SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade
Kuniamuthur, Coimbatore – 641008
Phone : (0422)-2678001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

Curriculum & Syllabi

Regulation 2022

2023-2027 Batch

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade
Kuniamuthur, Coimbatore – 641008
Phone : (0422)-2678001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Batch 2023-2027)

VISION OF THE INSTITUTION

- To Produce Globally Competitive Engineers with High Ethical Values and Social Responsibilities

MISSION OF THE INSTITUTION

- To impart the highest quality state-of-the-art technical education by providing impetus to innovation, research, and development and empowering students with entrepreneurship skills
- To instil ethical values, imbibe a sense of social responsibility, and strive for societal well-being
- To identify the needs of society and offer sustainable solutions through outreach programs

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VISION OF THE DEPARTMENT

- To prepare professionals with high technical, research and entrepreneurial skills as well as ethical values who will contribute to the computational world

MISSION OF THE DEPARTMENT

- To develop human resources with the ability and attitude to adapt to emerging technological changes through academic and research-oriented events
- To identify current socio, economic problems of national and international significance and provide solutions through competency centers
- To impart ethics, social responsibilities and necessary professional, entrepreneurial and leadership skills through student lead activities

I.	PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)
To enable graduates to	
PEO 1	Be successful in their career in industries associated with Computer Science and Engineering
PEO 2	Comprehend, analyze, design, and create novel products and solutions for the real-life problems
PEO 3	Possess professional and ethical attitude, effective communication skills, team working skills, multi-disciplinary approach, and an ability to relate engineering issues to broader social context
PEO 4	Exhibit leadership qualities and progress through life-long learning

II.	PROGRAMME OUTCOMES (POs)
PO 1	**Engineering Knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems
PO 2	**Problem Analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
PO 3	**Design/ Development of Solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
PO 4	**Conduct Investigations of Complex Problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
PO 5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
PO 6	**The Engineer and Society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
PO 7	**Environment and Sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
PO 8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
PO 9	**Individual and Team Work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
PO 10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions
PO 11	**Project Management and Finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as

	a member and leader in a team, to manage projects and in multidisciplinary environments
PO 12	**Life-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

III. PROGRAMME SPECIFIC OUTCOMES (PSOs)	
The Graduates of B.E – CSE programme will be able to:	
PSO 1	Apply the fundamental knowledge for problem solving and analysis as well as conduct investigations in computer science and engineering for sustainable development
PSO 2	Design and develop the solutions for real time problems and implement them by using modern software tools in lieu of deploying them in the society for its growth
PSO 3	Communicate effectively, adopt ethics and engage in life-long learning

IV.	**MAPPING OF PEOs WITH POs**											
PEO	**POs**											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
1	3	3	3	2	2	3	-	1	1	1	2	1
2	3	3	3	3	3	2	1	1	1	2	2	2
3	-	-	-	-	-	3	3	3	2	2	2	2
4	-	-	-	-	-	-	-	-	3	3	3	3
	1- low, 2 - medium, 3 - high, '-' - no correlation											

V. MAPPING OF PEOs WITH PSOs

	PSO 1	**PSO 2**	**PSO 3**
PEO 1	2	2	2
PEO 2	3	1	-
PEO 3	1	3	2
PEO 4	-	-	3

AUTONOMOUS CURRICULUM AND SYLLABI

Regulations 2022 (2023-2027 Batch)

SEMESTER I							
S.No	**Course Code**	**Course Title**	**L/T/P**	**Contact Hrs/Wk**	**Credits**	**Ext/Int**	**Cat.**
Theory (Internal 40 Marks & External 60 Marks)							
1	23MA101	Mathematics - I	3/1/0	4	4	60/40	BSC
2	23EC111	Digital Logic Design and Computer Architecture	3/1/0	4	4	60/40	ESC
3	23TA101	Heritage of Tamils / தமிழர் மரபு	1/0/0	1	1	60/40	HSMC
Theory with Practical (Internal 50 Marks & External 50 Marks)							
4	23IT101	Application Development Practices	1/0/4	5	3	50/50	ESC
5	23CS101	Problem Solving using C++	1/0/4	5	3	50/50	ESC
6	23EN101	Oral and Written Communication Skills	2/0/2	4	3	50/50	HSMC
Mandatory Course (Internal 100 Marks)							
7	23MC101	Mandatory Course – I (Induction Programme)	3 Weeks				MC
TOTAL				**23**	**18**	**700**	

SEMESTER II							
S.No	**Course Code**	**Course Title**	**L/T/P**	**Contact Hrs/Wk**	**Credits**	**Ext/Int**	**Cat.**
Theory (Internal 40 Marks & External 60 Marks)							
1	23MA201	Mathematics - II	3/1/0	4	4	60/40	BSC
2	23AS101	Applied Science	4/0/0	4	4	60/40	BSC
3	23TA201	Tamils and Technology / தமிழரும் தொழில்நுட்பமும்	1/0/0	1	1	60/40	HSMC
Theory with Practical (Internal 50 Marks & External 50 Marks)							
4	23CS201	Data Structures and Algorithms	1/0/4	5	3	50/50	PCC

5	23CD201	Database Management Systems	1/0/4	5	3	50/50	PCC
6	23CY201	Java Programming	1/0/4	5	3	50/50	PCC
7	23CD202	Object Oriented Analysis and Design	3/0/2	5	4	50/50	PCC
Practical (Internal 60 Marks & External 40 Marks)							
8	23AS102	Applied Science Laboratory	0/0/4	4	2	40/60	BSC
Mandatory Course (Internal 100 Marks)							
9	23MC102	Mandatory Course - II (Environmental Sciences)	1/0/0	1	0	0/100	MC
TOTAL				**34**	**24**	**900**	

SEMESTER III							
S.No	**Course Code**	**Course Title**	**L/T/P**	**Contact Hrs/Wk**	**Credits**	**Ext/Int**	**Cat.**
Theory (Internal 40 Marks & External 60 Marks)							
1	23MA301	Mathematical Foundation for Computer Science	3/1/0	4	4	60/40	BSC
Theory with Practical (Internal 50 Marks & External 50 Marks)							
2	23IT301	Web Technology using React	1/0/4	5	3	50/50	PCC
3	23CD301	Software Product Design	3/0/2	5	4	50/50	ESC
4	23CY202	Operating Systems	3/0/2	5	4	50/50	PCC
5	23CS301	Advanced Java Programming	1/0/4	5	3	50/50	PCC
6	23AD402	Design and Analysis of Algorithms	1/0/4	5	3	50/50	PCC
Mandatory Course (Internal 100 Marks)							
7	23MCXXX	Mandatory Course - III	1/0/0	1	0	0/100	MC
		TOTAL		**30**	**21**	**700**	

SEMESTER IV							
S.No	**Course Code**	**Course Title**	**L/T/P**	**Contact Hrs/Wk**	**Credits**	**Ext/Int**	**Cat.**
Theory (Internal 40 Marks & External 60 Marks)							
1	23GE301	Universal Human Values	3/0/0	3	3	60/40	HSMC
2	23CS401	Theory of Computation	3/1/0	4	4	60/40	PCC
3	23CS402	Algorithms of Internet	3/1/0	4	4	60/40	PCC
Theory with Practical (Internal 50 Marks & External 50 Marks)							
4	23CS403	Computer Networks	3/0/2	5	4	50/50	ESC
5	23AD403	Managing Cloud and Containerization	1/0/4	5	3	50/50	PCC
Practical (Internal 60 Marks & External 40 Marks)							
6	23IT402	Web Frameworks using Rest API	0/0/4	4	2	40/60	PCC
7	23ME305	Design Thinking and Idea Lab	0/0/2	2	1	40/60	HSMC
Mandatory Course (Internal 100 Marks)							
8	23MCXXX	Mandatory Course - IV	1/0/0	1	0	0/100	MC
		TOTAL		**28**	**21**	**800**	

SEMESTER V							
S.No	**Course Code**	**Course Title**	**L/T/P**	**Contact Hrs/Wk**	**Credits**	**Ext/Int**	**Cat.**
Theory (Internal 40 Marks & External 60 Marks)							
1	23IT501	Cryptography and Network Security	3/1/0	4	4	60/40	PCC
2	23CS501	Mobile and Edge Computing	3/0/0	3	3	60/40	PCC
Theory with Practical (Internal 50 Marks & External 50 Marks)							
3	23ITC01	Internet of Things	2/0/2	4	3	50/50	PCC
4	23CS502	Software Testing	1/0/4	5	3	50/50	PCC
5		Professional Elective - I	2/0/2	4	3	50/50	PEC
6		Professional Elective - II	2/0/2	4	3	50/50	PEC
Mini Project (Internal 100 Marks)							
7	23CS503	Application Development	0/0/6	6	3	0/100	PRJ
TOTAL				**30**	**22**	**700**	

SEMESTER VI							
S.No	**Course Code**	**Course Title**	**L/T/P**	**Contact Hrs/Wk**	**Credits**	**Ext/Int**	**Cat.**
Theory (Internal 40 Marks & External 60 Marks)							
1	23CSC03	Quantum Computing	3/0/0	3	3	60/40	PCC
Theory with Practical (Internal 50 Marks & External 50 Marks)							
2	23CSC04	Principles of Compiler Design	3/0/2	5	4	50/50	PCC
3	23CSC02	Machine Learning Techniques	2/0/2	4	3	50/50	PCC
4	23ADC01	Artificial Intelligence and Its Application	2/0/2	4	3	50/50	PCC
5		Professional Elective – III	2/0/2	4	3	50/50	PEC
6		Professional Elective - IV	2/0/2	4	3	50/50	PEC
Mini Project (Internal 100 Marks)							
7	23CS601	Capstone Project	0/0/6	6	3	0/100	PRJ
TOTAL				**30**	**22**	**700**	

SEMESTER VII							
S.No	**Course Code**	**Course Title**	**L/T/P**	**Contact Hrs/Wk**	**Credits**	**Ext/Int**	**Cat.**
Theory (Internal 40 Marks & External 60 Marks)							
1		Open / Emerging/ Industrial Elective- I	3/0/0	3	3	60/40	OEC
2		Open / Emerging/ Industrial Elective- II	3/0/0	3	3	60/40	OEC
3		Open / Emerging/ Industrial Elective- III	3/0/0	3	3	60/40	OEC
Theory with Practical (Internal 50 Marks & External 50 Marks)							
4		Professional Elective - V	2/0/2	4	3	50/50	PEC
5		Professional Elective - VI	2/0/2	4	3	50/50	PEC
Project (Internal 60 Marks & External 40 Marks)							
6	23CS701	Project - I	0/0/6	6	3	40/60	PRJ
Internship (Internal 100 Marks)							
7	23CS702	Employability Enhancement Skills (Internship)	28 Days		2	0/100	PRJ
TOTAL				**23**	**20**	**700**	

SEMESTER VIII							
S.No	**Course Code**	**Course Title**	**L/T/P**	**Contact Hrs/Wk**	**Credits**	**Ext/Int**	**Cat.**
Theory (Internal 40 Marks & External 60 Marks)							
1	23CS801	Project - II	0 / 0 / 24	24	12	60/40	PRJ
TOTAL				**24**	**12**	**100**	

SCHEME OF CREDIT DISTRIBUTION – SUMMARY											
S. No.	Stream	Credits/Semester								C	%
		I	II	III	IV	V	VI	VII	VIII		
1	Humanities & Social Sciences Including Management (HSMC)	4	1		4					9	5.6
2	Basic Sciences (BSC)	4	10	4						18	11.2
3	Engineering Sciences (ESC)	10		4	4					18	11.2
4	Professional Core (PCC)		13	13	13	13	13			65	41.6
5	Professional Electives (PEC)					6	6	6		18	11.2
6	Open/Emerging/ Industry (OEC)							9		9	5.6
7	Project Work (PROJ)					3	3	5	12	23	13.7
8	Mandatory Course (MC) / Spoken Hindi									Non-Credit	
Total		18	24	21	21	22	22	20	12	160	100

STRUCTURE FOR UNDERGRADUATE ENGINEERING PROGRAM			
S. No.	**Course Work - Subject Area**	**AICTE Suggested Credits**	**SKCET Credits (161)**
1	Humanities and Social Sciences (HS), including Management;	**16***	**9**
2	Basic Sciences (BS) including Mathematics, Physics, Chemistry, Biology;	**23***	**18**
3	Engineering Sciences (ES), including Materials, Workshop, Drawing, Basics of Electrical/Electronics/Mechanical/Computer Engineering, Instrumentation;	**29***	**18**
4	Professional Subjects-Core (PC), relevant to the chosen specialization/branch; (May be split into Hard (no choice) and Soft (with choice), if required	**59***	**65**
5	Professional Subjects – Electives (PE), relevant to the chosen specialization/ branch;	**12***	**18**
6	Open Subjects- Electives (OE), from other technical and/or emerging subject areas;	**9***	**9**
7	Project Work, Seminar and/or Internship in Industry or elsewhere.	**15***	**23**
8	Mandatory Courses (MC)	**Non-credit**	
Total		**163**	**160**
**Minor Variations is allowed as per need of the respective disciplines*			

HUMANITIES & SOCIAL SCIENCES INCLUDING MANAGEMENT (9 Credits)							
S.No.	**Course Code**	**Course Title**	**L/T/P**	**Contact hrs./Wk.**	**C**	**Ext/ Int**	**Cat.**
1	23EN101	Oral & Written Communication Skills	2/0/2	4	3	50/50	HSMC
2	23GE301	Universal Human Values	3/0/0	3	3	60/40	HSMC
3	23TA101	Heritage of Tamils	1/0/0	1	1	60/40	HSMC
4	23TA201	Tamils and Technology	1/0/0	1	1	60/40	HSMC
5	23ME305	Design Thinking and Idea Lab	0/0/2	2	1	40/60	HSMC

BASIC SCIENCE COURSES (18 Credits)							
S.No.	**Course Code**	**Course Title**	**L/T/P**	**Contact hrs./Wk.**	**C**	**Ext/ Int**	**Cat.**
1	23MA101	Mathematics I	3/1/0	4	4	60/40	BSC
2	23MA201	Mathematics II	3/1/0	4	4	60/40	BSC
3	23AS101	Applied Science	4/0/0	4	4	60/40	BSC
4	23AS102	Applied Science Laboratory	0/0/4	4	2	40/60	BSC
5	23MA301	Mathematical Foundation for Computer Science	3/1/0	4	4	60/40	BSC

ENGINEERING SCIENCE COURSES (18 Credits)							
S.No.	**Course Code**	**Course Title**	**L/T/P**	**Contact hrs./Wk.**	**C**	**Ext/ Int**	**Cat.**
1	23EC111	Digital Logic Design and Computer Architecture	3/1/0	4	4	60/40	ESC
2	23CY101	Networking and Communication	3/0/2	5	3	50/50	ESC
3	23IT101	Application Development Practices	1/0/4	5	3	50/50	ESC
4	23CS101	Problem Solving using C++	1/0/4	5	4	50/50	ESC
5	23CD301	Software Product Design	3/0/2	5	4	50/50	ESC

PROFESSIONAL CORE COURSES (67 Credits)							
S. No.	**Course Code**	**Course Title**	**L/T/P**	**Contact hrs./Wk.**	**C**	**Ext/ Int**	**Cat.**
1	23CS201	Data Structures and Algorithms	1/0/4	5	3	50/50	PCC
2	23CD201	Database Management Systems	1/0/4	5	3	50/50	PCC
3	23CY201	Java Programming	1/0/4	5	3	50/50	PCC
4	23CD202	Object Oriented Analysis and Design	3/0/2	5	4	50/50	PCC
5	23IT301	Web Technology using React	1/0/4	5	3	50/50	PCC
6	23CS301	Advanced Java Programming	1/0/4	5	3	50/50	PCC
7	23CY202	Operating Systems	3/0/2	5	4	50/50	PCC
8	23AD301	Design and Analysis of Algorithms	1/0/4	5	3	50/50	PCC
9	23IT402	Web Frameworks using Rest API	0/0/4	4	2	40/60	PCC
10	23AD403	Managing Cloud and Containerization	1/0/4	5	3	50/50	PCC
11	23CS402	Algorithms of Internet	3/1/0	4	4	60/40	PCC
12	23CS401	Theory of Computation	3/1/0	4	4	60/40	PCC
13	23IT501	Cryptography and Network Security	3/0/0	3	3	60/40	PCC
14	23CSC03	Quantum Computing	3/0/0	3	3	60/40	PCC
15	23CSC04	Principles of Compiler Design	3/0/2	5	4	50/50	PCC
16	23ITC01	Internet of Things	2/0/2	4	3	50/50	PCC
17	23CS502	Software Testing	1/0/4	5	3	50/50	PCC
18	23CS501	Mobile and Edge Computing	3/0/0	3	3	60/40	PCC
19	23ADC01	Artificial Intelligence and Its Application	2/0/2	4	3	50/50	PCC
20	23CSC02	Machine Learning Techniques	2/0/2	4	3	50/50	PCC

PROFESSIONAL ELECTIVE COURSES (18 Credits)							
S. No.	**Course Code**	**Course Title**	**L/T/P**	**Contact hrs./Wk.**	**C**	**Ext/ Int**	**Cat.**
ELECTIVE STREAM I – MACHINE LEARNING ENGINEERING							
1	23AD902	Exploratory Data Analysis (EDA) using Python	2/0/2	4	3	50/50	PEC
2	23AD905	Statistical Methods and Basic Machine Learning Models	2/0/2	4	3	50/50	PEC
3	23IT901	Advanced ML Techniques	2/0/2	4	3	50/50	PEC
4	23IT902	NLP	2/0/2	4	3	50/50	PEC
5	23IT903	Computer Vision	1/0/4	5	3	50/50	PEC
6	23AD906	GenAI Advanced Prompt Engineering & LLMs	1/0/4	5	3	50/50	PEC
ELECTIVE STREAM II - DATA ANALYST WITH ML ESSENTIALS							
1	23AD901	Data Storytelling and Visualization	2/0/2	4	3	50/50	PEC
2	23AD902	Exploratory Data Analysis (EDA) using Python	2/0/2	4	3	50/50	PEC
3	23AD903	Problem solving with Analytical & Design Thinking	2/0/2	4	3	50/50	PEC
4	23AD904	PowerBI	2/0/2	4	3	50/50	PEC
5	23AD905	Statistical Methods and Machine Learning Models	2/0/2	4	3	50/50	PEC
6	23AD906	GenAI Advanced Prompt Engineering & LLMs	1/0/4	5	3	50/50	PEC
ELECTIVE STREAM III - CLOUD IT ADMINISTRATION							
1	23CS901	Implementing and Administering Enterprise Networks	2/0/2	4	3	50/50	PEC
2	23CS902	Linux System Administration	2/0/2	4	3	50/50	PEC
3	23CS903	Information Security Systems	2/0/2	4	3	50/50	PEC
4	23CS904	Low-Code No-Code Application Building	2/0/2	4	3	50/50	PEC
5	23CS905	Virtualization, Cloud Computing and SysOps	1/0/4	5	3	50/50	PEC

6	23CS906	Continuous Monitoring and Observability (AWS)	1/0/4	5	3	50/50	PEC
ELECTIVE STREAM IV – CYBER SECURITY							
1	23CS901	Implementing and Administering Enterprise Networks	2/0/2	4	3	50/50	PEC
2	23CS902	Linux System Administration	2/0/2	4	3	50/50	PEC
3	23CS903	Information Security Systems	2/0/2	4	3	50/50	PEC
4	23CY901	Cloud Computing and Containerized Virtual Infrastructure	2/0/2	4	3	50/50	PEC
5	23CY902	Penetration Testing	1/0/4	5	3	50/50	PEC
6	23CY903	Security Operations of Information systems	1/0/4	5	3	50/50	PEC
ELECTIVE STREAM V – FULL STACK SOFTWARE DEVELOPMENT (for Minor)							
1	23CS911	Managing and Querying Database (RDBMS) MySQL / Postgre SQL	2/0/2	4	3	50/50	PEC
2	23CS912	Java / Python: Object-Oriented Programming	2/0/2	4	3	50/50	PEC
3	23IT911	Web Development and UI/UX Essentials	2/0/2	4	3	50/50	PEC
4	23IT912	Build Single-Page Applications using React	2/0/2	4	3	50/50	PEC
5	23IT913	Build Back-end Application using Spring Boot / FAST API	1/0/4	5	3	50/50	PEC
6	23AD906	GenAI Advanced Prompt Engineering & LLMs	1/0/4	5	3	50/50	PEC

OPEN/ EMERGING/ INDUSTRY					
S. No.	**Course Code**	**Course Title**	**L/T/P**	**Contact hrs./Wk.**	**C**
1.	23CS001	Programming with Data Structures	2/ 0/ 2	4	3

PROJECT WORK (23 Credits)							
S. No.	**Course Code**	**Course Title**	**L/T/P**	**Contact hrs./Wk.**	**C**	**Ext/ Int**	**Cat.**
1	23CS503	Application Development	0 / 0 / 6	6	3	0/100	PRJ
2	23CS603	Capstone Project	0 / 0 /6	6	3	0/100	PRJ
3	23CS701	Project - I	0 / 0 /6	6	3	40/60	PRJ
4	23CS702	Employability Enhancement Skills (Internship)	28 Days		2	0/100	PRJ
5	23CS801	Project - II	0 / 0 / 24	24	12	60/40	PRJ

PROFESSIONAL ELECTIVE COURSES: VERTICALS				
Vertical I - ML Engineering	**Vertical II - Data Analyst with ML Essentials**	**Vertical III - Cloud IT Administration**	**Vertical IV - Cybersecurity**	**Vertical V – Full Stack Software Development (Minor)**
Exploratory Data Analysis (EDA) using Python	Data Storytelling and Visualization	Implementing and Administering Enterprise Networks	Implementing and Administering Enterprise Networks	Managing and Querying Database (RDBMS) MySQL / Postgre SQL
Statistical Methods and Basic Machine Learning Models	Exploratory Data Analysis (EDA) using Python	Linux System Administration	Linux System Administration	Java / Python: Object-Oriented Programming
Advanced ML Techniques	Problem solving with Analytical & Design Thinking	Information Security Systems	Information Security Systems	Web Development and UI/UX Essentials
NLP	PowerBI	Low-Code No-Code Application Building	Cloud Computing and Containerized Virtual Infrastructure	Build Single-Page Applications using React
Computer Vision	Statistical Methods and Machine Learning Models	Virtualization, Cloud Computing and SysOps	Penetration Testing	Build Back-end Application using Spring Boot / FAST API
GenAI Advanced Prompt Engineering & LLMs	GenAI Advanced Prompt Engineering & LLMs	Continuous Monitoring and Observability (AWS)	Security Operations of Information systems	GenAI Advanced Prompt Engineering & LLMs

INTERN (2 Credits)							
S. No.	**Course Code**	**Course Title**	**L/T/P**	**Contact hrs./Wk.**	**C**	**Ext/ Int**	**Cat.**
1	23CS702	Employability Enhancement Skills (Internship)	28 Days		2	0/100	PRJ

VALUE ADDED COURSES (Based on student's interest)						
S. No.	**Course Code**	**Course Title**	**L/T/P**	**Contact hrs./Wk.**	**C**	**Sem**
1	23VA200	Cassandra	1 / 0 / 0	1	1	III
2	23VA201	MongoDB	1 / 0 / 0	1	1	IV
3	23VA202	Edge Computing	1 / 0 / 0	1	1	V
4	23VA203	PyCharm	1 / 0 / 0	1	1	VI
5	23VA204	Generative AI	1 / 0 / 0	1	1	VI

MANDATORY COURSES (01) (Courses conducted either by internal faculty or through MOOCs)					
S. No.	**Course Code**	**Course Title**	**L/T/P**	**Contact hrs./Wk.**	**C**
1	23MC101	Induction Programme			0
2	23MC102	Environmental Sciences	1 / 0 / 0	1	0
3	23MC103	Soft Skills	1 / 0 / 0	1	0
4	23MC104	Stress Management	1 / 0 / 0	1	0

L: Lecture **T:** Tutorial **P:** Practical **C:** Credit **O:** Outside Class hours **Cat.:** Category

HSMC : Humanities and Social Sciences including Management
BSC : Basic Science Courses
ESC : Engineering Science Courses
PCC : Professional Core Courses
PEC : Professional Elective Courses
OEC : Open and Emerging Elective Courses
PRJ : Project Work
INT : Internship
MC : Mandatory Course

Definition of Credit:

L – Lecture	1 Hr. Lecture (L) per week	1 credit
T – Tutorial	1 Hr. Tutorial (T) per week	1 credit
P - Practical/Practice (Project and Industry based Courses)	1 Hr. Practical (P) per week	0.5 credit

SEMESTER – V

23IT501	CRYPTOGRAPHY AND NETWORK SECURITY	3/1/0/4
Nature of Course:	G (Theory Analytical)	
Pre requisites:	Networking and Communication / Computer Networks	

Course Objectives:

1	To outline the security goals of cryptography
2	To recognize various modern cryptographic techniques and their applications
3	To illustrate public key encryption and hash functions
4	To impart understanding on digital signatures and authentication applications

Course Outcomes:
Upon completion of the course, students shall have ability to

C501.1	Interpret the OSI security architecture and classical encryption techniques.	[AP]
C501.2	Apply the Symmetric and Asymmetric Cryptographic algorithms in real-time examples	[AP]
C501.3	Examine the applications of Cryptographic Hash Functions and Message Authentication Codes	[AP]
C501.4	Develop a model for Digital signature system and authentication system	[AP]
C501.5	Apply techniques to enhance the security in different applications and networks	[AP]

INTRODUCTION **20**

Concepts of Cyber security– CIA Triad – OSI Security Architecture (attacks, services, mechanisms)- Cryptography - Network Security – Classical Encryption techniques - Symmetric ciphers - Substitution Techniques - Transposition Techniques- Data Encryption Standard – DES example - The Strength of DES - Block Cipher Design Principles -Advanced Encryption Standard, AES Structure-AES Transformation Functions- AES Key Expansion – AES Example- Multiple Encryption and Triple DES

PUBLIC-KEY ENCRYPTION AND HASH FUNCTIONS **20**

Fermat's and Euler's Theorem - Testing of primality -The Chinese remainder theorem - Public Key Cryptography: RSA- The RSA Algorithm- Diffie-Hellman (ElGamal) - Cryptographic Hash Functions - Applications of Cryptographic Hash Functions -Two Simple Hash Functions - Secure Hash Algorithm (SHA)- SHA 3 - Message Authentication Codes – Requirements – Functions - MACs Based on Hash Functions: HMAC.

NETWORK SECURITY APPLICATIONS **20**

Digital Signatures: Introduction -ElGamal/Schnorr Digital Signature Scheme
Authentication Applications: Remote User-Authentication Principles - Kerberos - Transport-Level Security: Web Security Considerations - Transport Layer Security – HTTPS - Secure Shell (SSH)- Wireless Network Security: Wireless Security - Mobile Device Security- Network Endpoint Security: Firewalls - Intrusion Detection Systems - Malicious Software - Distributed Denial of Service Attacks
Case Study: Hardening CISCO Devices based on Cryptography and Security Protocols.

TOTAL PERIODS:	**60**

Text Books:

1	William Stallings, "Cryptography and Network Security - Principles and Practice", 8th Edition, Pearson, 2023.

2	Behrouz A. Forouzon, “Cryptography and Network security”, 3rd Edition, Tata McGraw Hill, 2015.
3	Atul Kahate, “Cryptography and Network Security”, 3rd Edition, Tata McGraw-Hill, 2017.
Reference Books:	
1	Bernard L Menezes, and Ravinder Kumar “Cryptography, Network Security and Cyber Laws”, Cengage Learning India Pvt Limited, 2018.
2	Charlie Kaufman, Radia Perlman, Mike Speciner, “Network Security: Private Communication in a Public World”, 2nd Edition, Pearson Education, 2011
3	Bruce Schneier, “Applied Cryptography: Protocols, Algorithms, and Source Code in C”, 2nd Edition, Wiley, 2015.
Web References:	
1	https://crypto.stanford.edu/~dabo/cs255/syllabus.html
2	http://www.tutorialspoint.com/cryptography/
3	https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3523710
Online Resources:	
1	https://www.coursera.org/learn/crypto
2	https://www.edx.org/learn/network-security/rochester-institute-of-technology-network-security?authuser=0
3	https://onlinecourses.nptel.ac.in/noc22_cs90/preview

23CS501	MOBILE AND EDGE COMPUTING	3/0/0/3

Nature of the Course: D (Theory application)

Pre-requisite(s): Computer Networks

Course Objectives:

1	To provide the students with an understanding of the cellular concept frequency reuse, handoff strategies.
2	To infer the concept of adhoc networks and Wireless Application Protocol in design of wireless networks.
3	To explore the design considerations, scheduling algorithms, task synchronization, communication mechanisms, memory management
4	To gain solid understanding of edge computing and its role in the emerging field of cyber-physical systems.

Course Outcomes:
Upon completion of the course, students shall have ability to

CO1	Summarize the fundamentals of wireless communication and determine the suitable medium access control techniques	U
CO2	Elaborate the concepts and protocol architectures of GSM and satellite systems	AN
CO3	Develop a Adhoc network model using MANET and WAP	AP
CO4	Design edge computing solutions, including architectures, models, and platforms	AP
CO5	Develop knowledge of resource management techniques in edge computing, including task scheduling algorithms, resource allocation algorithms, and load balancing algorithms.	AP

INTRODUCTION TO MOBILE COMPUTING **15**

Wireless transmission –Frequencies for radio transmission –Signals –Antennas –Signal Propagation – Multiplexing –Spread spectrum –cellular systems-MAC-Motivation –SDMA –FDMA –TDMA –CDMA, GSM: Mobile services -System architecture -Radio interface -Protocols -Localization and calling – Handover

MOBILE NETWORK AND TRANSPORT LAYER **15**

MANET: Characteristics, Classifications, Routing-proactive and reactive. WAP-Architecture, Mobile IP: Goals, assumptions and requirements – Entities and terminologies – IP packet delivery – Agent discovery – Registration – Tunneling and Encapsulation – Dynamic Host Configuration Protocol-Mobile ad-hoc Networks –Improvement on TCP for mobile and wireless network

EDGE COMPUTING **15**

Overview of edge computing and its significance in distributed systems. Edge computing architectures, models, and platforms. Comparison of edge computing with cloud computing and fog computing. Resource management in edge computing and its challenges. Resource management techniques for edge computing, including task scheduling algorithms, resource allocation algorithms, and load balancing algorithms.

TOTAL PERIODS	**45**

Text Books:

1	Schiller J., “Mobile Communication”, 2nd Edition, Pearson Education, New Delhi, 2014.
2	Anitha Kumari, G. Sudha Sadasivam, D. Dharani and M. Niranjanamurthy,“Edge Computing Fundamentals, Advances and Applications”, CRC Press, 2022.

Reference Books:	
1	Raj Kamal, “Mobile Computing”, 3rd edition, Oxford University PressInc. New Delhi, 2019
2	Rajkumar Buyya and Satish Narayana Srirama, “Fog and Edge Computing Principles and Paradigms”, John Wiley & Sons, Inc.
Web References:	
1	https://fs.wp.odu.edu/wp-content/uploads/sites/1756/2016/11/IT401-Course-Syllabus.pdf
2	https://www.redhat.com/en/topics/edge-computing
3	https://archive.nptel.ac.in/courses/106/106/106106147/
4	https://onlinecourses.nptel.ac.in/noc24_cs66/preview

23ITC01	INTERNET OF THINGS	2/0/2/3
Nature of Course	F (Theory Programming)	
Prerequisites	-	
Course Objectives:		
1	To understand the concepts of the Internet of Things (IoT) and its design methodologies.	
2	To summarize the architecture of the Internet of Things (IoT).	
3	To familiarise with the design constraints and challenges of applications.	
4	To design basic solutions using embedded platforms for IoT.	
5	To assess suitable cloud solutions for IoT systems.	
Course Outcomes		
Upon completion of the course, students shall have ability to		
CO1	Understand the core principles and architecture of IoT systems to enable the development of IoT-based solutions.	[U]
CO2	Interpret design constraints and challenges in developing IoT systems.	[AN]
CO3	Develop cost-effective IoT solutions and interface with embedded platforms like Arduino, Raspberry Pi, or similar boards.	[AP]
CO4	Analyze cloud platforms for deploying, managing, and ensuring the security of IoT solutions.	[AN]
CO5	Apply data analytics to IoT data using Hadoop and MapReduce, demonstrating effective IoT design and implementation.	[AP]

INTRODUCTION TO INTERNET OF THINGS AND IOT DESIGN METHODOLOGIES **15**

Characteristics of IoT, Physical Design of IoT, IoT Protocols, IoT Reference Architecture, Logical Design of IoT, IoT Levels & Deployment Templates, Domain Specific IoT, IoT Vs M2M Vs IoE, IoT Design Methodologies.

SYSTEM HARDWARE FOR IOT **5**

Sensors and Actuators – Hardware Kits: Arduino, Node MCU, Raspberry Pi. **Arduino UNO:** Physical Design – Interfaces – Arduino IDE – Arduino Programming with examples: Digital IO – Analog IO – Serial Communication – Condition and Looping statements. **Raspberry Pi**: Physical Design – Interfaces – Pi programming using Python with examples – Python Packages for IoT.

CLOUD AND DATA ANALYTICS FOR IOT **10**

Cloud: Types of Cloud - IoT with Cloud challenges - Selection of cloud for IoT applications - Fog and Edge computing for IoT - Cloud security aspects for IoT applications. **Data Analytics for IoT:** Overview of Hadoop ecosystem – MapReduce architecture –
MapReduce Job Execution – MapReduce Schedulers.
Case study: Internet of Medical Things (IoMT)

TOTAL (THEORY): 30 PERIODS

List of Experiment

1. Study and Configuration of Arduino kit / Node MCU / Raspberry PI.

2. Basic Programming using Arduino / Raspberry PI: a. LED and Switch Interface b. Analog & Digital Sensor Interface c. Serial Communication d. Local display of sensor data using LCD e. Display of Sensor values in Mobile handset using Bluetooth 3. Basic Programming using NodeMCU. a. Remote control of Electrical appliances using Mobile handset and Wi-Fi b. Local Web server using NodeMCU and displaying Sensor values. 4. Design and development of a System using LM35 temperature sensor. 5. Design and development of a System using MQ5 sensor. 6. Design and development of a System using Soil Moisture sensor. 7. Design and development of a System using PIR sensor. 8. Design and development of a System using Heart beat sensor.	
TOTAL (Lab) PERIODS: 30	
TOTAL PERIODS: 60	
Text Books:	
1	Arshdeep Bahga and Vijay Madisetti, “Internet of Things: A Hands-on Approach”, 1st Edition, Universities Press, 2015.
2	Hanes David, Salgueiro Gonzalo, Grossetete Patrick, Barton Rob, Henry Jerome, “IoT Fundamentals: Networking Technologies, Protocols and Use Cases for the Internet of Things”, 1st Edition, Pearson Education, 2017.
3	Mark Torvalds, “Arduino Programming: Step-by-step guide to mastering Arduino hardware and software”, 2nd Edition, 2018.
Reference Books:	
1	Srinivasa K. G.,Siddesh G. M., Hanumantha Raju R., “Internet of Things”, 1st Edition, Cengage Learning India Pvt. Ltd., 2018.
2	Raj Kamal, “Internet of Things”, 1st Edition, McGraw Hill Education, 2017.
3	Dr. Simon Monk, “Programming the Raspberry Pi: Getting Started with Python”, 2nd Edition, McGraw-Hill Education, 2016.
4	Arshdeep bahga, Vijay madisetti, “Internet of Things A Hands-On Approach”, 1st Edition, Paperback, 2023.
5	Blum, “Arduino Programming”, 1st Edition, Pearson Education, 2020.
Web References:	
1	https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT
2	https://it.uw.edu/guides/internet-of-things-iot/iot-resources/
3	https://docs.arduino.cc/tutorials/
4	https://www.hackster.io/iot
5	https://www.coursera.org/learn/introduction-iot-boards
6	https://www.edx.org/learn/big-data/curtin-university-iot-programming-and-big-data
7	https://onlinecourses.nptel.ac.in/noc22_cs53/preview

23CS502	SOFTWARE TESTING	1/0/4/3
Nature of Course	F (Theory Programming)	
Pre requisites	-	
Course Objectives:		
1	To provide students with an understanding of Core Testing concept.	
2	To learn the functional and non-functional testing.	
3	To understand the different types of User Acceptance testing and end-to-end testing.	
4	To get familiarize with the best practices of Testing.	
Course Outcomes		
Upon completion of the course, students shall have ability to		
C502.1	Plan and apply the appropriate level of testing within the context of a software development application to the satisfaction of its beneficiaries.	[AP]
C502.2	Analyze specific and measurable test cases to ensure coverage and traceability to requirements	[A]
C502.3	Understand the problem of reporting techniques, metrics, and testing status reports and communicate testing results to colleagues, managers, and end users.	[U]
C502.4	Apply testing models, processes and practices appropriate for the software development lifecycle model of a project	[AP]
C502.5	Apply principles and practices of test-driven development to improve testing quality and reduce delivery times	[AP]
C502.6	Inspect the various testing processes towards the continuous delivery of a software product.	[A]

INTRODUCTION TO AUTOMATION TESTING WITH SELENIUM **5**

What is Software Testing, Why Software Testing, Benefits of Software Testing, Software Test Levels, Unit Testing, Integration Testing, System Testing, Acceptance Testing, Software Test Types, Functional testing, Non-functional testing, Change Related Testing. Test Scenario Design - Functional and non-functional test scenarios, identify and write business critical scenarios. Test Case Design - pre-requisites, test steps and expected results for test cases, Positive and negative testcases for each scenario, Test case prioritization, Test case optimization technique. RTM, DSR. Classes and Objects, Inheritance, and Polymorphism, Exception Handling, Collections, and, Collections (List), JDBC Connectivity, Creating CRUD OPERATION JDBC Connectivity

WORKING WITH SELENIUM **5**

Selenium webdriver - Maven Configuration, WebDriver Commands, Navigation Command, Selenium locators - Selenium Locators Basics (id, name),Xpath and css locators, Selenium WebElement - Handling of Form Elements, Synchronization Methods, Selenium Exceptions, Keyboard and mouse handling, Alert handling, Iframe Handling in Selenium, Java Script Execution, Handling WebTable and calendar. Introduction to TestNG - TestNg Introduction, Advantage of testNg , testNG Annotations, Test data preparation, Generation of TestNG Reports, Implicit wait and Explicit wait.

TESTING FRAMEWORK **5**

Testing Frameworks - Data driven testing using Apache POI, POM. Extent Reports - HTML Report Generation using Extent Reports, Attaching Screenshot in HTML Report.

Log4j - configuring log4j Property files, Log4j - parameters for Properties file, Log levels and logging using log4j, Hybrid framework implementation., Creating the POM with a Hybrid framework folder structure, Implementing the Hybrid framework in POM.	
TOTAL (Theory) PERIODS	**15**
Lab Component:	
1	Develop a program to automate the login process for a specified webpage using Selenium.
2	Write an automation script using Selenium to handle form elements on a given website.
3	Create an automation script with Selenium to interact with specific web elements on a designated webpage.
4	Implement automation scripts using TestNG, prioritizing different test cases for efficient testing on a given website.
5	Develop automation scripts with TestNG, incorporating seven levels of logging for detailed analysis while testing a specified website.
6	Execute application tests using designed test cases and generate an HTML report for a comprehensive overview.
7	Design and implement a hybrid framework for a ticket booking system, along with associated test cases.
8	Develop a hybrid framework and associated test cases for a hotel room booking system.
9	Design and develop a hybrid framework and relevant test cases for a hospital appointment application.
10	Implement a hybrid framework and associated test cases for an e-commerce application.
11	Develop a hybrid framework and design test cases for comprehensive testing of an insurance website.
TOTAL (Theory) PERIODS	**60**
TOTAL PERIODS	**75**
Text Books:	
1	Rex Allen Jones II, “Absolute Beginner, Part 1 Selenium Webdriver for Functional Automation Testing”, 1st Edition, Createspace Independent Pub, 2016
2	S Basu, “Selenium with Python Simplified for Beginners”, 1st Edition, 2020
3	Paul Watson, “Selenium webdriver with Node.js: Beginner's Guide”, 1st Edition, CreateSpace Independent Publishing Platform, 2016.
Reference Books:	
1	Satya Avasarala, “Selenium Web Driver Practical Guide”, 1stEdition, Packt Publishing Limited, 2014
2	Sujay Raghavendra, "Python Testing with Selenium: Learn to Implement Different Testing Techniques Using the Selenium WebDriver", 1stEdition, Apress, 2020.
3	Pinakin Ashok Chaubal, “Selenium Framework Design in Keyword-Driven Testing: Automate Your Test Using Selenium”, 1stEdition, BPB Publications, 2020.
Web References:	
1	https://www.coursera.org/projects/building-test-automation-framework-using-selenium-and-testng
2	https://www.edx.org/professional-certificate/delftx-automated-software-testing

3	https://onlinecourses.nptel.ac.in/noc22_cs12/preview
4	https://www.nextgenerationautomation.com/post/selenium-coding-exercises
5	https://www.studytonight.com/maven/build-and-test-maven-project
6	https://www.tutorialspoint.com/selenium-for-software-testing-getting-started/index.asp

23CS503	APPLICATION DEVELOPMENT	0/0/6/3
Nature of Course	M (Practical Application)	
Prerequisites	-	
Course Objectives:		
1	To discuss the essence of front-end development skills	
2	To impart the knowledge of React components used in Spring boot development platforms	
3	Ability to understand and use Setup Cloud API	
4	To deploy and test the React App used in Spring Boot	
5	To learn the Spring Cloud concepts using Docker	
Course Outcomes		
Upon completion of the course, students shall have ability to		
CO1	Identify the basic concepts and design issues of React.	[U]
CO2	Understand the principles of process and Spring boot.	[U]
CO3	Illustrate the approaches in scheduling and Spring Cloud to apply in real world problems.	[AP]
CO4	Apply concepts of Micro services Communication to the issues that occur in Real time applications.	[AP]
CO5	Identify issues related to Docker, API Gateway.	[AP]
CO6	Examine common React, Availability and Scalability.	[A]

REACT INTRODUCTION

Components, Routes, State, Props, hooks, Higher Order Functions, Axios and Services, Ant Design. Redux: Core Concept, Data Flow, Store, Actions, Pure function, Reducers, Devtools, Middleware, Webpack, Redux Integration. Spring boot: Annotations, Beans, Configuration, HTTP Methods, Crud, Postman Overview. Spring Security: Authentication, Authorization, Security Implementation. Configure Security, Authentication Manager, HTTP Security, Circular Reference Error. JWT Implementation: JWT Overview, JWT Libraries, Helper Methods, Token Generation and Validation, Implementing JWT Authorization, Filter. OAUTH Implementation: Introduction, Sample flow, Authorization code grant type flow, Implicit grant flow, Password Grant Type flow, Client, Credential Grand type flow, Refresh token Grand type flow, Validating token, Oauth2 integration with Spring Security. Building Micro services: Monolith Architecture and Challenges of Monolith Architecture, What is Micro services & How It Solves the Challenges of Monolith Architecture, Micro services Architecture Benefits and Best Practices, Understanding Spring Cloud and It's Important Modules, Micro service Applications and It's Port Mapping.

MICROSERVISES COMMUNICATION OVERVIEW

Micro services Communication using Rest Template, Micro services Communication using Web Client, Micro services Communication using Spring Cloud Open Feign - Understanding service Registry – Spring Cloud Netflix Eureka Server Implementation, Update on Using Spring Boot 3 Version, Register Micro service as Eureka Client, Update on using Spring Boot 3 Version, Register Micro service as Eureka Client, Running Multiple Instances of Micro service, Load Balancing with Eureka, Open Feign and Spring Cloud Load Balancer API gateway using Spring Cloud gateway: Understanding API Gateway - Create and Set up API Gateway Micro service, Update on Using Spring Boot 3 Version,

Register API-Gateway as Eureka Client to Eureka Server, Configuring API Gateway Routes and Test using Postman Client, Using Spring Cloud Gateway to Automatically Create Rout.

CENTRALIZED CONFIGURATIONS USING SPRING CLOUD CONFIG SERVER

How to Use Spring Cloud Config Server, Create and Setup Spring Cloud Config Server Project in IntelliJ IDEA, Update on Using Spring Boot 3 Version, Register Config-Server as Eureka Client, Set up Git Location for Config Server, Refactor Department-Service to use Config Server, Refactor Employee-Service to use Config Server, Refresh Use case - No Restart Required After Config Changes, REACT Frontend Micro service: Create React App using Create React App Tool, Adding Bootstrap in React Using NPM, Write HTTP Client Code to Connect React App with API-Gateway (REST API Call), Create a React Component and Integrate with API Gateway Microservice, RabbitMQ Core Concepts: RabbitMQ Architecture, Install and Setup RabbitMQ using Docker, Explore RabbitMQ using RabbitMQ Management UI, Create and Setup Spring Boot 3 Project in IntelliJ, Connection Between Spring Boot and RabbitMQ, Configure RabbitMQ in Spring Boot Application, Create RabbitMQ Producer, Create REST API to Send Message, Create RabbitMQ Consumer, Configure RabbitMQ for JSON Message Communication, Create RabbitMQ Producer to Produce JSON Message, Create REST API to Send JSON Object, Create RabbitMQ Consumer to Consume JSON Message, Dockering Spriing boot App : Install Docker Desktop, General Docker Workflow, Create Spring Boot Project and Build Simple REST API, Create Docker file to Build Docker Image, Build Docker Image from Dockerfile, Run Docker Image in a Docker Container, Push Docker Image to Docker Hub, Pulll Docker Image from DockerHub.

COURSE GUIDELINES:

1. Students choose a project topic from a list of approved options or propose their own idea from the area specified in the content and Faculty Coordinator/guide approval required for student-proposed projects
2. Number of students in the project team should be maximum of 4 and Every student shall have a project guide.
3. The entire semester shall be utilized by the students to do their project work by receiving the directions from the project guide.
4. Teams should submit a project proposal, including objectives, scope, timeline, and resources and Faculty Coordinator/guide reviews and approves the proposal.
5. Students should choose projects in line with the Departmental Mission, Vision, and Program Outcomes.
6. Teams should work on their projects, following the approved plan and Regular meetings with faculty advisors for progress updates and guidance.
7. Students should attend periodic reviews to present the progress of the project to faculty and peers' team and Evaluation is based on project outcomes, presentation quality, and teamwork.
8. Teams submit a final project report, including results, conclusions, and recommendations as specified in the guidelines issued by the COE.
9. Students should not be involved in unethical behavior, such as plagiarism, copyright violations, etc while working on projects and when submitting project reports.
10. Every student team will be required to prepare and submit two (2) copies plus (no. of students) copies of the Project report of typical length 30 – 60 pages (excluding Appendices).
11. A final external project viva-voce examination will be conducted to evaluate the student's Individual and team performance based on project outcomes, presentations, reports, and teamwork by an Internal and External Examiners.

TOTAL PERIODS: 60

Text Books:	
1	Taze, M. "Engineer's Survival Guide: Advice, Tactics, and Tricks After a Decade of Working at Facebook, Snapchat", 1st edition, Microsoft Paperback, 2021.
2	Blokdyk, G. "Secure Microservices: A Complete Guide", 1st edition, Paperback, 2021.
3	King, T. H. "AWS: The Ultimate Guide from Beginners to Advanced for the Amazon Web Services", 2020 edition, Paperback – Import, 2019.
Reference Books:	
1	Zacker, C. "Exam Ref PL-900: Microsoft Power Platform", 1st edition, Paperback, 2021.
Web References:	
1	https://awscloud.in/

SEMESTER – VI

23CSC03	Quantum Computing	3/0/0/3

Nature of the Course: D (Theory Application)

Pre-requisite(s): -

Course Objectives:

1	To introduce the core concepts of classical and quantum information and computation.
2	To familiarize students with the fundamental principles of quantum mechanics applied in quantum computing.
3	To explore quantum states, qubit representations, and essential quantum gates.
4	To understand the design and implementation of quantum circuits and operations.
5	To provide introductory knowledge of quantum algorithms and their applications in solving computational problems

Course Outcomes:
Upon completion of the course, students shall have ability to

CO1	Understand the fundamentals of quantum mechanics and represent quantum states	[U]
CO2	Apply quantum postulates to describe and manipulate single and multi-qubit systems.	[AP]
CO3	Construct quantum circuits using quantum gates.	[AP]
CO4	Analyze entanglement and quantum communication protocols, such as superdense coding and teleportation.	[AN]
CO5	Apply basic quantum algorithms, including Deutschs' and Simon's algorithms, to solve computational problems	[AP]
CO6	Apply Quantum Fourier Transform (QFT) in period-finding and explore its role in Shor's factoring algorithm.	[AP]

INTRODUCTION TO QUANTUM MECHANICS **15**

History of quantum computation and quantum Information- The Circuit Model of Computation – Review of Complex numbers- Vector spaces- -Linear Algebra and Dirac Notation: Dual vectors -Operators-Inner product -outer product- Quantum postulates-Quantum Measurement-Quantum Bits: single Qubits& multiple qubits Diarc and matrix representation- tensor products - Bloch sphere: Representation of a different basis

QANTUM GATES AND QUANTUM CIRCUITS **15**

Classical reversible gates and circuits- Single qubit gates and its operation: Pauli gates, Hadamard gate, Rotational gates – superpositions- Multi qubits gates and its operation: CNOT, CCNOT, Swap and Controlled U Gates. Entanglement- Bell state circuits- Quantum Series and Parallel circuits- super dense coding- Quantum Teleportation

QUANTUM ALGORITHMS **15**

Introduction to quantum algorithms- Phase kick back- oracle function- Deutsch's Algorithm - Deutsch-Jozsa Algorithm – Simon's Algorithm – Quantum Fourier transform- period findings using QFT- Grover's search algorithm-Shor's factoring algorithm - Case study: Quantum Cryptanalysis of RSA and Symmetric Encryption Using Shor's and Grover's Algorithms

TOTAL PERIODS **45**

Text Books:	
1	Lala, P. K. "Quantum Computing: A Beginner's Introduction", McGraw-Hill, 2020.
2	Nielsen, A., Chuang, I. L. "Quantum Computation and Quantum Information", 10th edition, Cambridge University Press, 2010.
3	Wong, T. G. "Introduction to Classical and Quantum Computing",1st edition, Rooted Grove, Omaha, Nebrask..
Reference Books:	
1	McMahon, D. "Quantum Computing Explained", 1st edition, Wiley-Interscience, John Wiley & Sons, 2008.
2	Griffiths, D. J. "Introduction to Quantum Mechanics", 2nd edition, Prentice Hall, New Jersey, 1995.
3	Kaye, P., Laflamme, R., Mosca, M. "An Introduction to Quantum Computing", 1st edition, Oxford University Press, New York, 2007.
4	Yanofsky, N. S., Mannucci, M. A. "Quantum Computing for Computer Scientists", 1st edition, Cambridge University Press, 2008.
Web References:	
1	https://learning.quantum.ibm.com/course/basics-of-quantum-information
2	https://learning.quantum.ibm.com/course/fundamentals-of-quantum-algorithms
3	https://nptel.ac.in/courses/106106232
4	https://youtube.com/playlist?list=PLxhaPrr4aQ9lnBEOoy7r6KNlrTG_obLgr&si=ABI4QGsRJ6G7-eVB

23CSC04	PRINCIPLES OF COMPILER DESIGN	3/0/2/4
Nature of Course	H (Theory Technology)	
Prerequisites	-	

Course Objectives:

1	To teach students with the fundamental principles of language translation and compiler design.
2	To train students to apply theoretical concepts in the development of a lexical analyzer and parser.
3	To introduce student wih the various code generation techniques to translate high-level language constructs into efficient machine code.
4	To enable students to explore code optimization strategies and their impact on performance and runtime environments.
5	To equip students with hands-on experience in constructing a functional compiler using LEX and YACC tools.

Course Outcomes:

Upon completion of the course, students shall have ability to:

C202.1	Develop a program to scan source code and identify tokens such as keywords, identifiers, literals, and symbols.	[AP]
C202.2	Implement parsing techniques such as LL(1) and LR parsing to generate parse trees.	[AP]
C202.3	Construct annotated parse trees for language constructs and Implement symbol tables and activation records for scope/access control..	[AP]
C202.4	Design a frontend (analysis phase) and backend (synthesis phase) components of a compiler.	[AP]
C202.5	Compare optimization strategies using empirical profiling and select a method to enhance program efficiency.	[AN]

LEXICAL AND SYNTAX ANALYSIS **15**

Phases of a compiler – Lexical Analysis: Role of Lexical Analyzer – Input Buffering – Specification of Tokens – Recognition of Tokens. Finite Automata – Regular expression to Automata – Minimization of DFA.Syntax Analysis: Role of the parser – Context-Free Grammars – Top-Down parsing: Recursive Descent Parsing – Predictive Parsing. Bottom-up parsing: Shift Reduce Parsing – LR Parsers – Error Handling and Recovery in Syntax Analyzer– LEX and YACC tools.

SEMANTIC ANALYSIS AND INTERMEDIATE CODE GENERATION **15**

Introduction to Semantic Analysis – Type Checking. Intermediate Code Generation: Intermediate Languages– Declarations – Assignment Statements – Boolean Expressions – Case Statements – Back patching – Procedure Calls.Run Time Environments: Source Language Issues – Storage Organization – Storage Allocation strategies.

CODE GENERATION AND CODE OPTIMIZATION **15**

Issues in the design of code generator – The Target Machine - Basic Blocks and Flow Graphs – A simple Code generator. Code Optimization: Principal Sources of Optimization – DAG representation of Basic Blocks – Peephole Optimization– Optimization of Basic Blocks – Introduction to Global Data Flow Analysis.

TOTAL PERIODS:	**45**

Laboratory Component:

List of Experiment

1. Implementation of lexical analyzer using C and LEX TOOL.
2. Implementation of a calculator that takes an expression (with digits, + and *), computes and print its value, using YACC.
3. Implementation of a parser using LEX and YACC.
4. Implementation of symbol table
5. Implementation of Predictive parsing.
6. Implementation of Shift Reduce Parsing Algorithm.
7. Implementation of LR parsing.
8. Implementation of front end of a compiler that generates the three-address code for a simple language with One data type integer, arithmetic operators, relational operators, variable declaration statement, one conditional construct, one iterative construct and assignment statement.
9. Implementation of back end of the compiler which takes the three-address code as input and produces assembly language instructions that can be assembled and run using an 8086 assembler. The target assembly instructions can be simple move, add, sub, and jump.
10. Implementation of the code optimizer phase of a compiler that eliminates dead code and common sub-expressions

TOTAL (Lab) PERIODS: 30

TOTAL PERIODS: 75

Text Books:

1	Alfred Aho, Ravi Sethi, Jeffrey D Ullman, Monica S. Lam, “Compilers Principles, Techniques and Tools”, 2nd Edition, Pearson Education Asia, 2013
2	T.G Manikumar, M Ganga Durga,” Principles of Compiler Design”, First Edition, MJP Publisher, 2021.
3	Dhamdhere, D. M. "Compiler Construction: Principles and Practice", 2nd edition, Macmillan India Ltd., New Delhi, 2008.

Reference Books:

1	Fischer, C. N., LeBlanc, R. J. "Crafting a Compiler with C", 1st edition, Benjamin Cummings, 2010.
2	Alblas, H., Nymeyer, A. "Practice and Principles of Compiler Building with C", 1st edition, PHI, 2001.
3	Louden, K. C. "Compiler Construction: Principles and Practice", 1st edition, Thompson Learning, 2003.

Web References:

1	https://gatecse.in/category/compiler-design/
2	www.tutorialspoint.com/compiler_design

23CSC02	MACHINE LEARNING TECHNIQUES	2/0/2/3
Nature of Course	H (Theory Technology)	
Prerequisites	-	

Course Objectives:

1	To understand the concepts and mathematical foundations of machine learning and types of problems tackled by machine learning
2	To explore the different supervised learning techniques including ensemble methods
3	To learn different aspects of unsupervised learning
4	To learn the role of probabilistic methods for machine learning
5	To understand the basic concepts of neural networks and deep learning

Course Outcomes:

Upon completion of the course, students shall have ability to:

C202.1	Apply mathematical concepts like linear algebra, probability, and vector calculus to solve machine learning problems.	[AP]
C202.2	Evaluate different machine learning problems, including supervised and unsupervised learning, along with their applications and challenges.	[AP]
C202.3	Design supervised learning models and analyze their performance using techniques like cross-validation and ensemble methods.	[AN]
C202.4	Construct probabilistic models, including Naïve Bayes and Hidden Markov Models, for sequence modeling and classification tasks.	[AP]
C202.5	Assess deep learning architectures like CNNs and RNNs and compare them to traditional machine learning techniques	[AN]

INTRODUCTION AND MATHEMATICAL FOUNDATIONS **10**

What is Machine Learning? Need –History – Definitions – Applications - Advantages, Disadvantages & Challenges -Types of Machine Learning Problems – Mathematical Foundations - Linear Algebra & Analytical Geometry -Probability and Statistics- Bayesian Conditional Probability -Vector Calculus & Optimization - Decision Theory - Information theory

SUPERVISED AND UNSUPERVISED LEARNING **10**

Introduction-Discriminative and Generative Models -Linear Regression - Least Squares -Under-fitting / Overfitting -Cross-Validation – Lasso Regression- Classification - Logistic Regression- Gradient Linear Models -Support Vector Machines –Kernel Methods -Instance based Methods - K-Nearest Neighbours - Tree based Methods –Decision Trees –ID3 – CART - Ensemble Methods –Random Forest - Evaluation of Classification Algorithms- Introduction - Clustering Algorithms -K – Means – Hierarchical Clustering - Cluster Validity - Dimensionality Reduction –Principal Component Analysis – Recommendation Systems - EM algorithm.

PROBABILISTIC METHODS, NEURAL NETWORKS AND DEEP LEARNING **10**

Introduction -Naïve Bayes Algorithm -Maximum Likelihood -Maximum Apriori -Bayesian Belief Networks -Probabilistic Modelling of Problems -Inference in Bayesian Belief Networks – Probability Density Estimation - Sequence Models – Markov Models – Hidden Markov Models- Neural Networks – Biological Motivation- Perceptron – Multi-layer Perceptron – Feed Forward Network – Back Propagation-Activation and Loss Functions- Limitations of Machine Learning – Deep Learning– Convolution Neural Networks – Recurrent Neural Networks.

TOTAL PERIODS: 30

List of Experiment

1. Implementation of Regression Models – Linear, Polynomial, Ridge & Lasso Regression on real-world data (House Price Prediction)
2. Decision Tree and Ensemble Learning – Implement Decision Tree, Random Forest, and XGBoost for classification tasks (Iris/Titanic dataset)
3. Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) – Train and evaluate classifiers on Handwritten Digit Recognition (MNIST)
4. Artificial Neural Networks (ANN) – Build a feedforward neural network for classification using Backpropagation (MNIST dataset)
5. Convolutional Neural Networks (CNN) – Train a CNN model for image classification using CIFAR-10/Fashion-MNIST dataset
6. Unsupervised Learning – Implement K-Means, Hierarchical Clustering, and DBSCAN for customer segmentation
7. Dimensionality Reduction & Anomaly Detection – Apply PCA & t-SNE for feature reduction, Autoencoders for fraud detection
8. Model Deployment & Explainable AI – Deploy a trained ML model using Flask/FastAPI and analyze model predictions using SHAP/LIME

TOTAL (Lab) PERIODS: 30

TOTAL PERIODS: 60

Text Books:

1	Christoph Molnar, "Interpretable Machine Learning - A Guide for Making Black Box Models Explainable", 1st edition, Creative Commons License, 2020
2	Jason Bell, “Machine Learning – Hands on for Developers and Technical Professionals”, Second Edition, Wiley, 2020.
3	Francois Chollet, “Deep Learning with Python”, Second Edition, Manning Publications, 2020.
4	Andriy Burkov, “Machine Learning Engineering”,1ST Edition, True Positive Inc, 2020

Reference Books:

1	Josh Starmer, PhD, "The StatQuest Illustrated Guide to Machine Learning!!!: Master the Concepts, One Full-Color Picture at a Time, from the Basics All the Way to Neural Networks. BAM!", Illustrated Edition, Packt Publishing, 2022.
2	Sebastian Raschka and VahidMirjalili, “Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2”, 3rd Edition, Packt Publishing Ltd, 2019.
3	Mitchell, T. "Machine Learning", 1st edition, McGraw-Hill, 2017.
4	Arnold, T., Kane, M., Lewis, B. W. "Computational Approach to Statistical Learning", 1st edition, CRC Press, 2019.
5	Murphy, Kevin P. "Machine Learning: A Probabilistic Perspective", 1st edition, MIT Press, 2012.

Web References:

1	https://www.youtube.com/channel/UCtYLUTtgS3k1Fg4y5tAhLbw
2	https://onlinecourses.nptel.ac.in/noc16_cs18/preview 2.
3	https://freevideolectures.com/course/2257/machine-learning 3.
4	https://www.coursera.org/learn/machine-learning

23ADC01	ARTIFICIAL INTELLIGENCE AND ITS APPLICATIONS	2/0/2/3

Nature of Course	F (Theory Programming)
Prerequisites	-

Course Objectives:

1	To explore AI's history, definitions, and ethical dimension of AI and differentiate it from Machine Learning and Deep Learning.
2	To study search algorithms, logic-based reasoning, and knowledge representation methods for problem-solving in AI.
3	To gain insights into knowledge representation, logic-based reasoning and expert systems.
4	To implement AI algorithms in domains such as Natural Language Processing (NLP), Computer Vision, and Decision-Making systems.
5	To investigate bias, fairness, accountability, and ethical considerations in designing and implementing AI technologies.

Course Outcomes:

Upon completion of the course, students shall have ability to:

CO1	Assess various Artificial Intelligence (AI) techniques and identify their fundamental concepts.	[U]
CO2	Apply problem-solving techniques such as search algorithms and game theory in AI systems.	[AP]
CO3	Implement knowledge representation and reasoning techniques for intelligent decision-making.	[AP]
CO4	Demonstrate Machine Learning algorithms and AI frameworks to solve real-world problems	[AP]
CO5	Design AI-based applications for NLP, Computer Vision, and Robotics. Evaluate the ethical considerations and social impacts of AI in various domains	[AP]

FOUNDATIONS OF ARTIFICIAL INTELLIGENCE **10**

Definition and History of AI - AI vs Machine Learning vs Deep Learning - Applications of AI in Healthcare, Finance, and Robotics - AI Ethics and Challenges - Turing Test and AI Approaches (Symbolic AI vs Sub-symbolic AI). PROBLEM SOLVING in AI **-** Problem Formulation - Uninformed Search (BFS, DFS) - Informed Search (A*, Greedy Search) - Constraint Satisfaction Problems (CSPs) - Adversarial Search and Game Theory (Minimax Algorithm, Alpha-Beta Pruning)

KNOWLEDGE, REASONING AND LEARNING **10**

Logic in AI: Propositional and First-Order Logic - Knowledge Representation: Semantic Networks, Frames, Ontologies - Inference Mechanisms: Forward Chaining, Backward Chaining - Introduction to Expert Systems - Fundamentals of Language -Probabilistic Language Processing -NLP Basics - Reinforcement Learning and AI Agents

AI IN ACTION – PERCEPTION, COMMUNICATION & APPLICATIONS **10**

Information Retrieval & Extraction – Perception: Image Formation, Object Recognition – AI in Computer Vision (Image Recognition) – AI in Edge Computing and IoT- Case Study - AI Applications in Smart Cities, Healthcare, and Business

TOTAL (Theory) PERIODS:	**30**

List of Experiment

1. Implement a simple AI-based Python script
2. Exploring AI applications in real-world case studies.
3. Implementing BFS and DFS in Python.
4. Implement A* and Greedy Best-First Search to find the shortest path in a grid.
5. Implement an AI agent to play Tic-Tac-Toe using the Minimax algorithm.
6. Implement a basic propositional logic reasoning system.
7. Build a simple knowledge base using First-Order Logic.
8. Create a semantic network to represent relationships between objects.
9. Design a basic expert system for medical diagnosis
10. Perform tokenization, stemming, and stop-word removal on sample text.
11. Implement a speech-to-text system using Python's speech recognition library.
12. Implementing AI for a real-world problem (e.g., predicting student performance).

TOTAL (Lab) PERIODS: 30

TOTAL PERIODS: 60

Text Books:

1	Stuart Russell & Peter Norvig, "Artificial Intelligence: A Modern Approach", 4th Edition, Pearson, 2021.
2	K. R. Chowdhary, Fundamentals of Artificial Intelligence, 1st Edition, Springer, 2020.
3	Elaine Rich & Kevin Knight, "Artificial Intelligence", 3rd Edition, McGraw-Hill, 2017.

Reference Books:

1	Wolfgang Ertel, Introduction to Artificial Intelligence, 2nd Edition, Springer, 2018.
2	Mariusz Flasiński, Introduction to Artificial Intelligence, Springer, 2016.
3	Stephen Lucci & Danny Kopec, Artificial Intelligence in the 21st Century: A Living Introduction, 2nd Edition, Mercury Learning and Information, 2016

Web References:

1	https://www.coursera.org/learn/introduction-to-ai
2	https://www.coursera.org/learn/ai-for-everyone
3	https://ai.google/get-started/for-developers/
4	https://nptel.ac.in/courses/106102220
5	https://nptel.ac.in/courses/106106140

23CS603	Capstone Project	0/0/6/3
Nature of Course	M (Practical Application)	
Pre-Requisites	Programming Languages	

Course Objectives:

1	To identify a problem area and showcasing a strong understanding of the selected domain.
2	To explore the latest advancements within their selected field of study.
3	To understand and adhere to ethical standards and professional practices in software development.

Course Outcomes:
Upon completion of the course, students shall have ability to

C603.1	Identify a problem and carry out a thorough study on the chosen problem	[AP]
C603.2	Analyze ongoing developments in the chosen domain and demonstrate technical knowledge pertaining to the same.	[A]
C603.3	Apply suitable tools, techniques, Algorithms, frameworks to solve the practical problems.	[AP]
C603.4	Develop a solution for the chosen problem and validate the results.	[C]

Course Guidelines:

Introduction: Identify domain, Framing Problem Statement, Dataset Collection, Analyze the Techniques, Organize the work flow. **Experiments:** Develop software life cycle model, Implement, and provide solution for the chosen problem statement, Validate the result, and provide the documentation for findings.

1. The entire semester shall be utilized by the students to do their Mini project work by receiving the directions from the project guide.
2. Every student shall have a project guide who is the member of the faculty of the institution for the in-house project or an industry mentor from the industry as project guide for an industry/internship project.
3. Identification of project guide has to be completed by the end of previous semester of the project work to be carried out.
4. The duration may be used for library reading, laboratory work, literature survey, computer analysis or field work as assigned by the guide and also to present periodical seminars about the progress made in the project.
5. Number of students in the project team should be maximum of 4.
6. Students can select project topics from the thrust areas.
7. Projects can be Research Based, Application Based, or Multidisciplinary.
8. Students can choose projects in line with the Departmental Mission, Vision and Program Outcomes.
9. Students can identify the project area / title, obtain the consent of faculty to guide them.
10. Students can make use of college subscribed E-resources like IEEE, ScienceDirect and Elsevier to choose base papers and thereby do literature surveys.
11. After project guide allocation, the student team must meet the respective project guide and update about the status of project periodically.
12. While working on the project, every student team must keep a project diary and record all relevant information. The diary must be verified and signed by the project guide which will be the periodic progress report and submitted during the project review to the project coordinator.

13. Students should not be involved in unethical behaviour, such as plagiarism, copyright violations, etc while working on projects and when submitting project reports.
14. The progress of the project will be evaluated on a continuous basis by conducting periodic internal reviews. The review committee may be constituted by the Head of the Department.
15. A final external project viva-voce examination will be conducted to evaluate the student project work based on oral presentation and the project report by an Internal and External Examiner.
16. Every student team will be required to prepare and submit two (2) copies plus (no. of students) copies of the Project report of typical length 30 – 60 pages (excluding Appendices).
17. The final report shall be in typewritten form as specified in the guidelines issued by the COE.
18. As outcome of the project, students are motivated to publish papers in Scopus Indexed Journals or present the project work in International Conferences.

PROFESSONAL ELECTIVES

Stream I - ML ENGINEERING

23AD902	EXPLORATORY DATA ANALYSIS USING PYTHON	2/0/2/3
Nature of Course	F (Theory Programming)	
Prerequisites	-	

Course Objectives:

1	To utilize NumPy and Pandas for data manipulation and structuring.
2	To implement querying and transforming data using Data Frame operations.
3	To familiarize learners with descriptive and exploratory visualizations using Python libraries.
4	To teach statistical concepts like central tendency, dispersion, and correlation in the context of EDA.
5	To enable learners to apply best practices of univariate, bivariate, and multivariate analysis using Python and GenAI tools.

Course Outcomes:

Upon completion of the course, students shall have ability to:

C902.1	Apply NumPy and Pandas to organize and manipulate structured data for analysis.	[U]
C902.2	Apply Query and transform Data Frames using filters, aggregations, and CRUD operations.	[AP]
C902.3	Create descriptive and exploratory visualizations using Matplotlib and Python libraries.	[AP]
C902.4	Apply descriptive statistics and explore central tendency, dispersion, and correlation.	[AP]
C902.5	Develop a comprehensive EDA project that demonstrates multivariate analysis, integrates SQL data sources, and uses GenAI tools for insights.	[AP]

DATA STRUCTURING WITH PYTHON LIBRARIES **10**

Array Structures - 1D/2D NumPy Arrays, Indexing, Slicing, Broadcasting. Data Structures - Series, DataFrames, Data Types, Null Handling. Data Manipulation - Sort, Drop, Apply, Map, Merge, Join, groupby, concat. Data Types and Conversion – astype(), to_numeric(). Basic Data Cleaning – dropna(), fillna(), replace().

QUERY TRANSFORMATION DATA AND DESCRIPTIVE ANALYSIS **10**

Data Filtering - Conditional Filters, Logical Operators, Query Method. Aggregation - Mean, Sum, Count, Min, Max, Custom Aggregations. CRUD Operations - Insert, Update, Replace, Delete rows/columns. SQL Integration – Sqlite3, Connectors, Executing Queries from Python.

Descriptive Analysis - Mean, Median, Mode, Range, Standard Deviation, Variance. Correlation Analysis - Pearson Coefficient, Covariance Matrix. Visualization: Matplotlib Charts, Seaborn Charts, Histograms, Boxplots, Scatterplots, Pairplots.

EDA BEST PRACTICES AND MULTIVARIATE ANALYSIS **10**

EDA Techniques - Univariate, Bivariate, Multivariate Analysis. GenAI Tools - AI-generated Summary, Code Assistance, Automated Charts. Project and Summative Assessment - Perform a complete exploratory data analysis using NumPy, Pandas, and Matplotlib. Integrate SQL datasets and use GenAI tools for generating summaries, visuals, and insights.

List of Experiments

1. Create and manipulate 1D and 2D NumPy arrays for basic numeric operations.
2. Construct Pandas Series and Data Frames from lists, dictionaries, and CSV files.
3. Apply filtering and aggregation techniques on Data Frames using groupby and pivot table.
4. Perform insert, update, and delete operations on Data Frames to simulate CRUD logic.
5. Connect Python with a SQLite database and retrieve data into a Data Frame.

6. Calculate mean, median, mode, standard deviation, and correlation on sample datasets.
7. Create visualizations like bar charts, histograms, scatter plots, and box plots using Matplotlib.
8. Perform univariate and bivariate analysis on a dataset using appropriate visuals and stats.
9. Demonstrate multivariate analysis using pair plots or heatmaps on complex datasets.
10. Complete a mini project that uses GenAI tools to generate exploratory insights and visual summaries.

Total Theory	**30 periods**
Total Lab	**30 periods**
Total	**60 periods**

Text Books:

1	"Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython" by Wes McKinney, Third Edition, O'Reilly Media, 2022.
2	"Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python" by Peter Bruce, Andrew Bruce, and Peter Gedeck, Second Edition, O'Reilly Media, 2020.

Reference Books:

1	"Data Science from Scratch: First Principles with Python" by Joel Grus, Second Edition, O'Reilly Media, 2019.
2	"Python Data Science Handbook: Essential Tools for Working with Data" by Jake VanderPlas, O'Reilly Media, 2016.
3	"Hands-On Exploratory Data Analysis with Python: Perform EDA Techniques to Understand, Summarize and Investigate your Data" by Suresh Kumar Mukhiya and Usman Qamar, Packt Publishing, 2020.

23AD905	STATISTICAL METHODS AND BASIC MACHINE LEARNING MODELS	2/0/2/3
Nature of Course	F (Theory Programming)	
Prerequisites	-	

Course Objectives:

1	To develop a foundational understanding of probability theory and sampling methods essential for statistical inference.
2	To apply and interpret hypothesis testing techniques to validate assumptions using real-world datasets.
3	To perform data preparation and preprocessing for effective use in machine learning applications.
4	To explore and analyze regression and classification models for identifying patterns and making predictions.
5	To gain practical experience in implementing clustering algorithms to discover hidden structures in data.

Course Outcomes:

Upon completion of the course, students shall have ability to:

C905.1	Understand and apply probability concepts and sampling techniques for statistical inference.	[AN]
C905.2	Analyse data distributions and perform hypothesis testing using Z-test, T-test, and Chi-Square test	[AN]
C905.3	Prepare and transform data effectively for building machine learning models.	[AP]
C905.4	Apply linear regression techniques to analyse relationships between variables	[C]
C905.5	Implement classification models like logistic regression and K-Nearest Neighbors for predictive analytics and perform clustering using K-Means through a course-end project simulating real-world practices.	[AP]

PROBABILITY, SAMPLING, AND STATISTICAL INFERENCE **10**

Probability and Prediction- Basic probability- Joint and conditional probability- Real-world probability applications- Sampling and Normal Distributions- Sampling techniques- Sampling distribution- Properties of the normal distribution- Central Limit Theorem- Statistical Inference and Hypothesis Testing(Parametric) - Formulating hypotheses, Z-test and T-test for means, p-value interpretation, confidence intervals. Hypothesis Testing(Nonparametric) - Chi-Square goodness-of-fit test, test for independence in contingency tables.

DATA PREPARATION AND SUPERVISED LEARNING MODELS **10**

Data Preparation for Machine Learning- Handling missing values- Encoding categorical variables- Data normalization and scaling-Train-test split-Regression Models-Simple linear regression: line of best fit, correlation vs causation, residual analysis -Multiple linear regression: multicollinearity, variable selection, coefficient interpretation, model assumptions- Classification Models- Logistic regression: binary classification, sigmoid function, threshold tuning- Confusion matrix, ROC-AUC.

UNSUPERVISED LEARNING AND MODEL EVALUATION **10**

K-Nearest Neighbor (KNN) - Distance metrics, choosing k, model training and prediction, performance evaluation. K-Means Clustering - Centroid initialization, distance calculation, elbow method, cluster interpretation and visualization. Model Evaluation -Accuracy, precision, recall, F1-score, cross-validation- Project / Summative Assessment- Implementation of regression, classification, and clustering models- Use of real-world datasets- Model analysis and reporting.

LIST OF LAB EXPERIMENTS

1. Probability Estimation: Simulate and compute simple, joint, and conditional probabilities using

Python.
2. Sampling and Visualization: Apply simple random and stratified sampling on datasets and visualize distributions.
3. Normal Distribution Analysis: Plot and interpret normal distribution with Python and identify z-scores.
4. Hypothesis Testing - Z and T Test: Perform one-sample and two-sample z-test and t-test on numerical data.
5. Chi-Square Test: Conduct Chi-square goodness of fit and independence tests for categorical variables.
6. Data Cleaning and Scaling: Perform handling of missing values, outlier detection, and standardization.
7. Simple Linear Regression: Implement and evaluate a linear regression model using one independent variable.
8. Multiple Linear Regression: Build a model using multiple features and interpret coefficients and R^2.
9. Classification using Logistic Regression and KNN: Train and evaluate logistic regression and KNN models using classification metrics.
10. K-Means Clustering Project: Apply clustering on a real dataset, use elbow method to select k, and interpret results.

Total Theory	**30 periods**
Total Lab	**30 periods**
Total	**60 periods**

Text Books:

1	"Statistics for Data Science and Business Analysis" by Bhushan Patil, BPB Publications, 2020.
2	"Introduction to Machine Learning with Python: A Guide for Data Scientists" by Andreas C. Müller and Sarah Guido, O'Reilly Media, 2016.

Reference Books:

1	"Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python" by Peter Bruce, Andrew Bruce, and Peter Gedeck, Second Edition, O'Reilly Media, 2020.
2	"Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron, Third Edition, O'Reilly Media, 2022.
3	"Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili, Third Edition, Packt Publishing, 2019.

23IT901	ADVANCED MACHINE LEARNING TECHNIQUES	2/0/2/3
Nature of Course	D (Theory Application)	
Pre requisites	-	

Course Objectives:

1.	To equip learners with the knowledge of advanced supervised learning techniques for accurate classification tasks.
2.	To develop the ability to select optimal models using cross-validation and robust performance metrics.
3.	To introduce ensemble learning methods for improved generalization and predictive performance.
4.	To provide hands-on experience in unsupervised learning using hierarchical and density-based clustering methods.
5.	To enable learners to perform dimensionality reduction and explore smart tools like GenAI for automating ML workflows.

Course Outcomes

Upon completion of the course, students shall have ability to

C901.1	Illustrate advanced supervised learning techniques like Naive Bayes, SVM, and Decision Trees for data classification tasks.	[U]
C901.2	Apply unsupervised learning techniques using Hierarchical Clustering and DBSCAN to discover patterns in unlabelled data.	[AP]
C901.3	Identify ensemble learning techniques such as Random Forest, Boosting, and Stacking for improved predictive accuracy.	[AP]
C901.4	Examine dimensionality reduction techniques like PCA and LDA to enhance model performance and interpretability.	[AN]
C901.5	Develop automation of machine learning workflows using Generative AI and smart ML tools for streamlined development.	[AP]

SUPERVISED AND UNSUPERVISED LEARNING TECHNIQUES **10 Hours**

Naive Bayes Classification - Probabilistic modelling, prior and likelihood, Gaussian and Multinomial Naive Bayes, real-world applications. Support Vector Machines (SVM) - Hyperplane and margins, kernel tricks, soft vs hard margin, model tuning. Decision Trees for Classification - Tree construction, entropy, Gini index, overfitting and pruning.
Hierarchical Clustering - Dendrograms, agglomerative vs divisive, linkage criteria, cluster interpretation. DBSCAN Clustering - Density-based clustering, core points, epsilon-neighbourhood, handling noise/outliers.

MODEL EVALUATION AND ENSEMBLE LEARNING **10 Hours**

Model Validation and Selection - K-Fold, hold-out, stratified K-Fold, LOO cross-validations, model performance comparison. Random Forest & Ensemble Learning - Bagging, feature importance, OOB error, Random Forest classifier. Boosting and Stacking Models - AdaBoost basics, model stacking architecture.

DIMENSIONALITY REDUCTION TECHNIQUES, ML WORKFLOW AUTOMATION AND GENAI TOOLS **10 Hours**

Dimensionality Reduction Techniques - PCA for variance preservation, LDA for class separation. ML Workflow Automation - GenAI-powered model recommendation tools, Auto ML pipelines,

smart feature engineering.	
Total Hours	**30**

Laboratory Component:

List of Experiments

1. Naive Bayes Classifier: Apply Gaussian/Multinomial Naive Bayes for text or numeric classification.
2. SVM Implementation: Train and test an SVM model with different kernels (linear, RBF).
3. Decision Tree Classifier: Build a tree-based classifier and visualize decision rules.
4. Hierarchical Clustering: Use dendrograms to visualize clusters and interpret structure.
5. DBSCAN Clustering: Implement density-based clustering and tune epsilon and minPts.
6. Cross-Validation: Perform k-fold cross-validation to compare multiple models.
7. Random Forest Model: Train and evaluate a random forest classifier on a structured dataset.
8. Boosting Techniques: Apply AdaBoost or XGBoost to improve model performance.
9. PCA and LDA: Apply PCA and LDA to reduce dimensions and plot transformed data.
10. AutoML Tools/GenAI Integration: Use GenAI toolkits (e.g., Google Vertex AI, AutoML, or GenAI notebooks) for automated model selection and optimization.

Total Hours: 30

Text Books:	
1.	Christopher M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006
2.	Aurélien Géron, 2. “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”, Third Edition, O'Reilly Media, 2022
Reference Books:	
1.	Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012
2.	Andreas C. Müller, Sarah Guido, “Introduction to Machine Learning with Python: A Guide for Data Scientists”, O'Reilly Media, 2016
3.	Sebastian Raschka, Vahid Mirjalili, “Python Machine Learning”, Third Edition, Packt Publishing, 2019

23IT902	NATURAL LANGUAGE PROCESSING	2/0/2/3
Nature of Course	D (Theory Application)	
Pre requisites	-	
Course Objectives:		
1.	To introduce learners to statistical methods for analysing and interpreting time series data.	
2.	To equip students to build and evaluate time series forecasting models using classical methods like AR, MA, ARIMA, SARIMA, and SARIMAX.	
3.	To enable learners to integrate Generative AI tools to enhance forecasting accuracy and automate predictions.	
4.	To provide knowledge of Natural Language Processing (NLP) techniques to clean, process, and analyze unstructured text data.	
5.	To train learners to implement text vectorization, rule-based matching, and various text classification models including fastText.	
Course Outcomes		
Upon completion of the course, students shall have ability to		
C902.1	Interpret statistical characteristics of time series data for effective modeling and forecasting.	[U]
C902.2	Build classical time series models such as AR, MA, ARMA, ARIMA, SARIMA, and SARIMAX.	[AP]
C902.3	Apply preprocessing and cleaning of unstructured textual data using NLP techniques	[AP]
C902.4	Examine rule-based matching, vectorization, and classification for text-based applications.	[AN]
C902.5	Analyse the impact of text classification models including binary, multi-class, and multi-label using fastText and other tools.	[AN]

TIME SERIES FUNDAMENTALS & CLASSICAL FORECASTING MODELS 10 Hours

Time Series Statistics and Exploration - Time-dependent features, trend, seasonality, autocorrelation, stationarity, ACF and PACF. AR and MA Models - Autoregressive and Moving Average models, lag selection, parameter tuning. ARMA and ARIMA Models - Combining AR and MA, differencing for stationarity, model diagnostics, error metrics. Seasonal Models (SARIMA & SARIMAX) - Seasonal components, exogenous variables, parameter selection, forecasting.

UNDERSTAND NLP FUNDAMENTALS AND TEXT PREPROCESSING 10 Hours

Text Processing Fundamentals - Tokenization, stop word removal, stemming, lemmatization, part-of-speech tagging. Text Cleaning and Annotation -Removing noise, entity recognition, labelling strategies. Rule-Based NLP - Pattern matching with spaCy, custom rules, phrase matching, entity ruler.

TEXT CLASSIFICATION MODELS AND USE OF GENAI FOR NLP 10 Hours

Text Vectorization Techniques - Bag-of-Words, TF-IDF, word embeddings, feature selection for modeling. Text Classification – Basics: Binary, multi-class, and multi-label approaches, confusion matrix, evaluation metrics. Advanced Classification with fastText - Text preprocessing, training

classifiers, model tuning. GenAI for NLP and Forecasting - Apply GenAI to automate text preprocessing and enhance the extraction of insights from unstructured data.	
Total Hours	**30**

Laboratory Component:

List of Experiments

1. Time Series Analysis: Plot and interpret components like trend, seasonality, and residuals.
2. Stationarity Testing: Apply ADF test and visualize rolling statistics.
3. AR/MA Modeling: Build and evaluate Autoregressive and Moving Average models.
4. ARIMA/SARIMA Modeling: Apply ARIMA and SARIMA models with proper order selection.
5. SARIMAX Implementation: Incorporate exogenous variables into SARIMA models.
6. GenAI for Forecasting: Use Generative AI tools to automate and improve time series predictions.
7. Text Cleaning and Tokenization: Preprocess raw textual data with annotation and normalization.
8. Rule-Based Matching: Implement phrase and pattern matching using rule-based NLP techniques.
9. Vectorization Techniques: Convert text to numerical format using CountVectorizer or TF-IDF.
10. Text Classification: Build and evaluate binary, multi-class, and multi-label classifiers using fastText, use GenAI for NLP.

Total Hours: 30

Text Books:	
1.	Galit Shmueli, Kenneth C. Lichtendahl Jr, “Practical Time Series Forecasting with R: A Hands-On Guide”, 2nd Edition, 2016
2.	Daniel Jurafsky, James H. Martin, “Speech and Language Processing”, International Edition, 2008.
References:	
1.	Wes McKinney, “Python for Data Analysis”, 2nd Edition, O'Reilly Media, 2017
2.	Steven Bird, Ewan Klein, Edward Loper, “Natural Language Processing with Python”, O'Reilly Media, 2009.
3.	Online GenAI documentation (OpenAI, Google Vertex AI) for workflow automation

23IT903	COMPUTER VISION	1/0/4/3
Nature of Course	F (Theory Programming)	
Pre requisites		

Course Objectives:

1.	To introduce learners to the foundational principles of deep learning and its application to real-world predictive problems.
2.	To build proficiency in developing deep learning architectures using TensorFlow or PyTorch.
3.	To enable learners to apply LSTM and BERT models for textual analysis such as recommendation systems and sentiment classification.
4.	To teach core computer vision techniques including image processing, classification, detection, and segmentation using CNNs.
5.	To encourage the use of Generative AI tools for critique, interpretation, and explainability in deep learning models.

Course Outcomes

Upon completion of the course, students shall have ability to

C903.1	Outline core concepts and architectures of deep learning for predictive modeling using TensorFlow or PyTorch.	[U]
C903.2	Apply LSTM and BERT models for building recommendation engines and performing sentiment analysis on text data.	[AP]
C903.3	Develop computer vision techniques including image processing, classification, and segmentation.	[AP]
C903.4	Utilize Convolutional Neural Networks (CNNs) to classify images and detect patterns in visual data.	[AP]
C903.5	Make use of Generative AI tools to improve model critique, interpretation, and explainability in computer vision tasks.	[AP]

DEEP LEARNING FUNDAMENTALS AND TEXT MODELS **5 Hours**

Deep Learning Foundations - Neurons, activation functions, forward and backward propagation. Core Building Blocks: Dense layers, dropout, batch normalization, loss functions, optimizers. Predictive Modeling with DL Model definition, training deep networks, evaluation techniques.
Text-Based DL Models - Sentiment analysis using BERT, tokenizer, transformer architecture, fine-tuning. Sequential Models – LSTM - Recurrent layers, building a recommendation system, sequence prediction.

IMAGE PROCESSING & CNNs **5 Hours**

Image Processing Fundamentals - OpenCV operations, grayscale conversion, edge detection, image transformation. Image Classification with CNN - Filters, feature maps, pooling layers, CNN architecture, model training.

TRANSFER LEARNING & DETECTION **5 Hours**

Transfer Learning for Images - Pretrained models, feature extraction, fine-tuning. Object Detection Techniques, Bounding boxes, real-time detection basics. Semantic Segmentation - Pixel-wise

classification, U-Net architecture, real-time applications. Generative AI Integration - Utilize Generative AI to streamline neural network design and hyperparameter tuning, accelerating the development of deep learning models. DL Tools and Frameworks - TensorFlow/PyTorch, Keras usage; debugging, model saving.	
Total Hours	**15**
Laboratory Component:	
List of Experiments	
1. Build a basic neural network using TensorFlow or PyTorch to predict numeric output. 2. Perform sentiment classification using pre-trained BERT on a Twitter dataset. 3. Build a recommendation engine using LSTM with sequential user-item data. 4. Load and preprocess images using OpenCV and apply edge detection filters. 5. Construct and train a CNN from scratch on a dataset like CIFAR-10 or Fashion-MNIST. 6. Implement transfer learning using a pretrained model like ResNet or VGG16 for image classification. 7. Detect objects in an image using bounding boxes and a YOLO or SSD approach. 8. Perform semantic segmentation using U-Net and visualize segmentation maps. 9. Use Grad-CAM or similar techniques to interpret CNN predictions visually. 10. Build a GenAI-powered notebook that critiques model performance and explains misclassifications. 11. Compare image classification accuracy using CNNs vs Transfer Learning techniques. 12. Present a course-end Project using a full pipeline from problem statement to deployment-ready model.	
Total Hours: 60	

Text Books:	
1.	François Chollet, “Deep Learning with Python”,1st Edition, Manning Publisher, 2017.
2.	Aurélien Géron, “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”, 3rd Edition, O'Reilly, 2022.
References:	
1.	Sebastian Raschka, Vahid Mirjalili, “Python Machine Learning”, Packt Publishing, 2017.
2.	Online TensorFlow and PyTorch documentation
3.	Generative AI documentation for automated model tuning and interpretability

23AD906	GENAI ADVANCED PROMPT ENGINEERING & LLMS	1/0/4/3
Nature of Course	M (Practical Application)	
Prerequisites	Basic knowledge of Python programming and fundamentals of prompt engineering.	

Course Objectives:

1	To apply advanced prompt engineering techniques to craft effective prompts and optimize outputs from generative AI models.
2	To programmatically interact with large language models (LLMs) using Java libraries and APIs such as OpenAI and Hugging Face for AI-driven applications.
3	To design and implement API-driven workflows for tasks like content generation, summarization, and text analysis using GenAI services.
4	To address ethical considerations in generative AI, covering areas like bias mitigation, privacy protection, regulatory compliance, and fairness in AI-generated content.
5	To integrate generative AI functionalities into Java-based applications through foundational techniques, enabling the development of practical, AI-enhanced solutions.

Course Outcomes:

Upon completion of the course, students shall have ability to:

C906.1	Analyze advanced prompt engineering techniques to optimize GenAI outputs.	[AN]
C906.2	Implement programmatic interactions with LLMs using Java libraries and APIs (e.g., OpenAI, Hugging Face).	[AP]
C906.3	Design API-driven workflows for specific AI tasks such as content generation, summarization, and text analysis.	[AP]
C906.4	Analyze ethical practices in GenAI, including bias mitigation, privacy safeguards, adherence to regulatory frameworks, and ensuring fairness.	[AN]
C906.5	Apply foundational skills to integrate GenAI into practical applications, preparing for more advanced AI application-building courses.	[AP]

PROMPT ENGINEERING AND LLM INTEGRATION EXPLORING GENAI CAPABILITIES **5**

Introduction to Prompt Engineering: Basics of Prompt Engineering, Prompt Structure, Optimization Techniques, Complex Prompts, Common Pitfalls, Troubleshooting. Interacting with LLMs Programmatically: Java Libraries, API Keys, Request Handling, Response Processing, Key Functions, Error Handling, Rate Limits. Exploring GenAI Capabilities: Text Generation, Summarization, Sentiment Analysis, Text Classification, Language Translation, Use Cases, Output Quality.

API WORKFLOWS AND GENAI APPLICATIONS ETHICS, FAIRNESS, AND RESPONSIBLE AI **5**

API-Driven Workflows: API Workflows, Workflow Design, Automation Scripts, Workflow Efficiency, Model Integration, Security Considerations. Case Studies and Real-World Applications: GenAI Solutions, Customer Service, Education, Marketing, Domain-Specific Challenges, Effectiveness Evaluation. Ethics and Responsible AI: Biases in AI, Bias Mitigation, Privacy Safeguards, Regulatory Compliance, Fairness, Ethical Frameworks, Accountability Measures.

CAPSTONE PROJECT AND ASSESSMENT **5**

Project and Summative Assessment: Project Planning and Design, Solution Development, Testing and Debugging, Review and Refactoring, Project Documentation and Presentation, Summative Course-end Assessment.

TOTAL THEORY : 15 PERIODS

Lab Experiments

1. Design an optimized few-shot prompt that generates creative product descriptions while minimizing hallucinations and ensuring factual accuracy.
2. Develop a Java script that programmatically interacts with an LLM API to generate contextual customer support responses.
3. Create and evaluate different variations of a chain-of-thought prompt for summarizing long-form research papers, ensuring conciseness without losing key insights. Compare the performance with a direct summarization prompt.
4. Implement an interactive chatbot using OpenAI API that refines user-generated prompts. Use the ReAct prompting method to enable reasoning and error correction before responding to the user.
5. Build an API-driven workflow in Java that automates blog content generation and stores the output in a database.
6. Develop an automation pipeline that takes raw user feedback, processes it using sentiment analysis, and categorizes responses for actionable insights. Use chain-of-thought reasoning to improve the classification of nuanced responses.
7. Integrate multiple AI models (e.g., OpenAI + Hugging Face) in a workflow to generate content followed by toxicity detection.
8. Implement a Java-based few-shot language translation tool using an LLM API and evaluate its performance across different dialects.
9. Develop an AI-driven resume screening tool that extracts and classifies key information from job applications.

TOTAL LAB	**60 PERIODS**
TOTAL	**75 PERIODS**

Text Books:

1	"You Look Like a Thing and I Love You" by Janelle Shane (for ethical perspectives).
2	OpenAI, Anthropic, or Cohere documentation for prompt engineering and API usage.
3	"Hands-On GenAI with Python" (online resources / platforms like GitHub & Medium tutorials).

Reference Books:

1	OpenAI Cookbook — https://github.com/openai/openai-cookbook
2	Hugging Face API and Inference Documentation.
3	Ethical AI resources from Google, IBM, and Partnership on AI.
4	Flask and Fast API documentation for real-world web integration.

STREAM II - DATA ANALYSIS WITH ML ESSENTIALS

23AD901	DATA STORYTELLING AND VISUALIZATION	2/0/2/3
Nature of Course	M (Practical Application)	
Prerequisites	-	

Course Objectives:

1	To introduce learners to the fundamentals of data types, Excel functions, and spreadsheet operations.
2	To train learners in data cleaning, transformation, and preparation techniques using Excel tools.
3	To analyze data summarization using PivotTables and apply basic statistical concepts.
4	To understand in creating effective data visualizations to represent patterns, trends, and outliers.
5	To express the ability to craft compelling data narratives using dashboards and visual summaries.

Course Outcomes:

Upon completion of the course, students shall have ability to:

C901.1	Identify data categories and apply essential Excel functions to manipulate data efficiently	[U]
C901.2	Apply cleaning and transformation techniques to prepare datasets for analysis.	[AP]
C901.3	Generate summaries and insights using PivotTables and basic statistical techniques.	[AP]
C901.4	Evaluate meaningful visualizations for descriptive data analysis.	[AP]
C901.5	Design a real-world data storytelling project using dashboards and visual narratives based on industry-oriented practices.	[AP]

EXCEL DATA HANDLING AND FUNCTIONS **10**

Data Types and Functions – Data categories, structured vs. unstructured, XLOOKUP(), IF(), SUMIF(), SUMIFS, COUNTIF(), COUNTIFS, AVERAGEIF(), Logical Operators. Data Import and Cleanup – Import from text, remove blanks, trim, find and replace, remove duplicates.

ANALYSIS WITH PIVOTTABLES AND VISUAL TECHNIQUES **10**

PivotTables and Charts – Grouping, filtering, calculated fields, PivotCharts, Central Tendency Analysis – Mean, median, mode. Outlier Detection – Visual spotting, IQR, z-score basics - Bar and Column Charts – Categorical comparisons, stacked vs. clustered. Pie and Donut Charts – Composition representation. Histograms and Scatterplots – Distribution analysis, pattern recognition. Trend Lines and Custom Formatting – Forecasting, dynamic visuals.

DATA STORYTELLING, CORRELATION AND ASSESSMENTS **10**

Correlation and Spread – Variance, standard deviation, correlation coefficient. Storytelling with Data – Visual hierarchy, narrative building. Dashboard Design – Layout planning, dashboards, slicers, interactivity, interpretation - Apply storytelling techniques to a real-world dataset using Excel to design an interactive dashboard and narrative presentation.

TOTAL THEORY: 30 PERIODS

List of Experiments

1. Classify different data types and apply functions like XLOOKUP, COUNTIF, and IF to perform logical operations for a given business scenario.
2. Import messy data from CSV/TEXT and clean it using trimming, find & replace, and removal of duplicates.

3. Create a PivotTable to summarize sales data by category, region, and time.
4. Generate a PivotChart to compare product performance across quarters.
5. Plot bar and pie charts using cleaned data to depict category-wise sales contribution.
6. Use histogram and scatter plot to analyze data spread and identify correlation.
7. Apply statistical formulas to calculate mean, median, mode, and detect outliers.
8. Design a multi-section Excel dashboard using slicers, conditional formatting, and charts.
9. Narrate a short data story integrating visuals and insights derived from an Excel analysis.
10. Apply data cleaning, pre-processing, visualization and storytelling techniques to a real-world dataset using Excel to design an interactive dashboard.

TOTAL LAB **TOTAL**	**30 PERIODS** **60 PERIODS**

Text Books:	
1	"Data Visualization with Excel Dashboards and Reports" — Dick Kusleika.
2	"Storytelling with Data: A Data Visualization Guide for Business Professionals" — Cole Nussbaumer Knaflic.
Reference Books:	
1	Microsoft Excel Online Documentation.
2	Blogs and Tutorials on Excel PivotTables, Charts, and Dashboards.
3	Data storytelling case studies and online courses (LinkedIn Learning, Coursera).

23AD903	PROBLEM SOLVING WITH ANALYTICAL & DESIGN THINKING	2/0/2/3

Nature of Course	F (Theory Programming)
Prerequisites	Nil

Course Objectives:

1	To identify and deconstruct real-world problems through root cause analysis and user-centric research.
2	To generate innovative ideas using structured ideation techniques and frame actionable problem statements.
3	To organize, prioritize, and visualize ideas through the creation of rapid prototypes and user flow mockups.
4	To explore solutions with users and refine designs through iterative feedback and continuous improvement.
5	To design and present real-world solutions and plan further improvements through collaborative decision-making.

Course Outcomes:

Upon completion of the course, students shall have ability to:

C903.1	Identify root causes of real-world problems and build user empathy through research tools and techniques.	[AN]
C903.2	Generate multiple creative solutions and translate user needs into actionable problem statements.	[AN]
C903.3	Organize and prioritise ideas using mapping tools and create low-fidelity prototypes.	[AN]
C903.4	Conduct usability testing and refine designs based on structured user feedback and observation.	[AN]
C903.5	Apply analytical and design thinking practices in an end-to-end industry-oriented problem-solving project.	[AP]

PROBLEM DISCOVERY AND RESEARCH **10**

Problem Identification & Understanding – Spot surface-level vs deeper issues – Apply 5 Whys and Fishbone tools – Break down complex challenges – Frame a clear problem statement. Understanding the User's World – Create Empathy Maps – Draft User Journey Maps – Identify pain points and behavior patterns – Use journey maps to define the real problem. Defining the Problem Statement – Synthesise key user insights – Frame a Point of View (POV) – Write focused "How Might We" questions – Align problem with user/business needs.

IDEATION, PROTOTYPING AND VISUALISATION **10**

Generating Potential Solutions – Run Crazy 8s sketch activity – Use SCAMPER technique Build on team ideas – Avoid early filtering or judging. Prioritising & Organising Ideas – Cluster ideas using Affinity Mapping – Build early process flows – Use Impact–Effort matrix – Select one idea to pursue. Translate ideas into quick draft – Sketch step-by-step user flows – Draw simple screen layouts – Build quick paper-based mockups – Walk others through the concept.

TESTING, REFINEMENT AND FINAL PROJECT EXECUTION **10**

Testing with Users – Conduct usability tests – Use Think-Aloud Protocol – Observe and take silent notes – Identify confusion or friction points. Refine designs using user feedback – Identify what to improve from feedback – Prioritise using Impact–Effort mapping – Document decisions in a changelog – Refine prototypes. **Final Project Execution** – Project and Summative Assessment involving real-world problem definition, user research, ideation, prototyping, testing, and final presentation.

TOTAL THEORY: 30 PERIODS

List of Experiments

1. Root Cause Analysis: Use 5 Whys and Fishbone diagram to identify the core of a given problem.
2. Empathy Mapping: Create empathy maps and journey maps based on user interviews or personas.
3. Problem Framing: Write Point of View (POV) statements and corresponding “How Might We” questions.
4. Idea Generation: Conduct a Crazy 8s activity to generate 8 creative solutions in 8 minutes.
5. SCAMPER Exercise: Apply SCAMPER technique to improve an existing product or service.
6. Idea Prioritisation: Use Affinity Mapping and Impact–Effort Matrix to select a feasible solution.
7. Prototyping: Build a low-fidelity paper prototype and prepare a walk-through of the concept.
8. User Testing: Conduct Think-Aloud usability testing with 2–3 users and document observations.
9. Iteration Log: Maintain a changelog to record design decisions and refinements based on feedback.
10. Mini Project: Execute a mini end-to-end design sprint (Empathize → Define → Ideate → Prototype → Test) on a campus or community problem.

TOTAL LAB	**30 PERIODS**
TOTAL	**60 PERIODS**

Text Books:

1	Tim Brown, Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation, Harper Business.
2	Jeanne Liedtka, Design Thinking for the Greater Good: Innovation in the Social Sector, Columbia Business School Publishing.

Reference Books:

1	IDEO.org, The Field Guide to Human-Centered Design.
2	Kelley, T., The Art of Innovation.
3	Selected online case studies and design thinking toolkits (IDEO, Stanford d.school).

23AD904	POWER BI	2/0/2/3
Nature of Course	M (Practical Application)	
Prerequisites	-	

Course Objectives:

1	To identify the complete workflow of extracting, cleaning, and transforming data using Power Query.
2	To gain knowledge in preparing structured datasets and designing efficient data models for scalable analytics.
3	To understand the use of DAX expressions for aggregations, measures, and implementing business logic.
4	To be familiar with the process of designing interactive dashboards and applying effective data storytelling techniques.
5	To analyse AI-powered analytics features and the secure publishing and management of Power BI reports for collaboration.

Course Outcomes:

Upon completion of the course, students shall have ability to:

C904.1	Extract, clean, and transform raw datasets using Power Query and ETL features in Power BI.	[AN]
C904.2	Structure data and implement robust data models for performance and scalability.	[AN]
C904.3	Use DAX to build dynamic metrics, perform business calculations, and derive insights.	[AP]
C904.4	Design compelling dashboards with interactive visuals for effective data storytelling.	[C]
C904.5	Apply advanced DAX and AI analytics to generate predictive insights and deliver a course-end project simulating real business reporting.	[AP]

DATA PREPARATION AND TRANSFORMATION **10**

Data Extraction and Transformation - Power Query Editor- ETL techniques - connecting to data sources - cleansing and shaping data. Data Preparation and Structuring - Handling missing data - data types - appending and merging queries - data formatting best practices. Data Modeling and Optimization - Creating Relationships - star and snowflakes schema design - calculated columns – role playing dimensions.

MODELING AND DAX CALCULATIONS **10**

Model Optimization - Relationship management - model size reduction - use of aggregations - performance tuning. Basic DAX Calculations – Syntax - calculated columns vs measures – aggregations- conditional – logical - time intelligence functions. Advanced DAX - Context transition, CALCULATE -FILTER, scenario-based analysis, KPIs.

REPORTING, VISUALIZATION AND PROJECT EXECUTION **10**

Data Visualization and Storytelling – Chart selection - visual formatting – narrative driven dashboards – drill through - bookmarks. Interactive Reports – Slicers – filters – tooltips – buttons - custom visuals - cross-filtering. AI-Powered Analytics - Q&A visuals - Smart Narrative - Decomposition Tree - Key Influencers. Project and Summative Assessment including DAX model building - interactive dashboard creation - AI insights - and secure report sharing.

TOTAL THEORY: 30 PERIODS

LIST OF LAB EXPERIMENTS

1. Data Cleaning in Power Query: Load a messy CSV and perform basic cleaning such as removing nulls, renaming columns, and filtering rows.
2. ETL Workflow: Connect multiple data sources (CSV, Excel), perform merge and append operations in Power Query.
3. Model Creation: Create fact and dimension tables from the dataset and establish relationships for a sales dashboard.
4. Calculated Columns and Measures: Use DAX to create calculated columns (e.g., Profit = Sales - Cost) and dynamic measures (e.g., Year-over-Year Growth).
5. Time Intelligence: Create Month-to-Date and Year-to-Date measures using DAX time intelligence functions.
6. Dashboard Design: Develop a sales performance dashboard using line charts, KPI cards, slicers, and bar charts.
7. Drill-through and Bookmarks: Add page navigation and drill-through capabilities to a dashboard with customized tooltips.
8. AI Insights Integration: Use decomposition tree and key influencers visuals for AI-assisted insights.
9. Secure Report Publishing: Publish a report to Power BI Service, set access controls, and schedule data refresh.
10. Mini Capstone Project: Execute a complete reporting cycle including data cleaning, modelling, DAX, visualization, and secure publishing for a retail business use case.

TOTAL LAB	**30 PERIODS**
TOTAL	**60 PERIODS**

Text Books:

1	Alberto Ferrari & Marco Russo, The Definitive Guide to DAX: Business intelligence with Microsoft Excel, SQL Server Analysis Services, and Power BI, Microsoft Press.
2	Daniil Maslyuk, Mastering Microsoft Power BI, Packt Publishing.

Reference Books:

1	Reza Rad, Power BI from Rookie to Rock Star, RADACAD.
2	Ruth Pozuelo Martinez, Learning Power BI, Curbal.
3	Official Microsoft Power BI Documentation and Community resources.

STREAM III - CLOUD IT ADMINISTRATION

23CS901	IMPLEMENTING AND ADMINISTERING ENTERPRISE NETWORKS	2/0/2/3

Nature of Course	D (Theory Application)

Course Objectives:

1	To introduce the fundamentals of network design using OSI and TCP/IP models for various networking scenarios.
2	To develop logical networking skills for small to large-scale networks including LAN, MAN, and WAN environments.
3	To train learners to configure routers and switches using Cisco's Internetworking Operating System (IOS).
4	To equip students with skills in network traffic monitoring, analysis, and troubleshooting using industry-grade tools.
5	To familiarize learners with the use of GenAI tools in detecting network anomalies and strengthening network security.

Course Outcomes:

Upon completion of the course, students shall have ability to

C901.1	Analyse communication protocols using OSI and TCP/IP models and apply them to design efficient networks.	AN
C901.2	Design and implement logical networks for LAN, MAN, and WAN using simulation and configuration tools.	AP
C901.3	Configure routers and switches using Cisco IOS to establish and secure reliable network connectivity.	AP
C901.4	Troubleshoot and optimize network performance using techniques like subnetting, partitioning, and IP planning.	AN
C901.5	Build an end-to-end network project integrating GenAI-based anomaly detection with tools like Wireshark and SNMP, adhering to industry practices.	AP

NETWORK FOUNDATIONS AND LOGICAL NETWORK DESIGN **10**

OSI and TCP/IP models, data encapsulation and de-encapsulation, Ethernet standards, types of cables and connectors, network interface devices, switching basics, collision and broadcast domains, LAN, MAN, WAN architectures, IP addressing and subnetting, VLSM and CIDR, network partitioning, logical topology planning, VLAN creation and configuration, trunking and inter-VLAN routing.

ROUTING AND SWITCHING **10**

Router and switch boot process, Cisco IOS command-line interface, static routing, RIP and OSPF configuration, routing table management, switch configuration, MAC address tables, port security settings,

NETWORK ANALYSIS & ANOMALY DETECTION **10**

Traffic sniffing using Wireshark, protocol and packet analysis, identifying latency and packet loss, SNMP-based monitoring with Zabbix, event correlation, GenAI-based anomaly detection, alert generation and reporting. End-to-end network design and simulation, router and switch configuration, traffic monitoring using Wireshark, integration of GenAI tools for anomaly detection, documentation and reporting of findings.

TOTAL PERIODS:	**30**

Lab Component:		
1	Simulate OSI layer communication using Packet Tracer and analyze encapsulation and de-encapsulation at each layer.	
2	Design a LAN topology in GNS3 and assign static IPs to configure connectivity among nodes.	
3	Perform subnetting and configure appropriate addressing schemes for segmented networks.	
4	Create and configure VLANs and enable inter-VLAN routing in a switch-based topology.	
5	Configure and verify static routing and RIP protocol on multiple routers.	
6	Use Wireshark to capture live packets, filter protocols like ICMP, HTTP, DNS, and analyze headers.	
7	Diagnose network issues using tools like ping, traceroute, and show commands in Cisco IOS.	
8	Monitor network performance using Zabbix with SNMP configurations to track CPU, memory, and interface traffic.	
9	Detect and classify network anomalies using GenAI prompts applied to Wireshark data logs.	
10	Complete an integrated project to design, configure, monitor, and report on a secure and efficient network using simulation and GenAI tools.	
	TOTAL (Theory) PERIODS	**30**
	TOTAL PERIODS	**60**

TEXT BOOKS:	
1	Behrouz A. Forouzan, “Data Communications and Networking”, 5th Edition, McGraw-Hill Education, 2017
REFERENCE BOOKS:	
1	Wayne Lewis, “Cisco Certified Network Associate (CCNA) Routing and Switching 200-120 Official Cert Guide Library”, Cisco Press, 2013
2	Richard Deal, “CCNA Routing and Switching Portable Command Guide”, 4th Edition, Cisco Press, 2016
3	Chris Sanders, “Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems”, 3rd Edition, No Starch Press, 2007
4	William Stallings, “Network Security Essentials: Applications and Standards”, 6th Edition, Pearson, 2016

23CS902	LINUX SYSTEM ADMINISTRATION	20/2/3
Nature of Course	D (Theory Application)	
Pre-requisite(s): -		
Course Objectives:		
1	To introduce learners to Linux system architecture, essential commands, and shell environment.	
2	To equip students with skills to install Linux distributions, manage partitions, and configure file systems.	
3	To enable management of users, groups, permissions, services, and disk operations.	
4	To teach automation of administrative tasks using Bash scripting and GenAI tools.	
5	To train learners to configure network services, implement security hardening, and centralized monitoring in a Linux environment.	
Course Outcomes: Upon completion of the course, students shall have ability to		
C902.1	Install and configure Linux systems with manual disk partitioning and filesystem setup.	AP
C902.2	Manage users, groups, files, permissions, and services for secure system operations.	AN
C902.3	Troubleshoot Linux services and processes to maintain availability and performance.	AN
C902.4	Automate administrative tasks using Bash scripting and integrate GenAI tools effectively.	AN
C902.5	Complete a capstone project applying Linux administration concepts and industry practices in a simulated environment.	AN

LINUX INSTALLATION AND USER MANAGEMENT **10**

Linux installation using ISO files, manual disk partitioning, Linux boot process, file system structure and types, working with mount and unmount commands, terminal usage and basic Linux commands, package management in Debian and RedHat systems, User and group management, setting file and directory permissions using chmod and chown, managing access control lists (ACL), disk partitioning using fdisk and parted, managing Linux processes and services, service status monitoring using systemctl and journalctl.

AUTOMATION AND SHELL SCRIPTING **10**

Writing and debugging Bash scripts, automating routine tasks with cron and at, managing process priorities using nice and renice, multitasking in shell, system update and upgrade operations, using GenAI tools to generate Linux administrative scripts,

NETWORKING AND MONITORING SERVICES **10**

Configuring network interfaces and services, using nmtui and Network Manager, implementing DHCP and DNS services, enabling remote access with SSH and RDP, centralized logging with rsyslog and syslog-ng, log management using logrotate, SNMP-based monitoring using Zabbix.Capstone project integrating Linux installation, configuration, service management, scripting, and monitoring tasks, demonstrating real-world system administration and automation practices using VirtualBox, GenAI, and remote tools.

TOTAL PERIODS:	**30**

Lab Component:		
1	Install Ubuntu or CentOS with manual partitioning using VirtualBox.	
2	Create, modify, and delete users and groups, and enforce password aging policies.	
3	Configure and test file and directory permissions using chmod, chown, and ACL.	
4	Partition disks and create file systems using fdisk, mkfs, and mount them manually.	
5	Monitor and restart services using systemctl and diagnose logs using journalctl.	
6	Write a Bash script to automate the creation of users and configuration of a firewall.	
7	Schedule automated backup tasks using cron and implement basic recovery with rsync.	
8	Configure static IPs, test DHCP and DNS settings, and access systems remotely via SSH.	
9	Implement centralized logging using rsyslog and rotate logs with logrotate.	
10	Generate and validate Linux admin scripts using GenAI for real-time task automation.	
	TOTAL (Theory) PERIODS	**30**
	TOTAL PERIODS	**60**

TEXT BOOKS:	
1	Tommy Singleton, “Linux Administration Handbook”, 2nd Edition, Pearson Education, 2006
REFERENCE BOOKS:	
1	Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, Dan Mackin, “UNIX and Linux System Administration Handbook”, 5th Edition, Pearson, 2017
2	William E. Shotts Jr., “The Linux Command Line: A Complete Introduction”, No Starch Press, 2012.
3	Mark G. Sobell, “A Practical Guide to Linux Commands, Editors, and Shell Programming”, 4th Edition, Pearson, 2017
4	Nemeth, Hein, Snyder, Whaley, “Linux Administration: A Beginner’s Guide”, 8th Edition, McGraw-Hill Education, 2020

23CS903	INFORMATION SECURITY SYSTEMS	2/0/2/3
Nature of Course	D (Theory Application)	
Pre-requisite(s): -		
Course Objectives:		
1	To introduce learners to the foundational principles and core objectives of information security, including the CIA triad, asset protection, and risk assessment methodologies.	
2	To train students to implement secure authentication and enforce access control mechanisms in enterprise systems.	
3	To equip learners with knowledge and tools to build secure network perimeters using firewalls, IDS/IPS systems, and encrypted communication protocols.	
4	To enable students to understand and apply cryptographic techniques for secure data exchange and identity verification.	
5	To leverage GenAI tools to enhance threat detection, malware analysis, and develop real-world security responses.	
Course Outcomes: Upon completion of the course, students shall have ability to		
C903.1	Explain the CIA triad, conduct risk assessments, and align security implementations with industry standards.	U
C903.2	Design and enforce authentication policies using passwords, hashing techniques, and multi-factor authentication.	AP
C903.3	Configure and manage network security defences such as firewalls, IDS/IPS, and VPNs to prevent external threats.	AN
C903.4	Apply symmetric and asymmetric cryptography, generate keys, and secure data using encryption, digital signatures, and PKI.	AP
C903.5	Use GenAI and open-source tools to analyse and mitigate malware threats and implement security controls in practical scenarios.	AP

FUNDAMENTALS OF INFORMATION SECURITY **10**

CIA triad concepts, assets and vulnerabilities, threats and attacks, security goals and objectives, risk analysis methods, information security frameworks and compliance standards (ISO 27001, NIST), security governance and security policiesm,Password policy enforcement, hashing and salting, multi-factor authentication methods, access control models (MAC, DAC, RBAC), implementation of IAM (Identity and Access Management), directory services, SSO integration.

NETWORK PERIMETER PROTECTION **10**

Firewall types and configuration (stateful, stateless), VPN setup and tunneling protocols, IDS/IPS concepts and implementation (Snort, Suricata), secure network protocol usage (SSH, HTTPS, SFTP), transport layer security, port scanning and hardening,

CRYPTOGRAPHY AND DATA SECURITY **10**

Symmetric and asymmetric encryption techniques (AES, RSA), hashing algorithms (SHA, MD5), Kerberos, LDAP, digital signature generation and verification, public key infrastructure (PKI), usage of OpenSSL for encryption and signing, certificate generation and validation.

Capstone project on secure system design including cryptographic implementation, firewall and VPN setup, malware traffic detection using GenAI-based tools. Summative evaluation through documentation, testing, and practical defense strategy presentation.		
	TOTAL (THEORY) PERIODS:	**30**
Lab Component:		
1	Analyse and document security gaps in a given system using the CIA triad model.	
2	Implement password hashing using SHA256 and validate salted hash results.	
3	Set up multi-factor authentication using tools like Google Authenticator or DUO.	
4	Simulate RBAC (Role-Based Access Control) in a Linux or web application environment.	
5	Configure iptables or pfSense to create firewall rules and log denied packets.	
6	Capture and analyse potentially malicious traffic using Wireshark filters.	
7	Encrypt and decrypt sensitive files using OpenSSL command-line utilities.	
8	Create and use public/private key pairs for signing and verification tasks.	
9	Use a GenAI-based tool to simulate malware detection and analyse log patterns.	
10	Final integrated project to demonstrate understanding of access control, cryptographic security, and perimeter defence mechanisms.	
	TOTAL (PRACTICAL) PERIODS	**30**
	TOTAL PERIODS	**60**
TEXT BOOKS:		
1	William Stallings, “Cryptography and Network Security: Principles and Practice”, 8th Edition, Pearson Education, 2023	
2	Michael E. Whitman and Herbert J. Mattord, “Principles of Information Security”, 7th Edition, Cengage Learning, 2023	
REFERENCE BOOKS:		
1	Behrouz A. Forouzan, “Cryptography and Network Security”, McGraw Hill Education, 2007	
2	Mark Rhodes-Ousley, “Information Security: The Complete Reference”, 2nd Edition, McGraw Hill Education, 2012	
3	Nitesh Dhanjani, “Network Security Tools: Writing, Hacking, and Modifying Security Tools”, O’Reilly Media, 2005	
4	Shon Harris and Fernando Maymi, “CISSP All-in-One Exam Guide”, 8th Edition, McGraw Hill Education, 2019	

23CS904	LOW-CODE NO-CODE APPLICATION BUILDING	2/0/2/3
Nature of Course	D (Theory Application)	
Pre-requisite(s): -		
Course Objectives:		
1	To grasp foundational concepts such as programming logic techniques, databases, user interface elements, and basic application logic to create robust LCNC solutions.	
2	To get familiar with LCNC tools, HTML5/CSS3, ChatGPT, and discover their applications in streamlining development.	
3	To create form-based applications with customized user interfaces for data collection and analysis.	
4	To build workflow-driven applications that optimize business processes using automation triggers and logic.	
5	To design and customize reports within LCNC applications to effectively visualize and communicate data insights.	
Course Outcomes: Upon completion of the course, students shall have ability to		
C904.1	Explain LCNC development principles, identify use cases, and select suitable tools and techniques.	AN
C904.2	Design user interfaces and develop applications using drag-and-drop components and responsive layout techniques.	AP
C904.3	Develop dynamic form-based applications with validation, conditional logic, and data collection.	AN
C904.4	Integrate APIs and configure workflow automation using triggers and business logic.	AN
C904.5	Visualize application data using built-in reporting tools and finalize a comprehensive project solution.	AN

FOUNDATIONS OF LCNC AND UI DESIGN **10**

Low-code no-code principles, citizen development and its role in business, benefits and limitations of LCNC, programming logic concepts, working with databases and user interface elements, overview of LCNC platforms, basic HTML5 and CSS3 for customization, ChatGPT for assistance in development, API integration basics, fundamentals of user-centered design, creating responsive interfaces, visual hierarchy and usability principles,

WORKFLOW AUTOMATION AND DATA VISUALIZATION **10**

Form design using LCNC platforms, dynamic fields and conditional logic, multi-step forms and layout optimization, input validation and error handling, designing application logic, process automation through workflows, creating and managing custom triggers, integrating external services into workflows, debugging and optimizing automated processes. Designing and embedding interactive dashboards, choosing effective visualization types, working with real-time data sources, report creation and layout customization, grouping and filtering data, using built-in connectors for cloud services, configuring custom APIs for data fetch and sync, troubleshooting integration issues, testing system interoperability and performance,

ROLE-BASED ACCESS AND SECURITY PRACTICES **10**

User authentication techniques, configuring user roles and access permissions, implementing secure data handling and validation, best practices in managing application sessions and

credentials, audit logging and activity tracking, protecting APIs and third-party integrations, enforcing secure deployment guidelines across LCNC applications. Final capstone project integrating UI design, form development, workflow automation, API integration, and reporting, project planning and scope definition, iterative development and testing, debugging and error resolution, final deployment and presentation, course-end summative assessment and individual reflection.		
	TOTAL (THEORY) PERIODS:	**30**
Lab Component:		
1	Create a simple LCNC-based application to collect user data using form components.	
2	Design a responsive user interface layout with text fields, dropdowns, and buttons.	
3	Implement form validation rules and configure multi-step navigation between form sections.	
4	Automate a workflow that sends notifications based on form submissions using triggers.	
5	Create dynamic dashboards with charts to visualize collected data in real-time.	
6	Integrate a third-party API into your application and display external data in a table.	
7	Build and export a custom report with filtered and grouped data insights.	
8	Use a GenAI tool like ChatGPT to generate field logic and UI text recommendations.	
9	Configure user roles, permissions, and access control for multi-level users.	
10	Develop a capstone LCNC project integrating forms, workflows, API calls, and reporting features.	
	TOTAL (PRACTICAL)) PERIODS	**30**
	TOTAL PERIODS	**60**
TEXT BOOKS:		
1	Lim Mei Ying, Sriram Krishnan, “Low-Code Application Development with Appian”, 1st Edition, Packt Publishing, 2022.	
2	Mark Goodyear, “HTML5 and CSS3: Visual QuickStart Guide”, Peachpit Press, 2011.	
REFERENCE BOOKS:		
1	Jan Vanthienen and Monique Snoeck, “Low-Code Development: A Practical Guide for Business Transformation”, Springer, 2021	
2	Matt Cavanagh, “Building Low-Code Applications with Mendix”, 1st Edition, Packt Publishing, 2021.	
3	Jon Duckett, “HTML and CSS: Design and Build Websites”, 1st Edition, Wiley, 2011.	
4	George Reese, “Cloud Application Architectures: Building Applications and Infrastructure in the Cloud”, O’Reilly Media, 2009	
5	Vijay Kumar Velu, Robert Beggs, “Mastering Kali Linux for Advanced Penetration Testing”, 3rd Edition, Packt Publishing, 2019	

23CS905	VIRTUALIZATION, CLOUD COMPUTING & SYSOPS	1/0/4/3
Nature of the Course: F (Theory Programming)		
Course Objectives:		
1	To understand core principles of IT virtualization, containerization, and cloud computing.	
2	To learn to implement and manage virtualized infrastructure and storage systems.	
3	To use Docker containers for efficient application deployment.	
4	To explore AWS cloud services and architectures for scalable and resilient deployments.	
5	To integrate GenAI tools for enhancing security and cloud operations.	
6	To design and implement fault-tolerant cloud solutions with real-time use cases.	
Course Outcomes: Upon completion of the course, students shall have ability to		
C905.1	Understand the foundational concepts of virtualization, containerization, and cloud deployment models.	U
C905.2	Implement and manage virtual machines, storage, and networking using hypervisors and cloud-based tools.	AP
C905.3	Explore containerization using Docker and distinguish it from traditional virtualization approaches.	AN
C905.4	Design secure and resilient cloud architectures using GenAI tools and AWS services.	AP
C905.5	Gain proficiency in AWS SysOps tasks including resource provisioning, automation, and infrastructure monitoring.	AP

VIRTUALIZATION AND CLOUD FOUNDATIONS **5**

Capital expenditure vs operational expenditure in IT, cloud deployment patterns and service models (IaaS, PaaS, SaaS), just-in-time provisioning concepts, hypervisor types and architecture, creating and managing virtual machines using VirtualBox, introduction to virtual storage and networking, types of virtual networks and configurations.

CONTAINERIZATION AND INFRASTRUCTURE MANAGEMENT **5**

Containerization fundamentals, how Docker differs from hypervisors, building Docker images with Dockerfile, running and managing Docker containers, creating and managing Docker networks, use cases for containerized workloads, integrating containers in cloud environments.

Utilizing GenAI tools for security analysis and compliance, introduction to AWS cloud infrastructure, regions and availability zones, AWS identity and access management (IAM), managing user accounts and billing with AWS Organizations, designing and implementing secure Virtual Private Clouds (VPC), configuring access control and network segmentation, introduction to CloudTrail for auditing.

SYSOPS ON AWS AND SCALABLE DEPLOYMENTS **5**

Provisioning applications on AWS using EC2, managing block storage with EBS, hosting static data using S3, working with RDS for relational databases, configuring auto-scaling and load balancing, monitoring services with AWS tools, deploying fault-tolerant architectures using AWS services, hands-on experience with real-time deployment and infrastructure automation.

Final project focused on designing, implementing, and deploying a secure and scalable cloud-native application, infrastructure planning using VirtualBox and AWS, integration of Docker containers, applying GenAI tools for security evaluation, monitoring and optimization, project documentation, presentation, and summative evaluation.		
	TOTAL (THEORY) PERIODS:	**15**
Lab Component:		
1	Install and configure VirtualBox to create and manage multiple virtual machines.	
2	Design and deploy a virtual network using host-only and bridged network adapters in VirtualBox.	
3	Set up and test Docker containers by creating a Dockerfile and launching multi-container applications.	
4	Create custom Docker networks and connect multiple containers for inter-container communication.	
5	Provision and configure an EC2 instance on AWS and access it using SSH.	
6	Create and attach Elastic Block Store (EBS) volumes to EC2 instances and perform mount operations.	
7	Design and deploy a Virtual Private Cloud (VPC) with public and private subnets using AWS.	
8	Use AWS IAM to create users, groups, and policies, and test access control scenarios.	
9	Enable and analyze CloudTrail logs for auditing AWS account activities.	
10	Plan and execute a project to deploy a secure, fault-tolerant web application using EC2, S3, RDS, and Docker containers.	
	TOTAL (PRACTICAL)) PERIODS	**60**
	TOTAL PERIODS	**75**
TEXT BOOKS:		
1	Tom Clark , “Designing and Implementing Linux Virtualization”, Packt Publishing,	
2	Karl Matthias, Sean P. Kane ,”Docker: Up & Running: Shipping Reliable Containers in Production”, 3rd Edition O’Reilly Media, 2023.	
3	Michael Wittig, Andreas Wittig, “Amazon Web Services in Action”, 3rd Edition Manning Publications, 2023.	
REFERENCE BOOKS:		
1	James Turnbull, “The Docker Book: Containerization is the New Virtualization”, Turnbull Press, 1st Edition, 2019.	
2	Eric Ligman, “AWS Certified SysOps Administrator Official Study Guide”, Wiley, 2017.	
3	Thomas Erl, “Cloud Computing: Concepts, Technology & Architecture”, Prentice Hall, 2013.	
4	Stuart Scott, “AWS Certified Solutions Architect Study Guide: Associate (SAA-C03) Exam”, 4th Edition, Wiley, 2022.	
5	Benjamin Caudill and Karl Gilbert, “Hands-On AWS Penetration Testing with Kali Linux”, Packt Publishing Limited, 2019.	

23CS906	CONTINUOUS MONITORING AND OBSERVABILITY	1/0/4/3
Nature of Course	F (Theory Programming)	

Pre-requisite(s): -

Course Objectives:

1	To understand the need for configuration management and infrastructure as code in cloud environments.
2	To learn to automate server configurations using Ansible and manage resources through AWS CloudFormation.
3	To apply observability principles to monitor AWS infrastructure using CloudWatch metrics, events, and alarms.
4	To configure monitoring dashboards and set up alerting mechanisms for proactive cloud operations.
5	To analyse AWS logs to troubleshoot, detect issues, and implement response strategies using GenAI tools.

Course Outcomes:

Upon completion of the course, students shall have ability to

C906.1	Understand the fundamentals of Configuration Management and Infrastructure as Code using Ansible and AWS CloudFormation.	U
C906.2	Apply Ansible to automate configuration tasks and manage cloud inventory using playbooks, variables, and modules.	AP
C906.3	Develop proficiency in designing and deploying AWS infrastructure using CloudFormation templates and stacks.	AN
C906.4	Monitor AWS infrastructure by setting up CloudWatch dashboards, custom metrics, events, and alarms.	AN
C906.5	Analyse AWS logs to detect patterns, optimize performance, and ensure system health.	AN

CONFIGURATION MANAGEMENT WITH ANSIBLE **5**

Introduction to Configuration Management, Introduction to Ansible, Defining Inventory Files, Ansible Playbooks, Group and Host Variables, Ansible Modules and Galaxy,

INFRASTRUCTURE AUTOMATION AND MONITORING AWS INFRASTRUCTURE **5**

AWS CloudFormation Overview, Writing YAML Templates, Defining Resources, Stack Management, Template Validation, Updating and Deleting Stacks, Linting Best Practices. Introduction to Observability, AWS CloudWatch Metrics, Creating Dashboards, Monitoring EC2, EBS, RDS, ELB, Creating Custom Metrics, Common Graphs and Visualizations.

CLOUDWATCH EVENTS, LOGS, AND ALERTS **5**

CloudWatch Events and Triggers, setting up Alarms, Using AWS Logs for Troubleshooting, Log Insights, Automating Alert Responses, Using GenAI for Alert Analysis

Real-world implementation of monitoring solution using Ansible and CloudFormation, with complete observability via CloudWatch. Includes alert configuration and log analysis using GenAI.

TOTAL (THEORY) PERIODS: 15

Lab Component:

1	Install and configure Ansible on a Linux system and define a basic inventory file.
2	Write and execute an Ansible playbook to configure Apache web server on EC2 instances.

3	Use Ansible variables and handlers to customize service deployment and restart services on change.
4	Create a CloudFormation template to deploy an EC2 instance and attach an EBS volume.
5	Validate and deploy a CloudFormation stack and update it to reflect infrastructure changes.
6	Set up CloudWatch to monitor EC2 instance metrics and create a custom dashboard.
7	Configure CloudWatch to collect and display metrics for EBS, RDS, and ELB services.
8	Create a CloudWatch alarm to notify when CPU utilization exceeds a defined threshold.
9	Analyse AWS CloudTrail logs to track API activities and detect unauthorized changes.
10	Simulate and respond to an infrastructure alert using automation and observability best practices.
	TOTAL (PRACTICAL)) PERIODS **60**
	TOTAL PERIODS **75**

TEXT BOOKS:	
1	Jeff Geerling, “Ansible for DevOps: Server and Configuration Management for Humans”, 1st Edition, Midwestern Mac, 2015.
2	Michael Wittig, Andreas Wittig, “Amazon Web Services in Action”, 3rd Edition, Manning Publications, 2023.
REFERENCE BOOKS:	
1	Lorin Hochstein, “Ansible: Up and Running — Automating Configuration Management and Deployment the Easy Way”,3rd Edition, O’Reilly Media, 2022.
2	Kief Morris, “Infrastructure as Code: Dynamic Systems for the Cloud Age”, 2nd Edition, O’Reilly Media, 2020.
3	Eric Ligman, “AWS Certified SysOps Administrator Official Study Guide”, 3rd Edition, Wiley, 2024.
4	Mike Julian, “Practical Monitoring: Effective Strategies for the Real World”, 1st Edition, O’Reilly Media, 2017.

STREAM IV - CYBER SECURITY

23CY901	CLOUD COMPUTING AND CONTAINERIZED VIRTUAL INFRASTRUCTURE	2/0/2/3

Nature of Course	D (Theory Application)

Pre-requisite(s): -

Course Objectives:

1	To understand the fundamental concepts of cloud computing, virtualization, and containerization.
2	To learn to deploy and manage virtual machines using hypervisor technologies.
3	To explore virtual networking and storage concepts in a cloud-based environment.
4	To implement containerization using Docker for lightweight and scalable workload management.
5	To apply GenAI tools to enhance cloud infrastructure security and risk analysis

Course Outcomes:

Upon completion of the course, students shall have ability to

C901.1	Explain cloud computing models, virtualization techniques, and infrastructure provisioning.	U
C901.2	Deploy and manage virtual machines using tools like VirtualBox.	AP
C901.3	Configure and manage virtual storage and networking systems.	AP
C901.4	Build and manage containerized applications using Docker.	AP
C901.5	Secure containerized and virtualized environments using GenAI-powered tools.	AP

CLOUD COMPUTING AND VIRTUALIZATION CONCEPTS **10**

Capital Expenditure vs. Operational Expenditure, Cloud Deployment Patterns, Service Models (IaaS, PaaS, SaaS), Just-in-Time Provisioning, Auto-scaling and Elasticity, Types of Hypervisors and their use cases

VIRTUAL MACHINE, STORAGE MANAGEMENT AND DOCKER CONTAINERIZATION **10**

Creating Virtual Machines using Oracle VirtualBox, Snapshot and Cloning, Resource Allocation and Optimization, Virtual Storage Concepts (Local and Remote), Creating and Managing Virtual Disks in Linux. Types of Virtual Networking (NAT, Bridged, Host-Only), IP Configuration and Access, Containerization with Docker, Docker vs. Hypervisors, Creating Docker Images with Dockerfile, Running and Networking Docker Containers,

GENAI-DRIVEN CLOUD SECURITY **10**

Security Risks in Virtualized Infrastructure, Cloud Security Principles, Role-Based Access Control, Introduction to GenAI Tools for Threat Detection, GenAI-based Cloud Security Automation and Best Practices. Plan and implement a secure virtual and containerized cloud infrastructure; include hypervisors, Docker, and GenAI security components. Activities include architecture design, deployment, testing, documentation, and presentation.

TOTAL (THEORY) PERIODS: 30

Lab Component:

1	Set up a virtual machine in Oracle VirtualBox and install a Linux OS.
2	Simulate just-in-time provisioning and auto-scaling using cloud deployment tools.
3	Configure virtual networks (NAT and bridged) and verify VM-to-VM connectivity.

4	Attach and partition virtual storage in a Linux-based virtual machine.
5	Create a Dockerfile and build a custom container image.
6	Deploy multiple Docker containers and link them via Docker networks.
7	Set up port forwarding in Docker and expose containerized services.
8	Compare and analyze resource usage between VMs and Docker containers.
9	Use a GenAI tool to identify vulnerabilities in a containerized setup.
10	Design and present a project on secure cloud infrastructure using virtualization and containers
	TOTAL (PRACTICAL)) PERIODS **30**
	TOTAL PERIODS **60**

TEXT BOOKS:

1	Virtualization Essentials — Matthew Portnoy, Wiley, 2nd Edition.
2	Docker: Up & Running — Shipping Reliable Containers in Production — Karl Matthias, Sean P. Kane, O'Reilly Media, 3rd Edition

REFERENCE BOOKS:

1	Mastering Virtual Machine Manager 2019 — Sarmad Ali, Packt Publishing, Latest Edition.
2	The Docker Book: Containerization is the New Virtualization — James Turnbull, Turnbull Press, Latest Edition.
3	AWS Certified SysOps Administrator Official Study Guide — Eric Ligman, Wiley, Latest Edition.
4.	Practical Cloud Security: A Guide for Secure Design and Deployment — Chris Dotson, O'Reilly Media, Latest Edition.

23CY902	PENETRATION TESTING	1/0/4/3
Nature of Course	F (Theory Programming)	
Prerequisite: -		
Course Objectives:		
1	To introduce students to ethical hacking principles, attack methodologies, and threat modeling.	
2	To enable learners to identify and exploit operating system, network, and web application vulnerabilities.	
3	To train students in using penetration testing tools and techniques to simulate real-world cyberattacks.	
4	To empower learners to evaluate and bypass security controls like firewalls, IDS/IPS, and authentication mechanisms.	
5	To enable learners to develop GenAI-based scripts for automated penetration testing and real-time vulnerability detection	
Course Outcomes: Upon completion of the course, students shall have ability to		
C902.1	Understand core ethical hacking concepts, vulnerabilities, and enumeration techniques	U
C902.2	Perform targeted exploitation of operating systems, networks, applications, and web services.	AP
C902.3	Evaluate and optimize tools for penetration testing, privilege escalation, and post-exploitation.	AN
C902.4	Evaluate and enhance security control effectiveness using standard frameworks and GenAI tools.	AN
C902.5	Design and implement a comprehensive penetration testing project simulating real-world threat scenarios.	AP

FUNDAMENTALS OF EXPLOITATION **5**

Vulnerability Types and Exploits, System Enumeration Techniques, Operating System Attacks, Malware Threats and Payloads, Privilege Escalation Methods,

WIRELESS EXPLOITS AND APPLICATION EXPLOITATION TECHNIQUES **5**

Network Scanning and Mapping, Exploiting Network Vulnerabilities, Wireless Network Attacks, Bluetooth and Mobile Exploits, Sniffing, Spoofing, and Hijacking. Web Application Vulnerabilities, SQL Injection and XSS, API and Webhook Exploits, Web Shells and Backdoors, Bypassing Security Controls

AUTOMATION AND GAP ANALYSIS **5**

GenAI for Target Enumeration, Automated Exploit Scripting, Honeypots and Detection Evasion, Security Controls Mapping, NIST, CIS Benchmark Evaluation.

Project and Summative Assessment – Learners will execute a full-scale penetration test project that includes reconnaissance, exploitation, automation using GenAI, and security control evaluation

TOTAL PERIODS: **15**

Lab Component:		
1	Perform enumeration and vulnerability scanning using Nmap and Nikto.	
2	Exploit an operating system vulnerability using Metasploit on Kali Linux.	
3	Bypass user authentication using password-cracking tools like Hydra or John the Ripper.	
4	Exploit a web application vulnerability such as SQL Injection or Cross-Site Scripting (XSS).	
5	Simulate privilege escalation in a Linux or Windows environment.	
6	Capture and analyze network traffic using Wireshark for session hijacking.	
7	Exploit a vulnerable wireless network using tools like Aircrack-ng.	
8	Develop a basic GenAI-assisted Python script for automated reconnaissance and scanning.	
9	Map enterprise security controls and analyze configuration gaps using OWASP and CIS benchmarks.	
10	Conduct a full-cycle penetration testing simulation including reporting and recommendations.	
	TOTAL (Lab) PERIODS	**60**
	TOTAL PERIODS	**75**

TEXT BOOKS:	
1	The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws — Dafydd Stuttard, Marcus Pinto, Wiley, 2nd Edition.
2	Penetration Testing: A Hands-On Introduction to Hacking — Georgia Weidman, No Starch Press, Latest Edition.
3	Kali Linux Revealed: Mastering the Penetration Testing Distribution — Raphael Hertzog, Jim O'Gorman, Mati Aharoni, Offensive Security Press, Latest Edition.

REFERENCE BOOKS:	
1	The Hacker Playbook 3: Practical Guide To Penetration Testing — Peter Kim, Secure Planet LLC, Latest Edition.
2	Hacking: The Art of Exploitation — Jon Erickson, No Starch Press, 2nd Edition.
3	Metasploit: The Penetration Tester's Guide — David Kennedy, Jim O'Gorman, Devon Kearns, Mati Aharoni, No Starch Press, Latest Edition.
4	Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning — Gordon Fyodor Lyon, Insecure.Com LLC, Latest Edition.
5	Advanced Penetration Testing: Hacking the World's Most Secure Networks — Wil Allsopp, Wiley, Latest Edition.
6	OWASP Testing Guide — OWASP Foundation, Latest Version (Open Source).
7	Relevant GenAI in Cybersecurity Reports and Tools — Industry White Papers, Latest.

23CY903	SECURITY OPERATIONS OF INFORMATION SYSTEMS	1/0/4/3
Nature of Course	F (Theory Programming)	
Prerequisite: -		
Course Objectives:		
1	To introduce foundational principles of incident response and security operations in modern information systems.	
2	To train students to detect, respond to, and recover from various cybersecurity threats using real-time monitoring tools.	
3	To equip learners with forensic investigation methods for effective threat identification and evidence preservation.	
4	To enable students to apply and assess cybersecurity controls using industry frameworks like NIST and MITRE ATT&CK.	
5	To guide learners in generating and automating incident response playbooks using GenAI and other modern tools.	
Course Outcomes: Upon completion of the course, students shall have ability to		
C903.1	Describe the core concepts of incident response, policies, and operational security procedures	U
C903.2	Apply containment, eradication, and recovery strategies to manage cybersecurity incidents.	AP
C903.3	Investigate and analyze digital threats using forensic techniques and open-source tools.	AN
C903.4	Evaluate and improve GenAI-driven incident response playbooks based on threat scenarios.	AN
C903.5	Develop and present a functional SOC dashboard or case study applying threat detection and mitigation frameworks.	AP

FOUNDATIONS OF INCIDENT RESPONSE **5**

Incident types and lifecycle, Policy creation and compliance, Roles and responsibilities in IR, Stakeholder communication plan

CONTAINMENT RECOVERY TACTICS AND FORENSICS INVESTIGATION SECURITY CONTROLS **5**

, Initial response procedures, Containment strategies, Eradication methods, Recovery and post-incident activities. Log collection and preservation, Host and network forensics, Chain of custody, Timeline analysis and documentation.

GENAI INTEGRATION **5**

Threat intelligence mapping, SIEM dashboards and alerts, Security control implementation, Playbook generation using GenAI.

Develop and present an end-to-end SOC use-case scenario integrating SIEM, forensic investigation, and automated GenAI-based playbook.

TOTAL PERIODS:	**15**

Lab Component:	
1	Configure and deploy Wazuh SIEM and perform basic log ingestion.
2	Create an incident response policy document tailored to a small enterprise.

3	Simulate a containment and eradication scenario using Linux or Windows systems.
4	Collect and preserve forensic evidence from a compromised virtual machine.
5	Use Wazuh to generate alerts for brute-force attacks and unauthorized access.
6	Integrate MITRE ATT&CK mappings into a SIEM dashboard for threat tracking.
7	Develop a basic time-series chart from logs using Wazuh or ELK stack.
8	Generate a GenAI-assisted incident response playbook for ransomware.
9	Conduct a network traffic investigation and produce a report.
10	Design a custom SOC dashboard displaying security control effectiveness
	TOTAL (Lab) PERIODS **60**
	TOTAL PERIODS **75**

TEXT BOOKS:	
1	Incident Response & Computer Forensics — Jason Luttgens, Matthew Pepe, Kevin Mandia; McGraw-Hill Education; 3rd Edition.
2	The Practice of Network Security Monitoring: Understanding Incident Detection and Response — Richard Bejtlich; No Starch Press; Latest Edition.
3	Cybersecurity Incident Response: How to Contain, Eradicate, and Recover from Incidents — Eric C. Thompson; Packt Publishing; Latest Edition.
REFERENCE BOOKS:	
1	Blue Team Handbook: Incident Response Edition — Don Murdoch; CreateSpace Independent Publishing; Latest Edition.
2	Digital Forensics and Incident Response: Incident response techniques and procedures to respond to modern cyber threats — Gerard Johansen; Packt Publishing; 3rd Edition.
3	NIST Computer Security Incident Handling Guide (SP 800-61 Rev. 2) — National Institute of Standards and Technology; Free Public Document; Latest Revision.
4	MITRE ATT&CK® for Enterprise: Design and Implementation Guide — MITRE Corporation; Open Source White Papers & Official ATT&CK Documentation.
5	Practical Threat Intelligence and Data-Driven Threat Hunting — Valentina Costa-Gazcon; Packt Publishing; Latest Edition.
6	Wazuh Documentation and ELK Stack Official Guides — Official Open Source Resources; Latest.

STREAM V - FULL STACK SOFTWARE DEVELOPMENT

23CS911	MANAGING AND QUERYING DATABASE (RDBMS) MySQL / POSTGRE SQL	2/0/2/3
Nature of Course	D (Theory Application)	
Pre-requisite(s): -		
Course Objectives:		
1	To introduce students to the structure and functioning of relational databases and the role of RDBMS.	
2	To enable students to design normalized database schemas using Entity-Relationship (ER) diagrams and apply integrity constraints.	
3	To teach students how to perform SQL operations to retrieve, manipulate, and manage data effectively using DDL, DML, and DQL commands.	
4	To equip learners with the ability to ensure data consistency through transactions, triggers, stored procedures, and automation.	
5	To familiarize learners with semi-structured data and explore the fundamentals of NoSQL for handling flexible data models.	
Course Outcomes: Upon completion of the course, students shall have ability to		
C911.1	Explain the role of RDBMS in organizing, securing, and ensuring data integrity.	U
C911.2	Design relational databases using ER diagrams and normalization to ensure minimal redundancy and high consistency.	AP
C911.3	Construct and execute SQL queries using various clauses, joins, and subqueries to manage and retrieve data.	AP
C911.4	Implement stored procedures, triggers, and transactions to automate and maintain consistency in data operations.	AN
C911.5	Build a practical database application integrating real-world scenarios and applying SQL automation and NoSQL for scalable solutions.	AP

QUERYING WITH SQL BASICS **10**

SELECT Queries - MySQL Setup, SELECT, WHERE, AND/OR, comparison operators, String Functions, Date Functions. Joins - Inner Join, Outer Join, Left Join, Right Join, Primary Key, Foreign Key. Data Summarization - Aggregation Functions, Group By Clause, Having Clause, Order By Clause. Subqueries - Scalar Query, Tabular Query, Correlated/Non-Correlated Subqueries.

DATA MANIPULATION, SCHEMA DESIGN AND CONSISTENCY MANAGEMENT **10**

DML Queries: Insert Query, Update Query, Delete Query, Index. Data Integrity – Primary Key, Foreign Key, UNIQUE, NOT NULL, CHECK. Database Design - Entities, Attributes, Entity Relationships, Normalization (1NF, 2NF, 3NF). DDL Commands - Create Command, Alter Command, Constraints (NOT NULL, UNIQUE, CHECK), Grant – Revoke Permissions. Database Consistency: ACID properties, Locking Mechanisms, Temporary Tables, Views. ACID Properties – Atomicity, Consistency, Isolation, Durability

AUTOMATING AND EXTENDING DATABASES **10**

Automate Database Operations - Create Procedure, Create Trigger, Schedule Operations, Repetitive Task Automation. JSON and NoSQL - JavaScript Object Notation – JSON, JSON

Datatype, JSON_OBJECT(), JSON_EXTRACT(), NoSQL Databases, XDevAPI, XDevAPI CRUD Methods. Design and implement a fully functional database system including schema design, query operations, procedures, and automation aligned with a real-world industry scenario.		
	TOTAL PERIODS:	**30**
Lab Component:		
1	Write SQL queries using SELECT, WHERE, ORDER BY, and GROUP BY on sample employee or sales database.	
2	Perform INNER JOIN, LEFT JOIN, and SELF JOIN operations to combine multiple related tables.	
3	Construct scalar and correlated subqueries to filter and retrieve nested data.	
4	Insert, update, and delete data from tables using DML commands while maintaining integrity constraints.	
5	Create ER diagrams using sample case studies and convert them into normalized schemas.	
6	Define tables with PRIMARY KEY, FOREIGN KEY, and apply constraints using DDL.	
7	Demonstrate ACID properties with transactions using COMMIT and ROLLBACK.	
8	Write a stored procedure to calculate and return derived information (e.g., monthly sales report).	
9	Create a trigger to log changes in a table when new rows are inserted or updated.	
10	Parse and manipulate semi-structured data using JSON functions in MySQL and demonstrate a basic NoSQL structure.	
	TOTAL (LAB) PERIODS	**30**
	TOTAL PERIODS	**60**
TEXT BOOKS:		
1	Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, “Database System Concepts”, Seventh Edition, McGraw-Hill Education, 2020.	
2	Russell J.T. Dyer, “Learning MySQL and MariaDB: Heading in the Right Direction with MySQL and MariaDB”, O’Reilly Media, 2015.	
REFERENCE BOOKS:		
1	Ramez Elmasri and Shamkant B. Navathe, “Fundamentals of Database Systems”, Seventh Edition, Pearson Education, 2017.	
2	Allen G. Taylor, “SQL For Dummies”, Eighth Edition, Wiley, 2018.	
3	Adam Fowler, “NoSQL For Dummies”, Wiley, 2015.	

23CS912	JAVA / PYTHON: OBJECT-ORIENTED PROGRAMMING	2/0/2/3
Nature of Course	D (Theory Application)	
Pre-requisite(s): -		
Course Objectives:		
1	To introduce students to the basics of Java programming, covering core concepts such as variables, data types, operators, control structures, loops, and arrays.	
2	To equip students with problem-solving skills by teaching them how to create and use functions, manipulate strings, and apply recursion.	
3	To train students to implement sorting and searching algorithms, and understand access modifiers for efficient data handling and application security	
4	To develop students' understanding of object-oriented programming (OOP) concepts, including classes, objects, inheritance, and polymorphism, to write structured, reusable, and maintainable code.	
5	To teach students to design scalable and robust Java applications using advanced OOP principles such as abstract classes, interfaces, and encapsulation for real-world problem-solving.	
Course Outcomes: Upon completion of the course, students shall have ability to		
C912.1	Learn the basics of Java programming, including variables, data types, loops, arrays, and decision-making structures.	U
C912.2	Develop problem-solving skills by creating functions, manipulating strings, and applying recursion to solve complex problems.	AP
C912.3	Dive into sorting, searching algorithms, and access modifiers to handle large datasets efficiently and secure your data.	AP
C912.4	Gain a solid understanding of OOP concepts like classes, objects, inheritance, and polymorphism to write clean, reusable, and maintainable code.	AN
C912.5	Design Robust Applications: Learn to design scalable Java applications using abstract classes, interfaces, and other advanced OOP principles for real-world problem-solving.	AP

JAVA FUNDAMENTALS AND MODULAR PROGRAMMING **10**

Introduction to Java: Program Structure, println() Method, Variables, Datatypes, Operators, Scanner class. Control Structures: Relational Operators, Logical Operators, Ternary Operator, if Statement, if…else Statement, if…else if Statement, switch case, for Loop, while Loop, do…while Loop. Arrays: 1-Dimensional Arrays, 2-Dimensional Arrays, Array Operations, Debug Errors.

Modular Programming: Function Definition, Arguments, Parameters, Return Value, Exception Handling, try…catch Block, finally, throw and throws. Strings: String, String Buffer, String Builder, String Operations. Sorting Searching: Sorting, Bubble Sort, Searching, Linear Search, Recursion

OBJECT-ORIENTED PROGRAMMING ESSENTIALS **10**

Introduction to Object Oriented Programming: Class, Object, Constructor, Default Constructor, Parameterized Constructor, this Keyword, Access Specifiers – private, public, protected, static, final.

INHERITANCE, POLYMORPHISM, AND INTERFACES **10**

Inheritance and Polymorphism: Inheritance Types, super Keyword, Polymorphism, Overloading, Overriding, Static Binding, Dynamic Binding. Abstract Classes and Interfaces: Abstract Class, Abstract Methods, Interface.

Project and Summative Assessment: Project Planning and Design, Solution Development, Testing and Debugging, Review and Refactoring, Project Documentation and Presentation, Summative Course-end Assessment.

TOTAL PERIODS:	**30**

Lab Component:

1	Implement a grading system that takes a student's marks and prints their grade based on a predefined scale.
2	Write a Java program to print the Fibonacci series up to n terms using a loop. Also do the same with the help of recursive function.
3	Write a program that checks if a given string is a palindrome.
4	Implement a program that reverses a string using StringBuilder or StringBuffer.
5	Implement a recursive function to calculate the greatest common divisor (GCD).
6	Write a program that takes two numbers as input and handles division by zero using try-catch-finally.
7	Create a custom exception called InvalidAgeException that gets thrown if a user enters an age below 18 for registration.
8	Create an "Employee Management System" using an abstract class for common attributes and behaviours, extended by specific employee types through inheritance. Use access modifiers for encapsulation, an interface for additional functionality like tax calculation, and demonstrate method overriding in subclasses.
9	Develop a Banking System with a base class for bank accounts, using static variables for shared attributes. Extend it to modify withdrawal behaviour via method overriding, enforce a minimum balance, and use the final keyword to restrict modifications while ensuring encapsulation.
TOTAL (LAB) PERIODS	**30**
TOTAL PERIODS	**60**

TEXT BOOKS:

1	Herbert Schildt , "Java: The Complete Reference", Twelfth Edition, McGraw-Hill Education, 2022.
2	Kathy Sierra and Bert Bates , "Head First Java", Second Edition, O'Reilly Media, 2005.

REFERENCE BOOKS:

1	Cay S. Horstmann, "Core Java Volume I – Fundamentals", Twelfth Edition, Pearson Education, 2022.
2	Joshua Bloch, "Effective Java", Third Edition, Addison-Wesley, 2018.
3	Barry A. Burd, "Beginning Programming with Java for Dummies", Eighth Edition, Wiley, 2021.

23IT911	WEB DEVELOPMENT AND UI/UX ESSENTIALS	2/0/2/3
Nature of Course	D (Theory Application)	
Pre requisites	Nil	

Course Objectives:

1.	To teach students how to build responsive websites that automatically adapt to various screen sizes and devices, ensuring a consistent and optimal user experience.
2.	To equip students with the ability to use the Bootstrap framework to rapidly develop responsive, mobile-friendly layouts with a consistent and professional look.
3.	To enable students to add interactivity to web pages using modern JavaScript (ES6), incorporating features that make web applications dynamic and engaging.
4.	To familiarize students with using generative AI tools like ChatGPT to assist in generating, optimizing, and troubleshooting for increased productivity.
5.	To train students in applying modern web development practices, including testing, debugging, and refining web pages to deliver seamless functionality and user experience.

Course Outcomes

Upon completion of the course, students shall have ability to

C911.1	Build websites that automatically adjust to different screen sizes, ensuring optimal user experience across all devices.	[AP]
C911.2	Choose the latest HTML and CSS standards to structure and style web pages, creating rich, accessible, and visually appealing designs.	[AP]
C911.3	Make use of the power of the Bootstrap framework to accelerate the development of responsive and consistent layouts.	[AP]
C911.4	Model interactive features to your pages using modern JavaScript (ES6), making your web applications engaging and dynamic.	[AP]
C911.5	Utilize ChatGPT to assist in generating HTML/CSS code and JavaScript functionality, speeding up the development process and improving productivity.	[AP]

WEBPAGE STRUCTURE, STYLING, AND RESPONSIVE DESIGN 10 Hours

Webpage: HTML Document Anatomy, Markup Language, Tags, Paired Tags, Unpaired Tags, HTML5 Elements – Heading, Paragraph, Image, Anchor, List, Live Server Deployment, Emmet Toolkit. Semantic Design: Semantic Elements, Header, Nav, Article, Footer, Section, Table, ARIA Labels. Styling: Cascading Style Sheets, CSS Selectors, CSS Properties, Page Layouts, CSS Box Model, Block Element, Inline Element, Position Properties, Z-Index, Flexbox Model. Responsive Webpage Design: Responsive Design Patterns, CSS Flexbox, CSS Grid, CSS Media Query, Bootstrap Framework, Bootstrap Grid System, Bootstrap Components, Bootstrap Utilities

JAVASCRIPT BASICS, CONTROL STRUCTURES, UNIT TESTING, ARRAYS, OBJECTS, AND API INTEGRATION 10 Hours

Introduction to JavaScript: Variables, Constants, Declarations, Hoisting, Primitive Datatypes, Literals, Template Literals, Operators, typeof operator. Control Structures: Conditional Constructs, if statement, if…else statement, if…else if statement, Nested if, switch case, Loop Constructs, for loop, while loop, do…while loop. Modular Programming: Function Definition, Function Call, Arguments, Parameters, Return Values, Variable Scope, Default Parameters, In-built Functions. Unit Testing: Unit Testing, Mocha, Chai, describe() Function, it() Function, Mocha Hooks.

Arrays and Objects: Data modelling, Object, Property, Methods, Array Structure, Array Length, Indexing, for…of Loop, Array Destructuring, Spread Operator, Array Methods, Arrow Function. JSON and REST APIs: JavaScript Object Notation – JSON, Conversions, Client-Server Architecture, Representational State Transfer – REST, HTTP, Request, Response, HTTP Methods, Status Codes

DOM MANIPULATION AND ASYNCHRONOUS PROGRAMMING **10 Hours**

DOM: Document Object Model - Elements, Properties and Methods, Events, Form Validation. JavaScript Promise Object: Synchronous, Asynchronous, Callbacks, Promise Object, Axios API.

TOTAL PERIODS	**30**

Laboratory Component:

List of Experiments

1. Create a webpage using HTML5 semantic elements to showcase two historical landmarks of your country, including a header, main content section with images, and a footer.
2. Style a navigation bar using CSS properties and selectors to include a brand logo, menu items, and a call-to-action button.
3. Implement a responsive webpage with a fixed header and footer while allowing the main content to scroll, demonstrating the use of CSS position properties and the Z-index.
4. Create a simple calculator which log the results along with their data types using the type of operator.
5. Create a function that takes a number as input and returns its factorial.
6. Write a function that checks if a given number is prime or not.
7. Create a shopping cart system where users can add, remove, and view cart items dynamically using DOM manipulation.
8. Implement form validation for a login form, ensuring the username is at least 5 characters long. Validate a password field to require at least one uppercase letter, one number, and a minimum of 8 characters. Display error messages dynamically if form validation fails, preventing submission.
9. Create a JavaScript program that fetches weather data asynchronously using the Axios API and displays temperature and conditions.
10. Create a JS program to fetch posts from https://jsonplaceholder.typicode.com/posts and display the title and body of the first 5 posts. Handle failed responses gracefully.

TOTAL (LAB) PERIODS: 30

TOTAL PERIODS: 60

Text Books:

1.	Jon Duckett, “HTML and CSS: Design and Build Websites”, Wiley, 2011.
2.	Marijn Haverbeke, “Eloquent JavaScript: A Modern Introduction to Programming”, 4th Edition, No Starch Press, 2020.

Reference Books:

1.	Jon Duckett, “JavaScript and JQuery: Interactive Front-End Web Development”, Wiley, 2014.
2.	Benjamin Jakobus, Jason Marah, “Learning Bootstrap 5”, Packt Publishing, 2021.

23IT912	BUILD SINGLE-PAGE APPLICATIONS USING REACT	2/0/2/3
Nature of Course	D (Theory Application)	
Pre requisites	Nil	

Course Objectives:

1.	To introduce students to the fundamentals of React, covering key concepts like components, JSX syntax, the virtual DOM, and declarative rendering to build dynamic single-page applications (SPAs).
2.	To teach students how to manage component state using React Hooks and retrieve data from remote servers to create interactive and dynamic application behaviour.
3.	To equip students with the skills to write and run unit tests using Jest and React Testing Library to ensure the functionality and reliability of React applications.
4.	To train students in styling React components using modern CSS techniques and Material Design principles to develop clean, responsive, and user-friendly interfaces
5.	To enable students to handle user input through forms, build flexible and validated forms using react-hook-form, and implement seamless navigation across multiple views using React Router.

Course Outcomes

Upon completion of the course, students shall have ability to

C912.1	Outline React fundamentals, including components, JSX, the virtual DOM, and declarative rendering techniques to build dynamic SPAs.	[U]
C912.2	Model component state using React Hooks and fetch data from remote servers for dynamic application behaviour.	[AP]
C912.3	Apply unit testing with Jest and React Testing Library to ensure application reliability.	[AP]
C912.4	Model React components using modern CSS techniques and Material Design for responsive, user-friendly interfaces.	[AP]
C912.5	Build user interactions with forms, create flexible forms using react-hook-form, and implement navigation with React Router for multi-page apps.	[AP]

REACT FUNDAMENTALS AND SPA STRUCTURE **10**

React Fundamentals: Components, JSX, Virtual DOM, Declarative Rendering. Static Single Page Applications (SPA): SPA vs MPA (Multi-Page Application), Structure, Design, and Layout. Rendering components: Conditional rendering, lists and keys, props

STATE MANAGEMENT AND DATA FETCHING **10**

State Management: useState, Component State, Event Handling, lift state up. Data Fetching: Remote Data Retrieval, useEffect.

STYLING, FORM HANDLING IN REACT, ADVANCED FORMS AND ROUTING **10**

Styling Components: CSS Modules, CSS in JS, Material Design. Form Handling: Controlled Components, Forms in React.

Extensible Forms: react-hook-form, Form Validation, Handling Dynamic Inputs. Routing: React Router, View Navigation.

TOTAL PERIODS	**30**

Laboratory Component:

List of Experiments

1. Create a React component that displays a personalized welcome message using JSX and updates dynamically based on user input.
2. Implement a navigation bar component with static links and explain the advantage of using reusable components in SPAs.
3. Create a button component that accepts a label as a prop and displays an alert with a custom message when clicked.
4. Write a simple unit test using Jest and React Testing Library to check if a button click updates the displayed count.
5. Lift state up by creating a parent component that tracks a selected item and updates child components when the selection changes.
6. Implement a to-do list using controlled components, where users can add, edit, and delete tasks using form inputs.
7. Create a form using react-hook-form that validates user input, ensuring the email and password fields meet specific criteria.
8. Use useRef to manage focus on an input field when a button is clicked in a form component.
9. Build a React Router setup with at least three pages (Home, About, Contact) and navigation links between them.
10. Fetch product details from an API and display them in a styled card layout using Material UI components.
11. Design a responsive navigation bar using Material UI components and ensure it adapts to different screen sizes.
12. Implement a login form using Material UI, applying proper form validation and displaying error messages when input criteria are not met.

TOTAL (LAB) PERIODS: 30

TOTAL PERIODS: 60

Text Books:

1.	Alex Banks, Eve Porcello, “Learning React: Modern Patterns for Developing React Apps”, Third Edition, O’Reilly Media, 2023
2.	Stoyan Stefanov, “React Up and Running: Building Web Applications”, O’Reilly Media, 2016.

Reference Books:

1.	Anthony Accomazzo, Nate Murray, Ari Lerner, “Fullstack React: The Complete Guide to ReactJS and Friends”, Fullstack.IO, 2020.
2.	Adam Freeman, “Pro React 16”, Apress, 2019
3.	Juho Vepsäläinen, “Testing React Applications with Jest and React Testing Library”, Leanpub, 2021.

23IT913	BUILD BACK-END APPLICATION USING SPRING BOOT / FAST API	1/0/4/3
Nature of Course	F (Theory Programming)	
Pre requisites	-	
Course Objectives:		
1.	To train students to write unit and integration tests for Java applications using JUnit 5 and Mockito, ensuring code reliability and correctness.	
2.	To equip students with the skills to manage and manipulate data effectively using Java Collections such as Lists, Sets, and Maps with generics for type-safe data handling.	
3.	To teach students to build scalable and modular Java applications using Spring Boot, applying Inversion of Control (IoC) and core annotations.	
4.	To enable students to implement data persistence in Java applications using Hibernate and JPA within Spring Boot projects for efficient database operations.	
5.	To guide students in developing RESTful APIs with Spring Boot, performing CRUD operations, validating input, and handling exceptions to build backend services.	
Course Outcomes		
Upon completion of the course, students shall have ability to		
C913.1	Apply unit and integration tests for Java applications using JUnit 5 and Mockito.	[AP]
C913.2	Model data using Java Collections like Lists, Sets, and Maps with generics	[AN]
C913.3	Build Java apps using Spring Boot using IoC and annotations with data persistence using Hibernate and JPA.	[AP]
C913.4	Develop RESTful APIs with Spring Boot, handling CRUD operations, validations, and exceptions	[AP]
C913.5	Apply MongoDB as a NoSQL data storage	[AN]

JUNIT TESTING AND JAVA COLLECTIONS **5**

JUnit Testing with Maven: Introduction to JUnit 5, adding JUnit to Maven projects, lifecycle annotations, assertion methods, parameterized tests. Java Collections: Working with Lists, Sets, and Maps using generics; ArrayList operations, iterating with Iterators, sorting, searching, and performing CRUD operations on collections. Unit Testing Collections: Writing JUnit tests for verifying collection operations.

SPRING BOOT BASICS, REST API DEVELOPMENT, HIBERNATE, JPA, AND MONGODB INTEGRATION **5**

Spring & Spring Boot Basics: Introduction to Spring Framework, benefits of Spring Boot, project setup with Spring Initializr, Inversion of Control (IoC), Dependency Injection, and core annotations. Building RESTful APIs: REST architecture, creating endpoints with @RestController, entity mapping, and integrating Spring Data JPA repositories.

Hibernate & JPA: Introduction to Hibernate, differences with JPA, configuration in Spring Boot, entity lifecycle management, Session, Session Factory, and Entity Manager usage.

CRUD Operations in REST APIs: Implementing Create, Read, Update, Delete operations, HTTP status codes, API naming conventions, and testing APIs using Postman. MongoDB Integration: Configuring Spring Boot with MongoDB, defining documents with @Document, performing CRUD operations using Mongo Repository, and executing basic queries.

EXCEPTION HANDLING AND APPLICATION TESTING **5**

Exception Handling & Input Validation: Using @Controller Advice and @Exception Handler for global error management, validating inputs with annotations like @Valid and @NotNull, creating custom exceptions, and structured error responses. Unit & Integration Testing with JUnit & Mockito: Writing unit tests for services and repositories, mocking dependencies using Mockito, and performing integration tests with @SpringBootTest.

Total Hours	**15**

Laboratory Component:

List of Experiments

1. Write unit tests using JUnit 5 to verify method's correctness, using assertions and lifecycle annotations.
2. Create a HashSet of employee IDs and a TreeSet of department names. Add, remove, and check for the presence of elements. Also, create a HashMap<String, String> mapping employee IDs to names.
3. Create a Student entity with fields id, name, and course. Configure Hibernate in your Spring Boot project and use JpaRepository to persist and retrieve student records in an H2 in-memory database.
4. Develop a REST API for managing Product records with Create, Read, Update, and Delete operations using @RestController and JpaRepository. Test your API using Postman and check the response codes.
5. Add validation to the Product entity so that the name cannot be blank and the price must be positive. Handle invalid input using @ControllerAdvice and return custom error messages for exceptions.
6. Create a Customer document class with fields id, name, and city. Configure MongoDB with Spring Boot and use MongoRepository to perform CRUD operations. Write query methods to find customers by city.
7. Write unit tests for the service layer of your Product API using Mockito to mock repository dependencies. Also, create an integration test using @SpringBootTest to test your API's end-to-end behavior.

TOTAL (LAB) PERIODS: 60

TOTAL PERIODS: 75

Text Books:	
1.	Craig Walls, “Spring Boot in Action”, Manning Publications, 2016.
2.	Brandon Pepin, “Beginning Spring Boot 3: Build and Deploy Applications with Spring Boot 3, Spring Cloud, and Spring Native”, Apress, 2023.
Reference Books:	
1.	Craig Walls, “Spring in Action”, Sixth Edition, Manning Publications, 2022.
2.	Felipe Gutierrez, “Pro Spring Boot 3”, Apress, 2023.
3.	Christian Bauer, Gavin King, “Java Persistence with Hibernate”, Second Edition, Manning Publications, 2015.