Shri Vile Parle Kelavani Mandal's

Dwarkadas J. Sanghvi College of Engineering

(Autonomous College Affiliated to the University of Mumbai)

Scheme and detailed syllabus (DJS23)

Third Year B. Tech

in

Artificial Intelligence (AI) and Data Science

(Semester VI)

Prepared by: - Board of Studies in Artificial Intelligence (AI) and Data Science

To be Recommended by: - Academic Council of Dwarkadas J. Sanghvi College of Engineering

To be Approved by: - Governing Body of Dwarkadas J. Sanghvi College of Engineering

with effect from Academic Year 2025-26

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Scheme for Third Year Undergraduate Program in Artificial Intelligence (AI) and Data Science: SEM VI (Autonomous)
Academic Year 2025-26

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 (A)	Term Test 1 (TT1)	Term Test 2 (TT2)	Term Test 3 (TT3)	Term Test Total (TT1 + TT2 + TT3)	Term Work	CA Total (B)			
1	DJS23SCPC601	Deep Learning	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	
	DJS23SLPC601	Deep Learning Laboratory	--	2	--	1	2	--	--	--	25	25	--	--	--	--	25	25	50	1	4
2	DJS23SCPC602	Large Language Models	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	
	DJS23SLPC602	Large Language Models Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	4
3	DJS23SLPC603	Advanced Java Programming Laboratory	--	4	--	2	2	--	--	--	25	25	--	--	--	--	25	25	50	2	2
4@	DJS23SCPE611	Advanced Data Structures	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	
	DJS23SLPE611	Advanced Data Structures Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	4
	DJS23SCPE612	Financial Time Series Analysis	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	
	DJS23SLPE612	Financial Time Series Analysis Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	4
	DJS23SCPE613	Probabilistic Graph Models	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	
	DJS23SLPE613	Probabilistic Graph Models Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	4
5@	DJS23SCPE614	Big Data Analytics	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	
	DJS23SLPE614	Big Data Analytics Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	4
	DJS23SCPE615	System Design	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	
	DJS23SLPE615	System Design Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	4
	DJS23SCPE616	Blockchain Technology	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	
	DJS23SLPE616	Blockchain Technology Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	4
6	DJS23SCMD601	MLOps	2	--	--	2	2	60	--	--	--	60	15	15	10	40	--	40	100	3	
7	DJS23SLMD601	MLOps Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	3
8	DJS23IPSCX04	Innovative Product Development IV	--	2	--	1	2	--	--	--	25	25	--	--	--	--	25	25	50	1	1
9	DJS23ICHSX09	Constitution of India	1	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	
		Total	**15**	**14**		**22**	**24**	**300**	**100**		**75**	**475**	**75**	**75**	**50**	**200**	**175**	**375**	**850**	**22**	**22**

@ Elective Course

Prepared by | Checked by | Head of the Department | Vice Principal | 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 Vice-Principal Principal

Program: B. Tech in Artificial Intelligence (AI) and Data Science **T.Y. B. Tech** **Semester: VI**
Course: Deep Learning (DJS23SCPC601)
Course: Deep Learning Laboratory (DJS23SLPC601)

Prerequisite: Natural Language Processing, Machine Learning.

Course Objectives: The Objectives of course are

1. To introduce fundamental concepts of artificial neural network and different learning algorithms: supervised and unsupervised neural networks
2. Develop in-depth understanding of the key techniques in designing Deep Network and GAN.
3. To expose Deep Network based methods to solve real world complex problems.
4. To explore applications and challenges in deep learning

Course Outcomes: Students will be able to

1. Understand the fundamentals of deep neural networks and their training mechanisms.
2. Apply optimization and regularization techniques to improve model performance.
3. Design and implement CNN models for supervised learning tasks.
4. Develop solutions for sequence learning applications using recurrent networks.
5. Analyze unsupervised learning techniques for dimensionality reduction and data reconstruction.
6. Evaluate recent trends in adversarial networks and generative models.

Deep Learning (DJS23SLPC601)		
Unit No	**Description**	**Duration**
1	**Supervised Learning Networks :** Feedforward DNN Perceptron, Representational power of Perceptron, The Perceptron Training Rule, Multilayer perceptron: Delta training rule; Multilayer Networks: A differentiable Threshold Unit (Sigmoid Neurons), Representational Power of Feedforward Networks; Activation functions: Tanh, Logistic, Linear, Softmax, ReLU, Leaky ReLU, Loss functions: Squared Error loss, Cross Entropy, Choosing output function and loss function	07
2	**Optimization:** Learning with backpropagation: EBPTA, Learning Parameters: Gradient Descent (GD), Stochastic and Mini Batch GD, Momentum Based GD, Nesterov Accelerated GD, AdaGrad, Adam, RMSProp, Convergence and local minima, stopping criteria. Regularization: Regularization for Deep Learning: Parameter Norm Penalties, Dataset Augmentation, Noise Robustness, Early Stooping, Sparse Representation, Dropout.	06
3	**Convolutional Neural Networks:** Convolution operation, Padding, Stride, Relation between input, output and filter size, CNN architecture: Convolution layer, Pooling Layer, Weight Sharing in CNN, Fully Connected NN vs CNN, Variants of basic Convolution function, 2D Convolution ConvNet Architectures: LeNet Architecture, AlexNET: Architecture, ResNet : Architecture, ConvNeXt, EfficientNET, Applications: object detection and recognition tasks, medical image analysis, image classification	06
4	**Sequence Modelling:** Sequence Learning Problem, Unfolding Computational graphs, Recurrent Neural	07

	Network, Bidirectional RNN, Backpropagation Through Time (BTT), Limitation of vanilla RNN, Vanishing and Exploding Gradients, The Long Short-Term Memory, GRU, Deep recurrent Networks. Applications: Sentiment analysis, stock prices or market trends	
5	**Unsupervised Learning Networks:** Kohonen Self-Organizing Feature Maps – architecture, training algorithm Autoencoders:Introduction, comparison with PCA, Linear Autoencoder, Undercomplete Autoencoder, Overcomplete Autoencoders, Regularization in Autoencoders, Denoising Autoencoders, Sparse Autoencoders, Contractive Autoencoders, Variational Autoencoders (VAEs) Applications: image compression, feature extraction, risk assessment and fraud detection	**08**
6	**Adversarial Networks** Generative Vs Discriminative Modeling, Generative Adversarial Networks (GAN) Architecture, GAN challenges: Oscillation Loss, Mode Collapse, Uninformative Loss, Hyperparameters, Tackling GAN challenges, Wasserstein GAN, Cycle GAN, Neural Style Transfer Diffusion Models: Introduction, Comparison with GANs Applications: Image synthesis or style transfer, Data Augmentation	**08**
	Total	**42**

List of Experiments:

1. Implement Boolean gates using perceptron.
2. Implement representation power of perceptron.
3. Implement backpropagation algorithm from scratch.
4. Train CNN Models for Image Classification Tasks.
5. Evaluate the Effect of Optimizer SGD on Model Performance.
6. Evaluate the Effect of Optimizer Adam on Model Performance.
7. Compare the Performance of PCA and Autoencoders on Dimensionality Reduction Tasks.
8. Sequence Classification Using RNN or GRU (e.g., Sentiment Analysis or Activity Recognition).
9. Anomaly detection using Self-Organizing Network.
10. Compare the performance of PCA and Autoencoders on a given dataset.
11. Train Variational Autoencoders (VAEs) for Image Reconstruction.
12. Build Generative adversarial model for fake (news/image/audio/video) prediction.
13. Generate Synthetic Data Using Diffusion Models and Evaluate Results.
14. Mini Project

Minimum eight 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.

Textbooks:

1. Asaton Zhang, Zhacary Lipton, Mu Li and Alex Smola , Dive into Deep Learning: December 2023
2. Simon Prince ,Understanding Deep Learning, MIT Press, Dec2023
3. Simon Haykin, Neural Networks and Learning Machines, Pearson Prentice Hall, 3rd Edition, 2010
4. S. N. Sivanandam and S. N. Deepa, Introduction to Soft Computing, Wiley India Publications, 3rd

Edition, 2018
5. M. J. Kochenderfer, Tim A. Wheeler. —Algorithms for Optimization‖, IT Press
6. David Foster, Generative Deep Learning, O'Reilly Media, 2019
7. Denis Rothman, Hands-On Explainable AI (XAI) with python, Packt, 2020

Reference Books:

1. Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, An MIT Press, 2016
2. François Chollet, Deep Learning with Python, Manning Publication, 2017. Xu, Alex, System Design Interview – An Insider's Guide
3. Josh Patterson, Adam Gibson, Deep Learning: A Practitioner's Approach, O'Reilly Publication, 2017
4. Andrew W. Trask, Grokking, Deep Learning, Manning Publication, 2019
5. John D. Kelleher, Deep Learning, MIT Press Essential Knowledge series, 2019
6. Douwe Osinga. —Deep Learning Cookbook‖, O'REILLY, SPD Publishers, Delhi

Web Links:

1. Deep Learning: https://vlab.spit.ac.in/ai/#/experiments
2. Deep learning book https://www.deeplearningbook.org/
3. Deep learning all videos: https://www.cse.iitm.ac.in/~miteshk/CS6910.html
4. Deep Learning Specialization: https://www.coursera.org/specializations/deep-learning

Online Resources:

1. Deep Learning, IIT Ropar NPTEL course by Prof. Sudarshan Iyengar,Dr. Padmavati https://nptel.ac.in/courses/106106184

Prepared by Checked by Head of the Department Vice Principal Principal

Program: B. Tech in Artificial Intelligence (AI) and Data Science **T.Y. B. Tech** **Semester: VI**

Course: Large Language Models (DJS23SCPC602)

Course: Large Language Models Laboratory (DJS23SLPC602)

Prerequisite: Natural Language Processing, Deep Learning

Course Objectives: The Objectives of course are

1. Introduce the fundamental concepts and applications of Generative AI
2. Provide in-depth understanding of Transformer architecture, the core building block of most Large Language Models (LLMs)
3. Explore various LLM architectures and techniques like BERT, prompt engineering, and fine-tuning
4. Develop problem-solving abilities for common system design challenges
5. Introduce students to Multimodal LLMs that can process and understand different data modalities

Course Outcomes: Students will be able to

1. Introduce the fundamental concepts and applications of Generative AI and to provide in-depth understanding of Transformer architecture, the core building block of Large Language Models (LLMs)
2. Explore various LLM architectures and techniques like BERT, GPT-3, T5 and Large Reasoning Models
3. Apply prompt engineering techniques for effective LLM interaction and understand the concept of Retrieval Augmented Generation (RAG) and its role in LLMs
4. Understand the different data modalities using Multimodal Architectures

Large Language Models (DJS23SCPC602)		
Unit No	**Description**	**Duration**
1	**Introduction to Generative AI &Transformer Architecture:** Domains of Generative AI, Text Generation, Image Generation, Music Generation, Video Generation. Limitations of RNN & LSTM, Tokenization, Transformer Architecture : encoders, decoders, attention mechanisms - types, Self-attention vs Flash Attention, feed-forward layer, RETRO Transformer, Reinforcement Learning with AI Feedback (RLAIF), Reinforcement Learning from Human Feedback (RLHF)	08
2	**Language Models - Unveiling the Power of Words :** BERT (Bidirectional Encoder Representations from Transformers) and its applications, exploring other notable LLM architectures (e.g.GPT-3, T5), Mixture of Experts (MoE), various benchmarks to evaluate LLMs – perplexity, BLEU score, Needle in a Haystack	06
3	**Large Reasoning Models :** Deep seek- v2: Multi head Latent Attention, Deep seek MoE, Knowledge Distillation, Mistral 7-B Architecture: sliding Window attention, Grouped Query Attention, Titans: Learning to memorize at test time, Knowledge Distillation, QWQ models	05
4	**Prompt Engineering &Agentic AI:** Introduction to prompt, examples of prompt, prompt engineering, prompt techniques, zero shot, one shot, few-shot learning, Agentic AI- a chain of thought , ReAct , self-consistency, Tree of thought, Multimodal CoT, Graph prompting, Large Action Models(LAMs), LLM based Agents, Auto Gen	07

5	**Retrieval Augmentation & Generation (RAG) and Fine-tuning for LLMs:** Understanding Retrieval and vector, vector storage: vector indexing and retrieval Algorithms: Annoy, HSNV, Inverted File System, LSH, vector quantization techniques: Scalar, Product, Binary, vector libraries, vector databases, Loading and retrieving in Lang Chain, Document loaders, Retrievers in Lang Chain. Fine-tuning: Quantization, PEFT, Full-Fine-tuning vs LoRA vs QLoRA, Fine-Tuning LLMs for different downstream	08
6	**Multimodal Architectures - Beyond Text** Introduction to Multimodal LLMs, Exploring architectures for Multimodal LLMs: Vision Transformer, Next GPT, GPT-4V, **Vision-Language Fusion Models** – BLIP-2, Flamingo, LLaVA, MiniGPT-4,**Temporal & Cross-Modal Models** – VideoLLaMA, ImageBind, Empowering Time Series Analysis with Large Language Models	08
	Total	**42**

List of Experiments:

1. Case study on Applications of Generative AI
2. Case study on role of Artificial Intelligence in achieving the Sustainable Development Goals
3. Fine Tuning Pre-trained Model On Custom Dataset (synthetic data) Using Transformer
4. Build your own LLM from scratch
5. Query PDF using Lang Chain and Pine cone
6. Fine Tune Mistral7-B With Custom Dataset Using LoRA And QLoRA Techniques
7. Using in-built tools and creating custom tools for ReAct agent in Langchain.
8. Question Answering Application using LLM based agents.
9. Understanding various retrievers in Langchain
10. Case study on comparison of Large Reasoning Models
11. Understanding multimodal models like Gemini vision
12. Build a simple multimodal generative model that combines text and image inputs to generate captions
13. Fine-tune or evaluate open-source BLIP-2, LLaVA, or MiniGPT-4 on image captioning, visual question answering (VQA), and instruction-following.
14. Mini project

Minimum ten 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

Textbooks:

1. Ben Auffarth, Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT, and other LLMs by Packt Publishing, 2023
2. Valentina Alto, Modern Generative AI with ChatGPT and OpenAI Models, by Packt Publishing, 2023.
3. Jay Alammar, Maarten Grootendorst, Hands-On Large Language Models, by O'Reilly, 2023
4. Thushan Ganegedara, Natural Language Processing with TensorFlow, by Packt Publishing, Second Edition, 2022

Reference Books:

1. David Foster, Generative Deep Learning, O'Reilly, 2020
2. Lewis Tunstall, Leandro von Werra & Thomas Wolf," Natural Language Processing with Transformers", 2022
3. Sebastian Raschka, Build a Large Language Model (From Scratch), ISBN 9781633437166

Web Links:

1. Mixture of Experts: Mixture of Experts Explained (huggingface.co)
2. PEFT:Efficient Model Fine-Tuning for LLMs: Understanding PEFT by Implementation | by Shivansh Kaushik | Medium
3. Various benchmarks to evaluate LLMs: LLM Benchmarks: Understanding Language Model Performance (humanloop.com)
4. Types of attention mechanism: Understanding and Coding the Self-Attention Mechanism of Large Language Models From Scratch (sebastianraschka.com)
5. Agents| RAG: Intro to LLM Agents with Langchain: When RAG is Not Enough | by Alex Honchar | Mar, 2024 | Towards Data Science

Online Resources:

1. React| Agent: Teaching LLMs to Think and Act: ReAct Prompt Engineering | by Bryan McKenney | Medium
2. LLM based Agents : Superpower LLMs with Conversational Agents | Pinecone
3. RAGAS: Evaluating RAG pipelines with Ragas + LangSmith (langchain.dev)
4. Model distillation: LLM distillation demystified: a complete guide | Snorkel AI
5. Sentence classifier |BERT: Classify text with BERT | Text | TensorFlow

Prepared by | Checked by | Head of the Department | Vice Principal | Principal

Program: B. Tech in Artificial Intelligence (AI) and Data Science **T.Y. B. Tech** **Semester: VI**
Course: Advanced Java Laboratory (DJS23SLPC603)

Prerequisite: Core Java and OOP concepts

Course Objectives: The Objectives of course are

1. To familiarize students with advanced object-oriented concepts and design patterns in Java for creating scalable applications
2. To enable students to optimize data handling through the Java Collections Framework, generics, and the Streams API
3. To equip students with skills to design, build, and secure web applications using Spring and Spring Boot frameworks, with a focus on database connectivity

Course Outcomes: Students will be able to

1. Apply advanced object-oriented programming concepts, including design patterns, SOLID principles, and Java features such as collections, streams, and concurrency, to develop modular, efficient, and maintainable software solutions
2. Analyze and implement enterprise-level Java applications integrating JDBC, ORM (Hibernate/JPA), and Spring-based frameworks to build database-driven, component-oriented, and RESTful web services
3. Design and evaluate secure, scalable, and deployable microservice architectures using Spring Boot, Spring Security, and containerization tools (e.g., Docker), ensuring performance optimization and reliable system integration

Advanced Java Laboratory (DJS23SLPC603)		
Unit No	**Description**	**Duration**
1	**Design Patterns:** Introduction to design patterns and implementation: Singleton, Factory, Observer, Strategy	04
2	**SOLID Principles**: Understanding and applying SOLID principles for better design and maintainability	04
3	**Interfaces and Abstract Classes**: Java Cryptography Architecture (JCA); securing Spring Boot applications using Spring Security; authentication and authorization techniques .Advanced usage of interfaces, abstract classes, and default methods	04
4	**Collections & Generics:** Lists (ArrayList, LinkedList), Sets (HashSet, TreeSet), Maps (HashMap, LinkedHashMap), and use of Generics for type safety	04
5	**Java Streams:** Streams API concepts: creating streams from collections/arrays, and performing map, filter, reduce, and collect operations	04
6	**Concurrency & Advanced Concepts:** Multithreading (threads, synchronization, executors); Java Concurrency utilities (Locks, Semaphores, CompletableFuture); Virtual Threads (Project Loom); Performance tuning and profiling	04
7	**Java Reflection & Exception Handling:** Accessing and manipulating class properties at runtime, dynamic invocation, annotations (predefined/custom); advanced exception handling and logging using frameworks like SLF4J/Log4J	04
8	**Java Database Connectivity (JDBC):** Connecting to databases using JDBC, executing SQL queries, managing results, and handling transactions	04

9	**Object-Relational Mapping (ORM):** Overview of Hibernate and JPA; creating applications using Hibernate for seamless database interaction	04
10	**Introduction to Enterprise Framework (Spring):** Servlets overview; Spring Framework concepts: Inversion of Control (IoC), Dependency Injection (DI), and modular structure of Spring applications	04
11	**Spring Core:** Understanding Beans, Application Context, Bean Lifecycle; configuring Spring using XML and annotations	04
12	**Spring Boot & RESTful Services:** Advantages of Spring Boot over traditional Spring, project setup, building REST APIs, Spring Data JPA, and repository patterns	04
13	**Securing Applications:** Java Cryptography Architecture (JCA); securing Spring Boot applications using Spring Security; authentication and authorization techniques	04
14	**Microservices & Deployment Tools:** Fundamentals of Microservices, architecture & design principles, Spring Boot microservices, Eureka API Gateway, load balancing, and Introduction to Containerization (Docker) for deployment	04
	Total	**56**

The Term Work will be calculated based on Laboratory Performance, Mini Project and Quiz

Textbooks:

1. Herbert Schildt, Java: The Complete Reference, 13th Edition, McGraw Hill Education, 2023
2. Mark Heckler, Spring Boot: Up and Running, 1st Edition, O'Reilly Media, 2021
3. Craig Walls, Spring in Action, 6th Edition, Manning Publications, 2022
4. Bert Bates, Kathy Sierra and Elisabeth Freeman, Head First Design Patterns, 2nd Edition, O'Reilly Media, 2020

Reference Books:

1. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, 1st Edition, Addison-Wesley Professional, 1994
2. Brian Goetz and Tim Peierls, Java Concurrency in Practice, 1st Edition, Addison-Wesley Professional, 2006
3. Dinesh Rajput, Mastering Spring Boot 2.0, 2nd Edition, Packt Publishing, 2020
4. Greg L. Turnquist, Learning Spring Boot 3.0, 3rd Edition, Packt Publishing, 2022
5. Joshua Bloch, Effective Java, 3rd Edition, Addison-Wesley Professional, 2018

Online Resources:

1. Core Java & Collections
   Oracle Java Tutorials: https://docs.oracle.com/javase/tutorial/collections/
   GeeksforGeeks – Java Collections: https://www.geeksforgeeks.org/collections-in-java/
2. Streams API
   Java Streams Guide – Jenkov.com: https://jenkov.com/tutorials/java-collections/streams.html
3. Concurrency & Project Loom
   Oracle Docs – java.util.concurrent:
   https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/package-summary.html
   OpenJDK Project Loom Page: https://openjdk.org/projects/loom
4. Reflection & Annotations
   Baeldung – Java Reflection Tutorial: https://www.baeldung.com/java-reflection

5. JDBC & Hibernate
   Oracle JDBC Guide: https://docs.oracle.com/javase/tutorial/jdbc/
   Hibernate Official Documentation: https://hibernate.org/orm/documentation/
6. Spring & Spring Boot
   Spring Framework Docs: https://docs.spring.io/spring-framework/docs/current/reference/html/
   Spring Boot Reference Guide: https://docs.spring.io/spring-boot/docs/current/reference/html/
7. Microservices & Security
   Spring Cloud Docs: https://spring.io/projects/spring-cloud
   Spring Security Guide: https://spring.io/guides/topicals/spring-security-architecture/
8. General Learning Support
   NPTEL – Advanced Java Programming: https://onlinecourses.nptel.ac.in/noc20_cs58/preview
   W3Schools Java Tutorial: https://www.w3schools.com/java/

Prepared by | Checked by | Head of the Department | Vice Principal | Principal

Program: Artificial Intelligence (AI) and Data Science **T.Y. B.Tech** **Semester: VI**
Course: Advanced Data Structures (DJS23SCPE611)
Course: Advanced Data Structures Laboratory (DJS23SLPE611)

Prerequisite: Concepts of Data structures, Discrete mathematics, Analysis of Algorithm and Basics of Machine Learning

Course Objectives: The objective of the course is,

1. To provide conceptual and practical knowledge of Advanced Data Structures and Analysis of Algorithms.

Course Outcomes: Students will be able to

1. Understand the concept of time complexity and its importance in analyzing algorithms and to Explore the complexity analysis of popular machine learning algorithms.
2. Explore balanced search tree data structures and spatial data structures used in geometric and spatial applications.
3. Apply graph algorithms to solve real-world problems related to network flows, matching, and optimization.
4. Analyze amortized complexities and utilize advanced string data structures to design efficient string processing and pattern-matching solutions.
5. Apply advanced hashing methods for efficient data applications.
6. Apply advanced heap structures to develop efficient computational solutions.

Advanced Data Structures (DJS23SCPE611)		
Unit No	**Description**	**Duration**
1	**Classification of Algorithms:** Algorithm Classes: P, NP, NP Hardness and NP Completeness Complexity Analysis of Machine Learning Algorithms Training Time Complexity and Testing Time Complexity, Train/Test Complexity of Linear Regression, Train/Test Complexity of Naïve Bayes Classifier Other Patterns on arrays: Two pointers, Slow and Fast pointers, Sliding Window, Binary Search on Answer, Prefix Sum, Bit wise manipulation.	**05**
2	**Analysis of Data Structures:** Amortized Complexity- Aggregate Method, Accounting Method, Potential Method Data Structures for String: Tries and Compressed Tries, Suffix Tree and Suffix array, Bitwise Tries, Compressed Tries, Ternary Search Tree, String Searching with application.	**06**
3	**Balanced Search Tree:** Height Balance and Weight Balance Trees, Red-Black Tree, Splay Tree, AVL Tree, B Tree, B+ Tree ,Skip List, Randomized BST, Tango Tree with application.	**10**
4	**Heap and Operations**: Heap ordered Tree, Leftist Heap, Skew Heap, tournament Tree, Binomial Heap vs Binary heap, Pairing Heap, Double Ended Heap, Multidimensional Heaps, Van Emde Boas Priority Queues, Treap with application, Disjoint Set Union. Top K	**06**

	elements, Two heaps-MaxHeap and MinHeap	
5	**Spatial Data Structures:** Interval, Segment, Range, Priority Search Tree, KD Tree, Quad Tree, OCTree, R Tree with application. Graph based algorithms: Residual Network, Augmenting Path, Edmonds-Karp Implementation, Bipartite Matching (using BFS/DFS approach)	10
6	**Hash Tables:** Universal Families of Hash Functions, Perfect Hash Functions, Cuckoo Hashing, Probabilistic Data Structures: Bloom filters, Count-Min Sketch, HyperLogLog. Locality Sensitive Hashing, Hash Tree (Merkle Tree) with application.	05
	Total	42

List of Experiments:

1. Experiment on Amortized Analysis
2. Experiment on Bitwise Tries.
3. Experiment on Tries (Core Operations, Prefixes and Suffixes Based Problems.
4. Experiment on Balanced Search Trees (Red/Black Tree, B Tree, B+ Tree)
5. Experiment on Advanced Data Structure (AVL - Core Operations)
6. Experiment on Heaps and Priority Queues
7. Experiment on Graph-Based Algorithms (Ford-Fulkerson Method)
8. Experiment on Graph-Based Algorithms (Edmonds-Karp Method)
9. Experiment on Spatial data structure.
10. Experiment on Hash Tables.
11. Development of New Algorithm by Students Based on Any One Topic of Above Mentioned Syllabus

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.

Textbooks:

1. Peter Brass, Advanced Data Structures, 1st Edition, Cambridge University Press, 2008.
2. Suman Saha, Shailendra Shukla, Advanced Data Structures Theory and Applications, 1st Edition, CRC Press and Taylor & Francis, 2019.
3. Giuseppe Bonaccorso, Machine Learning Algorithms by Packt, 2019
4. Harsh Bhasin, Algorithms Design and Analysis, Oxford, 1st Edition, 2015.
5. Introduction to Algorithms by Thomas H Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein, Third Edition, 2009.

Reference Books:

1. Dinesh Mehta and Sartaj Sahni, Handbook of Data Structures and Applications, 1st Edition, Chapman & Hall/CRC, 2005.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Introduction to Algorithms, 3rd Edition, The MIT Press, 2009.
3. Daniel R. Page, Advanced Data Structures: An Introduction to Data Structures and Algorithms, Kindle Edition, 2020.

4. Vijay V. Vajirani, Approximation Algorithms, Springer, 2003

Web Links:

1. Train/Test Complexity and Space Complexity of Linear Regression :Train/Test Complexity and Space Complexity of Linear Regression | by Writuparna Banerjee | Level Up Coding (gitconnected.com)
2. Computational Complexity of ML Models :Computational Complexity of ML Models | by Paritosh Kumar | Analytics Vidhya | Medium
3. The Power of Bloom Filters:The Power of Bloom Filters: A Comprehensive Guide | by Chiranjeet Baruah | Medium

Online Resources:

1. Introduction to Min Hash: ML Security Pro Tips: Understanding MinHash in a Security Context | by Melanie Beck | AI/ML at Symantec | Medium
2. Probabilistic Data Structures Decoded: Enhancing Performance in Modern Computing | by Naman Agrawal | Towards Data Science

Prepared by	Checked by	Head of the Department	Vice Principal	Principal

Program: B. Tech in Artificial Intelligence (AI) and Data Science **T.Y. B. Tech Semester: VI**

Course: Financial Time Series Analysis (DJS23SCPE612)

Course: Financial Time Series Analysis Laboratory (DJS23SLPE612)

Prerequisite: Statistics for Data Science, Machine Learning, Deep Learning

Course Objectives: The Objectives of course are

1. Learn basic analysis of time series data and preprocessing.
2. Study classical time series models
3. Learn basic nonlinear and multivariate time series
4. Explore advanced machine learning and deep learning approaches for improving forecasting accuracy

Course Outcomes: Students will be able to

1. Explain the fundamental structure, properties, and statistical characteristics of financial time-series data.
2. Preprocess and transform financial data using smoothing, normalization, and resampling for analysis and model building
3. Develop and evaluate univariate, nonlinear, and multivariate models for forecasting and volatility analysis
4. Design and compare an advanced ML, DL, and hybrid forecasting models for financial prediction and risk management

Financial Time Series Analysis (DJS23SCPE612)		
Unit No	**Description**	**Duration**
1	**Introduction & Time Series Fundamentals:** Financial data structure: prices, returns, volume, volatility, Asset classes and types of financial time series, return calculations: simple, log, and cumulative returns, Data frequency: tick, daily, monthly; irregular and missing data handling, Characteristics of financial data: volatility clustering, heavy tails, leverage effect, non-stationarity. Time Series Fundamentals Components: trend, seasonality, cyclical, and irregular components, Types of time series (discrete, continuous, univariate, multivariate), Models of time series analysis (deterministic vs stochastic models), Types of time series patterns and different types of data, Simple descriptive techniques and trend identification and measurement, Time Series Decomposition: Additive and Multiplicative Decomposition Models	09
2	**Preprocessing, Smoothing & Data Preparation:** Preprocessing: Handling missing values and outliers (Z-score, Hampel filter, interpolation), Date-time handling, resampling, and alignment of multi-asset data Smoothing Techniques: Moving average, exponential smoothing (SES, Holt's, Holt–Winters), Rolling mean/variance and trend extraction. Normalization and scaling techniques (Min–Max, Z-score) for ML readiness.	05
3	**Stationarity, Correlation and Model Identification:** Concepts of stationarity ,weak and strict stationarity, White noise, random walk, and martingale difference sequences in finance, Unit root testing (ADF, KPSS) and differencing for stationarity, Transformations for variance stabilization (log, Box–Cox), Correlogram analysis: ACF and PACF — plotting and interpretation, Identifying AR and MA Orders, Model order selection using AIC/BIC, Volatility visualization and autocorrelation of squared returns.	06
4	**Univariate Models and One-Step-Ahead Forecasting:** Linear stochastic models: AR, MA, theoretical properties of time series with a MA (1)	08

	and MA (2) model, simulating from an autoregressive process, AR (∞) representation of MA(P) model and MA(∞) representation of AR(P) model, ARMA, ARIMA, Seasonal ARIMA (SARIMA) and Box–Jenkins methodology, Backshift and lag operator representation. One-step-ahead prediction: Concept and derivation for AR(p), MA(q), ARMA(p,q), Durbin–Levinson algorithm, Model diagnostics and residual analysis (Ljung–Box test), Rolling and walk-forward forecasting evaluation.	
5	**Non Linear Time series:** The ARCH model: Feature of an ARCH, interpretation of ARCH model, The GARCH model: Existence of stationary solution of a GARCH(1,1), Value-at-Risk (VaR) estimation using volatility forecasts, Monte Carlo Simulation: Random walk simulation for asset price forecasting and risk estimation (VaR and Expected Shortfall). Multivariate Time Series Models Motivation for multivariate analysis: interconnected markets, Conditional independence,Partial correlation and coherency between time series, Vector AR (VAR) Models, Granger causality, co-integration.	07
6	**Machine Learning, Deep Learning & Hybrid Forecasting** Forecasting evaluation metrics: RMSE, MAPE, Theil's U, Diebold–Mariano test. Feature engineering: windowing and lag creation. Machine Learning models: Linear Regression, Deep Learning models: CNN, RNN, LSTM, GRU, Attention, and Transformers for time-series data. Hybrid Models: Combining ARIMA and LSTM for improved forecasting accuracy.	07
	Total	**42**

List of Experiments:

1. Preprocess and smooth financial data; handle missing values and outliers
2. Smoothing techniques
3. Autoregression Models for Time Series Forecasting with Python
4. Apply ARIMA - SARIMA models to predict / forecast the number of
5. Implement Durbin levinson algorithm for one step ahead forecasting.
6. Implement a multivariate time series regression model.
7. Implement ARCH and GARCH on a financial time series dataset.
8. Develop VAR and cointegration models for multi-asset relationships.
9. Build an LSTM or ARIMA–LSTM hybrid forecasting model.
10. Mini Project

Minimum eight 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.

Textbooks:

1. Ruey S. Tsay, Analysis of Financial Time Series, Third edition, Wiley,2010
2. D. Garfield, Time Series Analysis for Beginners: Comprehensive Introduction, May Reads, 2024
3. W. W. S. Wei, Time Series Analysis: Univariate and Multivariate Methods, 2nd ed., Pearson, 2023
4. D. C. Montgomery, C. L. Jennings, and M. Kulahci, Introduction to Time Series Analysis and Forecasting, 2nd ed., Wiley, 2015
5. W. Palma, Time Series Analysis, Wiley, 2016

6. B. V. Vishwas and A. Patel, Hands-on Time Series Analysis with Python, 1st ed., Apress, 2020
7. J. D. Hamilton, Time Series Analysis, Princeton University Press, 1994
8. T. Dunning and E. Friedman, Time Series Databases: New Ways to Store and Access Data, 1st ed., O'Reilly Media, 2019

Reference Books:

1. T. A. Atwan, Time Series Analysis with Python Cookbook: Practical Recipes for Exploratory Data Analysis, Data Preparation, Forecasting, and Model Evaluation, Packt Publishing, 2022
2. M. S. Paolella, Linear Models and Time- Series Analysis: Regression, ANOVA, ARMA and GARCH, Wiley, 2018
3. W. Enders, Applied Econometric Time Series, 4th ed., Wiley, 2014
4. C. Chatfield, Time-Series Forecasting, 1st ed., Chapman & Hall/CRC, 2001
5. A. Nielsen, Practical Time Series Analysis, O'Reilly Media, 2019
6. R. H. Shumway and D. S. Stoffer, Time Series Analysis and Its Applications: With R Examples, 4th ed., Springer, 2017

Web Links:

1. Practical Time Series Analysis: https://www.coursera.org/learn/practical-time-series-analysis?msockid=0fba0991cc8d64af281a1ae9cd3f657a
2. Introduction to Time Series: https://www.coursera.org/learn/illinois-tech-introduction-to-time-series?irclickid=xGjxa83B5xyPUjXVPt3FCX4hUksy38R1qQ47R80&irgwc=1&utm_medium=partners&utm_source=impact&utm_campaign=315774&utm_content=b2c&utm_campaignid=CoToNet%20-%20Tecnologias%20de%20Informa%C3%A7%C3%A3o%2C%20Unipessoal%20Lda&utm_term=14726_CR_1213620_
3. Demand Prediction Using Time Series: https://www.coursera.org/learn/demand-prediction-using-time-series?msockid=0fba0991cc8d64af281a1ae9cd3f657a
4. A PDF lecture/teaching material: https://web.stat.tamu.edu/~suhasini/teaching673/time_series.pdf

Online Resources:

1. Quantra – Financial Time Series Analysis Course | Stock Market Prediction Using Time Series Analysis
2. Applied Time-Series Analysis By Prof. Arun K. Tangirala | IIT Madras https://onlinecourses.nptel.ac.in/noc21_ch28/preview
3. Time Series Modelling and Forecasting with Applications in R By Prof. Sudeep Bapat | IIT Bombay: https://onlinecourses.nptel.ac.in/noc25_cs71/preview

Prepared by | Checked by | Head of the Department | Vice Principal | Principal

Program: B. Tech in Artificial Intelligence (AI) and Data Science **T.Y B. Tech** **Semester: VI**
Course: Probabilistic Graph Models (DJS23SCPE613)
Course: Probabilistic Graph Models Laboratory (DJS23SLPE613)

Prerequisite: Machine learning, Probability

Course Objectives: The Objectives of course are

1. Provide a comprehensive understanding of the fundamental concepts and principles of probabilistic and deterministic graph theory
2. Develop the ability to apply various graph theoretical algorithms and techniques for problem solving and analysis
3. Enable students to integrate theoretical knowledge of graph theory to model, analyze, and solve complex real-world problems in computing and related domains

Course Outcomes: Students will be able to

1. Explain the basic fundamentals of probabilistic graph theory
2. Illustrate various principles of graph theory and algorithms
3. Integrate core theoretical knowledge of graph theory to solve problems

Probabilistic Graph Models (DJS23SCPE613)		
Unit No	**Description**	**Duration**
1	**Fundamentals:** Structured Probabilistic Models: probabilistic graphical model, Representation, Inference, Learning and, Marginal and Joint Distributions, Independence and Conditional Independence. Bayesian Networks: Independence and Separation, from distribution to graphs, Markov properties and minimalism, Examples (Markov chain: HMM, diagnostic system, etc.)	**07**
2	**Markov Networks:** Boltzmann machine and Ising models, Markov random field, Parameterization, Cliques and potentials, Markov Network Independencies, Factor graphs. Gaussian Network Models and Exponential Family: Multivariate Gaussians, Gaussian Bayesian network, Gaussian random feilds, Exponential families, Entropy and Relative Entropy, Projections.	**07**
3	**Inference: Exact inference: Variable elimination:** Analysis of Complexity, Variable elimination, Belief propagation (message passing) on trees, Sum- and Max-product algorithms, Clique tree, Variable Elimination in a Clique Tree	**07**
4	**Sampling Methods:** MCMC method, Gibbs sampling Algorithm, Importance sampling, Particle filtering. Approximate inference: Loopy belief propagation, Variational inference and optimization view of inference, Mean field approach	**07**
5	**Parameter learning:** Parameterizing graphical models, Parameter estimation in fully observed Bayesian networks: (Maximum likelihood estimation, Bayesian parameter estimation, Example: HMM), Parameter estimation in fully observed Markov networks: Maximum likelihood estimation (Iterative Proportional Fitting (IPF), Generalized Iterative Scaling (GIS)), Parameter estimation in partially observed graphical models (Expectation-Maximization	**06**

	(EM) - Example: HMM), Learning Conditional Random Fields. Nonparametric Learning: Gaussian processes, Dirichlet processes, Indian Buffet processes	
6	**Structure learning:** Score based approach, Chow-Liu algorithm for Bayesian networks, l1-regularized convex optimization for Markov random fields, Low-rank regularized learning of latent variable models.	08
	Total	42

List of Experiments:

1. Implement Discrete Bayesian Networks
2. Implementation of Alarm Bayesian Network
3. Implementation of Linear Gaussian Bayesian Networks (GBNs)
4. Implementation of Monty Hall Problem using Bayesian Network
5. Implementation of Exact inference in Bayesian Networks
6. Implementation of Inference in Discrete Bayesian Network
7. Implementation of Causal Inference
8. Implementation of Parameter Learning in Discrete Bayesian Network
9. Implementation of Learning using Chow-Liu Algorith
10. Implementation of learning Tree-augmented Naive Bayes (TAN)

Minimum eight 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

Textbooks:

1. Koller, D. and Friedman, N. Probabilistic Graphical Models: Principles and Techniques. MIT Press 2009

Reference Books:

1. Jensen, F. V. and Nielsen, T. D. Bayesian Networks and Decision Graphs. Information Science and Statistics. Springer, 2nd edition , 2002.
2. Marloes Maathuis, Mathias Drton, Steffen Lauritzen, Martin Wainwright, Handbook of Graphical Models, Routledge Taylor and Francis group, 2020
3. Kevin P. Murphy Machine Learning: A Probabilistic Perspective. 4th Printing. MIT Press 2013
4. Barber, D. Bayesian Reasoning and Machine Learning. Cambridge University Press, 1st edition 2011
5. Bishop, C. M. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, 2nd printing 2011
6. Wainwright, M. and Jordan, M. (2008). Graphical Models, Exponential Families, and Variational Inference. Foundations and Trends in Machine Learning, 1:1–305

Web Links:

1. Comprehensive tutorial on Bayesian Networks and learning.
   https://www.ee.columbia.edu/~vittorio/Lecture12.pdf?
2. Online course specialization from Stanford University: Probabilistic Graphical Models Specialization.
   https://www.coursera.org/specializations/probabilistic-graphical-models?utm_source=chatgpt.com

Online Resources:

1. Introductory article – "A Brief Introduction to Graphical Models and Bayesian Networks: https://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html?utm_source=chatgpt.com
2. Practical tutorial using Python and pgmpy: https://pgmpy.org/detailed_notebooks/1.%20Introduction%20to%20Probabilistic%20Graphical%20Models.html?utm_source=chatgpt.com

Prepared by | Checked by | Head of the Department | Vice Principal | Principal

Program: B. Tech in Artificial Intelligence (AI) and Data Science **T.Y. B. Tech** **Semester: VI**
Course: Big Data Analytics (DJS23SCPE614)
Course: Big Data Analytics Laboratory (DJS23SCPE614)

Prerequisite: Databases, Python, R, Linux OS

Course Objectives: The Objectives of course are

1. To introduce the fundamental concepts of Big Data, its characteristics, lifecycle, and significance in business analytics and decision-making
2. To develop an understanding of the Hadoop framework and its ecosystem components for distributed storage and large-scale data processing
3. To enable students to apply Big Data tools and techniques such as Spark, Hive, Pig, and NoSQL databases for data analytics, machine learning, and real-time data processing applications

Course Outcomes: Students will be able to

1. Understand Big Data fundamentals, lifecycle, and challenges
2. Understand Big Data fundamentals, lifecycle, and challenges
3. Implement distributed storage and processing using Hadoop and its ecosystem tools
4. Work with NoSQL databases and manage unstructured data
5. Apply Apache Spark and MLlib for scalable data analytics
6. Analyze streaming and textual data using modern real-time processing frameworks

Big Data Analytics (DJS23SCPE614)		
Unit No	**Description**	**Duration**
1	**Introduction to Big Data and Hadoop Ecosystem:** Introduction to Big Data: Definition , Characteristics(Five Vs Volume, Velocity, Variety, Veracity, Value), Drivers and Types of Big Data, Applications and Case Studies (e.g., Healthcare, Retail, IoT, Social Media, Finance), Big Data Analytics Lifecycle and its Business Importance, Societal and Ethical Issues in Big Data (privacy, bias, accountability), Overview of Apache Hadoop, Hadoop Architecture and Components: : HDFS, YARN, and MapReduce.	08
2	**Hadoop Distributed Processing and Ecosystem Tools:** Hadoop Distributed File System (HDFS): Overview, Installation, Shell Commands, Java API. MapReduce Programming Model: Map, Shuffle, Reduce Phases, Grouping by Key, Combiners, and Execution Flow. Data Serialization and File Formats in Hadoop. Hadoop Ecosystem Overview: Apache Hive: Architecture, Metastore, Schema, HiveQL (Filtering, Sorting, Aggregation), Comparison with RDBMS,Apache Pig: Data Model, Grunt Shell, Pig Latin Scripts, Grouping and Joining Operations, UDFs, Apache Zookeeper: Coordination Service, Role in Cluster Management, Integration with HBase	11
3	**NoSQL Databases and HBase:** Introduction to NoSQL Databases, Relational vs NoSQL Data Stores, CAP Theorem, ACID vs BASE, Types of NoSQL Databases: Key-Value (Redis), Document-based (MongoDB), Column-based (HBase, Cassandra), Graph-based (Neo4j). HBase: Architecture and Internals, Data Model: Tables, Column Families, Regions, Schema Design and Advanced Indexing,	09

	Integration with Zookeeper. Cassandra Overview and Data Model (Keyspace, Tables, Columns)	
4	**Apache Spark and Machine Learning for Big Data:** Introduction to Apache Spark, Spark Architecture and In-Memory Computation, Working with RDDs, DataFrames, and Datasets, Spark SQL for Structured Data Processing. Introduction to Spark MLlib: Supervised and Unsupervised Learning, Parallel K-Means Algorithm. Introduction to Apache Mahout — Scalable Machine Learning	10
5	**Real-Time Data Processing:** Streaming Data Analytics ;Real-Time vs Batch Processing, Data Ingestion with Apache Kafka, Stream Processing with Apache Storm, Spark Streaming;DStreams and Micro-batching	04
	Total	42

List of Experiments:

1. Installation of Hadoop on a single node cluster
2. Execution of HDFS Commands
3. Execution of MapReduce Program (Sorting, Word Count)
4. Hive Operations (Create, Insert, Query)
5. Pig Commands and Script Execution
6. Installation and Configuration of Apache Spark, Execution of MLlib Algorithm
7. CRUD Operations in MongoDB (Document-based Data Store)
8. Analyze non-relational data using HBase
9. Analyze non-relational data using Cassandra
10. Real-Time Data Ingestion using Apache Kafka
11. Mini Project

Minimum eight 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

Textbooks:

1. Chris Eaton,Dirk derooset , Understanding Big data McGraw Hill,2012.
2. Kyle Banker,Piter Bakkum, Shaun Verch, MongoDB in Action ,Dream tech Press,2016
3. Balaswamy Vaddeman, Beginning Apache Pig-Big Data Processing Made Easy Apress',2016
4. Tom White, Hadoop: The Definitive Guide, Third Edition, O'Reilley,2012.
5. Eric Sammer, Hadoop Operations, Reilly,2012

Reference Books:

1. Paul Zikopoulos, Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Understanding Big Data: Analytics for Enterprise Class Hadoop and streaming Data,The McGraw-Hill Companies, 2012Bass, Len, Paul Clements, and Rick Kazman, Software Architecture in Practice
2. Vignesh Prajapati, Big data analytics with R and Hadoop, SPD 2013
3. E. Capriolo, D. Wampler, and J. Rutherglen, "Programming Hive", O'Reilley, 2012
4. Alan Gates, "Programming Pig", O'Reilley, 2011

Web Links:

1. Tableau Tutorial for Beginners in 20 Minutes | Complete Tableau Training for Beginners | Simplilearn - YouTube
2. Power BI Full Course | Power BI Tutorial For Beginners | Power BI Course | Simplilearn - YouTube
3. Power BI Tutorial (tutorialspoint.com)
4. Tableau Tutorial (tutorialspoint.com)

Online Resources:

1. Big Data Computing: https://onlinecourses.nptel.ac.in/noc20_cs92/preview
2. Google Data Analytics Professional Certificate, https://www.coursera.org/professional-certificates/google-data-analytics
3. NoSQL, Big Data, and Spark Foundations Specialization, https://www.coursera.org/specializations/nosql-big-data-and-spark-foundations

Prepared by Checked by Head of the Department Vice Principal Principal

Program: B. Tech in Artificial Intelligence (AI) and Data Science **T.Y. B. Tech** **Semester: VI**
Course: System Design (DJS23SCPE615)
Course: System Design Laboratory (DJS23SCPE615)

Prerequisite: Concepts of Data structures, Basic SQL, HTTP fundamentals, Python programming with functions and classes, Command-line skills

Course Objectives: The Objectives of course are

1. Introduce fundamental system design concepts through hands-on implementation
2. Build practical skills in designing and implementing scalable software systems
3. Bridge the gap between theoretical concepts and real-world application.
4. Develop problem-solving abilities for common system design challenges
5. Foster understanding of trade-offs in architectural decisions through experimentation

Course Outcomes: Students will be able to

1. Explain and apply core system design concepts including scalability, load balancing, caching, and database partitioning
2. Design and implement RESTful APIs with authentication and multiple endpoints
3. Compare and contrast different architectural patterns and database technologies
4. Implement basic reliability mechanisms including failover, logging, and monitoring
5. Build end-to-end systems for real-world problems like URL shorteners and chat applications
6. Create architecture diagrams and technical documentation for system design

System Design (DJS23SCPE615)		
Unit No	**Description**	**Duration**
1	**Introduction and Core Concepts:** Overview of System Design, Simple definition, uses in software engineering, High-level vs Low-level Design: Easy examples (Instagram feed flow vs function code), Scalability Basics: Vertical vs Horizontal scaling, pros/cons with simple analogiesCaching: simple types (browser cache, server cache), basic strategies (write-through/write-back simplified), Reliability, Latency, Performance vs Efficiency, CAP Theorem, ACID vs BASE, Data Partitioning and Replication Basics, CDN (Content Delivery Network), Message Queue (Kafka, RabbitMQ, SQS), Rate Limiting and Throttling, Transitioning from Open-Source to Commercial Use, Understanding Open-Source Licenses, Commercial Licensing Models	09
2	**Databases and Data Management**: JSON documents, flexible schema. Sharding Basics: split data across multiple databases; Horizontal vs vertical partitioning with visual demos; Partitioning strategies: Range-based, hash-based, geography-based; Sharding challenges: Cross-shard queries, hotspots. Replication & Backup Basics: Backup, read scaling, disaster recovery; Simple backup strategies: Full backups, incremental backups; Replication lag and consistency challenges. Consistency: All nodes see the same data; Availability: System responds to all requests; Partition Tolerance: System works despite network failures; Introduction to CAP: DB Indexing and Query Optimization, Database Sharding and Partitioning, Normalization vs	09

	Denormalization, Read Replica and Write Master Design, Transaction Management and Isolation Levels, Data Modeling Techniques	
3	**System Design Trade-offs:** Use simple analogies (single chef vs specialized chefs), REST API Basics: REST, resources, HTTP methods (GET/POST), status codes, API Gateways: Concept only (routing & security role) — no heavy tools introduced	07
4	**Reliability and Fault Tolerance:** Synchronous vs Asynchronous replication (conceptual only), Failover: active-active vs active-passive (visually explained), Disaster recovery basics (RTO/RPO with simple examples), Monitoring Basics: latency, errors, CPU	07
5	**Architectural Patterns :**Classic High-Level System Design Problems URL Shortener: Define requirements, sketch schema design, endpoint architecture (/shorten and /go/), storage strategies, caching layer implementation, Simple Chat/Messaging System: Client-server architecture using sockets, message storage strategies, multi-client communication patterns, Parking Lot Management: System requirements, slot availability management, basic concurrency handling concepts (locks and threading), Layered (N-tier) Architecture, Microservices Architecture, Monolithic Architecture (and evolution to microservices), Serverless Architecture, Peer-to-Peer Architecture, Service-Oriented Architecture (SOA)	10
	Total	42

List of Experiments:

1. Build a simple Python API (/hello) using Flask or FastAPI
2. Write a Python script to simulate 50 requests and measure response time
3. Implement manual scaling by running API on two ports with Python round-robin logic to forward requests
4. Add a Python dictionary as a cache and compare response time before vs after caching
5. Create a SQLite database to store simple user and task data, query it with Python and display results
6. Implement manual sharding: store odd IDs in one file, even IDs in another, fetch based on ID
7. Take backup (copy file), delete data, restore backup and observe data recovery in action
8. Extend API to have multiple routes: /add-task, /get-task, /delete-task and test using Postman or browser
9. Separate routes into different Python files to simulate microservices
10. Implement simple authentication (API key in header) manually
11. Simulate failover: Stop main API and start backup API manually
12. Add logging to API (count number of requests, track errors)
13. Write a simple Python script to alert (print message) when error rate > 20%
14. Design and implement URL Shortener with /shorten and /go/ endpoints, SQLite storage, and caching layer
15. Create console-based chat using Python socket library with message storage and two-client exchange
16. Build CLI program for Parking Lot Management with slot booking/unbooking and concurrency handling

Minimum eight 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

Textbooks:

1. Kendall, K. E., Systems Analysis and Design , Pearson Prentice Hall, 2011

2. Whitten, J. L., Systems Analysis and Design Methods , Irwin / McGraw-Hill 2008
3. Object Oriented Systems Analysis and Design Using UML, McGraw-Hill Higher Education,2011
4. Operating Systems: Internals and Design Principles, Pearson,2014
5. Distributed Operating Systems: Concepts and Design, PHI Learning Pvt. Ltd 1998

Reference Books:

1. Dennis, Wixom, and Roth, System Analysis and Design , John Wiley & Sons, 2018
2. Bass, Len, Paul Clements, and Rick Kazman, Software Architecture in Practice, Addison-Wesley Professional,2021
3. Xu, Alex, System Design Interview – An Insider's Guide , Shroff/Byte Code LLC, 2023

Online Resources:

1. Principles of Engineering System Design: https://nptel.ac.in/courses/107106009
2. NOC:Embedded System Design with ARM: https://nptel.ac.in/courses/106105193
3. CS50: Introduction to Computer Science: https://www.edx.org/cs50
4. Algorithms Specialization: https://www.coursera.org/specializations/algorithms

Prepared by Checked by Head of the Department Vice Principal Principal

Program: Artificial Intelligence (AI) and Data Science **T.Y. B.Tech.** **Semester: VI**
Course: Blockchain Technology (DJS23SCPE616)
Course: Blockchain Technology Laboratory (DJS23SLPE616))

Prerequisite: Networking Fundamentals, Distributed Operating Systems

Course Objectives: The objective of the course is,

1. To understand emerging Blockchain Technology and its relevance to cryptography
2. To demonstrate the use of cryptography required for Blockchain.
3. To understand smart contracts, wallets, and consensus protocols.
4. To design and develop Blockchain applications.

Course Outcomes: Students will be able to

1. Describe basic knowledge of Blockchain technology
2. Explain methods for securing blockchain networks, including cryptography and consensus protocols.
3. Use various tools for Blockchain implementation.
4. Analyze the real-world applications of Blockchain technology.

Blockchain Technology (DJS23SCPE616)		
Unit No	**Description**	**Duration**
1	**Introduction to Blockchain Technology:** The Model of Decentralization, Distributed Systems for Decentralization, Blockchain framework, Characteristics of Blockchain, Block structure, Block header, Types of Blockchain: Public, Private and Hybrid Blockchain.	07
2	**Basic Crypto primitives:** Cryptographic Primitives, Cryptographic Hash, Hash Functions, SHA-256, Puzzle Friendly, Secure Hash Algorithm, Hash Pointers, Merkle Tree, Hash Chain, Construction of Chain of Blocks, Public Key Cryptography, Digital Signature.	07
3	**Bitcoin and Consensus:** The Evolution of Cryptocurrencies: Design Goals for Cryptocurrency Development **Introduction to Bitcoin:** Bitcoin block, bitcoin P2P network, Transactions, Bitcoin mining, double spending attack, Forks, The Monopoly Problem-51% attack **Consensus:** Consensus Approach, Consensus Algorithms: Proof-of-Stake (PoS), Proof-of-Work (PoW), Proof-of-Burn (PoB), Proof-of-Elapsed Time (PoET), State Machine Replication as a Consensus, Crash Fault Tolerance, PAXOS, Byzantine Fault Tolerant (BFT), BFT Consensus, Practical BFT.	10
4	**Ethereum:** Ethereum and its Components, Ethereum Virtual Machine (EVM), Ethereum Ecosystem, Transaction, Comparison between Bitcoin and Ethereum, test-networks, Smart Contracts, Introduction to solidity programming, Ganache,	07

	MetaMask.	
5	**Hyperledger:** Introduction to Hyperledger Fabric, Key features of Hyperledger fabric, Hyperledger Fabric Architecture, Ethereum v/s Hyperledger framework, Fabric Test Network, Hyperledger Consensus, Fabric Transaction Flow, Hyperledger Tools and Libraries, Hyperledger Fabric Chaincode	07
6	**Blockchain Allied Technologies:** Blockchain in DeFi: Case Study on any of the Blockchain Platforms, Blockchain in Healthcare, Blockchain and Artificial Intelligence, Blockchain for Trustworthy AI in Financial Services, Blockchain and Machine Learning, Blockchain and IoT.	04
	Total	**42**

List of Experiments:

1. To create basic Blockchain with sample transactions and print it
2. To implement Merkle root from the transactions and verify the validity of transactions using it.
3. To implement Proof of Work (PoW) concept in Bitcoin Mining and demonstrating it.
4. To analyse and implement Unspent Transaction Outputs (UTXOs) in Bitcoin and demonstrate the transactions using UTXOs.
5. To create and deploy Smart Contract using Solidity and Remix IDE.
6. To perform Embedding Wallet and transaction using Solidity and MetaMask.
7. To implement blockchain using Geth (Go-Ethereum).
8. To implement local Blockchain using tools viz. Ganache and Truffle.
9. To interacting with the Ethereum Blockchain Using Web3.js
10. To install Hyperledger Fabric and demonstrate its usability.
11. To query and invoke transactions on Fabric Test Network

Minimum eight 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

Textbooks:

1. Imran Bashir , Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, 3rd Edition, Packt Publishing, 2020
2. Kumar Saurabh , Ashutosh Saxena, Blockchain Technology: Concepts and Applications , 1st Edition, Wiley Publication, 2020
3. S. Shukla, M. Dhawan, S. Sharma, S. Venkatesan, -Blockchain Technology: Cryptocurrency and Applications, Oxford University Press, 2019
4. Cryptography and Network Security – Principles and Practice by William Stallings, Pearson 2017

Reference Books:

1. Antony Lewis, Basics of Bitcoins and Blockchain, Mango Publishing, 2021

2. Blockchain for Beginners, Yathish R and Tejaswini N, SPD,22 September 2019
3. Blockchain Basics, A non-Technical Introduction in 25 Steps, Daniel Drescher, Apress,1 January 2017
4. Blockchain with Hyperledger Fabric, Luc Desrosiers, Nitin Gaur, Salman A. Baset, Venkatraman Ramakrishna, Packt Publishing,18 November 2020
5. Beginning Blockchain: A Beginner's Guide to Building Blockchain Solutions, Bikramaditya Singhal, Gautam Dhameja, Priyansu Sekhar Panda, Apress,2018

Web Links :

1. Hyperledger: https://www.hyperledger.org/use/tutorials
2. Ethereum: https://ethereum.org/en/developers/
3. TutorialsPoint: https://www.tutorialspoint.com/solidity/index.htm
4. MetaMask: https://docs.metamask.io/guide/

Online Resources :

1. Blockchain and its Applications, By Prof. Sandip Chakraborty, Prof. Shamik Sural IIT Kharagpur https://onlinecourses.nptel.ac.in/noc23_cs47/preview
2. Blockchain Architecture Design and Use Cases, By Prof. Sandip Chakraborty & Dr. Praveen Jayachandran | IIT Kharagpur and IBM, https://onlinecourses.nptel.ac.in/noc19_cs63/preview
3. Blockchain, By Dr.Mayank Aggarwal ,Gurukul Kangri Vishwavidyalaya,Haridwar https://onlinecourses.swayam2.ac.in/aic21_ge01/preview

Prepared by | Checked by | Head of the Department | Vice Principal | Principal

Program: B.Tech. in Artificial Intelligence (AI) and Data Science T.Y. B. Tech Semester: VI
Course: MLOps (DJS23SCMD601)
Course: MLOps Laboratory (DJS23SLMD601)

Prerequisite:

1. Knowledge of Linux Operating system, installation and configuration of services and command line basics
2. Basics of Machine Learning
3. Knowledge Development Life cycle, development frameworks and DevOps

Course Objectives: The Objective of course is

1. The objective of this course is to understand the fundamentals of MLOps and its significance in the ML lifecycle.
2. Students will learn various tools and technologies used in MLOps to design and build scalable ML pipelines.
3. Students will get exposure to deploy ML models.
4. Students will learn techniques for monitoring, debugging, and optimizing ML systems.
5. Finally, students will explore methods for reproducibility, version control, and model governance.

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

1. To understand the fundamentals of software engineering, Agile methodologies, and DevOps practices for efficient and continuous software development.
2. Understand the Fundamentals of MLOps and Its Role in ML Lifecycle Management.
3. Develop Skills in Data and Model Management for Reproducibility and Scalability.
4. Build and Automate ML Model Training Pipelines.
5. Implement Model Deployment and Serving Strategies Using CI/CD.
6. Monitor and Optimize ML Model Performance and Infrastructure.

MLOps (DJS23SCMD601)		
Unit No.	**Description**	**Duration**
1	**Fundamentals of Software Engineering, Agile, and DevOps** Introduction to Software Engineering – Process framework, Software Development Life Cycle (SDLC) and Process Models: Incremental and Evolutionary. Introduction to DevOps – Definition, Objectives, Continuous Integration & Deployment, Containers and Virtual Development Environments, Configuration Management Tools. Overview of Agile Process – Need for Agile Software Development, Agile Manifesto and Principles, Overview of Agile Models: Scrum, Kanban, and Lean Software Development, Business Benefits of Software Agility.	07
2	**Introduction to Machine Learning Operations** Overview of MLOps and its importance, Understanding the challenges in deploying and managing ML models, ML development lifecycle, Role of MLOps in the ML development lifecycle, Introduction to DevOps and its application to ML, MLOps in Practice.	07

3	**Data Management, Model Development and Training for MLOps** Model Development and Training for MLOps, Data versioning and reproducibility, Data preprocessing and feature engineering pipelines, Data validation and monitoring, Data quality assurance and governance, Model versioning and tracking, Model training pipelines and automation, Hyperparameter tuning and model selection, Model evaluation and validation techniques	08
4	**Model Deployment and Serving, Continuous Integration and Delivery (CI/CD) for ML** Model packaging and containerization (e.g., Docker), Infrastructure provisioning and orchestration (e.g., Kubernetes), Deploying models as scalable services, managing model endpoints and versioning, Version control and collaboration (e.g., Git), Building reproducible ML pipelines, Automated testing and code quality checks, Continuous integration and deployment strategies.	06
5	**Monitoring and Performance Optimization** Monitoring model performance and behavior, Real-time and batch monitoring techniques, Logging and error tracking in ML systems, Performance optimization and scalability considerations.	06
6	**Cloud Platforms and Infrastructure for MLOps** Introduction to cloud platforms (e.g., AWS, Azure, GCP) key services of each platform, Deploying ML models on cloud infrastructure, managing resources and scaling ML workloads, Cost optimization strategies for ML systems.	08
	TOTAL	42

List of experiments:

1. Write code for a simple user registration form for an event. To Study DevOps: Principles, Practices, and DevOps Engineer Role and Responsibilities
2. To configure Scrum and Kanban boards in JIRA, create sprints, and visualize issue progress.
3. Setting up a Version Control System (VCS) for ML Projects:
    a. Experiment with popular VCS tools like Git and create a repository for ML projects.
    b. Learn to track code changes, collaborate with team members, and manage different branches.
4. Creating a Continuous Integration (CI) Pipeline:
    a. Build a CI pipeline using tools like Jenkins, Travis CI, GitLab CI or Github Action
    b. Automate the process of building, testing, and validating ML models with each code commit.
5. Containerization with Docker:
    a. Containerize ML models and their dependencies using Docker.
    b. Experiment with Docker images, containers, and Dockerfile configurations.
6. Orchestrating ML Workflows with Kubernetes:
    a. Deploy ML models as scalable and resilient services using Kubernetes.
    b. Experiment with deploying, managing, and scaling ML workloads in Kubernetes clusters.
7. Experiment Tracking and Management:
    a. Use tools like MLflow or Neptune.ai to track experiments, log metrics, and manage model versions.
    b. Explore features like hyperparameter tuning, model registry, and experiment reproducibility.

8. Continuous Deployment (CD) for ML Models:
    a. Implement a CD pipeline to automate the deployment of ML models to production.
    b. Deploy an ML model for image classification using AWS SageMaker.
9. Monitoring and Alerting:
    a. Set up monitoring and alerting systems to track model performance, data drift, and anomalies.
    b. Experiment with Prometheus, Grafana, or DataDog to visualize and monitor ML system metrics while evaluating SLA/SLO targets and understanding the difference between monitoring and logging.
10. Model Performance Optimization:
    a. Explore techniques for optimizing model performance, such as model quantization, pruning, or distillation.
    b. Experiment with different optimization approaches and measure their impact on model efficiency.
11. A/B Testing and Experimentation:
    a. Design and conduct A/B tests to compare the performance of different ML models or algorithms.
    b. Experiment with statistical analysis and hypothesis testing to evaluate model improvement understand the importance of model governance and compliance in regulated industries.
    c. Perform A/B testing by comparing two model versions while ensuring system reliability through health checks, API connectivity validation, and stable database connections
12. Infrastructure as Code (IaC) for ML:
    a. Use tools like AWS Cloud Formation to manage basic instance creation in ML infrastructure.
    b. Experiment with provisioning and automating the setup of ML environments.

    Minimum eight 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:

Textbooks:

1. Noah Gift , "Practical MLOps: A Guide to Building Real-World Machine Learning Systems", O'Reilly, First Edition, September 2021.
2. Mark Treveil, Nicolas Omont, "Introducing MLOps: How to Scale Machine Learning in the Enterprise", O'Reilly Media, First Edition, January 5, 2021
3. Emmanuel Raj, "Engineering MLOps: Rapidly build, test, and manage production-ready machine learning life cycles at scale", Packt Publishing Limited, 1st edition, 19 April 2021

Reference Books:

1. Hannes Hapke and Catherine Nelson, "Building Machine Learning Pipelines: Automating Model Life Cycles with TensorFlow", O'Reilly, First Edition, 19 July 2020.
2. Chris Fregly, Antje Barth, "Data Science on AWS: Implementing End-to-End Continuous Machine Learning Pipelines", O'Reilly, First Edition, 9 May 2021.
3. Sridhar Alla, Suman Kalyan Adari, "Beginning MLOps with MLFlow: Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure", Apress publication, 1st edition, 8 December 2020.

Web Link:

1. Machine Learning Operations, https://ml-ops.org/
2. MLOps: Continuous delivery and automation pipelines in machine learning,

https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

3. Why You Need MLOps, https://www.nvidia.com/en-us/software/run-ai/#referrer=site&domain=run-ai

Online Courses:

1. MLOps | Machine Learning Operations Specialization, https://www.coursera.org/specializations/mlops-machine-learning-duke
2. Machine Learning in Production, https://www.coursera.org/learn/introduction-to-machine-learning-in-production
3. Machine Learning Operations (MLOps): Getting Started, https://www.coursera.org/learn/mlops-fundamentals

Prepared by | Checked by | Head of the Department | Vice Principal | Principal

Program: B. Tech in Artificial Intelligence (AI) and Data Science **T.Y. B. Tech Semester: VI**

Course: Constitution of India (DJS23ICHSX09)

Prerequisite: NA

Course Objectives: The Objectives of course are

1. To provide basic information about Indian constitution.
2. To identify individual role and ethical responsibility towards society.
3. To understand human rights and its implications.

Course Outcomes: Students will be able to

1. Have general knowledge and legal literacy and thereby to take up competitive examinations.
2. Understand state and central policies, fundamental duties.
3. Understand Electoral Process, special provisions.
4. Understand powers and functions of Municipalities, Panchayats and Co-operative Societies.
5. Understand Engineering ethics and responsibilities of Engineers.
6. Understand Engineering Integrity & Reliability.

Constitution of India (DJS23ICHSX09)		
Unit No	**Description**	**Duration**
1	**Introduction to the Constitution of India:** The Making of the Constitution and Salient features of the Constitution. Preamble to the Indian Constitution. Fundamental Rights & its limitations.	02
2	**Directive Principles of State Policy:** Relevance of Directive Principles, State Policy, Fundamental Duties. Union Executives – President, Prime Minister, Parliament, Supreme Court of India.	02
3	**State Executives:** Governor, Chief Minister, State Legislature, High Court of State. Electoral Process in India, Amendment Procedures, 42nd, 44th, 74th, 76th, 86th & 91st Amendments.	03
4	**Special Provisions:** For SC & ST, Special Provision for Women, Children & Backward Classes, Emergency Provisions	02
5	**Human Rights:** Meaning and Definitions, Legislation Specific Themes in Human Rights, Working of National Human Rights Commission in India, Powers and functions of Municipalities, Panchayats and Co- Operative Societies	03
6	**Scope & Aims of Engineering Ethics:** Responsibility of Engineers and Impediments to Responsibility. Risks, Safety and liability of Engineers, Honesty, Integrity & Reliability in Engineering	02
	Total	14

Text books:

1. Durga Das Basu, Introduction to the Constitution on India, (Students Edition) Prentice Hall EEE, 19th / 20th Edition, 2001
2. Charles E. Haries, Michael S. Pritchard and Michael J. Robins, Engineering Ethics, Thompson Asia, 2003

Reference Books:

1. M. V. Pylee, "An Introduction to Constitution of India", Vikas Publishing, 3rd Edition, 2003
2. M. Govindarajan, S. Natarajan, V. S. Senthilkumar, "Engineering Ethics", Prentice Hall of India Pvt. Ltd. New Delhi, 2013
3. Brij Kishore Sharma, "Introduction to the Constitution of India", PHI Learning Pvt. Ltd., New Delhi, 7th Edition 2015
4. Latest Publications of Indian Institute of Human Rights, New Delhi

Online Resources:

1. www.nptel.ac.in
2. www.hnlu.ac.in
3. www.nspe.org
4. www.preservearticles.com

Prepared by Checked by Head of the Department Vice Principal Principal

Program: B. Tech in Artificial Intelligence (AI) and Data Science T.Y. B. Tech Semester: VI
Course: Innovative Product Development IV (DJS23IPSCX04)

Course Objectives: The objectives of the course are

1. To acquaint the students with the process of identifying the need (considering a societal requirement) and ensuring that a solution is found out to address the same by designing and developing an innovative product
2. To familiarize the students with the process of designing and developing a product, while they work as part of a team
3. To acquaint the students with the process of applying basic engineering fundamentals, so as to attempt at the design and development of a successful value added product
4. To inculcate the basic concepts of entrepreneurship and the process of self-learning and research required to conceptualize and create a successful product

Course Outcome: Students will be able to

1. Identify the requirement for a product based on societal/research needs
2. Apply knowledge and skills required to solve a societal need by conceptualizing a product, especially while working in a team
3. Use standard norms of engineering concepts/practices in the design and development of an innovative product
4. Draw proper inferences through theoretical/ experimental/simulations and analyse the impact of the proposed method of design and development of the product
5. Develop interpersonal skills, while working as a member of the team or as the leader
6. Demonstrate capabilities of self-learning as part of the team, leading to life-long learning, which could eventually prepare themselves to be successful entrepreneurs
7. Demonstrate product/project management principles during the design and development work and also excel in written (Technical paper preparation) as well as oral communication

Guidelines for the proposed product design and development:

- Students shall form a team of 3 to 4 students (max allowed: 5-6 in extraordinary cases, subject to the approval of the department review committee and the Head of the department)
- Students should carry out a survey and identify the need, which shall be converted into conceptualization of a product, in consultation with the faculty supervisor/head of department/internal committee of faculty members
- Students in the team shall understand the effective need for product development and accordingly select the best possible design in consultation with the faculty supervisor
- Students shall convert the best design solution into a working model, using various components drawn from their domain as well as related interdisciplinary areas
- Faculty supervisor may provide inputs to students during the entire span of the activity, spread over 2 semesters, wherein the main focus shall be on self-learning
- A record in the form of an activity log-book is to be prepared by each team, wherein the team can record weekly progress of work. The guide/supervisor should verify the recorded notes/comments and approve the same on a weekly basis
- The design solution is to be validated with proper justification and the report is to be compiled in a standard format and submitted to the department. Efforts are to be made by the students to try and publish a technical paper, either in the institute journal, "Techno Focus: Journal for Budding Engineers" or at a suitable publication, approved by the department research committee/ Head of

the department

- The focus should be on self-learning, capability to design and innovate new products as well as on developing the ability to address societal problems. Advancement of entrepreneurial capabilities and quality development of the students through the year long course should ensure that the design and development of a product of appropriate level and quality is carried out, spread over two semesters, i.e. during the semesters V and VI

Guidelines for Assessment of the work:

- The review/ progress monitoring committee shall be constituted by the Head of the Department. The progress of design and development of the product is to be evaluated on a continuous basis, holding a minimum of two reviews in each semester
- In the continuous assessment, focus shall also be on each individual student's contribution to the team activity, their understanding and involvement as well as responses to the questions being raised at all points in time
- Distribution of term work marks during semester V shall be as given below:
  - Marks awarded by the supervisor based on performance 10
  - Marks awarded by Review Committee 10
  - Quality of Documentation 05

Review/progress monitoring committee may consider the following points during the assessment.

- In the semester V, the entire design proposal shall be ready, including components/system selection as well as the cost analysis
- Two reviews will be conducted based on the presentation given by the student's team

Tentative rubrics for reviews will be as follows:
Review 1:

i. Implementation Details & Status (60% project implementation)
ii. Design & System Specifications
iii. Presentation Quality
iv. Contribution as a team member and Punctuality
v. Project Documentation.

Review 2:

i. Implementation Details & Status (90% project implementation)
ii. Draft copy of research paper.
iii. Draft copy of Copy right or Patent if applicable .

First review is based on readiness of building the working prototype.
Second review shall be based on a presentation as well as the demonstration of the working model. This review will also look at the readiness of the proposed technical paper presentation of the team.

The overall work done by the team shall be assessed based on the following criteria:

1. Quality of survey/ need identification of the product
2. Clarity of Problem definition (design and development) based on need
3. Innovativeness in the proposed design
4. Feasibility of the proposed design and selection of the best solution
5. Cost effectiveness of the product
6. Societal impact of the product

7. Functioning of the working model as per stated requirements
8. Effective use of standard engineering norms
9. Contribution of each individual as a member or the team leader
10. Clarity on the write-up and the technical paper prepared

Guidelines for Assessment of Semester Reviews:

- The write-up should be prepared as per the guidelines given by the department
- The design and the development of the product shall be assessed through a presentation and demonstration of the working model by the student team to a panel of Internal and External Examiners, preferably from industry or any research organisations having an experience of more than five years, approved by the Head of the Institution. Students are compulsorily required to present the outline of the technical paper prepared by them during the final review in semester VI

Prepared by | Checked by | Head of the Department | Vice Principal | Principal