B.Tech.(Computer Engineering)

Programme Curriculum (2021 Course)

Semester- VII

Artificial Intelligence

TEACHING SCHEME		EXAMINATION SCHEME		CREDIT SCHEME	
	Hours/Week		**Marks**		**Credits**
Lecture:	3 Hours/Week	**University Examination**	**60 Marks**	**Theory**	03
Practical:	2 Hours/Week	**Internal Assessment**	**40 Marks**	**Practical**	01
		Term Work	**25 Marks**		
		Practical	**25 Marks**		
		Total	**150 Marks**	**Total**	04

Course Objective:

To make student aware of basics of Artificial Intelligence (AI), Knowledge representation methods, learning concept and basics of artificial neural network.

Prerequisite:

Data structures, Algorithmic Strategies, Discrete Mathematics

Course Outcomes: On completion of the course, students will have the ability to:

7. Describe the challenges in developing AI based systems
8. Apply appropriate problem-solving strategy to solve a particular problem
9. Use appropriate knowledge representation method
10. Describe components of planning system
11. Apply the various knowledge representation strategies
12. Describe the plan generation systems

Unit I: Artificial Intelligence and its Issues **06 Hours**

Definitions - Importance of AI, Evolution of AI - Applications of AI, Classification of AI systems with respect to environment, Knowledge Inferring systems and Planning, Uncertainty and towards Learning Systems.

Unit II: Problem Solving and Search Strategies **06 Hours**

Problem solving by Search, Problem space - State space, Blind Search - Types, Performance measurement. Breadth First Search, Depth First Search, Depth Limited Search, Iterative Deepening Depth First Search, Bidirectional Search ,Comparison of Uninformed search Strategies, Searching with partial information,Sensor-lessproblems,Generate&test,HillClimbing,BestFirstSearch,A*andAO*Algorithm,Constraintsatisfaction,Game playing Minimax Search, Alpha-Beta Cutoffs, Waitingf or Quiescence

Unit III Knowledge Representation and Reasoning **06 Hours**

Logical systems Knowledge Based systems, Propositional Logic Constraints, Predicate Logic First Order Logic, Inference in First Order Logic, Ontological Representations and applications, Overview Definition of uncertainty, Bayes Rule Inference, Belief Network, Utility Based System, Decision Network.

Unit IV Planning **06 Hours**

Basic plan generation systems, Components of planning system Advanced, Strips plan generation systems, K-strips strategic explanation Planning with state-space

search – partial-order planning – planning graphs – planning and acting in the real world

Unit V : Learning **06 Hours**

Learning: Learning concept, Supervised and unsupervised learning, Learning from observation - Inductive learning – Decision trees – Explanation based learning – Statistical Learning methods - Reinforcement Learning

06 Hours

Unit VI :

basic, comparison of human brain and machine, biological neuron, general neuron model, activation functions, Perceptronlearningrule,applicationsandadvantagesofneuralnetworks

Textbooks

1. Deepak Khemani , "Artifitial Intelligence", Tata Mc Graw Hill Education ,2013
2. G. Luger, "Artificial Intelligence: Structures and Strategies for complex problem solving", Fourth Edition, Pearson Education

Reference Books

J. Nilsson, "Artificial Intelligence: A new Synthesis", Elsevier Publishers.

List of Practical Experiments

1. Apply A* and A* algorithm and solve a search problem.
2. Write a program to conduct uninformed and informed search
3. Represent the knowledge for the given problem
4. Write a program to solve water-jug problem.
5. Write a menu driven program to display set of questions to user and give answer of selected question
   .For Ex.Menu will have following set of questions
   i). Who likes apple?
   ii) Does anybody like apple?
   iii) Is it true that nobody likes apple?
   iv) Who likes apple as well as enjoys playing cricket and piano?
   v) Does anybody play at least one instrument?
   vi) Who likes to play chess, drink buttermilk but does not play any instrument?
   vii) Who share at least one hobby and at least one instrument?
   viii) Who are the persons sharing common instruments but no hobbies are in common?
   Ask the user to enter the question no. The programs should display the answer.
6. Write a program to solve travelling salesman problem.
7. Write a program to solve monkey-banana problem
8. Write a program to solve tower of Hanoi problem.
9. Write a program to solve Tic-Tac-Toe problem.
10. Write a program to solve 8-puzzle problem.

Project Based Learning - Provisional List of Projects

1. Handwritten digits recognition.
2. Lane line detection.
3. Spam classifier
4. Optimal path finder
5. Fire detection and localization using surveillance.

6. Next character predictor
7. Chatbot
8. Game development
9. Recommender system
10. Face recognition
11. Dog and Cat classification

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Virtualisation and Cloud Computing

TEACHING SCHEME		EXAMINATION SCHEME		CREDIT SCHEME	
	Hours/Week		**Marks**		**Credits**
Lecture:	3 Hours/Week	**University Examination**	**60 Marks**	**Theory**	03
Practical:	2 Hours/Week			**Practical**	01
		Internal Assessment	**40 Marks**		
		Term Work	**25 Marks**		
		Oral	**25 Marks**		
		Total	**150 Marks**	**Total**	04

Course Objective:

To understand the various distributed system models and evolving computing paradigms
To gain knowledge in virtualization of computer resources
To realize the reasons for migrating into cloud
To introduce the various levels of services that can be achieved by a cloud.
To describe the security aspects in cloud and the services offered by a cloud.

Prerequisite: Basic knowledge Operating System

Course Outcomes: On completion of the course, students will have the ability to:

1. Ability to understand various service delivery models of a cloud computing
2. Ability to understand the ways in which the cloud can be programmed and deployed.
3. Ability to understand the virtualization and cloud computing concepts
4. Assess the comparative advantages and disadvantages of Virtualization technology
5. Analyse authentication, confidentiality, and privacy issues in cloud computing
6. Identify security implications in cloud computing

Unit I: Cloud Computing Fundamentals **06 Hours**

Cloud Computing Fundamentals: Definition of Cloud computing, Roots of Cloud Computing, Layers and Types of Clouds, Desired Features of a Cloud, Cloud Infrastructure Management, Infrastructure as a Service Providers, Platform as a Service Providers. Computing Paradigms: High-Performance Computing, Parallel Computing, Distributed Computing, Cluster Computing, Grid Computing.

Unit II: Virtualization **06 Hours**

Virtualization: Introduction to Cloud Computing- Cloud issues and challenges – Properties – Characteristics – Service models, Deployment models. Cloud resources: Network and API – Virtual and Physical computational resources – Data-storage. Virtualization concepts – Types of Virtualizations- Introduction to Various Hypervisors – High Availability (HA)/Disaster Recovery (DR) using Virtualization, Moving VMs.

Unit III: Service Models **06 Hours**

Service Models: Infrastructure as a Service (IaaS) – Resource Virtualization: Server, Storage, Network – Case studies. Platform as a Service (PaaS) – Cloud platform & Management: Computation, Storage – Case studies. Software as a

Service (SaaS) – Web services – Web 2.0 – Web OS – Case studies – Anything as a service (XaaS) – Microservices.

Unit IV: Cloud Programming and Software Environments **06 Hours**

Cloud Programming and Software Environments: Cloud Programming and Software Environments – Parallel and Distributed Programming paradigms – Current technologies – Programming support of App Engines – Emerging Cloud software Environment.

Unit V: Cloud Access **06 Hours**

Cloud Access: authentication, authorization and accounting – Cloud Provenance and meta-data – Cloud Reliability and fault-tolerance – Cloud Security, privacy, policy and compliance- Cloud federation, interoperability and standards.

06 Hours

Unit VI: Cloud Technologies And Advancements And SLA Management

Hadoop – MapReduce – Virtual Box -- Google App Engine – Programming Environment for Google App Engine – Open Stack – Federation in the Cloud – Four Levels of Federation – Federated Services and Applications – Future of Federation.

SLA Management in cloud computing: Traditional Approaches to SLO Management, Types of SLA, Life Cycle of SLA, SLA Management in Cloud.

Textbooks

1. Cloud Computing Principles and Paradigms, by Rajkumar Buyya.
2. Essentials of cloud Computing: K. Chandrasekhran, CRC press, 2014
3. Michael Miller, Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, Que Publishing, August 2008.
4. Cloud Computing, A Practical Approach, Anthony T Velte, Toby J Velte, Robert Elsenpeter, TMH

Reference Books

1. Cloud Computing: A Practical Approach, Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Tata McGraw Hill,rp2011.
2. Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, George Reese, O'reilly, SPD,rp2011.
3. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Kumaraswamy, Shahed Latif, O'Reilly, SPD, rp2011

List of Assignments

1. Installation and configuration of own Cloud.
2. Implementation of Virtualization in Cloud Computing to Learn Virtualization Basics, Benefits of Virtualization in Cloud using Open Source Operating System.
3. Study and implementation of infrastructure as Service using Open Stack.
4. Write a program for Web feed using PHP and HTML.
5. Write a Program to Create, Manage and groups User accounts in own Cloud by Installing Administrative Features.
6. Design and develop custom Application using Salesforce Cloud
7. creating an AMI for Hadoop and implementing short Hadoop programs on the Amazon Web Services platform.
8. Creating an Application in SalesForce.com using Apex programming Language

9. Design an Assignment to retrieve, verify, and store user credentials using Firebase Authentication, the Google App Engine standard environment, and Google Cloud Data store.

Project Based Learning - Provisional List of Projects

1. Data Science Assignment Help in Microsoft Azure Specify the necessary environment as a Docker file.
2. Cloud based VM resources for application hosting
3. Configurable deployment of cloud applications using the Docker container
4. Big Data analytics on unstructured text data using Microsoft Azure.
5. Hadoop and MapReduce in Microsoft HDInsight.
6. Azure Machine Learning for sentiment analysis
7. Cloud Computing Mashup/Docker Project
8. Deployment to a publicly hosted Linux VM [Azure or AWS will be appropriate here.]

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Scalable Computing

TEACHING SCHEME		EXAMINATION SCHEME		CREDIT SCHEME	
	Hours/Week		**Marks**		**Credits**
Lecture:	3 Hours/Week	**University Examination**	**60 Marks**	**Theory**	03
Practical:	2 Hours/Week			**Practical**	01
		Internal Assessment	**40 Marks**		
		Term Work	**25 Marks**		
		Oral	**25 Marks**		
		Total	**150 Marks**	**Total**	04

Course Objective: - This course covers computer systems topics that are essential for students engaging in computational and data sciences. It introduces topics on architecture, OS and data structures. Learn advanced topics on tree/graph data structures, HPC/GPGPU programming and Big Data platforms.

Prerequisite: Basic knowledge of computer systems, data structures and programming, and algorithms. However, the course will have a rapid pace and students are expected to pick up the skills rapidly through self-learning.

Course Outcomes: On completion of the course, students will have the ability to:

1. Understand challenges in efficient execution of large-scale parallel applications
2. Identify Big Data and its Business Implications
3. Understand Grid computing services and its practical use
4. Understand data center network and parallel code analysis
5. Identify techniques for self-updating
6. To have knowledge of the fundamentals of Green Computing.

Unit I : **Parallel Computing** **06 Hours**

Why parallel computing? Shared memory and distributed memory parallelism, Amdahl's law, speedup and efficiency, supercomputers.
Scalability, benchmarking, performance modeling, impact of network topologies, parallel code analysis and profiling.

Unit II : **Big data Computing** **06 Hours**

Types of Big Data, Design goals of Big Data platforms, and where in the systems landscape these platforms fall. Distributed programming models for Big Data, including Map Reduce, Stream processing and Graph processing. Runtime Systems for Big Data platforms and their optimizations on commodity clusters and Clouds. Scaling data Science algorithms and analytics using Big Data platforms.

Unit III : Grid Computing **06 Hours**

Introduction to Open Grid Services Architecture (OGSA) – Motivation – Functionality Requirements – Practical & Detailed view of OGSA/OGSI – Data intensive grid service models – OGSA services.

Unit IV: Data Center Computing **06 Hours**

Introduction of data centre, Core components of data centre, data centre infrastructure, types of data centre, Key parameters of data centre, data center network designs, software-defined networks (SDN), virtualization technologies, data center security, traffic engineering, resource management, and green data centers.

Unit V: Automatic Computing **06 Hours**

Autonomic Computing: Objective, element of automatic computing- Self-organizing network. Self-healing systems. Self-optimization. Characteristic of automated computing, component of computing, four areas of automatic computing (self-configuration, self-healing (error correction), self-optimization (automatic resource control for optimal functioning) and self-protection (identification and protection from attacks in a proactive manner) defined by IBM. Example of Automatic Computing.

Unit VI: Green Computing **06 Hours**

Green IT Fundamentals: Business, IT, and the Environment – Green computing: carbon foot print, scoop on power – Green IT Strategies: Drivers, Dimensions, and Goals – Environmentally Responsible Business: Policies, Practices, and Metrics.

Textbooks

1. Data-Intensive Text Processing with MapReduce, Jimmy Lin and Chris Dyer, 1st Edition, Morgan & Claypool Publishers, 2010
2. Kai Hwang, Geoffery C. Fox and Jack J. Dongarra, "Distributed and Cloud Computing: Clusters, Grids, Clouds and the Future of Internet", First Edition, Morgan Kaufman Publisher, an Imprint of Elsevier, 2012.
3. Bhuvan Unhelkar, ―Green IT Strategies and Applications-Using Environmental Intelligence‖, CRC Press, June 2014.
4. Peter S Pacheco, An Introduction to Parallel Programming, Morgan Kaufmann, 2011.

Reference Books

1. Dirk Deroos et al., Hadoop for Dummies, Dreamtech Press, 2014.
2. JL Hennessy and DA Patterson, Computer Architecture: A Quantitative Approach, 4th Ed., Morgan Kaufmann/Els India, 2006.
3. Alin Gales, Michael Schaefer, Mike Ebbers, ―Green Data Center: steps for the Journey‖, Shroff/IBM rebook, 2011.

List of Assignments

1. Basics of MPI (Message Passing Interface)
2. Study of MPI collective operations using 'Synchronization'
3. Installation of Hadoop: File Management tasks in Hadoop
4. Case study on Task Assignment for grid computing.
5. Google's Green Data Centers: Network POP Case Study
6. The Case for Automated Planning in Autonomic Computing

7. Green Computing case study: calls for proposing solutions for the Arabian Gulf Oil Company.

Project Based Learning:

1. Applied Parallel Programming Languages
2. Network simulator for parallel computing
3. Big Data for cybersecurity.
4. Health status prediction.
5. Anomaly detection in cloud servers.
6. Recruitment for Big Data job profiles
7. Sound-Based Computer Automation Using Python
8. Home Automation System Using a Simple Android App

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Elective I: Software Testing & Quality Assurance

TEACHING SCHEME		EXAMINATION SCHEME		CREDIT SCHEME	
	Hours/Week		**Marks**		**Credits**
Lecture:	4 Hours/Week	**University Examination**	**60 Marks**	**Theory**	04
Tutorial:	1 Hours/Week			**Tutorial**	01
		Internal Assessment	**40 Marks**		
		Total	**100 Marks**	**Total**	05

Course Objective:
Students will learn: Introduce basic concepts of software testing.
Understand white box, block box, object oriented, web based and cloud testing.
Know in details automation testing and tools used for automation testing.
Understand the importance of software quality and assurance software systems development.

Prerequisite:
Programming Language, DBMS, JavaScript and HTML 5.

Course Outcomes: On completion of the course, students will have the ability to:
7. Understand fundamental concepts in software testing such as manual testing,
8. Design and Develop project test plan, design test cases, test data, and conduct test
9. Apply recent automation tool for various software testing for testing software.
10. Understand fundamental concepts of Selenium IDE, Selenium RC, Selenium Web driver, Selenium Grid.
11. Apply different approaches of quality management, assurance, and quality standard.
12. Apply and Analyze effectiveness Software Quality Tools

Unit I: Testing methodology **08 Hours**

Goals of Software Testing, Software Testing 10 Methodology Definitions, Model for Software Testing, Effective Software Testing vs Exhaustive Software Testing, Software Failure Case Studies, Software Testing Terminology, Software Testing Life Cycle (STLC), Software Testing methodology, Verification and Validation, Verification requirements, Verification of high level design, Verification of low level design, validation.

Unit II: Testing techniques **08 Hours**

Dynamic Testing: Black Box testing: boundary value analysis, equivalence class testing, state table based testing, cause-effect graphing based testing, error guessing. White box Testing Techniques: logic coverage criteria, basis path testing, graph matrices, loop testing, data flow testing, mutation testing. Static Testing. Validation Activities: Unit validation, Integration, Function, System, Acceptance Testing. Regression Testing: Progressive vs. Regressive, regression testing produces quality software, regression testability, objectives of regression testing, regression testing types, define problem, regression testing techniques.

Unit III : Test Management **08 Hours**

Test organization, Test planning, Detailed Test design and Test Process specification. Software Metrics: need, definition and classification of software matrices. Testing Metrics for Monitoring and Controlling the Testing Process:

attributes and corresponding matrics, estimation model for **testing effort**, information flow matrix used for testing, function point and test point. Efficient Test Suite Management, test suite prioritization its type, techniques and measuring effectiveness.

Unit IV: Test Automation and Testing Tools **08 Hours**

Need, categorization, 8 Automation selection and cost in testing tool, guidelines for testing tools. Study of testing tools: WinRunner, QTP, LoadRunner, Test Director and IBM Rational Functional Tester, Selenium etc.

Unit V : Testing Object Oriented Software **08 Hours**

OOT basics, Object- 5 for Specialized oriented testing. Environment Testing Web based Systems: Web based system, web technology evaluation, traditional software and web-based software, challenges in testing for web-based software, testing web-based testing, Testing a data warehouse.

08 Hours

Unit VI : Quality Software Quality Management

McCall's quality factors 3 Management and Criteria, ISO 9126 quality characteristics, ISO 9000:2000, software quality management.

Textbooks

5. William E. Lewis" Software Testing and Continuous Quality Improvement, CRC Press.
6. Dorothy Graham, Erik van Veenendaal, Isabel Evans, Rex Black, Foundations of Software Testing, Cengage Learning.

Reference Books

4. M. G. Limaye, Software Testing: Principles, Techniques and Tools, TMH
5. Paul C. Jorgenson, Software Testing: A Craftsman's Approach, CRC Press

Project Based Learning - Provisional List of Projects

1. Customer Experience Management
2. Android Local Train Ticketing System
3. Android Task Monitoring
4. Android Women Safety App
5. Personality Analysis
6. Online Election System
7. Analyzing sentiments of Facebook Users: A Software System
8. Detecting Evil Applications on Online Social Networks.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Elective I: Mobile Operating System

TEACHING SCHEME		EXAMINATION SCHEME		CREDIT SCHEME	
	Hours/Week		**Marks**		**Credits**
Lecture:	4 Hours/Week	**University Examination**	**60 Marks**	**Theory**	04
Tutorial:	1 Hours/Week			**Practical**	01
		Internal Assessment	**40 Marks**		
		Total	**100 Marks**	**Total**	05

Course Objective: To make students aware of mobile operating system, framework of mobile operating system, security aspects.

Prerequisite: Basic knowledge of Operating System.

Course Outcomes: On completion of the course, students will have the ability to:

1. To study the basic concept of mobile operating system.
2. To implement the concept of multiprogramming.
3. To gain the knowledge of digital certificate and Security.
4. To understand the framework of mobile OS.
5. Learn the concepts of Linux OS.
6. To implement case study

Unit I: Introduction to Mobile Operating Systems **08 Hours**

Brief History of Mobile Operating Systems, OS-Interfaces, Multilevel Views of OS, Categories, Small and Specialized OS, 64-Bit OS, Processes and Threads, System Performance and Models: Performance of Computer Systems, Performance Metrics, Workload and System Parameters, Simulation Models: Types, Discrete-Event Model, Stochastic Model.

Unit II: Multiprogramming **08 Hours**

System with Multiprogramming, Processor Scheduling, Synchronization, Deadlocks, File Management, Memory Management: Process Address Space, Contiguous Memory Allocation, Non Contiguous Memory Allocation, Virtual Memory, Paging with Virtual Memory.

Unit III: Security and Protection **08 Hours**

Components for Security and Protection, Physical Security, User Authentication, Protection, Secure Communications, Digital Certificates, System Vulnerabilities, Invasive and Malicious Software, Defending the System and User, Intrusion Detection Management.

Unit IV: Mobile Ecosystems **08 Hours**

Application Framework, Developing a Mobile Strategy, Mobile Information Architecture, Mobile Design: Elements of Mobile Design, Ubiquity in the Mobile Web, Mobile Web Development.

Unit V: Introduction to Linux **08 Hours**

Command Line Interface, Files and Directories, Shell Variables, Script Files, Connecting a Remote Linux Server. Java Modelling Framework, Java and Posix Threads.

Unit VI: Case Study **08 Hours**

Android SDK, iOS, Windows, Mobile Web Apps vs. Mobile Applications

Textbooks:

1. Jose M Garrido, Richard Schlesinger, Kenneth Hoganson, Principles of Modern Operating Systems.
2. By Brian Fling, Mobile Design and Development: Practical concepts and techniques for Creating Mobile Sites and Web Apps, O'Reilly Publications.

Reference Books:

Mobile Operating Systems, Gerard Blokdyk

Project Based Learning:

1.Understand the file system of Linux.
2.Implement Shell Scripting of Linux.
3. Bus pass management system using Android Studio.
4.Online Book Store: Ecommerce Application.
5. Agri Shop For Farmers Online Shopping Android Application.
6. COVID-19 (corona) Online Test Results & availability booking of Hospitals based Mobile App.
7. Online Voting System Project Application.
8. On Road Vehicle Breakdown Assistance (ORVBA) Finder Project

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Elective I: Fundamentals of Fog & Edge Computing

TEACHING SCHEME		EXAMINATION SCHEME		CREDIT SCHEME	
	Hours/Week		**Marks**		**Credits**
Lecture:	4 Hours/Week	**University**	**60 Marks**	**Lecture:**	04
Tutorial:	1 Hours/Week	**Examination**		**Tutorial:**	01
		Internal Assessment	**40 Marks**		
		Total	**100 Marks**		05

Course Objective:

1.Introduce cloud computing and enabling technologies
2.Explore the need for fog and edge computation
3.Impart the knowledge to log the sensor data and to perform further data analytics

Prerequisite:

1. Principles of Cloud Computing
2. Python Programming
3. Java programming

Course Outcomes:

1. Understand the principles, architectures of fog
2. Understand the communication and management of fogs
3. Understand storage and computation in fogs
4. Design and Implement Internet of Everything (IoE) applications through fog computing architecture computing
5. Analysis the performance of the applications developed using fog architecture
6. Understand the security and privacy issues of fog computing

Unit I: Internet of Things (IoT) and New Computing Paradigms **08 Hours**

Introduction-Relevant Technologies-Fog and Edge Computing Completing the Cloud-Hierarchy of Fog and Edge Computing-Business Models-Opportunities and Challenges

Unit II: Challenges in Federating Edge Resources **08 Hours**

Introduction-Methodology-Integrated C2F2T Literature by Modeling Technique-Integrated C2F2T Literature by Use-Case Scenarios-Integrated C2F2T Literature by Metrics-Future Research Directions

Unit III: Management and Orchestration of Network Slices in 5G, Fog, Edge, and Clouds **08 Hours**

Introduction-Background-Network Slicing-Network Slicing in Software-Defined Clouds-Network Slicing Management in Edge and Fog- Internet of Vehicles: Architecture, Protocol and Security Seven layered model architecture for Internet of Vehicles- IoV: Network Models, Challenges and future aspects.

Unit IV: Optimization Problems in Fog and Edge Computing **08 Hours**

Preliminaries-The Case for Optimization in Fog Computing-Formal Modeling Framework for Fog Computing-Metrics-Further Quality Attributes-Optimization Opportunities along the Fog

Architecture-Optimization Opportunities along the Service Life Cycle-Toward a Taxonomy of Optimization Problems in Fog Computing.

Unit V: Middleware for Fog and Edge Computing: Design Issues **08 Hours**

Need for Fog and Edge Computing Middleware-Design Goals-State-of-the-Art Middleware Infrastructures-System Model-Proposed Architecture-Case Study Example

Unit VI: Technologies in Fog Computing and Applications **08 Hours**

Fog Data Management-Motivating Example: Smart Building-Predictive Analysis with Fog Torch Machine Learning in Fog Computing-Data Analytics in the Fog-Data Analytics in the Fog Architecture-Configurations Tracking-Fog Computing Model for Evolving Smart Transportation Applications-Testing Perspectives of Fog-Based IoT Applications-Legal Aspects of Operating IoT Applications in the Fog

Textbooks:

1. Buyya, Rajkumar, and Satish Narayana Srirama, eds, Fog and edge computing: principles and paradigms, 1st edition, John Wiley & Sons, 2019.
2. John Mutumba Bilay , Peter Gutsche, Mandy Krimmel and Volker Stiehl ,SAP Cloud Platform Integration: The Comprehensive Guide, 2nd edition, Rheinwerg publishing, 2019

Reference Books

1. Bahga, Arshdeep, and Vijay Madisetti. Cloud computing: A hands-on approach, 1st edition, CreateSpace Independent Publishing Platform, 2013.
2. Ovidiu Vermesan, Peter Friess, Internet of Things –From Research and Innovation to Market Deployment, 1 st edition,River Publishers, 2014
3. Michael Missbach, Thorsten Staerk, Cameron Gardiner, Joshua McCloud, Robert Madl, Mark Tempes, George Anderson, SAP on Cloud, 1 st edition, Springer, 2016

Project Based Learning:

1. Scheduling for Deep Reinforcement Learning-Based Offloading in Vehicle Edge Computing
2. Multilevel vehicular edge-cloud computing networks with advanced deep learning-based computational offloading
3. In Wireless Metro Area Networks, Optimal Cloudlet Location and User to Cloudlet Allocation
4. Joint Management and Cloud Unloading for Mobile Applications at the Optimal Level
5. Mobile Cloud Computing: Distributed Mega Pricing for Effective Application Offloading
6. An Edge Node Resource Management Framework
7. Increasing the Reliability of Cloud Services by Using a Proactive Fault-Tolerance Approach
8. Task assignment for mobile edge computing that considers user mobility
9. Deadline-Aware Portable Edge Computing Systems Task Scheduling
10. A Privacy-Preserving Data Gathering Scheme for IoT Applications Assisted by Mobile Edge Computing

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Elective I: System Thinking

TEACHING SCHEME		EXAMINATION SCHEME		CREDIT SCHEME	
	Hours/Week		**Marks**		**Credits**
Lecture:	4 Hours/Week	**University**	**60 Marks**	**Lecture:**	04
Tutorial:	1 Hours/Week	**Examination**		**Tutorial:**	01
		Internal Assessment	**40 Marks**		
		Total	**100 Marks**		05

Course Objective:
Systems thinking is a holistic approach to analysis that focuses on the way that a system's constituent parts interrelate and how systems work over time and within the context of larger systems.

Prerequisite: Basic knowledge of System Thinking

Course Outcomes: On completion of the course, students will have the ability to:

1. Provide an overview of the history, research and perspectives into systems thinking
2. Understand and document system thinking objectives
3. Establish a basic understanding of systems thinking terminology, theories, processes, methods, language and tools.
4. Evaluate when it is appropriate to apply thinking methods, i.e. reductionist methods (ex. data collection, scientific method, etc.) as opposed to applying systems thinking methods (ex. ,Systems Engineering, Breakthrough Thinking/Smart Questions, etc.)
5. Describe and model solutions that will enable system thinking ex. (mind maps, feedback & causal loops, behaviour over time diagrams
6. Apply systems engineering and analysis techniques to various problems. (socio - technical, supply chain, value chain / lean, etc.)

Unit 1 - Introduction to System Thinking **08 Hours**

Introduction: Definitions & Concepts, What is Systems Thinking?, The importance and Purpose & Principles of Systems Thinking, Systems Thinking tools and techniques, The Systems Thinking Process/Protocol for Business, Applying Systems Thinking to Engineering, System Thinking in Management.

Unit II - System Engineering, Data Analytics and System Thinking **08 Hours**

The fourth industrial revolution, Integrating Reliability Engineering with System Engineering, Software Cybernetics, Using Modeling & Simulations ,Risk Management, An Integrated Approach to Safety & Security Based on Systems Theory, Applied system Thinking, The System in System Thinking Applied System Thinking Approach, Soft System Methodology, Systemigram, Casual Loop Diagrams, Intervention Points, Approach , Tools & Methods- Final Thoughts

Unit III - System Thinking in Software Engineering (Say Agile Approach) **08 Hours**

Principles of System Thinking for Software Development, The critical role of systems thinking in software development, IT Project Management with System Thinking, Applying system thinking to model-based software engineering.

Unit IV - System Thinking in Project Management **08 Hours**

Systems Thinking for Project Management, The Need for Systems Thinking in Project Management, Systems thinking and its latent potential in project Planning, Systems thinking and its latent, potential in project implementation and control, How to Apply Systems Thinking in Managing Projects, Managing Project Risks, Improving Decisions in Projects, Systems Approaches: Hard Systems Thinking, System Dynamics, Organizational Cybernetics, Complexity Theory, Strategic Assumption Surfacing and Testing Interactive Planning, Soft Systems Methodology, Critical Systems Heuristics, Team Syntegrity, Postmodern Systems Thinking

Unit V - System Thinking for Intelligent Systems **08 Hours**

Engineering Intelligent Systems: Systems Engineering and Design with Artificial Intelligence, Visual Modelling, and Systems Thinking, Artificial Intelligence, Science Fiction, and Fear, The Intelligence in the System: How Artificial Intelligence Really Works, What Is Artificial Intelligence?, Modelling of Intelligent System Thinking in Complex Adaptive Systems, Systems Thinking and AI applications

Unit VI - System thinking future research and Case studies **08 Hours**

Systems thinking and complexity ideas in Research, Key themes and perspectives, A systematic review of the use of key, Methodology, Analysis, Results, Discussion, Limitations, Ways Forward, Conclusion.

Textbooks:

1. Engineering Intelligent Systems, Systems Engineering and Design with Artificial Intelligence, Visual Modelling, and Systems Thinking By Barclay R. Brown · 2022
2. Systems Engineering, Building Successful Systems By Howard Eisner · 2011
3. Systems Engineering in the Fourth Industrial Revolution, Big Data, Novel Technologies, and Modern Systems Engineering

Reference Books

1. The Journey to Enterprise Agility Systems Thinking and Organizational Legacy By Daryl Kulak, Hong Li · 2017
2. Mastering Project Management James P. Lewis.

Project Based Learning:

1. Understand Events, Patterns behaviour, System and Mental Model.
2. Understand and implement key concepts of System Thinking.
3. Create a list of different possible solutions. Evaluate the solutions to see whether they are realistic.
4. Conduct small tests of change to see whether an improvement can be made.
5. To understand the interrelationships within a system's structure.
6. To perform problem solving in complex system.
7. To develop the systems-thinking skills that thought leaders across the globe consider critical for 21st-century life.
8. Understand and implement different levels of systems thinking maturity.
9. Study different types of Tools for Systems Thinkers.
10. Observe events or data, to identifying patterns of behaviour overtime, to surfacing the underlying structures that drive those events and patterns.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Programming Technologies and Tools Laboratory – VI

Teaching Scheme		Examination Scheme		Credit Scheme	
Hours/Week			Marks		Credits
Lecture:	**01 Hours/Week**	**Practical**	**50 Marks**	**Lecture**	**01**
Practical:	**02 Hours/Week**			**Practical**	**01**
		Total	**50 Marks**	**Total**	**02**

Course Objective: To develop applications in various domain by applying programming skills using Python Libraries.

Prerequisite: Understanding of basic python programming and OOPs concepts.

Course Outcomes: On completion of the course, students will have the ability to:

1. To learn how to use regular expression in Python for searching.
2. To develop the ability to write database applications in Python.
3. To develop the skill of data science using python
4. To develop the ability of Data Visualization using Python
5. To develop the skill of designing Graphical user Interfaces in Python•
6. To develop the ability to implement machine learning and deep learning applications.

Unit I : Python Regular Expressions **02 Hours**

RegEx Module, Regexes in Python and Their Uses, The match Function, Find all Function, The search Function ,Matching vs searching ,Search and Replace ,Regular Expression Modifiers

Unit II: Databases with Python **02 Hours**

Python MySQL Database access, Install the MySQL db and other Packages, **Create Database, Create table, Insert, Select, Where, Order By, Delete, Drop Table, Update,** Introduction to Python MongoDB, SQLite.

Unit III : Data Science using Python **02 Hours**

Introduction to NumPy, Installation of NumPy, Creating Arrays, **Array Indexing ,** NumPy **Data Types ,**NumPy **Array Shape ,NumPy Functions ,Basic operations on single array ,Searching and sorting.**

Unit IV: Data Visualization in Python: **02 Hours**

Pandas: Installation of Pandas, Import Pandas, Read CSV Files, Viewing the Data, Pandas **-Cleaning Data,** Removing Duplicates, Difference between Pandas and NumPy, Pandas Time Series

Python Matplotlib: Installation of Matplotlib ,Basic plotting ,Bar Plot ,Line ,Pie chart, Histogram

Unit V: GUI in Python **02 Hours**

Python Tkinter , Widgets, Python Tkinter Geometry, Python Tkinter place() method, Button ,Introduction to PyQT

Unit VI: Machine Learning using Python **02 Hours**

What is Scikit-Learn (Sklearn), Loading data, Splitting, Train Test Data, Introduction to TensorFlow

Textbooks

1. Hands-on Data Analysis and Visualization with Pandas Paperback – 1 January 2020 by Purna Chander Rao. Kathula
2. Machine learning and Deep learning using Python and Tensor flow by Venkata Reddy Konasani and Shailendra Kadre

Reference Books

- MySQL for Python: Database Access Made Easy Kindle Edition by Albert Lukaszewski
- Python GUI Programming with Tkinter, Alan D. Moore, O'Reilly Media, Inc.

List of Laboratory Exercises

- Write a program to implement algorithm that searches for an element in a list.
- Write a program to implement Databases (MySQL, MongoDB).
- Write a program to implement Calculator in Python.
- Write a program to implement GUI in Python.
- Write a program to implement simple graphs using Matplotlib in Python.
- Write a program to generate random numbers for Gaussian distribution using Numpy.

Syllabus for Unit Tests: NA

Project Stage -I

TEACHING SCHEME	EXAMINATION SCHEME	CREDIT SCHEME
		Credits
		Theory 00
Practical: 2 Hrs/Week		**Practical:** 03
	Term Work : 50 Marks	**Term Work**
	Oral : 50 Marks	**Oral:**
	Total 100 marks	**Total** **03**

Course Pre-requisites:
Basics of Software engineering, Software testing and knowledge of core computer engineering subjects.

Course Objectives:

- To provide in depth outline for software project planning.

Course Outcomes: On completion of the course, students will have the ability to:

7. Review and understand how previous experiences had an impact on affective states and intellectual performance.
8. Identify and define the problem.
9. Demonstrate the ability of decision-making to propose solution.
10. Design an algorithm to solve the problem.
11. Demonstrate an ability to work as a team member.
12. Perform requirement analysis process and decide feasibility.

1. The project will be undertaken preferably by a group of at least 3- 4 students who will jointly work and implement the project over the academic year. The work will involve the design of a system or subsystem in the area of Computer Engineering.
2. If the project is chosen a hardware project it will involve the designing a system or subsystem or upgrading an existing system. The design must be implemented into a working model with necessary software interfacing and a user manual.
3. If the project is chosen in the pure Software Application it must involve the detail Software Design Specifications, Data Structure Layout, File Design, Testing with complete documentation and user interface, with life cycle testing and as an executable package.
4. The group will select a project with the approval of the guide (Staff members assigned) and submit the name of the project with a synopsis of 2 or 3 pages in the month of August in the academic year. A preliminary study report by the group must be submitted and certified at the end of seventh Semester.
5. It is expected that at least one research paper is published by each group with guide.
The project report stage-I will contain the details.
Problem definition and requirement specification, acceptance test procedure (ATP).
a) System definition, requirement analysis.
b) System design with UML.
c) Documentation and references.

Documentation will use UML approach with Presentation, Category, Use Case, Class Diagrams, etc.
Project report must be checked for plagiarism from respective guide

Internship

TEACHING SCHEME	EXAMINATION SCHEME	CREDIT SCHEME
		Credits
Practical: --		**Practical:** 03
	Termwork : 25 Marks	
	Oral : 25 Marks	**Termwork**
		Oral
	Total 50 marks	**Total** **03**

Course Pre-requisites:
Professional Skills, Knowledge of core computer engineering subjects.

Course Objectives:

- To provide exposure for the students on technology /tools for software development.in practical engineering fields.
- To identify their skills, values, beliefs, interests and personal abilities for professional growth.

Course Outcomes: On completion of the course, students will have the ability to:

- Propose a solution to solve real world problems with the help of technology.
- Apply software engineering principles.
- Evaluate and compare the various methodologies to solve a real-world problem.
- Demonstrate use of modern software development tools.
- Prepare and present a report on industrial training.
- Identify social and ethical responsibilities and develop skills to compete for lifelong learning.

As a part of the B. Tech Computer Engineering curriculum, Internship is a Practical course, which the students B. Tech Computer Engineering should undergo in reputed Private / Public Sector / Government organization / companies as industrial training of 60 days to be undergone by the student in the summer vacation after the semester VI. Examination and Oral examination will be conducted at the end of the semester VII

The Internship Report:

Internship report should be prepared by each student duly signed by respective guide. The report is expected to demonstrate, development of practical and professional skills in Engineering through technical experience and application of theoretical knowledge. Development of skills in dealing with people, and communication skills form part of the training experience. Students should seek advice from their employers to ensure that no confidential material is included into the report. The student should be able to present the report to prospective employers.

The following should be observed:

- Duration of Internship.
- Preliminary information
- Technical report/diary references should be made in the text to books, technical papers, standards etc., used during the training period and should be listed.
- Finally, a conclusion should include comprehensive comments on the type and value of experience gained, and how this relates to your professional career.
- A copy of the report and training certificate should be submitted to his/her employer, another copy to the Department (through the respective Adviser).
- Students should also retain a personal copy of the report.

B.Tech.(Computer Engineering)

Programme Curriculum (2021 Course)

Semester- VIII

Machine Learning

TEACHING SCHEME		EXAMINATION SCHEME		CREDIT SCHEME	
	Hours/Week		**Marks**		**Credits**
Lecture:	3 Hours/Week	**University Examination**	**60 Marks**	**Lecture:**	3
Practical:	2 Hours/Week			**Practical:**	1
		Internal Assessment	**40 Marks**		
		Term Work	**25 Marks**		
		Practical	**25 Marks**		
		Total	**150 Marks**	**Total**	4

Course Objectives: To provide a strong foundation of Machine Learning concepts and techniques.

Prerequisite:
The students should be aware Discrete Mathematics, Database Management System, Engineering Mathematics, Programming Languages.

Course Outcomes: On completion of the course, students will have the ability to:

1. Explain significance of Machine Learning
2. Understand different paradigms of Machine Learning.
3. Understand various machine learning models.
4. Explain supervised learning algorithms.
5. Explain unsupervised learning algorithms.
6. Tackle real world problems in the domain of machine learning, Information Retrieval.

Unit I Introduction to machine learning: **06 Hours**

Introduction to Learning Systems, Structure of Learning System, Testing vs Training, Learning vs Designing, Goal and Applications of Machine Learning, Examples of Machine Learning Problems, Need of Learning, Machine Intelligence.

Unit II Machine Learning Techniques: **06 Hours**

Types of data in Machine Learning, Structures of data, Introduction Machine Learning Techniques: Supervised Learning (SL), Semi Supervised Learning (SSL), Unsupervised Learning (USL), Data quality and remediation, Data Pre-Processing: Dimensionality reduction, Feature subset selection.

Unit III Machine Learning Models **06 Hours**

Steps to choose Machine Learning Technique, Machine Learning Models with Examples: Linear based Models, Logic Based and Algebraic Models, Probabilistic Models.

Unit IV: Supervised Learning **06 Hours**

What is Supervised Learning? Types of Supervised Learning, Classification: What is Classification? Types: Naive Bayes Classifier, Decision Trees, Support Vector Machines, Rule based Classification, Backpropagation, Associative Classification, Classifier Accuracy Measures, Precision and Recall Measures. Regression and types Types: Linear Regression, Logistic

Regression, Classification vs Regression, Issues Regarding Classification, and Regression, Assessing performance of Regression

Unit V: Unsupervised Learning: **06 Hours**

Introduction to clustering, Types: K Means clustering Algorithm, Mixture Models, Hierarchical Clustering, Anomaly Detection, Neural Networks, Self-Organizing Map (SOM), Applications of Unsupervised Learning.

Unit VI: Trends in Machine Learning **06 Hours**

Ensemble methods for increasing accuracy: Bagging and Boosting, multitask learning, online learning and Sequence prediction, Data Streams and Active Learning, Introduction to Deep Learning and Reinforcement Learning, Case Study: Latest Machine Learning Tools.

Textbooks

1. Machine Learning (McGraw-Hill International Editions Computer Science Series
2. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow
3. Machine Learning: A First Course for Engineers and Scientists, by Andreas Lindholm (Author), Niklas Wahlström (Author), Fredrik Lindsten (Author)

Reference Books

1. T.M. Mitchell, "Machine Learning", McGraw Hill.
2. C.M. Bishop, "Pattern Recognition and Machine Learning", Springer
3. Ethem Alpaydin, "Introduction to Machine Learning"

List of Assignments

1. Introduction to Learning Systems (Structure, Goals, Need, Applications, Examples).
2. Explain in detail steps to choose Machine Learning Algorithm. Differentiate between different Machine Learning Models.
3. Study and implement Decision Tree using R/Python Programming.
4. Study and implement Support Vector Machines using R/Python Programming.
5. What is Regression? Implement Linear Regression using R/Python Programming.
6. Examine Classification and Regression. What are the issues regarding classification and regression.
7. WAP to Implement Handwritten Digit Classification.
8. Study and implement K-Means clustering algorithm.
9. Study and implement Neural Network to Read Handwriting.
10. Case study on SCIKIT-LEARN, WEKA tool for machine learning.

Project Based Learning :

1. Movie Recommendations with Movie lens Dataset.
2. Sales Forecasting with Walmart.
3. Stock Price Predictions.
4. Human Activity Recognition with Smartphones.
5. Wine Quality Predictions.
6. Breast Cancer Prediction.
7. Iris Classification.

8. Movie Recommendations with Movie lens Dataset

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Data Storage Networking

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	04 Hours/Week	University Examination:	60 Marks		
		Internal Assessment:	40 Marks	Lecture	04
		Total	100 Marks	Total	04

Course Objective:
Student should be able to understand the different data storage technologies and able to design data storage solutions for an organization.

Prerequisite: Computer Network

Course Outcomes: On completion of the course, students will have the ability to:

1. Understand the design of a data centre and storage requirements.
2. Use the various types of storage and their properties.
3. Describe physical and virtualization of storage.
4. Understand the backup, archiving regarding recovery and business continuity.
5. Describe the backup/ recovery topologies.
6. Describe local replication and remote replication technologies.

Unit I **08 Hours**

Introduction to Storage Technology: Information storage, evolution ofstorage technology and architecture, data centre infrastructure, key challenges in Managing information, information lifecycle. Storage system Environments: components of storage system environment, Disk Drive components, Disk Drive Performance, fundamental laws governing disk performance, logical components of the host, application requirements and disk performance.

Unit II **08 Hours**

Data Protection: RAID: Implementation of RAID, RAID array components, RAID levels, RAID comparison, RAID Impact on disk performance, host spares. Intelligent Storage System: Components of an Intelligent Storage System, Intelligent Storage array, concepts in Practice: EMC CLARIION and Symmetric.

Unit III **08 Hours**

Direct – Attached Storage and Introduction to SCSI :Types of DAS,DAS benefits and limitations, disk drive interfaces, introduction to parallel SCSI, SCSI command model. Storage Area Networks: fibre channel, The SAN and Its evolution, components of SAN, FC connectivity, Fibre channel ports, fibre channel architecture, zoning, fiber channel login types, concepts in practice: EMC Connectix.

Unit IV **08 Hours**

Network attached storage: general purpose servers vs NAS Devices, benefits of NAS, NAS file I/O, components of NAS, NAS Implementations, NAS file sharing protocols, NAS I/O operations, factors effecting NAS Performance and availability, concepts in practice: EMC Celerra. IP SAN: iscsi, fcip. Content – addressed storage: Fixed content and Archives, types of archives, features and benefits of CAS, CAS Architecture, object storage and retrieval in CAS, CAS Examples, concepts in practice: EMC Centera.

08 Hours

Unit V

Storage Virtualization: Formas of Virtualization, SNIA Storage virtualization taxonomy, storage virtualization configurations, storage virtualization challenges, types of storage virtualization, concepts in practice: EMC In vista, Rainifinity. Introduction to business continuity: information availability, BC terminology, BC planning life cycle, Failure analysis, business impact analysis, BC technology solutions

Unit VI **08 Hours**

concepts in practice: EMC Power path. Backup and recovery: backup purpose, backup considerations, backup granularity, recovery considerations, backup methods, backup process, backup and restore operations, backup topologies, backup in NAS environments, backup technologies, concepts in practice: EMC Networker, EMC Disk Library(EDL).

Textbooks

- Mauricio Arregoces, Data Center Fundamentals, Cisco Press; 1st edition, 2003.
- Robert Spalding, Storage Networks: The Complete Reference, Tata McGraw Hill, Osborne, 2003
- Meeta Gupta, Storage Area Network Fundamentals, Pearson Education Limited, 2002.

Reference Books

- G. Somasundaram, Alok Shrivastava, Information Storage and Management, EMC Education Series, Wiley, Publishing Inc., 2011.
- Gustavo Santana, Data Center Virtualization Fundamentals: Understanding Techniques and Designs for Highly Efficient Data Centres with Cisco Nexus, UCS, MDS, and Beyond, Cisco Press; 1 edition, 2013.

Project Based Learning:

1. Implement different storage technology (File, Block and Object storage).
2. Implement Configuration of RAID on your Computer.
3. Create and Implement Google cloud console account and projects.
4. Study about implementation of EMC CLARIION and Symmetric.
5. Comparing SAN with standalone storages in server
6. Modeling /simulation of FC SAN -- Fibre Channel Storage Area Network
7. Implementation of FCoE – Fibre Channel over Ethernet.
8. Work flow management in cloud storage network.
9. How to model your project with assessments of storage and I/O workload requirements.
10. Cloud federation and also in green data centers.
11. Credential and also trust management in storage networking
12. Dynamic resource (resource as Data Storage) provisioning

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Data Visualization

TEACHING SCHEME		EXAMINATION SCHEME		CREDIT SCHEME	
	Hours/Week		**Marks**		**Credits**
Lecture:	3 Hours/Week	**University Examination**	**60 Marks**	**Lecture:**	03
Practical:	2 Hours/Week			**Practical:**	01
		Internal Assessment	**40 Marks**		
		Term Work	**25 Marks**		
		Practical	**25 Marks**		
		Total	**150 Marks**		04

Course Objective: Students will learn: the value of visualization, specific techniques in information visualization and scientific visualization, and how understand how to best leverage visualization methods.

Prerequisite: Programming Language, DBMS, JavaScript and HTML5

Course Outcomes: On completion of the course, students will have the ability to:

1. Explore various data visualization techniques.
2. Apply appropriate data visualization techniques to provide trends/insights for the dataset.
3. Apply visualization tools / techniques for various data analysis tasks.
4. Apply visualization tools / techniques for large datasets.
5. Apply advanced data visualization tools and techniques.
6. Given the application context for given data set, Design the information Dashboard for access information based on user criteria.

Unit I: Introduction to Data Visualization: **06 Hours**

Acquiring and Visualizing Data, Simultaneous acquisition and visualization, Applications of Data Visualization, Keys factors of Data Visualization (Control of Presentation, Faster and Better JavaScript processing, Rise of HTML5, Lowering the implementation Bar) Exploring the Visual Data Spectrum: charting Primitives (Data Points, Line Charts, Bar Charts, Pie Charts, Area Charts), Exploring advanced Visualizations (Candlestick Charts, Bubble Charts, Surface Charts, Map Charts, Infographics). Making use of HTML5 CANVAS, Integrating SVG.

Unit II: Basics of Data Visualization – Tables: **06 Hours**

Reading Data from Standard text files (.txt, .csv, XML), Displaying JSON content Outputting Basic Table Data (Building a table, Using Semantic Table, Configuring the columns), Assuring Maximum readability (Styling your table, Increasing readability, Adding dynamic Highlighting), Including computations, Using data tables library, relating data table to a chart

Unit III: Visualizing data Programmatically: **06 Hours**

Creating HTML5 CANVAS Charts (HTML5 Canvas basics, Linear interpolations, A Simple Column Chart, Animations), Starting with Google charts (Google Charts API Basics, A Basic bar chart, A basic Pie chart, Working with Chart Animations).

Unit IV: Introduction to D3.js: **06 Hours**

Getting setup with D3, Making selections, changing selection's attribute, Loading and filtering External data: Building a graphic that uses all of the population distribution data,

Data formats you can use with D3, Creating a server to upload your data, D3's function for loading data, Dealing with
Asynchronous requests, Loading and formatting Large Data Sets

Unit V: Advanced Data Visualization: **06 Hours**

Making charts interactive and Animated: Data joins, updates and exits, interactive buttons, Updating charts, Adding transactions, using keys.

Unit VI: Information Dashboard Design: **06 Hours**

Introduction, Dashboard design issues and assessment of needs, Considerations for designing dashboard-visual perception, Achieving eloquence, Advantages of Graphics _Library of Graphs, Designing Bullet Graphs, Designing Sparklines, Dashboard Display Media, Critical Design Practices, Putting it all together - Unveiling the dashboard.

Textbooks:

1. Jon Raasch, Graham Murray, Vadim Ogievetsky, Joseph Lowery, "JavaScript and jQuery for Data Analysis and Visualization", WROX.
2. Ritchie S. King, Visual story telling with D3" Pearson
3. Ben Fry, "Visualizing data: Exploring and explaining data with the processing environment", O'Reilly, 2008.
4. A Julie Steele and Noah Iliinsky, Designing Data Visualizations: Representing Informational Relationships, O'Relly.

Reference Books:

1. Scott Murray, Interactive Data Visualization for Web, O'Relly
2. Nathan Yau, "Data Points: Visualization that means something", Wiley, 2013.

List of Assignments

14. Setup Environment for All the Tools
15. Develop the following Program Using HTML5 CANVAS and SVG TAG
    a.Develop the Different basic Graphical Shapes using HTM5 CANVAS
    b.Develop the Different Advanced Graphical Shapes using HTM5 CANVAS
    c.Develop the Different basic Graphical Shapes using HTM5 SVG
    d.Develop the Different Advanced Graphical Shapes using HTM5 SVG
16. Develop Following Program Using HTML5 and JavaScript
    a.Develop the simple bar chart usingTML5 CANVAS
    b.Read the data .txt file and draw Data Table
    c.Read the data .txt file and draw Simple Bar Chart
    d.Read the data .csv file and draw Data Table
17. Develop Following Program Using HTML5 and JavaScript
    a.Read the data .csv file and draw Column Bar Chart
    b.Read the data XML file and draw Data Table
    c.Read the data XML file and draw Simple Chart
    d.Read JSON Data and draw Data Table
    e.Read JSON Data and draw Simple Chart
18. Develop Following Program Using HTML5 and D3.js and Canvas.js
    a.Showing the data as a column chart (simple)
    b.Showing the data as a stacked column chart
    c.Showing the Data as a column chart for four age group
    d.Showing the data as a Line chart (single, fewer and multiple lines)
    e.Showing the data as a Pie Chart (single and multiple pie)
    f.Showing the data as a Bar Chart (Simple and multiple)
19. Develop Following Program Using HTML5 and Google Chats API and Map API

a.Using Google Charts API Basics draw charts like a Bar chart
b.Using Google Charts API Basics draw charts like a Line chart

20. Develop Following Program Using HTML5 and Google Chats API and Map API
    a.Using Google Charts API Basics draw PieChart.
    b.Using Google Charts API Basics draw Donut Chart.
    c.Using Google Charts API Basics draw Candle Chart
21. Develop Following Program Using HTML5 and Google Chats API and Map API
    a.Using Google Charts API Basics draw other types of Chart.
    b.Using Google API read JSON file and create Google Map.
22. Development of Dashboard.
23. Case Study

Project Based Learning :

1. Scatter Plot with Matplotlib.
2. Horizontal Bar Chart using Pandas.
3. Boxplot with Seaborn.
4. Histogram with Plotnine (ggplot).
5. Stacked Bar Plot.
6. Heatmaps.
7. Interactive Plot with Plotly (using Cufflinks).
8. Basic Interactive Binned Scatter Plot with Altair.
9. Correlogram.
10. Interactive Time Series Visualization.
11. Interactive Sunburst Charts.
12. Race Bar Chart.
13. Interactive Choropleth Map.

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Elective II: Intelligent Autonomous Systems & Robotics

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	04 Hours/Week	University Examination:	60 Marks	Lecture	04
Tutorial	01 Hours/Week	Internal Assessment:	40 Marks	Tutorial	01
		Total	100	Total	05

Course Objective: To provide students with a working knowledge of methods for design and analysis of robotic and intelligent autonomous systems.

Prerequisite: Artificial Intelligence.

Course Outcomes: On completion of the course, students will have the ability to:

13. Familiarise with anatomy of Robots
14. Obtain forward and inverse kinematic models of robotic manipulators
15. Classify Robot End effectors.
16. Classify different types of sensors
17. Plan trajectories in joint space & Cartesian space
18. Familiarise with different types of Robotics applications

Unit I Introduction **08 Hours**

Types of Robots, Robotic system and robot Anatomy specification of robots, Robot configurations-PPP, RPP, RRP, RRR; features of SCARA, PUMA Robots; Classification of robots based on motion control methods and drive technologies, Flexible automation

Unit II Robot kinematics and dynamics **08 Hours**

Forward and Reverse Kinematics, open kinematic vs closed kinematic chain; degrees of freedom, Kinematic equation using Homogeneous Transformation, inverse Kinematics, Robot arm Dynamics, The Denavit-Hartenberg (D-H) representation, The arm equation, direct kinematics problems (upto 3DOF), Mobile Robot Kinematics

Unit III Effectors **08 Hours**

Classification of End effectors - mechanical grippers, special tools, Magnetic grippers, Vacuum grippers, adhesive grippers, Active and passive grippers, hooks and scoops, selection and design considerations of grippers in robot. Gripper force analysis and gripper design,

Unit IV Sensors and intelligent robots **08 Hours**

AI and automated manufacturing, Sensing system, Types of sensors, Robot Vision system, Design and control of sensor integrated Robot hand, Sensors for Mobile Robots, Sensor classification, Characterizing sensor performance, Representing uncertainty, Wheel/motor sensors, Heading sensors, Accelerometers, Inertial measurement unit (IMU), Ground beacons, Active ranging, Motion/speed sensors, Vision sensors

Unit V Trajectory planning **08 Hours**

Path planning, Trajectory Planning. Joint space trajectory planning- cubic polynomial, linear trajectory with parabolic blends, trajectory planning with via

points; Cartesian space planning, Point to point vs continuous path planning. Obstacle avoidance methods- Artificial Potential field, A* algorithms

Unit VI Applications **08 Hours**

Robot Pose Maintenance and Localization: Simple Landmark Measurement, Servo Control, Recursive Filtering, Global Localization. Mapping: Sensorial Maps, Topological Maps, Geometric Maps, Exploration. Robots in Practice: Delivery Robots, Intelligent Vehicles, Mining Automation, Space Robotics, Autonomous Aircrafts, Agriculture, Forestry, Domestic Robots.

Textbooks

1. Deepak Khemani, "A First Course in Artificial Intelligence", McGraw Hill Education(India), 2013, ISBN :978-1-25-902998-1
2. Robotics Technology and Flexible Automation, Second Edition, S. R. Deb
3. Introduction to Robotics (Mechanics and control), John. J. Craig, Pearson Education Asia 2002.
4. Michael Jenkin, Gregory, " Computational Principals of Mobile Robotics", Cambridge University Press, 2010, ISBN : 978-0-52- 187157-0

Reference Books

4. Sicilliano, Khatib , "Handbook of Robotics", Springer
5. Andries P.Engelbrecht-Computational Intelligence: An Introduction, 2nd Edition-Wiley India- ISBN: 978-0-470-51250-0

Project Based Learning:

1. Robotic Arm Controlled by Touch Screen Display
2. Metal Detector Robotic Vehicle
3. **Chess Playing Robot**
4. **Autonomous Underwater Robot**
5. Sensor Guided Robotics
6. Smart Umbrella with Solar Cell
7. Artificial Intelligence-Based Chatbot for Appliance Control
8. Autonomous Quadcopter Docking System
9. Solar Panel Cleaning System Using Arduino
10. Pick and Place Robotic Vehicle

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Elective II: Deep Learning

TEACHING SCHEME		EXAMINATION SCHEME		CREDIT SCHEME	
	Hours/Week		**Marks**		**Credits**
Lecture:	4 Hours/Week	**University Examination**	**60 Marks**	**Lecture:**	**04**
Tutorial:	1 Hours/Week			**Tutorial:**	**01**
		Internal Assessment	**40 Marks**		
		Total	**100 Marks**		**05**

Course Objective: Introduce major deep learning algorithms, the problem settings, and their applications to solve real world problems.

Prerequisite: Linear Algebra, Statistics, probability, Machine learning

Course Outcomes: On completion of the course, students will have the ability to:

1. To Understand a wide variety of learning algorithms
2. To study the concepts of deep learning
3. To enable the students to know deep learning techniques to support real-time applications
4. Identify the deep learning algorithms which are more appropriate for various types of learning tasks in various domains.
5. Design and implement various deep supervised learning architectures for text & image data and design and implement various deep learning models and architectures
6. Apply various deep learning techniques to design efficient algorithms for real-world applications

Unit I: Basics of Neural Networks: Neural Networks basics – Binary Classification, Logistic Regression, Gradient Descent, Derivatives, Computation graph, Vectorization, Vectorizing logistic regression – **Shallow neural networks:** Activation functions, non-linear activation functions, Backpropagation, Data classification with a hidden layer. **08 Hours**

Unit II: Deep Neural Networks: Deep L-layer neural network, Forward and Backward propagation, Deep representations, Parameters vs Hyperparameters, Effective training in Deep Net- early stopping, Dropout, Batch Normalization, Instance Normalization, Group Normalization, Building a Deep Neural Network (Application). **08 Hours**

Unit III: Supervised Learning with Neural Networks: Practical aspects of Deep Learning: Train/Dev / Test sets, Bias/variance, Overfitting and regularization, Linear models and optimization, Vanishing/exploding gradients, Gradient checking. **08 Hours**

Unit IV: Logistic Regression, Convolution Neural Networks, RNN and Backpropagation – Convolutions and Pooling – Optimization algorithms: Mini-batch gradient descent, exponentially weighted averages, RMSprop, learning rate decay, problem of local optima, Batch norm – Parameter tuning process. Unsupervised Learning with Deep Network, Autoencoders. **08 Hours**

Unit V: Neural Network Architectures: Recurrent Neural Networks, Adversarial NN, Spectral CNN, Self-Organizing Maps, Restricted Boltzmann Machines, Recent Trends in Deep Learning Architectures, Residual Network, Skip Connection Network, Fully Connected CNN etc. **08 Hours**

Unit VI Long Short-Term Memory Networks (LSTM) and Deep Reinforcement Learning: TensorFlow, Keras or MatConvNet for implementation. Generative Modeling with DL, Variational Autoencoder, Generative Adversarial Network Revisiting Gradient Descent, Momentum Optimizer, RMSProp, Adam **08 Hours**

Textbooks

1. Deep Learning (Adaptive Computation and Machine Learning series) Hardcover, by Aaron Courville, Ian Goodfellow, Yoshua Bengio
2. Deep Learning for Natural Language Processing: Applications of Deep Neural Networks to Machine Learning Tasks by Pearson Learn IT
3. Advanced Deep Learning with Keras by Rowel Atienza
4. Deep Learning with Python Paperback – 22 December 2017
5. Advanced Deep Learning with Keras by Rowel Atienza

Reference Books

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron
2. Machine Intelligence: Demystifying Machine Learning, Neural Networks and Deep Learning by Suresh Samudrala

Project Based Learning:

1. KNN (K - nearest neighbor) method
2. Artificial Neural Network (ANN)
3. Convolutional Neural Network (CNN)
4. Recurrent Neural Network (RNN)
5. Deep Neural Network (DNN)
6. Deep Belief Network (DBN)
7. Back Propagation
8. Stochastic Gradient Descent

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Elective II: Blockchain & Cryptocurrency

Teaching Scheme		Examination Scheme		Credit Scheme	
Hours/Week			Marks		Credits
Lecture:	04 Hours/Week	University Examination:	60 Marks	Lecture	04
Tutorial:	01 Hours/Week	Internal Assessment:	40 Marks	Tutorial	01
		Total	100 Marks	Total	05

Course Objective:
To get acquainted with the concept of Block and Blockchain.
To analyze the applications& case studies of Blockchain.

Prerequisite:
Cyber Security, Network security, Distributed networks, Object Oriented programming language.

Course Outcomes: On completion of the course, students will have the ability to:
7. Describe the basic concept of Block chain.
8. Associate knowledge of consensus and mining in Block chain.
9. Summarize the bit coin crypto currency at an abstract level.
10. Apply the concepts of keys, wallets and transactions in the Bit coin network.
11. Interpret the knowledge of Bit coin network, nodes and their roles.
12. Illustrate the applications of Block chain and analyze case studies.

Unit I Introduction to Block chain **08 Hours**
Structure of a Block, Block Header, Block Identifiers: Block Header Hash and Block Height, The Genesis Block, Linking Blocks in the Block chain, Merkle Trees and Simplified Payment Verification (SPV).

Unit II Consensus and Mining **08 Hours**
Decentralized Consensus, Byzantine General's Problem, Independent Verification of Transactions, Mining Nodes, Aggregating Transactions into Blocks, Constructing the Block header, Mining the Block, Successfully Mining the Block, Validating a New Block, Assembling and Selecting Chains of Blocks, Block chain Forks

Unit III Introduction to Bit coin **08 Hours**
What is Bit coin and the history of Bit coin, Getting the first bit coin, finding the current price of bit coin and sending and receiving bit coin, Bit coin Transactions.

Unit IV Concepts of Bit coin **08 Hours**
Keys and addresses, Wallets and Transactions: Public Key Cryptography and Crypto currency, Private and Public Keys, Bit coin Addresses, Base58 and Base58Check Encoding, Nondeterministic (Random) Wallets, Deterministic (Seeded) Wallets, HD Wallets (BIP-32/BIP-44), Wallet Best Practices, Using a Bit coin Wallets, Transaction Outputs and Inputs, Transaction Fees, Transaction Scripts and Script Language, Turing Incompleteness, Stateless Verification,

Script Construction (Lock + Unlock), Pay-to-Public-Key-Hash (P2PKH), Bitcoin Addresses, Balances, and Other Abstractions

Unit V Bit coin Networks **08 Hours**

Peer-to-Peer Network Architecture, Node Types and Roles, Incentive based Engineering The Extended Bitcoin Network, Bitcoin Relay Networks, Network Discovery, Full Nodes, Exchanging "Inventory", Simplified Payment Verification (SPV) Nodes, Bloom Filters, SPV Nodes and Privacy, Encrypted and Authenticated Connections, Transaction Pools

Unit VI Blockchain Applications & case studies **08 Hours**

Domain-Specific Applications: FinTech, Internet of Things, Industrial and Manufacturing, Energy, Supply chain & Logistics, Records & Identities, Healthcare Case studies related to cryptocurrencies Concept of Altcoin.

Textbooks

6. Mastering Bitcoin, PROGRAMMING THE OPEN BLOCKCHAIN, 2nd Edition by Andreas M. Antonopoulos, June 2017, Publisher(s): O'Reilly Media, Inc. ISBN:9781491954386.
7. Blockchain Applications: A Hands-On Approach", by Arshdeep Bahga, Vijay Madisetti, Paperback – 31 January 2017

Reference Books

4. "Mastering Blockchain", by Imran Bashir, Third Edition, Packt Publishing
5. "Mastering Ethereum: Building Smart Contracts and Dapps Paperback" by Andreas Antonopoulos, Gavin Wood, Publisher(s): O'Reilly Media

Project Based Learning - Provisional List of Projects

1. Smart Contract: Development of smart block-based contact for project development
2. Crypto-wallet: Creating a Crypto wallet for handling cryptocurrency
3. Cryptocurrency: ERC-20 tokens & creating own cryptocurrency using solidity for Ethereum.
4. Blockchain-based Lottery – Picking a Winner from various Blockchain Nodes taking part in a lottery.
5. Install and Use Ganache, Flask and Postman.
6. Remix-Ethereum IDE
7. Simple Smart Contract for Bank with withdraw and deposit functionality.
8. Smart Contract for storing and retrieving information of Degree Certificates.
9. Simple Python program to create a Block class that contains index, timestamp, and previous hash. Connect the blocks to create a Blockchain

Syllabus for Unit Tests

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Elective II: Docker and Kubernetes

TEACHING SCHEME		EXAMINATION SCHEME		CREDIT SCHEME	
	Hours/Week		**Marks**		**Credits**
Lecture:	4 Hours/Week	**University Examination**	**60 Marks**	**Theory**	04
Tutorial:	1 Hours/Week			**Tutorial**	01
		Internal Assessment	**40 Marks**		
		Total	**150 Marks**	**Total**	05

Course Objective:
To make student aware of basics of Docker and Kubernetes Knowledge representation methods, learning concept and basics of it.

Prerequisite:
DevOps, Linux, AWS, and Docker concepts, CI + CD pipeline

Course Outcomes: On completion of the course, students will have the ability to:

1. Describe the challenges in developing Docker Syatem
2. Apply appropriate problem-solving strategy to solve a particular problem
3. Use appropriate knowledge representation method
4. Describe components of planning system
5. Apply the various knowledge representation strategies
6. Describe the plan generation systems

Unit I: Introduction to Docker **08 Hours**

What is Docker , manipulating container with docker client , Docker run in detail, Container lifecycle and Log output, Purpose of IT flag,
Starting with shell, Docker Architecture

Unit II: Building Custom Images through Docker Server **08 Hours**

Creating Docker image, Base Image , Build Process, Tagging in image .Node Server Setup , Base image issues , Copying build files, Container port mapping, Specifying a Working Directory , Unnecessary Rebuilds, Minimizing Cache Busting and Rebuilds.

Unit III Docker Architecture and CICD with AWS **08 Hours**

The Docker daemon, The Docker client, Docker registries

Docker objects, Development work flow, Docker volume , Shorthand with Docker Compose, Multi-Step Docker Builds, Github Setup, Travis CI Setup, Travis YML File Configuration, AWS Elastic Beanstalk, Travis Config for Deployment, Automated Deployments, Exposing Ports Through the Dockerfile, Workflow With Github, Redeploy on Pull Request Merge, Single Container Deployment Issues, Checkpoint Catchup, Adding Postgres as a Service, Docker-compose Config, Production Dockerfiles, Creating the EB Environment, Managed Data Service Providers, Verifying Deployment

Unit IV Kubernetes **08 Hours**

Overview , Features , Kubernetes - Cluster Architecture, Kubernetes - Master Machine Components, etcd, API Server, Controller Manager, Scheduler, Kubernetes - Node Components, Docker, Kubelet Service, Kubernetes Proxy Service, Kubernetes - Master and Node Structure

Unit V: Kubernetes services **08 Hours**

Service without Selector, Service Config File with Selector , Multi-Port Service Creation, Types of Services ClusterIP , NodePort, Load Balancer , Types of Pod , Kubernetes - Replication Controller, Kubernetes - Replica Sets, Kubernetes – Volumes , Kubernetes – Secrets , Kubernetes - Network Policy

08 Hours

Unit VI : Advanced Kubernet

API, Kubectl , Kubectl Commands, Creating an App, Autoscaling, Dashboard Setup, Monitoring

Textbooks

1. The Docker Book James Turnbull in 2014
2. Docker up and running Karl Matthias and Sean P

Reference Books

Dr. Gabriel N. Schenker, “Containerize your Apps with Docker and Kubernetes”.

Project Based Learning - Provisional List of Projects

1. Creating a docker image
2. Running the first container
3. Retrieving container logs
4. Creating CI/CD pipeline
5. Writing script for build automation
6. With the help of App infrastructure isolation (Deploying multiple apps on docker)
7. Use Multi-tenancy support(working with different computing environments)
8. Creation of microservices architecture
9. Installing docker on VM
10. Build and deploy the multistage image

Syllabus for Unit Tests:

Unit Test -1	Unit – I, Unit – II, Unit - III
Unit Test -2	Unit – IV, Unit – V, Unit - VI

Programming Technologies and Tools Laboratory – VII

Teaching Scheme		Examination Scheme		Credit Scheme	
	Hours/Week		Marks		Credits
Lecture:	01 Hours/Week	Term Work	25 Marks		
Practical:	02 Hours/Week	Practical	25 Marks	Lecture	01
			50 Marks	Practical	01
		Total	50 Marks	Total	02

Course Objective:

3. Understand role of blockchain in Web 3.0.
4. Understand the bitcoin blockchain platform and its terminologies.
5. Understand Ethereum architecture and enterprise blockchain.
6. Study and developed smart contracts, DAPPS for different application.

Prerequisite:

1. Cyber Security, Network security, Distributed networks
2. Object Oriented programming language.

Course Outcomes:

1. Differentiate between Web 2.0 and Web 3.0 with respect to various applications.
2. Elaborate the bitcoin mining, DLT, Consensus algorithm.
3. Analyse the Ethereum architecture.
4. Analyse the Hyperledger Fabric architecture.
5. Design smart contract and DAPP for real time application.
6. Illustrate blockchain integration with emerging technologies and security issues.

Unit I: Fundamentals of Blockchain **02 Hours**

Challenges Faced by Modern Businesses, Features of Blockchain, Building Blocks of Blockchain, Introduction to Blockchain Pillars, Why Blockchain Platform: Platform types, Public, Private, technology requirements for implementation. Distributed Ledger, Introduction to cryptography-Encryption and Decryption-Ciphers-Cryptography using arithmetic modulo primes-hashing algorithms-SHA-256 algorithm-Application of SHA algorithm, Web 2.0 and Web 3.0.

Unit II: Bitcoin Blockchain **02 Hours**

Introduction to Bitcoin, Bitcoin Wallets, Bitcoin Block, Bitcoin Transaction, Bitcoin Network, Operation of Bitcoin Blockchain, Blockchain Architecture – Block, Hash, Distributer P2P, Structure of Blockchain- Consensus mechanism: Proof of Work, Bitcoin (BTC) – Genesis Block, Buy Bitcoin, Transactions, Unspent Transaction Output (UTXO), Bitcoin Mining, Value of Bitcoin, Advantages and Disadvantages

Unit III: Ethereum Blockchain **02 Hours**

Introduction, Ethereum components: miner and mining node, Ethereum virtual machine, Ether, Gas, Transactions, accounts, swarm and whisper, Ethash, end to end transaction in Ethereum, architecture of Ethereum Dapp Architecture, DAO.

Unit IV: Ethereum Smart Contracts **02 Hours**

Smart Contract, Smart Contract Lifecycle, Solidity, Solidity State and Variable Types, Solidity Functions, Solidity Compilation and
Deployment, mapper function, ERC20 and ERC721 Tokens, comparison between ERC20 & ERC721, ICO, use cases of smart, contract, smart Contracts: Opportunities, Risks.

Unit V Enterprise Blockchain **02 Hours**

Introduction to Hyperledger, tools and frameworks, Hyperledger Fabric, Comparison between Hyperledger Fabric & Other Technologies, Distributed Ledgers. Hyperledger Fabric Architecture, Components
of Hyperledger Fabric: MSP, Chain Codes etc., Transaction Flow, Advantages of Hyperledger Fabric Blockchain, working of Hyperledger Fabric, Creating Hyperledger network, Case Study of Supply chain.
management using Hyperledger. Hyperledger Fabric (A): Decomposing the consensus process, Hyperledger fabric components, Chain code Design and Implementation Hyperledger Fabric (B): Beyond Chain code: fabric SDK and Front End (b) Hyperledger composer tool.

Unit VI: Blockchain integration and Research challenges **02 Hours**

Integrating Blockchain with cloud, IoT, AI, ERP, End to end blockchain integration, Risks and Limitations of Blockchain: Privacy & Security. Criminal Use of Payment Blockchains, The “Dark” Side of Blockchain.

Textbooks

9. “Mastering Bitcoin, PROGRAMMING THE OPEN BLOCKCHAIN” , 2nd Edition by Andreas M. Antonopoulos

10. William Stallings, “Cryptography and Network security Principles and Practices”, Pearson/PHI,2017

11. Mastering Ethereum, Building Smart Contract and Dapps, Andreas M. Antonopoulos Dr. Gavin Wood, O'reilly.

Reference Books

1. Mastering Blockchain”, by Imran Bashir, Third Edition,Packt Publishing
2. Blockchain with Hyperledger Fabric,LucDesrosiers, Nitin Gaur, Salman A. Baset, Venkatraman Ramakrishna,Packt Publishing
3. Atul Kahate, Cryptography and Network Security, Tata Mc Grawhill, India, 2019.

List of Experiments:

1. Working of Blockchain Transaction, DLT(https://andersbrownworth.com/blockchain/)
2. Implement program to convert given text in to hashes using SHA 256 algorithm.
3. Create simple wallet transaction from one account to another account using Metamask.
4. Connect Metamask to a Ganache Test Network
5. Ether Transaction Using Ganache.
6. Write Hello World smart contract in a higher programming language (Solidity).

7. Write simple smart contract for User identity management using Solidity language.
8. Write simple smart contract for Crowd fund ERC20 token

    1. User creates a campaign.
    2. Users can pledge, transferring their token to a campaign.
    3. After the campaign ends, campaign creator can claim the funds if total amount pledged is more than the campaign goal.
    4. Otherwise, campaign did not reach its goal, users can withdraw their pledge.

9. Build NFT Application for Fan engagement and gaming rewards
10. Write smart contract for Tracking property details in real estate
    Create DAPP for Protecting sensitive medical data in healthcare
11. Build NFT Application by writing smart contract
    English Auction
    English auction for NFT.
    Auction
    - Seller of NFT deploys this contract.
    - Auction lasts for 7 days.
    - Participants can bid by depositing ETH greater than the current highest bidder.
    - All bidders can withdraw their bid if it is not the current highest bid.

    After the auction
    - Highest bidder becomes the new owner of NFT.
    - The seller receives the highest bid of ETH.
12. Creating a Business Network using Hyperledger Fabric

Syllabus for Unit Tests: NA

Project Stage -II

TEACHING SCHEME	EXAMINATION SCHEME	CREDIT SCHEME	
			Credits
Practical: 4 Hrs/Week		Practical	06
	Term Work : 100 Marks	Term Work	
	Oral : 100 Marks	Oral	
		Term Work	
	Total 200 marks	Total	06

Course Pre-requisites:
Basics of Software engineering, Software testing and knowledge of core computer engineering subjects.

Course Objectives:

- To provide in depth outline for software project planning and development.

Course Outcomes: On completion of the course, students will have the ability to:

- Use appropriate software development tool for the proposed problem.
- Deign test cases and perform testing.
- Perform collaboratively towards a common purpose.
- Demonstrate self-advocacy skills and self-reliant behaviour.
- Demonstrate the ability to develop and maintain satisfying interpersonal. relationships.
- Evaluate and conclude the results with documentation.

1 The project will be undertaken preferably by a group of at least 3- 4 students who will jointly work and implement the project over the academic year. The work will involve the design of a system or subsystem in the area of Computer Engineering.

2. If the project is chosen a hardware project it will involve the designing a system – subsystem or upgrading an existing system. The design must be implemented into a working model with necessary software interfacing and a user manual.

3. If the project is chosen in the pure Software Application it must involve the detail Software Design Specifications, Data Structure Layout, File Design, Testing with complete documentation and user interface. With life cycle testing and as an executable package.

The group will submit at the end of Semester-VIII,

i) The workable project.

ii) The details of Research paper published in National/International paper conferences/journals for the project work carried out.

iii) Project Report complete in all aspects, 3 copies for the institute and 1 copy of each student in the group for certification.

The examiner in consultation with the guide will assess the term work.

Oral examination will be based on the project work completed by the candidate.

Project report must be checked for plagiarism from respective guide.

The project report will contain the following details:

1. Problem definition and requirement specification, acceptance tests procedure (ATP).
2. System definition, requirement analysis.
3. System design.
4. System implementation-code documentation –dataflow diagram / algorithm.
5. Test results and procedure, test report as per ATP.
6. Platform choice, use.
7. Appendix tools used, references.
8. Documentation will use UML approach with Presentation, Category, Use Case, Class Diagrams, etc.