Shri Vile Parle Kelavani Mandal’s

Dwarkadas J. Sanghvi College of Engineering

(Autonomous College Affiliated to the University of Mumbai)

Scheme and Detailed Syllabus (DJS22)

Final Year B.Tech

in

Information Technology

(Semester VII and VIII)

Revision: 2 (2025)
With effect from the Academic Year: 2025-2026

1st July 2025

Scheme for Final Year Undergraduate Program in Information Technology: Semester VII (Autonomous)
(Academic Year 2025-2026)

Sr. No	Course Code	Course	Teaching Scheme				Semester End Examination(A)						Continuous Assessment (B)					Aggregate (A+B)	Credits earned	
			Theory (hrs.)	Practical (hrs.)	Tutorial (hrs.)	Credits	Duration (hrs)	Theory	Oral	Pract	Oral & Pract	SEE Total (A)	Term Test 1 (TT1)	Term Test 2 (TT2)	Total (TT1 & TT2)	Term Work Total	CA Total (B)			
1	DJS22ITC701	Service Oriented Architecture	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	4
	DJS22ITL701	Service Oriented Architecture Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	
2	DJS22ITC702	Parallel and Distributed Computing	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	4
	DJS22ITL702	Parallel and Distributed Computing Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	
3 @	DJS22ITC7011	Wireless Sensor Network	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	4
	DJS22ITL7011	Wireless Sensor Network Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	
	DJS22ITC7012	User Centered Design	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ITL7012	User Centered Design Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	
	DJS22ITC7013	Business Analytics	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ITL7013	Business Analytics Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	
	DJS22ITC7014	Deep Learning	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ITL7014	Deep Learning Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	

	DJS22ITC7015	Blockchain Technology	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ITL7015	Blockchain Technology Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	
4#	DJS22ILO7011	Product Lifecycle Management (PLM)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	3
	DJS22ILO7012	Management Information System (MIS)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO7013	Operations Research (OR)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO7014	Cyber Security and Laws (CSL)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO7015	Personal Finance Management (PFM)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO7016	Energy Audit and Management (EAM)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO7017	Disaster Management and Mitigation Measures (DMM)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO7018	Science of Wellbeing (SW)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO7019	Research Methodology (RM)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO7020	Public Systems and Policies (PSP)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
5	DJS22ITP703	Project Stage – I	--	4	--	2	--	--	50	--	--	50	--	--	--	50	50	100	2	2
		Total	**12**	**10**	**0**	**17**	**08**	**260**	**125**	**0**	**0**	**385**	**80**	**60**	**140**	**125**	**25**	**650**	**17**	

@Any 1 Elective Course

#Any 1 Institute Professional Elective

Prepared By Checked By Head of the Department Vice Principal Principal

Scheme for Final Year Undergraduate Program in Information Technology: Semester VIII (Autonomous) (Academic Year 2025-2026)

Sr. No.	Course Code	Course	Teaching Scheme				Semester End Examination (A)						Continuous Assessment (B)					Aggregate (A+B)	Credits earned	
			Theory (hrs.)	Practical (hrs.)	Tutorial (hrs.)	Credits	Duration (hrs)	Theory	Oral	Pract	Oral & Pract	SEE Total (A)	Term Test 1 (TT1)	Term Test 2 (TT2)	Total (TT1 & TT2)	Term Work Total	CA Total (B)			
1	DJS22ITC801	Semantic Web Technology	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	4
	DJS22ITL801	Semantic Web Technology Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	
2	DJS22ITC802	Design Patterns	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	4
	DJS22ITL802	Design Patterns Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	
3 @	DJS22ITC8011	Industrial Internet of Things	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	4
	DJS22ITL8011	Industrial Internet of Things	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	
	DJS22ITC8012	Gamification	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ITL8012	Gamification Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	
	DJS22ITC8013	Predictive Analytics	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ITL8013	Predictive Analytics Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	
	DJS22ITC8014	Advanced Machine Learning	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ITL8014	Advanced Machine Learning Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	
	DJS22ITC8015	Advanced Security	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ITL8015	Advanced Security Laboratory	--	2	--	1	--	--	25	--	--	25	--	--	--	25	25	50	1	

4#	DJS22ILO8021	Project Management (PM)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	**3**
	DJS22ILO8022	Entrepreneurship Development and Management (EDM)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO8023	Corporate Social Responsibility (CSR)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO8024	Human Resource Management (HRM)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO8025	Corporate Finance Management (CFM)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO8026	Logistics and Supply Chain Management (LSCM)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO8027	IPR and Patenting (IPR)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO8028	Digital Marketing Management (DMM)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO8029	Environmental Management (EM)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
	DJS22ILO8030	Labour and Corporate Law (LCL)	3	--	--	3	2	65	--	--	--	65	20	15	35	--	35	100	3	
5	DJS22ITP803	Project Stage – II	--	10	--	5	--	--	100	--	--	100	--	--	--	100	100	200	**5**	**5**
		Total	**12**	**16**	--	**20**	**08**	**260**	**175**	--	--	**435**	**80**	**60**	**140**	**175**	**315**	**750**	**20**	

@Any 1 Elective Course
#Any 1 Institute Professional Elective

Prepared By Checked By Head of the Department Vice Principal Principal

Continuous Assessment (A):

Course	Assessment Tools	Marks	Time (hrs.)
Theory	a. One Term test (based on 40 % syllabus)	20	1
	b. Second Term test (next 40 % syllabus) / presentation / assignment / course project / group discussion / any other.	15	1
	Total marks (a + b)	35	--
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.	65	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

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

Course: Service Oriented Architecture (DJS22ITC701)
Course: Service Oriented Architecture Laboratory (DJS22ITL701)

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 completion of the course, learner will 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	**Introduction to SOA and Web Services:** Concepts of Distributed Computing, Fundamental of SOA, Evolution of SOA, Characteristics of SOA, Comparing SOA to client-server and distributed internet architectures, Architectures and Standards, Directory services, SOAP, REST WSDL, UDDI, Computing with Services, W3C, and Service Description.	**05**
2	**Principles of SOA:** Service orientation and object- orientation, SOA Standards Stack, SOA with Web Services, Service composition guidelines – Entity-centric business service design, Application-centric business service design, Task-centric business service design.	**05**
3	**Message Queue and WS-* Extension:** Message Exchange Pattern, Enterprise messaging with Message Broker, Asynchronous vs. Synchronous message handling, Queues and Topics, Messaging Topologies, WS-Addressing, WS-Reliable Messaging, WS- Policy, WS-Metadata Exchange, WS-Security.	**10**
4	**Service-Oriented Computing:** Service Life Cycle, Service Creation, Service Design and Build, Service Deployment, Publish Web service using UDDI, Service Discovery, Service Selection, Service Composition, Service Execution and Monitoring, Modeling Business Processes with Business Process Execution Language (BPEL).	**07**
5	**SOA Platforms:** SOA support in J2EE, RESTful Web Services - Java (JAX-RS), JAXB with JSON-B (Java API for JSON Binding), JSON-B (Java API for JSON Binding), SOA support in python, GraphQL as query language for APIS, Wrapping GraphQL with existing RESTful, SOAP APIs.	**06**
6	**Microservices:** Evolution, Monolithic Architecture, SOA vs Microservice, Microservice and API, Microservices Architecture, Microservices Design, Domain Driven Design, Microservice Architecture Decisions, Microservices Security, Service-to-Service Authentication and Authorization.	**06**

Suggested Lab Experiments

1. Develop a simple REST API that performs CRUD operations using Spring boot.
2. Develop a REST API using frameworks like Flask (Python).
3. Develop a basic SOAP service using Java (JAX-WS).
4. Implement JSON-B or Jackson in a Java-based REST API.
5. Implement communication between microservices using **gRPC**, a high-performance RPC framework.
6. Develop an integration service that calls multiple services (e.g., Payment Service, Shipping Service) and returns the result as a composite response.
7. Using AXIS 2 framework and TOMCAT create a simple calculator web service and create a java client to consume this web service.
8. Build and consume a GraphQL API.
9. Create a microservice using Docker.
10. Create any full-fledged system using microservice architecture.

Books Recommended:

Textbooks:

1. Thomas Erl, "Service Oriented Architecture: Concepts, Technology, and Design", 1st Edition, Pearson education, 2006.
2. Munindar P. Singh and Michael N. Huhns, "Service-Oriented Computing: Semantics, Processes, Agents", 1st Edition, John Wiley & Sons, Ltd., 2005.
3. Sam Newman, "Building Microservices", 2nd Edition, O'Reilly Media, 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.

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

Course: Parallel and Distributed Computing (DJS22ITC702)
Course: Parallel and Distributed Computing Laboratory (DJS22ITL702)

Pre-requisite: Knowledge of Computer Networks and Operating System

Course Objectives: The objective of this course is to introduce the fundamentals of parallel and distributed computing that includes system architecture, programming model, design & implementation and performance analysis of these systems. The course also introduces concepts related to message passing interface, GPU, multithreaded programming and cloud computing.

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

1. Develop parallel computing solutions to programming problems.
2. Develop distributed applications.
3. Provide appropriate cloud computing solutions.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to Parallel Processing Architecture:** Introduction to Parallel Processing, Parallelism in sequential machines, Abstract model of Parallel computer, multiprocessor architecture, pipelining, array processors, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA).	04
2	**Parallel Programming:** Data Dependency Analysis: Types of Dependencies, Loop and Array Dependence, Loop Dependence Analysis, Shared Memory Programming, General model of Shared Memory Programming: Process Creation, Mutual Exclusion, Examples, General model of Shared Memory Programming, Principles of Message-Passing Programming, The Building Blocks: Send and Receive Operations, MPI: Message Passing Interface Section, Topologies and Embedding, Overlapping Communication with Computation, Collective Communication and Computation Operations, Groups and Communicators, OpenMP: Standard for Directive Based Parallel Programming, Dense Matrix Algorithms: Matrix-Matrix Multiplication, Solving a System of Linear Equations. **Performance Measures:** Speedup, execution time, efficiency, cost, scalability, Effect of granularity on performance, Scalability of Parallel Systems, Amdahl's Law, Gustafson's Law, Performance Bottlenecks.	08
3	**GPU Architecture and Programming:** GPU architecture, parallel algorithm design, data parallelism, CUDA (Compute Unified Device Architecture): A general-purpose parallel computing architecture, kernel, thread organization, memories hierarchy, heterogeneous programming, Introduction to CUDA Programming, data parallelism using multi-GPU computing.	04
4	**Introduction to Distributed Computing:** Definition of Distributed Computing, Goals of Distributed Computing, Distributed Computing Models, Software Concepts, and Issues in designing Distributed System, Multithreading, Clock Synchronization, Physical and Logical Clocks, Election Algorithms, Mutual Exclusion, Distributed Transactions, Deadlocks. Difference between distributed and parallel architectures and examples of both.	07

5	**Communication:** Message Passing, 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.	**08**
6	**Cloud Computing and Virtualization:** Cloud Computing definition, Components of Cloud Computing, Cloud types: NIST and Cloud Cube Models, Cloud Deployment Models and Service Models, Cloud Computing architecture, Advantages and Disadvantages of Cloud Computing. Virtualization: Characteristics of virtualized environment, Type I & Type II Hypervisors, Taxonomy of virtualization, Implementation Levels of Virtualization, Virtualization of CPU, Memory and I/O Devices, Technology Examples: KVM, Xen, Vmware and HyperV. Exploring the Components of Amazon Web Services such as EC2, S3, EBS.	**08**

Suggested Lab 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
   a. Multiplication
   b. Parallel Sort
7. Creating distributed applications using RPC / RMI.
8. Implementation of Election Algorithms.
9. Implementation of Mutual Exclusion Algorithms.
10. Implementation of SaaS, PaaS and IaaS using AWS.
11. Basics Programs using Go Language.
12. Concurrent Programming using Go routines.
13. Concurrent Programming using Go channels.
14. Basics Programs using Julia Language.
15. Concurrent Programming using Julia.
16. Implementation of Julia Ecosystem.

Books Recommended:

Textbooks:

1. M. Sasikumar, Dinesh Shikhare and P. Ravi Prakash, “Introduction to Parallel Processing”, 2nd Edition, PHI, 2014.
2. Sunita Mahajan and Seema Shah, “Distributed Computing”, 2nd Edition, Oxford, 2010.
3. Andrew S. Tanenbaum and Maarten van Steen, “Distributed Systems: Principles and Paradigms”, 3rd Edition, Prentice Hall of India Private Limited, 2017.
4. Ananth Grama, Anshul Gupta, George Karypis and Vipin Kumar, “Introduction to Parallel Computing”, 2nd Edition, Pearson Education, 2007.
5. Kai Hwang and Naresh Jotwani, “Advanced Computer Architecture: Parallelism, Scalability, Programmability”, 2nd Edition, McGraw Hill, 2010.
6. Edward Kandrot and Jason Sanders, “CUDA by Example – An Introduction to General Purpose GPU Programming”, Addison-Wesley Professional, 2010.

7. Benedict R. Gaster, Lee Howes, David R Kaeli, Perhaad Mistry and Dana Schaa, “Heterogeneous Computing with OpenCL”, 2nd Edition, Elsevier, 2013.

Reference Books:

1. Pradeep K. Sinha, “Distributed Operating Systems”, 1st Edition, Prentice Hall of India Private Limited, 2012.
2. Georg Hager and Gerhard Wellein, “Introduction to High Performance Computing for Scientists and Engineers”, Chapman & Hall / CRC Computational Science series, 2011.
3. Michael J. Quinn, “Parallel Programming in C with MPI and OpenMP”, 1st Edition, McGraw-Hill International Editions, Computer Science Series, 2008.
4. Kai Hwang and Zhiwei Xu, “Scalable Parallel Computing: Technology, Architecture, Programming”, McGraw Hill, 1998.
5. Laurence T. Yang and MinyiGuo, “High- Performance Computing: Paradigm and Infrastructure”, Wiley, 2006.

Web Resources:

1. Introduction to Parallel Computing from Livermore Computing.
2. Links to Parallel and Network Programming Resources MPI, OpenMP, posix threads, socket programming, CUDA.
3. Some Cluster and Distributed Systems Papers
4. https://golang.org/doc/tutorial/
5. https://levelup.gitconnected.com/goroutines-and-channels-concurrent-programming-in-go-9f9f8495c34d
6. https://docs.julialang.org/en/v1/
7. https://julialang.org/learning/tutorials/
8. https://www.run.ai/guides/multi-gpu#:~:text=Model%20parallelism&text=Using%20this%20method%2C%20you%20split,synchronizing%20data%20between%20the%20splits.
9. https://huggingface.co/docs/transformers/en/perf_train_gpu_many
10. https://pytorch.org/tutorials/beginner/blitz/data_parallel_tutorial.html

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

Course: Wireless Sensor Network (DJS22ITC7011)
Course: Wireless Sensor Network Laboratory (DJS22ITL7011)

Pre-requisite: Knowledge of Analog and digital communication, Computer Network.

Course Objectives: The objective of this course is to provide a comprehensive introduction to wireless sensor technology. The course familiarizes students with wireless technologies and applications of WSN. Students will understand the design of emerging wireless transmission technology and systems. They will also learn to analyze the routing protocols for any wireless network.

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

1. Specify the requirements for the hardware and software solutions for energy-efficient sensor network.
2. Analyze various critical parameters in deploying a WSN.
3. Apply appropriate algorithms to improve existing or to develop new WSN applications.
4. Design a WSN for given sensor data using microcontroller, transceiver, middleware and operating system.
5. Work effectively as a member of the team.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to Wireless Communication:** Fundamentals of Wireless Communication, Advantages, limitations, and application, DSSS and FHSS, Frequency Spectrum: Radio and Infrared; Overview of wireless generations: 1G: Cellular, 2G: Mobile Radio, 3G: UMTS, 4G: LTE, , 5G	**06**
2	**Evolution of Wireless Technologies:** Multiple Access Technique: TDMA, FDMA, CSMA, CDMA, Wireless Technology Overview and its architectures: GSM, GPRS, EDGE, CDMA, UMTS and LTE	**07**
3	**Basic Wireless Sensor Technology**: Sensor Node Technology, Hardware and Software, Sensor Taxonomy, Wireless sensor network operating environment, Trends in Wireless sensor networks, commercially available sensor nodes –Imote, IRIS, Mica Mote, EYES nodes, BTnodes, TelosB, Sunspot.	**07**
4	**Types and Applications of WSN**: MANET & VANET, Application, Advantage, and limitations, Introduction, Background, Range of Applications, WSN Applications - Home Control – Building Automation - Industrial Automation - Medical Applications - Reconfigurable Sensor Networks - Highway Monitoring - Military Applications	**07**
5	**Emerging Wireless Transmission Technology and Systems:** Wireless Technologies: Campus Applications, Bluetooth, WLAN, ZigBee, Hotspot/WiMAX, MAN/WAN Applications	**06**
6	**Routing Protocols for Wireless Networks**: Issues in designing a routing protocol for ad hoc Wireless networks, Classifications of routing protocols, On-Demand Routing Protocols: Ad Hoc On-Demand Distance-Vector Routing Protocol, Hybrid Routing Protocols: Zone Routing Protocol	**06**

Lab guidelines for mini project:

1. The mini project work is to be conducted by a group of three students.
2. Each group will be associated with a Subject In-Charge/ mini project mentor. The group should meet with the

concerned faculty during Laboratory hours and the progress of work discussed must be documented.

3. The students may do survey for different application using different types of sensors for their mini project.
4. Each group will identify the Hardware (Motes from different Motes families) & sensor configuration and software requirement for their mini project problem statement.
5. Design your own circuit board using multiple sensors etc.
6. Installation, configure and manage your sensors in such away so that they can communicate with each other.
7. Work with operating system, emulator like contiki cooja and do coding to for input devices on sensors.
8. Create and interface using Mobile/Web to publish or remotely access the data on Internet.
9. Each group along with the concerned faculty shall identify a potential problem statement, on which the study and implementation is to be conducted.
10. Each group may present their work in various project competitions and paper presentations.
11. A detailed report is to be prepared as per guidelines given by the concerned faculty.

List of Projects not limited to:

1. Home Control.
2. Building Automation.
3. Industrial Automation.
4. Medical Applications.
5. Reconfigurable Sensor Networks.
6. Highway Monitoring.
7. Military Applications.
8. Civil and Environmental Engineering Applications.
9. Wildfire Instrumentation.
10. Habitat Monitoring.
11. Nanoscopic Sensor Applications.

Books Recommended:

Textbooks:

1. Holger Karl and Andreas Willig, “Protocols and Architectures for Wireless Sensor Networks”, 1st Edition, Wiley Interscience, 2007.
2. T.L Singal, “Wireless Communications”, McGraw Hill Education, 2017.
3. Kazem Sohraby, Daniel Minoli and Taieb Znati, “Wireless Sensor Networks Technology, Protocols, and Applications”, Wiley, 2010.

Reference Books:

1. C. Siva Ram Murthy and B.S. Manoj, “Ad Hoc Wireless Networks Architectures and Protocols”, Pearson Education, 2006.
2. S. Sitharama Iyengar and Nandan Parameshwaran, Vir V. Phoha, N. Balakrishnan, Chuka D. Okoye, “Fundamentals of Sensor Network Programming: Applications and Technology”, Wiley, 2010.
3. Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)”, Orient Blackswan, 2015.

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

Course: User Centered Design (DJS22ITC7012)
Course: User Centered Design Laboratory (DJS22ITL7012)

Pre-requisite: Knowledge of Software Engineering and Web Programming,

Course Objectives: The objective of the course is to explore various user research methods and information architecture and to use them in interaction design, visual design and functional Layout Design. The course also introduces students to usability testing which is performed on various designs.

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

1. Identify user requirements.
2. Design UI/UX using appropriate methods.
3. Generate test report using usability testing.
4. Work effectively as a member of the team.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction:** What is UX Design? What is UI Design? UX Design Deliverables, seven factors that influence user experience, UX design laws and their uses, Elements used in User Experience Design, and How it works together. **Design Thinking Overview:** What is Design Thinking; Design Mindset; Brief History; Various Models; Design Thinking Life Cycle, how it connects with other methods & Key Stages; 10 commandments.	**04**
2	**User Research and Analysis:** What is UX research? What's the difference between good and bad UX research? What are the five steps of UX research? User research importance, KPIs in User Experience. **Understanding User Problem:** Involves identifying the stages of problem solving, applying the IDEAL framework for problem-solving, and recognizing the different types of problems: Known-Known (KK), Known-Unknown (KU), and Unknown-Unknown (UU). What is empathy; Sympathy vs Empathy; how to empathize, various tools. Empathy Map. Web Content Accessibility Guidelines (WCAG 2.2). **Root cause analysis for problem definition:** AEIOU, Questions Builder, 6-WH, Persona Types, Building a Persona, Journey Map (why, benefits, how to, types of key components), HMW Statement. **User research methodologies:** Qualitative and Quantitative analysis, user interviews, focused group discussion, expert reviews, tools for user research.	**08**
3	**Interaction Design:** What is Ideation? How to ideate the problem definition? Ideation Techniques (mind map, SCAMPER, Crazy-8, Brain-writing, NABC). **Cognitive Psychology Basics:** How users perceive and process information, Mental models and their impact on design. **Visual Hierarchy & Information Architecture**: Introduction to Information architecture, card sorting, open card sorting, semi-closed card sorting, closed card sorting, using excel as a tool for card sorting, Understanding Information architecture, creating IA for different industries, IA types and structures. **Emotional Design:** Using colors, typography, Use of imagery and iconography and visual elements to evoke trust, excitement, or calmness depending on the product type (e.g., urgency for flash sales in e-commerce, serenity for meditation apps).	**08**

	Functional Layout Design: Z-Pattern, F-Pattern, and Browsing vs. Searching vs. Discovery, Page Framework, The Fold, The Axis of Interaction Forms	
4	**Wireframing and Prototyping**: Paper Prototyping, Low fidelity & High-Fidelity Prototypes, build your own Prototyping, Prototyping Tools for UI/UX Designers — How to Choose the Right One? Designing a Web / Mobile App. **Usability Testing:** Learning the process of conducting usability tests for digital products, Usability testing methodologies – task-based user testing, A/B testing, lab-based user testing, remote user testing, moderated & unmoderated user testing.	**07**
5	**Product Design and Analytics:** Types of products & solutions, an overview of using analytics tools, such as Google Analytics, to measure performance goals on your website or application. **UX Writing: Language, Directives, & Jargon:** Microcopy & User Guidance: Writing clear, actionable, and concise copy for buttons, instructions, and error messages. **Interaction with Development Team:** Understand how to collaborate and communicate with Developers, prepare for handing off designs, how to effectively present your ideas, and collaborate with internal and external stakeholders (clients, business analysts, project manager), and Different ways to share the intangible, receive feedback in critiques and make modifications based on feedback.	**08**
6	**UX Design for futuristic technologies:** UX processes for AR/VR, what is Imagineering? What are the various tools & devices used to design? Field of view for AR/VR, the difference in Designing for 2D & 3D interfaces, Gesture, gaze, and voice control, getting familiar with AR is, Challenges of 3D interaction design, conversational UI design	**04**

Suggested Lab Experiments:

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

1. Understanding the User Needs: Conduct interviews with potential users to gather insights into their pain points, needs, and desires.
2. Create empathy maps based on user interviews to identify their behaviors, needs, and feelings.
3. User Personas Creation: Develop detailed user personas based on collected data to represent the core user groups.
4. Identify the root causes of user pain points by analyzing data collected during the empathy phase and develop clear problem statements based on user needs and insights.
5. Ideate: Conduct a brainstorming session where team members ideate and share ideas without restrictions.
6. Apply the Scamper method to enhance idea generation.
7. Prototyping: Create low and high-fidelity prototypes to represent the proposed solution and gather early feedback.
8. Conduct usability testing on the prototype to identify potential improvements.
9. Validation through User Feedback: Validate the prototype's effectiveness and user satisfaction by testing in real-world conditions.
10. Assess the ethical considerations in design projects and explore leadership strategies in managing design teams.

Books Recommended:

Textbooks:

1. Garrett and J. J., “The Elements of User Experience: User-Centered Design for the Web and Beyond”, 2nd edition New Riders Publishing, 2011.
2. Yablonski and J., “Laws of UX: Using Psychology to Design Better Products & Services”, 2nd edition, O’Reilly Media, 2020.

3. Rosenfeld, L., Morville and P., “Information Architecture: For the Web and Beyond”, 4th edition, O’Reilly Media, 2015.
4. Krug, S., “Don't Make Me Think: A Common Sense Approach to Web Usability”, 2nd edition, New Riders Publishing, 2014.
5. Cypher, “A., Designing for Augmented Reality and Virtual Reality: UX for a New Medium”, Springer, 2018.

Reference Books:

1. Brown, T., “Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation”, HarperBusiness, 2009.
2. Kuniavsky, M., “Observing the User Experience: A Practitioner’s Guide to User Research”, 2nd edition, Morgan Kaufmann, 2003.
3. Gothelf, J., Seiden, J., “Lean UX: Applying Lean Principles to Improve User Experience”, O’Reilly Media, 2013.

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

Course: Business Analytics (DJS22ITC7013)
Course: Business Analytics Laboratory (DJS22ITL7013)

Pre-requisite: Data warehousing and mining, Statistical Analysis, Big Data Analytics.

Course Objectives: This course aims to equip students with a comprehensive understanding of business analytics, differentiating between various analytical methods such as descriptive, diagnostic, predictive, prescriptive, and cognitive analytics. Participants will learn to identify and frame business problems analytically, utilize diverse data types and analytical tools, and apply visualization techniques to drive data-driven decision-making. Additionally, the course will explore the implications of augmented analytics and the ethical considerations surrounding advanced analytical practices.

Course Outcomes: On completion of the course, learner will be able to:
1. Apply analytical techniques for strategic decision-making.
2. Analyze ethical issues in analytics.
3. Work effectively as a member of a team.

Detailed Syllabus: (unit wise)		
Unit	**Description**	**Duration**
1	**Introduction:** Meaning, Definition and Importance of Business Analytics – Analytics v/s Analysis –Business Analytics v/s Business Intelligence and Data Mining – Applications of Analytics – Different Kinds of Analytics (Descriptive Analytics, Diagnostic Analytics, Predictive Analytics, Prescriptive Analytics, Cognitive Analytics) – Types of Analytical Tools – Identifying Problems & Opportunities through Data Analytics – Framing a Business Problem as an Analytical Problem, structured methodologies-CRISP-DM	**06**
2	**Descriptive Analytics:** Introduction to Descriptive Analytics, Data Types and Scales, Types of Data Measurement Scales, Population and Samples, different sampling methods, Chebyshev's Theorem, Measures of Shapes, Types of Data: Structured data (e.g., sales figures, survey results), Time-series data (e.g., monthly sales over time), Data Collection and Management: Sources of data, Data cleaning and pre-processing, Descriptive Statistics: Measures of central tendency (mean, median, mode), Measures of variability (range, variance, standard deviation), Data Analysis – Data visualization – visualization techniques – Bar charts, histograms, box plots, pie charts, Tables, Charts, Cross-tabulations, Storytelling through data, Dashboard design principles, Dash boards.	**07**
3	**Diagnostic Analytics:** Introduction to Diagnostic Analytics-Difference between descriptive and diagnostic analytics, Types of Data: Historical data (e.g., sales data from previous years), Categorical data (e.g., customer demographics), Geospatial data, Techniques for Root Cause Analysis: Techniques (e.g., Fishbone diagrams, 5 Whys), Statistical techniques (e.g., correlation, regression analysis), Data Exploration and Visualization: Scatter plots, heat maps, and correlation matrices.	**07**
4	**Prescriptive Analytics:** Introduction to Prescriptive Analytics-Definition and significance in decision-making, Types of Data: Operational data (e.g., resource allocation data) Simulation data (e.g., scenarios generated for analysis), Web Data, Optimization Techniques: Linear programming and constraint optimization, Simulation methods (e.g., Monte Carlo simulation) Visualization Techniques: Decision trees,	**07**

	flowcharts, and optimization dashboards, Decision Analysis: techniques for decision-making, Scenario analysis and sensitivity analysis	
5	**Augmented Business Intelligence**: Introduction to Augmented BI, highlighting the evolution from traditional BI and self-service BI to augmented analytics. Understanding the augmented BI roadmap—from data reporting to data augmentation. Major components include Infrastructure (cloud-native solutions and SaaS), Analytics (machine learning, cognitive and conversational BI, data relationships, insight generators), and Share (data stories, decision support systems). **Augmented Analytics:** Definition, The Five I's of Augmented Analytics, Overcoming the Limitations of Traditional Analytics Approaches, Augmented Workflows, The Benefits of Augmented Analytics, Overcoming Bias, Key Enablers of Augmented Analytics: Automation and AI, Artificial Intelligence: The Five Archetypes, The Limitations of Augmented Analytics, The Challenges of Augmented Analytics	**07**
6	**Ethical Considerations**: Data privacy and security, addressing bias and fairness in algorithms, and compliance with regulations such as GDPR. Challenges and limitations: Data quality issues, interpretation of automated insights, and the dynamics of human-machine collaboration. **Future Trends in Business Analytics and BI:** The evolving role of AI and machine learning, emerging technologies and innovations in BI, and strategies for preparing for the future workforce.	**05**

Suggested Lab Experiments:

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

1. Understanding the User Needs: Conduct interviews with potential users to gather insights into their pain points, needs, and desires.
2. Data Cleaning and Pre-processing: to clean a dataset by handling missing values, removing duplicates, and normalizing data.
3. Descriptive Statistics Analysis: Analyze a dataset using statistical measures (mean, median, mode, variance, standard deviation) and visualize results using Matplotlib (Python) or ggplot2 (R).
4. Data Visualization: Create various visualizations (bar charts, histograms, pie charts) analyze trends.
5. Time-Series Analysis: Perform time-series analysis on historical sales data using libraries like statsmodels (Python) or forecast (R) and visualize trends.
6. Correlation and Regression Analysis: Conduct correlation analysis and regression modeling on a dataset.
7. Prescriptive Analytics with Optimization: to perform linear programming and optimization for resource allocation in a business scenario.
8. Cognitive Analytics with NLP: Implement Natural Language Processing techniques using to analyze customer reviews for sentiment analysis.
9. Augmented Analytics: Create dashboards in Power BI or Tableau that incorporate augmented analytics features, such as AI-driven insights and anomaly detection.
10. Data Exploration with Geospatial Data: Analyze and visualize geospatial data using libraries like Geopandas or Folium to uncover geographic trends in business data.
11. Mini project: Analyse a business dataset, clean and process it, and use various analytics techniques (descriptive, prescriptive, augmented) to draw actionable insights. Present findings through visualization tools and an AI-powered dashboard, simulating a real-world business decision-making process.

Books Recommended:

Textbooks:

1. Jeffrey D. Camm, James J Cochran, Michael J. Fry, Jeffrey W. Ohlmann, David R. Anderson, “Business Analytics with MindTap”, 4th Edition, Cengage Learning, 2024.
2. James R. Evans, “Business Analytics”, 3rd Edition, Pearson, 2019.
3. Willi Weber, Tobias Zwingmann, “Augmented Analytics”, 1st Edition, O’Reilly Media, Inc., May 2024.

Reference Books:

1. S. Christian Albright, Wayne L. Winston, “Business Analytics: Data Analysis and Decision Making with MindTap”, 7th Edition, Cengage Learning, 2022.
2. Judith Hurwitz, Marcia Kaufman, Adrian Bowles, “Cognitive Computing and Big Data Analytics”, 1st Edition, John Wiley & Sons, 2012.
3. Alice LaPlante, “What Is Augmented Analytics?”, 1st Edition, O’Reilly Media, Inc., 2019.

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

Course: Deep Learning (DJS22ITC7014)
Course: Deep Learning Laboratory (DJS22ITL7014)

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 completion of the course, learner will be able to:

1. Interpret working of different types of neural networks.
2. Select the suitable deep learning architecture for various real-world applications.

Detailed Syllabus: (unit wise)		
Unit	**Description**	**Duration**
1	**Fundamentals of Deep Learning:** Overview of 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, Gradient Descent: Concept of optimization, basic gradient descent algorithm, introduction to backpropagation.	**06**
2	**Supervised Learning with Feedforward Neural Networks:** Feedforward Neural Networks: Architecture, forward pass, weight initialization, 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, Regularization: Introduction to overfitting and underfitting, L2 regularization (Ridge), L1 regularization (Lasso), Elastic Net.	**07**
3	**Regularization and Model Tuning:** Overfitting and Underfitting: Diagnosing and addressing overfitting and underfitting in models. Regularization Techniques: 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.	**04**
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, the vanishing and exploding gradient problem.	**06**

	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.	
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.	**06**

Suggested Lab Experiments:

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

1. Implement a single-layer perceptron from 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 pretrained models like BERT.

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, 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.

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

Course: Blockchain Technology (DJS22ITC7015)
Course: Blockchain Technology Laboratory (DJS22ITL7015)

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 completion of the course, learner will 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 Block chain:** 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. Consensus mechanism, Consensus Algorithm PoW, PoS, DPoS, issues with PoW, Distributed Consensus, Incentives and Proof of Work, Consensus without Identity.	**06**
3	**Introduction to Bitcoin:** 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	**Introduction to Ethereum:** Overview of Ethereum Blockchain, Key Features, 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.	**06**
5	**Introduction to Hyperledger** – 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, Digital Trust Networks, Beyond Cryptocurrency, Blockchain and Sustainable Technologies, World Economic Forum blockchain	**06**

	development toolkit. Blockchain and AI: Data Security, AI-Powered Applications, Decentralized AI Models.	

Suggested Lab Experiments:

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

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.

Books Recommended:

Textbooks:

1. Tiana Laurence, “Blockchain for Dummies”, 2nd Edition, Wiley Publications, 2017.
2. Daniel Drescher, “Block Chain Basics”, 1st Edition, Apress, 2017.
3. Daniel Drescher, “Blockchain Basics: A Non-Technical Introduction in 25 Steps”, 1st Edition, Apress publisher, 2017.
4. 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/

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

Course: Project Stage – I (DJS22ITP703)

Course Objectives: To introduce the students to professional engineering practice by providing them with an opportunity to work on an open-ended engineering problem. Typically, the students would apply knowledge from different areas or courses, which they have studied in their curriculum using methods, tools, and techniques, which they learned to a real-world scenario. Students would have to apply not only their engineering knowledge and proficiencies (hard skills), but also to demonstrate their competence in generic, professional skills (soft skills). It also emphasizes the importance of life-long learning as a fundamental attribute of graduate engineers.

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

1. Discover potential research areas in the field of IT.
2. Survey several available literatures in the related field of study.
3. Compare the several existing solutions for research challenges.
4. Design the solution for the research plan.
5. Summarize the findings of the study conducted.
6. Work effectively as a member of the team.

Guidelines

1. The department must allocate 1 day in the VII semester every week.
2. The project work is to be conducted by a group of three students
3. Each group shall identify a potential research area/problem domain, on which the study is to be conducted.
4. Each group will be associated with a project mentor/guide. The group should meet with the project mentor/guide periodically and record of the meetings and work discussed must be documented.
5. Students will do a rigorous literature survey of the problem domain by reading and understanding at least 3-5 research papers from current superior quality national/international journals/conferences. (Papers selected must be indexed by Scopus/IEEE/Springer/ACM etc.). The list of papers surveyed must be clearly documented.
6. Students will design and implement (30-40%) the system in Sem VII.
7. The project assessment for term work will be done at least two times at department level by giving presentation to panel members which consist of at least three (3) members as Internal examiners (including the project guide/mentor) appointed by the Head of the department of respective Program.
8. A report is to be prepared summarizing the findings of the literature survey. A comparative evaluation of the different techniques surveyed is also to be done.
9. Every team must publish their work in national / international conference/journals (if possible, publish in Scopus indexed journals).

Evaluation Scheme:

Semester End Examination (A):

Laboratory:

1. Each group will be jointly evaluated by a team of Internal and External Examiners approved by the University of Mumbai.
2. An oral exam will be conducted on the project done by the students.

Continuous Assessment (B):

Laboratory: (Term work)

1. Each team must give a presentation/demo to the Internal Panel consisting of 3 domain experts.
2. Each team will prepare a report that will summarize the results of the literature survey, analysis and design as a in SEM VII. The list of papers surveyed must be clearly documented.

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

i. Term Work shall consist of full Project-I on above guidelines/syllabus.
ii. The final certification and acceptance of term work will be subject to satisfactory performance and upon fulfilling minimum passing criteria in the term work.

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

Course: Semantic Web Technology (DJS22ITC801)
Course: Semantic Web Technology Laboratory (DJS22ITL801)

Pre-requisite: Knowledge of XML and database management systems.

Course Objectives: The objective of the course is to introduce students to the fundamental concept of the semantic web along with its advantages and limits. The course will also enable students to understand and use ontologies in the context of the semantic web.

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

1. Apply foundational concepts of Semantic Web technologies.
2. Model ontologies using Resource Description Framework (RDF) and Web Ontology Language (OWL).
3. Query ontologies using SPARQL.
4. Apply Semantic web technologies to real world applications.

Detailed Syllabus: (unit wise)		
Unit	**Description**	**Duration**
1	**Introduction-Foundation of Semantic Web Technologies:** Introduction, Todays' web, Current web vs Semantic Web, Semantic Web Technologies, A layered approach. Knowledge Graphs, Linked Data, and their applications in search engines and AI	**04**
2	**Simple Ontologies in RDF and RDF Schema:** Introduction to RDF, Syntax for RDF (XML-based), RDF Serialization, Advanced Features, RDF Schema (Lightweight Ontology): Basic Ideas, The Language, Simple Ontologies in RDF Schema, Encoding of Special Data structures, An Example, Ontology Querying in SPARQL.	**10**
3	**Web Ontology Language:** Introduction, OWL and RDF/RDFS, OWL language: three sublanguages and description, OWL Species, Layering of OWL, Examples, Explainable AI and Knowledge Representation, Advanced Reasoning Techniques with OWL.	**08**
4	**Ontology Engineering and Semantic Web Tools:** Introduction, Constructing Ontologies Manually, Reusing Existing Ontologies, Semiautomatic Ontology Acquisition, Ontology Mapping, On-To-Knowledge Semantic Web Architecture, Quality Assurance of Ontologies, Modular Ontologies: Divide and Conquer.	**06**
5	**Sematic Web Tools:** Software Tools for Ontology Engineering and Management, RDF Tools, FOAF, DublinCore, Ontology Design and Management using the Protege editor, Ontology Programming with the Jena API, Knowledge Graph Construction Tools.	**05**
6	**Applications:** Horizontal Information Products at Elsevier, Openacademia: Distributed Publication Management (distributed, semantic-based publication management), Flink: the social networks of the Semantic Web Community, Bibster: Data Exchange in a Peer-to-Peer System, Data Integration at Audi, e-Learning. Applications in AI and Machine Learning, Healthcare and Biomedical Informatics, IoT and Smart Cities, E-commerce personalization using knowledge graphs	**06**

Suggested Lab Experiments:
Experiments based on content such as RDF document preparation, OWL ontology etc.

1. Create a simple knowledge graph focusing on relationships and queries.
2. Design of Ontology using RDF.

3. Design of Ontology using RDFS.
4. Design of Ontology using OWL.
5. Querying Ontology using SPARQL.
6. Write Python scripts to load, query, and manipulate RDF/OWL data.
7. Design of any domain specific Ontology in Protégé.
8. Ontology Programming with Jena API.
9. Case Study: WordNet.
10. Case Study: Dbpedia.
11. Case study: Pizza Ontology.
12. Case Study: Applications of Semantic Web Technology.

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

List of Open-Source Software/learning website:

1. http://www.linkeddatatools.com/
2. http://opensemanticframework.org/
3. https://jena.apache.org/ Books Recommended

Books Recommended:

Text books:

1. Grigoris Antoniou, Paul Groth, Frank van Harmelen and Rinke Hoekstra, "A Semantic Web Primer", 3rd Edition MIT Press, 2012.
2. Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, "Foundations of Semantic Web Technologies", 1st Edition Chapman and Hall, 2009.
3. Peter Mika, "Social Networks and the Semantic Web", 1st Edition, Springer, 2007.

Reference Books:

1. Dean Allemang, and James Hendler, "Semantic Web for the Working Ontologist", 2nd Edition, Morgan Kaufmann, 2011.
2. John Hebeler, Matthew Fisher, Ryan Blace, Andrew Perez-Lopez, and Mike Dean (Foreword), "Semantic Web Programming", 1st Edition, Wiley, 2009.
3. David Wood, Marsha Zaidman, Luke Ruth, and Michael Hausenblas, "Linked Data: Structured Data on the Web", 1st Edition, Manning, 2014.
4. Bob DuCharme, "Learning SPARQL: Querying and Updating with SPARQL 1.1", 2nd Edition, O'Reilly Media, 2013.
5. Web Toby Segaran, Colin Evans, and Jamie Taylor, "Programming the Semantic Web, Build Flexible Applications with Graph Data", 2nd Edition, O'Reilly Media, 2009.

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

Course: Design Patterns (DJS22ITC802)
Course: Design Patterns Laboratory (DJS22ITL802)

Pre-requisite: Software Engineering, SOA, Knowledge of Programming.

Course Objectives: The objective of this course is to introduce students to a wide range of software design patterns and outlines the differences between creational, structural, and behavioral patterns. This course extends object-oriented analysis and design by incorporating design patterns to create interactive applications. Through a survey of established design patterns, students will gain a foundation for more complex software applications.

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

1. Apply the most suitable design pattern to address a given application design problem.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to Design Patterns:** What is a Design Pattern, Describing Design Patterns, How Design Patterns Solve Design Problems, Selecting a Design Pattern, Using a Design Pattern.	06
2	**Creational and Structural Patterns:** Abstract Factory, Builder, Factory Method, Prototype, Singleton; Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy.	10
3	**Behavioral Patterns:** Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor.	08
4	**Understanding SOA Design Patterns:** Fundamental Terminology, Object-Oriented Patterns, Software Architecture Patterns, Enterprise Application Architecture Patterns, EAI Patterns, SOA Patterns, Pattern Notation, Symbols, and Figures.	06
5	**Service Design Patterns:** Service Identification Patterns, Service Definition Patterns, Service Implementation Patterns, Service Security Patterns, Service Contract Design Patterns, Service Governance Patterns.	05
6	**Strategic Architecture Considerations:** “Compound” vs. “Composite”, Compound Patterns and Pattern Relationships, Increased Federation, Increased Intrinsic Interoperability, Increased Vendor Diversification Options, Increased Business and Technology Alignment.	04

Suggested Lab Experiments:

1. Design a UML class diagram for Abstract Factory Pattern and implement the same for any real life scenario. (For example, implement a code to list the courses offered to students based on college and department using Abstract Factory Pattern).
2. Code in any language of your choice to prevent Singleton Pattern from Reflection, Serialization and Cloning.
3. Draw class diagram for Flyweight Pattern and implement the same to demonstrate the working of Counter Strike Game (Instead of creating objects for each player, use Flyweight Pattern to create only 2 objects one for Terrorists class and other for Counter Terrorists class, and reuse them).
4. Implement the CoR Pattern using any language of your choice for the given problem statement. (A system in which several managers and executives can respond to a purchase request or hand it off to a superior. You are free to have your own set of rules to approve the orders.).
5. Implement the Observer Pattern using any language of your choice for any real life scenario. (For example, implement a system using Observer Pattern in which registered investors are notified every time a stock changes value).

6. Design a UML class diagram for Strategy Pattern and implement the same using any language of your choice. (For example, code in such a way that the system should encapsulate different sorting algorithms in the form of sorting objects. This should allow clients to dynamically change sorting strategies).
7. Case study on Enterprise Integration Patterns. (Students can refer to the Bond Trading System case study listed here: https://www.enterpriseintegrationpatterns.com/BondTradingCaseStudy.html).
8. Case study on: SOA Metadata Centralization Pattern.
9. Case study on: Compound Pattern (For example, students can be asked to form groups and discuss on how MVC architecture can be dissected into patterns).
10. Case Study on: “Why enterprise architecture maximizes organizational value” (Students can refer to the article here: https://www.cio.com/article/228396/why-enterprise-architecture-maximizes-organizational-value.html).

Books Recommended:

Textbooks:

1. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley Professional, 1994.
2. Thomas Erl, “SOA Design Pattern”, Pearson, 2008.

Reference Books:

1. Eric Freeman, Elisabeth Robson, Bert Bates, 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. in Information Technology **Final Year** **Semester: VIII**

Course: Industrial Internet of Things (DJS22ITC8011)
Course: Industrial Internet of Things Laboratory (DJS22ITL8011)

Pre-requisite: Overview of IoT.

Course Objectives: The objective of this course is to provide a comprehensive introduction to the concept of the Industrial Internet of Things, or IIoT. The students will learn to apply IIoT in manufacturing, and the businesses that should be considered to implement this technology. The course familiarizes students with considerations that include information technology infrastructure, the business value of implementing IIoT, and what needs to happen across the organization to ensure successful implementation.

Course Outcomes: On completion of the course, learner will be able to:
1. Explore Industry 4.0 and IIoT technologies, architectures, standards, and protocols.
2. Examine the technological developments that will shape the industrial landscape in the future.
3. Work effectively as a member of team.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to Industrial Internet of Things (IIoT):** Evolution of the Internet, What is Industrial Internet?, Why Industrial Internet and Why Now?, Key IIoT Technologies, Opportunities and Benefits.	**04**
2	**IIoT Use Cases and the Innovators of Industrial Internet:** Use Cases (Health Care, Oil and Gas Industry, Smart Office, Retail); Miniaturization, Cyber Physical Systems (CPS), Wireless Technology, IP Mobility, Network Functionality Virtualization (NFV), Network Virtualization, SDN (Software Defined Networks), The Cloud and Fog, M2M Learning and Artificial Intelligence, 3D Printing, People versus Automation.	**08**
3	**IIoT Reference Architecture:** Traditional M2M architecture, IIC (Industrial Internet Committee) Reference Architecture, Industrial Internet Architecture Framework (IIAF), Viewpoints (Business, Usage, Functional), Architectural Topology, The Three-Tier Topology (Edge, Platform, Enterprise), Connectivity, Key Functional Characteristics of Connectivity.	**08**
4	**Designing the Industrial Internet Systems:** Proximity Network, WSN Edge Node (WSN Network Protocols, Low-Power Technologies, Designing Low-Power Device Networks), Legacy Industrial Protocols (RS232 Serial Communications, Field Bus Technologies), Modern Communication Protocols (Industrial Ethernet, Encapsulated Field Bus, Standard Ethernet), Wireless Communication Technologies (IEEE 802.15.4, Bluetooth Low Energy, ZigBee and ZigBee IP, Z-Wave, Wi-Fi Backscatter, RFID, NFC, 6LoWPAN, RPL).	**09**
5	**IIoT Middleware Transport Protocols:** TCP/IP, UDP, RTP, CoAP, Middleware Software Patterns (MQTT, XMPP, AMQP, DDS), IIoT Middleware Architecture.	**06**
6	**Securing the IIoT and Introduction to Industry 4.0:** Security in Manufacturing, PLCs and DCS, Securing the OT, Potential Security Issues - Network Level and System Level, Identity Access Management, Defining Industry 4.0, Four Main Characteristics of Industry 4.0, Industry 4.0 Design Principles, Building Blocks of Industry 4.0.	**04**

Lab guidelines for mini project:
1. The mini project work is to be conducted by a group of three students (four in extreme case; call can be taken by subject in-charge).
2. The group should meet with the concerned faculty during laboratory hours and document the progress of work.

3. The students should be given sufficient time (6-8 hrs) to do survey for finalizing their mini project topic using Raspberry Pi / Arduino / ARM Cortex / Intel Galileo etc.
4. Each group should identify a potential problem statement on which the study and implementation is to be conducted and will also identify the hardware and software requirements for their mini project.
5. Once the topic has been finalized, students either can buy the required components by themselves or can request the college to provide the components.
6. Concerned faculty will do the term work assessment after seeing the group's presentation and overall implementation of the mini project.
7. Each group may present their work in various project competitions and paper presentations.
8. A detailed report is to be prepared as per guidelines given by the concerned faculty.

Books Recommended:

Textbooks:

1. Alasdair Gilchrist, "Industry 4.0: The Industrial Internet of Things", Apress, 2017.
2. Anandarup Mukherjee, Chandana Roy, and Sudip Misra, "Introduction to Industrial Internet of Things and Industry 4.0", CRC Press, 2020.

Reference Books:

1. Giacomo Veneri, Antonio Capasso, "Hands-On Industrial Internet of Things", Packt, 2018.
2. Carolina Machado, J. Paulo Davim, "Industry 4.0: Challenges, Trends, and Solutions in Management and Engineering", CRC Press, 2020.

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

Course: Gamification (DJS22ITC8012)
Course: Gamification Laboratory (DJS22ITL8012)

Pre-requisite: Basic knowledge of HCI, UI/UX.

Course Objectives: The course introduces the students to the application of game-design elements and game principles. The objective of the course is to develop problem-solving capabilities using gamification.

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

1. Design games using gamification principles.
2. Work effectively as a member of a team.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to Game Design:** Motivation, Types of games, Different aspects of game design; Different components in a game, Game engines, Design Schemas, Game Design Fundamentals.	04
2	**The Design Process:** Iterative Design, Commissions, Game creation, Game Modification, Game Analysis, Design Process, Scripted Game Deign, Play Testing. **Game Mechanics and Dynamics:** Feedback and Re-enforcement, designing for engagement Game Mechanics in depth. **Rules of Digital Games:** Introduction, Types of Rules: constitutive, operational, and implicit, Case Study: Rules of Tetris, Why Rules.	08
3	**Foundations of Gamification:** Definition of Gamification, Why Gamify, Examples and Categories, Gamification in Context, Resetting Behavior, Replaying History, Gaming foundations: Fun Quotient, Evolution by loyalty, status at the wheel, the House always wins, Case studies.	06
4	**Opponent Moves in Gamification:** Reclaiming Opposition: Counter gamification, Gamed Agencies: Affectively Modulating Our Screen-and App-Based Digital Futures, Remodeling design, Game Mechanics.	04
5	**The Octalysis Framework:** The 8 Core Drives of Gamification, Left Brain (Extrinsic Tendency) vs. Right Brain (Intrinsic Tendency) Drives, White Hat vs Black Hat Gamification, How to Apply Level 1 Octalysis to Actual Systems, Introduction to Level II Octalysis, Casestudies.	12
6	**Developing Thinking:** Re-framing Context: Communicology, Apparatus, and Post-history, Concepts Applied to Video games and Gamification, Rethinking 'playing the game' with Jacques Henriot, To Play Against: Describing Competition in Gamification, Player Motivation: Powerful Human Motivators, Why People Play, Player types, Social Games, Intrinsic verses Extrinsic Motivation, Progression to Mastery. Case studies for Thinking: Tower of Hanoi.	05

Suggested Lab Experiments:

1. Analyze a game and describe it in terms of its core elements, game mechanics, rules.
2. Gamification Definition Video: Create a video, animation, or screencast up to ten minutes long, which explains the concept of gamification. Imagine you are describing to a friend or relative what this course is about, and why it’s an important topic. To the extent possible, anticipate and address possible misunderstandings. Humor and creativity are encouraged.
3. Spend some time playing a casual online/mobile game, such as Candy Crush Saga, Clash of Clans, or Words

with Friends. (These are just examples; it can be any game of your choosing, so long as you didn’t already use it for a prior assignment.) Analyze the techniques the game uses to motivate players to participate, and to keep playing. Are they effective? Why or why not?

4. Identify two games. Do a comparative analysis that explains which system you think is most successful, and why. Give specific examples of design aspects that you find effective or ineffective.
5. Casual Games: Spend some time playing a casual online/mobile game, such as Candy Crush Saga, Clash of Clans, or Words with Friends. (These are just examples; it can be any game of your choosing.) Answer the following questions, drawing on the concept discussed in the course: Is the game fun? Why or why not? What could a business learn from this game?
6. Application Comparison: Compare the use of gamification in two of the four application categories, viz., Marketing, Workplace, Learning, Behavior Change. How would a successful gamification system differ in the two situations, and how would it be similar? In which do you think gamification can be more effective?
7. Mini Project.

Books Recommended:

Text books:

1. Chou, Yu-kai. “Actionable gamification: Beyond points, badges, and leaderboards”, Packt Publishing Ltd, 2019.
2. Katie Salen and Eric Zimmerman, “Rules of Play: Game Design Fundamentals”, MIT Press, 2003.
3. Ernest Adams, “Fundamentals of Game Design”, 3rd Edition, New Riders, 2013.
4. Mathias Fuchs, Sonia Fizek,Paolo Ruffino, Niklas Schrape, “Rethinking Gamification”, Meson Press, 2014.
5. Gabe Zichermann, Christopher Cunningham, “Gamification by Design”, O’Reilly, 2011.
6. Byron Reeves and J. Leighton Read, “Total Engagement: Using Games and Virtual Worlds to Change the Way People Work and Businesses Compete”, Harvard Business Press, 2009.
7. Kevin Werbach and Daniel Hunter, “For the Win: How Game Thinking Can Revolutionize Your Business”, Wharton Digital Press, 2012.

Online References:

1. Scott Nicholson, “A User-Centered Theoretical Framework for Meaningful Gamification”, Proceedings of the 8th Games Learning and Society Conference, 2012.
2. B.J. Fogg, “A Behavior Model for Persuasive Design”, Proceedings of the 4th International Conference on Persuasive Technology, ACM, 2009.
3. Joey Lee and Jessica Hammer, “Gamification in Education: What, How, Why Bother?”, Academic Exchange Quarterly 15.2, 2011.
4. Steffen P. Walz and Sebastian Deterding, eds., “The Gameful World: Approaches, Issues, Applications”, MIT Press, 2015.
5. Juho Hamari, Vili Lehdonvirta, “Game Design as Marketing: How Game Mechanics Create Demand for Virtual Goods,” International Journal of Business Science and Applied Management, 2010.
6. Roger E. Pedersen, “Game Design Foundations”, 2nd Edition, Jones & Bartlett Learning, 2009.

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

Course: Predictive Analytics (DJS22ITC8013)
Course: Predictive Analytics Laboratory (DJS22ITL8013)

Pre-requisite: Knowledge of Statistical Analysis and Business Analytics.

Course Objectives: The objective of this course is to gain understanding of the computational foundations in Data Science and develop critical inferential thinking.

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

1. Apply prediction modeling techniques to interpret data into actionable insights.
2. Select a suitable model to carry out the prediction.
3. Evaluate and compare various models to enhance predictive accuracy.

Detailed Syllabus: (unit wise)

Unit	Description	Duration
1	**Introduction to Predictive Analytics:** Introduction to Analytics, Analytics Overview, Why Analytics, Predictive Analytics, Why Do We Need Analytics? Game-Changing Innovations, Types of Analytics, Predictive Analytics, Predicting, Predicting Binary Outcomes, Examples, Trees and Other Predictive Models, Data Sources, Parametric vs. Non-Parametric Models, Predictive Analytics vs. Business Intelligence.	07
2	**Simple Linear Regression:** Simple Linear Regression (SLR), Regression Analysis, Types of Regression, SLR Model Evaluation, SLR Estimation & Prediction, Applications of SLR, Demonstration of SLR Using Python, Categorical Predictors, Unveiling Categorical Predictors, Nominal vs. Ordinal Categorical Predictors, The Power of Prediction, Data Visualization, Data Transformations, Choosing the Right Tool, Beyond Normality, Model Building, Common Model Building Techniques, Crafting a Model, Influential Points, Key Characteristics of Influential Points, Ridge and Lasso Regression.	07
3	**Multiple Linear Regression:** Multiple Linear Regression, Introduction to Multiple Regression, Characteristics of Multiple Regression, Interpretation of Multiple Regression Coefficients, Partial and Part Correlation, Hypothesis Testing, Partial F-test and Variable Selection Method, Dummy Variables, Interaction Variables in MLR, MLR Estimation and Assumptions, MLR Model Building, MLR Model Deployment, Multicollinearity and Other Regression Pitfalls, Model Diagnostics, Shrinkage of Regression Coefficients and Predictive Analysis.	08
4	**Multivariate Analysis and Model Evaluation:** Introduction to Multivariate Analysis, Types of Multivariate Analysis, Example of Multivariate Analysis: Factor Analysis, Correlation Matrix, Extraction of Factors, Factor Loadings, Interpretation of Factors. Model Evaluation and Deployment, Model Validation, Rule Induction Using CHAIO, Automating Models for Categorical and Continuous targets, Comparing and Combining Models, Meta-Level Modeling, Assessing Model Performance, Updating a Model.	07
5	**Time Series Analysis:** Time Series Analysis, Introduction to Time Series, Time Series Objects, Trends and seasonality variation, Decomposition of time series, Time series with deterministic components, Seasonal models, Smoothing and decomposition, Autocorrelation, Exponential smoothing, Holt-Winters method, Forecasting, Stationarity, Moving average models, Autoregressive moving average (ARMA) model.	05

6	**Time Series Analysis and LLMs:** LLMs for Predictive Modeling, Training LLMs for Time Series, LLMs for Forecasting, Prompt Engineering for Time Series Analysis: Techniques for Effective Prompts, Financial forecasting using LLMs, Application of LLMs in Financial Forecasting.	**05**

Suggested Lab Experiments:

(Tools: SAS and SPSS):

1. To perform Analysis on a given dataset using SAS.
2. To perform basic operations (Sorting file, split file, Compute, Recode, Select cases) on dataset in SPSS.
3. To implement PCA (Principal Component Analysis) for dimensionality reduction.
4. To implement Multiple Linear Regression (MLR) using R on a given dataset and predict next values.
5. To implement Stepwise Regression in R and SAS.
6. To perform ROC (Receiver Operating Characteristics) Curve analysis using R.
7. To implement Time series analysis and decomposition using a given dataset for predicting trend, seasonal, and random components.
8. To implement forecasting using Exponential Smoothing, Holt-Winters.
9. To implement a neural network model for forecasting.
10. To train a model using hold-out and cross-validation techniques. Analyze performance stability across folds and explore overfitting tendencies with each approach.
11. To apply CHAID on a dataset with categorical outcomes (e.g., customer segmentation) to identify distinct decision rules.
12. To explore LLMs for financial forecasting using prompts for time series data and predicting future financial trends.

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, Pallavi Chavan, “Predictive Analytics with SAS and R: Core Concepts, Tools, and Implementation”, 1st Edition, Apress springer, 2024.

Reference Books:

1. Walter R. Paczkowski, “Predictive and Simulation Analytics: Deeper Insights for Better Business Decisions”, Springer, 2023.
2. Kattmuri S. Sarma, “Predictive Modeling with SAS Enterprise Miner: Practical Solutions for Business Applications”, 2nd Edition, 2016.
3. Edward W. Frees, Glenn Meyers, Richard A. Derrig, “Predictive Modeling Applications in Actuarial Science” Volume 2, Case Studies in Insurance (International Series on Actuarial Science), Cambridge press, 2016.
4. 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

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

Course: Advanced Machine Learning (DJS22ITC8014)
Course: Advanced Machine Learning Laboratory (DJS22ITL8014)

Pre-requisite: Knowledge of data mining, machine learning, statistics.

Course Objectives: This course will provide students with a solid foundation for both applying and optimizing machine learning to complex real world problems.

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

1. Select an appropriate machine learning model to address real-world scenarios.
2. Apply natural language processing techniques to solve real-world problems.

Detailed Syllabus: (unit wise)		
Unit	**Description**	**Duration**
1	**Reinforcement Learning**: Definitions, key concepts (agent, environment, actions, rewards), Markov Decision Processes (MDPs), Tabular & approximate solution methods: dynamic programming, Monto-Carlo Methods, Q-learning and SARSA Model, Deep Q-Networks (DQNs), Deep Deterministic Policy Gradient (DDPG), Exploration vs. Exploitation: Epsilon-greedy, Temporal difference learning, eligibility traces; planning and learning; practical applications of Reinforcement Learning (AlphaGo Zero, Robotics and Industrial Automation).	**06**
2	**Generative Adversarial Networks (GANs):** Introduction to GANs, Components-Generator and Discriminator, Adversarial training process. Issues and challenges in GANs: Mode collapse, training instability, evaluation metrics (Inception Score, FID). **Variants of GANs**: Conditional GANs, CycleGANs, StyleGANs, and their applications.	**08**
3	**Graph Neural Networks:** Basic graph concepts and graph terminologies, architecture of GNNs, Graph Convolutional Networks (GCNs): Supervised vs unsupervised learning on graph data, Graph Convolution Operation, Graph Attention Networks (GATs), higher-order GNNs.	**08**
4	**Natural Language Processing:** Introduction, Text preprocessing, Language models, text classification, sentiment analysis, named entity recognition, Transformers and BERT: Architecture of transformers, pre-training, and fine-tuning BERT, Prompt Engineering, Techniques for designing effective prompts for model interaction and performance optimization.	**06**
5	**Transfer Learning:** Basics, pre-trained Model Approach, Freezing, Fine-tuning. Transfer Learning Strategies: Inductive Learning, Inductive Transfer, Transductive Transfer Learning, Unsupervised Transfer Learning; Types of Deep Transfer Learning: Domain Adaptation, Domain Confusion, One-shot Learning, Zero-shot Learning, Multitask Learning; Types of Transferable Components: Instance transfer, Feature-representation transfer, Parameter transfer, Relational-knowledge transfer; Transfer Learning Challenges: Negative Transfer, Transfer Bounds; Applications: Transfer learning for NLP/ Audio/ Speech/ Computer Vision.	**07**
6	**Instance-Based Learning:** Learning Vector Quantization (LVQ), Locally Weighted Learning (LWL), and case-based reasoning (CBR). **Explainable AI (XAI)**: Importance of interpretability, methods for model explainability (LIME, SHAP).	**04**

	Federated Learning: Overview, architecture, and challenges of training models on decentralized data. **Self-Supervised Learning:** Introduction, methods and techniques. Introduction to Neuro-Symbolic AI and Causal Analytics.	

Suggested Lab Experiments:

1. Apply Reinforcement Learning for Solving Control Tasks using Q-Learning and DQN.
2. Demonstrate the working of GAN on any suitable dataset.
3. Employ Graph Neural Networks to solve any real-world problems.
4. Implement Sentiment Analysis/ Question Answering using Natural Language Processing.
5. Perform Transfer Learning for NLP/ Audio/ Speech/ Computer Vision.
6. Implement Explainable AI to solve problems in various domains.
7. Mini Project based on the concepts relevant to the syllabus.

Books Recommended:

Text books:

1. Laura Graesser Wah Loon Keng, "Foundations of Deep Reinforcement Learning", 1st Edition, Pearson Education, 2020.
2. Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012.
3. Ian Goodfellow, Yoshua Bengio and Aaron Courville, "Deep Learning", MIT Press, 2016.

Reference Books:

1. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", 2nd Edition, O'Reilly Media, Inc, 2019.
2. Andrew W. Trask, Grokking, "Deep Learning", Manning publication, 2019.
3. François Chollet, "Deep Learning with Python", Manning publication, 2017.
4. Charu C. Aggarwal, "Neural Networks and Deep Learning: A Textbook", Springer; 1st Edition, 2018.
5. Umberto Michelucci, "Applied Deep Learning. A Case-based Approach to Understanding Deep Neural Networks", Apress, 2018.

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

Course: Advanced Security (DJS22ITC8015)
Course: Advanced Security Laboratory (DJS22ITL8015)

Prerequisites: 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 completion of the course, learner will 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.	**06**
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.	**06**

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.	**06**

Suggested Lab Experiments:

1. Conduct various forms of security assessment using different tools available in Kali Linux.
2. Analyse the CIA Triad in various case studies.
3. Implement Bell-LaPadula and Biba models in a virtual environment.
4. Demonstrate attack techniques like phishing and DDoS. using tools like LOIC (Low Orbit Ion Cannon).
5. Implement QKD protocols and key exchange scenarios using simulation software.
6. Analyze the weaknesses of DES or AES using open source tools like Wireshark, SageMath etc.
7. Perform vulnerability scans on a virtual network using tools like Nessus.
8. Simulate SOC operations using open-source tools like ELK Stack for incident response monitoring and analysis.
9. Apply ISTQB security testing processes by designing security test cases for any software application.
10. Conduct a digital forensics investigation on a simulated breach using forensic tools like FTK Imager or Autopsy.
11. Configure a SIEM solution using a tool like Splunk to collect logs, set up alerts, and monitor security events.
12. Case study - Analysis of a real-world cybercrime case and its legal implications.

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

Books Recommended:

Text Books:

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. in Information Technology **Final Year** **Semester: VIII**

Course: Project Stage - II (DJS22ITP803)

Course Objectives: To introduce the students to professional engineering practice by providing them with an opportunity to work on an open-ended engineering problem. Typically, the students would apply knowledge from different areas or courses, which they have studied in their curriculum using methods, tools, and techniques, which they learned to a real-world scenario. Students would have to apply not only their engineering knowledge and proficiencies (hard skills), but also to demonstrate their competence in generic, professional skills (soft skills). It also emphasizes the importance of life-long learning as a fundamental attribute of graduate engineers.

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

1. Develop the proposed solution using appropriate techniques.
2. Test the developed system for its correctness using appropriate techniques.
3. Work effectively as a member of the team.

Guidelines

1. The department must allocate 2 days in the Semester VIII every week.
2. Students will do coding and testing in Semester VIII.
3. Each group along with its guide/mentor shall identify an appropriate technique/s for testing the developed system.
4. The project assessment for term work will be done at least two times at department level by giving presentation to panel members which consist of at least three (3) members as Internal examiners (including the project guide/mentor) appointed by the Head of the department of respective Program.
5. A report is to be prepared summarizing the findings of the literature survey, coding and testing.
6. Every team must publish their work in national / international conference/journals (if possible, publish in Scopus indexed journals).

Evaluation Scheme:

Semester End Examination (A):

Laboratory:

1. Each group will be jointly evaluated by a team of Internal and External Examiners approved by the University of Mumbai.
2. An oral exam will be conducted on the project done by the students.

Continuous Assessment (B):

Laboratory: (Term work)

1. Each team must give a presentation/demo to the Internal Panel consisting of 3 domain experts.
2. Each team will prepare a report that will summarize the results of the literature survey, coding and testing as a product in SEM VIII. The list of papers surveyed must be clearly documented.

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

i. Term Work shall consist of full Project-II on above guidelines/syllabus.
ii. The final certification and acceptance of term work will be subject to satisfactory performance and upon fulfilling minimum passing criteria in the term work.