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)

Final Year B. Tech

in

INFORMATION TECHNOLOGY

(Semester VII and VIII)

Revision: 3 (2026)

With effect from the Academic Year: 2026-2027

1st July 2026

SEM VII/VIII (ACADEMICS BATCH) COMPULSORY COURSES			SEM VII/VIII (OJT BATCH) COMPULSORY COURSES
Program Core Courses	Service Oriented Architecture (3+1 Credits)		Internship (OJT) cum Project (Credits:18)
Multidisciplinary Minor Course	Deep Learning (3+1 Credits)		
Skill Enhancement Courses	Coursera (MOOC) - 1 Credit		Coursera (MOOC) – 1 Credit
Experiential Learning Courses	Major Project (4 Credits)		
	Research Ecosystem (2 Credits)		
	Disaster Management and Preparedness (Audit)		
SEM VII/VIII (ACADEMICS BATCH) ELECTIVE COURSES			**SEM VII/VIII (OJT BATCH) ELECTIVE COURSE**
ELECTIVE BUCKET 1 (Credits: 3+1)	**ELECTIVE BUCKET 2 (Credits: 3)**		**ELECTIVE 1 (Online) (Credits: 3)**
Design Patterns	Responsible AI		Biomedical Informatics
Software Testing & Quality Assurance	Multimodal Computer Vision		Agent Modeling in Finance
Predictive Analytics	Agentic AI		Language Modeling
Blockchain Technology	Advanced Security		Nature Inspired Learning Algorithms
High Performance Computing	Edge Computing		Quantum Computing
TOTAL CREDITS:	**22**		**22**

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 Final Year Undergraduate Program in Information Technology: Autonomous

(Academic Year 2026-2027)

SEMESTER VII & VIII

Sr. No.	CourseCode	Course	Teaching Scheme				Semester End Examination (A)						Continuous Assessment (B)						Aggregate (A+B)	Credits Earned	
			Theory (hrs)	Practical (hrs)	Tut (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	DJS23ICPC701	Service Oriented Architecture	3	--	--	**3**	2	60	--	--	--	**60**	15	15	10	40	--	**40**	**100**	**3**	**4**
	DJS23ILPC701	Service Oriented Architecture Laboratory	--	2	--	**1**	--	--	25	--	--	**25**	--	--	--	--	25	**25**	**50**	**1**	
2@	DJS23ICPE711	Design Patterns	3	--	--	**3**	2	60	--	--	--	**60**	15	15	10	40	--	**40**	**100**	**3**	**4**
	DJS23ILPE711	Design Patterns Laboratory	--	2	--	**1**	--	--	25	--	--	**25**	--	--	--	--	25	**25**	**50**	**1**	
	DJS23ICPE712	Software Testing and Quality Assurance	3	--	--	**3**	2	60	--	--	--	**60**	15	15	10	40	--	**40**	**100**	**3**	
	DJS23ILPE712	Software Testing and Quality Assurance Laboratory	--	2	--	**1**	--	--	25	--	--	**25**	--	--	--	--	25	**25**	**50**	**1**	
	DJS23ICPE713	Predictive Analytics	3	--	--	**3**	2	60	--	--	--	**60**	15	15	10	40	--	**40**	**100**	**3**	
	DJS23ILPE713	Predictive Analytics Laboratory	--	2	--	**1**	--	--	25	--	--	**25**	--	--	--	--	25	**25**	**50**	**1**	
	DJS23ICPE714	Blockchain Technology	3	--	--	**3**	2	60	--	--	--	**60**	15	15	10	40	--	**40**	**100**	**3**	
	DJS23ILPE714	Blockchain Technology Laboratory	--	2	--	**1**	--	--	25	--	--	**25**	--	--	--	--	25	**25**	**50**	**1**	
	DJS23ICPE715	High Performance Computing	3	--	--	**3**	2	60	--	--	--	**60**	15	15	10	40	--	**40**	**100**	**3**	
	DJS23ILPE715	High Performance Computing Laboratory	--	2	--	**1**	--	--	25	--	--	**25**	--	--	--	--	25	**25**	**50**	**1**	
3@	DJS23ICPE721	Responsible AI	3	--	--	**3**	2	60	--	--	--	**60**	15	15	10	40	--	**40**	**100**	**3**	**3**
	DJS23ICPE722	Multimodal Computer Vision	3	--	--	**3**	2	60	--	--	--	**60**	15	15	10	40	--	**40**	**100**	**3**	
	DJS23ICPE723	Agentic AI	3	--	--	**3**	2	60	--	--	--	**60**	15	15	10	40	--	**40**	**100**	**3**	

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

	DJS23ICPE724	Advanced Security	3	--	--	**3**	2	60	--	--	--	**60**	15	15	10	40	--	**40**	**100**	**3**	
	DJS23ICPE725	Edge Computing	3	--	--	**3**	2	60	--	--	--	**60**	15	15	10	40	--	**40**	**100**	**3**	
4	DJS23ICMD701	Deep Learning	3	--	--	**3**	2	60	--	--	--	**60**	15	15	10	40	--	**40**	**100**	**3**	**4**
	DJS23ILMD701	Deep Learning Laboratory	--	2	--	**1**	--	--	25	--	--	**25**	--	--	--	--	25	**25**	**50**	**1**	
5	DJS23ITELX12	Research Ecosystem	--	--	2	**2**	--	--	--	--	--	**--**	--	--	--	--	50	**50**	**50**	**2**	**2**
6	DJS23IPELX13	Major Project	--	8	--	**4**	--	--	--	--	50	**50**	--	--	--	--	50	**50**	**100**	**4**	**4**
7	DJS23ICSC701	DevOps and AI on AWS Specialization	1	--	--	**1**	--	--	--	--	--	--	15	15	10	40	--	**40**	**40**	**1**	**1**
8	DJS23ICHSX11	Disaster Management and Preparedness (Audit)	2	--	--	-	--	--	--	--	--	--	--	--	--	--	--	--	--	--	**--**
9@	DJS23ICPE811	Biomedical Informatics	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	**40**	**100**	**3**	**3**
	DJS23ICPE812	Agent Modeling in Finance	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	**40**	**100**	**3**	
	DJS23ICPE813	Language Modeling	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	**40**	**100**	**3**	
	DJS23ICPE814	Nature Inspired Learning Algorithms	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	**40**	**100**	**3**	
	DJS23ICPE815	Quantum Computing	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	**40**	**100**	**3**	
10	DJS23IPELX14	Internship (OJT) cum Project	--	36	--	**18**	2	--	--	--	200	**200**	--	--	--	--	**400**	**400**	**600**	**18**	**18**
11	DJS23ICSC801	Multi-Agent Systems and Production AI	**1**	**--**	**--**	**1**	**--**	**--**	**--**	--	**--**	**--**	**15**	**15**	**10**	**40**	**--**	**40**	**40**	**1**	**1**
		Total	**19**	**50**	**2**	**44**	**20**	**300**	**75**	**--**	**250**	**625**	**105**	**105**	**70**	**280**	**575**	**855**	**1480**	**44**	

@Any 1 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

Course: Service Oriented Architecture (DJS23ICPC701)
Course: Service Oriented Architecture Laboratory (DJS23ILPC701)

Pre-requisite: Knowledge of programming, Distributed Computing

Course Objectives: The objective of this course is to introduce students to Service Oriented Architecture with its characteristics and advantages. Students will also understand the distinction between client-server, two-tier, three-tier and enterprise architectures. The course also teaches the basics of web services and introduces SOAP, REST, WSDL and UDDI. It highlights the SOAP and REST architecture along with its importance and standards.

Course Outcomes: On successful completion of this course, students should be able to:
1. Create web services using development tools.
2. Build SOA-based solutions for intra-enterprise and inter-enterprise applications.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**SOA Foundations and Modern API Design:** Concepts of Distributed Computing, Fundamental of SOA (evolution of SOA, characteristics, principles), Comparing SOA to client-server and distributed internet architectures, SOAP and WSDL, RESTful API design (resources, HTTP methods, statelessness), OpenAPI 3.x (contract-first design, schema definition, API documentation), API versioning strategies.	06
2	Service Design Principles and Communication Protocols: Service orientation vs object-orientation, SOA standards stack, service composition guidelines, service contract design, API-first design, REST vs gRPC, gRPC fundamentals, Protocol Buffers, .proto schema, code generation and service communication models.	07
3	**Event-Driven Architecture and Messaging:** Message exchange patterns, message brokers, queues and topics, messaging topologies, Apache Kafka fundamentals, Event-Driven Architecture, CQRS (Overview, Benefits, Limitations), Saga Pattern for distributed transactions – Choreography vs. Orchestration, WS-Security overview.	8
4	**API Gateways, Service Discovery, Business Process Modeling and Resilience:** API Gateway concepts, Service Discovery concepts (Consul / Eureka), BPMN 2.0 (Events, Activities, Gateways, Pools and Lanes), Circuit Breaker, Retry, Bulkhead, Timeout (Resilience4j), Load balancing strategies (client-side vs. server-side); Cloud-native API management overview (AWS API Gateway / Azure API Management as examples).	08
5	**Microservices Architecture, DDD and Security:** Microservices architecture principles – single responsibility, loose coupling, independent deployment; Decomposition strategies (by business capability, by subdomain); Domain-Driven Design (DDD) – Bounded Contexts, Aggregates, Ubiquitous Language; OAuth 2.0 authorization flows – Authorization Code, Client Credentials; JWT – structure, claims, signing, validation, expiry handling.	07
6	**Distributed Systems Design & Architecture Trade-offs:** Real-world SOA case studies – how Netflix, Uber, and Amazon decomposed monoliths; Architecture trade-off framework (when to use REST vs. gRPC vs. Kafka for a given scenario); SOA governance (service versioning strategy, backward compatibility, deprecation policies); System design – analyse, critique, and propose improvements to given architectures.	06

List of Laboratory Experiments:

1. Design an OpenAPI 3.x spec, generate Spring Boot server stubs, and implement CRUD endpoints.
2. Build a REST API using FastAPI with Pydantic schema validation and auto-generated OpenAPI docs.
3. Define a .proto schema, generate stubs, and implement a gRPC server with Unary and Server-Streaming RPCs.
4. Implement a Kafka producer-consumer pipeline with consumer groups and a Dead Letter Queue for failed messages.
5. Implement an Order Processing system using CQRS and a choreography-based Saga pattern.
6. Configure Spring Cloud Gateway with routing, rate limiting, and JWT validation. Test with valid and tampered tokens.
7. Register two services with Eureka, simulate a service failure, and observe automatic deregistration via health checks.
8. Implement Circuit Breaker, Retry, and Bulkhead patterns using Resilience4j between two Spring Boot services.
9. Implement OAuth 2.0 Authorization Code flow with JWT. Validate claims, test expiry, and demonstrate scope-based access control.
10. Given a monolithic e-commerce specification, produce a microservices decomposition diagram, DDD Bounded Context map, and protocol choices (REST/gRPC/Kafka) with justification.

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

Books Recommended:

Text books:

1. Thomas Erl, “Service Oriented Architecture: Concepts, Technology, and Design”, 1st Edition, Pearson Education, 2006.
2. Thomas Erl, “SOA Principles of Service Design”, 1st Edition, Pearson Education, 2008.
3. Sam Newman, “Building Microservices: Designing Fine-Grained Systems”, 2nd Edition, O'Reilly Media, Inc., 2021.

Reference Books:

1. Mark D. Hansen, “SOA Using Java™ Web Services”, 1st Edition, Prentice Hall, 2007.
2. Thomas Erl, “Web Service Contract Design & Versioning for SOA”, 1st Edition, Pearson, 2008.
3. Gregor Hohpe and Bobby Woolf, “Enterprise Integration Patterns: Designing, Building and Deploying Messaging Solutions”, 1st Edition, Addison-Wesley Professional, 2003
4. Martin Kleppmann, “Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable and Maintainable Systems”, 1st Edition, O’Reilly Media, 2017.

Other Resources:

1. https://www.coursera.org/learn/google-cloud-java-spring?utm_source=chatgpt.com (Building Scalable Java Micro Services with Spring Boot and Spring Cloud)

Course: Design Patterns (DJS23ICPE711)
Course: Design Patterns Laboratory (DJS23ILPE712)

Pre-requisite: Object Oriented Programming, Software Engineering.

Course Objectives: The course aims to build a strong foundation in object-oriented software design by introducing standard design patterns and their significance in modern software development. It enables students to understand structured approaches for designing flexible, maintainable, and reusable software systems using proven design principles and patterns. The course also familiarizes learners with the evolution of software design from traditional object-oriented systems to enterprise architectures, service-oriented architectures (SOA), and modern distributed systems used in real-world applications.

Course Outcomes: On successful completion of this course, students should be able to:

1. Apply appropriate creational, structural, and behavioral design patterns to develop flexible and maintainable object-oriented software solutions.
2. Analyze system design problems and formulate service-oriented architecture solutions using appropriate SOA design patterns.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to Design Patterns:** Definition and characteristics of design patterns, Elements of design patterns (Intent, Structure, Participants, Consequences), Design principles: SOLID, DRY, Open-Closed Principle, Pattern classification (Creational, Structural, Behavioral), Selecting and applying design patterns.	04
2	**Creational Design Patterns:** Singleton, Factory Method, Abstract Factory, Builder, Prototype.	06
3	**Structural Design Patterns:** Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy.	08
4	**Behavioral Design Patterns:** Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor.	09
5	**Foundations and Evolution of SOA Design Patterns:** SOA Design Pattern Fundamentals (Design pattern, Compound pattern, Pattern language, Pattern catalog), Object-Oriented Patterns, Software Architecture Patterns, Enterprise Application Architecture (EAA) Patterns, Enterprise Integration Architecture (EAI) Patterns.	08
6	**Service-Oriented Design and Composition Patterns:** Service Inventory Patterns (Service Normalization, Utility Abstraction, Process Centralization), Service Design Patterns (Service Encapsulation, Service Data Replication, Contract Centralization), Service Composition Patterns (Capability Composition, Service Messaging, Atomic Service Transaction).	07

List of Laboratory Experiments:

1. Implement the Singleton Pattern in any language of your choice ensuring prevention against Reflection, Serialization, and Cloning.
2. Design a UML class diagram for the Abstract Factory Pattern and implement the same for any real-life scenario. (For example, implement a system to list courses offered to students based on college and department using Abstract Factory Pattern.)
3. Design a UML class diagram for the Flyweight Pattern and implement the same to demonstrate the working of a game scenario. (For example, instead of creating objects for each player in a Counter Strike game, create shared objects for Terrorists and Counter Terrorists and reuse them.)

4. Design a UML class diagram for the Proxy Pattern and implement the same for a real-life scenario. (For example, implement a virtual proxy for image loading or a protection proxy for access control in a system.)
5. Implement the Chain of Responsibility Pattern for a real-life scenario. (For example, a system where multiple managers and executives can approve, or forward purchase requests based on defined rules.)
6. Implement the Observer Pattern for a real-life scenario. (For example, a system where registered investors are notified whenever a stock price changes.)
7. Design a UML class diagram for the Strategy Pattern and implement the same. (For example, encapsulate different sorting algorithms and allow dynamic selection of sorting strategy at runtime.)
8. Implement a Compound Pattern by combining multiple design patterns to solve a complex problem. (For example, implement a system similar to MVC architecture or Duck Simulator that combines patterns such as Composite, Observer, Strategy, and Adapter to demonstrate coordinated behavior.)
9. Case study: Analyze the evolution of design patterns from Object-Oriented Patterns to Software Architecture Patterns, Enterprise Application Architecture (EAA), Enterprise Integration Architecture (EAI), and SOA Design Patterns. Illustrate mappings with suitable examples.
10. Case study: Study and analyze Enterprise Integration Patterns using a real-world case study (https://www.enterpriseintegrationpatterns.com/BondTradingCaseStudy.html).

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

Books Recommended:

Textbooks:

1. Anand Rajaraman and Jeff Ullman, “Mining of Massive Datasets”, 2nd Edition, Cambridge University Press, 2014.
2. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, “Design Patterns: Elements of Reusable Object- Oriented Software”, Addison-Wesley Professional, 1994.
3. Thomas Erl, “SOA Design Pattern”, Pearson, 2008.

Reference Books:

1. Eric Freeman, Elisabeth Robson, Bert Bates and Kathy Sierra, “Head First Design Patterns”, O’Reilly Media, Inc., 2004.
2. Vaskaran Sarcar, “Design Patterns in C#: A Hands-on Guide with Real-World Examples”, Apress, 2018.

Program: B. Tech **Information Technology** **Final Year** **Semester: VII/VIII**

Course: Software Testing and Quality Assurance (DJS23ICPE712)
Course: Software Testing and Quality Assurance Laboratory (DJS23ILPE712)

Pre-requisite: Software Engineering.

Course Objectives: The course covers quality software development practices, including testing techniques, quality models, lifecycle models, and unit-to-acceptance testing processes.

Course Outcomes: On successful completion of this course, students should be able to:

1. Apply appropriate software testing techniques to derive effective test cases.
2. Analyze software quality attributes for effective software evaluation.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Testing Foundations:** Software Quality Concepts and Testing Motivation, Verification and Validation, Faults Errors Failures and Defects, Testing Levels, Testability Observability and Controllability, Testing Maturity and Quality Mindset, Introduction to automation testing, Common types of automation tests.	06
2	**Testing Techniques & Agile Practices:** Unit Testing, Defect Prevention, Dynamic Unit Testing, Debugging, XP Unit Testing, System Integration Testing, Integration Techniques, Software-Hardware Integration, Agile Testing Principles, Test-Driven Development, Behavior-Driven Development, Acceptance Test-Driven Development, Continuous Testing, Regression in Agile.	08
3	**Advanced and Unique Testing Methods:** Symbolic execution, proof-based testing, mutation testing, fault-based testing, domain testing, input-space analysis, grammar-based testing, specification mutation, counterexample-based test generation, object-oriented software testing, web and API testing, testing of distributed and cloud-based systems, performance and reliability testing, security testing, usability testing, compliance testing, embedded systems testing, real-time systems testing, and database testing.	06
4	**Testing Across the Software Development Lifecycle:** Test Testing in the software development lifecycle, Test planning and documentation, Requirements traceability, Static and proactive quality assurance, Test execution and monitoring, Metric based monitoring of test execution, Defect reporting, Defect classification and analysis, Process improvement and maturity models, Using test and defect metrics for process improvement.	08
5	**Test Automation & Infrastructure:** Test Automation Architecture, Test Harnesses and Drivers, Stubs and Mocks, Scaffolding, GenAI in Software Test Automation, Automated Test Case Structure, AI-Augmented Code Reviews and Static Analysis, CI/CD-Aligned Regression Automation, Tool Selection, Version Control, Test Maintenance.	08
6	**Software Quality: Five Views of Software Quality,** McCall‟s Quality Factors and Criteria, Quality Factors Quality Criteria, Relationship between Quality Factors and Criteria, Quality Metrics, ISO 9126 Quality Characteristics, ISO 9000:2000 Software Quality Standard ISO 9000:2000 Fundamentals, ISO 9001:2000 Requirements, ISO/IEC 25010: Software product quality model , ISO 9000:2015, ISO 9001:2015.	06

List of Laboratory Experiments:

1. Prepare a test case verification document for a given scenario.
2. Detailed Test Plan in IEEE format for given case study.

3. White Box Testing on Units/Modules for given case study.
4. Black Box Testing on Units/Modules for given case study
5. To design test cases for given problem statement based on Decision Table Testing method.
6. To study software Automation Testing with JMeter
7. Study of Automation Software Testing with JUnit.
8. To study software Automation Testing tool WinRunner for Setting Up the GUI Map
9. Use Git/GitHub for version control in a software project.
10. To study software Automation Testing tool WinRunner for Checking GUI Objects.
11. To study software Automation Testing tool WinRunner Creating Data-Driven Tests.
12. To generate a resilient Playwright/TypeScript login test using Gen-AI.

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

Books Recommended:

Text books:

1. Yogesh Singh, "Software Testing", Cambridge University Press, 2012
2. Paul Ammann and Jeff Offutt, "Introduction to Software Testing", Cambridge University Press, 2008.
3. Kshirasager Naik and Priyadarshi Tripathy, "Software Testing and Quality Assurance", Wiley, 2008.
4. Mauro Pezze and Michal Young, "Software Testing and Analysis: Process, Principles, and Techniques", Wiley, 2007.

Reference Books:

1. Srinivasan Desikan and Gopalaswamy Ramesh,"Software Testing", Pearson Education, 2006
2. Paul C. Jorgensen, "Software Testing - A Craftsman's Approach", CRC Press, 1995.
3. Boris Beizer, "Software Testing Techniques", Van Nostrand Reinhold, 2nd Edition, 1990.
4. Bob Hughes, Mike Cotterell and Rajib Mall, "Software Project Management", 5th Edition, Mcgraw-Hill, 2009.

Course: Predictive Analytics (DJS23ICPE713)
Course: Predictive Analytics Laboratory (DJS23ILPE713)

Pre-requisite: Knowledge of Statistical Analytics and Machine Learning.

Course Objectives: The objective of this course is to develop a strong understanding of the computational foundations of Data Science and to enhance critical and inferential thinking skills.

Course Outcomes: On successful completion of this course, students should be able to:

1. Apply predictive modeling techniques to analyze data and derive actionable insights.
2. Select and implement appropriate models for solving prediction problems.
3. Evaluate and compare different models to improve predictive performance and accuracy.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to Predictive Analytics and Problem Framing:** Introduction to Analytics, Analytics Overview, Types of Analytics (Descriptive, Diagnostic, Predictive, Prescriptive), Business Analytics vs Business Intelligence vs Data Mining, Why Analytics and its role in industry applications, Identifying business problems and opportunities using data analytics, Framing business problems as analytical problems, CRISP-DM methodology, Data sources and types, Data collection and data management.	06
2	**Data Understanding, Preprocessing and Visualization:** Sampling methods: random, stratified, systematic, Descriptive statistics: mean, variance, skewness, kurtosis, Data cleaning: handling missing values, outliers, noise, Data transformation and normalization, Exploratory Data Analysis, Data visualization techniques: histograms, boxplots, scatter plots, Correlation matrices and covariance analysis, Trees and other Predictive Models.	07
3	**Regression and Model Building:** Simple Linear Regression, Regression analysis, types of regression, SLR estimation and prediction, model evaluation metrics (R^2, RMSE, MAE), Applications of SLR, Multiple Linear Regression (MLR), characteristics and interpretation of regression coefficients, hypothesis testing (t-test, F-test, partial F-test), multicollinearity, Kednal's Tau, dummy variables and categorical predictors, interaction variables, feature engineering, encoding categorical variables, scaling and transformation techniques, model building, model diagnostics, ridge and lasso regression.	07
4	**Multivariate Analysis and Model Evaluation:** Introduction to multivariate analysis, types of multivariate techniques, factor analysis, principal component analysis, correlation matrix and extraction of factors, factor loadings and interpretation, model validation techniques, model evaluation and performance metrics: precision, recall, F1-score, rule induction using CHAID, comparing and combining models, meta-level modeling, model deployment concepts, monitoring and updating models.	07
5	**Time Series Analysis and Forecasting:** Introduction to time series, Time Series Databases, time series components: trend, seasonality, irregular variations, decomposition of time series: additive and multiplicative models, autocorrelation (ACF), moving average models, autoregressive models, exponential smoothing techniques, Holt and Holt-Winters methods.	08

6	**Modern Predictive Analytics, LLMs and Ethical AI:** LLMs for predictive modeling, prompt engineering techniques for analytics, applications in financial forecasting, augmented analytics, AutoML and no-code analytics tools, human-machine collaboration, ethical considerations in analytics, bias and fairness in algorithms, data privacy and security, responsible AI practices.	**07**

List of Laboratory Experiments:

1. To perform data preprocessing, data analysis techniques on sample dataset.
2. To perform basic operations (Sorting file, split file, Compute, Recode, Select cases) on dataset in SPSS.
3. To perform model building and developing using IBM watsonx.ai Auto AI.
4. To apply principal component analysis (PCA) and linear discriminant analysis (LDA) on Large Dataset.
5. To implement, compare, and critically analyze Ridge Regression (L2 regularization) and Lasso Regression (L1 regularization).
6. To demonstrate and implement Forward Selection, Backward Elimination, and Stepwise Regression techniques for feature selection
7. To implement Dummy Regression and One Hot Encoding & compare them.
8. To implement SMA, DMA, Exponential Smoothing, Holt's winter coefficient technique for forecasting.
9. To study and implement LLM models for financial forecasting.
10. To apply CHAID on a dataset with categorical outcomes (e.g., customer segmentation) to identify distinct decision rules.
11. Evaluating Foundation Models Using Amazon Bedrock
12. Monitoring and Evaluating Forecasting Results Using InfluxDB

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

Books Recommended:

Text books:

1. Dean Abott, "Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst", Wiley Publication, 2014.
2. Peter Christie, Jim Georges, Jeff Thompson and Chip Wells, "Applied Analytics Using SAS Enterprise Miner", SAS Institute Inc., 2016.
3. Ramchandra Mangrulkar and Pallavi Chavan, "Predictive Analytics with SAS and R: Core Concepts, Tools, and Implementation", 1st Edition, Apress Springer, 2024.

Reference Books:

1. Adrian McEwen and Hakim Cassimally, "Designing the Internet of Things", 1st Edition, Wiley, 2013.
2. Walter R. Paczkowski, "Predictive and Simulation Analytics: Deeper Insights for Better Business Decisions", Springer, 2023.
3. Kattmuri S. Sarma, "Predictive Modeling with SAS Enterprise Miner: Practical Solutions for Business Applications", 2nd Edition, Apress, 2016.
4. Edward W. Frees, Glenn Meyers and Richard A. Derrig, "Predictive Modeling Applications in Actuarial Science", Volume 2, Case Studies in Insurance (International Series on Actuarial Science), Cambridge Press, 2016.
5. IBM ICE Publication, "Predictive & Advanced Analytics", (https://www.ibm.com/docs).

Web Resources:

1. https://www.ibm.com/docs/en/spss-statistics
2. https://www.analyticsvidhya.com/
3. https://towardsdatascience.com/deep-dives

Course: Blockchain Technology (DJS23ICPE714)
Course: Blockchain Technology Laboratory (DJS23ILPE714)

Pre-requisite: Computer Networking, Cryptography and Hashing, Data Structures.

Course Objectives: The course will provide students with conceptual understanding of working of blockchain technology. In this course, students will also be able to design and deploy smart contracts and distributed applications using solidity language, which will also help them analyze the use of blockchain technology to innovate and improve business processes.

Course Outcomes: On successful completion of this course, students should be able to:

1. Survey different blockchain platforms, their architectures and applications.
2. Develop smart contracts on Ethereum framework using solidity language.
3. Analyze the real-world problems that can be solved using blockchain technology.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to Blockchain:** Prerequisites, Blockchain Myths, Blockchain and Decentralization, Blockchain as Disruption Technology, History, Features of Blockchain, Present Growth, Predicted Market, Blockchain Types, Blockchain Framework, A Block and its Structure, Scaling Blockchain, Blockchain DApps and Use Cases.	07
2	**Decentralization of Network**: Essentials of Blockchain Programming, Cryptography Primitives, Hash Functions, Public Key Cryptography, Merkle Tree. Decentralized Storage, Consensus mechanism, Consensus Algorithm PoW, PoS, DPoS, issues with PoW, Distributed Consensus, Incentives and Proof of Work, Consensus without Identity.	07
3	**Bitcoin Fundamentals:** Bitcoin, What Is Bitcoin, History, Predicted Market, Wallet, Digital Keys and Addresses, Addresses in Bitcoin, Transaction, Digital Signature, Mining and Consensus in Bitcoin. Byzentine fault tolerance, 51 % attacks, Double spending, Mining and consensus (Mining Incentives and Strategies), Energy consumption.	07
4	**Ethereum Fundamentals and Development Tools:** Overview of Ethereum Blockchain, Key Features, Ethereum 1.0 vs Ethereum 2.0, EVM, History of Ethereum, Ledger to State Machine, Ethereum Network, Smart Contracts, Challenges in Implementing Smart Contracts, Smart Contract Life Cycle, Introducing Solidity, Global Variables, Ethereum Development Tools, Ethereum Transactions, Gas and Transaction Fees. Ethereum development standard.	07
5	**Hyperledger Framework and Enterprise Blockchain** – Hyperledger, Introduction to Hyperledger, Hyperledger Architecture, Hyperledger Community and Development, Hyperledger Smart Contracts (Chaincode), The Functioning of Hyperledger, Hyperledger Projects, Hyperledger Consortiums and Networks, Hyperledger and Blockchain as a Service (BaaS). Hyperledger Fabrics, Hyperledger Iroha.	07
6	**Blockchain use cases beyond Blockchain:** Blockchain's Role in the Metaverse, Web 3.0, Digital Scarcity and Ownership of Virtual Assets, Building Trust and Security in the Decentralized Metaverse, DeFi, NFT, Beyond Cryptocurrency, Blockchain and Sustainable Technologies, World Economic Forum blockchain development toolkit. Blockchain Security and Privacy, Zero Knowledge Proofs, Decentralized Identifiers, Self-Sovereign Identity (SSI), Decentralized AI Models, Blockchain Challenges: Scalability, Security, Privacy and Interoperability.	07

List of Laboratory Experiments:

1. Install the blockchain server and establish a peer-to-peer network of nodes.
2. Building a Private Blockchain with Docker and Ethereum.
3. Create and Deploy Your Private Blockchain on MultiChain.
4. Deploying and interacting with a token contract.
5. Generate the genesis block for the Orderer node and start ordering service (solo node) in the bootstrapping network.
6. Installing Truffle and Ganache and deploying your first smart contract.
7. Perform tests on complied programs to check for possible errors using tools such as Remix IDE, Ropsten Testnet, Ganache and Truffle.
8. Setting Up Development Environment Using Hyperledger Composer.
9. Building a Blockchain PoC using Hyperledger Composer.
10. Perform basic querying of transactions, nodes, blocks using Blockchains, Ethereum Query Language, Bitcoin and Explorer, Ethereum Explorer.
11. Perform file storage using Inter-Planetary File System (IPFS).
12. Perform verifying of LLM Response Authenticity Using Blockchain.

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

Books Recommended:

Text books:

1. Tiana Laurence, "Blockchain for Dummies", 2nd Edition, Wiley Publications, 2017.
2. Daniel Drescher, "Blockchain Basics: A Non-Technical Introduction in 25 Steps", 1st Edition, Apress publisher, 2017.
3. Roberto Infante, "Building Ethereum Dapps", 1st Edition, Manning Publication, 2019.

Reference Books:

1. Ramchandra Mangrulkar and Pallavi Chavan, "Blockchain Essentials: Core Concepts and Implementations", 1st Edition, Apress Springer, 2024.
2. Josh Thompson, "Blockchain: The Blockchain for Beginnings, Guild to Blockchain Technology and Blockchain Programming", 1st Edition, Create Space Independent Publishing Platform, 2017.
3. Andreas Antonopoulos, Gavin Wood, "Mastering Ethereum: Building Smart Contracts and DApps", 1st Edition, O'Reilly Publications, 2018.

Web Resources:

1. https://www.frontiersin.org/journals/blockchain
2. https://hyperledger-fabric.readthedocs.io/en/release-2.5/
3. https://www.talentica.com/blogs/llms-empowering-the-blockchain-ecosystem/

Course: High Performance Computing (DJS23ICPE715)
Course: High Performance Computing Laboratory (DJS23ILPE715)

Pre-requisite: Fundamentals of C programming, Operating Systems, Computer Architecture, and Algorithms.

Course Objectives: The course introduces concepts of Parallel and High-Performance Computing, parallel programming models, GPU computing, and performance optimization techniques. It also covers modern HPC technologies and their applications in AI, cloud computing, and large-scale data processing.

Course Outcomes: On successful completion of this course, students should be able to:

1. Develop parallel computing solutions to programming problems.
2. Analyze the performance and scalability of parallel systems using appropriate metrics and techniques.
3. Apply communication in multiprocessing environment.
4. Apply advanced HPC technologies for real-world applications.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Foundations of Parallel & High-Performance Computing:** Need for Parallel Computing, Abstract model of Parallel computer, multiprocessor architecture, pipelining, array processors, Levels of Parallelism: Instruction, Data, Task, Thread, Parallel vs Distributed Systems, Flynn's Classification, Shared Memory vs Distributed Memory, Tightly Coupled vs Loosely Coupled Systems, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access, Real-world applications of HPC (IT, AI, Big Data).	**05**
2	**Parallel Programming:** Data Dependency Analysis: Types of Dependencies, Loop and Array Dependence, Loop Dependence Analysis, General model of Shared Memory Programming: Process Creation, Mutual Exclusion Principles of Message-Passing Programming, Send and Receive Operations **MPI**: Message Passing Interface Section, Topologies and Embedding, Overlapping Communication with Computation, Collective Communication and Computation Operations, Groups and Communicators, Programming using MPI. **OpenMP**: Standard for Directive Based Parallel Programming, Dense Matrix Algorithms: Matrix-Matrix Multiplication, Programming using OpenMP.	**08**
3	**Performance Analysis & Optimization:** Execution Time, Speedup, Efficiency, Cost, Effect of granularity on performance, Scalability of Parallel Systems, Amdahl's Law and Gustafson's Law, Performance Bottlenecks. Strong vs Weak Scalability, Performance Bottlenecks, GPU Optimization Techniques: Memory Coalescing, Warp Divergence, Loop Unrolling.	**06**
4	**GPU Architecture & CUDA Programming:** Evolution of GPUs, CPU vs GPU Architecture. CUDA Architecture, CUDA Programming Model: Kernels, Threads, Blocks, Grids. CUDA Memory Hierarchy: Global, Shared, Constant, Local Memory, Unified Memory, CUDA-based Case Studies (IT & AI workloads) Introduction to CUDA Programming, data parallelism using multi-GPU computing.	**08**

5	**Multiprocessing Communication:** Introduction to Message Passing, Advantages and features of Message Passing, Message Format, Message Buffering, Multi Data Gram Messaging, Group Communication. **Remote Procedure Call (RPC):** Basic RPC Operations, Parameter Passing. **Remote Object Invocation:** Distributed Objects, binding a Client to an Object, Static Vs Dynamic RMI, Parameter Passing, Java RMI. Message Oriented Communication: Persistence and synchronicity in communication, Message Oriented Transient and Persistent Communications. **High-Performance Cluster Networking:** Introduction to HPC Clusters, Network Interconnects for HPC, InfiniBand Architecture, Remote Direct Memory Access (RDMA) Concepts and Benefits, Low-Latency Communication in Parallel Systems. **Cluster Resource Management and Scheduling:** Introduction to Slurm Workload Manager, Job Submission, Resource Allocation, Queue Management, Batch and Interactive Jobs in HPC Environments.	**08**
6	**Advanced HPC Topics:** **Pthreads**: POSIX thread architecture and programming model, Parallel programming using Pthreads examples. **GPU Libraries**: cuBLAS Library- Introduction to cuBLAS, Performance optimization using cuBLAS, cuDNN Library- Convolution, pooling, and activation operations, Applications of cuDNN in AI workloads. **HPC in Cloud**: Cloud HPC, HPC-as-a-Service (HPCaaS), Virtualization and containerization in HPC, TPU architecture and working principles, Comparison of CPU, GPU, and TPU, TPU applications in machine learning and deep learning HPC with ML Frameworks: TensorFlow, PyTorch.	**07**

List of Laboratory Experiments:

1. Basics of MPI (Message Passing Interface).
2. Implementation of advanced MPI Programs.
3. Basics of OpenMP API.
4. Shared Memory Programming using OpenMP API.
5. Setting up CUDA Development Environment.
6. Programming in CUDA.
7. Distributed Computing using PyTorch.
8. Basics Programs using Go Language.
9. Concurrent Programming using Go routines.
10. Concurrent Programming using Go channels.
11. Deploying an HPC Cluster on AWS Parallel Cluster and Executing MPI/CUDA Jobs.

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

Books Recommended:

Text books:

1. Georg Hager and Gerhard Wellein “Introduction to High Performance Computing for Scientists and Engineers” CRC Press, 2019.
2. Ananth Grama et al., “Introduction to Parallel Computing”, Pearson Education, 2003.
3. Michael J. Quinn, “Parallel Programming in C with MPI and OpenMP”, McGraw Hill, 2003.

Reference Books:

1. David B. Kirk and Wen mei W. Hwu, "Programming Massively Parallel Processors", 4th Edition, Morgan Kaufmann Publishers, 2022.
2. Jason Sanders and Edward Kandrot, "CUDA by Example", Addison-Wesley Professional, 2010.
3. Brian Tuomanen, "Hands on GPU Programming with Python and CUDA", Packt Publishing, 2018.
4. Edson Borin et al., "High Performance Computing in Clouds", CRC Press, 2023.
5. Robert Robey and Yuliana Zamora, "Parallel and High-Performance Computing", Manning Publications, 2021.
6. Douglas Eadline, "High Performance Computing for Dummies", Wiley Publishing, 2009.

Program: B. Tech **Information Technology** **Final Year** **Semester: VII/VIII**

Course: Responsible AI (DJS23ICPE721)

Prerequisite: Machine Learning, Artificial Intelligence, Computer Programming Fundamentals, and Ethics & Technology Governance

Course Objectives: This course aims to introduce the principles and practices of Responsible Artificial Intelligence, covering AI, machine learning, and generative models across the AI lifecycle. It helps students understand ethical risks, bias, fairness, explainability, privacy, security, and governance of AI systems. The course also enables learners to apply Responsible AI frameworks, assurance methods, and real-world practices across critical application domains.

Course outcomes: On successful completion of this course, students should be able to:

1. Interpret the foundations, principles, and need for Responsible AI.
2. Evaluate AI systems for fairness, transparency, privacy, robustness, and accountability.
3. Apply practical tools and documentation methods for responsible AI development.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Foundations of Responsible AI:** Introduction to AI, ML, Generative AI, and foundation models, AI lifecycle, Need for Responsible AI : harms, failures, societal impact, Core dimensions of Responsible AI: fairness, transparency, explainability, privacy, safety, security, accountability, human oversight, Stakeholders in AI systems, Overview of major Responsible AI frameworks and standards, Introduction to documentation and governance artefacts.	06
2	**Bias, Fairness, and Data Responsibility**: Introduction to Bias in AI lifecycle, Types of Bias and Sources of bias in AI - historical bias, sampling bias, measurement bias, label bias, representation bias, evaluation bias, Fairness in AI, Common fairness metrics: demographic parity, equal opportunity, disparate impact, false positive / false negative disparities, Bias detection in structured datasets, Bias mitigation strategies, Responsible data practices, Bias detection and Mitigation in Responsible AI - A Case Study.	08
3	**Explainable AI (XAI)**: Introduction to Explainable AI, interpretability, explain ability, and transparency, Interpretable models vs black-box models, Types of Explanations – Global, Local, feature Importance, rule based explanations, explanation methods - Model-Specific, Model-Agnostic, Partial dependence plots, Surrogate models, Popular XAI Techniques, Evaluation of Explanations, role of XAI in building Responsible AI, Explainable AI in practise – A Real World Case Study.	08
4	**Privacy, Security, Robustness, and Safety in AI**: Privacy risks in AI: Personal data exposure, re-identification, leakage from training data, Privacy-preserving AI concepts: anonymization / pseudonymization, differential privacy overview, federated learning overview, secure data handling, AI security threats: adversarial examples, data poisoning, model theft, prompt injection, Robustness and safety testing, Red teaming and stress testing. Monitoring safety and incident response basics.	07
5	**AI Assurance, Audit, and Continuous Monitoring**: Scope of AI assurance, AI assurance lifecycle: pre-deployment, deployment, post-deployment, Accountability, traceability, and auditability in AI systems, Documentation artefacts: model cards, system cards ,datasheets, risk registers, incident logs ,compliance checklists, Logging and record-keeping for AI systems, Internal AI audits and external AI audits, Verification and validation of AI systems, Monitoring drift, failures, and performance after deployment, Feedback loops, logging, escalation, and	08

	governance for model updates, End-to-end Responsible AI deployment workflow, Post-deployment review and corrective action planning.	
6	**Responsible AI in Practices**: Applications of XAI, AI Guardrails for Safe and Responsible AI Systems, Types of Guardrails, Responsible Data Governance, Responsible Corporate Governance of AI Systems, Responsible AI in Healthcare Governance, Artificial Intelligence, Law and National Security, Case Study – Guardrails in AI based Healthcare Decision Support Systems.	**05**

Books Recommended:

Text books:

1. Virginia Dignum, "Responsible Artificial Intelligence: Designing AI for Human Values", 1st Edition, Springer Nature, 2019.
2. Solon Barocas, Moritz Hardt and Arvind Narayanan, "Fairness and Machine Learning: Limitations and Opportunities", 1st Edition, MIT Press, 2023.
3. Christoph Molnar, "Interpretable Machine Learning: A Guide for Making Black Box Models Explainable", 2nd Edition, Leanpub, 2022.

Reference Books:

1. Michael Kearns and Aaron Roth, "The Ethical Algorithm: The Science of Socially Aware Algorithm Design", 1st Edition, Oxford University Press, 2019.
2. Reza Shokri, Catherine Dwork, Aaron Roth and Thomas DeCristofaro (Editors), "Privacy-Preserving Machine Learning", 1st Edition, Springer, 2018.

Online Resources:

1. Responsible AI for Developers: Fairness & Bias | Coursera
2. Responsible AI for Developers: Interpretability & Transparency | Coursera
3. Responsible AI for Developers: Privacy & Safety | Coursera

Program: B. Tech **Information Technology** **Final Year** **Semester: VII/VIII**

Course: Multimodal Computer Vision (DJS23ICPE722)

Pre-requisite: Basic Mathematics, C Programming, Java, Python.

Course Objectives: The course aims to provide students with a strong foundation in computer vision and multimodal learning, including image representation, feature extraction, and classical vision techniques. It enables learners to understand and apply deep learning models such as CNNs and modern architectures for visual understanding tasks. The course further focuses on multimodal data representation, fusion strategies, and vision–language models for applications like image captioning and visual question answering. It also exposes students to video, audio, and real-world multimodal applications while addressing ethical considerations and emerging trends in AI.

Course Outcomes: On successful completion of this course, students should be able to:

1. Interpret the fundamental concepts of computer vision, digital image representation, and multimodal data.
2. Implement deep learning models, including CNN-based architectures like ResNet, for image classification and object detection tasks.
3. Apply multimodal representation, fusion techniques, and vision–language models for tasks like image captioning and visual question answering.
4. Develop solutions for video, audio, and real-world multimodal applications while considering ethical and societal implications.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Foundations of Computer Vision and Multimodal Learning:** Introduction to Computer Vision, digital image representation, basic image processing concepts, introduction to machine learning in computer vision, concept of multimodal data, types of modalities (image, text, audio, sensor data), importance of multimodal learning, challenges and applications. **Feature Extraction:** Points and patches, Feature detectors, Feature descriptor, Feature matching, Feature tracking, finding gradients using Masks: Sobel, Prewitt, Edge detection, Canny, HOG, GLOH, Gaussian derivative filters, Gabor Filters.	**08**
2	**Deep Learning for Visual Understanding:** Introduction to deep learning in computer vision, neural networks and representation learning, Convolutional Neural Networks (CNN), convolution, pooling and activation layers, transfer learning, feature extraction using pretrained CNN models, popular architectures (AlexNet, VGG, ResNet), image classification. CNN, RCNN, FAST RCNN, FASTER RCNN, YOLO, SSD	**07**
3	**Multimodal Representation, Embeddings and Fusion Techniques:** Multimodal data representation, feature embedding for image, text, audio and video, cross-modal representation learning, embedding spaces, dimensionality reduction techniques (PCA, autoencoders), multimodal chunking and embedding generation, vector representation, vector databases and similarity search, multimodal fusion strategies: early fusion, late fusion, hybrid fusion, multimodal retrieval systems, retrieval-augmented generation (RAG), integration of multimodal context with Large Language Models (LLMs), and applications of multimodal search and question-answering systems.	**07**
4	**Vision–Language Models:** Introduction to vision–language learning, The Families of VLMs, A Guide to VLM Training, Approaches for Responsible VLM Evaluation, image captioning techniques, visual question answering (VQA), multimodal transformers, contrastive learning, cross-attention mechanisms, vision–language representation learning and applications.	**08**

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

	Extending VLMs to Videos: Early work on Videos based on BERT, enabling text generation using an early-fusion VLM, using a pretrained LLM, Challenges in leveraging video data	
5	**Video, Audio and Multimodal Scene Understanding:** Video representation and temporal modeling, action recognition in videos, audio-visual learning, multimodal emotion recognition, speech and lip-reading systems, multimodal event detection and scene understanding.	**06**
6	**Applications and Emerging Trends:** Multimodal perception in autonomous systems, multimodal medical image analysis, multimodal systems in robotics and human–computer interaction, smart surveillance systems, ethical issues and bias in multimodal AI, future research directions.	**06**

Books Recommended:

Text books:

1. Richard Szeliski, "Computer Vision: Algorithms and Applications", Springer-Verlag London Limited, 2011.
2. Ian Goodfellow, Yoshua Bengio and Aaron Courville, "Deep Learning", MIT Press, Reprint Edition with updates, 2022.
3. Louis-Philippe Morency, "Multimodal Machine Learning", Cambridge University Press, 2023
4. Rafel C. Gonzalez and Richard E. Woods, "Digital Image Processing", 4th Edition, Pearson Education Asia, 2018.

Reference Books:

1. Hassaballah, M., & Awad, A. A., "Deep learning in computer vision: Principles and applications", CRC Press, 2020.
2. S. Kumar and M. H. Kolekar, "Multimodal Machine Learning: Techniques and Applications", Cham, Switzerland Springer, 2020.
3. M. Y. Yang, Y. Guo, and J. Feng, "Multimodal Scene Understanding: Algorithms, Applications and Deep Learning", Cambridge, MA, USA: Academic Press, 2019.

Program: B. Tech **Information Technology** **Final Year** **Semester: VII/VIII**

Course: Agentic AI (DJS23ICPE723)

Pre-requisite: Artificial Intelligence, Machine Learning, Deep Learning

Course Objectives: Discuss the fundamentals of Agentic AI and the frameworks - how they work, how they differ, and how they apply to real world scenario. Address the risks and responsibilities of AI, from deep fakes and misinformation to legal compliance and ethical governance. Enable learners to understand ethical, safe, and responsible agentic AI usage.

Course Outcomes: On successful completion of this course, students should be able to:

1. Apply Large Language Models (LLMs), Generative AI and Agentic AI models for real-world problems.
2. Apply agentic AI frameworks and incorporate ethical, safe, and responsible AI principles when developing agentic solutions.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction Agentic AI:** Generative AI and Foundation Models, Agentic AI, Agentic Workflow, Traditional AI Working vs Agentic AI Working, Generative AI vs Agentic AI, AI Agents vs Agentic AI, Applications. **Transformer Architecture:** Attention Mechanism, Self-Attention, SoftMax, Transformer Positional Encoding. **Embedding & Similarity:** Vector Embedding, Semantic Similarity, Cosine Similarity.	**07**
2	**Large Language Models (LLMs):** Tokens, Context Windows, Balancing Act, Inference vs Training, Challenges of Long Context Windows. **Prompt Engineering:** Zero-shot Prompting, Few-shot Prompting, Chain-of-thought Prompting, Structured Outputs (JSON). **Agentic AI Design Patterns:** ReAct Framework, Task Decomposition, Plan-then-Execute.	**07**
3	**Retrieval-Augmented Generation (RAG):** Document chunking strategies, Retrieval pipelines, Similarity Search, Hybrid Search, Retrieval Probability, Reranking, Context Injection. **RAG with Memory:** Short-Term Memory, Long-Term Memory, Episodic Memory, Vector Memory.	**07**
4	**Agentic AI Architectures:** Types of Agentic Architectures: Single-agent architectures, Multi-Agent Architectures. Single vs Multi-Agent Systems. **Autonomous Agents:** AI Agents vs Autonomous Agents, Working of Autonomous Agents. Feedback Loops: Internal Self-Reflection Loops, Human-in-the-Loop (HITL), Goal-oriented execution, Error recovery, Long-running workflows. **Multi-Agent Systems (MAS):** Architectures of MAS: Reactive Architecture, Deliberative (Cognitive) Architecture, Hybrid Architecture. Agent communication, Role specialization, Role-based agents.	**07**
5	**Agent Evaluation & Monitoring:** Hallucination detection: Hallucination Rate. Benchmarking: Accuracy, Precision, and Recall. Agent Tracing and Observability: Task Completion Rate. Latency Analysis, Reliability Scoring. Tools for Agent Evaluation: Langfuse. Monitoring: AgentOps. **Responsible Agentic AI:** Prompt Injection Attacks, Data Leakage and Privacy, Sandboxing and Access Control, Security Risks and Ethical Challenges of Agentic AI.	**07**

6	**Orchestration Framework:** Model Context Protocol (MCP) Architecture, Single-Agent Frameworks: OpenAI Agents SDK, LangChain. Multi-Agent Frameworks: CrewAI, AutoGen, LangGraph, LlamaIndex. AI Coding Agents: Cursor and Kiro. **Case Studies:** AI customer support agent, AI coding assistant, AI operations assistant, Prior authorization workflows, multi-agent workflow platform.	**07**

Books Recommended:

Textbooks:

1. Dhivya Nagasubramanian, "Agentic AI for Engineers: Architecting Goal-Driven Systems", Apress, 2026.
2. Zt AhBo, "Agentic AI Engineering", ArgoLong Publishing, 2025.
3. Ryan Rad, "The Agentic AI book - From Language Models to Multi-Agent Systems", Leanpub, 2026.
4. Wirtz, "Professor Jochen. Agentic Artificial Intelligence: Harnessing AI Agents to Reinvent Business, Work, and Life", WS Professional**,** 2025.
5. James Phoenix and Mike Taylor, "Prompt Engineering for Generative AI", O'Reilly Media, Inc., 2024.

Reference Books:

1. Sinan Ozdemir, "Building Agentic AI", Addison-Wesly, Hoboken, Pearson Education, Inc., 2026.
2. Devesh Rajadhyax, "Building Generative AI for Enterprise: A practical guide to developing GenAI systems to solve business problems," BPB Publications, May 2026, ISBN-13: 978-9378541353.
3. Kaushik Bar, "The Agentic AI Handbook: Concepts, Design Patterns, and Future Directions", 2024.
4. Pratik Bhavsar, "Mastering AI Agents", Galileo, 2025.
5. Manish Gupta, "Building Applications with AI Agents", BPB Publications, 2025.
6. Chip Huyen, "AI Engineering", O'Reilly Media, Inc., 2024.
7. Paul Iusztin and Maxime Labonne, "LLM Engineer's Handbook: Master the art of engineering large language models from concept to production", Packt Publishing, 2024.
8. Jay Alammar and Maarten Grootendorst, "Hands-On Large Language Models", O'Reilly Media, Inc. 2024.
9. Valentina Alto, "Building LLM Powered Applications: Create intelligent apps and agents with large language models", Packt Publishing, 2024.

Online Resources:

1. Tanmoy Chakraborty and Soumen Chakraborti, IIT Delhi, IIT Bombay. Introduction to Large Language Models (LLMs). https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs88
2. V. Susheela Devi, IISc Bangalore. Artificial Intelligence: Concepts and Techniques. https://onlinecourses.nptel.ac.in/e-learning/preview/noc25_cs159
3. Arijit Sur, IIT Guwahati. Generative AI for Computer Vision. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs104
4. Dr. Anik Mukherjee, IIM Calcutta. NPTEL+ eLEARN. Applications of GenAI and Agentic AI in Business. https://elearn.nptel.ac.in/product-page/applications-of-genai-and-agentic-ai-in-business

Program: B. Tech **Information Technology** **Final Year** **Semester: VII/VIII**

Course: Advanced Security (DJS23ICPE724)

Pre-requisite: Computer Networks, Information Security.

Course Objectives: In this course, students will gain an in-depth knowledge of key security concepts, principles, and technologies essential to modern cybersecurity. Beginning with foundational security principles, the threat landscape and attack vectors. The student will further explore advanced cryptography topics, The Blue Team Principles, ISTQB security testing processes, DFIR, CSIR and SIEM.

Course Outcomes: On successful completion of this course, students should be able to:

1. Analyze different security principles and models.
2. Apply advanced cryptographic algorithms.
3. Design defensive network and endpoint architectures
4. Design Security Operations Centers (SOC) framework to integrate vulnerability management, threat intelligence, and automated response systems.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to Advanced Security:** Overview of Advanced Security Concepts- Security Principles (CIA Triad), Security Models (Bell-LaPadula, Biba), Importance of Security in Modern Infrastructure- Impact of Security Breaches, Case Studies of Major Security Incidents. Threat Landscape and Attack Vectors- Types of Threat Actors (Hacktivists, Nation States, Cybercriminals), Common Attack Techniques (Phishing, Ransomware, DDoS)	**06**
2	**Introduction to Advanced Cryptography:** Modern Cryptographic Algorithms- Symmetric vs. Asymmetric Cryptography. Quantum Cryptography- Quantum Key Distribution (QKD), Post-Quantum Cryptography. Cryptographic Protocols and Standards- Secure Multiparty Computation Homomorphic Encryption.	**08**
3	**Blue Team Principles: Defensive network** – architecture and monitoring, Endpoint Architecture and Monitoring, Security Data Locations, Authentication, Authorization, and Accounting, Defending Network Infrastructure, Intrusion Prevention Systems and Firewalls, Name Resolution Attacks and Defence, Securing Private and Public Cloud Infrastructure.	**07**
4	**Security Operations Center (SOC):** What is a SOC, SOC types, Security Operations (SecOps), The People, Processes, and Technology, Vulnerability Management, Automation, Improvement, and Tuning, Security Operations Framework – SOC elements and processes, Security Operations Infrastructure and Automation, Threat Prevention and Intelligence, Introduction to Network Operation Centre (NOC), NOC v/s SOC.	**07**
5	**Security Testing Processes:** ISTQB Security Testing Processes, Security Test Planning, Security Test Design, Security Testing Throughout the Software Lifecycle, Testing Security Mechanisms -System Hardening, Authentication and Authorization, Encryption, Malware Scanning, Data Obfuscation.	**07**
6	**Digital Forensics:** Definition and scope of digital forensics, Importance of digital forensics in criminal investigations, Laws and regulations affecting digital forensics, Ethical issues in digital investigations, Steps in a digital forensic investigation, Overview of the digital forensics process Active Defence, DFIR Core Concepts, Digital Forensics and Incident Response, Modern DFIR Cyber Security Incident Response - CSIR Plan, CSIR Models, SIEM, SIEM Architecture, SIEM – Logging, Evaluation, Analytics, Detection, Cyber-crime and Cyber law.	**07**

Books Recommended:

Textbooks:

1. Charles P. Pfleeger, Shari Lawrence Pfleeger, and Jonathan Margulies, “Security in Computing”, Prentice Hall Professional, 2023.
2. William Stallings, “Cryptography and Network Security: Principles and Practice”, Pearson India, 2023.
3. Michael A. Nielsen and Isaac L. Chuang, “Quantum Computation and Quantum Information”, Cambridge, 2012.
4. Joseph Muniz, Gary McIntyre and Nadhem AlFardan, “Security Operations Center: Building, Operating, and Maintaining your SOC”, Pearson Education, 2015.

Reference Books:

1. Chris Sanders and Steve Smith, Alan J White and Ben Clark, “Practical Network Security: Monitoring, Policy, and Incident Response”, Blue Team Field Manual, 2017.
2. Rex Black, “Advanced Software Testing - Vol. 3: Guide to the ISTQB Advanced Certification as an Advanced Technical Test Analyst”, Rocky Nook, 2014.

Program: B. Tech **Information Technology** **Final Year** **Semester: VII/VIII**

Course: Edge Computing (DJS23ICPE725)

Pre-requisite: Cloud Computing.

Course Objectives: To develop a comprehensive understanding of edge computing concepts, architectures, and resource management strategies, while addressing security and privacy challenges across diverse domains. It focuses on enabling students to apply edge computing models and infrastructure for real-world problem solving.

Course Outcomes: On successful completion of this course, students should be able to:

- Apply edge computing principles and paradigms across diverse networking architectures.
- Analyze edge intelligence, data analytics, and resource optimization in edge computing environments.
- Assimilate challenges, privacy issues, and practical applications of edge computing across various domains to determine appropriate deployment strategies.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Fundamentals of Edge Computing:** The Evolutionary History, Mobile Edge Computing, Fog Computing, Cloud-Sea Computing, Rapid Growth Period, Intelligence Integration Period. The Basic Concept and Key Characteristics of Edge Computing, Edge Computing vs. Cloud Computing, Advantages and Challenges of Edge Computing.	08
2	**Architecture and Components of Edge Computing:** Edge Infrastructure, Introduction to Edge Computing Architecture, Different Grades/Layers of Edge, Capabilities of Edge Infrastructure, New Progress of Edge Computing Architecture, Edge Computing Models, Overview and Definitions, Collaborative Edge Computing Models, Choosing the Right Model, Networking in Edge Computing, Introduction and Development Process of Edge Computing-Network Integration, Edge Computing-Network Architectures.	07
3	**Edge Intelligence:** Formal Definition, Hardware and Software Support, Technologies Enabling Edge Intelligence, Edge Intelligent System Design and Optimization. **Edge Data Analytics:** Architecture of Edge Analytics, Machine Learning for Edge Devices, Edge Analytics: Case Study	07
4	**Resource management in edge computing:** Resource management in edge computing and its challenges. Resource management techniques for edge computing, including task scheduling algorithms, resource allocation algorithms, and load balancing algorithms. Case studies and applications of resource management in edge computing, such as mobile edge computing, and autonomous vehicles.	08
5	**Privacy and Bias in Edge Computing: Privacy** in Edge Computing, Accessibility and Digital Divide. **Challenges and Solutions in Edge Computing:** Programmability and Data Management, Security, Privacy, and Service Management, Deployment Strategies and Integration, Foundations and Business Models.	06
6	**Case Studies and Practical Applications:** Manufacturing, Telecommunications, Healthcare, Smart Cities, Internet of Things (IoT), Retail, and Autonomous Vehicles. **Cloud-Based Edge Infrastructure Deployment using Edge Zones and Content Delivery Networks (CDNs),** Realization of Edge Computing in Healthcare and Storage Security in Edge Environments.	06

Books Recommended:

Text books:

1. Lanyu Xu, Weisong Shi, "Edge Computing: Systems and Applications", 1st Edition, John Wiley & Sons, 2025.
2. K. Anitha Kumari, G. Sudha Sadasivam, D. Dharani, M. Niranjanamurthy, "Edge Computing: Fundamentals, Advances and Applications", 1st Edition, CRC Press (Taylor & Francis Group), 2022.
3. Ajit Singh, "Edge Computing", 1st Edition, Ajit Singh Publications, 2024.

Reference Books:

1. Taheri, Javid, and Shuiguang Deng (Eds.), "Edge Computing: Models, Technologies and Applications", 1st edition, Institution of Engineering and Technology (IET), 2020.
2. Xu, Lanyu, and Weisong Shi, "Artificial Intelligence and Machine Learning for Edge Computing", CRC Press, 2024.
3. Fadi Al-Turjman, "Edge Computing: From Hype to Reality", Springer International Publishing, 2018.
4. Cao, Jie, et al., "Edge Computing: A Primer", Springer International Publishing, 2018.

Web Resources:

1. https://nptel.ac.in/courses/106104449
2. https://www.coursera.org/learn/security-at-the-edge-first-course-1
3. https://www.mindluster.com/course/11921/Edge-computing

Program: B. Tech **Information Technology** **Final Year** **Semester: VII/VIII**

Course: Deep Learning (DJS23ICMD701)
Course: Deep Learning Laboratory (DJS23ILMD701)

Pre-requisite: Machine Learning

Course Objectives: The course will familiarize students with the concept of deep learning and enable them to assimilate to different types of neural networks. The course also aims to expose students to the idea of auto-enders.

Course Outcomes: On successful completion of this course, students should be able to:

1. Apply suitable deep learning models for real-world problems.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Fundamental of Deep Learning:** Overview: Machine Learning and its types, Evolution of Deep Learning and its Applications. Fundamentals of Neural Networks: Introduction to Biological neuron, NN architecture, perceptron, multilayer perceptron (MLP). Activation Functions: Step function, sigmoid, tanh, ReLU (Rectified Linear Unit) functions, leaky ReLU. Learning and Loss Functions: cost functions, cross-entropy loss, mean squared error.	06
2	**Supervised Learning with Feedforward Neural Networks:** Feedforward Neural Networks: Architecture, forward pass, weight initialization problem, vanishing gradient problem, exploding gradient problem. Backpropagation: weights adjustment, chain rule of differentiation, Optimization Techniques: Stochastic Gradient Descent (SGD), Momentum-based SGD, RMSProp, Adam Optimizer, learning rate scheduling, adaptive learning rates.	07
3	**Regularization and Model Tuning:** Overfitting and Underfitting: Diagnosing and addressing overfitting and underfitting in models. Regularization Techniques: L2 regularization (Ridge), L1 regularization (Lasso), Elastic Net, Dropout, Batch Normalization, Weight Regularization. Model Tuning: Hyperparameter tuning, Grid Search, Random Search, manual tuning, selecting optimal parameters for deep learning models. Cross-Validation: K-fold cross-validation, leave-one-out cross-validation, training-validation-test split, data leakage.	05
4	**Convolutional Neural Networks (CNN):** CNN Architecture: Layers in CNN, convolutional layers, pooling layers (max pooling, average pooling), fully connected layers, flattening. Convolution Operations: Filters (kernels), strides, padding (same vs valid), dilation, depth. Feature Extraction: feature maps, edge detection, hierarchical feature learning. Popular CNN Architectures: AlexNet,VGGNet, GoogLeNet/Inception, ResNet Applications: Image classification, object detection (YOLO, R-CNN), face recognition, style transfer.	10
5	**Sequence Modelling:** Recurrent Neural Networks (RNN) Architecture: Sequential data handling, recurrence, shared weights, time steps, and memory cells. Backpropagation Through Time (BPTT): Error backpropagation over multiple time steps. Variants of RNN: Long Short-Term Memory (LSTM), Gated Recurrent Units (GRUs). Sequence-to-Sequence Learning Applications: Time series forecasting. Natural Language Processing (NLP): sentiment analysis, text generation, language modelling, machine translation.	07
6	**Unsupervised Learning and Dimensionality Reduction:** Kohonen Self-Organising Feature Maps- Architecture, training algorithm, Kohonen Self Organising Motor Map. Autoencoders: Compression and reconstruction, bottleneck layer, undercomplete and overcomplete autoencoders. Variational Autoencoders (VAE): Probabilistic approach, latent space representation, generating new samples.	07

List of Laboratory Experiments:

1. Implement a single-layer perceptronrom scratch and train it to solve a simple binary classification problem.
2. Explore different activation functions (Sigmoid, ReLU, Tanh) and their impact on model performance.
3. Implement backpropagation manually for a small neural network and observe the weight update process.
4. Explore the effect of different optimization algorithms (SGD, Adam) and hyperparameters on the convergence of a neural network.
5. Implement a basic CNN model for image classification using a dataset like CIFAR-10 or MNIST.
6. Use a pretrained CNN model (AlexNet, VGGNet, or GoogLeNet) for image classification.
7. Implement a simple RNN for time series prediction (e.g., stock price prediction).
8. Implement an LSTM or GRU model for text generation based on a character-level dataset.
9. Build an autoencoder and use it for dimensionality reduction or noise removal on the MNIST dataset.
10. Use transfer learning for text classification tasks (e.g., sentiment analysis) with pre-trained models.

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

Books Recommended:

Textbooks:

1. John Krohn, Grant Beyleveld and Aglae Bassens, "Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence", Pearson, 2022.
2. Francois Chollet, "Deep Learning with Python", 2nd Edition, Manning Publications, 2018.
3. Josh Patterson and Adam Gibson, "Deep Learning", O'Reilly Media, Inc., 2017.
4. Umberto Michelucci, "Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection", Apress, 2019.
5. Simon Haykin, "Neural Networks and Learning Machines", 3rd Edition, Pearson Prentice Hall, 2010.

Reference Books:

1. Yegnanarayana, B., "Artificial Neural Networks", PHI Learning Pvt. Ltd, 2009.
2. Satish Kumar, "Neural Networks: A Classroom Approach", Tata McGraw-Hill Education, 2004.
3. Raul Rojas, "Neural networks: A Systematic Introduction", Springer Science & Business Media, 2013.
4. David Foster, "Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play", O'Reilly Media, 2019.

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 & 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:
	  ▪ Research paper
	  ▪ Product/Startup idea
	Activities:
	• Convert Industry/ Societal problem into research problem.
	• Analyze research articles and extract research gaps.
		05

2	**Experimental Design, Tools & Validation Techniques** • Advanced research methodologies: ▪ Experimental, simulation, data-driven, design thinking	**06**

	• 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).	
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.	**05**
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.	**06**
5	**Integrated Research, Proposals 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 • 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	**06**

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)

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:

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

- Apply technical knowledge gained from previous courses to identify real-life, industry, or societal problems and design appropriate solutions.
- Demonstrate the technical skills required to design, develop, build, and test engineering systems, products, software, or hardware using modern engineering and IT tools.
- 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.
- Develop and demonstrate the ability to test, validate, debug, and troubleshoot engineering systems (including software, hardware, and process components) and analyse the results.
- 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 students' 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

Assessment Criteria (mapped CO)	Max	Excellent	Satisfactory	Below Exp.
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.

Criteria	Excellent	Satisfactory	Below Expectation
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, 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.	Procedures mostly followed with occasional oversight. Needs some guidance in selecting tools / equipment and is tentative in operating them. Aware of errors but accounts for them only minimally.	Does not follow a proper procedure. Cannot select appropriate tools / equipment or operates them incorrectly, 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	A paper draft or IPR document is prepared and largely follows the required format. Novelty	No paper / IPR document, or a weak draft that does not follow the required format. Novelty or

Criteria	Excellent	Satisfactory	Below Expectation
	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.	and contribution are stated but only partially developed, with minor gaps in supporting evidence or formatting.	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.

Course: DevOps and AI on AWS Specialization (DJS23ICSC701)

Coursera Link: https://www.coursera.org/specializations/devops-ai-aws

Number of hours: 26 Hrs., Beginner level

About the Course:

This course provides a comprehensive introduction to Generative AI, DevOps, and AIOps for developing and managing intelligent cloud-native applications on AWS. Learners explore the fundamentals of Generative AI, Amazon Bedrock, prompt engineering, and AI-assisted application development.
The course covers modern DevOps practices, including Continuous Integration and Continuous Deployment (CI/CD), infrastructure automation, and deployment strategies for building reliable and scalable software systems. It also introduces monitoring and observability techniques using cloud-native tools to enhance application performance and reliability.
Finally, learners explore AIOps concepts such as anomaly detection, predictive analytics, event correlation, and automated incident response. By the end of the course, participants will be able to develop AI-powered applications, automate software delivery workflows, and implement intelligent operational practices in cloud environments.

Course Outcomes: On successful completion of this course, student should be able to

1. Apply generative AI concepts, Amazon Bedrock services, and prompt engineering techniques to develop AI-powered cloud applications.
2. Design and implement DevOps practices, including Continuous Integration and Continuous Deployment (CI/CD), for automated software development and delivery.
3. Utilize Infrastructure as Code (IaC), monitoring, and observability tools to automate cloud infrastructure management and enhance application reliability.
4. Analyze and implement AIOps techniques for intelligent monitoring, anomaly detection, predictive analytics, and automated incident response in cloud environments.

Detailed Syllabus:

Unit	Description	Duration
1	**Generative AI Application Development with Amazon Bedrock** Introduction to Generative AI and Large Language Models (LLMs), Overview of AI-Enhanced Applications, Amazon Bedrock Architecture and Services, Prompt Engineering Techniques, Amazon Q Developer for AI-Assisted Development, Integration of Generative AI Features into Applications, Application Upgrade and Deployment Workflow, Amazon Bedrock Model Customization Concepts, Responsible AI Principles and Practices, Case Study: AI-Powered TravelGuideAI Application.	**06**
2	**DevOps Practices and CI/CD Automation for AI Applications** Introduction to DevOps and DevOps Lifecycle, DevOps for Generative AI Applications, Version Control and Branching Strategies, Continuous Integration (CI) Concepts and Practices, Continuous Deployment (CD) Strategies, Automated Build, Testing and Deployment Workflows, CI/CD Pipeline Design and Implementation, Application Release Management, Deployment Automation using Amazon CodeDeploy, Reliability Engineering and Operational Excellence, Continuous Release and Delivery Practices.	**07**

3	**Infrastructure Automation, Monitoring and Observability** Infrastructure as Code (IaC) Fundamentals, AWS Cloud Development Kit (AWS CDK), Automated Infrastructure Provisioning, Cloud-Native Application Deployment, Monitoring and Observability Fundamentals, Metrics, Logs and Traces, Amazon CloudWatch and CloudWatch Anomaly Detection, Distributed Tracing using AWS X-Ray Insights, AI-Assisted Monitoring and Alerting, Performance Analysis and Optimization, Reducing Deployment Risks and Human Errors.	**07**
4	**AIOps for Intelligent Operations and Automated Remediation** Introduction to AIOps, AIOps Architecture and Frameworks, Machine Learning for IT Operations, Operational Challenges in Large-Scale Cloud Environments, Intelligent Monitoring and Observability, Log Analytics and Operational Intelligence, Anomaly Detection Techniques, Event Correlation and Root Cause Analysis, Predictive Analytics for Incident Prevention, AI-Driven Alert Management, Automated Incident Response and Remediation, Self-Healing Systems, Performance Optimization using AIOps, Case Study: AIOps for Cloud-Native Applications.	**06**
	Total	**26**

Continuous Assessment (B)

Evaluation Scheme:

1. The identified courses shall focus on developing AI-related competencies and skills among students. Each selected course shall carry one (1) credit.
2. Faculty members shall act as mentors for the Coursera courses identified for various semesters of the program.
3. One faculty mentor shall be assigned to each division of the program for effective coordination and monitoring of this activity.
4. The assigned faculty mentor shall complete and review the course content before assigning the course to students.
5. The term work for these one-credit courses shall form a part of the Internal Continuous Assessment (ICA)component.
6. The Term work (ICA) component shall carry **40 marks**

The distribution of marks for term work shall be as follows:

- **15 Marks**: Coursera Course Completion and Certification
- **15 Marks**: Faculty-curated quiz conducted through the Coursera/Any other platform
- **10 Marks**: Internal faculty evaluation through quizzes, assignments, presentations, or project

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.

Detailed Syllabus: (unit-wise)

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). • Role of growing population and subsequent industrialization, urbanization and changing lifestyle of human beings in frequent occurrences of manmade disasters. • Significance of studying various aspects of disasters - magnitude, frequency, effects of disasters, India's vulnerability to disasters, and the impact of disasters on National development.	**08**
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. • Structural & Non-Structural Mitigation: Disaster-safe designs, land-use planning. • Science & Technology: Use of information management, Geo informatics like RS, GIS, GPS and remote sensing mitigation measures. • Do's and don'ts in case of disasters and effective implementation of relief aids	**08**
3	**Disaster Preparedness & Response:** • Preparedness Planning, Early Warning Systems (EWS), & Communication. • Emergency Response: Search & Rescue, Logistics, Medical Aid. • Psychological Response & Management (Trauma, Stress).	**08**

	• Role of IT, Media, Govt., NGOs, & Community. Scope and responsibilities of National Institute of Disaster Management (NIDM) and National disaster management authority (NDMA) in India.	
4	**Recovery, Rehabilitation & Reconstruction:** • Post-disaster damage assessment. • Rehabilitation, Reconstruction, & Livelihood Restoration. • Sanitation, Hygiene, & Waste Management. Ways to raise finance for relief expenditure, role of government agencies and NGO's in this process, Legal aspects related to finance raising as well as overall management of disasters	**06**
5	**Policy, Governance & Capacity Building:** • National Disaster Management Authority (NDMA) & Legislation. • Institutional Mechanisms & Community Mobilization. Non-Structural Mitigation: Community based disaster preparedness, risk transfer and risk financing, capacity development and training, awareness and education, contingency plans.	**06**
6	**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	**06**
	Total Number of Hours	**42**

Course: Biomedical Informatics (DJS23ICPE811)

Pre-requisite: Database Management Systems, Basic Statistics, Familiarity with Machine Learning or Data Analytics concepts is desirable.

Course Objectives: The course aims to introduce the concepts of biomedical and health informatics along with healthcare systems. It helps students understand healthcare data management, information systems, and interoperability standards. It also enables students to explore healthcare analytics, artificial intelligence techniques, and emerging digital health technologies.

Course Outcomes: On successful completion of this course, students should be able to:

1. Explain biomedical informatics concepts, healthcare systems, and data management practices.
2. Apply interoperability standards, data exchange mechanisms, and system design principles in healthcare.
3. Analyze healthcare data using analytics and AI techniques and evaluate digital health technologies and applications.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Foundations of Biomedical Informatics:** Introduction to biomedical and health informatics, types including biomedical, health and clinical informatics, evolution of healthcare information systems, healthcare delivery systems and stakeholders, healthcare data types and DIKW hierarchy, clinical workflows and overview of patient lifecycle.	04
2	**Healthcare Information Systems:** Healthcare data lifecycle, Electronic Health Records (EHR) and EMR, structure and components of EHR systems, Hospital Information Systems (HIS) architecture, clinical databases and healthcare repositories, digital workflows in hospitals, data quality issues in healthcare, Case study of end-to-end hospital workflow.	07
3	**Healthcare Applications and System Design:** Telemedicine systems and architecture, mobile health (mHealth) applications, patient portals and consumer health systems, remote healthcare delivery models, Human-Computer Interaction (HCI) in healthcare, usability and user-centered design principles, challenges in healthcare interfaces, Case study on usability issues in healthcare systems.	08
4	**Healthcare Data Standards and Interoperability:** Need for interoperability in healthcare systems, types including syntactic and semantic interoperability, Health Information Exchange (HIE), medical coding systems such as ICD and SNOMED, HL7 standards, Fast Healthcare Interoperability Resources (FHIR), REST APIs and data formats such as JSON and XML, integration of healthcare applications, data exchange workflows and challenges, case study on inter-hospital data exchange.	10
5	**Healthcare Data Analytics and Decision Support:** Introduction to healthcare data analytics, data sources in healthcare systems, data preprocessing concepts, machine learning for disease prediction, Clinical Decision Support Systems (CDSS), Natural Language Processing (NLP), medical image informatics and AI applications, Computer-Aided Diagnosis (CAD), evidence-based medicine, Case Study on AI-based disease prediction and clinical decision support system using healthcare data analytics, machine learning, and medical image analysis.	08

6	**Digital Health Systems and Future Trends:** Internet of Medical Things (IoMT) and wearable devices, remote patient monitoring systems, cloud computing in healthcare, big data in healthcare systems, blockchain in healthcare data management, privacy and security including HIPAA concepts, healthcare quality, safety and outcome measurement, ethical and legal issues in biomedical informatics, Case Study on Smart remote patient monitoring system using IoMT wearables, cloud computing, and secure healthcare data management.	**05**

Books Recommended:

Textbooks:

1. Robert E. Hoyt and Ann Yoshihashi, "Health Informatics: Practical Guide for Healthcare and IT Professionals", 7th Edition, Lulu Press, 2019.
2. Edward H. Shortliffe and James J. Cimino, "Biomedical Informatics: Computer Applications in Health Care and Biomedicine", 4th Edition, Springer, 2014.

Reference Books:

1. H. K. Huang, "Biomedical Information Technology", Academic Press, 2010.
2. R. S. H. Istepanian, S. Laxminarayan and C. S. Pattichis, "M-Health: Emerging Mobile Health Systems", Springer, 2006.
3. Kevin Beaver, "Healthcare Informatics: Improving Efficiency and Productivity", McGraw-Hill Education, 2016.
4. J. Mantas, A. Hasman and M. S. Househ, "Health Informatics: An Interdisciplinary Approach", IOS Press, 2017.

Coursera Courses:

1. Health Informatics, Johns Hopkins University, Coursera
2. AI for Medicine Specialization, DeepLearning.AI, Coursera

Program: B. Tech **Information Technology** **Final Year** **Semester: VII/VIII**

Course: Agent Modeling in Finance (DJS23ICPE812)

Pre-requisite: Basic programming, probability/statistics, and introductory economics/finance concepts.

Course Objectives: To develop a comprehensive understanding of Agent-Based Modelling including its behavior, learning mechanisms, and system dynamics. It also enables learners to Analyze and evaluate financial markets and heterogeneous agent models.

Course Outcomes: On successful completion of this course, students should be able to:

1. Analyze the fundamental principles of Agent-Based Modelling and their impact on system behavior.
2. Investigate financial market structures and heterogeneous agent models to interpret complex phenomena.
3. Design and implement agent-based simulation frameworks to develop solutions for real-world financial problems

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Foundations of ABM:** Introduction to Agent based Modelling (ABM), ABM Features, agents, environment, interactions, bounded rationality, ABM as recursive Systems, Rationality, Behavior, and Expectations of ABM, model assumptions, parameter selection.	06
2	**Agents' Learning and Interaction:** Agents' Behavior and Heuristics, Agents' Adaptation and Learning- Statistical, Fitness, Social, Individual vs. Social Learning, Modelling Interaction, Interaction Networks, static and Dynamic Networks, Empirical Validation of ABM.	06
3	**Financial Markets and Market Microstructure:** Structure of financial markets, traders and institutions, order books, price formation, market efficiency, stylized facts in finance, returns and volatility, limitations of classical models.	06
4	**Heterogeneous Agent Models:** Need for Heterogeneity, Taxonomy, Brock-Hommes Model (Core Framework), Lux-Marchesi Model (Herding & Opinion), Zero-intelligence traders, adaptive market hypothesis, asset pricing models, bubbles and crashes, volatility clustering.	06
5	**Simulation and Implementation:** Agent based Simulation workflow, Initialization, calibration, and parameter tuning, sensitivity analysis, and computational aspects of ABM.	09
6	**ABM Application and Policy Analysis:** ABM in stock markets and portfolio management, Systemic risk modelling, Policy design and impact analysis, ABM for behavioral finance and Crisis modelling.	09

Case Studies suggested:

1. Panic Selling During a Market Crash
2. Retail Investors Influenced by Social Media
3. High-Frequency Trading and Price Discovery
4. Cryptocurrency Bubble Formation
5. The 2008 Global Financial Crisis
6. Adaptive Traders in a Volatile Market
7. Brock-Hommes Model and Asset Price Dynamics
8. Impact of Circuit Breakers on Market Stability
9. Zero-Intelligence Traders vs. Rational Traders
10. Portfolio Management Under Behavioural Biases

Books Recommended:

Textbooks:

1. Domenico Delli Gatti, Giorgio Fagiolo, Mauro Gallegati, Matteo Richiardi and Alberto Russo, "Agent-Based Models in Economics a Toolkit", Cambridge University Press, 1997.
2. Giovanni Gallegati et al., "Introduction to Agent-Based Economics", Elsevier, 2017.
3. Blake LeBaron, "Agent-based Computational Finance in the Handbook of Computational Economics", Brandeis University, 2005.
4. Shu-Heng Chen and Mak Kaboudan, "Handbook of Computational Economics", Elsevier, 1996.
5. Nigel Gilbert, "Agent-Based Models", SAGE Publications, 2008.

Reference Books:

1. Nigel Gilbert, "Agent-Based Models", SAGE Publications, 2008.
2. Eric Bonabeau and Nigel Gilbert, "Agent-Based Modeling and Simulation", Springer, 2017.
3. Guenter Iori, Giulia I. B. et al., "Agent-Based Modelling for Financial Markets", Springer, 2012.

Program: B. Tech **Information Technology** **Final Year** **Semester: VII/VIII**

Course: Language Modeling (DJS23ICPE813)

Pre-requisite: Artificial Intelligence, Deep Learning.

Course Objectives: The course aims to introduce fundamental concepts of language modeling, including tokenization, probability, and N-gram models, while developing an understanding of neural language models and transformer architectures, and providing knowledge of fine-tuning, alignment techniques, and real-world applications of large language models.

Course Outcomes: On successful completion of this course, students should be able to:

1. Apply language modeling techniques for real-world Natural Language Processing applications.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to Language Modeling:** Definition and Purpose of Language Modeling (LM), Modelling process, Linguistic unit (text, sentence, token, symbol), Tokenization, Text Probability, Sentence Probability, N-gram Models, Neural Models, LM vs LLM. Hallucination, bias mitigation, and ethical considerations.	05
2	**N-gram LMs:** N-gram Definition and Types, Process, Markov property, Smoothing, Generation. Evaluating Language Models: Cross-Entropy and Perplexity, Cross-Entropy and Log-Likelihood of a Text.	07
3	**Large Language Models:** Token Embeddings and Contextual Representations, Vocabulary Construction and Subword Tokenization, Pre-training Objectives: Causal Language Modelling (CLM) and Masked Language Modelling (MLM), Training Corpus and Data Preparation, Scaling Laws, Emergent Abilities, In-Context Learning, Evolution from Neural Language Models to Large Language Models.	07
4	**Transformer Architectures:** Attention Mechanisms: Self-attention, multi-head attention, and positional encoding. Architecture Types: BERT (BERT Base and BERT Large), GPT, LLama2.	08
5	**Fine-Tuning:** Transfer Learning & Adaptation: Fine-tuning methods, Parameter-Efficient Fine-Tuning (PEFT), and LoRA (Low-Rank Adaptation). Model Optimization: Model distillation, pruning, and quantization for efficiency. Large-Scale LLM Training: Distributed training, data parallelism, model parallelism, pipeline parallelism, GPU clusters, and challenges in large-scale pre-training.	07
6	**Alignment and Applications:** Prompting and Reasoning: Zero-shot/few-shot prompting, Chain of Thought (CoT), and Tool Use (agents). Retrieval-Augmented Generation (RAG): Connecting LLMs to external data sources and vector databases. Human Alignment: Reinforcement Learning from Human Feedback (RLHF) and Direct Preference Optimization (DPO).	08

Books Recommended:

Textbooks:

1. Daniel Jurafsky and James H. Martin "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models", 3rd Edition, Pearson Education**,** 2026.
2. Steven Bird, Ewan Klein and Edward Loper, "Natural Language Processing with Python", O'Reilly Media, 2009.
3. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana, "Practical Natural Language Processing", O'Reilly Media, 2020.

Reference Books:

1. Christopher D. Manning and Hinrich Schütze, "Foundations of Statistical Natural Language Processing", MIT Press, 1999.
2. Yoav Goldberg, "Neural Network Methods for Natural Language Processing", Morgan & Claypool, 2017.
3. Denis Rothman, "Transformers for Natural Language Processing", 2nd Edition, Packt Publishing, 2021.
4. Tong Xiao and Jingbo Zhu, "Foundations of Large Language Models", Springer, 2024.

Program: B. Tech **Information Technology** **Final Year** **Semester: VII/VIII**

Course: Nature Inspired Learning Algorithms (DJS23ICPE814)

Pre-requisite: Algorithm Design and Analysis, Artificial Intelligence.

Course Objectives: Bio-inspired algorithms and computation have become increasing popular in the last two decades. The algorithms such as ant colony algorithms, firefly algorithms and particle swarm optimization have been applied in almost every area of science and engineering. This course covers novel algorithms to solve real-life, complex problems, combining well-known nature inspired algorithms with new concepts.

Course Outcomes: On successful completion of this course, students should be able to:

1. Identify the natural phenomena inspiring the given algorithm.
2. Apply appropriate nature-inspired algorithms to solve the given problem.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Nature Inspired Learning Algorithms:** From Nature to Computing: philosophy, models, metaphors, Characteristics: Adaptation, self-organization, emergence, feedback, Agents, parallelism, distributed intelligence, Optimization basics: single-objective vs multi-objective optimization, Heuristics, metaheuristics, hyper-heuristics, Fitness landscape, exploration vs exploitation.	06
2	**Genetic Algorithms:** Basic Principles of Genetics, Fitness Function; Selection: Selective Pressure, Random Selection, Proportional Selection, Tournament Selection, Rank-Based Selection, Boltzmann, Convergence and schema theorem, Differential Evolution (DE), Overview of Evolutionary Programming & Genetic Programming, Genetic algorithms - Mathematical foundation, Genetic problem solving, crossover and mutation. Genetic algorithms and Markov process, applications of genetic algorithms Applications and limitations.	07
3	**Swarm Intelligence and Ant Colony Optimization:** Particle Swarm Optimization: Basic Particle Swarm Optimization: Global Best PSO, Local Best PSO, Velocity Components; Basic PSO parameters, Single Solution Particle Swarm Optimization: Guaranteed Convergence PSO, Social-Based Particle Swarm Optimization, Hybrid Algorithms, Sub-Swarm Based PSO, Multi-Start PSO Algorithms, Repelling Methods, Binary PSO; Application.	08
4	**Ant Algorithm:** Simple Ant Colony Optimization, Ant Colony Optimization Meta-Heuristic, Cemetery Organization and Brood Care, Division of Labor, Artificial Bee Colony algorithms - ABC basics, ABC in optimization, multi-dimensional bee colony algorithms, hybrid ant system, ACO in combinatorial optimization, variations of ACO, applications of bee algorithms, application in solving Travelling Salesman Problem.	07
5	**Modern Metaheuristic Optimization Algorithms:** Selected nature inspired techniques - Hill climbing, simulated annealing, Gaussian adaptation, Cuckoo search, Firefly algorithm, SDA algorithm, bat algorithm, case studies. Other nature inspired techniques - Social spider algorithm, Cultural algorithms, Harmony search algorithm, Intelligent water drops algorithm, Flower pollination algorithm, Bacterial Foraging Optimization (BFO), Biogeography-Based Optimization (BBO), Cultural Algorithms & Opposition-Based Learning case studies. Applications and case studies.	08
6	**Artificial Immune Models:** Natural Immune System: Classical view, Antibodies and Antigens, White Cells, Immunity types, Network Theory, Danger Theory; Artificial Immune Models: Artificial Immune system algorithm, classical view models, Clonal Selection Theory: CLONALG; Network Theory Models; Danger Theory Models; Application: Intrusion	06

	Detection, introduction to DNA Computing (concept + Adleman experiment), Bone Marrow Model. Hybrid algorithms (GA+PSO, ACO+local search, etc.), Performance metrics: convergence, diversity, scalability, Parameter tuning and sensitivity analysis, Recent trends and research directions.	

Books Recommended:

Textbooks:

1. Leandro Nunes de Castro, "Fundamentals of Natural Computing: Basic Concepts, Algorithms and Applications", Chapman & Hall/CRC Press, 1st Edition, 2007
2. Andries P. Engelbrecht, "Computational Intelligence: An Introduction", Wiley, Latest available edition: 2nd Edition, 2007 (reprints up to 2023 available)
3. Dan Simon, "Evolutionary Optimization Algorithms", Wiley, 1st Edition, 2013.

Reference Books:

1. D. Floreano and C. Mattiussi, "Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies", MIT Press, 2008.
2. Russell C. Eberhart, Yuhui Shi and James Kennedy, "Swarm Intelligence: The Morgan Kaufmann Series in Evolutionary Computation", 1st Edition, Morgan Kaufmann Publishers, 2001.
3. Sam Jones (Editor), "Bio Inspired Computing-Recent Innovations and Applications", 2nd Edition, Clanrye International, 2015.
4. Yang Xiao (Editor), "Bio-Inspired Computing and Networking", CRC Press, 2016.
5. Xin-She Yang, "Nature-Inspired Optimization Algorithms", 1st Edition, Elsevier, 2014.
6. Marco Dorigo and Thomas Stützle, "Ant Colony Optimization", 1st Edition, MIT Press, 2004
7. Albert Y. Zomaya, "Handbook of Nature-Inspired and Innovative Computing", 1st Edition, Springer, 2006.

Program: B. Tech **Information Technology** **Final Year** **Semester: VII/VIII**

Course: Quantum Computing (DJS23ICPE815)

Pre-requisite: Statistics for Data Science, and Machine Learning.

Course Objectives: To equip students with fundamental and applied knowledge of quantum computing, including quantum principles, algorithms, cryptography, error correction, and quantum machine learning for solving computational problems.

Course Outcomes: On successful completion of this course, students should be able to:

1. Understand basic concepts of quantum computing, qubits, and quantum phenomena.
2. Analyze quantum gates, circuits, and compare classical and quantum computation.
3. Apply quantum algorithms, cryptography, and error correction techniques.
4. Understand NISQ systems and basic concepts of quantum machine learning and deep learning.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Mathematical Foundations:** Complex Numbers and Conjugation, Vectors, Vector Space, Basic set, Dirac Notation, Ket and Bra, Inner Product, Linearly Dependent and Independent Vectors, Dual Vector Space, Qubit representation**.**	06
2	**Quantum Computing vs. Classical Computing:** History of quantum computation and quantum information, Quantum State, Bloch sphere, Physical quantum phenomena: Spin, Quantum superposition, Interference and Entanglement. Logic Gates and Circuits: Boolean Algebra, Logic Gates, Circuits, Universal Gates, Gates and Computation Quantum Gates and Circuits: Qubits, Single-qubit gates (Pauli, Hadamard), Multi-qubit gates (CNOT, SWAP, Toffoli), Quantum circuits, No-cloning theorem, Bell states, Quantum teleportation.	10
3	**Quantum Computing algorithms:** Classical computations on a quantum computer, Quantum parallelism, Quantum key distribution, Superdense coding, quantum teleportation, applications of teleportation, probabilistic versus quantum algorithms, phase kick-back, Quantum phase estimation and quantum Fourier Transform, eigenvalue estimation, Shor's Factorization Algorithm, Grover's Search Algorithm, Quantum algorithms summarized.	08
4	**Quantum Cryptography algorithm:** Cryptography using principles of quantum computing, No-cloning theorem, Quantum key distribution Algorithm, Quantum secret sharing Algorithm.	05
5	**Quantum error correction and NISQ:** Quantum noise and decoherence, Error correction basics, Shor code, Steane code, Syndrome measurement, Fault tolerance, NISQ computing, Noise models, Error mitigation.	06
6	**Quantum Machine Learning & Deep Learning:** Variational Quantum Circuits (VQC), Parameterized Quantum Circuits, Data encoding methods: Quantum variational classifier, Quantum Support Vector Machine, Quantum Neural Networks (QNN), Quantum Convolutional Neural Networks (QCNN), Training challenges: Barren plateaus (vanishing gradients), Practical training issues.	07

Books Recommended:

Textbooks:

1. Parag K. Lala, "Quantum Computing", 1st Edition, McGraw Hill, 2020.
2. Chris Bernhardt, "Quantum Computing for Everyone", 1st Edition, MIT Press, 2020.

Reference Books:

1. Jack D. Hidary, "Quantum Computing: An Applied Approach", 2nd Edition, Springer, 2021.
2. Johan Vos, "Quantum Computing in Action", 1st Edition, Manning Publications, 2022.

NPTEL Courses:

1. https://nptel.ac.in/courses/106106232
2. https://nptel.ac.in/courses/106104845
3. https://onlinecourses.nptel.ac.in/noc23_cs04

Course: Multi-Agent Systems and Production AI (DJS23ICSC801)

Coursera Course: Advance Multi-Agent Systems and Production AI with LangSmith

Link: https://www.coursera.org/programs/nmims-university-faculty-development-program-kjd4b/learn/advanced-multi-agent-systems-and-production-ai-with-langsmith?authProvider=nmims&source=search

Number of hours: 26 Hrs., Beginner level

About the Course:
Advance Multi-Agent Systems & Production AI with LangSmith is an advanced course that teaches learners how to build intelligent AI applications using multi-agent systems. Participants will gain practical skills in tool calling, retrieval-augmented generation (RAG), context management, debugging, safety, scalability, embeddings, and large language model (LLM) applications, enabling them to create more capable and autonomous AI solutions.
The course also focuses on deploying and managing production-ready AI systems with LangSmith and related technologies. Learners will work with LangChain, vector databases, Model Context Protocol (MCP), AI orchestration, agentic workflows, and data persistence to design, monitor, and optimize scalable AI applications for real-world use cases.

Course Outcomes: On successful completion of this course, student should be able to

1. Implement tool-calling mechanisms using LLM function interfaces.
2. Construct multi-agent workflows with task delegation and communication.
3. Explain the principles of MCP and its role in standardizing agent tools.
4. Deploy agent systems with memory persistence and safety guardrails.

Detailed Syllabus:

Unit	Description	Duration
1	**Tool-calling and Function Execution Patterns** Advance Multi Agent system and Production, What LLMs should and Should Not Tools as API contracts, Why Schema Validation is Mandatory, Safe Tools Execution Wrapper, Agents as Execution Loops, Live Agents Execution Demo Debugging with Traces, Tools Chaining Concepts.	**06**
2	**Multiagent Architectures and Coordination** Chained Agent Execution, Failure, Retries & Guardrails, Bridge to Multi-Agent Coordination, Multi-Agent Coordination, Why Single Agents Break at Scale Simulating Multi-Agent Dialogue: Case Study (Research Agents), CrewAI - What it is? CrewAI - Minimal Working Example, Demo: Multi-Agent Workflow for Research Summarization	**06**
3	**Model Context Protocol (MCP) and Tool Integration** Why MCP Standardization for Agent Interoperability, Core Components Schemas, Endpoints, and Contracts, Creating an MCP Server Using FastAPI, Schema Registration and Discovery Workflows, testing a validating MCP endpoints, Integrating MCP Clients with Existing Agents, Chaining MCP Tools for Multi-Step Tasks, Demo: Agent Using MCP for Web Search + Summarization	**07**

4	**Production, Evaluation and Scaling** AWS Production, Evaluation and Scaling for Agents System, Deployment and Persistence, Implementing Persistent Memory- DynamoDB + ChromaDB, Evaluation and Safety Guardrails, Building Safety Guardrails: Prompt Filters and Red-Teaming, Scaling and Monitoring in Production, Scaling Architectures- Horizontal vs Vertical.	**07**
	Total	**26**

Continuous Assessment (B)

Evaluation Scheme:

1. The identified course, **Advanced Multi-Agent Systems & Production AI with LangSmith**, shall focus on developing advanced AI competencies, including multi-agent system design, agent orchestration, evaluation, monitoring, and deployment of production-ready AI applications. Each selected course shall carry one (1) credit.
2. Faculty members shall act as mentors for the Coursera courses identified for various semesters of the program.
3. One faculty mentor shall be assigned to each division of the program for effective coordination and monitoring of this activity.
4. The assigned faculty mentor shall complete and review the course content before assigning the course to students.
5. The term work for these one-credit courses shall form a part of the Internal Continuous Assessment (ICA)component.
6. The Term work (ICA) component shall carry **40 marks**

The distribution of marks for term work shall be as follows:

- **15 Marks**: Coursera Course Completion and Certification
- **15 Marks**: Faculty-curated quiz conducted through the Coursera/Any other platform
- **10 Marks**: Internal faculty evaluation through quizzes, assignments, presentations, or project