Shri Vile Parle Kelavani Mandal's

Dwarkadas J. Sanghvi College of Engineering

(Autonomous College Affiliated to the University of Mumbai)

Course Structure and Syllabus

of

Final Year Engineering

in

Computer Engineering

Prepared by:- Board of Studies in Computer Engineering

Recommended by:- Academic Council of D. J. Sanghvi College of Engineering

Approved by:- Governing Body of D. J. Sanghvi College of Engineering

Revision: 3 (2023)

With effect from the Academic Year: 2026-2027

B. TECH. IN COMPUTER ENGINEERING SEM - VII SCHEME

Sr. No.	Course Code	Course	Teaching Scheme				Semester End Examination (SEE) - A						Continuous Assessment (CA) - B						Aggregate (A+B)	Credits Earned	
			Theory (Hrs)	Practical (Hrs)	Tutorial (Hrs)	Credits	Duration (Hrs)	Theory	Oral	Pract	Oral & Pract	SEE Total	Term Test 1	Term Test 2	Term Test 3	Term Test Total (TT1	Term Work	CA Total (B)			
1	DJS23CCPC701	Natural Language Processing	3	--	--	3	2	60	--	--	--	60	15	15	10	40	- -	40	100	3	4
	DJS23CLPC701	Natural Language Processing Laboratory	--	2	--	1	2	--	25	--	--	25	- -	- -	- -	- -	25	25	50	1	
#2	DJS23CCPE711	Agentic AI	3	--	--	3	2	60	--	--	--	60	15	15	10	40	- -	40	100	3	4
	DJS23CLPE711	Agentic AI Laboratory	--	2	--	1	2	--	25	--	--	25	- -	- -	- -	- -	25	25	50	1	
	DJS23CCPE712	Distributed Computing	3	--	--	3	2	60	--	--	--	60	15	15	10	40	- -	40	100	3	4
	DJS23CLPE712	Distributed Computing Laboratory	--	2	--	1	2	--	25	--	--	25	- -	- -	- -	- -	25	25	50	1	
	DJS23CCPE713	Metaverse	3	--	--	3	2	60	--	--	--	60	15	15	10	40	- -	40	100	3	4
	DJS23CLPE713	Metaverse Laboratory	--	2	--	1	2	--	25	--	--	25	- -	- -	- -	- -	25	25	50	1	
#3	DJS23CCPE714	Social Network Analysis	3	--	--	3	2	60	--	--	--	60	15	15	10	40	- -	40	100	3	4
	DJS23CLPE714	Social Network Analysis Laboratory	--	2	--	1	2	--	25	--	--	25	- -	- -	- -	- -	25	25	50	1	
	DJS23CCPE715	Virtual Infrastructure Systems	3	--	--	3	2	60	--	--	--	60	15	15	10	40	- -	40	100	3	4
	DJS23CLPE715	Virtual Infrastructure Systems Laboratory	--	2	--	1	2	--	25	--	--	25	- -	- -	- -	- -	25	25	50	1	
	DJS23CCPE716	Internet of Everything	3	--	--	3	2	60	--	--	--	60	15	15	10	40	- -	40	100	3	4
	DJS23CLPE716	Internet of Everything Laboratory	--	2	--	1	2	--	25	--	--	25	- -	- -	- -	- -	25	25	50	1	
4	DJS23CCMD701	Vulnerability Assessment and Penetration Testing	2	--	--	2	2	60	--	--	--	60	15	15	10	40	- -	40	100	2	2
5	DJS23ITELX12	Research Ecosystem	--	--	2	2	--	--	--	--	--	--	- -	- -	- -	- -	50	50	50	2	2
6	DJS23IPELX13	Major Project	--	8	--	4	2	--	--	--	50	50	- -	- -	- -	- -	50	50	100	4	4
7	DJS23ICHSX11	Disaster Management and Preparedness	2	--	--	--	--	--	--	--	--	--	- -	- -	- -	- -	--	--	--	--	--
8	DJS23CCSC701	AI for Language Generation and Understanding	1	--	--	1	--	--	--	--	--	--	15	15	10	40	--	40	40	1	1
		Total	**14**	**14**	**2**	**21**	**16**	**240**	**75**	**0**	**50**	**365**	**75**	**75**	**50**	**200**	**175**	**375**	**740**	**21**	

B. TECH. IN COMPUTER ENGINEERING SEM - VIII SCHEME

Sr. No.	Course Code	Course	Teaching Scheme				Semester End Examination (SEE) - A						Continuous Assessment (CA) - B						Aggregate (A+B)	Credits Earned	
			Theory (Hrs)	Practical (Hrs)	Tutorial (Hrs)	Credits	Duration (Hrs)	Theory	Oral	Pract	Oral & Pract	SEE Total	Term Test 1	Term Test 2	Term Test 3	Term Test Total (TT1	Term Work	CA Total (B)			
#1	DJS23CCPC811	Quantum Computing	3	--	--	3	2	60	--	--	--	60	15	15	10	40	- -	40	100	3	3
	DJS23CCPC812	Human Computer Interaction	3	--	--	3	2	60	--	--	--	60	15	15	10	40	- -	40	100	3	
	DJS23CCPC813	Geographic Information System	3	--	--	3	2	60	--	--	--	60	15	15	10	40	- -	40	100	3	
2	DJS23IPELX14	Internship (OJT) cum Project	--	--	--	18	2	--	--	--	200	200	--	--	--	--	400	400	600	18	18
3	DJS23CCSC801	Multi-Agent Systems	1	--	--	1	--	--	--	--	--	--	15	15	10	40	--	40	40	1	1
		Total	**4**	**36**	**0**	**22**	**4**	**60**	**0**	**0**	**200**	**260**	**30**	**30**	**20**	**80**	**400**	**480**	**740**	**22**	

# indicates any one subject

Prepared by Checked by Head of the Department Principal

Continuous Assessment (A):

Course	Assessment Tools	Marks	Time (mins)
Theory	a. Term Test 1 (based on 40 % syllabus)	15	45
	b. Term Test 2 (on next 40 % syllabus)	15	45
	c. Assignment / course project / group discussion / presentation / quiz/ any other.	10	--
	Total marks (a + b + c)	40	--
Audit course	Performance in the assignments / quiz / power point presentation / poster presentation / group project / any other tool.	--	As applicable
Laboratory	Performance in the laboratory and documentation.	25	
Tutorial	Performance in each tutorial & / assignment.	25	
Laboratory &Tutorial	Performance in the laboratory and tutorial.	50	

The final certification and acceptance of term work will be subject to satisfactory performance upon fulfilling minimum passing criteria in the term work / completion of audit course.

Semester End Assessment (B):

Course	Assessment Tools	Marks	Time (hrs.)
Theory / * Computer based	Written paper based on the entire syllabus.	60	2
	* Computer based assessment in the college premises.		
Oral	Questions based on the entire syllabus.	25	As applicable
Practical	Performance of the practical assigned during the examination and the output / results obtained.	25	2
Oral & Practical	Project based courses - Performance of the practical assigned during the examination and the output / results obtained. Based on the practical performed during the examination and on the entire syllabus.	As per the scheme	2

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **Final Year B.Tech.** **Semester: VII**

Course: Natural Language Processing (DJS23CCPC701)

Course: Natural Language Processing Laboratory (DJS23CLPC701)

Pre-requisite:

1. Formal Languages and Automata Theory
2. Python Programming
3. Machine Learning

Objectives:

1. Understand the theoretical foundations and basic principles of Natural Language Processing.
2. Apply preprocessing techniques, syntactic and semantic analysis to process and analyze text data.
3. Explore statistical and machine learning methods for solving NLP tasks such as classification, tagging, and translation.
4. Design and implement NLP applications using modern tools and frameworks with consideration of ethical issue

Outcomes: On completion of the course, the learner will be able to:

1. Understand stages of NLP and apply morphological analysis techniques on any real-world text.
2. Apply appropriate techniques such as tokenization, POS tagging and syntactic parsing to real-world text.
3. Apply feature extraction, word sense disambiguation and similarity for semantic analysis of text.
4. Apply pragmatics and discourse segmentation to understand structure and context of text.

Natural Language Processing (DJS23CCPC701)		
Unit	**Description**	**Duration**
1	**Introduction**: Levels of NLP, Knowledge in Speech and Language Processing, Ambiguity in Natural Language, Challenges Of NLP, Regular Expression, Finite Automata, Finite State Transducers (FST), Morphology Analysis: Survey of English Morphology, Inflectional Morphology & Derivational Morphology, Lemmatization, Morphological Parsing With FST, Lexicon Free FST Porter Stemmer, Word and Sentence Tokenization, Detection and Correction of Errors, Minimum Edit Distance with Backtracking. **Self-learning Topics:** Survey of modern NLP applications (ChatGPT, Gemini, Claude, Perplexity), Tools: Hugging Face Playground, OpenAI Playground	5
2	**Word Level Analysis**: N – Grams, Unigrams/Bigrams Language Models, Corpora, Computing the Probability of Word Sequence, Training and Testing. Perplexity And Entropy: Smoothing and Backup, Zipf's Law, Add One Smoothing **Self-learning Topics:** Modern Tokenization Techniques for NLP Byte Pair Encoding (BPE), WordPiece and SentencePiece Tokenization, Tools/Frameworks: Hugging Face Tokenizers, SentencePiece	6
3	**Syntax Analysis**: Part-Of-Speech Tagging (POS) - Open and Closed Words. Tag Set for English (Penn Treebank), Rule Based POS Tagging,	8

	Transformation Based Tagging, Stochastic POS Tagging and Issues –Multiple Tags & Words, Unknown Words. Hidden Markov Model (HMM), Maximum Entropy, And Conditional Random Field (CRF). Context-Free Grammars, Normal Forms for grammar Parsing with CFG: Top-down parsing, Bottom-up parsing, Basic top-down parser, Problems with the Basic Top-down Parser: Left Recursion, Ambiguity Earley parser, Probabilistic CFG, Probabilistic CYK **Self-learning Topics:** Dependency Parsing and Universal Dependencies (UD) using spaCy or Stanza	
4	**Semantic Analysis**: Lexical Semantics, Attachment for Fragment of English-Sentences, Noun Phrases, Verb Phrases, Prepositional Phrases, Relations Among Lexemes & Their Senses: Homonymy, Polysemy, Synonymy, Hyponymy, WordNet, Feature Extraction and Word Embeddings: Vector semantics, Traditional Word Embedding: TF-IDF Static Word Embedding: Word2Vec - Continuous Bag of Words (CBOW), Skipgram, GloVe Word Sense Disambiguation: WSD using Indirect approaches: selectional restriction-based disambiguation, selectional association-based disambiguation, WSD using Direct approaches: Supervised, Unsupervised, Bootstrapping methods, Dictionary based: Lesk algorithm **Self-learning Topics:** Contextual Word Embeddings (ELMo, BERT Embeddings, Sentence-BERT) using Sentence Transformers	8
5	**Pragmatics & Discourse Segmentation**: Introduction to Discourse, Cohesion and Coherence in Discourse, Discourse connectives, Reference Resolution, Reference Phenomena, Features for Pronominal Anaphora Resolution, Anaphora Resolution: Hobbs algorithm, Centering algorithm, Log-linear model, Co-reference resolution, Evaluation of Coreference resolution **Self-learning Topics:** Knowledge Graphs and Semantic Web for NLP (RDF, OWL, DBpedia, Wikidata)	7
6	**NLP application for text classification:** NLP application pipeline, Evaluating Classification Model, Develop a text classification application, **Transformers**: Encoder-Decoder Framework, Attention Mechanisms, Transformer architecture, Challenges with Transformers, Transfer Learning in NLP. **Self-learning Topics:** Prompt Engineering and Foundation Model Applications using Hugging Face Hub, Ollama, or LM Studio	8
	Total	**42**

Natural Language Processing Laboratory (DJS23CLPC701)	
Exp.	**Suggested Experiments**
1	Study and Critical Analysis of a Research Paper on NLP Applications.
	To collect a text corpus from various sources and perform preprocessing tasks such as tokenization, stop-word removal, and cleaning.
2	To implement text normalization techniques including stemming, lemmatization, case folding, and lexical processing for improved text representation.
3	To construct and analyze N-Gram language models for predicting word sequences and estimating language probabilities.
4	To develop and evaluate a rule-based Part-of-Speech (POS) tagging system using linguistic rules and patterns.

5	To implement a statistical POS tagging system using Hidden Markov Models (HMM) and assess its tagging accuracy.
6	To design and apply Context-Free Grammar (CFG) rules for syntactic parsing and sentence structure analysis.
7	To perform probabilistic parsing using statistical parsing techniques for determining the most likely parse tree of a sentence.
8	To conduct semantic analysis using WordNet for exploring lexical relationships such as synonymy, antonymy, and hypernymy.
9	To extract textual features and represent words and documents using vector semantics techniques.
10	To generate word embeddings and perform Word Sense Disambiguation (WSD) for understanding contextual word meanings.
11	To analyze discourse structures and implement coreference resolution for identifying entity references across text.
12	To develop and evaluate machine learning models for automated text classification tasks.
13	To implement transformer-based text classification using transfer learning techniques and evaluate model performance on benchmark datasets.

Batchwise laboratory work of minimum 10 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Batchwise tutorial sessions are to be conducted on topics which would help the learner to identify/analyze the problem and to apply problem solving techniques learnt.

Books Recommended:

Text Books:

1. Tanmoy Chakraborty, Introduction to Large Language Models, Generative AI for Text. Wiley, 2024
2. Denis Rothman, Transformers for Natural Language Processing, Packt Publishing, 2nd Edition, 2022
3. Speech and Language Processing, 2nd Edition, Jurafsky and Martin, Prentice Hall, 2000

Reference Books:

1. Jurafsky, D., & Martin, J. H., Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models (3rd ed., draft), Self-published, 2025.
2. Manning and Schutze, "Statistical Natural Language Processing", MIT Press; 1 st edition, 1999.
3. James Allen. Natural Language Understanding. The Benajmins/Cummings Publishing Company Inc. 1994.
4. Tom Mitchell. Machine Learning. McGraw Hill, 1997.
5. Frederick Jelinek, "Statistical Methods of Speech Recognition", MIT Press, 1997.

Web Resources:

1. Natural Language Processing, IIT Kharagpur
   https://nptel.ac.in/courses/106105158
2. Applied Natural Language Processing
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc20_cs87

Program: Computer Engineering **Final Year B.Tech.** **Semester: VII**

Course: Agentic AI (DJS23CCPE711)

Course: Agentic AI Laboratory (DJS23CLPE711)

Pre-requisite:

1. Artificial Intelligence
2. Machine Learning

Objectives:

1. Understand the fundamental concepts, architectures, and lifecycle of Agentic AI systems.
2. Apply LLMs, agent frameworks, memory systems, and RAG techniques to develop intelligent agents.
3. Analyze and evaluate multi-agent systems, agent performance, deployment strategies, and real-world applications.

Outcomes: On completion of the course, the learner will be able to:

1. Explain the concepts, architectures, and operational characteristics of Agentic AI systems.
2. Develop LLM-based agents using prompt engineering, agent frameworks, tool integration, and memory mechanisms.
3. Analyze memory retrieval, RAG techniques and multi-agent collaboration for autonomous problem solving.
4. Examine the performance, limitations and application scenarios of Agentic AI systems.

Agentic AI (DJS23CCPE711)		
Unit	**Description**	**Duration**
1	**Evolution of Agentic AI:** Evolution from traditional AI and Large Language Models (LLMs) to Agentic AI, Intelligent agents and the Perception–reasoning–action cycle **Agent architectures** and components, Agent lifecycle, capabilities, and levels of autonomy, Types of Agents: Reactive, deliberative, hybrid, task-oriented and goal-oriented agents **Agent design principles** and task decomposition, ReAct Framework, fundamentals of multi-agent systems, Comparison of rule-based systems and agent-based systems Real-world applications of Agentic AI **Self-learning Topics:** Survey of Commercial AI Agents, Comparative Study of AI Assistants (ChatGPT, Gemini, Claude, Copilot, Perplexity)	5
2	**Large Language Models and Agent Development:** Introduction to Large Language Models (LLMs), role of LLMs in Agentic AI, agents versus chains, prompting techniques - zero-shot, few-shot, chain-of-thought (CoT), and role prompting, structured prompting for controlling agent behavior and outputs, tool integration and function calling	9

	Agent development frameworks: LangChain, LangGraph, CrewAI, AutoGen and Hugging Face Agents, Development of simple LLM-based agents. **Self-learning Topics:** Low-Code/No-Code AI Agent Development Platforms (n8n, Flowise, Dify, Langflow, PydanticAI)	
3	**Memory, Retrieval and RAG Systems for AI Agents:** **Agent Memory Architectures:** Short-term, long-term, episodic, and semantic memory, Embeddings and Vector Representations, Text chunking and similarity techniques **Vector Stores and Knowledge Retrieval:** Storage and retrieval of contextual knowledge, Concepts of semantic search **Retrieval-Augmented Generation (RAG):** RAG architecture and retrieval-augmented memory loops, Agentic RAG, Query transformation, hybrid search, and re-ranking; Context compression and context engineering strategies, Evaluation of retrieval and RAG performance. **Self-learning Topics:** Advanced Retrieval Techniques (GraphRAG, Multimodal RAG and Long-Context LLMs)	9
4	**Multi-Agent Systems and Agent Collaboration:** Fundamentals of multi-agent systems, agent collaboration and coordination, communication between agents, agent roles and responsibilities; distributed problem solving and task allocation, cooperative and competitive agent behaviors **Multi-Agent Architectures:** Planner–executor and supervisor–worker architectures, multi-agent workflows **Self-learning Topics:** AI Agent Security, Governance and Trustworthy Multi-Agent Systems	8
5	**Agent Evaluation and Deployment:** **Agent Evaluation Metrics:** Success metrics, autonomy, correctness, reliability, and robustness, self-reflection, critique, and iterative improvement, benchmarking and performance assessment of AI agents, Human-in-the-Loop (HITL) systems and approval workflows **Agent Deployment Fundamentals:** Basics of agent deployment and execution environments; monitoring and maintenance of agent systems; limitations, challenges, and failure modes of autonomous agents. **Self-learning Topics:** LLMOps, AgentOps and AI Observability Platforms (LangSmith, LangFuse, Phoenix)	6
6	**Applications and Future Trends of Agentic AI:** Applications of Agentic AI in healthcare, finance, education, manufacturing, and smart cities; case studies of AI assistants, research agents, coding agents, and enterprise AI agents; emerging trends in multimodal agents, autonomous workflows **Self-learning Topics:** Need for Agent Orchestration in Complex AI Systems, Basic Orchestration Architectures and Coordination Strategies	5
	Total	**42**

Agentic AI Laboratory (DJS23CLP711)	
Exp.	**Suggested experiments**
1	**Setup and Configuration of Agentic AI Development Environment** Set up a Python-based development environment for building LLM agents, including API key configuration, virtual environment management, and a first call to an LLM endpoint to verify connectivity. *Tools: Python 3.x, venv/conda, OpenAI/Anthropic API or Ollama (local), VS Code, Jupyter Notebook.*
2	**Design of Reactive and Goal-Oriented Agents** Implement and compare a simple reactive agent (stimulus–response) with a goal-oriented agent that selects actions to achieve a defined objective in a small environment. *Tools: Python, OpenAI/Anthropic API, simple grid/task environment.*
3	**Implementation of ReAct-based Agent** Build an agent using the ReAct (Reasoning + Acting) paradigm that interleaves chain-of-thought reasoning with tool actions to solve a multi-step query. *Tools: Python, LangChain, an LLM API, a search/calculator tool.*
4	**Prompt Engineering Techniques for Agents** Apply and evaluate zero-shot, few-shot, chain-of-thought, and role-based prompting strategies, observing their effect on agent reasoning quality and output reliability. *Tools: Python, LLM API, prompt templates.*
5	**Tool Calling and Function Integration** Expose custom Python functions (e.g., calculator, weather, database lookup) as callable tools and enable the LLM to invoke them via structured function calling. *Tools: Python, OpenAI/Anthropic function-calling API, custom tool functions.*
6	**Development of a LangChain-based Agent** Construct an agent using LangChain components — chains, prompts, and a tool-equipped agent executor — to autonomously answer queries requiring external tools. *Tools: Python, LangChain, LLM API, LangChain tools (Search, Python REPL).*
7	**Development of a Simple LangGraph Workflow** Design a stateful agent workflow as a graph with nodes and conditional edges using LangGraph, demonstrating branching and looping control flow. *Tools: Python, LangGraph, LangChain, LLM API.*
8	**Implementation of Agent Memory using Conversation History** Add short-term and summary-based memory to a conversational agent so it retains context across turns, and observe behaviour with and without memory. *Tools: Python, LangChain memory modules, LLM API.*
9	**Embeddings and Semantic Search using Vector Stores** Generate text embeddings, store them in a vector database, and perform semantic similarity search over a document corpus. *Tools: Python, embedding model (OpenAI/Sentence-Transformers), FAISS/ Chroma/ Pinecone.*
10	**Development of a Basic RAG System** Build a Retrieval-Augmented Generation pipeline that retrieves relevant chunks from a knowledge base and grounds the LLM's response in the retrieved context. *Tools: Python, LangChain, vector store (Chroma/FAISS), LLM API.*
11	**Query Expansion and Context Optimization in RAG**

	Improve a baseline RAG system using query expansion/rewriting, re-ranking, and chunking strategies, and compare answer relevance before and after optimization. *Tools: Python, LangChain, re-ranker (e.g., Cohere/BGE), vector store, LLM API.*
12	**Development of a Two-Agent Collaborative System using CrewAI** Create two role-specialized agents (e.g., researcher and writer) that collaborate on a shared task with defined roles, goals, and task delegation. *Tools: Python, CrewAI, LLM API, web search tool.*
13	**Implementation of Planner–Executor Multi-Agent Workflow** Design a multi-agent system where a planner agent decomposes a complex task into sub-tasks and an executor agent carries them out, coordinating through a shared state. *Tools: Python, LangGraph/CrewAI/AutoGen, LLM API.*
14	**Evaluation of Agent Performance** Evaluate agent and RAG outputs using metrics such as task success rate, faithfulness, relevance, latency, and token cost, using an automated evaluation framework. *Tools: Python, RAGAS/DeepEval/LangSmith, LLM API.*

Batchwise laboratory work of minimum 10 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Batchwise tutorial sessions are to be conducted on topics which would help the learner to identify/analyze the problem and to apply problem solving techniques learnt.

Books Recommended:

Text Books:

1. Michael Lanham, GPT Agents in Action, Manning Publications; 1st Edition, 2024.
2. Roberto Infante, AI Agents and Applications, Manning Publications, 2026.
3. Elliot R. Stroud, Agentic AI with LangChain and LangGraph, Independently Published, 2025.

Reference Books:

1. Srinivasan Venkataraman and Harrison Chase, Building Agentic AI Systems: A Developer's Guide to LLM Agents, LangChain and LangGraph, O'Reilly Media, 2024.
2. Sinan Ozdemir, LLM Engineers' Handbook: Retrieval-Augmented Generation, Prompting, Agents, and Production AI, Packt Publishing, 2024.
3. Jerry Liu, AI Engineering: Building Applications with Foundation Models, O'Reilly Media, 2024.
4. Tom Taulli and Gaurav Deshmukh, Building Generative AI Agents: Using LangGraph, AutoGen, and CrewAI, Apress, 2025.
5. Olivier Blais, Practical GenAI: Building Real-World Generative AI Products with LLMs, RAG, Agents, and Workflows, O'Reilly Media, 2024.
6. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Pearson Education; 4th Edition, 2021.

Web Resources:

1. Applications of GenAI and Agentic AI in Business
   https://elearn.nptel.ac.in/product-page/applications-of-genai-and-agentic-ai-in-business
2. https://www.youtube.com/watch?v=nOJrNwhokw4

Program: Computer Engineering **Final Year B.Tech.** **Semester: VII**

Course: Distributed Computing (DJS23CCPE712)

Course: Distributed Computing Laboratory (DJS23CLPE712)

Pre-requisite:

1. Computer Networks
2. Operating Systems
3. Database Management Systems

Objectives:

1. To provide students with contemporary knowledge in distributed systems.
2. To equip students with skills to analyze and design distributed applications.
3. To provide master skills to measure the performance of distributed synchronization algorithms

Outcomes: On completion of the course, the learner will be able to:

1. Understand the fundamental concepts, architectures, elements, and middleware technologies used in distributed systems, including RPC, RMI, and object-based middleware.
2. Analyze synchronization techniques in distributed systems, including clock synchronization, mutual exclusion, and related synchronization algorithms.
3. Understand resource management, process management, consistency models, and replication management in distributed computing environments.
4. Apply the concepts of distributed file systems to analyze and compare systems such as NFS and AFS and understand the design of large-scale distributed applications.

Distributed Computing (DJS23CCPE712)		
Unit	**Description**	**Duration**
1	**Introduction to Distributed Systems:** Characterization of Distributed Systems: Issues, Goals, and Types of distributed systems, Distributed System Models, Hardware concepts, Software Concept. Middleware: Models of Middleware, Services offered by middleware, Client Server model. **Self-learning Topics:** Microservices architecture and the service mesh, Service-Oriented Architecture (SOA) vs. microservices, Serverless and Function-as-a-Service (FaaS) models	4
2	**Communication:** Layered Protocols, Interprocess communication (IPC): MPI, Remote Procedure Call (RPC), Remote Object Invocation, Remote Method Invocation (RMI) Message Oriented Communication, Stream Oriented Communication, Group Communication **Self-learning Topics:** gRPC and Protocol Buffers, Event-driven architecture and event sourcing, Apache Kafka and distributed stream processing	8
3	**Synchronization:** Clock Synchronization, Logical Clocks, Election Algorithms, Mutual Exclusion, Distributed Mutual Exclusion-Classification of mutual Exclusion Algorithm, Requirements of Mutual Exclusion Algorithms, Performance measure. Non Token based Algorithms: Lamport Algorithm, Ricart–Agrawala's Algorithm, Maekawa's Algorithm	8

	Token Based Algorithms: Suzuki-Kasami's Broardcast Algorithms, Singhal's Heurastic Algorithm, Raymond's Tree based Algorithm, Comparative Performance Analysis. **Self-learning Topics:** Distributed coordination with ZooKeeper and etcd, Hybrid logical clocks and Google Spanner's TrueTime, Conflict-free Replicated Data Types (CRDTs)	
4	**Resource and Process Management:** Desirable Features of global Scheduling algorithm, Task assignment approach, Load balancing approach, load sharing approach Introduction to process management, process migration, Threads, Virtualization, Clients, Servers, Code Migration **Self-learning Topics:** Container orchestration and Kubernetes scheduling, MapReduce and the Bulk Synchronous Parallel (BSP) model, Cluster resource managers (Apache Mesos, YARN)	8
5	**Consistency, Replication and Fault Tolerance:** Introduction to replication and consistency, Data-Centric and Client-Centric Consistency Models, Replica Management. Fault Tolerance: Introduction, Process resilience, Reliable client-server and group communication, Recovery **Self-learning Topics:** Modern consensus: Raft, Multi-Paxos, and PBFT, Distributed transactions: 2PC, 3PC, and the Saga pattern, Blockchain consensus (PoW, PoS) as distributed agreement, Chaos engineering for resilience testing	7
6	**Distributed File Systems and Name Services:** Introduction and features of DFS, File models, File Accessing models, File-Caching Schemes, File Replication Case Study: Distributed File Systems (DSF), Network File System (NFS), Andrew File System (AFS). Trends in Distributed Computing: Edge Computing, Cloud Computing, Fog Computing **Self-learning Topics: Distributed databases:** NoSQL, NewSQL, and sharding, Object storage and key-value stores (Dynamo, Cassandra), Distributed caching (Redis, Memcached), Distributed data processing with Apache Spark	7
	Total	**42**

Distributed Computing Laboratory (DJS23CLPE712)	
Exp.	**Suggested experiments**
1	To implement Client/server programming using RPC/RMI.
2	To implement a multi-thread application.
3	To implement Inter-process communication
4	To implement Group Communication
5	To implement Load Balancing Algorithm.
6	To implement Name Resolution protocol.

7	To implement the Election Algorithm.
8	To implement Clock Synchronization algorithms.
9	To implement Mutual Exclusion Algorithm.
10	To implement Deadlock management in Distributed systems
11	To implement Distributed File System
12	To implement a Case study on CORBA

Batchwise laboratory work of minimum 10 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Text Books:

1. Andrew S. Tanenbaum and Maarten Van Steen, ―Distributed Systems: Principles and Paradigms, 4th edition, Pearson Education, 2023.
2. George Coulouris, Jean Dollimore, Tim Kindberg, , "Distributed Systems: Concepts and Design", 4th Edition, Pearson Education, 2005.

Reference Books:

1. Shraddha Zanzat, Bhavana Karmore, Vishwajit Barbudhe, Distributed Systems, Notion Press, 2020.
2. M. L. Liu, ―Distributed Computing Principles and Applications‖, Pearson Addison Wesley, 2004.

Web Resources:

1. Distributed Systems
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc21_cs87
2. Cloud Computing and Distributed Systems
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs29

Program: Computer Engineering **Final Year B.Tech.** **Semester: VII**

Course: Metaverse (DJS23CCPE713)

Course: Metaverse Laboratory (DJS23CLPE713)

Pre-requisite:

1. Computer Graphics
2. Computer Vision
3. Artificial Intelligence

Objectives:

1. To introduce the Metaverse, its architecture, and enabling immersive technologies (VR/AR/MR).
2. To impart working knowledge of Virtual Reality and Blockchain/Web3 as applied to the Metaverse.
3. To develop multi-user, AI-augmented metaverse applications using open-source tools.
4. To examine security, privacy, ethical, and governance issues in metaverse deployments.
5. To build Digital Twins and Industrial Metaverse solutions and analyse their cyber-physical security requirements.

Outcomes: On completion of the course, the learner will be able to:

1. Explain the Metaverse architecture, ecosystem, and the role of VR/AR/MR.
2. Apply VR, spatial computing, and Blockchain/Web3 concepts to immersive worlds.
3. Design and develop multi-user, AI-driven metaverse applications using open-source tools.
4. Evaluate security, privacy, ethical, and governance challenges across consumer and industrial metaverse deployments.
5. Design Digital Twin and Industrial Metaverse solutions and assess their cyber-physical security requirements.

Metaverse (DJS23CCPE713)		
Unit	**Description**	**Duration**
1	**Introduction to the Metaverse** Definition, evolution from Web 1.0 → 3.0 and the Spatial Web; characteristics — persistence, interoperability, embodiment, real-time interaction, creator economy. **Seven-layer Metaverse architecture** (Infrastructure, Human Interface, Decentralization, Spatial Computing, Creator Economy, Discovery, Experience). Open vs walled-garden metaverses; consumer vs Industrial Metaverse. **Reference platforms**: Decentraland, Roblox, Horizon Worlds, NVIDIA Omniverse. **Standards**: Metaverse Standards Forum, OMA3, Khronos. **Self-learning Topics:** Tokenomics and virtual-economy modeling of metaverse platforms, Sustainability and carbon footprint of metaverse infrastructure (the "green metaverse"), The "Metaverse OS" vision and cross-platform asset portability	6
2	**Virtual Reality and Spatial Computing for the Metaverse** **VR concepts:** immersion, presence, simulator sickness; VR hardware — HMDs, controllers, haptics, eye/hand tracking; 3DoF vs 6DoF; tracking systems (inside-out, outside-in). **AR/MR and the XR continuum;** OpenXR and WebXR.	7

	Stereoscopic, foveated, and asynchronous time-warp rendering; motion-to-photon latency. **Spatial computing**: world anchoring, persistent maps, SLAM, plane detection, occlusion (extending Computer Vision). Spatial audio (HRTF, ambisonics). **Interaction techniques**: gaze, gesture, voice. **Self-learning Topics:** Varifocal, holographic, and light-field displays for VR/AR, Neural rendering and real-time photorealistic reconstruction, Full-body haptics, electrical muscle stimulation, and the tactile internet	
3	**Avatars and Multi-User Virtual Worlds** **Avatar types**: stylized, photorealistic, codec avatars. **Vision-driven avatar capture**: facial landmarks, blendshapes, pose and hand tracking, motion retargeting. Inverse kinematics, lip-sync, gaze, emotion transfer. Avatar interoperability: VRM, Ready Player Me, glTF, USD avatar schemas. **Multi-user system architecture**: client–server, P2P, hybrid; authoritative servers, dead reckoning, interest management, state synchronization. Edge/5G and cloud rendering. Social presence, embodiment, accessibility, harassment-resistant design. **Self-learning Topics:** Codec avatars and neural compression for photorealistic telepresence, Photogrammetry and MetaHuman-style digital human creation pipelines, Affective computing for emotion-driven, empathetic avatars	6
4	**AI and Generative Content in the Metaverse** Role of AI in scaling content and population. **LLM-driven NPCs and intelligent agents**; agentic loops, memory, goal-directed behaviour. Multimodal interaction: ASR, TTS, gesture/gaze recognition, emotion estimation. **Generative 3D**: text-to-image, text-to-3D, NeRF, Gaussian splatting, procedural world generation. Animation synthesis and real-time translation for avatars. AI-based content moderation. Latency, cost, and grounding challenges of LLMs in real-time worlds. **Self-learning Topics:** World models and 3D foundation models for generating explorable environments, Generative agent societies (autonomous, memory-driven multi-agent simulations), Diffusion-based 4D (dynamic 3D) content generation	7
5	**Blockchain, Web3, Privacy, Ethics and Governance** **Blockchain concepts**: decentralization, consensus, smart contracts, EVM-compatible chains. Tokens and NFTs (ERC-20/721/1155) for avatars, land and items; crypto wallets and Web3 authentication. Decentralized storage (IPFS), Decentralized Identity (DID) and DAOs for governance. **Security and privacy threats**: avatar impersonation, deepfakes, virtual harassment; privacy of immersive telemetry (gaze, gait, room geometry). **Regulations**: GDPR, India's DPDP Act, child-safety norms. Governance models and wellbeing concerns. **Future trends**: BCI, 6G, AR cloud. **Self-learning Topics:** Zero-knowledge proofs for privacy-preserving Web3 identity, Soulbound tokens (SBTs) and reputation systems in the metaverse, Post-quantum threats to blockchain-based metaverse assets	8
6	**Digital Twins, the Industrial Metaverse and Cyber-Physical Security** **Digital Twins**: digital model vs shadow vs twin; five-dimension twin model; real-time synchronization and predictive maintenance. Enabling stack — IoT sensing,	8

	MQTT/OPC-UA pipelines, simulation engines (NVIDIA Omniverse, Azure Digital Twins). **Standards**: ISO 23247, DTDL, Asset Administration Shell (AAS). **Industrial Metaverse**: distinction from the consumer metaverse; cyber-physical systems and the Industry 4.0/5.0 link; use cases — smart factory, smart-city, energy-grid and healthcare twins. **Cyber-Physical and OT Security**: IT/OT convergence; Purdue model, IEC 62443, network segmentation and zero-trust; twin-specific threats (false-data injection, sensor spoofing, replay, twin–physical desynchronization); securing pipelines (MQTT/OPC-UA over TLS); the safety–security interplay. **Self-learning Topics:** Federated digital twins and "twin-of-twins" architectures, Urban-scale city digital twins for planning and disaster simulation, AI-driven anomaly detection and self-healing in OT/cyber-physical systems, 6G and the tactile internet as enablers of the industrial metaverse	
	Total	**42**

Metaverse Laboratory (DJS23CLPE713)	
Exp.	**Suggested experiments**
1	Build a basic VR scene in the browser with primitive shapes, lighting, and 6DoF navigation.
2	Add spatial (3D positional) audio to a Three.js / A-Frame scene — attach a sound source to a 3D object and demonstrate distance attenuation and HRTF-based directionality as the user moves.
3	Create a markerless AR app that places a 3D model on a detected surface using the webcam.
4	Model a 3D object in Blender, export as glTF, and load it into a Three.js / A-Frame scene.
5	Implement hand-tracking / gesture interaction in a WebXR scene so the user can grab, move, and release a virtual object using tracked hands or controllers.
6	Create a 3D avatar using Ready Player Me / VRoid and import it into a web scene with idle animation.
7	Drive an avatar's face / pose using webcam landmarks.
8	Build a multi-user shared room where two clients see each other's avatars move in real time.
9	Add a chatbot/LLM-driven Non-Player Characters (NPC) to a 3D scene that answers user questions via text.
10	Write, deploy, and test a simple NFT smart contract for a metaverse asset on a test network.
11	Connect a Web3 wallet (MetaMask) to a 3D web scene to authenticate a user and display owned NFTs.
12	Generate a 3D asset or PBR texture using an open-source text-to-image / text-to-3D tool, then import and place it in a web scene — illustrating AI-driven content creation.
13	Build a Digital Twin demo — simulate sensor data via MQTT and visualize it on a 3D model in the browser.
14	Secure the Digital Twin pipeline — enable TLS and token-based authentication on the MQTT/OPC-UA channel, then inject and detect a false-data-injection (sensor-spoofing) attack on the twin.

Batchwise laboratory work of minimum **10** experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Batchwise tutorial sessions are to be conducted on topics which would help the learner to identify/analyze the problem and to apply problem solving techniques learnt.

Books Recommended:

Text Books:

1. Anjan Chatterjee, "Building the Metaverse: A Comprehensive Guide for Architects, Engineers, and Designers", CRC Press / Routledge, 2024
2. Ali Darejeh, Samad Sepasgozar, "Metaverse, Generative AI, and Brain–Computer InterfacesShaping the Future of the Digital Landscape", Routledge, 2026
3. Cathy Hackl, Dirk Lueth, Tommaso Di Bartolo, "Navigating the Metaverse: A Guide to Limitless Possibilities in a Web 3.0 World", Wiley, 2022
4. QuHarrison Terry, Scott "DJ Skee" Keeney, "The Metaverse Handbook: Innovating for the Internet's Next Tomorrow", Wiley, 2022
5. Kumar Saurabh, Ashutosh Saxena, "Blockchain Technology: Concepts and Applications", Wiley India, 2022

Reference Books:

1. Anand Nayyar, Akshi Kumar (Eds.), "Augmented and Virtual Reality", TMH / McGraw Hill, 2023
2. Pethuru Raj, Kavita Saini, Chellammal Surianarayanan, "Blockchain Technology and Applications", TMH / McGraw Hill, 2022
3. Y. Wang, Z. Su et al., "Metaverse Security and Privacy", Springer, 2024
4. Steve Aukstakalnis, "Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR", Pearson / Addison-Wesley, 2017

Web Resources:

1. Foundation for Virtual and Augmented Reality Systems, IIT Guwahati
   https://nptel.ac.in/courses/106103842
2. Mobile Virtual Reality and Artificial Intelligence
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc25_cs80

Program: Computer Engineering **Final Year B.Tech.** **Semester: VII**

Course: Social Network Analysis (DJS23CCPE721)

Course: Social Network Analysis Laboratory (DJS23CLPE721)

Pre-requisite:

1. Discrete Graph Theory
2. Computer Networks

Objectives:

1. To introduce the fundamentals of Social Network Analysis (SNA), graph representations, and network measures.
2. To study network growth models, link analysis, and community structure in real-world networks.
3. To apply link prediction, cascade behaviours, and anomaly detection techniques on social networks.
4. To explore graph representation learning and real-world applications of SNA.

Outcomes: On completion of the course, the learner will be able to:

1. Explain the foundations, levels, and measures of Social Network Analysis.
2. Apply network growth models and link-analysis algorithms (PageRank, SimRank, etc.) to real-world networks.
3. Analyze community structure, link prediction, and cascade behaviours in static and dynamic networks.
4. Design SNA-based solutions using graph representation learning for applications such as recommender systems, anomaly detection, and OSN security.

Social Network Analysis (DJS23CCPE721)		
Unit	**Description**	**Duration**
1	**Foundations of Social Network Analysis and Network Measures** **Networks and Society:** What is Social Network Analysis; why we study social networks; applications of SNA; preliminaries; three levels of SNA (micro, meso, macro); historical development; graph visualisation tools (Gephi, Cytoscape, NetworkX, igraph). **Network Measures:** Network basics — nodes, edges, directed/undirected, weighted graphs, adjacency representations; node centrality (degree, closeness, betweenness, eigenvector, Katz, PageRank-based); assortativity; transitivity and reciprocity; similarity measures; degeneracy and k-core decomposition. **Self-learning Topics:** Higher-order and hypergraph network analysis (beyond pairwise edges), Multilayer / multiplex networks, Temporal network measures and time-respecting paths	7
2	**Network Growth Models** Properties of real-world networks — small-world, scale-free, heavy-tailed degree distributions, clustering. Random network model (Erdős–Rényi). Ring lattice network model. Watts–Strogatz small-world model. Preferential attachment model (Barabási–Albert). Price's model for citation networks. Local-world	7

	network growth model. Network model with accelerating growth. Aging in preferential attachment. **Self-learning Topics:** Deep generative graph models (GraphRNN, graph diffusion models), Densification power laws and shrinking diameters in evolving networks	
3	**Link Analysis** Applications of link analysis. Signed networks; structural balance theory. Strong and weak ties (Granovetter). Link analysis algorithms — overview. PageRank — formulation, random surfer model, iterative computation, dangling nodes, teleportation. Personalised PageRank. DivRank — diversity-aware ranking. SimRank — structural similarity between nodes. PathSIM — meta-path-based similarity in heterogeneous networks. Comparative analysis of ranking algorithms. **Self-learning Topics:** Learning-to-rank on graphs, Trust and reputation propagation in networks, Axiomatic / fairness analysis of ranking algorithms	7
4	**Community Structure and Link Prediction** **Community Structure in Networks:** Applications of community detection; types of communities (disjoint, overlapping, local); community detection methods — modularity-based (Girvan–Newman, Louvain), spectral, label propagation; disjoint vs overlapping vs local community detection; community detection vs community search; evaluation metrics (modularity, NMI, F1, ARI). **Link Prediction:** Applications; temporal changes in a network; problem definition and evaluation (AUC, precision-at-k); heuristic models — common neighbours, Jaccard, Adamic–Adar, Katz, preferential attachment; probabilistic models; supervised random walk; information-theoretic model; latest trends (GNN-based link prediction). **Self-learning Topics:** Tracking dynamic community evolution over time, Community deception / hiding from detection algorithms, GNN-based overlapping community detection	8
5	**Cascade Behaviours, Network Effects and Anomaly Detection** **Cascade Behaviours:** Preliminaries and key terminology — diffusion, threshold, contagion. Cascade models — Linear Threshold and Independent Cascade. Probabilistic cascades. Epidemic models — SI, SIS, SIR, SEIR. Cascade prediction. Case study: the *Indignados* movement. **Anomaly Detection in Networks:** Outliers vs network-based anomalies; challenges; anomaly detection in static networks (structural, attribute-based); anomaly detection in dynamic networks (event-, change-, community-based). **Self-learning Topics:** Misinformation and rumour-propagation modelling, Coordinated inauthentic behaviour and large-scale bot-campaign detection, Competitive and time-critical influence maximization, Deep anomaly detection on dynamic graphs	6
6	**Graph Representation Learning and Applications** Graph Representation Learning: Machine learning pipelines for graph data; intuition and benefits of representation learning; criteria for good graph embeddings; graph representation learning pipeline; methods — DeepWalk, Node2Vec, LINE, GraphSAGE, GCN, GAT (introductory). Applications and	7

	Case Studies: Malicious activities on OSNs; sockpuppets in OSNs; collusion on online social networks; modelling the spread of COVID-19 using SNA; recommender systems based on SNA. **Self-learning Topics:** Temporal / dynamic graph neural networks (TGN), Graph transformers, Self-supervised and contrastive learning on graphs, Explainable GNNs (e.g., GNNExplainer), GraphRAG and LLM-augmented graph reasoning, Adversarial attacks and robustness of GNNs	
	Total	**42**

Social Network Analysis Laboratory (DJS23CLPE721)	
Exp.	**Suggested experiments**
1	**Network construction and visualization:** load a real-world network (e.g., Zachary's Karate Club / Facebook ego-net), compute basic statistics (nodes, edges, density, degree distribution), and visualize it.
2	**Node centrality analysis:** compute and compare degree, closeness, betweenness, eigenvector, and PageRank centralities; identify top-k influential nodes.
3	**Network properties:** compute clustering coefficient, transitivity, reciprocity, assortativity, and k-core decomposition for the chosen network.
4	**Network growth models:** generate Erdős–Rényi, Watts–Strogatz, and Barabási–Albert networks; compare degree distributions, clustering, and average path length with a real network.
5	**Link analysis algorithms:** implement / apply PageRank, Personalised PageRank, and SimRank on a chosen graph; rank top-k nodes and compare results.
6	**Community detection:** apply Girvan–Newman, Louvain, and Label Propagation; visualize and evaluate communities using modularity and NMI.
7	**Link prediction:** evaluate Common Neighbours, Jaccard, Adamic–Adar, and Preferential Attachment heuristics on a temporal split of the dataset; report AUC and precision-at-k.
8	**Epidemic spread modelling:** simulate SI / SIR / SEIR models on a real network and analyze how network structure affects spread (mini-version of the COVID-19 case study).
9	**Anomaly detection:** detect anomalous nodes/edges in a static network using structural features (degree, ego-net density, OddBall-style); visualize outliers.
10	**Graph representation learning:** generate node embeddings using Node2Vec / DeepWalk; perform node classification or link prediction; visualize embeddings with t-SNE.
11	**Influence maximization and cascade modelling:** implement the Linear Threshold and Independent Cascade models on a real network, then apply greedy seed selection to find the top-k most influential nodes and compare the resulting spread.
12	**Signed-network and structural-balance analysis:** load a signed network (e.g., Epinions / Slashdot), classify triads as balanced or unbalanced, measure the degree of structural balance, and relate findings to strong/weak ties.
13	**SNA-based recommender system:** build a friend- or item-recommender using network structure (common-neighbour / Adamic–Adar scoring or Node2Vec embeddings), then evaluate top-k recommendations against a held-out set.

14	**GNN-based link prediction / OSN security:** train a GCN or GraphSAGE model for link prediction (or sockpuppet/malicious-node classification) and compare its AUC against the heuristic baselines from Experiment 7.

Batchwise laboratory work of minimum **10** experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Batchwise tutorial sessions are to be conducted on topics which would help the learner to identify/analyze the problem and to apply problem solving techniques learnt.

Books Recommended:

Text Books:

1. Tanmoy Chakraborty, "Social Network Analysis", Wiley, 2024.
2. Maksim Tsvetovat and Alexander Kouznetsov, "Social Network Analysis for Startups", O'Reilly, 2011.
3. Krishna Raj P.M., Ankith Mohan, K.G. Srinivasa, "Practical Social Network Analysis with Python, Springer, 2018.
4. Charu C. Aggrwal, "Social Network Data Analytics", Springer, 2011.

Reference Books:

1. Niyati Aggrawal and Adarsh Anand, "Social Networks Modelling and Analysis", CRC Press, 2022
2. Stanley Wasserman, Katherine Faust, "Social Network Analysis: Methods and Applications", Cambridge University Press, 1994.
3. Stephen P Borgatti, Martin G Everett and Jeffrey C Johnson, "Analyzing Social Networks", SAGE, 2013.
4. Mehmet Kaya, Reda Alhajj, "Influence and Behavior Analysis in Social Networks and Social Media", Springer, 2019

Web Resources:

1. Social Network Analysis
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_cs117
2. Social Networks, IIT Ropar
   https://nptel.ac.in/courses/106106169

Program: Computer Engineering **Final Year B.Tech.** **Semester: VII**

Course: Virtual Infrastructure Systems (DJS23CCPE722)

Course: Virtual Infrastructure Systems Laboratory (DJS23CLPE722)

Pre-requisite:

1. Operating Systems
2. Computer Networks
3. Database Management Systems
4. Basic Cloud Computing concepts

Objectives:

1. Understand the architecture, components, and operation of virtualized IT infrastructure.
2. Learn virtualization methods for compute, storage, and networking.
3. Manage virtual machines and data center operations using industry tools.
4. Explore performance, security, and migration in virtual environments.
5. Understand how virtualization underpins modern cloud computing systems.

Outcomes: On completion of the course, the learner will be able to:

1. Explain the concepts and architecture of virtualization and cloud computing.
2. Configure virtual machines, networks, and storage resources using virtualization tools.
3. Analyze performance, scalability, and security aspects of virtualized systems.
4. Implement integrated virtualization and cloud infrastructure solutions.

Virtual Infrastructure Systems (DJS23CCPE722)		
Unit	**Description**	**Duration**
1	**Virtualized IT Infrastructure – Concepts and Cloud Foundations** Physical vs. Virtual IT Infrastructure, Virtualization in Data Centers – Hardware, Storage, Memory, and I/O virtualization, Types of Virtualization – Server, Desktop, Application, Network, Need for Virtualization; Benefits and Challenges, Cloud Computing Fundamentals – Definition, Characteristics, Service and Deployment Models (IaaS, PaaS, SaaS; Public, Private, Hybrid, Community Clouds), Relationship between Virtualization and Cloud Computing, Recent Trends & Emerging Technologies in Virtualized Environments **Self-learning Topics:** Multi-cloud and cloud-agnostic infrastructure strategies, Sovereign cloud and data-residency-driven deployment, Confidential computing and Trusted Execution Environments (TEEs)	6
2	**Virtual Machine Management** Hypervisors – Types and Architecture, Role of VMM (Virtual Machine Monitor), VM Lifecycle – Configuration, Provisioning, Scaling, and Deletion, VM Scheduling and Load Balancing, Migration – Live, Cold, and Storage Migration; Migration Process and Tools, Software and Workload Migration (P2V, V2V), QoS Parameters – Performance, Functionality, and Reliability, VM Hosting and Comparison (Windows vs. Linux) **Self-learning Topics:** Live-migration optimization: pre-copy vs. post-copy and RDMA-accelerated migration, Virtual Machine Introspection (VMI) for security, Unikernels as lightweight VM alternatives	7

3	**Hardware and Storage Virtualization** Abstracting Hardware – Concept, Need, and Solutions, CPU Virtualization Approaches & Algorithms, Memory Virtualization – Techniques, Ballooning, Shadow Paging, Storage Virtualization – Methods, Storage Aggregation, and Software-Defined Storage, I/O Virtualization Techniques (SR-IOV, PCI Passthrough), Role of Virtualization in Data Center Efficiency **Self-learning Topics:** GPU virtualization and vGPU for AI/ML workloads, Persistent-memory (PMEM/NVDIMM) virtualization, Composable / disaggregated infrastructure and CXL	7
4	**Network Virtualization** Virtual Interfaces, Virtual Switches and Routers, VNIC Profiles, VLANs, and Virtual Network Design, Network Modes – Bridged, NAT, Host-Only, Data Flow among VMs; TUN/TAP Drivers; Host-Only vs. Bridged Networking, Data Center Network Virtualization using ESXi, Software-Defined Networking (SDN) Overview, Optimizing Resource Utilization & Network Security in Virtual Environments **Self-learning Topics:** Network Function Virtualization (NFV) and service-function chaining, eBPF / XDP for high-performance virtual networking, Service mesh (Istio / Linkerd) and overlay encapsulation (VXLAN/GENEVE)	7
5	**Server Virtualization and Cloud Integration** Server Partitioning and Virtual Server Hosts, Deploying and Managing Virtual Servers (vSphere, Hyper-V, KVM, Xen), Remote Management Tools and Automation (vCenter, CLI, Web Console), Server Clustering and Network Load Balancing (NLB), Performance and Health Monitoring – vSphere Monitoring & Tools, Integration of Virtualization with Cloud Services (IaaS Implementation), Introduction to Containers and Orchestration (Docker, Kubernetes) **Self-learning Topics:** Infrastructure-as-Code (Terraform, Ansible, Pulumi), GitOps for declarative infrastructure management, Serverless containers (Fargate, Knative) and WebAssembly (Wasm) server runtimes, Kubernetes operators and custom resource definitions	7
6	**Virtualization Performance, Security, and Emerging Trends** Performance Issues – Resource Overcommitment, VM Sprawl, and Bottlenecks, Virtualization Security Risks – Hypervisor Vulnerabilities, VM and Migration Attacks, Backup, Fault Tolerance, and Disaster Recovery in Virtualized Systems, Security Hardening and Monitoring Strategies, Security Standards and Compliance for Virtualization – CIS Benchmarks, NIST SP 800-125/125A, ISO/IEC 27017, PCI-DSS virtualization guidance; Audit Mechanisms – hypervisor/vCenter audit logging, configuration-compliance scanning, baseline validation, SIEM integration and continuous compliance monitoring; Backup, Fault Tolerance, and Disaster Recovery in Virtualized SystemsData Center Automation and Orchestration Tools (OpenStack, vCloud Suite), Emerging Trends – Edge Computing, Serverless Architecture, Green IT **Self-learning Topics:** Confidential VMs (AMD SEV, Intel TDX) and full-memory encryption, Chaos engineering for resilience testing, FinOps — cost optimization for virtualized and cloud infrastructure, Zero-Trust architecture for virtualized workloads, Carbon-aware computing and sustainable data-centre design	8
	Total	**42**

Virtual Infrastructure Systems Laboratory (DJS23CLPE722)	
Exp.	**Suggested experiments**
1	**Installation of a Hypervisor** – Install and set up Type-1 and Type-2 hypervisors (e.g., VMware ESXi, VirtualBox).
2	**Create and Manage Virtual Machines** – Configure CPU, memory, and storage parameters; start, stop, and snapshot VMs.
3	**Virtual Networking Setup** – Design NAT, Bridged, and Host-Only networks between multiple VMs.
4	**Storage Virtualization** – Configure shared storage using iSCSI or NFS and attach to virtual hosts.
5	**Virtual Machine Migration** – Perform live and cold migrations of VMs between hosts.
6	**Resource Allocation and Scheduling** – Experiment with CPU and memory resource limits, shares, and reservations.
7	**Server Virtualization and Monitoring** – Deploy and manage virtual servers using VMware vCenter or Microsoft Hyper-V Manager.
8	**Network Load Balancing** – Configure and test load balancing between multiple virtual servers.
9	**Security in Virtual Environments** – Demonstrate access controls, snapshots for recovery, and identify hypervisor vulnerabilities.
10	**Cloud Integration** – Deploy a simple IaaS environment using OpenStack or AWS EC2, linking virtualization to cloud concepts.
11	**Containerization and Orchestration** – Build and run application containers using Docker (create images, manage volumes and networks), then deploy and scale a multi-container application on a Kubernetes cluster (minikube/kind); demonstrate pods, services, deployments, and auto-scaling, and compare the resource footprint of containers vs. virtual machines.
12	**Fault Tolerance and Disaster Recovery** – Configure High Availability (HA) and Fault Tolerance for virtual machines, schedule automated backups/snapshots, and simulate a host failure to demonstrate VM failover, recovery, and rollback using tools such as VMware HA/FT or KVM with Pacemaker/Corosync.

Batchwise laboratory work of minimum **10** experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Batchwise tutorial sessions are to be conducted on topics which would help the learner to identify/analyze the problem and to apply problem solving techniques learnt.

Books Recommended:

Text Books:

1. Matthew Portnoy – "Virtualization Essentials", Sybex / Wiley, 3rd Edition (2023).
2. Barrie Sosinsky, “Cloud Computing Bible”, Wiley, 2011
3. Jim Smith, Ravi Nair, “Virtual Machines - Versatile Platforms for Systems and Processes”, Morgan Kaufmann, 2005
4. John A. Davis, Steve Baca, Owen Thomas, “VCP-DCV for vSphere 8.x Cert Guide, O'Reilly, 5th Edition”, 2023

Reference Books:

1. Elver Sena Sosa, "Mastering VMware NSX® for vSphere®", John Wiley & Sons, Inc, 2020
2. Edward Haletky, "VMware vSphere and Virtual Infrastructure Security: Securing the Virtual Environment", Prentice Hall, 2009
3. Thomas Erl, Cloud Computing: Concepts, Technology & Architecture, Pearson, 2013.

Web Resources:

1. Cloud Infrastructure and Services
   https://onlinecourses.swayam2.ac.in/e-learning/preview/nou25_cs33
2. Cloud Computing and Distributed Systems, IIT Patna
   https://nptel.ac.in/courses/106104182

Program: Computer Engineering **Final Year B.Tech.** **Semester: VII**

Course: Internet of Everything (DJS23CCPE723)

Course: Internet of Everything Laboratory (DJS23CLPE723)

Pre-requisite:

1. Internet of Things
2. Computer Networks

Objectives:

1. To understand the fundamentals of IoT and its evolution to IoE, including architectures, components, industrial application domains, and challenges.
2. To analyse IoT/IoE devices, sensing technologies, communication networks, connectivity solutions, and system architectures.
3. To apply communication protocols, data management techniques, and analytics frameworks for intelligent IoE systems.
4. To evaluate security, governance, emerging technologies, and real-world IoE applications for designing future-ready connected systems.

Outcomes: On successful completion of the course, the learner will be able to:

1. Explain IoT fundamentals, IoT reference architectures, IoE concepts, components, benefits, and application areas.
2. Examine IoT devices, sensors, connectivity technologies, architectures, and deployment platforms used in IoE systems.
3. Assess IoT communication protocols, routing mechanisms, data lifecycle management, and analytics approaches for IoE applications.
4. Assess security, privacy, governance requirements, and emerging trends such as AIoT, Industry 4.0, Digital Twins, and Blockchain-enabled IoE solutions.

Internet of Everything (DJS23CCPE723)		
Unit	**Description**	**Duration**
1	**IoT Foundations and Evolution to IoE** • IoT Stack & Core Concepts: IoT Stack, Overview of IoT, Characteristics of IoT, IoT Reference Architecture (e.g. ITU-T Y.2060). • Evolution from IoT to IoE, IoE Components. IoE Four-Pillar Framework • IoE Application Areas & Challenges: Application areas of IoE, IoE Challenges - scalability, interoperability, power, bandwidth, security & privacy. • M2M, IOE & Disambiguation: M2M Technology, Understanding IoE, IoE Pyramid model, IoT 2.0 and IoE Benefits, IoT vs IoE vs M2M. **Self-learning Topics:** Web of Things (WoT) and W3C standards, Social Internet of Things (SIoT), Internet of Behaviours (IoB)	6
2	**Devices, Sensors and Connectivity** • Common IoT/IoE Devices: Introduction to Sensors, Actuators, Transducers and gateways. • Sensing Technologies: RFID in IoT, NFC (Near Field Communication) Connectivity Technologies: Short-range (Wi-Fi (802.11), Bluetooth / BLE, Zigbee (802.15.4), NFC), Low-Power Wide-Area	8

	Networks: LoRa / LoRaWAN, Sigfox, NB-IoT, LTE-M), Wired and Wireless device connectivity (5G/6G). **Self-learning Topics:** Backscatter communication and ambient (battery-free) IoT, Satellite IoT and non-terrestrial networks (NTN), Ultra-wideband (UWB) for precise indoor localization	
3	**IoE Architecture and Platforms** • Overview of Physical Design of IoT, Logical Design of IoT — IoT Communication Model, IoT Levels and Deployment Templates • Cisco IoE Reference Architecture • System Architecture: 3-tier/multi-tier models • Edge Computing and Fog Computing — architecture, comparison, Use Cases • IoT Devices & Software: IoT Devices, Software, Device Management Platform such as Google Cloud IoT, Cisco IOx etc. **Self-learning Topics:** Serverless / FaaS at the edge, KubeEdge and Kubernetes-based edge orchestration, Standardized digital-twin platforms and the Asset Administration Shell (AAS)	6
4	**Communication & Data Analytics** • Protocol Layer Overview: IoT Protocol Stack— mapping protocols to OSI layers; why IoT needs specialized protocols. • Application Layer Protocols: MQTT, CoAP, AMQP, DDS, REST • Network & Adaptation Layer Protocols: IPv6 for IoT, 6LoWPAN, IoT Security Protocols, OSCORE • IoT Routing Protocols: Data-centric and Flat-Architecture Routing, Flooding and Gossiping, SPIN Protocol family, LEACH Protocol. • Data Lifecycle – Collection, aggregation, transmission, and visualization. • Edge and Fog Analytics **Self-learning Topics:** Time-Sensitive Networking (TSN) for deterministic IoT, Matter and Thread for smart-home interoperability, Stream processing for IoT (Apache Kafka, Apache Flink), Information-Centric / Named Data Networking (NDN) for IoT	8
5	**Security and Governance in IoE** • Security Requirements beyond IoT – Trust, authentication, and secure orchestration across IoE layers. • Data Privacy and Governance – Regulations (GDPR), policy frameworks, and ethical considerations. • Security for Edge/Fog computing environments. • Blockchain for IoE Security – Decentralized trust and traceability in data sharing. **Self-learning Topics:** Zero-Trust architecture for IoT/IoE, Federated learning for privacy-preserving IoE analytics, Physically Unclonable Functions (PUFs) and hardware root-of-trust, Post-quantum cryptography for constrained devices	8
6	**Applications and Future Trends in IoE** • Domain Case Studies: Smart Home Automation, Smart Cities, Environment Monitoring, Transportation & Supply Chain / Logistics, Agriculture, Industrial IoT, Healthcare, Smart Energy / Smart Grid. • Emerging Trends: AIoT (AI + IoT) — edge inference, TinyML, on-device anomaly detection, intelligent actuators, IIoT 4.0 / Industry 4.0 — cyber-physical systems, Digital Twins Concept.	6

	Self-learning Topics: Agentic AI / LLM-driven IoT orchestration, Sustainable, green, and energy-neutral IoT, Tactile internet and haptic IoE Convergence of IoE with the Industrial Metaverse	
	Total	**42**

Internet of Everything Laboratory (DJS23CLPE723)	
Exp.	**Suggested experiments**
1	**Traffic Light Simulation using Arduino** Design and implement a basic traffic light control system using Red, Yellow, and Green LEDs connected to an Arduino. Program the LEDs to follow standard traffic signal timing sequences. **Tools:** Arduino Uno, LEDs, Resistors, Breadboard, Arduino IDE.
2	**IoE Architecture Setup** Configure a small-scale IoE environment integrating sensors, gateways, and cloud dashboards. Tools: Raspberry Pi, NodeMCU, ThingSpeak/AWS IoT.
3	**Connectivity and Data Transmission** Interfacing a Smoke Sensor with Arduino and Implementing MQTT/CoAP-Based Data Transmission with Latency and Throughput Analysis **Tools:** Arduino Uno, LEDs, Resistors, Breadboard, Arduino IDE, MQTT.
4	**Edge-Fog-Cloud Integration** Deploy a data processing pipeline that filters data at the edge and stores aggregate data in the cloud. Tools: Node-RED + AWS IoT or Azure IoT.
5	**IoE Data Analytics** Collect real-time IoE data and perform basic analytics (averages, anomalies, trends). Tools: Python (Pandas), Grafana, or Power BI for visualization.
6	**Machine Learning for IoE** Train a simple ML model (e.g., temperature anomaly detection) and integrate it with an IoE node for real-time decision making. Tools: TensorFlow Lite, Edge Impulse, or Python scikit-learn.
7	**Interoperability Experiment** Integrate heterogeneous devices (e.g., Zigbee, Wi-Fi, Bluetooth) in a single IoE network and test interaction through a middleware (e.g., Node-RED or MQTT broker).
8	**Implementation of Digital Twin** Create a digital twin of a physical device that replicates live sensor data on a virtual dashboard.
9	**IoE Security Demonstration** Implement end-to-end security using TLS/SSL for MQTT communication and observe changes in performance. Optional: Simulate token-based authentication (OAuth or JWT).
10	**IoE Smart Application Prototype** Develop a mini-project such as a Smart Parking, Smart Home Energy Monitor, or Smart Campus Dashboard integrating multiple IoE layers — sensing, transmission, analytics, and visualization.

11	**Long-Range Communication using LoRa/LoRaWAN** Establish a point-to-point or star-topology LoRa link between two end nodes and a gateway. Transmit sensor data (e.g., temperature/humidity) over a long distance and measure RSSI, SNR, packet delivery ratio, and effect of spreading factor on range vs. data rate.
12	**RFID/NFC-Based Asset Tracking and Access Control** Build a system that reads RFID/NFC tags to identify and log assets or grant access, pushing scan events with timestamps to a cloud dashboard. Demonstrate tag enrolment, authentication, and event logging**.**
13	**Blockchain-Enabled Data Integrity for IoE** Record IoE sensor readings on a permissioned/test blockchain to ensure tamper-evident, traceable data sharing. Compare a normal write vs. a tampered record and verify integrity through hashing.
14	**TinyML On-Device Anomaly Detection (AIoT)** Train and deploy a lightweight ML model directly on a microcontroller for on-device inference (e.g., vibration/motion or sound anomaly detection) without cloud dependence, and benchmark inference latency and memory footprint against a cloud-based approach.

Batchwise laboratory work of minimum **10** experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Batchwise tutorial sessions are to be conducted on topics which would help the learner to identify/analyze the problem and to apply problem solving techniques learnt.

Books Recommended:

Text Books:

1. Dr. Prateek Jain and Dr. Archana Sharma, "Transitioning to Internet of Everything (IoE) Key Technology Application and Recent Trends", BFC Publications, 2024.
2. T. Kavitha, V. Ajantha Devi, Internet of Everything: Smart Sensing Technologies, Nova Publishers, 2022
3. Arshdeep Bahga and Vijay Madisetti, "Internet of Things: A Hands-On Approach", 1st Edition, Universities Press, 2016.

Reference Books:

1. Hang Song, "Internet of Everything: Key Technologies, Practical Applications and Security of IoT", World Scientific, Tsinghua University Press, 2023.
2. Pethuru Raj & Anupama Raman, The Internet of Everything (IoE): IoT to Next-Gen IoE, CRC Press, 2019.
3. David Hanes et al., IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, Cisco Press, 2017.

Web Resources:

1. Introduction to Internet of Things by Prof. Rajbabu Velmurugan
   https://www.coursera.org/learn/introduction-to-internet-of-things
2. Introduction to Internet of Things, IIT Kharagpur
   https://onlinecourses.nptel.ac.in/noc22_cs53/preview

Program: Computer Engineering **Final Year B.Tech.** **Semester: VII**

Course: Vulnerability Assessment and Penetration Testing (DJS23CMD701)

Pre-requisite:

Basic understanding of Computer Networks, Fundamentals of Operating Systems, Basic Programming Knowledge, Web Technologies.

Objectives:

1. Introduce students to the fundamental principles of ethical hacking, penetration testing frameworks, and reconnaissance methodologies used in cybersecurity.
2. Develop skills in identifying and assessing security vulnerabilities in networks, systems, and applications using industry-standard scanning and analysis tools.
3. Provide hands-on experience in exploiting common vulnerabilities and implementing post-exploitation techniques in controlled environments.
4. Creating professional penetration testing documentation, communicating security findings, and recommending effective remediation strategies.

Outcomes: On completion of the course, the learner will be able to:

1. Understand and apply penetration testing methodologies, information gathering, reconnaissance, scanning, and enumeration techniques.
2. Analyze network and system vulnerabilities using automated and manual assessment techniques and prioritize security risks.
3. Exploit web application, network, and system vulnerabilities using industry-standard tools including OWASP Top 10 and Metasploit.
4. Create comprehensive penetration testing reports documenting vulnerabilities, risk assessment, and remediation recommendations.

Vulnerability Assessment and Penetration Testing (DJS23CMD701)		
Unit	**Description**	**Duration**
1	**Introduction to Penetration Testing:** Introduction to Information Security and Ethical Hacking, Difference between Vulnerability Assessment and Penetration Testing, Types of Penetration Testing: Black Box, White Box, and Grey Box, Penetration Testing Methodologies: OWASP, PTES, NIST, Legal and Ethical Considerations, Setting up Penetration Testing Lab Environment, Introduction to Kali Linux and Essential Tools. **Self-learning Topics:** Red team vs. blue team vs. purple team operations, Bug bounty programs and coordinated/responsible vulnerability disclosure, Threat-led penetration testing frameworks (TIBER-EU, CBEST)	4
2	**Information Gathering and Reconnaissance:** Passive Reconnaissance Techniques, OSINT (Open Source Intelligence) Gathering, DNS Enumeration and Zone Transfers, WHOIS Lookups and Domain Information, Active Reconnaissance and Footprinting, Network Scanning with Nmap, Port	5

	Scanning Techniques and Service Detection, Banner Grabbing and OS Fingerprinting. **Tools:** Maltego, theHarvester, Shodan, Censys, Recon-ng. **Self-learning Topics:** Attack Surface Management (ASM) platforms, Advanced OSINT automation and adversary-infrastructure tracking, SOCMINT and the ethics of open-source intelligence	
3	**Vulnerability Assessment and Analysis: Understanding** Vulnerabilities and Common Vulnerability Scoring System (CVSS), Vulnerability Databases: CVE, NVD, Exploit-DB, Automated Vulnerability Scanning, Network Vulnerability Scanning with Nessus and OpenVAS, Web Application Vulnerability Scanning, Manual Vulnerability Assessment Techniques, Analyzing Scan Results and False Positive Detection, Vulnerability Prioritization and Risk Assessment. **Tools:** Nessus, OpenVAS, Nikto, OWASP ZAP, Burp Suite **Self-learning Topics:** SBOM (Software Bill of Materials) and software supply-chain security, Exploit Prediction Scoring System (EPSS) and the CISA KEV catalog, Breach-and-Attack Simulation (BAS) for continuous security validation	5
4	**Web Application Penetration Testing:** OWASP Top 10 Vulnerabilities (Latest Version), SQL Injection: Detection and Exploitation, Cross-Site Scripting (XSS): Reflected, Stored, and DOM-based, Cross-Site Request Forgery (CSRF), Authentication and Session Management Flaws, Insecure Direct Object References (IDOR), Security Misconfigurations and Sensitive Data Exposure, XML External Entity (XXE) Attacks, Server-Side Request Forgery (SSRF), File Upload Vulnerabilities and Remote Code Execution, API Security Testing **Tools:** Burp Suite Professional, OWASP ZAP, SQLMap, XSSer **Self-learning Topics:** Cloud penetration testing (AWS / Azure / GCP misconfigurations), GraphQL and the OWASP API Security Top 10, DevSecOps — shifting security left with SAST / DAST / IAST in CI/CD	6
5	**System and Network Exploitation:** Exploitation Frameworks: Metasploit, Empire, Cobalt Strike, Buffer Overflow Attacks and Stack-based Exploitation, Password Attacks: Brute Force, Dictionary, Rainbow Tables, Credential Harvesting and Hash Cracking, Privilege Escalation: Windows and Linux, Post-Exploitation Techniques, Lateral Movement and Pivoting, Persistence Mechanisms, Wireless Network Penetration Testing. **Tools:** Metasploit, John the Ripper, Hashcat, Hydra, Mimikatz, PowerSploit **Self-learning Topics:** Active Directory attack paths and tiered-admin defence (conceptual), Container and Kubernetes security assessment, Mobile application penetration testing (Android / iOS), IoT / OT / ICS security assessment	4

6	**Reporting and Remediation:** Penetration Testing Report Structure and Components, Executive Summary and Technical Findings, Risk Rating and Vulnerability Classification, Evidence Collection and Documentation, Remediation Recommendations and Best Practices, Compliance and Industry Standards: PCI-DSS, ISO 27001, NIST, Effective Communication with Stakeholders, Remediation Verification and Retesting, Continuous Security Improvement, **Career Paths:** Certifications (OSCP, CEH, GPEN, GWAPT, PNPT) **Self-learning Topics:** Detection engineering and purple-teaming collaboration, Security metrics and KPIs for vulnerability-management programs, Security testing of AI/LLM applications — the OWASP LLM Top 10 and prompt-injection testing, AI-assisted offensive security and AI-driven defense	4
	Total	**28**

Books Recommended:

Text Books:

1. CEH Certified Ethical Hacker All-in-One Exam Guide" by Matt Walker, 6th Edition, McGraw-Hill Education, 2024
2. Penetration Testing: A Hands-On Introduction to Hacking" by Georgia Weidman, No Starch Press, 2014
3. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning" by Gordon Fyodor Lyon, Insecure.Com LLC, 2009
4. Metasploit: The Penetration Tester's Guide" by David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni, No Starch Press, 2011
5. Penetration Testing: A Hands-On Introduction to Hacking" by Georgia Weidman, No Starch Press, 2014
6. Reporting Security Vulnerabilities: A Practical Guide for Penetration Testers" by Andrew Davies and Dan Raywood, IT Governance Publishing, 2015

Reference Books:

1. Kali Linux Revealed: Mastering the Penetration Testing Distribution" by Raphaël Hertzog, Jim O'Gorman, and Mati Aharoni, Offsec Press, 2024 (2nd Edition
2. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto, 2nd Edition, Wiley, 2011
3. The Hacker Playbook 3: Practical Guide To Penetration Testing" by Peter Kim, Secure Planet LLC, 2018
4. Reconnaissance and Information Gathering" by Ben Clark, Packt Publishing, 2023
5. Real-World Bug Hunting: A Field Guide to Web Hacking" by Peter Yaworski, No Starch Press, 2019
6. Effective Vulnerability Management: Managing Risk in the Vulnerable Digital Ecosystem" by Chris Hughes and Nikki Robinson, Apress, 2021.

Web Resources:

1. Introduction to Information Security - I, IIT Madras
   https://nptel.ac.in/courses/106106129

2. Cyber Security and Privacy
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc23_cs127

Program: Computer Engineering **Final Year B.Tech.** **Semester: VII/VIII**

Course: Research Ecosystem (DJS23ITELX12)

Pre-requisites:

1. Basic understanding of Engineering concepts and problem-solving techniques.
2. Fundamental knowledge of project-based learning, data analysis, and domain-specific tools.
3. Basic exposure to project development, technical report writing, and emerging technologies.

Objectives:

1. To understand and identify complex engineering, societal, and industrial problems suitable for research through structured analysis of literature, patents, and real-world needs.
2. To learn and apply advanced research methodologies—including experimental, simulation-based, and data-driven approaches—for designing and validating engineering solutions.
3. To gain proficiency in selecting and using appropriate tools, software, and platforms (AI/ML, simulation, CAD, IoT, etc.) for experiment design, data acquisition, and performance evaluation.
4. To transform research insights into innovative prototypes, products, or startup concepts by engaging with innovation ecosystems, TRLs, and funding mechanisms.
5. To analyze socio-technical systems, ethics, and sustainability implications of engineering solutions, ensuring responsible and impactful outcomes.
6. To integrate systems thinking, multidisciplinary perspectives, policy awareness, future technology trends, and technical documentation skills to develop comprehensive research proposals and outputs (papers, patents, reports).

Outcomes: On completion of the course, the learner will be able to:

1. Analyze complex engineering and societal problems to evaluate literature and patents and formulate clear research gaps.
2. Design and apply advanced research methods and tools to acquire data and evaluate solution performance.
3. Transform research insights into prototypes or startup concepts using innovation frameworks and Technology Readiness Levels.
4. Evaluate ethical, socio-technical, and sustainability issues and create responsible engineering solutions.
5. Synthesize multidisciplinary knowledge to create research proposals and scalable, future-ready solutions.

Research Ecosystem (DJS23ITELX12)		
Unit	**Description**	**Duration**

1	**Research Gap Identification & Advanced Problem Formulation**
	• Identifying high-impact problems from:
	  ▪ Industry and Society
	  ▪ Emerging technologies (AI, IoT, Blockchain, EV, etc.)
	• Research gap identification techniques:
	  ▪ Systematic Literature Review (SLR)
	  ▪ Patent analysis
	• Defining objectives, scope, constraints, and feasibility of the identified problem
	• Writing problem statement for:
		5

	▪ Research paper ▪ Product/Startup idea **Activities:** • Convert Industry/ Societal problem into research problem. • Analyze research articles and extract research gaps.	
2	**Experimental Design, Tools & Validation Techniques** • Advanced research methodologies: ▪ Experimental, simulation, data-driven, design thinking • Tool selection: ▪ Computational tools and software, simulation and analysis platforms, system prototyping and testing platforms • Data acquisition: ▪ Data collection interfaces, Principles of data cleaning and organization (primary / secondary data) • Validation techniques: ▪ Accuracy and performance metrics ▪ Benchmarking with existing solutions • Result interpretation & visualization **Activities:** • Selection of Research Design for final year project. • Apply evaluation metrics (e.g., ML model accuracy, system efficiency).	6
3	**Research to Innovation, Product & Startup Ecosystem** • Research → Prototype → Product → Startup lifecycle • Intellectual Property Rights (IPR), patents, copyrights • Technology readiness levels (TRL) • Case studies: ▪ AI startups, EV ecosystem, Smart Cities • Industry-academia collaboration models • Funding opportunities (grants, incubation, government schemes - Anusandhan National Research Foundation (ANRF)) **Activities:** • Convert project into startup idea or product pitch. • Analyze failure of a technology/product.	5
4	**Socio-Technical Systems, Ethics & Sustainability** • Socio-technical systems: Interaction of people, technology and environment • Ethical issues: ▪ Plagiarism, AI bias, data privacy • Sustainability in Engineering: ▪ Green practices, resource optimization and lifecycle considerations • Impact assessment: ▪ Social, economic, environmental • Policies and governance (data protection laws, standards and compliance requirements) **Activities:** • Case study: Facial recognition ethics / smart surveillance. • Impact analysis of student's own project.	6
5	**Integrated Research, Policy and Future Engineering Challenges** • Systems thinking and holistic analysis • Multidisciplinary problem solving: ▪ Healthcare tech, climate tech, smart mobility • Writing research proposals: ▪ Problem → Methodology → Expected outcomes • Future trends: ▪ AI ethics, Quantum computing, Industry 5.0 / 6.0	6

	• Scaling solutions from lab to society **Activities:** • *Capstone Task:* ▪ Identify real-world problem. ▪ Apply research methodology. ▪ Propose: o Technical solution o Societal impact o Possible commercialization	
	Total	**28**

Term Work Guidelines:

The Term Work (TW) shall consist of continuous assessment based on tutorials activities and a final group presentation.

a) **Part A (25 Marks):**
Each student shall complete a minimum of five tutorials based on the syllabus topics during the semester. The tutorials may include activities such as research gap identification, literature survey, case study analysis, tool-based experimentation, impact analysis, technical report writing, and innovation-oriented problem solving.

b) **Part B (25 Marks):**
Students shall work in groups and select a project/research topic based on emerging engineering, industrial, or societal challenges. The group presentation shall include:

i. Identification of a real-world problem
ii. Application of appropriate research methodology
iii. Proposed technical solution
iv. Analysis of societal impact
v. Possible commercialization/startup opportunities

Books Recommended:

Reference Books:

- Kothari, C. R., Research Methodology: Methods and Techniques, New Age International Publishers, New Delhi, 2023.
- Booth, W. C., Colomb, G. G., Williams, J. M., Bizup, J., & FitzGerald, W. T., The Craft of Research, University of Chicago Press, Chicago, 2024.
- Montgomery, D. C., Design and Analysis of Experiments, Wiley, New York, 2020.
- Singh, Pooja, Katiyar, Aman, Rawat, Shalini, Dhuri, Archana, Research Methodology and IPR, RK Publications, India, 2025.
- Blok, Vincent (Ed.), Putting Responsible Research and Innovation into Practice: A Multi-Stakeholder Approach, Springer Cham, Switzerland, 2022.

Web References:

- Research Methodology: Quantitative Methods and Computer Application (https://onlinecourses.swayam2.ac.in/e-learning/preview/ini26_ma02)
- Research Methodology and IPR (https://onlinecourses.swayam2.ac.in/e-learning/preview/ntr26_ed28)
- Research and Publication Ethics (https://onlinecourses.swayam2.ac.in/e-learning/preview/nou26_ge46)

- Research Ethics and Plagiarism (https://onlinecourses.swayam2.ac.in/e-learning/preview/nou26_ge26)
- Quantitative and Mixed Methods Research for Management (https://onlinecourses.swayam2.ac.in/e-learning/preview/imb26_mg21)
- Doing Qualitative Research (https://onlinecourses.swayam2.ac.in/e-learning/preview/nou26_ge61)

Program: Computer Engineering **Final Year B.Tech.** **Semester: VII/VIII**

Course: Major Project (DJS23IPELX13)

Objectives

The primary purpose of the Major Project is to enable students to apply the knowledge and skills acquired across the programme to conceive, design, implement, and evaluate a comprehensive solution to a real-world problem. The project strengthens technical competence, communication, and teamwork by integrating design, development, technical writing, presentation, and collaborative work. It emphasises professional engineering practice and addresses a range of non-technical considerations such as economic factors, safety, reliability, ethics, sustainability, and environmental and societal impact.

Course Outcomes (CO)

On successful completion of the Major Project, the student will be able to:

CO1: Apply technical knowledge gained from previous courses to identify real-life, industry, or societal problems and design appropriate solutions.

CO2: Demonstrate the technical skills required to design, develop, build, and test engineering systems, products, software, or hardware using modern engineering and IT tools.

CO3: Apply project-management skills — scheduling work, procuring resources/components, budgeting, documenting technical and non-technical details, and working within defined deadlines as an effective team.

CO4: Develop and demonstrate the ability to test, validate, debug, and troubleshoot engineering systems (including software, hardware, and process components) and analyse the results.

CO5: Create technical reports and research papers in the prescribed format and effectively communicate and present the work to evaluators.

Scope and Nature of the Project Work

Final-year students have already undertaken project assignments in their pre-final year through Innovative Product Design / Mini Project work. In the final year, a group of a maximum of four students completes a comprehensive project based on the courses studied. The project work may be internally assigned or externally assigned by research institutes, industry, or similar organisations. Each group is assigned one faculty member as supervisor. The final-year project may be an extension of the Mini Project carried out in the pre-final year.

The main intention of the project work is to enable students to apply the knowledge and skills learnt across the programme to solve or implement a well-defined practical problem. The work may go beyond the scope of the curriculum or build upon courses studied, but the thrust should be on:

- Learning additional skills relevant to the chosen domain;
- Developing the ability to define, design, analyse, and implement a problem and carry it through to completion with proper planning;
- Learning behavioural science and professional ethics by working effectively in a group;
- Selecting a project area aligned with the student's intended higher education, employment, or self-employment goals;

- Choosing a topic that differs from, or meaningfully advances, the Mini Project topic;
- Using the opportunity to learn modern computational techniques, tools, and model development through an appropriate choice of project.

Continuous Assessment and Term Work

The department maintains a proper record of the progress of each project throughout the semester. At the end of the semester, the work is assessed for the award of Term Work (TW) marks by an approved internal faculty member appointed by the Head of the Institute, on the basis of:

- Scope and objectives of the project work;
- Extent and quality of the literature survey;
- Progress of the work (continuous assessment across reviews);
- Design, implementation, and analysis of the project work;
- Results, conclusions, and future scope;
- Report prepared in the prescribed University / institutional format.

Oral / Practical Examination: An approved external examiner together with an internal examiner appointed by the Head of the Institute assesses the project during the oral examination. The oral examination consists of a presentation by the group along with a demonstration of the work done. Each student is assessed individually for their contribution, understanding, and the knowledge gained.

Project Guide / Topic Allocation Policy

The project guide and topic-allocation policy is explained to students during the Orientation held in Semester VI. Each faculty member is allocated a minimum of one and a maximum of four project groups. As per the timeline in the academic calendar, the list of project topics and areas offered by each faculty member — three topics per faculty member — is displayed on the notice board. Allocation is carried out over successive rounds, and in each round students submit their preferences against the available topics and guides.

If a topic is preferred by only one group, that group is allocated to the corresponding guide and topic. If a topic is preferred by more than one group, the concerned faculty member reviews the applications and selects one group; the remaining groups carry forward to the next round. After each round, the list of groups and guides allotted so far is displayed, along with the topics and guides that remain available. Unallotted groups then submit fresh preferences from the available list. This process continues over successive rounds until every group has been allotted a guide, ensuring a uniform distribution of project groups among the faculty members.

Assessment Scheme

The Major Project carries a total of 100 marks — **Term Work (50)** assessed internally through continuous evaluation, and **Oral / Practical Examination (50)** assessed jointly by internal and external examiners. Each criterion is scored against three performance levels — Excellent, Satisfactory, and Below Expectation — as detailed below.

Rubric for Term Work (50 Marks)

Assessment Criteria (mapped CO)	Max	Excellent	Satisfactory	Below Exp.
Problem Definition, Objectives & Literature Survey (CO1)	**10**	8–10	5–7	0–4
Design, Methodology & Implementation Progress – Continuous Assessment (CO2)	**10**	8–10	5–7	0–4
Project Management & Teamwork (CO3)	**10**	8–10	5–7	0–4
Report / Thesis Writing – Written Communication (CO5)	**10**	8–10	5–7	0–4
Research Paper Writing / IPR – Patent or Copyright (CO5)	**10**	8–10	5–7	0–4
TOTAL	**50**			

Rubric for Oral / Practical / Presentation / Demonstration (50 Marks)

Assessment Criteria (mapped CO)	Max	Excellent	Satisfactory	Below Exp.
Concept & Problem Definition (CO1)	**5**	4–5	3	0–2
Design, Methodology & Implementation (CO2)	**15**	12–15	7–11	0–6
Project Management Skills (CO3)	**5**	4–5	3	0–2
Demonstration, Testing & Result Analysis (CO4)	**15**	12–15	7–11	0–6
Presentation & Oral Communication (CO5)	**10**	8–10	5–7	0–4
TOTAL	**50**			

Descriptive Assessment Rubric (Performance-Level Descriptors)

The following descriptors define each performance level and are applied consistently across all programmes. Examiners map a student's performance to the most appropriate level for each criterion before awarding the corresponding marks from the tables above.

Criteria	Excellent	Satisfactory	Below Expectation
Concept & Problem Definition	All major and minor objectives are identified and correctly prioritised. The problem statement is precise, relevant and well-scoped. All relevant information is gathered from valid sources and the proposed solution is well justified.	Major objectives identified but one or two minor ones missing or priorities not fully set. Problem statement mostly clear. Sufficient information from largely valid sources; solution mostly justified.	Several major objectives not identified. Problem statement vague or too broad. Insufficient information and/or invalid sources; proposed solution not justified.
Literature Survey	Comprehensive, current and well-organised survey of journals, standards, patents and existing products/systems. Research gap / novelty clearly identified and the chosen approach is well justified.	Adequate survey covering most key sources. Gap / novelty identified but only partially justified.	Sparse or outdated survey. Little or no identification of the gap; chosen approach not justified.
Design, Methodology & Implementation	Design is based on stated criteria, sound analysis and realistic constraints. Methodology is appropriate and well justified. Implementation is complete and robust and makes effective use of modern engineering / software tools.	Design is reasonable and methodology mostly appropriate. Implementation is largely complete; deeper analysis of alternatives or constraints could improve the outcome.	Only one solution considered or alternatives ignored. Weak methodology and incomplete implementation. Many constraints and criteria ignored.
Project Management & Teamwork	Responsibilities delegated fairly and every member contributes valuably. All members attend meetings and meet deliverable deadlines. Schedule, resources and budget are well managed.	Minor inequities in delegation; some members contribute more than others but all meet their responsibilities. Meetings mostly attended and deadlines mostly met.	Major inequities in delegation, with free-riders or dominating members. Meetings and deadlines frequently missed; poor planning of schedule and resources.
Demonstration, Testing & Result Analysis	Develops and follows logical testing / validation / experimental procedures. Selects appropriate tools,	Procedures mostly followed with occasional oversight. Needs some guidance in selecting tools / equipment and is	Does not follow a proper procedure. Cannot select appropriate tools / equipment or operates them incorrectly,

	equipment, platforms or datasets and operates them confidently. Aware of errors / limitations, accounts for them, and interprets and relates the results to the objectives and underlying theory.	tentative in operating them. Aware of errors but accounts for them only minimally.	needing frequent supervision. Makes no attempt to relate results to objectives.
Report / Thesis Writing (Written Communication)	Report is well organised and clearly written; the logic is easy to follow. Wording is precise; figures, tables and analyses enhance understanding. Grammatically correct, free of spelling errors, properly cited and free of plagiarism.	Report is mostly organised and clear; a few sections are harder to follow. Wording is good with minor exceptions; figures are consistent. Few grammatical / spelling errors that do not hinder the reader; some sources not fully cited.	Report lacks organisation and the logic is hard to follow. Figures absent or inconsistent. Grammatical and spelling errors hinder reading. Evidence of plagiarism or inadequate citation.
Research Paper Writing / IPR (Patent or Copyright)	A research / technical paper is drafted to publication standard in the required format, or a patent / copyright application is prepared and filed. Novelty and contribution are clearly articulated, claims and results are well supported, and the work is original with proper citations and no plagiarism.	A paper draft or IPR document is prepared and largely follows the required format. Novelty and contribution are stated but only partially developed, with minor gaps in supporting evidence or formatting.	No paper / IPR document, or a weak draft that does not follow the required format. Novelty or contribution is unclear or unsupported, with evidence of plagiarism or inadequate citation.
Presentation & Oral Communication	Slides are error-free and logically present the work and findings; they are readable and the graphics support the main ideas. Speakers are audible and fluent, do not rely on notes and answer questions accurately. Confident body language and eye contact convey strong engagement.	Slides are error-free and logical and mostly readable. Speakers are mostly fluent with minimal reliance on notes and answer most questions well. Slight discomfort or occasional distracting gestures.	Slides contain errors and lack logical flow; key aspects are absent and graphics confuse the audience. Speakers are inaudible or hesitant and rely heavily on notes, with difficulty answering questions. Body language conveys discomfort.

Program: Computer Engineering **Final Year B.Tech.** **Semester: VII**

Course: Disaster Management and Preparedness (DJS23ICHSX11)

Objectives:

1. To provide basic understanding of hazards, disasters and various types and categories of disaster occurring around the world.
2. To identify extent and damaging capacity of a disaster.
3. To study and understand the means of losses and methods to overcome /minimize it.
4. To understand roles and responsibilities of individual and various organizations during and after disaster.
5. To appreciate the significance of GIS, GPS in the field of disaster management.
6. To understand the emergency government response structures before, during and after disaster.

Outcomes: On completion of the course, the learner will be able to:

1. Apply disaster management principles & guidelines.
2. Conduct risk assessments.
3. Develop community awareness & participation.
4. Utilize Science & Technology tools (GIS, GPS).
5. Prepare disaster management plans.

Disaster Management and Preparedness (DJS23ICHSX11)		
Unit	**Description**	**Duration**
1	**Understanding Disasters & Hazards:** **Definition and types of disasters**: Natural, Man-made and hybrid disasters, **Study of Natural disasters**: Flood, drought, cloud burst, Earthquake, Landslides, Avalanches, Volcanic eruptions, Mudflow, Cyclone, Storm, Storm Surge, climate change, global warming, sea level rise, ozone depletion etc. **Study of Human/Technology Induced Disasters**: Chemical, Industrial and Nuclear disasters, internally displaced persons, road and train accidents Fire Hazards, terrorism, militancy, Hazard & Vulnerability profiles of India (seismic zones, flood-prone areas). India's vulnerability to disasters, and the impact of disasters on National development.	6
2	**Disaster Risk Reduction (DRR) & Mitigation:** **Disaster Management Cycle**: Prevention, Mitigation, Preparedness, Response, Recovery. Need for disaster prevention and mitigation, mitigation guiding principles, challenging areas, structural and non-structural measures for disaster risk reduction. Risk Assessment & Vulnerability Analysis. **Science & Technology**: Use of information management, Geo informatics like RS, GIS, GPS and remote sensing mitigation measure.	6
3	**Disaster Preparedness & Response:** Preparedness Planning, Early Warning Systems (EWS), & Communication. Emergency Response: Search & Rescue, Logistics, Medical Aid. Psychological Response & Management (Trauma, Stress).	4

	Role of IT, Media, Govt., NGOs, & Community.	
4	**Recovery, Rehabilitation & Reconstruction:** Post-disaster damage assessment. Rehabilitation, Reconstruction, & Livelihood Restoration. Sanitation, Hygiene, & Waste Management.	4
5	**Policy, Governance & Capacity Building:** National Disaster Management Authority (NDMA) & Legislation. Institutional Mechanisms & Community Mobilization. Non-Structural Mitigation: Community based disaster preparedness, capacity development and training, awareness and education, contingency plans. **Case studies on disaster (National /International):** Case study discussion of National Disasters: Tsunami (2004), Bhopal gas tragedy, Kerala and Uttarakhand flood disaster, 26th July 2005 Mumbai flood. Case study discussion of International Disasters: Hiroshima – Nagasaki (Japan), Cyclone Phailin (2013), Fukushima, Daiichi nuclear disaster (2011), Chernobyl meltdown	8
	Total	**28**

Books Recommended:

Reference Books and Reports:

1. Disaster Management, by Harsh K. Gupta, Universities Press Publications (2003).
2. Disaster Management: An Appraisal of Institutional Mechanisms in India, by O. S. Dagur, published by Centre for land warfare studies, New Delhi, 2011.
3. Introduction to International Disaster Management, by Damon Copolla, Butterworth Heinemann Elsevier Publications (2015).
4. Disaster Management Handbook, by Jack Pinkowski, CRC Press, Taylor and Francis group (2008).
5. Disaster management & rehabilitation, by Rajdeep Dasgupta, Mittal Publications, New Delhi (2007).
6. Natural Hazards and Disaster Management, Vulnerability and Mitigation, by R B Singh, Rawat Publications (2006).
7. Concepts and Techniques of GIS, by C. P. Lo Albert, K.W. Yonng, Prentice Hall (India) Publications (2006).
8. Risk management of natural disasters, by Claudia G. Flores Gonzales, KIT Scientific Publishing (2010).
9. Disaster Management – a disaster manger's handbook, by W. Nick Carter, Asian Development Bank (2008).
10. Disaster Management in India, by R. K. Srivastava, Ministry of Home Affairs, GoI, New Delhi (2011)
11. The Chernobyl Disaster: Legacy and Impact on the Future of Nuclear Energy, by Wil Mara, Marshall Cavendish Corporation, New York, 2011.
12. The Fukushima 2011 Disaster, by Ronald Eisler, Taylor & Francis, Florida, 2013.

(Learners are expected to refer reports published at national and international level and updated information available on authentic web sites.)

Prepared by Checked by Head of the Department Principal

Program: B.Tech. in Computer Engineering	Final Year B.Tech.	Semester: VII
Course: AI for Language Generation and Understanding (DJS23CCSC701)		

Prerequisite:

Machine Learning, Deep Learning (Neural Networks and RNNs), Engineering Mathematics (Linear Algebra, Calculus, and Statistics).

Course Objectives:

1. To explain the attention mechanism and how it overcomes the limitations of sequential seq2seq models.
2. To develop the ability to build Transformer-based models for machine translation, summarization, and question answering.
3. To enable the use of transfer learning with state-of-the-art pre-trained models such as BERT and T5.

Course Outcomes: On completion of the course, the learner will be able to:

1. Explain the attention mechanism using the queries, keys, and values formulation.
2. Build a neural machine translation model with attention and evaluate it using BLEU and ROUGE.
3. Construct Transformer architectures using self-attention and multi-head attention.
4. Develop a Transformer-based text summarization model.
5. Apply transfer learning with BERT and T5 for downstream NLP tasks.
6. Implement a question-answering system using Hugging Face pre-trained models.

AI for Language Generation and Understanding (DJS23CCSC701)		
Unit	**Description**	**Duration**
1	**Seq2Seq Models and the Attention Mechanism:** Shortcomings of traditional seq2seq models, the seq2seq model with attention, and the queries–keys–values formulation of attention; basic attention and scaled dot-product attention labs.	4
2	**Neural Machine Translation with Attention:** Setup for machine translation, teacher forcing, building an NMT model with attention; evaluation with BLEU and ROUGE-N scores, and decoding strategies — sampling, beam search, and Minimum Bayes Risk. English-to-German translation programming assignment.	4
3	**Transformer Architecture and Self-Attention:** Transformers vs. RNNs, transformer overview and applications, scaled dot-product attention, masked self-attention, multi-head attention, the transformer decoder, and positional encoding.	4
4	**Text Summarization with Transformers:** Building a transformer-based summarizer, masking, and positional encoding in practice; transformer summarizer programming assignment.	4
5	**Transfer Learning and Pre-trained Language Models:** Transfer learning in NLP, the ELMo–GPT–BERT–T5 family, Bidirectional Encoder Representations from Transformers (BERT) and its objective, fine-tuning BERT, the T5 model, multi-task training strategy, and the GLUE benchmark.	5
6	**Question Answering with BERT, T5 and Hugging Face:** Building question-answering models, SentencePiece and Byte-Pair Encoding, and using the Hugging Face library with base and fine-tuned models; question-answering programming assignment.	4

		Total	**25**

Web Resources:

https://www.coursera.or g/learn/attention-models-in-nlp

Program: Computer Engineering **Final Year B.Tech.** **Semester: VIII**

Course: Quantum Computing (DJS23CCPE811)

Pre-requisite:

1. Engineering Mathematics
2. Discrete Structures
3. Cryptography & Information Security
4. Basic programming in Python

Objectives:

1. To introduce the mathematical and quantum-mechanical principles underlying the qubit model of computation.
2. To impart knowledge of quantum gates, circuits, entanglement and the foundational and famous quantum algorithms.
3. To develop practical skills in quantum programming on simulators and NISQ hardware using open-source frameworks.
4. To analyze the impact of quantum algorithms on classical and post-quantum cryptography, and to create awareness of emerging quantum applications and trends.

Outcomes: On completion of the course, the learner will be able to:

1. Apply linear-algebraic and quantum-mechanical formalism to describe qubit states and operations.
2. Construct quantum circuits using single- and multi-qubit gates to demonstrate entanglement and teleportation.
3. Analyze foundational and famous quantum algorithms (Deutsch–Jozsa, Grover, Shor) and their complexity advantages.
4. Implement quantum programs on simulators and NISQ devices using open-source frameworks.
5. Analyze quantum communication and cryptographic protocols, including QKD and post-quantum cryptography.
6. Identify and evaluate emerging applications of quantum computing in optimization, machine learning and chemistry.

Quantum Computing (DJS23CCPE811)		
Unit	**Description**	**Duration**
1	**Introduction to Quantum Computing & Mathematical Foundations** • Motivation: limits of classical computing, Moore's law, Feynman's vision; classical vs quantum computation. • Brief recap of classical bits, gates, reversible logic (Toffoli, Fredkin) and computational complexity classes (P, NP, BQP). • Complex numbers and Euler's formula; vectors, inner product and norms; Dirac (bra-ket) notation. • Matrices, Hermitian and unitary operators; eigenvalues and eigenvectors; tensor (Kronecker) products. • Probability vs probability amplitude; the Born rule and quantum interference.	6

	• Brief history and timeline of quantum computing milestones (Feynman, Deutsch, Shor, Grover, IBM, Google supremacy, NISQ era). **Self-learning Topics:** Quantum complexity theory (QMA and the quantum PCP conjecture), ZX-calculus and diagrammatic reasoning about quantum circuits, Quantum information theory: von Neumann entropy and channel capacity	
2	**Qubits, Single-Qubit Gates and Measurement** • Postulates of quantum mechanics: state space, evolution, measurement and composite systems. • The qubit: superposition, computational basis $\|0\rangle$, $\|1\rangle$; geometric representation on the Bloch sphere. • Single-qubit gates: Pauli X, Y, Z; Hadamard (H); phase gates S and T; rotation gates Rx, Ry, Rz. • Quantum circuit notation; identity and global vs relative phase. • Measurement in the computational and Hadamard bases; projective measurement and collapse. • No-cloning theorem and its implications; reversibility of quantum gates. • Multi-qubit systems and tensor products; product vs entangled states; CNOT, CZ, SWAP, controlled-U, Toffoli; universal gate sets. • Bell states (H + CNOT); quantum teleportation and superdense coding. **Self-learning Topics:** Qudits and higher-dimensional quantum information, Continuous-variable quantum computing, Measurement-based (one-way) quantum computing and cluster states	7
3	**Quantum Algorithms** • Quantum parallelism, oracles and the role of interference. • Deutsch and Deutsch–Jozsa; Bernstein–Vazirani and Simon (bridges to period-finding). • Quantum Fourier Transform: definition, circuit, intuition; quantum phase estimation. • Grover's search: $O(\sqrt{N})$ speedup, oracle and diffusion operator; amplitude amplification. • Shor's factoring algorithm: period/order finding via QFT (overview); complexity advantages. **Self-learning Topics:** The HHL algorithm for solving linear systems, Quantum walks and their algorithmic applications, Amplitude estimation and quantum approximate counting, Dequantization and quantum-inspired classical algorithms	8
4	**Quantum Programming & Simulation** • Quantum programming platforms: Qiskit, Cirq, Q#, Amazon Braket; a first quantum program. • Quantum circuit design; statevector and other simulators; quantum assembly basics. • Execution on cloud quantum computers (IBM Quantum, AWS Braket, Azure Quantum). • Noise, decoherence, gate fidelity, T1/T2; the NISQ era. • Error correction at an intuitive level: bit-flip/phase-flip codes; surface code (overview). • Hands-on implementation of Deutsch and Grover; result visualization. **Self-learning Topics:** Fault-tolerant quantum computing and the threshold theorem, Logical qubits, magic-state distillation, and lattice surgery, Quantum	6

	error mitigation (ZNE, PEC) for NISQ devices, Pulse-level control and OpenPulse	
5	**Quantum Communication & Cryptography** • Quantum channels, state transmission, noise; quantum repeaters. • Quantum Key Distribution: BB84, B92, E91; eavesdropping detection and security intuition. • Shor's impact on RSA, Diffie–Hellman, ECC; "harvest-now, decrypt-later" threat. • Grover's impact on AES and hash functions — effective key-length halving. • Post-Quantum Cryptography: NIST standards (ML-KEM/Kyber, ML-DSA/Dilithium, SLH-DSA/SPHINCS+); lattice-, code-, hash-based families. • QKD vs PQC; the road to a quantum-safe Internet. **Self-learning Topics:** Quantum internet architecture and entanglement distribution, Device-independent and measurement-device-independent QKD, Quantum money and quantum digital signatures, Migrating real systems to PQC: crypto-agility and hybrid schemes	8
6	**Applications, Hardware & Emerging Trends** • Physical qubit implementations: superconducting, trapped ion, photonic, neutral atom; DiVincenzo criteria. • Optimization: combinatorial problems, quantum annealing, QAOA (introductory). • Quantum chemistry and VQE; quantum machine learning: feature spaces, variational classifiers (conceptual). • Quantum sensing and metrology (brief). • Emerging trends: quantum-inspired classical algorithms, India's National Quantum Mission, the quantum-safe transition roadmap. **Self-learning Topics:** Quantum advantage in practice — real gains vs. hype, Recent quantum-error-corrected and logical-qubit demonstrations, Topological qubits and Majorana-based approaches, Quantum-centric supercomputing and hybrid HPC–quantum workflows	7
	Total	**42**

Books Recommended:

Text Books:

1. Ahmed Banafa, "Introduction to Quantum Computing", River Publishers, 2023.
2. Thomas G. Wong, "Introduction to Classical and Quantum Computing", Rooted Grove, 2022.
3. Venkateswaran Kasirajan, "Fundamentals of Quantum Computing -Theory and Practice", Springer, 2021.
4. Maria Schuld, Francesco Petruccione, "Machine Learning with Quantum Computers", Springer International Publications, 2021.
5. Michael A. Nielsen and Isaac L. Chuang, "Quantum Computation and Quantum Information", 10th Anniversary Edition, Cambridge University Press, 2010.

Reference Books:

1. Vladimir Silva, "Practical Quantum Computing for Developers", Apress, 2018.
2. Eleanor Rieffel and Wolfgang Polak, "Quantum Computing: A Gentle Introduction", MIT Press, 2011.

3. Phillip Kaye, Raymond Laflamme and Michele Mosca, "An Introduction to Quantum Computing", Oxford University Press, 2007.
4. N. David Mermin, "Quantum Computer Science: An Introduction", Cambridge University Press, 2007.

Web Resources:

1. Introduction to Quantum Computing: Quantum Algorithms and Qiskit
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs89
2. Quantum Computing: Algorithms and Limitations
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs06

Program: Computer Engineering **Final Year B.Tech.** **Semester: VIII**

Course: Human Computer Interaction (DJS23CCPE812)

Pre-requisite:

Web Technologies; Experience in designing interfaces for applications and web sites. Basic knowledge of designing tools and languages like HTML, Java, etc.

Objectives:

1. Learn the foundation of human machine interaction.
2. Understand the importance of human psychology in designing good interfaces.
3. Be aware of mobile interaction design and its usage in day – to – day activities.
4. Understand various design technologies to meet user requirements.
5. Encourage to indulge into research in Machine Interaction Design.

Outcomes: On completion of the course, the learner will be able to:

1. Identify User Interface (UI) design principles.
2. Analysis of effective user-friendly interfaces.
3. Apply Interactive Design process in real world applications.
4. Evaluate UI design and justify.
5. Create application for social and technical task.

Human Computer Interaction (DJS23CCPE812)		
Unit	**Description**	**Duration**
1	**FOUNDATIONS OF HMI**: The Human: History of User Interface Designing, I/O channels, Hardware, Software and Operating environments, The Psychopathology of everyday Things, Psychology of everyday actions, Reasoning and problem solving. The computer: Devices, Memory, processing and networks. Interaction: Models, frameworks, Ergonomics, styles, elements, interactivity, Paradigms. **Self-learning Topics:** Study tools like Figma. Redesign UIs, test layouts.	7
2	**DESIGN & SOFTWARE PROCESS:** Mistakes performed while designing a computer system, Human interaction with computers, importance of human characteristics, human consideration, Human interaction speeds. Interactive Design basics, process, scenarios, navigation, Iteration and prototyping. HMI in software process: software life cycle, usability engineering, Prototyping in practice, design rationale. Design rules: principles, standards, guidelines, rules. Recognize the goals, Goal directed design process. Evaluation Techniques: Universal Design. **Self-learning Topics:** Design thinking and the double-diamond model, Design systems and component libraries (atomic design), Lean UX and agile-UX integration	8
3	**GRAPHICAL USER INTERFACE:** The graphical User Interface: Popularity of graphics, the concept of direct manipulation, graphical systems, Characteristics. Web user Interface: Interface popularity, characteristics. The merging of graphical Business systems and the Web. Principles of user interface design	8

	Self-learning Topics: Conversational UI and chatbot design, Voice user interfaces (VUI) and conversation design, Zero UI and ambient / invisible interfaces	
4	**SCREEN DESIGNING**: Design goals , Screen planning and purpose, organizing screen elements, ordering of screen data and content , screen navigation and flow, Visually pleasing composition, amount of information, focus and emphasis, presentation information simply and meaningfully, information retrieval on web, statistical graphics, Technological consideration in interface design. **Self-learning Topics:** Interactive data-dashboard and visualization design, Dark patterns and ethical interface design, Motion design and microinteractions	8
5	**INTERFACE DESIGN FOR MOBILE DEVICES**: Mobile Ecosystem: Platforms, Application frameworks: Types of Mobile Applications: Widgets, Applications, Games, Mobile Information Architecture, Mobile 2.0, Mobile Design: Elements of Mobile Design, Tools. **Self-learning Topics:** Wearable and smartwatch interface design, AR / VR / XR spatial interaction design, Cross-device and continuity experiences	6
6	**INTERACTION STYLES AND COMMUNICATION:** Windows: Characteristics, Components, Presentation styles, Types of Windows, Management, operations. Text messages: Words, Sentences, messages and text words, Text for web pages. Icons, Multimedia and colors **Self-learning Topics:** Gesture, gaze, and brain–computer interfaces (BCI), Haptics and tactile-feedback design, Natural-language interfaces and LLM-powered UX	6
	Total	**42**

Books Recommended:

Text Books:

1. Kasper Hornbæk, Per Ola Kristensson, Antti Oulasvirta, **"**Introduction to Human-Computer Interaction**",** Oxford University Press, 2026
2. J Dhilipan, V Saravanan, R Agusthiyar, Human Machine Interaction in the Digital Era Towards Conversational Artificial Intelligence, CRC Press, 2024.
3. Gerard Jounghyun Kim, "Human–Computer Interaction", CRC Press, 2015

Reference Books:

1. Guy A. Boy, The Handbook of Human-Machine Interaction: A Human-Centered Design Approach, CRC Press, 2011.
2. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, ―Human Computer Interaction‖, 3rd Edition, Pearson Education, 2004.

Web Resources:

1. Human Computer Interaction
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs70
2. User-Centric Computing for Human-Computer Interaction
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc20_cs45

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **Final Year B.Tech.** **Semester: VIII**

Course: Geographic Information System (DJS23CCPE813)

Pre-requisite:

1. Basic knowledge of computer systems, databases, and programming concepts.
2. Fundamental understanding of geography, maps, and spatial data representation.

Objectives:

1. To introduce the fundamental concepts, components, and applications of Geographic Information Systems (GIS).
2. To understand spatial and non-spatial data models and their representation in computer systems.
3. To study raster and vector data structures and compare their advantages and limitations.
4. To design and query spatial databases using spatial data types, spatial indexing, spatial SQL and PostGIS.
5. To develop knowledge of map digitization, editing, vector topology creation and validation, and attribute data management.
6. To understand Digital Elevation Models and triangulation methods (Delaunay/TIN), and apply GIS and Remote Sensing in Computer Engineering domains such as smart cities, navigation, and AI-based image processing.

Outcomes: After successful completion of the course, students will be able to:

1. Explain the fundamental concepts, components, and applications of GIS and Remote Sensing systems.
2. Differentiate raster and vector data models with respect to storage, processing, and application areas.
3. Implement spatial database queries using spatial data types, spatial indexing, spatial SQL and PostGIS.
4. Construct vector topology and attribute mappings through digitization, editing, and validation techniques.
5. Design Digital Elevation Models using triangulation (Delaunay/TIN) and grid structures for surface analysis.
6. Apply GIS and Remote Sensing techniques to real-world applications such as navigation, smart-city planning, and satellite image analysis.

Geographic Information System (DJS23CCPE813)		
Unit	**Description**	**Duration**
1.	**Introduction to GIS** Definition, sources of data, types of data, concept of space and time in GIS, spatial information theory, history of GIS, elements and components of GIS, objectives of GIS, hardware and software requirements of GIS, applications of GIS. **Self-Learning Topics:** Volunteered Geographic Information (VGI) and citizen-science mapping, Spatial Data Infrastructure (SDI) and open geospatial-data policy, Web GIS and cloud-based geospatial platforms	6
2.	**Spatial Data Models – Raster and Vector** Layers and coverage; conceptual models of spatial information; representation of spatial data models in computer — raster and vector models; raster encoding	7

	and compression (run-length, quadtree); vector primitives (point, line, polygon); comparative overview between raster and vector models with respect to storage, processing and application. **Self-Learning Topics:** 3D GIS and CityGML / 3D city models, LiDAR and point-cloud data processing, Spatio-temporal and moving-object data models	
3.	**Spatial Databases and Spatial Data Management** Review of DBMS models — hierarchical, network and relational structures, and their limitations for spatial data. **Spatial database concepts**: spatial data types (geometry vs geography); OGC Simple Features specification; spatial reference systems and SRID. **Spatial indexing**: R-tree, R+-tree, quadtree, grid index. Spatial SQL and spatial operators (intersection, containment, buffer, distance, spatial joins). **PostGIS**: architecture, geometry/geography types, ST_ spatial functions, spatial queries and relationships, raster support; spatial database design and query optimization. **Self-Learning Topics:** Distributed spatial processing (Apache Sedona / GeoMesa), NoSQL spatial databases (MongoDB geospatial, Elasticsearch), Vector tiles and Cloud-Optimized GeoTIFF (COG)	8
4.	**Vector Topology, Digitizing, Editing and Structuring of Map Data** **Digitizing**: manual, semi-automatic and automatic. Editing: error detection and correction, snapping and fuzzy tolerances. **Vector topology**: concept of topology; topological relationships — adjacency, connectivity, containment; planar enforcement. **Topological data structures**: arc–node structure, DIME, POLYVRT. Building topology, topological rules and validation, topology cleaning. Attribute data management, attribute map generation, and linking of spatial and attribute data. **Self-Learning Topics:** Map generalization and multi-scale representation, Conflation and map-matching algorithms, Geospatial metadata and quality standards (ISO 19115)	8
5.	**Digital Elevation Models and Triangulation** Need and sources of DEM. **Structures of DEM**: regular grid, contour/line-based, and triangulated irregular network (TIN). **Triangulation**: Delaunay triangulation and the empty-circle criterion; constrained Delaunay triangulation; Voronoi / Thiessen polygons and their duality with Delaunay. TIN generation and TIN vs grid trade-offs. **Surface interpolation**: IDW, spline, kriging (introductory); surface analysis — slope, aspect, and contour generation. **Self-Learning Topics:** Geostatistics deep-dive: variograms and kriging variants, Hydrological modelling and flood simulation from terrain, InSAR and photogrammetric DEM generation	7
6.	**Applications of Remote Sensing and GIS** GIS software development; spatial database management and optimization; GPS-based navigation and route optimization systems; smart city planning and infrastructure monitoring; satellite image processing using Artificial Intelligence and Machine Learning.	6

	Self-Learning Topics: Deep learning for satellite imagery (semantic segmentation, object detection), Google Earth Engine for planetary-scale geospatial analysis, City digital twins and GIS–BIM integration, GeoAI and spatial machine learning	
	Total	**42**

Books Recommended:

Textbooks:

1. Paul A. Longley, Michael F. Goodchild, David J. Maguire, David W. Rhind , Geographic Information Science and Systems, 5th Edition, Wiley, 2021.
2. Kang-Tsung Chang, Introduction to Geographic Information Systems, 10th Edition, McGraw-Hill Education, 2023.
3. Paul Bolstad, GIS Fundamentals: A First Text on Geographic Information Systems, 7th Edition, XanEdu Publishing, 2022.
4. Thomas Lillesand, Ralph W. Kiefer, Jonathan Chipman, Remote Sensing and Image Interpretation, 8th Edition, Wiley, 2022.
5. Peter A. Burrough, Rachael A. McDonnell, Christopher D. Lloyd, Principles of Geographic Information Systems, Oxford University Press, 2015.

Reference Books:

1. Surya S Durbha, Jibonananda Sanyal, Lexie Yang, Sangita S Chaudhari, Ujwala Bhangale, Ujwala Bharambe, Kuldeep Kurte, Advances in Scalable and Intelligent Geospatial Analytics Challenges and Applications, CRC Press, 2023.
2. Michael Zeiler, Modeling Our World: The ESRI Guide to Geodatabase Design–, ESRI Press, 2000.
3. David J. Maguire, Michael Batty, Michael F. Goodchild, GIS, Spatial Analysis and Modeling, ESRI Press, 2005.
4. John R. Jensen, Remote Sensing of the Environment: An Earth Resource Perspective, 3rd Edition, Pearson, 2013.
5. Sathheesh Gopi, R. Sathikumar, N. Madhu, Advanced Surveying: Total Station, GIS and Remote Sensing, Pearson Education, 2006.
6. Michael N. DeMers, Fundamentals of Geographic Information Systems, 5th Edition, Wiley, 2008.

Web Resources:

1. Geographic Information Systems
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_ce26
2. Remote Sensing and GIS
   https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_ce84

Program: Computer Engineering **Final Year B.Tech.** **Semester: VII/ VIII**

Course: Internship (OJT) cum Project (DJS23IPELX14)

Pre-requisite:

1. Fundamental knowledge of engineering concepts and programming
2. Basic understanding of professional ethics and communication skills
3. Exposure to laboratory courses and mini-projects
4. Teamwork and problem-solving skills

Objectives:

1. To provide practical exposure to industrial and professional environments.
2. To bridge the gap between academic learning and industry practices.
3. To develop technical, communication, teamwork, and managerial skills.
4. To enable application of engineering knowledge for solving real-world problems.
5. To familiarize students with modern tools, technologies, and professional ethics.
6. To prepare students for industry-oriented careers through project implementation and professional experience.

Outcomes: On completion of the course, the learner will be able to:

1. Apply engineering knowledge and technical skills in industrial environments.
2. Analyze and solve practical engineering problems using appropriate tools and techniques.
3. Demonstrate professional ethics, teamwork, communication, and organizational skills.
4. Utilize modern engineering tools and technologies for assigned tasks and projects.
5. Prepare technical reports, presentations, and project documentation effectively.
6. Adapt to professional work culture and demonstrate readiness for professional careers.

Overview of the OJT Process:

The On-Job Training (OJT) program begins with an introduction to workplace ethics, safety practices, organizational culture, and professional conduct to prepare students for industrial environments. Students are then exposed to industrial processes, workflow practices, tools, and modern technologies relevant to their domain of study.

During the technical training phase, students work on assigned industrial tasks and projects under the guidance of faculty mentors and industry supervisors. The program emphasizes the application of engineering concepts, teamwork, communication skills, and problem-solving abilities in real-world professional settings.

The later stages of the OJT focus on project implementation, report writing, technical documentation, and presentation preparation. Students are required to maintain logbooks and progress records throughout the training period. At the end of the semester, students present their project work through demonstrations and viva-voce examinations, followed by comprehensive evaluation and feedback.

Internship (OJT) cum Project Schedule, Duration and Credits

The Internship / On-Job Training (OJT) cum Project is an integral part of the Final Year B.Tech. program and carries a total of 18 academic credits toward completion of the degree requirements. Each

credit corresponds to 40 hours of work, resulting in approximately 550–600 hours of industrial training and project work over one semester.

The full-time OJT program is conducted during the designated semester as per the academic schedule. The duration of the OJT is one complete semester in the final year. Faculty mentors are assigned for mentoring, monitoring, and evaluating students throughout the OJT period. Approximately 50% of the students undergo OJT during Semester VII while the remaining students continue academic activities in the institute. In Semester VIII, the arrangement is interchanged to ensure equal academic and industrial exposure for all students.

To avoid academic scheduling conflicts and ensure fair evaluation, oral, practical, and End Semester Examinations for both Semester VII and Semester VIII will be conducted at the end of Semester VIII. Students are issued a combined marksheet covering both semesters.

Types of External OJT Opportunities

Sr. No.	Type of OJT	Examples / Description
1	Industry OJT	Industrial internships with or without stipend
2	Government / PSU OJT	BARC, SAMEER, Railways, ISRO, etc.
3	Research OJT	IITs, NITs, IISc, IIMs, and other research institutes
4	Institute-Based OJT	OJT at institutes approved by DJSCE and departments
5	Other Approved OJT	Any approved industrial/research work relevant to the domain

OJT / Internship Sources

To facilitate OJT and internship opportunities, AICTE has signed Memorandums of Understanding (MoUs) with various industries, government organizations, and training institutions. These collaborations provide students with industrial exposure, skill development opportunities, research internships, and professional training through multiple recognized platforms and organizations. Students may also approach organizations such as the Board of Apprenticeship Training (BOAT) and Board of Practical Training (BOPT) for additional training opportunities.

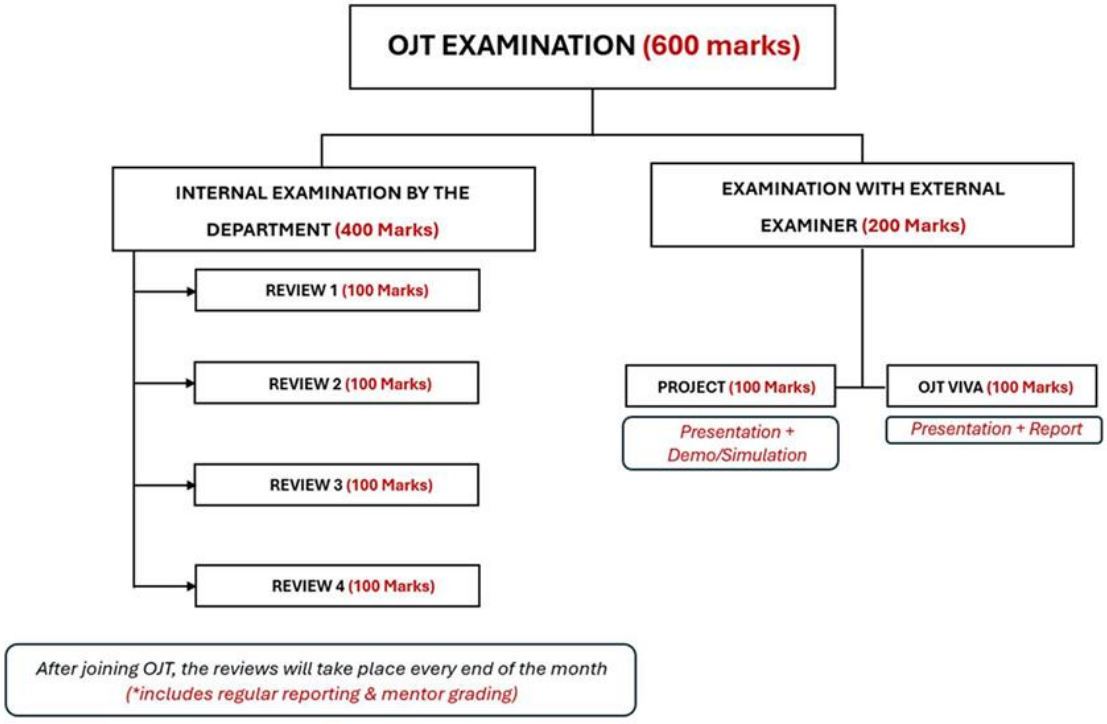
Major OJT / Internship Sources

Sr. No.	Organization / Platform	Nature of Opportunity
1	Internshala	Internship and industry training opportunities
2	NETiit (Taiwan)	International internship opportunities
3	HireMee	Skill-based internships and employability support
4	IGNCA	Research and cultural project opportunities
5	CCEI, Republic of Korea	International innovation and creative economy internships
6	IIWM, Bangalore	Waste management and sustainability projects
7	Engineering Council of India (ECI)	Engineering and professional development activities

8	Fourth Ambit	Industry training and technical internships
9	LinkedIn	Professional networking and internship opportunities
10	Telecom Sector Skill Council (TSSC)	Telecom and communication technology training
11	SCHOLARSMERIT	Internship and skill enhancement programs
12	Studenting Era	Student internship and academic support activities
13	MSME	Industry exposure and entrepreneurship opportunities
14	BOAT	Apprenticeship and industrial training programs
15	BOPT	Practical and apprenticeship-based training opportunities

Evaluation Metrics:

Sr. No.	Evaluation Component	Marks
1	Internal Review 1	100
2	Internal Review 2	100
3	Internal Review 3	100
4	Final Internal Review	100
5	End Semester Evaluation (OJT Viva/Presentation)	100
6	Project Report & Demonstration	100
	Total	**600**



Evaluation Parameters

The student shall be evaluated based on the following parameters:

- Understanding of assigned work
- Progress and task completion
- Technical skill development

- Learning ability and initiative
- Professional behaviour and discipline
- Communication and presentation skills
- Problem solving and contribution
- Quality of work and innovation
- Logbook maintenance and report documentation
- Application of engineering knowledge

Necessary Documents

Students are required to submit and maintain the following documents during the OJT period:

1. No Objection Certificate (NOC)
2. Undertaking for opting OJT
3. OJT Joining Form
4. Parent/Guardian Undertaking Form
5. Contact Details and OJT Objectives Form
6. Student Logbook / Daily Activity Record
7. Internal Review Evaluation Sheets
8. External Supervisor Evaluation Form
9. Internal Supervisor Feedback Form
10. Student Feedback Form
11. SDG Mapping Form
12. Final OJT Report
13. Project Report and Presentation
14. OJT Completion Certificate

Scheme of Examination

Component	Marks
Continuous Internal Assessment	400
End Semester OJT Evaluation	100
Project Report & Presentation	100
Total	**600 Marks**

Credits

- Total Credits: 18
- 1 Credit = 40 Hours of Work
- Minimum Duration: 550–600 Hours of OJT and Project Work over one semester.

Program: B.Tech. in Computer Engineering	Final Year B.Tech.	Semester: VIII
Course: Multi-Agent Systems (DJS23CCSC801)		

Prerequisite:

Python Programming and Foundations of Generative AI / Large Language Models.

Course Objectives:

1. To introduce the architecture and design principles of single- and multi-agent AI systems.
2. To develop the ability to build collaborative agent teams using tools, memory, guardrails, and orchestration with CrewAI.
3. To enable the deployment, monitoring, and evaluation of production-ready, scalable multi-agent systems.

Course Outcomes: On completion of the course, the learner will be able to:

1. Explain the foundations of AI agents and what makes multi-agent systems intelligent.
2. Build multi-agent systems that plan, reason, and collaborate to automate workflows.
3. Apply memory, guardrails, tools, and the Model Context Protocol to control agent behavior.
4. Orchestrate agents using collaboration patterns and CrewAI Flows.
5. Evaluate multi-agent system reliability through monitoring, tracing, and structured evaluation.
6. Analyze real-world enterprise deployments of AI agents across industries.

Multi-Agent Systems (DJS23CCSC801)		
Unit	**Description**	**Duration**
1	**Foundations of AI Agents:** What AI agents are, real-world use cases, what makes an AI agent intelligent, and why the AI agent revolution is happening now; building your first AI agent.	4
2	**Designing and Building Multi-Agent Systems:** Planning multi-agent systems, tuning behavior through context engineering, running systems in production, and tactics for debugging, observing, and optimizing; building multi-agent systems and an automated code reviewer.	4
3	**Controlling Agents — Memory, Guardrails and Tools:** Understanding agent workflows, incorporating memory and knowledge, controlling agents with guardrails and execution hooks, using and building custom tools, and adopting the Model Context Protocol (MCP); building a meeting-preparation co-pilot and extending the code reviewer.	4
4	**Orchestrating Multi-Agent Systems:** Agent collaboration and communication, the Agent-to-Agent (A2A) Protocol, and orchestrating agents with CrewAI Flows using sequential, parallel, hierarchical, hybrid, and async processes; building coordination patterns and a deep research flow.	4
5	**Reliability, Monitoring and Deployment:** Tactics for building reliable systems, monitoring and observability with tracing and sampling, CI/CD for agents, and training agents with human-in-the-loop feedback and structured evaluation (LLM-as-a-Judge); creating a Flow for automatic code review.	4
6	**Applying AI Agents in Business:** Agent applications across industries and functions, finding high-value use cases, and real-deployment case studies featuring Exa, Snyk, Weaviate, and AB InBev; the future of AI agents.	2
	Total	**22**

Web Resources:

https://www.coursera.o rg/learn/design-develop-and-deploy-multi-agent-systems-with-crewai

Prepared by Checked by Head of the Department Principal