ACADEMIC REGULATIONS,

COURSE STRUCTURE

&

SYLLABUS

R22 REGULATIONS

CHOICE BASED CREDIT SYSTEM (CBCS)

III & IV B.Tech.

Computer Science and Engineering

Applicable to batches admitted in the First year
from 2022-23 onwards

CVR COLLEGE OF ENGINEERING

An UGC Autonomous Institution with NAAC Grade 'A'

(Approved by AICTE & Govt. of Telangana and
Affiliated to JNT University, Hyderabad)
Vastunagar, Mangalpalli (V), Ibrahimpatan (M),
Ranga Reddy Dist., Pin – 501 510

CVR COLLEGE OF ENGINEERING

VISION

- To be a state of the art institution of engineering in pursuit of excellence, in the service of society.

MISSION

- To excel in providing quality education at undergraduate and graduate levels.
- To encourage research and innovation.
- To provide infrastructure and facilities to meet the latest technological needs.
- To establish Centres of Excellence through active interaction with industry.
- To nurture students towards holistic development with human values and ethics.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

- Towards a Global Knowledge Hub, striving continuously in pursuit of excellence in Education, Research, Consultancy, and Technological services to society.

MISSION

1. To produce the best quality Computer Science & Engineering professionals by imparting quality training, hands-on experience and value education.
2. To strengthen links with industry through partnerships and collaborative developmental works.
3. To attain self-sustainability and overall development through Research, Consultancy, and Development activities.
4. To extend technical expertise to other technical institutions of the region and play a lead role in imparting technical education.
5. To inculcate work ethics and commitment in students for their future endeavors to serve society.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs):

PEO 1 : Employability: Computer Science & Engineering graduates will acquire the capability to apply their knowledge and skills to solve various kinds of computational engineering problems.

PEO 2 : Professionalism: Graduates will inculcate a professional attitude, interdisciplinary approach, ethics, and ability to relate computer engineering issues with social awareness.

PEO 3 : Managerial skills: Graduates will possess managerial skills to face challenges in the profession by working harmoniously in a team with effective communication skills.

PEO 4 : Continuous learning: Graduates will continue to learn and adapt in a world of constantly evolving technologies and pursue research towards academic excellence.

PEO 5 : Adaptability: Graduates of Computer Science & Engineering will have soft skills to adapt to the diverse global environment.

PROGRAM OUTCOMES (POs):

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identity, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs):

PSO1: Successfully design and implement algorithmic paradigms by using efficient programming language constructs, formal tools, and frameworks.

PSO2: Develop scalable and reliable distributed applications and data analytics pipelines by employing industry-agnostic technologies and secure software engineering models.

PSO3: Adapt cloud computing ecosystems and machine learning algorithms to develop smart and sustainable solutions complying the ethics of society and eventually emerge as entrepreneurs.

CVR COLLEGE OF ENGINEERING

III B.Tech. Computer Science and Engineering

I Semester Course Structure

Regulations: R22-CBCS **With effect from the Academic Year 2024-25 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CS301	Computer Networking	PC	3	0	3	40	60	100	5
2	22CS302	Algorithms Design and Analysis	PC	3	0	3	40	60	100	7
3	22IT301	Web Technologies	PC	3	0	3	40	60	100	9
4	22CS303	Data Science	PC	3	0	3	40	60	100	11
	Professional Elective-I									
5	22CS304	Computer Graphics	PE	3	0	3	40	60	100	13
	22CS305	Distributed Databases								15
	22CS306	Microprocessors and Interfacing								17
	22CS307	Principles of Programming Languages								19
	22CS308	Digital Image Processing and Computer Vision								21
Practicals										
6	22CS331	Algorithms and Data Science Lab	PC	0	2	1	40	60	100	23
7	22IT331	Web Technologies Lab	PC	0	3	1.5	40	60	100	26
8	22CS332	Case Tools Lab	PC	0	2	1	40	60	100	28
9	22IT332	Mobile Application Development Lab	PC	0	3	1.5	40	60	100	30
		Total		**15**	**10**	**20**	**360**	**540**	**900**	
		Total Hours		**25**						

Service Courses to III B.Tech. IT I Semester

Regulations: R22-CBCS **With effect from the Academic Year 2024-25 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CS302	Algorithms Design and Analysis	PC	3	0	3	40	60	100	7
2	22CS304	Computer Graphics	PE	3	0	3	40	60	100	13
3	22CS305	Distributed Databases	PE	3	0	3	40	60	100	15
4	22CS308	Digital Image Processing and Computer Vision	PE	3	0	3	40	60	100	21

Note: Lecture Hours (L), Tutorials (T), Practicals (P), Drawing (D) & Credits (C)

HS: Humanities & Sciences **PE: Professional Electives**
PC: Professional Core **MC: Mandatory Course**

CVR COLLEGE OF ENGINEERING

III B.Tech. Computer Science and Engineering

II Semester Course Structure

Regulations: R22-CBCS **With effect from the Academic Year 2024-25 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CS351	Artificial Intelligence and Machine Learning	PC	3	0	3	40	60	100	33
2	22CS352	Full Stack Development	PC	3	0	3	40	60	100	35
3	22CS353	Cloud Computing and Devops	PC	3	0	3	40	60	100	37
4	22CS354/ 22CS309	Automata Theory and Compiler Design	PC	3	0	3	40	60	100	39
	Professional Elective-II									
5	22CS355	Crytography and Essentials of Network Security	PE	3	0	3	40	60	100	41
	22CS356	Adhoc Sensor Networks								43
	22CS357	Internet of Things								45
	22CS358	Distributed Systems								47
	22CS359	Aritificial Neural Networks and Graphical Models								49
Practicals										
6	22CS381	IoT and Cloud Computing Lab	PC	0	2	1	40	60	100	51
7	22CS382	Full Stack Development Lab	PC	0	2	1	40	60	100	53
8	22HS381/ 22HS331	Advanced English Communication and Soft Skills Lab	HS	0	2	1	40	60	100	57
9	22CS383	Industrial Oriented Mini Project/ Internship	PC	0	4	2	40	60	100	
		Total		**15**	**10**	**20**	**360**	**540**	**900**	
		Total Hours		**25**						
10	22HS352/ 22HS302/ 22HS253	Intellectual Property Rights	MC	3	0	0	100	0	100	59

Service Courses to III B.Tech. IT II Semester

Regulations: R22-CBCS **With effect from the Academic Year 2024-25 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CS351	Artificial Intelligence and Machine Learning	PC	3	0	3	40	60	100	33

Service Courses to III B.Tech. CSE(AIML), CSE(DS) and CSE(CS) I Semester

Regulations: R22-CBCS **With effect from the Academic Year 2024-25 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CS354/ 22CS309	Automata Theory and Compiler Design	PC	3	0	3	40	60	100	39

Note: Lecture Hours (L), Tutorials (T), Practicals (P), Drawing (D) & Credits (C)

HS: Humanities & Sciences **PE: Professional Electives**
PC: Professional Core **MC: Mandatory Course**

CVR COLLEGE OF ENGINEERING

IV B.Tech. Computer Science and Engineering

I Semester Course Structure

Regulations: R22-CBCS **With effect from the Academic Year 2025-26 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CS401	Linux Programming	PC	3	0	3	40	60	100	61
2	22HS401/ 22HS301/ 22HS351	Business Economics & Financial Analysis	HS	3	0	3	40	60	100	63
3	**Professional Elective-III**									
	22IT403	Quantum Computing	PE	3	0	3	40	60	100	65
	22IT404	Cloud Security								67
	22CY405	Blockchain Technologies								69
	22DT406	NoSQL Databases								71
	22AM408/ 22AM355	Augmented Reality and Virtual Reality								73
4	**Professional Elective-IV**									
	22CS402	Storage Management Systems	PE	3	0	3	40	60	100	75
	22CS403	Enterprise Applications and Architectural Patterns								77
	22IT405	Internet of Everything								79
	22CS404	Data Warehousing and Data Mining								81
	22CS405	Deep Learning and Applications								83
5		**Open Elective-I**	OE	3	0	3	40	60	100	
Practicals										
6	22CS431	Big Data Analytics and Platforms Lab	PC	0	2	1	40	60	100	85
7	22CS432	Visual Programming Lab	PC	0	2	1	40	60	100	89
8	22CS433	Project Work Stage-I	PC	0	6	3	40	60	100	
Total				**15**	**10**	**20**	**320**	**480**	**800**	
Total Hours				**25**						

Note: Lecture Hours (L), Tutorials (T), Practicals (P), Drawing (D) & Credits (C)

PE: Professional Electives **PC: Professional Core** **OE: Open Elenctive**
HS: Humanities & Sciences

CVR COLLEGE OF ENGINEERING

IV B.Tech. Computer Science and Engineering

II Semester Course Structure

Regulations: R22-CBCS **With effect from the Academic Year 2025-26 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22HS451/ 22HS404	Organizational Behaviour	HS	3	0	3	40	60	100	93
2	**Professional Elective-V**									
	22CS451	Natural Language Processing Models	PE	3	0	3	40	60	100	95
	22CS452	Penetration Testing and Incident Response								97
	22CS453	Robotic Process Automation								99
	22IT451	Design Patterns								101
	22IT452	Generative AI								103
3		**Open Elective-II**	OE	3	0	3	40	60	100	
Practicals										
4	22CS481	Project Work Stage–II	PC	0	22	11	40	60	100	
			Total	**9**	**22**	**20**	**160**	**240**	**400**	
			Total Hours	**31**						

Service Courses to IV B.Tech. IT, CSE(AIML) I Semester

Regulations: R22-CBCS **With effect from the Academic Year 2024-25 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CS453	Robotic Process Automation	PE	3	0	3	40	60	100	99

Note: Lecture Hours (L), Tutorials (T), Practicals (P), Drawing (D) & Credits (C)

PE: Professional Electives **PC: Professional Core** **OE: Open Elenctive**
HS: Humanities & Sciences

Course code: 22CS301

COMPUTER NETWORKING

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To introduce communication reference models and to understand the characteristics of the transmission media.
2 : To solve the optimal route establishment problems for data delivery using relevant metrics.
3 : To serve data at the end point level and to ensure reliable data delivery mechanisms.
4 : To model secured exchange of high-level data between two applications.

Unit I – Introduction

Uses of Computer Networks, Network Hardware, Network Software, Reference Models, Example Networks. **Physical Layer** - Guided Transmission Media, Wireless Transmission.

Unit II - Data Link Layer

Data Link Layer Design Issues, Error Detection and Correction, Elementary Data Link Protocols, Sliding Window Protocols, example of data link protocols.

Medium Access Control Sub-Layer -The Channel Allocation Problem, Multiple Access Protocols, Ethernet, Wireless LANs.

Unit III - Network Layer

Network Layer Design Issues, Routing Algorithms, Congestion Control Algorithms, Quality of Service. Internetworking, Network Layer in the Internet.

Unit IV - Transport Layer and Application Layer

Transport Service, Elements of Transport Protocols, Internet Transport Protocols- User Datagram Protocol, Transmission Control Protocol.

Application Layer -Domain Name System, Electronic Mail, World Wide Web. Simple Network Management Protocol.

Unit V – Information Security

Cryptography, security services, message confidentiality, message integrity, message authentication, digital signature.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Appreciate communication reference models along with PDUs and design a physical network based on the topological specifications using various communication media.
CO 2 : Design solutions for logical link layer transmission and control errors, control the medium access patterns through an established methodology.
CO 3 : Solve the optimal routing problems for static and dynamic networks and realizing various QoS parameters.
CO 4 : Demonstrate the significance of various applications using TCP and UDP protocol.
CO 5 : Implement CIA security mechanisms, and Network Security protocols.

Textbooks:

1. Computer Networks, Andrew S Tanenbaum, 4th Edition, Pearson Education, 2011.
2. Data Communications and Networking, Behrouz A. Forouzan, 5th Edition, TMH, 2009.

References:

1. Computer Networking: A Top-Down Approach, James F. Kurose, and K.W. Ross, 7th Edition, Pearson Education, 2017.
2. Introduction to Data Communications and Networking, W. Tomasi, Pearson Education, 2009.
3. Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson Education, 2008.

Course Code: 22CS302

ALGORITHMS DESIGN AND ANALYSIS

(Common to CSE, IT)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To emphasize upon the demands of real-world problems in engineering solutions
2 : To make students conversant with the various paradigms of algorithms
3 : To handcraft the performance analysis of designed solutions
4 : To take students through various optimization principles of ill-posed problems

Unit I – Fundamentals of algorithm analysis

Introduction- Definition of algorithm, algorithmic problem solving, pseudo code for expressing algorithms. Asymptotic notations- O, Ω, and θ notations. Performance analysis: Time and space complexity: count, tabular methods, examples on non-recursive, recursive algorithms. Recursive algorithms and recurrence relations - ToH problem, Amortized analysis.

Unit II - **Divide and Conquer**

Control abstraction, binary search algorithm and its complexity, Merge sort, its complexity, quick sort, its complexity. Graph traversals: Depth first search (dfs), breadth first search (bfs), articulation points, bi-connected components. Disjoint Sets: Union and Find.

Unit III - **Greedy paradigm**

Control abstraction, fractional knapsack problem, job sequencing problem, minimum cost spanning tree: Prim's algorithm, Kruskal's algorithm, Single source shortest path algorithm, Huffman coding.
Pattern matching algorithms: Knuth-Morris Pratt algorithm and brute force algorithm.

Unit IV - Dynamic programming and Backtracking

Dynamic programming **-** Control abstraction, Multistage Graphs, OBST, Travelling salesperson problem, reliability design, 0/1 knapsack problem.
Backtracking - n-queens problem, Graph coloring, Sum of subsets problem.

Unit V - **Branch and Bound, Complexity Theory**

Branch and Bound: General method, Applications: Travelling sales person problem, 0/1 knapsack problem, LC branch and bound solution, FIFO branch and bound solution, Game trees, Heuristics for search space reduction: alpha-beta pruning.
Np-hard and NP-complete problems: basic concepts, non-deterministic algorithms, NP-hard and NP-complete classes, Cook's theorem.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Analyze worst-case running times using asymptotic analysis of algorithms.
CO 2 : Describe the divide-and-conquer paradigm and Synthesize divide-and-conquer algorithms.
CO 3 : Define optimization problems and solve them through various greedy policies
CO 4 : Describe the dynamic-programming paradigm and synthesize dynamic-programming algorithms and analyze them.
CO 5 : Reduce the size of search space of the optimization problems by applying backtracking and branch and bound tools. Appreciate the Non-Deterministic modeling of algorithms.

Textbooks:

1. Fundamentals of algorithms, E. Horowitz and S.Sahni, 2nd edition, Galgotia Publications, 2010
2. Introduction to algorithms, T.H.Cormen, C.E.Leiserson, R.L.Rivest, and C.Stein, 2nd edition, PHI/Pearson Education, 2001.

References:

1. Introduction to Design and Analysis of Algorithms, A strategic approach, R C T Lee, Hang and TT Sai, TMH, 2012
2. Data structures and Algorithm Analysis in C++, Allen Weiss, 2nd edition, Pearson Education, 2002.
3. Design and Analysis of Algorithms, Aho, Ullman and Hopcroft, Pearson Education, 1974.
4. Algorithms, Richard Johnson Baugh, and Marcus Schaefer, Pearson Education, 2004.

Course Code: 22IT301

WEB TECHNOLOGIES

(Common to IT, CSE, CSE(AI&ML), CSE(CS) & CSE(DS))

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Prerequisites:

1. Must have knowledge in HTML 5 and CSS 3.
2. Must be knowledgeable on Java Technology.
3. Must be knowledgeable on any RDBMS.

Course Objectives:

1 : To learn a framework to create responsive web designing.
2 : To learn the client-side script and validations along with asynchronous programming.
3 : To introduce XML and work with data storage and interactivity using Java.
4 : To introduce Server-side programming with Java Servlets.
5 : To learn sending Dynamic Response from server using JSP.

Unit I – Working with CSS and its Framework

Introduction to CSS: Syntax structure, using style sheets, Box model. **CSS3:** Grid, Flexbox. Responsive Web Design using Media Queries, use of viewport, Transition, Animation.

CSS Framework: Bootstrap.

CSS Framework: Bootstrap (local and CDN usage, containers, 12 – column grid system, commonly used controls – Typography, Nav, Navbar, Carousel, Button, Card, Modal dialog, Table, forms, Breadcrumbs).

Unit II – Client- Side Scripting using JavaScript

JavaScript: Introduction to JavaScript, Data types, var, let, const., Control statements, Operators, Functions, fat arrows, Arrays, Objects, Destructuring, Strings, Date Objects, Events, DOM Manipulations, Regular Expressions.

Introduction to jQuery: Syntax, Selectors, Events, Effects.

Unit III – Data storage and manipulation

XML: Syntax, namespaces, DTD, Schema, XML Document Parsing.

JDBC: Design of JDBC, JDBC Configuration, Working with JDBC Statements, Scrollable and UpdatableResultSets, Rowset, MetaData, Transactions.

Unit IV – Server-side Scripting using Servlets

Web servers: An introduction to Web Servers, Web application structure and deployment in Tomcat. MVC Architecture, Servlet Technology: Servlets, Servlet lifecycle, The Servlet API packages and classand interface hierarchy, basic servlet program template, Handling requests and responses using form parameters, using ServletContext and ServletConfig objects, using initialization parameters (both context and config level), Session management (Cookies, Http Session, URL Rewriting, HiddenForm fields).

Unit V – Dynamic Response using JSP

Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Using Cookies and session for session tracking, connecting to database in JSP.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Build a custom website with HTML, CSS, and Bootstrap
CO 2 : Demonstrate JavaScript and its asynchronous nature of execution
CO 3 : Implement the Database Connectivity and Component Technologies like Beans
CO 4 : Develop and deploy Servlet based web applications
CO 5 : Develop Server-side programming using JSP

Textbooks:

1. Teach Yourself HTML, CSS, and JavaScript All in One, Julie C. Meloni, Jennifer Kyrnin, Sams 3rd Edition, Pearson Publication, 2019.
2. Head First Servlets and JSP, Bryan Basham, Kathy Sierra and Bert Bates, O'Reilly Media, 2nd Edition, 2008.
3. Core Java Volume II—Advanced Features, Cay S. Horstmann, 10th Edition, Prentice HallPublications, 2017.

References:

1. Responsive Web Design with HTML5 and CSS3, Ben Frain, Second Edition, Packt Publishing, 2015.
2. Beginning HTML, XHTML, CSS, and JavaScript, Jon Duckett, Wiley Publishing, Inc., 2010.
3. Core Servlets and JSPs, Martin Hall and Larry Brown, Volume I and II, Pearson.
4. E – Resource: https://www.w3schools.com/html/
5. E – Resource: https://developer.mozilla.org/en-US/docs/Learn/JavaScript
6. E – Resource: https://getbootstrap.com/

Course code: 22CS303

DATA SCIENCE

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To understand data science and descriptive analysis of data.
2 : To develop models to perform association, regression, and classification on data.
3 : To develop a comprehensive understanding of various Data analysis methods.

Unit I – Introduction to Data Science

Data science, Characteristics of Bigdata, Different steps in Data science process, Types of Data analytics.

Descriptive Analysis: Data Types and Scales, Types of Data Measurement Scales, Measures of Central Tendency, Measures of Variation, Similarity, and dissimilarity measures,

Data preprocessing: Data Cleaning, Data Integration, Data Transformation.

Unit II – Exploratory Data Analysis
Hypothesis testing: t-Test, z-Test, Chi-Square-Test.

Analysis of Variance (ANOVA): One-way, Two-way.

Multivariate Analysis: Mean Vector, Covariance, Correlation and Precision Matrices, Multivariate Data, Parameter Estimation, Estimation of Missing Values, Multivariate Normal Distribution.

Dimensionality Reduction: Principal Component Analysis and Multi-Dimensional Scaling.

Unit III – Predictive Analysis-I

Simple linear Regression: Model Building, Estimation of Parameters, Interpret coefficients, Validation of model, Outlier analysis. Bias, variance, and trade-off, Gradient descent, over and under fitting models.

Multiple linear regression: Ordinary Least Squares Estimation, Model building, Validation of model, coefficients of Multiple determination R^2 and adjusted R^2, Multicollinearity and Variance Inflation factor, Ridge, Lasso regression.

Unit IV- Predictive Analysis-II

Classification: Basic concepts, K – Nearest Neighbor classifier, Support Vector Machines, Nonlinear boundaries, Kernel functions, Bayes classifier/Graph analytics, Model Evaluation and Selection, Techniques to Improve Classification Accuracy.

Logistic Regression: Binary Logistic Regression, Estimation of Parameters, Interpretation of Parameters, Model Diagnostics, Classification Table, Sensitivity, and Specificity.

Unit V – Prescriptive Analysis

Forecasting Techniques: Time series data, Techniques, and accuracy, Moving average method, Single, double, triple exponential smoothing, Regression model for forecasting, Auto-Regression models, ARIMA Process.

Graph Analytics: Path analysis, Connectivity analysis, Community analysis, Centrality analysis, Social-Network Graphs, Communities and Clusters, Betweenness, The Girvan-Newman Algorithm.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Preprocess the data and can perform descriptive analysis
CO 2 : Implement multivariate data analysis and Hypothesis testing
CO 3 : Able to apply Regression models to solve real world problems.
CO 4 : Develop classification methods to solve problems
CO 5 : Understand prescriptive analysis using forecasting techniques and Graph analytics

Textbooks:

1. Business Analytics, U. Dinesh Kumar, Wiley publications, 2017.
2. Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman, Jeff Ullman, Cambridge University Press, 2016.

References:

1. Data Mining: Concepts and Techniques, Jiawei Han and Micheline Kamber, 4rd edition, Morgan Kaufmann Publishers, 2023.
2. Introduction to Machine Learning, Ethem Alpaydin, Francis Bach, 3rd edition Adaptive Computation and Machine Learning series 2014.
3. Principles of Data Science, Sinan Ozdemir, Packt publications, 2016.
4. Data Science in Theory and Practice, Maria Cristina Mariani, Osei Kofi Tweneboah and Maria Pia Beccar-Varela, John Wiley and Sons publishers, 2022.
5. Introduction to Data Mining, Pang-Ning Tan, Michael Stein batch, Vipin Kumar, Pearson publications, 2016.

Course Code: 22CS304

COMPUTER GRAPHICS

(Professional Elective - I)
(Common to CSE & IT)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : The aim of this course is to provide an introduction of fundamental concepts and theory of computer graphics.
2 : It presents the important drawing algorithm, polygon fitting, clipping and 2D transformation curves and an introduction to 3D transformation.
3 : To expose the students to the current and emerging technologies such as OpenGL and visualize 2D and 3D objects.
4 : It provides the basics of Animation techniques interface which allows students to develop programming skills in CG.

Unit I – Introduction

Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations, Frame buffers, and input devices.

Output primitives: Points and lines, line drawing algorithms (Bresenham's and DDA Algorithm), mid- point circle and ellipse algorithms

Polygon Filling: Scan-line algorithm, boundary-fill and flood-fill algorithms

Unit II - 2-D Geometrical transforms

2-D geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems

2-D viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland algorithms, Sutherland – Hodgeman , polygon clipping algorithm.

Unit III - 3-D Object representation

Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves. Basic illumination models, surface rendering methods and polygon rendering methods.

Unit IV - 3-D Geometric transformations

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations.

3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

Unit V – Computer Animation

Computer animation: Design of animation sequence, general computer animation functions, raster animation, key frame systems, motion specifications.

Visible surface detection methods: Classification, back-face detection, depth-buffer, BSP-tree methods and area sub-division method K-d Tree, OCTree and Ray-casting method.

Course Outcomes: At the end of the course, the student should be able to

CO1 : Know the application areas of computer graphics, overview of graphics systems and output primitives.
CO2 : Apply 2D geometric transforms, 2D viewing using transformation matrices.
CO3 : Acquire familiarity with the relevant mathematics of computer graphics.
CO4 : Apply 3D Geometric representations and transformations.
CO5 : Be able to design basic graphics application programs, including animation.

Textbooks:

1. Computer Graphics C version, Donald Hearn and M. Pauline Baker, Pearson Education, 2002.
2. A Programming approach: Computer Graphics, Steven Harrington, Second Edition, Tata McGraw-Hill, 1987.

References:

1. "Computer Graphics Principles & practice", Foley, Van Dam, Feiner and Hughes, second edition in C, Pearson Education, 2003.
2. "Computer Graphics", Zhigang Xiang, Roy Plastaock, second edition, Schaum's Outlines, 2000.
3. "Procedural elements for Computer Graphics", David F Rogers, 2nd edition, Tata McGraw-Hill, 2001.

Course Code: 22CS305

DISTRIBUTED DATABASES

(Professional Elective - I)
(Common to CSE, IT)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester-End Exam Duration	:	3 Hours

Course Objectives:

1 : To enrich the previous knowledge of database systems and expose the need for distributed database technology to confront the deficiencies of centralized database systems.
2 : To introduce basic principles and implementation techniques of distributed database systems.
3 : To make the students understand the concurrency and serializability in transaction management system.
4 : To acquire the knowledge on distributed database reliability and parallel database design.
5 : To equip students with principles and knowledge of parallel and object-oriented databases.

Unit I

Introduction: Distributed Data Processing, Distributed Database System, Promises of DDBSs,Problem areas.

Distributed DBMS Architecture: Architectural Models for Distributed DBMS, DDBMS Architecture.

Distributed Database Design: Alternative Design Strategies, Distribution Design Issues, Fragmentation, Allocation.

Unit II

Query Processing and Decomposition: Query processing objectives, characterization of queryprocessors, layers of query processing, query decomposition, localization of distributed data.

Distributed query Optimization: Query optimization, centralized query optimization, distributed queryoptimization algorithms.

Unit III

Transaction Management: Definition, properties of a transaction, types of transactions, distributed concurrency control: serializability, concurrency control mechanisms and algorithms, time-stamped and optimistic concurrency control algorithms, deadlock management.

Unit IV

Distributed DBMS Reliability: Reliability concepts and measures, fault-tolerance in distributed systems, failures in Distributed DBMS, local and distributed reliability protocols, site failures, and network partitioning.

Parallel Database Systems: Parallel database system architectures, parallel data placement, parallel query processing, load balancing, database clusters.

Unit V

Distributed object Database Management Systems: Fundamental object concepts and models, object distributed design, architectural issues, object management, distributed object storage, object query processing.

Object Oriented Data Model: Inheritance, object identity, persistent programming languages,persistence of objects, comparison OODBMS and ORDBMS

Course Outcomes: At the end of the course, the student should be able to

CO1 : Understand theoretical and practical aspects of distributed database systems.
CO2 : Identify various issues related to the development of distributed database systems.
CO3 : Understand serializability and concurrency control in distributed databases.
CO4 : Design a fault tolerant distributed system and able to run parallel query processing.
CO5 : Understand the design aspects of object-oriented database systems and related development.

Textbooks:

1. Principles of Distributed Database Systems, M. Tamer OZSU and Patuck Valduriez: Pearson Edition Asia, 2001.
2. Distributed Databases: Principles and Systems, Stefano Ceri and Giuseppe Pelagatti: McGraw-Hill 2017.

References:

1. "Database Systems: The Complete Book", Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom Second Edition, Pearson International Edition, 2008.

Course Code: 22CS306

MICROPROCESSORS AND INTERFACING

(Professional Elective - I)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To appreciate pipeline implementation Architecture and Programming of 8086.
2 : To interface real-world peripherals with the processor and controller.
3 : To design synchronous and asynchronous communication models.
4 : To design Microcontroller architecture, Memory organization, Instruction set with simple programs.
5 : To Learn and Understand the Real time control using programming serial communication and programming timers and counters.

Unit I - 8086 Architecture

8086 Architecture–Functional Diagram, Register Organization, Memory Segmentation, Signal Descriptions of 8086, Minimum and Maximum Modes, Physical Memory Organization, Timing Diagrams, Addressing Modes 0f 8086, Instruction set of 8086, Assembler Directives.

Unit II - Memory and I/O Interfacing

SRAM Interfacing and DRAM Interfacing, 8255 PPI Architecture, Various Modes of Operation of 8255 and Interfacing with 8086, Displays, Keyboard Interfacing, Interfacing Analog to Digital converter: ADC 0808/0809, Interfacing Digital to Analog converter: DAC 0800.

Unit III - Interrupts and Serial Communication Interface

Interrupt Structure of 8086, Vector Interrupt Table, Interrupt Service Routine, Interrupt Controller 8259 Architecture and interfacing with 8086. Introduction to DOS and BIOS Interrupts.

Serial Communication Standards, Serial Data Transfer Schemes, 8251 USART Architecture and Interfacing, RS-232.

Unit IV - 8051 Microcontroller

Overview of 8051 Microcontroller, Architecture, I/O ports, Memory Organization, Addressing Modesand Instruction Set of 8051, Simple Programs.

Unit V - 8051 Real Time Control

Interrupts, timer/counter and serial communication, Programming Timer Interrupts, Programming external hardware interrupts, programming the serial communication interrupts, Programming 8051 timers and counters.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Gain knowledge about pipelined processor 8086 and understand timing diagrams.
CO 2 : Interface the processor with peripheral devices.
CO 3 : Gain knowledge about interrupt structure and serial communication of 8086 microprocessor.
CO 4 : Master the 8051 architecture and programming

CO 5 : Implementing various real time controls like timers, interrupts, serial communications in 8051 microcontroller.

Textbooks:

1. Advanced Microprocessors and Peripherals, A. K. Ray and K. M. Bhurchandani, 2nd Edition, Tata McGraw-Hill, 2006.
2. The 8051 Microcontroller, Kenneth J. Ayala, 3rd Edition, Cengage Learning, 2010.

References:

1. Microprocessor and Interfacing, D.V. Hall, 2nd Edition, Tata McGraw-Hill, 2006.
2. Micro Computer system: 8086/8088 Family Architecture, Programming and Design, Liu and G. A. Gibson, 2nd Edition, Prentice Hall, 1986.
3. The 8051 Microcontroller and Embedded Systems, Muhammad Ali Mazidi and Janice Gillispie Mazidi, 2nd Edition, Pearson, 2008.

Course Code: 22CS307

PRINCIPLES OF PROGRAMMING LANGUAGES

(Professional Elective - I)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To Learn and understand Programming Language concepts, Environments, and syntax and semantics.
2 : To Understand conceptual design of high-level language implementation using data types, expressions, and statements.
3 : To implement the subprograms, co-routines using functions, design issues, abstract datatypes
4 : To understand and implement programs with concurrency, exception handling in various programing languages C++, Java, and C#.
5 : To learn and understand the Functional and imperative languages using LISP, scripting languages, key concepts, and Case study PHP programs.

Unit I – Preliminary Concepts

Preliminary Concepts: Reasons for Studying Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-Offs, Implementation Methods, Programming Environments
Syntax and Semantics: General Problem of Describing Syntax and Semantics, Formal Methods of Describing Syntax, Attribute Grammars, Describing the Meanings of Programs.

Unit II – Data Types, Expressions and Statements

Names, Bindings, and Scopes: Introduction, Names, Variables, Concept of Binding, Scope and Lifetime, Referencing Environments, Named Constants

Data Types: Introduction, Primitive Data Types, Character String Types, User defined Ordinal Types, Array, Associative Arrays, Record, Union, Tuple Types, List Types, Pointer and Reference Types, Type Checking, Strong Typing, Type Equivalence

Expressions and Statements: Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short Circuit Evaluation, Assignment Statements, Mixed-Mode Assignment Control Structures – Introduction, Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands.

Unit III - Subprograms

Subprograms and Blocks: Fundamentals of Sub-Programs, Design Issues for Subprograms, Local Referencing Environments, Parameter Passing Methods, Parameters that are Subprograms, Calling Subprograms Indirectly, Overloaded Subprograms, Generic Subprograms, Design Issues for Functions, User defined Overloaded Operators, Closures, Coroutines Implementation.

Subprograms: General Semantics of Calls and Returns, Implementing Simple Subprograms, Implementing Subprograms with Stack-Dynamic Local Variables, Nested Subprograms, Blocks, Implementing Dynamic Scoping

Abstract Data Types: The Concept of Abstraction, Introduction to Data Abstraction, Design Issues, Language Examples, Parameterized ADT, Encapsulation Constructs, Naming Encapsulations.

Unit IV – Concurrency, Exception Handling

Concurrency: Introduction, Introduction to Subprogram Level Concurrency, Semaphores, Monitors, Message Passing, Java Threads, Concurrency in Function Languages, Statement Level Concurrency.

Exception Handling and Event Handling: Introduction, Exception Handling in Ada, C++, Java, Introduction to Event Handling, Event Handling with Java and C#.

Unit V – Functional and Imperative Languages

Functional Programming Languages: Introduction, Mathematical Functions, Fundamentals of Functional Programming Language, LISP, Support for Functional Programming in Primarily Imperative Languages, Comparison of Functional and Imperative Languages.

Logic Programming Language: Introduction, an Overview of Logic Programming, Basic Elements of Prolog, Applications of Logic Programming.

Scripting Language: Pragmatics, Key Concepts.

Case Study: PHP – Values and Types, Variables, Storage and Control, Bindings and Scope, Forms creation, Inheritance in PHP.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Understand the role of theoretical formalisms, such as BNF for syntax and operational and denotational semantics for semantics
CO 2 : Understand the salient features in the landscape of programming languages like Data Types, Bindings, scope, expressions etc.
CO 3 : Analyze the different ways of sub program execution and ADT implementation across different programming languages.
CO 4 : Comprehend Concurrency and Exception Handling concepts and their implementation in various programming languages.
CO 5 : Understand the basics of functional programming languages logic programming languages and scripting languages.

Textbooks:

1. Concepts of Programming Languages 10/E, Robert. W. Sebesta, Pearson Education, 2016.
2. Programming Language Design Concepts, D. A. Watt, Wiley Dreamtech, 2007.

References:

1. Programming Languages, A.B. Tucker, R. E. Noonan, 2nd Edition, TMH,2007.
2. Programming Languages, K. C. Louden, 2nd Edition, Thomson, 2003
3. Introduction to Data Communications and Networking, W.Tomasi, Pearson Education, 2009.
4. Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson Education, 2008.

Course code: 22CS308

DIGITAL IMAGE PROCESSING AND COMPUTER VISION

(Professional Elective-I)
(Common to CSE, IT)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To offer background knowledge about Image representations and elementary transformations.
2 : To make the students understand the essential functionality of Image Processing.
3 : To introduce various transformations and filters for image processing enhancement, Restoration Segmentation.
4 : To develop statistical intuitions for extracting and representing the image features.

Unit I – Digital Image Fundamentals

Introduction, Origin, Steps in Digital Image Processing, Components, Elements of Visual Perception, Image Sensing and Acquisition, Image Sampling and Quantization, Relationships between pixels.

Unit II – Image Enhancement

Spatial Domain: Basic intensity transformation functions, Histogram processing, Fundamentals of Spatial Filtering, Smoothing and Sharpening Spatial Filtering.

Frequency Domain: Basics of filtering in frequency domain, Image Smoothing using lowpass frequency domain filters and Image Sharpening using high pass Filters.

Unit III – Image Restoration

Restoration: Noise Models, Restoration in the presence of Noise only- Spatial filtering, Periodic noise reduction, Frequency domain filtering, Inverse filtering, Wiener filtering, Reconstruction from projections.

Colour Image Processing: Colour models, Basics of full Colour image processing, Colour transformations, Colour image smoothing and Sharpening,

Unit IV – Image Compression and Segmentation

Image compression: Fundamentals, Huffman coding, Arithmetic coding, Run length coding, Predictive coding.

Morphological Processing: Erosion and Dilation.

Segmentation: Point, Line and Edge detection, Thresholding, Segmentation by Region Growing, Region splitting and merging.

Unit V – Feature extraction and Computer Vision

Boundary Preprocessing: Chain Code, Polygonal Approximation, Signature.
Boundary feature descriptors: Shape Number, Fourier Descriptor, Moments.
Regional feature Descriptors: Topological Feature, Texture, Corner Detection, Rectangle detection. Scale Space and Scale Selection; SIFT, SURF; HoG

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Understand the image representation and relationship between the pixels.
CO 2 : Design and develop filters for image enhancement in the Spatial and Frequency domains to improve quality of an image.
CO 3 : Implement restoration techniques to restore degraded images and Compressions techniques to reduce the size.
CO 4 : Apply Segmentation techniques to perform image into meaningful regions.
CO 5 : Demonstrate the Feature extraction methods and solve computer vision-based problems.

Textbooks:

1. Woods, Digital Image Processing, Rafael C. Gonzalez and Richard E. Fourth edition. Pearson Education, July 2018.
2. Computer Vision: A Modern Approach, Forsyth, A., D. and Ponce, J., Pearson Education, 2nd Edition, 2012.

References:

1. Pattern classification, Richard Duda. Hart and David Strok, John Wiley publishers, Second Edition 2012.
2. Image Processing. Analysis and Machine Vision, Milan Sonka, Vaciav Hlavac and Roger Boie, Second Edition, Thomson learning, 2001.
3. Pattern Recognition and Image Processing, Signal Processing and Communications Series, Sing-T. Bow, Second Edition, January 2002.

Course code: 22CS331

ALGORITHMS AND DATA SCIENCE LAB

Instruction	:	2 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1	Semester End Exam Duration	:	3 Hours

List of Experiments:

Algorithms

Task 1:

a) Implement Merge sort algorithm and plot its time complexity with reference to the size of the input.

b) Implement Quick sort algorithm and plot its time complexity regarding asymptotic notations (Best, average, and worst).

Task 2:

Write a program to identify the articulation points present in a graph.

Task 3:

Implement Job Sequencing with deadlines algorithm.

Task 4:

Implement Fractional Knapsack Algorithm.

Task 5:

Implement Dijkstra's algorithm to compute the shortest path through a graph.

Task 6:

Implement OBST using dynamic programming.

Task 7:

a) Implement sum of subset algorithm

b) Implement N-queen algorithm.

Task 8:

Implement Prim's algorithm.

Task 9:

Implement Kruskal's algorithm.

Task 10:

Implement graph coloring algorithm.

Data Science

Task 1:

1. Introduction to various libraries of Python for analyzing the datasets.
2. Exploring the various methods of Pandas and Numpy libraries on datasets.
3. Pre-processing and Visualization of the data.

Task 2:

1. Demonstrate the dimensionality reduction technique (PCA) on suitable dataset and compare the results before and after the reduction.
2. Demonstrate the dimensionality reduction technique (RFE) on suitable dataset and compare the results before and after the reduction.

Task 3:

1. Demonstrate Apriori Frequent Item Set Mining Algorithm on supermarket dataset to list the top 10 Association rules. Comment on the performance of the algorithm for different support and confidence threshold values.
2. Demonstrate FP-Growth Frequent Item Set Mining Algorithm on supermarket dataset to list the top 10 Association rules. Comment on the performance of the algorithm for different support and confidence threshold values.

Task 4:

1. Design and Demonstrate Regression model to predict the age of a person. Evaluate the performance of the model.

Task 5:

1. Implement the Decision Tree Classification model on Soybean dataset. Estimate the accuracy of model.
2. Implement the K-Nearest Neighbor Classification Technique on iris dataset. Estimate the accuracy of the model.

Task 6:

1. Design and implement Random Forest Classification model to predict if a loan will get approved or not for a bank customer dataset. Estimate the accuracy of the model.

Task 7:

1. Implement K-Means Clustering technique on a suitable dataset. Compute the performance measures of the clustering.
2. Implement Hierarchical Clustering technique on weather dataset. Compute the performance measures of the clustering.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Solve Sorting problems using Divide and Conquer Model.
CO 2 : Solve fractional knapsack, job sequencing with deadlines and single source shortest path problem using Greedy Method.
CO 3 : Solve OBST in dynamic programming and N-queen using backtracking
CO 4 : Demonstrate Data Science python libraries and pre-process the given data.
CO 5 : Design and Develop Prediction, Classification and Clustering models for real world applications

References:

1. Fundamentals of Algorithms, E. Horowitz and S.Sahni, 2nd edition, Galgotia Publications, 2010
2. Business Analytics, U. Dinesh Kumar, Wiley publications, 2017.

Course Code: 22IT331

WEB TECHNOLOGIES LAB

(Common to IT, CSE, CSE(AI&ML), CSE(CS) and CSE(DS))

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To learn the basics of HTML elements
2 : To learn the basics of java Console and GUI based programming
3 : To introduce XML and processing of XML Data with Java
4 : To introduce Server-side programming with Java Servlets and JSP
5 : To introduce Client-side scripting with JavaScript and AJAX.

Exercises:

1. Create a web page using the advanced features of CSS: Grid, Flexbox. And apply transition and animations on the contents of the web page
2. Make the web pages created in the above experiment as responsive web page with Bootstrap Framework.
3. Validate the registration, user login, user profile and payment pages using JavaScript. Make use of any needed JavaScript objects.
4. Build a scientific calculator.
5. JavaScript Program to demonstrate working of prototypal inheritance, closure, callbacks, promises and async / await.
6. Write an XML file which will display the Book information with the following fields: Title of the book, Author Name, ISBN number, Publisher name, Edition, Price
7. Define a Document Type Definition (DTD) and XML schema to validate the above created XML Documents
8. Write a java program to establish a connection to a database and execute simple SQL queries.
9. Write a java program to demonstrate the usage of JDBC in performing various DML statements. Use prepared statements and callable statements.
10. Write a java-based application to demonstrate the Updatable and Scrollable result sets.
11. Write a java program to access meta data of the SQL database.
12. Write a program to accept request parameters from a form and generate the response.
13. Write a program to accept Servlet Config and Servlet Context parameters.
14. Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and, pwd4 respectively. Write a servlet for doing the following functionalities

    a. Create a Cookie and add these four user ids and passwords to this Cookie.
    b. Read the user id and password entered into the Login form and authenticate with the values (user id and passwords) available in the cookies. If the person is a valid user (i.e., user-name and password match) you should welcome by name (user-name) else you should display the message "You are not an authenticated user ".
15. Develop a servlet to demonstrate the database access and update from a database.
16. Create a servlet to implement an authentication filter mechanism.
17. Develop a servlet to implement servlet context and session listeners.
18. Write a JSP which does the following job:
    a. Insert the details of the three users who register with the web site by using registration form.
    b. Authenticate the user when he submits the login form using the username and password from the database.
19. Write a JSP to demonstrate the usage of JSP standard actions.
20. Write a JSP to show the usage of various scripting elements.
21. Design and use a custom tag library.
22. Design a simple application using both Servlets and JSPs along with database access.

Note: Programs from 1 to 14 are mandatory and Programs from 15 to 22 are optional.

Course Outcomes: At the end of the course, the student will be able to

CO 1 : To build a custom website with HTML, CSS, and Bootstrap.
CO 2 : Demonstrate JavaScript, XML, DHTML and related Technologies.
CO 3 : Implement the Database Connectivity and Component Technologies like Beans.
CO 4 : Deploy the servlet technology & API Management.
CO 5 : Construct the fundamentals of JSP.

References:

1. Beginning HTML, XHTML, CSS, and JavaScript, Jon Duckett, Wrox Publications, 2010
2. Head First Servlets and JSP, Bryan Basham, Kathy Sierra and Bert Bates, O'Reilly Media, 2nd Edition, 2008.
3. Core Java: Volume II – Advanced Features, Cay Horstmann and Gary Cornell, Prentice Hall, 9th Edition, 2013 (Only Chapter 4 for Database Programming)

Course Code: 22CS332

CASE TOOLS LAB

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To understand the role of formal specifications in project design and be able to develop such specifications.
2 : To be able to design an interface and develop a prototype for a complex software system.
3 : To understand the role of testing in the software development cycle and be capable of developing a test plan.
4 : To be aware of and able to use Computer Aided Software Engineering (CASE) tool.

LIST OF PROGRAMS:

Task 1:

ATM Case Study and Online Railway Reservation System
Phases in software development project, overview and need. Understand problems in existing systems and perform system analysis: Requirement analysis, SRS

Task 2:

Requirement analysis, SRS of both case studies

Task 3:

To perform the function-oriented design: Data flow diagrams

Task 4:

To perform the function-oriented design: Structured chart

Task 5:

To perform the user's view analysis: Use case diagram

Task 6:

To draw the structural view diagram: Class diagram, object diagram.

Task 7:

To draw the behavioral view diagram: Sequence diagram, Collaboration diagram

Task 8:

To draw the behavioral view diagram: State-chart diagram

Task 9:

To draw the behavioral view diagram: Activity diagram

Task 10:

To draw the implementation view diagram: Component diagram.

Task 11:

To draw the implementation view diagram: deployment diagram

Task 12:

Version Control System, GIT

1. Working Locally with GIT
2. Working Remotely with GITHUB
3. Branching and Merging
4. Resolve merge Conflict
5. GIT reset and Stash operation
6. How to setup Git on Premises Hardware

Task 13:

1. Introduction about Maven project
2. Build, test and deploy a simple application in Maven.

Task 14:

Continuous Integration using Jenkins

1. Introduction of Jenkins
2. Install and setup Jenkins
3. Continuous Build and Deployment.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Develop requirement specifications for a software problem in hand
CO 2 : Perform functional oriented and object-oriented design
CO 3 : Implement the concepts of object oriented to develop a real-world application
CO 4 : Prepare test cases to rigorously test the application for ensuring quality.
CO 5 : Integrate developed code using Jenkins to simulate a CI/CD pipeline

References:

1. Software Engineering, Ian Sommerville , 9TH Edition, 2004.
2. Fundamentals of Software Engineering, Carlo Ghezzi, Jazayeri Mehdi, Mandrioli Dino, Prentice Hall, 2003.
3. The Unified Development Process, Ivar Jacobson, Grady Booch, James Rumbaugh, Pearson Education, 1999
4. Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Pearson Education, 1994.
5. Software Metrics: A Rigorous and Practical Approach, Norman E. Fenton, Shari Lawrence Pfleeger, PWs Pub, 1997.

Course Code: 22IT332

MOBILE APPLICATION DEVELOPMENT LAB

(Common to IT, CSE)

Instruction	: 3 Periods / week	Continuous Internal Evaluation	: 40 Marks
Tutorial	: –	Semester End Examination	: 60 Marks
Credits	: 1.5	Semester End Exam Duration	: 3 Hours

Course Objectives:

1 : To understand the fundamentals of the Kotlin programming language
2 : To acquire the skills of using Android Software Development Tools and categorize the different layouts.
3 : To use common Android UI components to create a basic User Interface, handle user input and Android lifecycle events.
4 : To enhance the ability to navigate across the various screens of application.
5 : To create well-designed Android applications using Kotlin that can connect to the Internet, store data.

Exercises

Task 1: Introduction to Kotlin

1. Write a Kotlin program that takes a nullable integer as input and print its Square if it is not null, or "Input is null" otherwise
2. Implement a number guessing game in which the user is prompted to enter a number between 1 and 100 until he or she guesses correctly. After every wrong guess, the user is told whether the guess was too high or too low.
3. Create a function in Kotlin that takes a name as input and prints the greeting message. Make the message customizable and provide a default message if no custom message is provided.

Task 2: Kotlin's Object Oriented Concepts

1. Create a Kotlin application for Rolling die using classes.
2. Create a Kotlin application to demonstrate the companion objects, getter and setter properties.

Task 3: Creation of Android Application

1. Create "Hello World" application. That will display "Hello World" in the middle of the screen in the red color with white background.
2. Write a program to demonstrate Activity Lifecycle.

Task 4: Understanding Activity and Intent

1. To understand Activity and Intent create a sample application with login module. (Check username and password), on successful login, go to next screen and on failing login, alert the user using Toast. Also pass username to next screen.
2. Create a Dice Roller Android app that has a Button to roll a dice and update the image on the screen.

Task 5:

1. Create a program with different types of dwellings (Shelters people live in like round hut, square cabin, round tower) that are implemented as a class hierarchy.
2. Create a tip calculator app with a working Calculate button.

Task 6:

1. Create a polished Affirmations app that uses a Recycler View to display a list of cards. Each card contains an image and affirmation text.
2. Create a dictionary app to implement navigation between screens using intents and adding an options menu.

Task 7:

1. Create a Words app to use a single activity and multiple fragments, and navigate between fragments with the Navigation Component.
2. Create an Unscramble game app where the user can guess the scrambled words. Use Live Data for the app's data (word, word count and the score) in the Unscramble app.

Task 8:

1. Create a Cupcake app that displays an order flow for cupcakes, allowing the user to choose the cupcake flavor, quantity, and pickup date
2. Create a cupcake ordering app that allows the user to send the order to another app and allows for canceling an order.

Task 9:

1. Create an Android Trivia app illustrates navigation patterns and controls. The app has several components:
2. In the title screen, shown on the left in the screenshot above, the user starts the game.
3. In the game screen with questions, shown in the middle above, the user plays the game and submits their answers.

Task 10:

1. The navigation drawer, shown on the right above, slides out from the side of the app and contains a menu with a header. The drawer icon opens the navigation drawer. The navigation-drawer menu contains a link to the About page and a link to the rules of the game.

Task 11:

1. Create the Guess the Word app, beginning with starter code. Guess The Word is two-player charades-style game, where the players collaborate to achieve the highest score possible.

Task 12:

1. Build an Inventory app that saves inventory items into the SQLite database.

Task 13:

1. Build an app that tracks sleep quality using database. The app uses a database to store sleep data over time. The app has two screens, represented by fragments.
2. The first screen, shown on the left, has buttons to start and stop tracking. The second screen, shown on the right, is for selecting a sleep-quality rating.

Task 14:

1. Create an app called Mars Real Estate, which shows properties for sale on Mars. This app connects to a web service to retrieve and display the property data, including details such as the price and whether the property is available for sale or rent

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Compare and contrast the language fundamentals of Kotlin.
CO 2 : Use Android Layouts and Connect Views with data through data binding.
CO 3 : Illustrate the Android Lifecycle mechanism and application architecture.
CO 4 : Develop the navigation patterns and display collections of data using recycler View.
CO 5 : Collaborate data persistence in the application, and use APIs to connect to internet, to store and retrieve data.

Textbooks:

1. Android Programming with Kotlin for Beginners, John Horton, Packt, 2019.

References:

1. Kickstart Modern Android Development with Jetpack and Kotlin, Catalin Ghita, PACKT PUBLISHING LIMITED, 2022.
2. https://developer.android.com/courses/android-basic-kotlin/course

Course code: 22CS351

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

(Common to CSE, IT)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To develop an understanding of the basic concepts of Artificial Intelligence.
2 : To analyze the nature of various advanced search strategies in AI.
3 : To design and develop model-based techniques to solve classification and clustering problems.
4 : To study the experiential leaning models and designing of adaptative models.

Unit I - Introduction

Introduction to AI, Intelligent Agents.

Solving Problems by Search: Uninformed Search Strategies: Breadth-first search, Uniform cost search, Depth-first search.

Informed (Heuristic) Search Strategies: Greedy best-first search, A* search, Heuristic Functions.

Search in complex environments: Local search and Optimization problems, Local Search in Continuous Spaces, Searching with Non-Deterministic Actions.

Adversarial Search: Optimal decisions in games, Heuristic Alpha-Beta search, Monte Carlo tree search.

Unit II - Knowledge, Reasoning and Planning

Logic Agents: Based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem Proving.

First-Order Logic: Syntax and
Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic.

Inference in First-Order Logic: Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

Unit III - Knowledge Representation

Knowledge Representation: Ontological Engineering, Categories and Objects, Events. Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.

Machine Learning: Attributes, categories of attributes, Estimation of Missing Values Classification, Types of Machine learning

Classification: Basic concepts, Naive Bayes, Decision Tree, Classification Trees, Pruning, Rule Extraction form Tree.

Unit IV - Supervised and Unsupervised Learning

Ensemble Methods: Bagging- Random Forest, Boosting- Gradient Boosting and Ada-boost.

Clustering: Basic Concepts and Methods, Cluster Analysis, Requirements,

Basic clustering methods: Partitioning Methods- k-Means, k-Medoids,
Hierarchical Methods- Agglomerative versus Divisive Hierarchical Clustering.
Distance Measures in algorithmic method- BIRCH, Chameleon

Evaluation of Clustering: Assessing Clustering Tendency, Determining the Number of Clusters, Measuring Clustering Quality.

Unit V - Reinforcement Learning

Introduction to Reinforcement Learning

Elements of RL: Agent, Environment, Reward

Markov Decision Processes (MDPs): Markov Processes and Markov Reward Processes

Introduction to MDPs: States, Actions, Rewards, Policy

Dynamic Programming: Policy Evaluation and Improvement, Value Iteration and Policy Iteration. **Temporal Difference Learning and Q-Learning:** TD Prediction, TD(0) and TD(λ), Q-Learning, Off-policy control using Q-learning.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Identify the scope for agent-based engineering solutions using AI based tools
CO 2 : Demonstrate advanced search strategies, perform search space reduction techniques using minmax algorithm
CO 3 : Apply knowledge representation, reasoning to AI-based solutions
CO 4 : Develop multiple-class classifiers and develop recommender systems.
CO 5 : Implement simple Q-Learning algorithm, based on Value iteration, and appreciate actor-critic model

Textbooks:

1. Artificial Intelligence: A Modern Approach, Stuart Russel and Peter Norvig, 4th Edition, Pearson, 2020.
2. Introduction to Machine Learning, Ethem Alpaydin, Francis Bach, 3rd edition Adaptive Computation and Machine Learning series 2014.

References:

1. Machine Learning, Anuradha Srinivasa Raghavan, Vincy Joseph, Paperback, Kindle edition, 2019.
2. Machine Learning using Python, Manaranjan Pradhan U Dinesh Kumar, Paperback, Wiley Publication, 2019.
3. Machine learning, Tom Mitchel, 2012 edition.

Course Code: 22CS352

FULL STACK DEVELOPEMENT

Instruction	: 3 Periods / week	Continuous Internal Evaluation	: 40 Marks Semester
Tutorial	: –	End Examination	: 60 Marks
Credits	: 3	Semester End Exam Duration	: 3 Hours

Course Objectives:

1 : To enhance code readability, maintainability, and developer productivity by adopting ES6 features and explore version control system.
2 : To gain proficiency and expertise in utilizing NPM (Node Package Manager) and developing RESTful APIs using Express framework.
3 : To explore how NoSQL databases, particularly MongoDB, can be leveraged to design scalable and flexible data solutions.
4 : To Gain proficiency in front-end development using Angular.
5 : To provide the knowledge of building modular and reusable components and create interactive user interfaces for MERN stack applications.

Unit I – Working with ES6 Features

Version Control System: Git, creating local and remote repositories. **Git commands:** init, status, add, commit, remote, push, pull, clone, Git Branching. Using **github.com** for collaborative software development.

ES6 Features extended: variable declaration using var, let, const, Prototypal Inheritance, Classes, Objects, Destructuring, Modules, Symbols, Function Generators, Understanding Callbacks, Closure, Promise, XHR: response, Asynchronous Task in JS, using async / await.

Unit II – Server-side JavaScript – Node JS

Getting started with Node.js: Introduction to Node.js, REPL, NPM. **Node Modules:** os, path, util, events, fs, buffers, streams, http. Building own API and consuming it. (**REST full API**).

Building an Express web application:
Introduction to Express, Installation of Express, create first Express application, application, request, and response objects, Configuring an Express application, rendering views using EJS.

Unit III – Data storage and manipulation

No SQL: Introduction to NoSQL, SQL vs NoSQL, Migrating from SQL to NoSQL database. Different Types of NoSQL databases, CAP Theorem, Sharding.

MongoDB: Introduction to MongoDB, Key features of MongoDB, MongoDB shell, MongoDB databases, MongoDB collections, MongoDB CRUD operations, Real – time database Firebase CRUD operations.

Express-MongoDB connectivity: Connect Express application with MongoDB using mongoose library. Managing user authentication, understanding authorization using Passport OAuth / JWT.
Type Script: Introduction to Type Script, basic types and any type, compiler options, Classes, Interfaces, Generics, Decorators.

Unit IV – Client-side JS Framework Angular

Angular Development: Introduction to Angular, versions, Angular Architecture, Components, Modules, Directives. **Data Binding:** Interpolation, Property Binding, Class Binding, Event Binding, Two – way data binding.

Angular Services: Creating Services, Creating API and Consuming, Dependency Injection.
Routing: Routes, Router Outlet, Router Links. Using HttpClient, Observables, Pipes.

Angular Forms: Template-driven forms and Reactive Forms, passing data from parent to child and passing data between siblings.

Unit V – Client-side JS Library React

Introduction to React JS: Motivation for using React, Key differentiators (Virtual DOM, One-way binding), JSX. **React Components:** Functional Component, Class Component, Render function, Component Lifecycle, State, Props.

Components inter communication: Pass data from parent to child, Pass data from child to parent, fetching data from an API using Axios. **React Routing:** Form Validations, Posting Data, React Router, Building & Deploying React App.

Course Outcomes: At the end of the course, the student will be,

CO 1 : Proficient in leveraging ES6 capabilities to streamline development tasks, improving code efficiency, and solving programming challenges with a modern JavaScript mindset and managing various versions of the product using git.

CO 2 : Able to develop the skills to construct scalable and maintainable web applications by harnessing the asynchronous, event-driven nature of Node.js and the modular architecture provided by Express.

CO 3 : Designing and implementing efficient NoSQL database solutions, understanding when and how to use document-oriented databases like MongoDB to address specific application requirements.

CO 4 : Developing the dynamic and single-page web applications using the MEAN stack and understand how to manage client-side routing, and UI components to deliver a seamless user experience.

CO 5 : Excelling in both server-side and client-side aspects of web application development using MERN stack.

Textbooks:

1. MEAN Web Development, Amos Q. Haviv, Second Edition, Packt Publications, November 2016.
2. "Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node", Vasan Subramanian, 2nd Edition, APress, 2017.

References:

1. Learning Node: Moving to the Server-Side, Shelly Powers, 2nd Edition, O'REILLY, 2016.
2. Getting MEAN with Mongo, Express, Angular, and Node, Simon D. Holmes and Clive Harber, Second Edition, Manning Publications, 2019.
3. Node.js, MongoDB and Angular Web Development, Brad Dayley, 2nd Edition, Addison-Wesley Professional, 2017.
4. https://angular-2-training-book.rangle.io.
5. https://www.atlassian.com/git
6. https://www.typescriptlang.org/docs/handbook/basic-types.html
7. https://firebase.google.com

Course Code: 22CS353

CLOUD COMPUTING AND DEVOPS

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours
			-		

Course Objectives:

1 : To impart concepts in fundamentals of cloud computing, principles, characteristics, and cloud deployment models.
2 : To Learn and understand Virtualization concepts.
3 : To explore various case studies on cloud service providers.
4 : Understand configuration management; continuous integration deployment, delivery and monitoring using DevOps tools such as Git, Jenkins in a practical, hands-on and interactive approach can be explored.

Unit I – Cloud Computing Concepts

Principles of Parallel and Distributed Computing, Introduction to Cloud computing, Cloud computing architecture, Cloud concepts and technologies, Cloud benefits and challenges, Cloud service delivery models – Infrastructure as a Service, Platform as a Service, Software as a Service, Cloud deployment models – public, private, hybrid.

Unit-II - Virtualization and Containers

Introduction, Characteristics of Virtualized Environments, Taxonomy of Virtualization Techniques, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples, Contemporary Virtualization Techniques, Containers, Container Orchestration, Docker and Kubernetes, Terraform.

Unit III – Cloud Case Studies

Case studies of Google Cloud Platform, Cloud Toolkit , Google App Engine, Amazon Web Services, Microsoft Windows Azure .

Unit IV – Introduction to CI / CD

Introduction to DevOps: What Is DevOps? DevOps Importance and Benefits, DevOps Principles and Practices, 7"Cs of DevOps Life Cycle for Business Agility, Continuous Planning, Continuous Development, Continuous Integration, Continuous Deployment, Continuous Testing, Continuous Delivery and Monitoring, Continuous Feedback, DevOps and Continuous Testing, Steps to be followed to choose the right DevOps Tools, Selecting the Right Tools, Challenges with DevOps Implementation, Must Do Things for DevOps.

Unit V – Devops Tool Suites

Tool Suites: Atlassian Tools, Phabricator

Orchestration: Jenkins-Features, Example of Reference Architecture, Microsoft TFS-Features, Example of Reference Architecture, TeamCity- Features, Example of Reference Architecture, Ansible Features, Example of Reference Architecture, Bamboo- Features, Example of Reference Architecture.

Source Code Management and Quality: Bitbucket- Features, Example of Reference Architecture, Crucible- Features, Example of Reference Architecture.

Course Outcomes: At the end of the course, the student should be able to:

CO 1: Understand the Cloud computing fundamental concepts.
CO 2: Learn about Virtualization, and contemporary virtualization concepts like Containers, and Dockers.
CO 3: Explore Cloud Computing case studies like Amazon Cloud, Google App Engine, and Microsoft Azure.
CO 4: Analyze need for continuous integration and appreciate continuous deployment in Industrial Scenario
CO 5: Implement GitHub and Jenkins for configuration management and continuous integration.

Textbooks:

1. Cloud Computing: Principles and Paradigms, Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, Wiley, 2011.
2. Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, S.Thamarai Selvi, McGraw-Hill, 2017.

References:

1. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Kumaraswamy, Shahed Latif, O'Reilly, SPD, RP 2011.
2. DevOps Tools from Practitioner's Viewpoint, Deepak Gaikwad, Viral Thakkar, Wiley, 2019.
3. Essentials of cloud Computing: K. Chandrasekhran, CRC press, 2014.
4. Distributed and Cloud Computing, Kai Hwang, Geoffery C. Fox, Jack J. Dongarra, Elsevier, 2012.

Course code: 22CS354/22CS309

AUTOMATA THEORY AND COMPILER DESIGN

(Common to CSE, CSE(AI&ML), CSE(CS) & CSE(DS))

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To introduce the concepts of regular languages, finite automata, regular expressions, and context free grammar.
2 : To make the students understand and implement top-down and bottom-up parsers.
3 : To make the students understand the intermediate code forms, type checking.
4 : To acquire knowledge on storage allocation strategies, symbol table management and code generation algorithms.

Unit I – Introduction to Automata

Languages, definitions, Regular Expressions, Regular Grammars, Acceptance of Strings and Languages, Finite Automaton Model, DFA, NFA, conversion of NFA to DFA, Conversion of Regular Expression to NFA, Chomsky hierarchy of Languages.

Unit II– Lexical Analysis and Top-down Parsing

Phases of compilation overview, Pass, Phase, Interpretation, Bootstrapping.
Context free grammars, Top-down Parsing: Parse Trees, Ambiguous Grammars, Backtracking, LL (1), Recursive Descent parsing, Predictive parsing, pre-processing steps for predictive processing.

Unit III – Bottom-Up Parsing and Syntax Directed Translation

Bottom-up parsing and handle pruning, LR (k) grammar parsing, LALR (k) grammars, Error Recovery in parsing, parsing ambiguous grammars, YACC parser generator.
Syntax Directed Translation, Attribute Grammars, Evaluation order for SDDs, Syntax Directed Translation schemas, Intermediate source program forms - AST, polish notation and 3 address code, DAG, Types and declarations, Type Checking, Equivalence of type expressions.

Unit IV – Code Optimization

Symbol table format, organization, Block structured languages, hashing, Block structure and non- block structure storage allocation: static, runtime and heap allocation for arrays, strings, and activation records.

Consideration for optimization, Scope of optimization, DAG representation, Basic blocks, partitioning into basic blocks, flow graphs, Compile Time Evaluation, Common Subexpression elimination, dead code elimination, Strength Reduction, Code Movement, Loop Invariant Method, Loop Fusion, Loop Unrolling, Induction Variables and Reduction in Strength.

Unit V – Code Generation

Absolute Code, Assembler Code, Register and Address Descriptors, Implementing Global Register Allocation, Usage Counts, Using DAG for register allocation, Simple Code generation Algorithm, Generic Code generation Algorithm.

Course Outcomes: At the end of the course, the student should be able to

CO 1: Covert NFA to DFA and regular expression to DFA.
CO 2: Design top-down and bottom-up parsers.
CO 3: Understand the concepts of type checking and intermediate code generation forms.
CO 4: Understand the concepts of storage allocation strategies, symbol table management and hashing
CO 5: Use DAG for generating assembly language code and able to generate relocatable machine code.

Textbooks:

1. Introduction to Automata Theory Languages and Computation, Hopcroft H.E. and Ullman J.D., Pearson Education, 2009.
2. Principles of Compiler Design, A.V Aho and J D Ullman, Pearson Education, 2002.

References:

1. Compiler Construction: Principles And Practice, Kenneth C. Louden, Thomson/D elmar Cengage Learning, 2006.
2. Lex & Yacc, Doug Brown, John Levine and Tony Mason, 2nd Edition, O'reilly Media, 1992.
3. Engineering a compiler, Keith Cooper and Linda Torczon, 2nd Edition, Morgan Kaufmann,2011.
4. Modern Compiler Construction in C, Andrew W. Appel, Cambridge University Press, 2004

Course code: 22CS355

CRYPTOGRAPHY AND ESSENTIALS OF NETWORK SECURITY

(Professional Elective - II)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : Understand the various Symmetric and Asymmetric Cryptographic algorithms.
2 : Appreciate various Key Management and Distribution along with Authentication schemes.
3 : Understand the different protocols used for Network security and System security.
4 : Introduce the different aspects of Application-level security.

Unit – I- Security and Cryptography concepts

Security Concepts: Introduction, Security Attacks, Security Services, Security Mechanisms, A model for Network Security.

Cryptography Concepts and Techniques: Introduction, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric keycryptography, steganography.

Symmetric key Ciphers: Symmetric Encryption principles, DES, AES, Block cipher modes of operation, Stream ciphers, RC4.

Unit – II- Introduction to Modular Arithmetic

Asymmetric key Ciphers: Public key cryptography principles, RSA algorithm, Elgamal Cryptography, Diffie-Hellman Key Exchange.
Key Management and Distribution: Symmetric Key Distribution Using Symmetric & Asymmetric Encryption, Distribution of Public Keys, Kerberos, X.509 Authentication Service, Public – Key Infrastructure

Unit – III-Cryptographic Hash Functions

Approaches to Message Authentication, Secure Hash Algorithm (SHA-512).
Message Authentication Codes: Authentication requirements, HMAC, Digital signatures, Elgamal Digital Signature Scheme.
IP Security: IP Security overview, IP Security architecture, Authentication Header, Encapsulating security payload, Combining security associations, Internet Key Exchange.
.

Unit – IV-Transport-level Security

Web security considerations, Secure Socket Layer and Transport Layer Security, HTTPS, Secure Shell (SSH).
E-Mail Security: Pretty Good Privacy, S/MIME

Unit – V- Malicious Software

Types of malicious software, Types of attacks, Countermeasures, DDoS attacks.

Intruders: Intrusion detection and Password Management.

Firewalls: Characteristics, Access Policy, Types of Firewalls, Firewall Basing, Location and Configurations

Course Outcomes: At the end of the course, the student will be able to:

CO 1 : Understand basic Cryptographic concepts and Symmetric Key algorithms.
CO 2 : Understand and analyze Public-Key Cryptography along with Key Management and Distribution
CO 3 : Analyze and design Hash and MAC algorithm and IP security.
CO 4 : Understand the Transport level security and Web security.
CO 5 : Understand the Intruders, Viruses and Firewalls.

Textbooks:

1. Cryptography and Network Security - Principles and Practice, William Stallings, PearsonEducation, 7th Edition, 2017
2. Network Security Essentials: Applications and Standards: William Stallings, Pearson Education, 6th Edition, 2018
3. Cryptography and Network Security: Atul Kahate, McGraw-Hill, 3rd Edition, 2017

References:

1. Cryptography and Network Security, C K Shyamala, N Harini, Dr T R Padmanabhan, WileyIndia, 1st Edition, 2006
2. Cryptography and Network Security: Forouzan Mukhopadhyay, Mc Graw Hill, 3rd Edition,2015.
3. Information Security, Principles, and Practice: Mark Stamp, Wiley India, 2011.
4. Principles of Computer Security: WM. Arthur Conklin, Greg White, TMH,2016.
5. Introduction to Network Security: CENGAGE Learning, Neal Krawetz, 2007.
6. Network Security and Cryptography: CENGAGE Learning, Bernard Menezes, 2010.

Course Code: 22CS356

ADHOC AND SENSOR NETWORKS

(Professional Elective - II)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : Understand MANETs, routing approaches, and protocols in wireless networks.
2 : Analyze data transmission challenges, TCP protocol, and solutions in MANETs.
3 : Explore Wireless Sensors, WSN classification, MAC layer, and adaptability.
4 : Study security in Ad Hoc Networks, including key management.
5 : Familiarize with TinyOS, languages, and node-level simulation tools.

Unit I - Introduction to Ad hoc Wireless Networks

Introduction to Ad hoc Wireless Networks: Characteristics of MANETs, Applications of MANETs, Challenges.
Routing in MANETs: Topology-based versus Position-based approaches, Topology based routing protocols, Position based routing, Other Routing Protocols.

Unit II - Data Transmission in MANETs

Data Transmission in MANETs: The Broadcast Storm, Multicasting, Geo-casting
TCP over Ad hoc Networks: TCP Protocol overview, TCP and MANETs, Solutions for TCP over ad hoc networks

Unit III - Basics of Wireless Sensors and Applications

Basics of Wireless Sensors and Applications: Introduction, The Mica Mote, Sensing and Communication Range, Design issues, Energy consumption, Clustering of Sensors, Applications of WSNs.
Data Retrieval in Sensor Networks: Classification of WSNs, MAC layer, Routing layer, High level application layer support, Adapting to the inherent dynamic nature of WSNs.

Unit IV – Sensor Operating System

Operating System: TinyOS, Imperative Language: nesC Dataflow style language: T1nyGALS, Node- Level Simulators: ns-2 and its sensor network extension, TOSSIM.

Unit V – Security in MANET

Security: Security in Ad hoc Wireless Networks, Key Management, Secure Routing, Cooperation in MANETs, Intrusion Detection Systems.
Sensor Network Platforms and Tools: Sensor Network Hardware, Sensor Network Programming Challenges, Node-Level Software Platforms.

Course Outcomes: At the end of the course, the student should be able to:

CO 1: Analyze MANETs, choose appropriate routing protocols for wireless networks.
CO 2: Address data transmission challenges, optimize TCP in dynamic environments.
CO 3: Design efficient Wireless Sensor Networks, considering various application scenarios.
CO 4: Develop applications using Tiny OS, simulate networks for analysis.

CO 5: Implement secure Ad Hoc Networks, manage keys, detect intrusions.

Textbooks:

1. Ad Hoc and Sensor Networks: Theory and Applications, Carlos De Morais Cordeiro and Dharma Prakash Agrawal, World Scientific Publications /Cambridge University Press, March 2006.
2. Wireless Sensor Networks: An Information Processing Approach, Feng Zhao and Leonidas Guibas , Elsevier Science imprint, Morgan Kauffman Publishers, Reprint 2009.

References:

1. Ad hoc Wireless Networks: Architectures and Protocols, C. Siva Ram Murthy and B.S. Murthy, Pearson Education, 2004.
2. Wireless Sensor Networks: Principles and Practice, Fei Hu, Xiaojun Cao, Auerbach / CRC Press, Taylor & Francis Group, 2010.
3. Wireless Ad hoc Mobile Wireless Networks: Principles, Protocols and Applications, Subir Kumar Sarkar et al. Auerbach Publications, Taylor & Francis Group, 2008.
4. Ad hoc Networking, Charles E. Perkins, Pearson Education, 2001.
5. Wireless Ad hoc Networking, Shih-Liri Wu and Yu-Chee Tseng, Auerbach Publications, Taylor & Francis Group, 2007

Course Code: 22CS357

INTERNET OF THINGS

(Professional Elective - II)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To impart necessary and practical knowledge of components of IoT.
2 : To demonstrate the various python packages, interfacing Raspberry Pi was used for application development.
3 : To Learn and understand the IoT and M2M fundamental concepts, and system management.
4 : To Learn and Understand the IoT and cloud computing Architecture, service models, Web application framework and APIs.
5 : To Implement the Domain specific IoT applications in Realtime environment like Home, city, energy, antiagriculture, health, Industry and Lifestyle.

Unit I –Introduction to Internet of Things

Introduction - Definition and Characteristics of IoT. Physical Design of IoT – IoT Protocols. Logical Design of IoT -IoT Communication Models, IoT Communication APIs. IoT Enabling Technologies – Wireless Sensor Networks, Cloud Computing, Big data analytics, Communication protocols, Embedded Systems. IoT Levels and Deployment Templates.

Unit II - Python packages and IoT Interfaces

JSON, XML, HTTPLib, URLLib, SMTPLib, RPi.GPIO.
IoT Physical Devices and Endpoints - Introduction to Raspberry PI, Interfaces (serial, SPI, I2C). Programming Raspberry PI with Python - Controlling LED, interfacing an LED and Switch, Interfacing a Light Sensor with Raspberry Pi.

Unit III – IoT and M2M

Introduction, M2M, Difference between IoT and M2M, SDN and NFV for IoT. IoT System Management with NETCONF-YANG - SNMP, NETCONF, YANG, IoT System Management with NETCONF-YANG, NETOPEER.

Unit IV – IoT Physical Servers and Cloud Offerings

Cloud Computing – Definition, Characteristics, Architecture, Service Models and Deployment Models, Virtualization Concepts. Introduction to Cloud Storage models and communication APIs, WAMP-AutoBahn for IoT, Cloud for IoT, Python Web Application Framework, Designing a RESTful Web API.

Unit V – Domain Specific IoTs

Introduction - Home, City, Environment, Energy Systems, Retail, Logistics, Agriculture, Industry, Health, and Lifestyle.

Course Outcomes: At the end of the course, the student should be able to

CO 1: Understand the characteristics, protocols and communication models required for logical design of IoT.

CO 2: Realize the hardware platforms for implementing and interfacing the IoT based board with different peripheral devices and serial communication devices.

CO 3: Gain knowledge on protocol stacks for IoT and M2M networks and configurations.

CO 4: Integrate devices and develop an application that can communicate through IoT Cloud.

CO 5: Implement various case studies and applications in IoT design.

Textbooks:

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015.
2. Learning Internet of Things, Peter Waher, Packt publisher, 2015.
3. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014.

References:

1. Cloud Computing: Principles and Paradigms, Raj Kumar Buyya, James Broberg and Andrzej M Goscinski, Wiley, 2013
2. Getting started with sensors: Measure the world with Electronics, Arduino, and Raspberry, Kimmokarvinen and tero Karvenien, First Edition, Shroff/O"Reilly, 2014.
3. Getting started with Raspberry Pi, Richardson Matt, Shroff Publishers & Distributers Private Limited,2012

Course code: 22CS358

DISTRIBUTED SYSTEMS

(Professional Elective - II)

Instruction	:	3 Periods/week	Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To introduce the characteristics of distributed systems and inter-process communication.
2 : To enrich the knowledge of processes, threads, and operating system architecture.
3 : To acquire knowledge on peer-to-peer systems and their applications.
4 : To make the students understand the concepts of transactions and concurrency control in distributed systems.
5 : To equip students with principles and knowledge of transactions with replicated data.

Unit I - Characterization of Distributed Systems

Introduction, Examples of Distributed systems, Resource sharing and web, challenges, System models - Introduction, Architectural and Fundamental models, Networking and Internetworking, Inter-process Communication, Distributed objects and Remote Invocation-Introduction, Communication between distributed objects, RPC, Events and notifications, Case study-Java RMI.

Unit II - Operating System Support

Introduction, OS layer, Protection, Processes and Threads, Communication and Invocation, Operating system architecture, Distributed File Systems-Introduction, File Service architecture.

Unit III Peer to Peer Systems

Introduction, Napster and its legacy, Peer to Peer middleware, Routing overlays,Overlay case studies- Pastry, Tapestry, Application case studies - Squirrel, Ocean Store.

Time and Global States-Introduction, Clocks, events and Process states, Synchronizing physicalclocks, logical time and logical clocks, global states, distributed debugging.

Coordination and Agreement-Introduction, Distributed mutual exclusion, Elections, Multicast communication, consensus, and related problems.

Unit IV - Transactions and Concurrency Control

Introduction, Transactions, Nested Transactions, Locks, Optimistic concurrency control, Timestamp ordering. Distributed Transactions-Introduction, Flat and Nested Distributed Transactions, Atomic commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery.

Unit V - Replication

Introduction, System model and group communication, Fault-tolerant services,Transactions with replicated data.

Distributed shared memory, Design and Implementation issues, and Consistency models.

Course Outcomes: At the end of the course, the student should be able to

CO 1: Understand the characteristics of distributed systems and inter-process communication.
CO 2: Understand the concepts of processes, threads, and operating system architecture.
CO 3: Design a peer-to-peer systems and their applications.
CO 4: Understand the concepts of transactions and concurrency control in distributed systems.
CO 5: Use replicated data in transactions in distributed systems.

Textbooks:

1. Distributed Systems Concepts and Design, G Coulouris, J Dollimore and T Kindberg, FourthEdition, Pearson Education, 2008.
2. Distributed Systems, S.Ghosh, Chapman & Hall/CRC, Taylor & Francis Group, 2010.

References:

1. Distributed Systems – Principles and Paradigms, A.S. Tanenbaum and M.V. Steen, PearsonEducation, 2016.
2. Distributed Computing, Principles, Algorithms and Systems, Ajay D. Kshemakalyani andMukesh Singhal, Cambridge, RP 2010.

Course code: 22CS359

ARTIFICIAL NEURAL NETWORKS AND GRAPHICAL MODELS
(Professional Elective - II)

Instruction	:	3 Periods/week	Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To develop an understanding of the basic concepts of Artificial Intelligence.
2 : To gain the rationale behind the capabilities of different ANN architectures.
3 : To impart knowledge representation, non-linear boundaries.
4 : To derive value from Random Fields

Unit I – Multilayer Perceptron and Back Propagation

Introduction: A Neural Network, Human Brain, Models of a Neuron, Neural Networks viewed as Directed Graphs.

Learning Process: Error Correction Learning, Memory Based Learning, Hebbian Learning, Competitive, Boltzmann Learning, Credit Assignment Problem.

Single Layer Perceptron: Adaptive Filtering Problem, Least Mean Square Algorithm, Perceptron –Convergence.

Multi-Layer Perceptron: Back Propagation Algorithm, XOR Problem, Heuristics to improve the performance. Regularization, Weight Initialization techniques, Momentum based training.

Unit II – Unsupervised Learning and Sequence Learning

Associative Memory Networks: Training algorithms for pattern association, BAM and ART1.
Self-Organization Maps (SOM): Two Basic Feature Mapping Models, Self-Organization Map, SOM Algorithm, Properties of Feature Map, Computer Simulations, Learning Vector Quantization, Adaptive Pattern Classification.

Sequence Learning: Recurrent Network Paradigm, BPTT Algorithm, Hopfield Networks.

Unit III – Graphical Models

Bayesian Networks: Definition, Example: Polynomial regression, Generative models, Discrete variables, Linear-Gaussian models, Conditional Independence, Three example graphs, D-separation.

Markov Random Fields: Conditional independence properties, Factorization properties, Illustration: Image de-noising, Relation to directed graphs, Inference in Graphical Models, Inference on a chain, Trees, Factor graphs, The sum-product algorithm, The max-sum algorithm, Exact inference in general graphs, Loopy belief propagation, Learning the graph structure.

Unit IV – Mixture Models and EM

Mixtures of Gaussians, Maximum likelihood, EM for Gaussian mixtures, EM for Bayesian linear regression, EM Algorithm in General.

Approximate Inference: Variational Inference, Factorized distributions, Properties of factorized approximations, Example: The univariate Gaussian, Model comparison, Illustration: Variational **Mixture of Gaussians**: Variational distribution, Variational lower bound, Predictive density, Determining the number of components, Induced factorizations

Unit V – Sampling Methods
Basic Sampling Algorithms, standard distributions, Rejection sampling, Adaptive rejection sampling, Sampling-importance-resampling, Sampling and the EM algorithm

Markov Chain Monte Carlo: Markov chains, Metropolis-Hasting's algorithm, Gibbs Sampling, Slice Sampling, Hybrid Monte Carlo Algorithm, Dynamical systems Hybrid Monte Carlo, Estimating the Partition Function.

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

CO 1 : Implement Backpropagation algorithm and solve non-liner classification problem
CO 2 : Solve semi-supervised learning-based SOM
CO 3 : Address real-world image restoration problem using Belief Propagation algorithm
CO 4 : Appreciate Variational principle based MLE approximation
CO 5 : Gain proficiency in different sampling methods

Textbooks:

1. Neural Networks a Comprehensive Foundations, Simon Haykin, 2nd Edition, Pearson, 2009.
2. Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer, 2006.

References:

1. Artificial Intelligence: Structures and Strategies for complex problem solving, G. Luger, 4thEdition, Pearson Education, 2002.
2. Artificial Intelligence: A new Synthesis, J. Nilsson, Elsevier Publishers, 1998.
3. Probabilistic Graphical Models, Koller, and N. Friedman, MIT Press, 2009.

Course Code: 22CS381

IoT AND CLOUD COMPUTING LAB

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

List of Experiments: IOT

Task 1:

Installing Node-RED on a Raspberry Pi through Remote Login.

Task 2:

Create a simple Node-RED flow that takes input from an inject node and displays output in a debug node. Add a function node to modify the payload in the flow.

Task 3:

Integrate an MQTT node into a Node-RED flow and subscribe to a topic.

Task 4:

Implement an HTTP request node to interact with an external API. (Twitter, WhatsApp)

Task 5:

Build a basic IoT dashboard using the Node-RED dashboard nodes.

Task 6:

Include widgets for displaying sensor data and control buttons.

Task 7:

Configure security settings for Node-RED, including user authentication.

Task 8:

Implement SSL/TLS for secure communication in a Node-RED instance.

List of Experiments: Cloud Computing

Task 1:

Create and Manage Cloud Resources
a) Tour of Google Cloud
b) Creating a Virtual Machine
c) Getting Started with Cloud Shell and g cloud

Task 2:

Kubernetes Engine:
a) Set Up Network and HTTP Load Balancers
b) Create and Manage Cloud Resources: Challenge Lab

Task 3:

Perform Foundational Infrastructure Tasks
a) Cloud Storage: Qwik Start - Cloud Console
b) IAM in AWS and setup
c) Cloud Functions
d) Cloud networking

Task 4:

Set Up and Configure a Cloud Environment
a) Cloud IAM: Qwik Start
b) Introduction to SQL for Big Query and Cloud SQL
c) Database Tasks in Cloud
d) Cloud Monitoring: Qwik Start
e) Managing Deployments Using Kubernetes Engine
f) Set Up and Configure a Cloud Environment in Google Cloud: Challenge Lab

Task 5:

1. Introduction to Amazon EC2
2. Introduction to Amazon Simple Storage Service (S3)

Task 6:

1. Introduction to Amazon Relational Database Service (RDS) - SQL Server)
2. AWS Identity and Access Management (IAM) Task

Task 7:

1. Management of Amazon Elastic Container Service
2. Hosting a static website and Deploying a web application on AWS

Task 8:

a. Continuously Querying top 10 songs in the song-list or chart with Kubernetes
b. Key Parameter Indicators visualization for Airline services using Kubernetes

Task 9:

a. Managing resources using Terraform
b. creating and running containers.

Course outcomes: At the end of the course, the student will be able to

CO 1 : Implement framing, error detection techniques of data link layer.
CO 2 : Design and implement different routing algorithms in network layer.
CO 3 : Design and Implement security algorithms in application layer.
CO 4 : Understand the fundamentals of creating and managing cloud
CO 5 : Identify the significance of Kubernetes engine and process of tasks execution

References:

1. Computer Networks, Andrew S Tanenbaum, 5th Edition, Pearson Education, 2011.
2. Data Communications and Networking, Behrouz A. Forouzan, 5th Edition, TMH, 2009.
3. Network Security Essentials: Applications and Standards, William Stallings, 4th Edition Pearson Education, 2011
4. Cryptography and Network Security, B.A. Forouzan and D. Mukhopadhyay, TMH, 2nd Edition, TMH. 2010.

Course Code: 22CS382

FULL STACK DEVELOPMENT LAB

Instruction	: 3 Periods / week	Continuous Internal Evaluation	: 40 Marks Semester
Tutorial	: –	End Examination	: 60 Marks
Credits	: 1.5	Semester End Exam Duration	: 3 Hours

Course Objectives:

1 : To enhance code readability, maintainability, and developer productivity by adopting ES6 features and exploring version control system.

2 : To gain proficiency and expertise in utilizing NPM (Node Package Manager) and developing RESTful APIs using Express framework.

3 : To explore how NoSQL databases, particularly MongoDB, can be leveraged to design scalable and flexible data solutions.

4 : To Gain proficiency in front-end development using Angular.

5 : To provide the knowledge of building modular and reusable components and create

Task 1: Managing the versions of the product using Version Control System (GIT)

1. Create a git repository local add and commit a simple web application consisting of 5 pages.
2. Create a remote repository in github.com. Push the above local repository to the github.com. Explore the push, pull and fetch options with remote repository.
3. Clone a remote repository into local directory, modify the implementation and push the updated changes back to remote repository.
4. Create branch and manage the work distributions. Merge all the branches and commit the changes.
5. Publish the application using GitHub pages.

Task 2: Implementing the advanced features of JavaScript

1. Working with Prototypal Inheritance and Classes.
2. Working with Object and Array Destructuring.
3. Working with Modules.
4. Working with Function Generators and Symbols.
5. Working with Closure.

Task 3: Asynchronous Execution of JavaScript

1. Working with higher order function in JavaScript.
2. Using Callback and creating a Callback Hell situation to understand the drawbacks.
3. Working with XHR: response.
4. Dealing with the Callback Hell situation using Promise. Exploring the different ways of creating and using promise in executing the asynchronous task.
5. Dealing with Promise chaining and async / await.

Task 4: Fetching server state using JavaScript (fetch)

1. Use fetch function to access remote data using the given api and display the data in the form of a table.
2. Use fetch function to read the weather details from openweathermap.org and display the details like city, min-temp, max-temp, humidity on the webpage for a given city.
3. From the same website read the weather forecast details for a given city and display the details like date – temperature in a table.
4. Plot a bar chart for the above implementation using date and temperature along X and Y axis respectively. Use ChartJS library.

Task 5: Node JS

1. Create custom / local modules and export them using various module patterns.
2. Explore the functionality of os, path, util and events modules.
3. Use the fs module for creating directories and files of different formats.
4. Write script to read and write the streaming data using readable and writable streams.

Task 6: Working with http

1. Create a http server listening request at port 3000. Process the request to provide different type of resources as response. (HTML, TEXT, JSON, etc.).
2. Create express server listening request at port 3000. Add different endpoints to provide access to the resources.

Task 7: Working with Express

1. Create a custom API for Users data and add different endpoints in express server to perform CRUD operations on the API. Test the endpoints using POSTMAN.
2. Use EJS view-engine to display the dynamic response. Display the data read from REST API in the form of a table in EJS.

Task 8: Working with model (MongoDB & Firebase)

1. Create a database in Mongo DB. Create Users collection and documents to the User collection. Perform all DB operations (CREATE, READ, UPDATE and DELETE) on the User collection.
2. Create a real time database in firebase for the student management system and explore the features of Firebase Real Time Database. Perform CRUD operations on the Real Time Database.

Task 9: Working with Express & Mongo DB

1. Create express server that has endpoints connecting to Users collection present in Mongo DB database using mongoose library and perform CRUD operation on that.

Task 10: Authentication and Authorization

1. Create express server that has authorized endpoint using JWT (JSON Web Token) library.
2. Create express server that connects to Mongo DB database to authenticate the user and generate the authorized token to access the protected endpoints.

Task 11: MEAN Stack Development

1. Create a user profile management system where users can update their profiles, including details like name, email, phone. Use Angular's data binding to ensure that changes made by users are instantly reflected in the UI.
2. Develop an angular application that interacts with the backend API and executes CRUD operations on it.

Task 12: Angular Routing

1. Develop angular application consisting of App, Home, About, Contact, Profile, Login and Register Components. Add a Navigation bar and navigate to the respective component using angular routing.
2. Develop a Single Page Application in Angular for User Management System that interacts with the backend database created in Task 8. Use Services and Http-Client to access the express endpoints of Task 9.

Task 13: MERN Stack Development

1. Create react functional and class components. Implement the lifecycle methods of react component.
2. Develop react application with App, Home, About and Contact components. Implement the use of react props and state in these components.

Task 14: React Routing

1. Develop a react application that demonstrates the routing feature to navigate across different components of react and pass the data in between the components.
2. Develop a SPA in react for User Management System that interacts with the backend API using Axios and perform CRUD operations on that.

Task 15: Single Page Application in Angular / React

A TODO application serves as a simple yet powerful tool to help individuals and teams organize their tasks, manage priorities, and enhance productivity. TODO applications provide a structured and efficient way for individuals and teams to manage tasks, prioritize work, and achieve their goals. Develop a Single Page TODO Application in Angular React to manage the daily tasks with the following features:

i. **Task Creation:** Allow users to create new tasks with a title, description, due date, and priority level. Provide a straightforward interface for entering task details.
ii. **Task Listing:** Display a list of all tasks with essential details. Tasks can be organized based on different criteria such as due date, priority, or completion status.
iii. **Task Editing and Updating:** Enable users to edit task details, including the ability to modify the title, description, due date, and priority. Changes should be reflected in real-time.
iv. **Task Deletion:** Provide the option to delete tasks that are no longer relevant or completed. Include a confirmation prompt to prevent accidental deletions.
v. **Task Completion:** Allow users to mark tasks as completed or mark them with a specific status. Completed tasks may be moved to a separate section or visually differentiated.
vi. **User Authentication and Authorization:** Implement user accounts with authentication to ensure data privacy. Differentiate between users and provide appropriate authorization levels.
vii. **Data Persistence:** Ensure that tasks are persistently stored, so users can access their TODO lists even after closing and reopening the application.

(NOTE: Task 1 – Task 14 are mandatory to complete in the labs)

Course Outcomes: At the end of the course, the student will be,

CO 1 : Proficient in leveraging ES6 capabilities to streamline development tasks, improving code efficiency, and solving programming challenges with a modern JavaScript mindset and managing various versions of the product using git
CO 2 : Able to develop the skills to construct scalable and maintainable web applications by harnessing the asynchronous, event-driven nature of Node.js and the modular architecture provided by Express
CO 3 : Designing and implementing efficient NoSQL database solutions, understanding when and how to use document-oriented databases like MongoDB to address specific application requirements.
CO 4 : Developing the dynamic and single-page web applications using the MEAN stack and understand how to manage client-side routing, and UI components to deliver a seamless user experience
CO 5 : Excelling in both server-side and client-side aspects of web application development using MERN stack

References:

1. MEAN Web Development, Amos Q. Haviv, Second Edition, Packt Publications, November 2016.
2. "Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node", Vasan Subramanian, 2nd Edition, APress, 2017.
3. Learning Node: Moving to the Server-Side, Shelly Powers, 2nd Edition, O'REILLY, 2016.
4. "Getting MEAN with Mongo, Express, Angular, and Node", Simon D. Holmes and Clive Harber, Second Edition, Manning Publications, 2019.
5. "Node.js, MongoDB and Angular Web Development", Brad Dayley, 2nd Edition, Addison-Wesley Professional, 2017.
6. https://angular-2-training-book.rangle.io.
7. https://www.atlassian.com/git
8. https://www.typescriptlang.org/docs/handbook/basic-types.html
9. https://firebase.google.com

Course Code: 22HS331/381

ADVANCED ENGLISH COMMUNICATION AND SOFT SKILLS LAB

(B.Tech III Year - Common to all Branches)

Instruction	:	2 Periods/week	Continuous Internal Evaluation	:	40 marks
Tutorial	:	-	Semester End Examination	:	60 marks
Credits	:	1	Semester End Exam Duration	:	2 hrs

Course Objectives:

1 : To equip students with the requisite communication skills for real-time environment.
2 : To prepare students for persuasive conversations in the professional sphere.
3 : To integrate time management and decision-making skills for better performance.
4 : To modify communication to suit diverse cultures.
5 : To sensitize students to handle emotions at workplace.

Unit I - Soft Skills & Interpersonal Communication

a) Effective Communication: types of communication-verbal & non-verbal, importance of communication, 7 Cs of communication, barriers to effective communication, communication according context, strategies for improving communication.
b) Intrapersonal & Interpersonal Skills: definition, how to start a conversation, self-introduction, self-concept, signs of high & low self-esteem, self-exploration, SWOT analysis.

Unit II - Oral Communication

a) Group Discussion- significance of GD, types of GD, opening strategies, roles of participants, evaluation parameters, dos and don'ts, types of topics, features of GD, mock GD.
b) Negotiation Skills- definition, process, outcome of negotiation, skills required, strategies of communication for negotiation.
c) Book-discussion- genres of book, importance of book-discussion, purpose of book-discussion, process of book-discussion, critical and analytical skills.

Unit III - Employability Skills

a) Team Dynamics- difference between team and group, types of teams, concepts related to team building, process of team building, roles of team player and qualities.
b) Time Management- concept of time-management, time logs, timewasters, time quadrant, priority list, tips of time management.
c) Decision-making & Problem Solving- strategies of decision-making, techniques of decision-making, problem-solving process.

Unit IV - Cross-cultural Communication

a) Pluralism: Introduction to Cross-cultural Management, Communicating across Cultures, high-low context culture, negotiating across Cultures, Motivation and Leadership across Cultures, Managing Global Teams, Global Manager.
b) Diversity & adaptability- race, gender, class, caste, religion.

Unit V - Emotional Intelligence

Concept of Emotional Intelligence: Intrapersonal Awareness, Intrapersonal Management, Conflict Management and Leadership, Anger Management.

Course Outcomes: At the end of the course, the students will

CO 1 : Gain proficiency in communication skills.
CO 2 : Emerge as rational speakers.
CO 3 : Efficiently Manage their professional career.
CO 4 : Acclimatize to diverse cultures.
CO 5 : Be empowered with skills required for self-management.

References:

1. Communication and Soft Skill development, Ashwini Deshpande, 1st Edition, Career Publications, 2017.
2. Effective Communication and Soft Skills, Nitin Bhatnagar and Mamta, Pearson, 2011.
3. Soft Skills & Life Skills, The Dynamics of Success, Nishitesh and Dr. Bhasker Reddi, BSC Publishers & Distributors, 2012.
4. Guide to Cross-Cultural Communications, Reynolds, Valentine and Verma, Pearson, 2010.
5. Emotional Intelligence: A Comprehensive Self Help Guide to Developing EQ, Managing Anger, and Improving your Relationships, Christopher Rance, Ingram Publishing 2019.
6. Unearthing your Emotional Intelligence, Deepa R, Notion Press, Paperback, 2020.

Course Code: 22HS352/302/253

INTELLECTUAL PROPERTY RIGHTS

(Common to B.Tech III Year all branches and II year CSBS branch)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	100 Marks
Tutorial	:	-	Semester End Examination	:	-
Credits	:	-	Semester End Exam Duration	:	-

Course Objectives:

1 : To impart basic concepts in IPR.
2 : To understand the various aspects of Trademarks.
3 : To create awareness on Law of Patents and Copyrights.
4 : To highlight relevance of Trade Secrets in any Trade/Business.
5 : To get elementary knowledge of International IPRs and New developments in IPR.

Unit I - Introduction to Intellectual Property

Introduction, types of intellectual property, international organizations, agencies and treaties, importance of intellectual property rights.

Unit II - Trade Marks

Purpose and function of trademarks, acquisition of trademark rights, protectable matter, selecting, and evaluating trademark, trademark registration processes.

Unit III - Law of Copyrights

Fundamental of copy right law, originality of material, rights of reproduction, rights to perform the work publicly, copy right ownership issues, copyright registration, notice of copy right, international copy right law.

Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer.

Unit IV- Trade Secrets

Trade secrete law, determination of trade secrete status, liability for misappropriations of trade secrets, protection for submission, trade secrete litigation.

Unfair competition: Misappropriation right of publicity, false advertising.

Unit V- New Development in Intellectual Property

New developments in trademark law; copyright law, patent law, intellectual property audits. International overview on intellectual property, international – trademark law, copyright law, international patent law, and international development in trade secrets law.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Understand concepts of intellectual property rights.
CO 2 : Evaluate trademark.
CO 3 : File for a patent.
CO 4 : Analyze the fairness in a competition.
CO 5 : Understand laws related to intellectual property rights.

Textbooks:

1. Intellectual Property Right, Deborah. E. Bouchoux, 4th Edition, Cengage Learning, 2012.
2. Intellectual Property Right – Unleashing the Knowledge Economy, Prabuddha Ganguli, 1st Edition, Tata McGraw-Hill Publishing Company Ltd, 2017.

References:

1. Intellectual Property Patents, Trademarks and Copyrights, Richard Stim, 2nd Edition, Cengage Learning, 2012.
2. Intellectual Property Rights under WTO, T. Ramappa, S. Chand, 2008.

Course Code: 22CS401

LINUX PROGRAMMING

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1: To develop the skills necessary for systems programming.

2: To model asynchronous event handling.

3: To establish efficient communication between two asynchronous processes.

4: To design various client & server communication models

Unit I – File System and Directory Structure

Files: Files concept, File System Structure, I-nodes, File Attributes, File types, Library functions, the standard I/O and formatted I/O in C, stream errors, kernel support for files, System calls, file descriptors, low level file access- File structure related system calls(File APIs), file and record locking, file and directory management – Directory file APIs, Symbolic links & hard links.

Unit II - Process and Signals

Process: Process concept, Kernel support for process, process attributes, process control - process creation, waiting for a process, process termination, zombie process, orphan process APIs.

Signals: Introduction to signals, Signal generation and handling, Kernel support for signal, Signal function, unreliable signal, reliable signal, kill, raise, alarm, pause, abort, sleep functions.

Inter Process Communication: Introduction to IPC, Pipes, FIFOs.

Unit III - IPC, Message Queues, Semaphores, Shared Memory and Socket Programming

Message Queues – Kernel support for messages, Unix system V APIs for messages, client/server example.

Semaphores: Kernel support for semaphores, Unix system V APIs for semaphores.
Shared Memory: Kernel support for shared memory, Unix system V APIs for shared memory, semaphore and shared memory example.

Sockets: Introduction to Sockets, Socket Addresses Structures, Byte ordering and manipulation functions, Socket related system calls for TCP sockets- Socket, connect, bind, listen, fork, exec, and close, Implementation of concurrent server, TCP Client Server programs, Normal startup, terminate and signal handling server process termination. TCP Client-Server Program and UNIX domain Sockets.

Unit IV - Socket Programming, Multithreaded Programming

Socket Options, Server and service models, Super Server, Elementary UDP Sockets: Introduction UDP Echo server function, lost datagram, summary of UDP example, Lack of flow control with UDP.

Multithreaded Programming: Differences between threads and processes, Thread structure and uses. Threads and Lightweight Processes, POSIX Thread APIs, Creating Threads, Thread Attributes, Thread Synchronization with Condition Variables and with Mutexes, Example programs.

Unit V –Advanced I/O

I/O Multiplexing and Socket options: I/O Models, Select Function, poll function.
Record Locking, Readn and Writen functions, Scatter and Gather IO, Memory Mapped IO. Asynchronous IO and Async Options. Remote login overview and RPC Transparency issues. FTP server configuration.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Make use of well-defined Korn shell utilities and develop menu driven Text processing Application.
CO 2 : Appreciate process abstraction and asynchronous event handling using signals.
CO 3 : Implement IPC Mechanisms, Messages Queues and synchronize the access patterns as a shared memory.
CO 4 : Design concurrent server programs based on various design alternatives.
CO 5 : Implement I/O multiplexing mechanisms.

Textbooks:

1. Advanced Programming in the UNIX Environment, W Richard Stevens and Stephen A Rago, 3rd Edition, Addison Wesley / Pearson Education Inc., 2013.
2. Unix System Programming using C++, T.Chan, PHI, 1999

References:

1. Unix Network Programming, W R Stevens, PHI, 2003.
2. Unix Internals: The New Frontiers, Uresh Vahalia, Pearson Education, 1995
3. Unix for Programmers and Users, Graham Glass and King Ables, 3rd Edition, Pearson Education, 2003.

Course Code: 22HS301/351/401

BUSINESS ECONOMICS AND FINANCIAL ANALYSIS

(Common to B.Tech III Year all branches (except CSE) and IV Year CSE branches)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To obtain the knowledge about types of Business Structures and features.
2 : To learn various principles of Managerial Economics and to make them effective business decision makers.
3 : To make the students understand functional areas and potential problems in economics for efficient utilization of resources.
4 : To have an overview on market structures and Competition.
5 : To gain knowledge on important elements.

Unit I – Introduction to Business and Economics

Business: Structure of Business Firm, Theory of Firm, Types of Business Entities, Limited Liability Companies, Sources of Capital for a Company, Non-Conventional Sources of Finance. Economics: Significance of Economics, Micro and Macro Economic Concepts, Concepts and Importance of National Income, Inflation, Money Supply and Inflation, Business Cycle, Features and Phases of Business Cycle. Nature and Scope of Business Economics, Role of Business Economist, Multidisciplinary nature of Business Economics.

Unit II - Demand and Supply Analysis

Elasticity of Demand: Elasticity, Types of Elasticity, Law of Demand, Measurement and Significance of Elasticity of Demand, Factors affecting Elasticity of Demand, Elasticity of Demand in decision making.

Demand Forecasting: Characteristics of Good Demand Forecasting, Steps in Demand Forecasting, Methods of Demand Forecasting.
Supply Analysis: Determinants of Supply, Supply Function and Law of Supply.

Unit III - Production, Cost, Market Structures & Pricing

Production Analysis: Factors of Production, Production Function, Production Function with one variable input, two variable inputs, Returns to Scale.

Cost analysis: Types of Costs, Short run and Long run Cost Functions.

Market Structures: Nature of Competition, Features of Perfect competition, Monopoly, Oligopoly, Mono polistic Competition.

Pricing: Types of Pricing, Product Life Cycle based Pricing, Break Even Analysis (Simple Problems).

Unit IV – Financial Accounting

Accounting concepts and Conventions, Accounting Equation, Double-Entry system of Accounting, Rules for maintaining Books of Accounts, Journal, Posting to Ledger, Preparation of Trial Balance, Elements of Financial Statements, Preparation of Final Accounts (Simple Problems).

Unit V – Financial Ratios Analysis

Concept of Ratio Analysis, Importance and Types of Ratios, Liquidity Ratios, Turnover Ratios, Profitability Ratios, Proprietary Ratios, Solvency, Leverage Ratios –Analysis and Interpretation (simple problems).

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Understand the concepts of micro and macro economics.
CO 2 : Analyze demand and forecast demand.
CO 3 : Evaluate markets and formulate competitive strategies.
CO 4 : Prepare financial statements.
CO 5 : Evaluate the financial strengths and weaknesses of a business by using ratio analysis.

Textbooks:

1. Business Economics - Theory and Applications, D. D. Chaturvedi, S. L. Gupta, International Book House Pvt. Ltd. 2013.
2. Financial Accounting, Dhanesh K Khatri, Tata McGraw-Hill, 2011.
3. Managerial Economics, Geethika Ghosh, Piyali Gosh, Purba Roy Choudhury, 2nd Edition, Tata McGraw-Hill Education Pvt. Ltd. 2012.

References:

1. Financial Accounting for Management, Paresh Shah, 2nd Edition, Oxford Press, 2015.
2. Financial Accounting, S. N. Maheshwari, Sunil K Maheshwari, Sharad K Maheshwari, 5th Edition, Vikas Publications, 2013.

Course Code: 22IT403

QUANTUM COMPUTING

(Professional Elective-III)
(Common to IT, CSE, CSE(AI&ML, DS and CS))

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Prerequisites: A course on "Data Structures"

Course Objectives:

1: To Understand the basic Concepts of Linear Algebra related to Quantum Computing.
2: To Master the basics of physics-oriented phenomenon related to Quantum Computing.
3: To get familiar with various Quantum Architecture and Quantum Algebra.

Unit I - Introduction to Essential Linear Algebra

Some Basic Algebra, Matrix Math, Vectors and Vector Spaces, Set Theory. Complex Numbers: Definition of Complex Numbers, Algebra of Complex Numbers, Complex Numbers Graphically, Vector Representations of Complex Numbers, Pauli Matrice, Transcendental Numbers. Database System, The Journey to Quantum Computing.

Unit II - Basic Physics for Quantum Computing

Quantum Physics Essentials, Basic Atomic Structure, Hilbert Spaces, Uncertainty, Quantum States, Entanglement.

Basic Quantum Theory: Further with Quantum Mechanics, Quantum Decoherence, Quantum Electrodynamics, Quantum Chromodynamics, Feynman Diagram Quantum Entanglement and QKD, Quantum Entanglement, Interpretation, QKE.

Unit - III Quantum Architecture

Further with Qubits, Quantum Gates, More with Gates, Quantum Circuits, The D-Wave Quantum Architecture. Quantum Hardware: Qubits, How Many Qubits Are Needed? Addressing Decoherence, Topological Quantum Computing, Quantum Essentials. Introduction to Query Language

Unit - IV Quantum Algorithms

Introduction, Deutsch's Algorithm, Deutsch-Jozsa Algorithm, Bernstein-Vazirani Algorithm, Simon's Algorithm, Shor's Algorithm, Grover's Algorithm

Unit - V Current Asymmetric Algorithms

RSA, Diffie-Hellman, Elliptic Curve. The Impact of Quantum Computing on Cryptography: Asymmetric Cryptography, Specific Algorithms, Specific Applications.

Course Outcomes: At the end of the course, the student will be able to

CO 1 : Understand basics of quantum computing
CO 2 : Understand physical implementation of Qubit
CO 3 : Understand Quantum algorithms and their implementation

CO 4 : Understand the Impact of Quantum Algorithms and its significance
CO 5 : Realize the importance of current Asymmetric Algorithms on Quantum computing

Textbooks:

1. Quantum Computation and Quantum Information, Nielsen M. A., Cambridge University Press
2. Quantum Computing Fundamentals, Dr. Chuck Easttom, Pearson.

References:

1. Quantum Computing for Computer Scientists, Noson S. Yanofsky and Mirco A. Mannucci, 1st Edition, Cambridge University Press, 2008.
2. Principles of Quantum Computation and Information, Benenti G., Casati G. and Strini G., Vol. Basic Tools and Special Topics, World Scientific Publishing Co. Pte. Ltd, 2007.
3. An Introduction to Quantum Computing Algorithms, Pittenger A. O., Springer-Verlag New York Inc, 2012.

Course Code: 22IT404

CLOUD SECURITY

(Professional Elective-III)
(Common to IT, CSE, CSE(CS))

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Prerequisites: Fundamentals of cyber security

Course Objectives:

1: Understand the fundamentals of cloud computing architectures based on current standards, SLA, and cloud security models to integrate.
2: Understand the industry security standards, regulatory mandates, audit policies and compliance requirements for Cloud based infrastructures.
3: Approaches to designing cloud services that meets essential Cloud infrastructure security and management.
4: Identify the known threats, risks, vulnerabilities, and privacy issues associated with Cloud based IT services and IAM services.

Unit -I Introduction to cloud computing and cloud security

Cloud Computing Architectural Framework: Cloud Computing Evolution, Essential Characteristics of Cloud Computing, Cloud Deployment Architecture, Cloud Deployment models, Cloud Service Models, SLA.

Introduction to Cloud Security: Introduction, Cloud Security Concepts, Cloud security Standards, CSA Cloud Reference Model, NIST Cloud Reference model.

Unit - II Compliance, audit and privacy issues

Compliance and Audit: Cloud customer responsibilities, Compliance and Audit Security Recommendations. Portability and Interoperability: Changing providers reasons, Changing providers expectations, Internal Policy compliance, governance risk compliance(GRC)

Cloud Security and Privacy Issues: Introduction, Cloud Security Goals/Concepts, Cloud Security Issues, Security Requirements for Privacy, Privacy Issues in Cloud.

Unit - III Cloud infrastructure security

Identity and Access Management (IAM) in cloud environments, Virtual Private Cloud (VPC) and network security, Secure provisioning and configuration management in the cloud. The Network Level, the Host Level, the Application Level. Security Management In Cloud, Availability management for SAAS, PAAS, IAAS.

Unit - IV Threat Model and Cloud Attacks

Introduction, Threat Model- Type of attack entities, Attack surfaces with attack scenarios, A Taxonomy of Attacks, Attack Tools-Network-level attack tools, VM-level attack tools, VMM attack tools, Security Tools, VMM security tools

Unit - V Identity and Access Management

The role of IAM in cloud security, Key concepts: authentication, authorization, and auditing, oAuth and SSO, User identity and access lifecycle management, Configuring IAM in cloud platforms (e.g., AWS, Azure). **Authorization and Role-Based Access Control (RBAC)** Role-based access control principles, Creating and managing roles in cloud platforms, Least privilege principle.

Identity Federation- Understanding identity federation, Cross-domain and cross-cloud identity integration, Use cases and challenges, Virtualization, Virtualization Security Recommendations.

Course Outcomes: At the end of the course, the student should be able to

CO1 : Demonstrate the growth of Cloud computing and cloud security, architecture, and different modules of implementation
CO2 : Evaluate the different types of cloud Compliance and Audit, Security and Privacy Issues.
CO3 : Access the security implementation flow, actions and responsibilities of stake holders using IAM
CO4 : Able to analyze the various threats and Attack tools details
CO5 : Able to implement based on roles and groups policy created and choose the type of virtualization to be used

Textbooks:

1. Cloud Security and Privacy, An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Kumaraswamy, Shahed Latif, Oreilly Media, 2009.
2. Cloud Security Attacks, Techniques, Tools, and Challenges, Preeti Mishra, Emmanuel S Pilli, Jaipur R C Joshi Graphic Era, 1st Edition, CRC press, 2022.

References:

1. Securing the Cloud, Cloud Computer Security Techniques and Tactics, Vic (J.R.), Winkler, Syngress, 2011.
2. Online documentation and tutorials from cloud service providers (e.g., AWS, Azure, GCP)

Course Code: 22CY405

BLOCKCHAIN TECHNOLOGIES

(Professional Elective-III)
(Common to CSE (CS), CSE & IT)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1: Conceptual understanding of the function of Blockchains as a method of securing distributed ledgers, how consensus on their contents is achieved.
2: To understand blockchain operations as distributed data structures and decision-making systems.
3: To evaluate "smart contract" capabilities and platforms, and examines their future directions, opportunities, risks, and challenges.

Unit - I - Introduction to Blockchain

The growth of Blockchain technology, Distributed systems, History of Blockchain and Bitcoin, Types of Blockchain, Consensus, CAP theorem and Blockchain**.**

Decentralization: Decentralization using block chain, Methods of decentralization, Routes to decentralization, Blockchain and full ecosystem decentralization, Smart contracts, Decentralized Organizations, Platforms for decentralization.

Unit - II Bitcoins

Introducing Bitcoin, Digital keys and addresses, Transactions**,** The structure of a block, Mining.

Bitcoin Network and Payments: The Bitcoin network, Wallets, Bitcoin payments, Innovation in Bitcoin, Bitcoin Clients, and APIs, Bitcoin installation, Alternative Coins, Bitcoin limitations.

Unit - III Smart contracts

History, Definition, Ricardian contracts, Introduction to Ethereum, Components of the Ethereum ecosystem, Further Ethereum, Programming languages.

Unit - IV Ethereum Development Environment

Test networks, Development Tools and Frameworks, Compilers, Solidity compiler (solc), Integrated Development Environments (IDEs).

Solidity language: Layout of a Solidity, Data Types: Reference types, Value types; Literals, Enums, Function types, Global variables, Control structures.

Unit - V Hyper ledger and beyond cryptocurrency

Projects under Hyperledger, Hyperledger as a protocol, the reference architecture, Requirements and design goals of Hyperledger Fabric, Hyperledger Fabric, Membership services, Blockchain services, Consensus services, Distributed ledger.

Beyond Cryptocurrency: Applications of blockchain in cyber security, integrity of information, E-Governance and other contract enforcement mechanisms, Limitations of blockchain as a technology, and myths vs. reality of blockchain technology.

Course Outcomes: At the end of the course, the student will be able to

CO 1 : Understand the structure of a blockchain and how it is better than a simple distributed database
CO 2 : Evaluate the blockchain based structure along with its potentials and its limitations
CO 3 : Understand smartǁ contract and what are its legal implications and what it can and cannot do
CO 4 : Attain awareness of the new challenges around blockchains and smart contracts
CO 5 : Understand the differences between the different blockchain structures and their specific uses

Textbooks:

1. Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, Imran Bashir, 3rd Edition, Packt Publishing, 2020.
2. Mastering Blockchain, Imran Bashir, Second Edition, Packt Publishing, March 2018.
3. Mastering Bitcoin Programming the Open Blockchain, Andreas M. Antonopoulos, 2nd Edition, O'Reilly Media, Inc., June 2017.

References:

1. Hyperledger Fabric - https://www.hyperledger.org/projects/fabric.
2. Hyperledger Tutorials - https://www.hyperledger.org/use/tutorials
3. Ethereum Development Resources - https://ethereum.org/en/developers

Course Code: 22DT406

NOSQL DATABASES

(Professional Elective – III)
(Common to CSE(DS), CSE & IT)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1: Define, compare and use the four types of NoSQL Databases (Document-oriented, key-value pairs, Column-oriented and Graph).
2: Demonstrate an understanding of the detailed architecture, define objects, load data, query data and performance-tune Column-oriented NoSQL databases.
3: Explain the detailed architecture, define objects, load data, query data and performance-tune Document-oriented NoSQL databases.

Unit - I Introduction to NoSQL

Why NoSQL? The Value of Relational Databases, Getting at Persistent Data, Concurrency, Integration, A (Mostly) Standard Model, Impedance Mismatch, Application and Integration Databases, Attack of the Clusters, The Emergence of NoSQL, Aggregate Data Models; Aggregates, Example of Relations and Aggregates, Consequences of Aggregate Orientation, Key-Value and Document Data Models, Column-Family Stores, Summarizing Aggregate-Oriented Databases. More Details on Data Models; Relationships, Graph Databases, Schemaless Databases, Materialized Views, Modeling for Data Access.

Unit - II Distribution Models

Single Server, Sharding, Master-Slave Replication, Peer-to-Peer Replication, Combining Sharding and Replication. Consistency, Update Consistency, Read Consistency, Relaxing Consistency, The CAP Theorem, Relaxing Durability, Quorums. Version Stamps, Business and System Transactions, Version Stamps on Multiple Nodes.

Unit - III Map-Reduce

Basic Map-Reduce, Partitioning and Combining, Composing Map-Reduce Calculations, A Two Stage Map-Reduce Example, Incremental Map-Reduce Key-Value Databases, What Is a Key-Value Store, Key-Value Store Features, Consistency, Transactions, Query Features, Structure of Data, Scaling, Suitable Use Cases, Storing Session Information, User Profiles, Preference, Shopping Cart Data, When Not to Use, Relationships among Data, Multioperation Transactions, Query by Data, Operations by Sets.

Unit - IV Document Databases

What is a Document Database? Features, Consistency, Transactions, Availability, Query Features, Scaling, Suitable Use Cases, Event Logging, Content Management Systems, Blogging Platforms, Web Analytics or Real-Time Analytics, E-Commerce Applications, When Not to Use, Complex Transactions Spanning Different Operations, Queries against Varying Aggregate Structure.

Unit - V Graph Databases

What is a Graph Database? Features, Consistency, Transactions, Availability, Query Features, Scaling, Suitable Use Cases, Connected Data, Routing, Dispatch, and Location-Based Services, Recommendation Engines, When Not to Use.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Acquire the knowledge of NoSQL databases
CO 2 : Articulate the distributed models
CO 3 : Construct the database using MapReduce model
CO 4 : Illustrate the document databases
CO 5 : Distinguish the graph databases with respect to social networks

Textbooks:

1. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Sadalage, P. & Fowler, Wiley Publications, 2019.

References:

1. NoSQL For Mere Mortals, Dan Sullivan, 1st Edition, Pearson Education India, 2015.
2. Making Sense of NoSQL: A guide for Managers and the Rest of us, Dan McCreary and Ann Kelly, 1st Edition, Manning Publication/Dreamtech Press, 2013.
3. MongoDB: The Definitive Guide- Powerful and Scalable Data Storage, Kristina Chodorow, 2nd Edition, O'Reilly Publications, 2013.

Course Code: 22AM408/22AM305

AUGMENTED REALITY AND VIRTUAL REALITY

(Professional Elective – III)
(Common to CSE(AI&ML), CSE & IT)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course objectives:

1: The objective of this course is to provide a foundation to the fast-growing field of AR and make the students aware of the various AR devices.
2: To give historical and modern overviews and perspectives on virtual reality. It describes the fundamentals of sensation, perception, technical and engineering aspects of virtual reality systems.

Unit - I Introduction to Augmented Reality

What Is Augmented Reality - Defining augmented reality, history of augmented reality, The Relationship Between Augmented Reality and Other Technologies-Media, Technologies, Other Ideas Related to the Spectrum Between Real and Virtual Worlds, applications of augmented reality Augmented Reality Concepts- How Does Augmented Reality Work? Concepts Related to Augmented Reality, Ingredients of an Augmented Reality Experience.

Unit -II AR Devices and Components

AR Components – Scene Generator, Tracking system, monitoring system, display, Game scene. AR Devices – Optical See- Through HMD, Virtual retinal systems, Monitor bases systems, Projection displays, Video see-through systems.

Unit - III Introduction to Virtual Reality

Defining Virtual Reality, History of VR, Human Physiology and Perception, Key Elements of Virtual Reality Experience, Virtual Reality System, Interface to the Virtual World-Input & output- Visual, Aural & Haptic Displays, Applications of Virtual Reality

Unit - IV Representing the Virtual World

Representation of the Virtual World, Visual Representation in VR, Aural Representation in VR and Haptic Representation in VR, Case Study: GHOST (General Haptics
Open Software Toolkit) software development toolkit.

Unit - V Visual Perception and Rendering

Visual Perception - Perception of Depth, Perception of Motion, Perception of Color, Combining Sources of Information, Visual Rendering -Ray Tracing and Shading Models, Rasterization, Correcting Optical Distortions, Improving Latency and Frame Rates.

Course Outcomes: At the end of the course, student should be able to

CO 1 : Describe how AR systems work and list the applications of AR.
CO 2 : Understand and analyze the hardware requirement of AR.
CO 3 : Describe how VR systems work and list the applications of VR.

CO 4 : Understand the design and implementation of the hardware that enables VR systems to be built.

CO 5 : Understanding visual perception and rendering mechanisms.

Textbooks:

1. Virtual Reality, Steven M. LaValle, Cambridge University Press, 2016.
2. AR Game Development, Allan Fowler, 1st Edition, A press Publications, 2018

References:

1. Understanding Virtual Reality: Interface, application, and design., Sherman, W. R., & Craig, A. B. 2018.
2. Augmented Reality: Principles & Practice, Schmalstieg / Hollerer, Pearson Education India; First edition, 2016.

Course Code: 22CS402

STORAGE MANAGEMENT SYSTEMS

(Professional Elective – IV)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To introduce information growth and challenges, storage systems and Data centers.

2 : To introduce Network-Attached Storage and intelligent storage systems.

3 : To solve the optimal route establishment problems for data delivery using relevant metrics.

4 : To serve data at the end point level and to ensure reliable data delivery mechanisms.

5 : To gain knowledge on important elements. To model secured exchange of high-level data between two applications

Unit I - Introduction to Information Storage and Data Center

Evolution of Storage Architecture, Data Center Infrastructure, Virtualization and Cloud Computing.

Data Center Environment: Application, Database Management System (DBMS), Host(compute), Connectivity, Host Access to Data, Direct-Attached Storage, Storage Design Based on Application, Introduction to Flash Drives.

Data Protection: RAID Implementation Methods, RAID Array Components, RAID Techniques, RAID Levels, RAID Impact on Disk Performance, RAID Comparison.

Unit II - Intelligent Storage System

Components of an Intelligent Storage System, Storage Provisioning, Types of Intelligent Storage Systems.

Network-Attached Storage: General-purpose Servers verses NAS Devices, Benefits of NAS, File Systems and Network File Sharing, Components of NAS, NAS I/O Operation, NAS Implementations, NAS File-Sharing Protocols, Factors Affecting NAS Performance, File-Level Virtualization.

Unit III - Fibre Channel Storage Area Networks

The SAN and its Evolution, Components of FC SAN, FC Connectivity, Switched Fabric ports, Fibre Channel Architecture, Fabric Services, Switched Fabric Login Types, Zoning, FC SAN Topologies, Virtualization in SAN.

IP SAN and FCoE: iSCSI, FCIP, FCoE.

Unit IV - Object-Based and Unified Storage

Object-Based Storage Devices, Content-Addressed Storage, CAS Use Cases, Unified Storage.

Introduction to Business Continuity: Information Availability, BC Terminology, BC Planning Life Cycle, Failure Analysis, Business Impact Analysis, BC Technology Solution.

Unit V - Backup and Archive

Backup Purpose, Backup Considerations, Backup Granularity, Recovery Considerations, Backup Methods, Backup Architecture, Backup and Restore Operations, Backup Topologies, Backup in NAS Environments, Backup Targets, Data Deduplication for Backup, Backup in Virtualized Environments, Data Archive, Archiving Solution Architecture.

Managing the Storage Infrastructure: Monitoring the storage infrastructure, Storage Infrastructure Management Activities, Storage Infrastructure Management Challenges, Developing an Ideal Solution, Information Life cycle Management, Storage Tearing.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Describe the large volume data storage systems and its privacy.
CO 2 : Design the intelligent storage systems and network attached storage.
CO 3 : Illustrate the Internet Protocol based storage area networks embedding.
CO 4 : Demonstrate the business storage devices and solutions.
CO 5 : Implement Backups and maintain to protect businesses data.

Textbooks:

1. Information Storage and Management: Storing, Managing and Protecting Digital Information in Classic, Virtualized and Cloud Environment, Somasundaram Gnanasundaram, Alok Shrivastava, 2nd Edition, EMC Corporation, Wiley publication, 2012.

References:

1. Building Storage Networks, Marc Farley, 2nd Edition, Tata McGraw-Hill /Osborne, 2001.
2. Storage Area Network Fundamentals, Meeta Gupta, 1st Edition, Pearson Education, 2002.
3. Building Storage Networks, Marc Farley, Tata McGraw-Hill, Osborne, 2001.
4. Storage Networks: The Complete Reference, Robert Spalding, Tata McGraw-Hill, Osborne, 2003.

Course Code: 22CS403

ENTEREPRISE APPLICATIONS AND ARCHITECTURAL PATTERNS

(Professional Elective – IV)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1: To identify Functional Areas of an Enterprise
2: To propose Enterprise Architectural Solutions for Functional Areas
3: To apply Suitable Design Patterns in Enterprise Applications
4: To explain Enterprise Application Frameworks

Unit I - Layering and Mapping

Layering: The Evolution of Layers in Enterprise Applications, The Three Principal Layers, Choosing Where to Run Your Layers.

Mapping to Relational Databases: Architectural Patterns, The Behavioral Problem, Reading in Data, Structural Mapping Patterns, Mapping Relationships, Inheritance, Building the Mapping, Double Mapping, Using Metadata, Database Connections, Some Miscellaneous Points

Unit II - Concurrency

Concurrency Problems, Execution Contexts, Isolation and Immutability, Optimistic and Pessimistic Concurrency Control, Preventing Inconsistent Reads, Deadlocks, Transactions, ACID, Transactional Resources, Reducing Transaction Isolation for Liveness, Business and System Transactions, Patterns for Offline Concurrency Control, Application Server Concurrency.

Unit III - Distribution and logic patterns

Distribution Strategies: The Allure of Distributed Objects, Remote and Local Interfaces, Where You Have to Distribute, Working with the Distribution Boundary, Interfaces for Distribution.

Domain Logic Patterns: Transaction Script, the Revenue Recognition Problem, Domain Model, Table Module Service Layer. Data Source Architectural Patterns: Table Data Gateway, Row Data Gateway, Active Record, Data Mapper

Unit IV - Behavioral and structural patterns

Object-Relational Behavioral Patterns: Unit of Work, Identity Map, Lazy Load.

Object-Relational Structural Patterns: Identity Field, Foreign Key Mapping, Association Table Mapping, Dependent Mapping, Serialized LOB, Single Table Inheritance, Class Table Inheritance, Concrete Table Inheritance, Inheritance Mappers, Object-Relational Metadata Mapping Patterns: Metadata Mapping, Query Object, Repository.

Unit V - Web and offline patterns

Web Presentation Patterns: Model View Controller, Page Controller, Front Controller, Template View, Transform View, Application Controller. Distribution Patterns: Remote Facade, Data Transfer Object.

Offline Concurrency Patterns: Optimistic Offline Lock, Pessimistic Offline Lock, Coarse-Grained Lock, Implicit Lock.

Course Outcomes: At the end of the course, the student should be able to:

CO 1 : Understand the Layering of Enterprise applications and their mapping to Relational Databases.
CO 2 : Illustrate the Concurrency Control mechanisms in Enterprise applications
CO 3 : Demonstrate the Distribution Strategies and Domain Logic Patterns.
CO 4 : Understand and implement the Object-Relational Behavioral Patterns.
CO 5 : Implement Web Presentation Patterns and Distribution Patterns.

Textbooks:

1. Patterns of Enterprise Application Architecture, Martin Fowler, 1st Edition, Addison-Wesley Education, 2002.

References:

1. The Software Architect Elevator: Redefining the Architect's Role in the Digital Enterprise, Gregor Hohpe, 1st Edition, Shroff/O'Reilly, 2020.
2. Simple SysML for Beginners: Using Sparx Enterprise Architect, David Hetherington, 1st Edition Asatte Press, 2019.
3. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions, Gregor Hohpe, Bobby Woolf, 1st Edition, Pearson Addison-Wesley Professional, 2003.

Course Code: 22IT405

INTERNET OF EVERYTHING (IoE)

(Professional Elective-IV)
(Common to IT & CSE)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Prerequisites: Internet of Things (IoT)

Course Objectives:

1 : Understanding of Industrial IOT Infrastructure and Issues.
2 : Monitoring of End Devices.
3 : Understanding of Smart City Environment.
4 : Familiarity of Internet of Medical Things.

Unit - I Industrial IOT

Definition, IoT v IIoT, Next Generation Sensors, Sensor's calibration and validate sensor measurements, placement of IoT devices, sensors, low-cost communication system design, Top application areas include manufacturing, oil & gas, Embedded systems in the Automotive and Transportation market segment.

Unit - II Monitoring of end devices

Enablement of data-driven digital equipment model to monitor assets and systems, Introduction to device localization and tracking; different types of localization techniques, Radio-Frequency Identification (RFID) and fingerprinting, Device diversity/heterogeneity issue in IIoT networks

Unit - III Implementation of industrial IoT using AIML

Data flow, big data and how to prepare data for machine learning algorithms, Machine Learning algorithms, supervised learning & Un-supervised learning algorithms, Basics of neural network, activation functions, back-propagation.

Unit - IV Smart City

IoT in smart city& their distinctive advantages like smart environment, smart streetlight, smart water management, Smart Road & Traffic, Smart Parking & waste management. The Fourth Revolution, LEAN Production Systems, Smart and Connected Business Perspective, Smart Factories.

Unit - V Internet of Medical Things

Introduction to IoT applications in smart healthcare& their distinctive advantages - Patient Health Monitoring System (PHMS), Mobile Health Things (m-health).

Emerging Technologies for Health and Medicine: Virtual Reality, Augmented Reality, Artificial Intelligence, Robotics, Industry 4.0.

Course Outcomes: The student will be able to

CO 1 : Gain knowledge of Industrial sensors and Embedded Systems
CO 2 : Usage of IOT data for ML
CO 3 : Able to connect and communicate with different IOT devices

CO 4 : Collecting and displaying Smart City data in proper format
CO 5 : Monitoring and control of patient health using Medical IOT

Textbooks:

1. Industry 4.0: The Industrial Internet of Things, Alasdair Gilchrist, Apress, 2016.
2. Designing, Developing, and Facilitating Smart Cities Urban Design to IoT Solutions, Vangelis Angelakis, Springer, 2019
3. Emerging Technologies for Health and Medicine: Virtual Reality, Augmented Reality, Artificial Intelligence, Internet of Things, Robotics, Industry 4.0, Dac-Nhuong Le, Wiley, 2019.

References:

1. Introduction to Industrial Internet of Things and Industry 4.0, Sudip Misra, Chandana Roy, Anadarup Mukherjee, CRC Press,2021
2. Introduction to IoT, S. Misra, A. Mukherjee, and A. Roy, Cambridge University Press, 2017

Course Code: 22CS404

DATA WAREHOUSING AND DATA MINING

(Professional Elective – IV)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1: To demonstrate the value of Data mining in solving real-world problems.
2: Demonstrate Understanding of foundational concepts underlying Data mining.
3: Demonstrate Understanding of algorithms commonly used to perform various Data mining tasks.
4: To Learn and understand the cluster analysis and clustering methods used in data mining.
5: To Learn and understand the Time series, text and web mining applications and data mining trends, challenges, and its applications.

Unit I – Introduction to Data Mining and Data Warehouse

Fundamentals of Data mining, Data Mining Functionalities, Classification of Data Mining systems, Data Mining Task Primitives, Data Warehouse, Integration of a Data Mining System with a Database or a Data Warehouse System, Multidimensional Data Model, A three tier Data Warehouse Architecture, OLAP Technology for Data Mining.

Unit II – Association Rule Mining

Data Characterization, Data Discrimination, Attribute-Oriented Induction.

Association Rule Mining: Basic Concepts, Efficient and Scalable Frequent Itemset Mining methods. Mining various kinds of Association rules, from Association Analysis to Correlation Analysis, Constraint-based Association Mining.

Unit III - Classification and Prediction

Classification: Classification by Decision Tree Induction, Bayesian Classification, Rule based classification, Eager learners, Lazy Learners, Soft computing approaches -rough set, Fuzzy logic.

Accuracy and Error measures: Evaluating the accuracy of a classifier or a predictor, Ensemble Methods.

Unit IV – Cluster Analysis

Introduction to Cluster Analysis: Types of Data in Cluster Analysis, A Categorization of **Major Clustering Methods**: Hierarchical Method-BIRCH, Density-Based Methods- DBSCAN and DENCLUE, Grid-Based Methods- STING, Model-Based Clustering Methods-Expectation Maximization

Clustering High-Dimensional Data: PROCLUS, Outlier Analysis.

Unit V – Time Series, Text and Web Mining

Mining Time-series data: mining sequence patterns in Transactional Databases, Text Mining, Mining the World Wide Web, VIPS and HITS algorithms.

Applications and Trends in Data Mining: Data Mining Applications, Major issues and challenges in Data Mining, Social Impacts of Data Mining.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Understand different data mining tasks and apply the algorithms most appropriate for addressing them.
CO 2 : Discover and Analyze interesting patterns from different kinds of databases.
CO 3 : Apply the techniques of classification and prediction to build and use supervised learning from datasets.
CO 4 : Apply the techniques of clustering to implement unsupervised learning systems.
CO 5 : Understand nature of time-series, web and text data to develop methodologies and application for such data analysis and mining.

Textbooks:

1. Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber and Jian Pei, 3rd Edition, Morgan Kaufmann Publishers/Elsevier, 2011.

References:

1. Data Mining Techniques, Arun K Pujari, 2nd Edition, University Press, 2013.
2. Introduction to Data Mining, Pang Ning Tan, Michael Steinbach and Vipin Kumar, Pearson Education, 2007.
3. Data Warehousing in the Real World, Sam Anahory and Dennis Murray, Pearson Education Asia, 2011.

Course Code: 22CS405

DEEP LEARNING AND APPLICATIONS

(Professional Elective-IV)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To develop an understanding of the basic concepts of Artificial Intelligence.
2 : To gain the rationale behind various discriminative and generative models of ANN
3 : To impart knowledge on combining the capabilities of different models
4 : To appreciate the encoder-decoder architectures that solve linguistics related problems.

Unit I – Multilayer Perceptron and Back Propagation

Introduction: Perceptron, Learning Algorithm, Feed forward Neural networks, Gradient descent and the back propagation algorithm, Unit saturation, the vanishing gradient problem, and ways to mitigate it. Various Activation Functions, ReLU Heuristics for avoiding bad local minima, Heuristics for faster training, Momentum, RMSProp, ADAMS, Nestors accelerated gradient descent, Bias-Variance trade off, Regularization, Dropout, weight initialization techniques – Random, He, Xavior.

Unit II – Deep Learning Models

Convolutional Neural Networks: Architectures, convolution/pooling/stride layers, Feature maps, Fully Connected Layer, Training. Transfer Learning, ALEXNet, VGG, ResNet.

Auto Encoders: Architecture, Training, Drawbacks, Denoising Autoencoders

Restricted Boltzmann Machine: Architecture, Training, Contrastive Divergence

Variational Auto Encoders: Architecture, Training, ELBO, KL- Divergence, Reparameterization.

Adversarial Generative Networks: Architecture, Cross-Entropy and MSE, MINMAX algorithm, Challenges in training

Unit III - Applications of Deep Learning to Computer Vision

Image segmentation, object detection, Image Denoising, Face Recognition, Image Reconstruction using VAE, Image generation with Generative adversarial networks, Object Tracking, YOLO4 for computer vision tasks.

Unit IV - Vector Space Model of Semantics, Sequence Models

Vector Representation of Words: one-hot-encoding, BoW, TF-IDF and information retrieval, Co-Occurrence Matrix, SVD, Word similarity.

Distributed Representations: Word2Vec, Continuous Skip-Gram Model, Continuous Bag-of-Words model (CBOW), GloVe, Evaluations and Applications in word similarity.

Sequence Models: Sequence Learning, Drawbacks of CNN, RNN Architecture, Template architectures, BPTT algorithm, Limitations of RNN, LSTM -architecture, Activation functions, gates, functions, training, LLMs- Encoder and Decoder based models, Transformer and BERT.

Unit V – Applications of Deep Learning to NLP

Analogy reasoning: Named Entity Recognition, Opinion Mining using Recurrent Neural Networks: Parsing and Sentiment Analysis using Recursive Neural Networks: Sentence Classification using Convolutional Neural Networks, Dialogue Generation with LSTMs, Image Captioning R-CNN, Attention models. Exposure to RASA and ChatGPT.

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

CO 1 : Implement Backpropagation algorithm and solve non-liner classification problem.
CO 2 : Solve image classification problem using transfer learning.
CO 3 : Derive a functionality from a generative model to create fake images, restore the images.
CO 4 : Handle the complexities of Natural Language processing and gain proficiency in designing language models.
CO 5 : Solve real world problems using CNN, RNN, and LSTM based RASA framework.

Textbooks:

1. Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press, 2016
2. The Elements of Statistical Learning, T. Hastie, R. Tibshirani, and J. Friedman, Springer, Second Edition, 2009.

References:

1. Pattern Recognition and Machine Learning, Bishop, C, M., Springer, 2006.
2. Artificial Neural Networks, Yegnanarayana, B., PHI Learning Pvt. Ltd, 2009.
3. Matrix Computations Golub, G.H., and Van Loan, C.F, JHU Press, 2013.
4. Neural Networks: A Classroom Approach, Satish Kumar, Tata McGraw-Hill, 2004.

Course Code: 22CS431

BIG DATA ANALYTICS AND PLATFORMS LAB

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To gain familiarity with distributed file systems and scalable computing paradigms.
2 : To derive actionable knowledge from big data.
3 : To handle stream data and implement predictive analytics.
4 : To implement scalable and resilient event-driven architectures.

Task 1:

1. Configure and Install Hadoop in Pseudo-Distributed Mode.
2. Using the command line interface, execute a series of operations on the Hadoop Distributed File System (HDFS):
    1. Copy Files to HDFS
    2. Copy Files from HDFS
    3. List the directories and files in HDFS
    4. Display the contents of file in HDFS
    5. Create a directory in HDFS
    6. Remove a directory/file in HDFS
    7. Count the number of files in HDFS
    8. Merge files in HDFS
    9. Concatenate two files and store them in HDFS
    10. Find the checksum of the file in HDFS

Task 2:

1. Experiment importing and exporting data into and from HDFS using Apache Sqoop
2. Demonstrate the utilization of Apache Flume for collecting log files generated by any application server and subsequently loading of the data into HDFS

Task 3:

Develop a Map Reduce program to calculate the frequency of a word in a file.

Task 4:

Develop a MapReduce program to analyze Uber dataset to find the days on which each basement has more trips using the following dataset.

The Uber dataset consists of four columns

Dispatching_base_number	Date	Active_vechiles	Trips

Task 5:

1. Conduct the following operations within Hive:
    1. Create a Datastore
    2. Listing Datastores
    3. Switching Datastore
    4. Viewing the current Datastore
    5. Dropping a Datastore
2. Use the following schema structure and construct tables using HiveQL.

Sales Order Table: sales_order

order_id INT,
customer_id INT,
product_id INT,
order_date TIMESTAMP,
order_amount DOUBLE,
quantity INT,
discount DOUBLE,
tax DOUBLE,
total_amount DOUBLE
PRIMARY KEY (order_id)

Product Table: product

product_id INT,
product_name STRING,
category STRING,
price DOUBLE,
manufacturer STRING,
date_added TIMESTAMP,
PRIMARY KEY (product_id)

Customer Table: customer

customer_id INT,
customer_name STRING,
email STRING,
phone STRING,
address STRUCT<street: STRING, city: STRING, state: STRING, zip: INT>,date_joined TIMESTAMP,PRIMARY KEY (customer_id)PARTITIONED BY (country STRING, state STRING);

a) Find Total Sales Amount
b) Find Top N product by Sales
c) Find Customer-wise Sales
d) Find Monthly Sales Trends
e) Find Product-wise Quantity Sold
f) Find the Highest-Priced Product in Each Category

Task 6:

1. Demonstrate the usage of Hive User Defined Function (UDF) to manipulate data in a Hive table.

Task 7:

Use the following schema structure and construct HBase tables:

Table Name: employee_details
Column Families: PersonalInfo, ProfessionalInfo
PersonalInfo: name, age, gender
ProfessionalInfo: designation, salary, department

1. Create employee_details HBase table
2. Insert ten records into the employee_detailstable
3. Display details of an employee with a specific name
4. Update the Salary of an employee with a specific name
5. Delete the record of an employee with a specific name

Task 8:

1. Demonstrate Hive and HBase for seamless data transfer between two storage systems with an example also verify in HBase.

Task 9:

1. Consider an Employee dataset that includes fields such as Employee ID, Name, Department, Salary, and Year of Joining. Apply the principles of the Resilient Distributed Dataset (RDD) to perform operations on this dataset.
    a) Load the Dataset into an RDD
    b) Count the number of records in the RDD
    c) Filter out employees who joined after 2014
    d) Map the RDD to a new RDD with only employee names
    e) Calculate the average salary of employees for each department

Task 10:

1. Write a PySpark program to read an employee_data.csv file, store it in DataFrame, and perform a series of operations on the DataFrame:
    1. Calculate the average salary for each department.
    2. Create a new column 'Salary Increase' that shows the percentage increase in salary for each employee compared to the average salary in their department
    3. Determine the number of years each employee has been with the company and add this information as a new column 'YearswithCompany'
    4. Create a new column 'Salary Category' that categorizes employees into different salary ranges (Low, Medium, High)
2. Write a PySpark program to ingest data from a database, store it in a Spark DataFrame, and subsequently execute the following operations:

    1. Get the details about the DataFrame.
    2. Check for Missing Values
    3. Replace Missing Values with Mean/Median/Most frequent items of the column.
    4. Save the data back to the Database.

Task 11:

1. Develop an experiment to showcase real-time data processing capabilities using the Apache Spark Streaming API utilizing any input source such as socket, file, or Kafka.

Task 12:

1. Utilize Spark ML API and build a Linear Regression Model to predict the house price for a Boston Housing Prices dataset.
2. Write a program to demonstrate the working of the decision tree-based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

Task 13:

1. Understand the GraphX Modeling of Page Rank algorithm using PySpark

Task 14:

1. Write a PySpark program to perform analytics on the employee_details table stored in HBase.

Task 15:

1. Set up Kafka locally, create topics and channels, and implement a simple event-handling system with subscribers.

Course Outcomes: At the end of the course, the student should be able to:

CO 1 : Distribute data across HDFS and solve Big Data problem using map-reduce framework offered by Hadoop
CO 2 : Populate the Hive data warehouse and run online analytics
CO 3 : Perform real-time data and graph analytics on stream data using Apache Spark and machine learning algorithms
CO 4 : Develop PySpark programs to connect with various storage systems
CO 5 : Setup, Configure, and Manage Apache Kafka in a local machine and implement event handling system

References:

1. HADOOP: The Definitive Guide, Tom White, 4th Edition, O'Reilly 2015.
2. Spark: The Definitive Guide, Bill Chambers and Matei Zaharia, 1st Edition, O'Reilly 2018.

Course Code: 22CS432

VISUAL PROGRAMMING LAB

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1: To provide hands-on experience on .NET Framework.
2: To appreciate the asynchronous event handling feature in .Net
3: To offer end-to-end program model for web application development

List of Experiments:

Task 1:

1. Demonstrate a Console Application program to validate Email-ID and username with 3 to 16 characters length and also can have all special characters using Regular Expressions.

2. String Manipulation with the String Builder and String Classes: Demonstrate some basic string manipulation using methods of both StringBuilder and String classes**.**

Task 2:

1. Working with callbacks and delegates in C#: Demonstrate the use of delegates, callbacks, and synchronous and asynchronous method invocation.

Task 3:

1. Working with Interface Inheritance: Demonstrate the interface inheritance using explicit interface Implementation.

Task 4:

Working with Inheritance:

Employee Management System

Tokyo Company wants to register the new employee to their data.

Write an application which generates EmployeeID, UserID, EmailID, Salary

Employee ID: It is the sequence of the employee in the organization.

1. Employee Sequence should be in between 1-9999
2. It should be always in 5 digits starting with 10000, if the employee sequence is 1 then Employee ID should be 10001, if sequence is 101 then Employee ID should be 10101

User ID: It is similar to EmployeeID, but first 2 letters in the first name is prefixed in UPPERCASE Example: If firstname of the employee is Rajesh and Employee Id is 10101 then UserId is RA10101

Email ID: UserID@Tokyo.com

Salary: of the employee

1. NetCompensation of the Employee is=No. Of Years of experience*1(1 represented for 1 LAKH)
2. HRA is 10% of the salary
3. Salary=NetCompensation+HRA

Class Requirements:
Person class:

Member Name	Type
First Name	String
LastName	String
Gender	String
DOB	DateTime

Employee class:

Member Name	Type
Department	String
Location	String
EmpSequence	int
YearsOfExperience	int

This class should have **parameterised constructor** with
First name, last name, department, location, empseq, years Of Experience

Methods to implement:
Public string GenerateEmailID()
Public double CalculateSalary()

Registration class:
Members
Static in EmployeeID
Static string UserID
This class should have **parameterised constructor** with
Firstname, lastname, empseq

Methods to implement:
Public void GenerateEmployeeID()
Public void GenerateUserID()

Inheritance requirement:
Base class: Person
Child class 1: Employee class inherits Person.
Child class 2: Registration class inherits Employee

Task 5:

1. Using Reflection in C#: Demonstrate how to gather information on various types included in any assembly by using the System.Reflection namespace and some main.NET base classes.

Task 6:
Working with Assemblies:

1. Demonstrate a console application by creating a Private Assembly and use it in different applications.
2. Demonstrate how to Create a Public assembly and store it in GAC and use it in all applications.

(Use the following case study for Tasks 7 to 10)

ToDo Planner

XYZ Inc. is a leading office automation service provider. They want to develop an application using which the users can plan and manage their ToDos. As a developer you need to develop a console based application to manage the list of tasks. You need to define the implementation for scheduling a task, postpone, prepone, set priority, task completion etc. You are provided with the class and respective members description and you are supposed to implement them accordingly.

Namespace: Planning App

Use List Generic Class to Implement this Application

The class and it's members descriptions provided below and you need to define them accordingly.
1. Define a class by the name, Task and define the following private fields inside the Task class.
2. TaskID as int
3. TaskDescription as string
4. TaskStartDate as Date Time
5. TaskEndDate as Date Time
6. TaskPriority as string
7. CompletionStatus as string
8. In the Task class define a parameterized constructor to initialize the private fields to assign the parameter values accordingly. Within the parameterized constructor ensure that a unique integer value is auto-generated and assigned to TaskID private field.
9. Define a class by the name, To Do Planner with the following specifications.
10. Define a generic list collection of type Task.
11. Define the below methods accordingly.

public bool AddTask(Task obj)

This method is to add a task to the tasks generic collection of type Task. If the parameter obj is null, the method should return false, else the method should add the task details to tasks generic collection and return true.

TaskPriority can be either high or low initially completion status can be pending
public bool DeleteTask(int intTaskID)

This method should delete a task based on the task ID provided. If the deletion successful return true else return false, If no task exists with the given task ID, return false.

public bool Update TaskStatus(int intTaskID, string strStatus)

This method should update a particular task's status. If the task found and able to update the CompletionStatus, return true. If the task is not found or unable to update the CompetionStatus return false. Updated CompletionStatus can be either "Inprogress" or "Completed"

public Task GetTasks()

This method should return the list of tasks as an Array.
Use List Generic Concept to implementthisprogram
List<Task> task=new List<Task>

Task 7:

1. Sending Mail with SmtpMail : Use a simple web form to demonstrate how to use the SmtpMail class in the .Net Framework.

Task 8:

1. Using the System.NetWebClient to Retrieve or Upload Data: Demonstrate how to create windows form that can use HTTP to download and save a resource from a specified URI, upload a resource to a specified URI, or read and write data through a stream object.

Task 9:

1. Working with ASP.NET Web Pages:

Create the ASP.Net Web Application that accepts Name, Password, age, email id, and userid. All the information entry is mandatory. Password should be reconfirmed; age should be within 21-30. Email id should be valid. User id should have atleast a capital letter and digit as well as length should be between 7 and 20 characters. Use all Validation Controls.

Task 10:

1. Working with LINQ:

Create a table with given fields

Field Name	Type
EmpNo	Number
EmpName	Varchar
EmpSal	Number
EmpJob	Varchar
EmpDeptNo	Number

For the given table design a web page to display the employee information from table to grid control. Use LINQ to ADO.NET

Student Management System Application Development (For Tasks 11 to 14)

Task 11:

1. Create Website Application for Student Management System with a master page which is linked to other web pages in the application.

Task 12:

1. Create 5 content pages and design it accordingly and use different Navigation controls to navigate between content pages.

Task 13:

1. Use ADO.NET for storing and manipulating the data. Develop the necessary forms for the better user Interface.

Task 14:

1. Convert the above application to a web application using ASP.NET and SQL Server. Use IIS to deploy the web application developed in ASP.NET.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Create Private and shared Assemblies
CO 2 : Develop Asynchronous Applications
CO 3 : Deploy Web Services
CO 4 : Understand the Language Integrated Query (LINQ) Library
CO 5 : Build Database applications using ADO.NET

References:

1. Professional C# 5.0 and .NET 4.5.1, Christian Nagel, Jay Glynn and Morgan Skinner, John Wiley & Sons Inc., 2014.
2. Beginning ASP.net 4.5.1in C# and VB, Imar Spaanjaars, Wrox Publication, 2014.
3. Microsoft Visual C# Step by Step, John Sharp, O'Reilly Media, Inc., 2013.
4. A Tester's Guide to .NET Programming, Randal Root and Mary Romero Sweeney, Apress, 2006.

Course Code: 22HS404/451

ORGANIZATIONAL BEHAVIOUR

(Common to B.Tech IV Year CSE, CSE (DS), CSE (CS) and IT branches)

Instruction	:	3 Periods /week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To understand the nature, scope of Organizational Behaviour along with basic concepts and its applications in contemporary organizations.
2 : To deeply understand the role of personality and attitude, theories of motivation, emotional intelligence.
3 : To appreciate the role and means of effective communication, decision making, strategies to reduce stress and conflict in organizations.
4 : To learn about the role of power and politics, Group dynamics, Teams in the modern workplace.
5 : To develop framework for high performance through leadership, job design, performance management and Quality of work life in organizations.

UNIT I - Organisational Behaviour

Definition, need and importance of organizational behaviour – Nature and scope – Frame work – Organizational behaviour models.

UNIT II - Individual Behaviour

Personality – types – Factors influencing personality – Theories – Learning – Types of learners – The learning process – Learning Theories – Organizational behaviour modification. Misbehaviour – Types – Management Intervention - Emotions – Emotional Labour – Emotional Intelligence – Theories. Attitudes – Characteristics – Components – Formation – Measurement – Values, Perceptions – Importance – Factors influencing perception – Interpersonal perception – Impression Management. Motivation – importance – Types – Effects on work behaviour.

UNIT III - Group Behaviour

Organization structure – Formation – Groups in organizations – Influence – Group dynamics - Emergence of informal leaders and working norms – Group decision making techniques – Team building – Interpersonal relations – Communication – Control.

UNIT IV - Leadership and Power

Meaning – Importance – Leadership styles – Theories of leadership – Leaders Vs Managers – Sources of power – Power centers – Power and Politics.

UNIT V - Dynamics of Organizational Behaviour

Organizational culture and climate – Factors affecting organizational climate – Importance. Job satisfaction – Determinants – Measurements – Influence on behaviour. Organizational change – Importance – Stability Vs Change – Proactive Vs Reaction change - the change process - Resistance to change – Managing change. Stress – Work Stressors – Prevention and Management of stress – Balancing work and Life. Organizational development – Characteristics – objectives – Organizational effectiveness.

Course Outcomes: After learning this course students be able to

CO 1 : Understand the nature of Organizational Behaviour and its applications in an organization.
CO 2 : Motivate themselves and others with a positive approach.
CO 3 : Work as a team member and build a good team.
CO 4 : Be good leaders.
CO 5 : Work effectively in an organization.

Textbooks:

1. Organizational Behaviour, Luthans, Fred: 10th Edition, McGraw-Hill, 2009.
2. Organizational Behaviour, Robbins, P. Stephen, 18th Edition Pearson, 2018.

References:

1. Organizational Behaviour, Mc Shane: 9th Edition, McGraw-Hill 2022.
2. Organizational Behaviour, Nelson: 3rd Edition, Thomson, 2008.
3. Organizational Behavior- Human Behavior at Work, Newstrom W. John & Davis Keith, 12th Edition, TMH, New Delhi, 2009.
4. Management and Organizational Behavior: An Integrated perspective, Pierce and Gardner, Thomson, 2009.
5. Behavioural Process at Work Pareek Udai: Oxford & IBH, New Delhi, 2009.
6. Organizational behaviour, Schermerhorn: 9th Edition, Wiley, 2008.
7. Organizational behaviour, Hitt: Wiley, 2008.

Course Code: 22CS451

NATURAL LANGUAGE PROCESSING MODELS

(Professional Elective –V)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1: To show how language related algorithms and techniques can be applied to important real-world problems (Spell Checking, Text Document Search, Part-of- Speech Tagging.

2: To emphasize scientific evaluation and language generation mechanisms.

3: To learn various applications of Natural Language Processing tools.

Unit I – Overview

Origins and challenges of NLP-Language and Grammar-Processing Indian Languages NLP Applications, Information Retrieval. Language Modeling: Various Grammar- based Language Models, Statistical Language Model. Tools for NLP (NLTK).

Unit II - Word Level Analysis

Regular Expressions, Finite-State Automata, Morphological Parsing, Spelling Error Detection and Correction-Words and Word classes, Part-of Speech Tagging.

Syntactic Analysis: Context-free Grammar, Constituency, Parsing, Probabilistic Parsing.
Word level analysis methods in NLTK tool.

Unit III - Semantic Analysis

Meaning Representation, Lexical Semantics, Ambiguity, Word Sense Disambiguation, Discourse Processing: cohesion, Reference Resolution, Discourse Coherence and Structure.
Semantic analysis methods in NLTK tool.

Unit IV - Natural Language Generation

Architecture of NLG Systems, Generation Tasks and Representations Application of NLG.
Machine Translation: Problems in Machine Translation, Characteristics of Indian Languages, Machine Translation Approaches, Translation involving Indian Languages.

Unit V - Information Retrieval

Design features of Information Retrieval Systems, Classical, Non-classical, Alternative Models of Information Retrieval, Evaluation of the IR System. Lexical Resources: World Net, Frame Net Stemmers, POS Tagger, Research Corpora.
Information retrieval methods in NLTK.

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

CO 1: Extract information from text automatically using concepts and methods from natural language processing (NLP) including stemming, n-grams, POS tagging, and parsing

CO 2: Develop speech-based applications that use speech analysis (phonetics, speech recognition, and synthesis).
CO 3: Analyze the syntax, semantics, and pragmatics of a statement written in a natural language.
CO 4: Develop a conversational agent that uses natural language understanding and generation.
CO 5: Evaluate the performance of NLP tools and systems.

Textbooks:

1. Natural Language Processing and Information Retrieval, Tanveer Siddiqui, U.S. Tiwary, Oxford University Press, 2023.
2. Natural Language Processing with Python: Analyzing text with Natural Language Toolkit, Steven Biord, Ewan Klein, and Edward Loper, ORielly Publications, January 2011.

References:

1. Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Daniel Jurafsky and James H Martin, 2nd Edition, Prentice Hall, 2013.
2. Natural Language Processing in Action: Understanding, analyzing, and generating text with Python, Hobson Lane, Hannes Hapke, Cole Howard, Manning publications 2019.

Course Code: 22CS452

PENETRATION TESTING AND INCIDENT RESPONSE

(Professional Elective –V)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : To describe Cybersecurity skills.
2 : Describe penetration testing tools and the benefits to an organization.
3 : Describe digital forensics and digital evidence.
4 : To test and exploit systems using various tools.
5 : To understand the impact of hacking in real time machines

Unit I - Introduction

Penetration Testing phases/Testing Process, types and Techniques, Blue/Red Teaming, Strategies of Testing, Non-Disclosure Agreement Checklist, Phases of hacking, Open source/proprietary Pentest Methodologies

Unit II - Information Gathering and Scanning

Information gathering methodology- Foot printing, Competitive Intelligence-DNS Enumerations-Social Engineering attacks, Port Scanning-Network Scanning-Vulnerability Scanning- NMAP scanning tool- OS Fingerprinting- Enumeration.

Unit III - Preparing for the Inevitable Incident

What is Incident Response, Concept of the Attack Life cycle, Goals of Incident Response, Incident Response Process, Initial Response, Investigation, Remediation,

Pre-Incident Preparation: Identifying Risk, Policies That Promote a Successful IR, working with Outsourced IT. Preparing the IR Team: Defining the Mission, Communication Procedures, Deliverables, Resources for the IR Team

Unit IV - Incident Detection and Characterization

Collecting Initial Facts, Maintenance of Case Notes, Setting Expectations with Management, Initial Development of Leads, Turning Leads into Indicators, The Life cycle of Indicator Generation, Resolving Internal Leads, Resolving External Leads

Unit V - Live Data Collection

When to Perform a Live Response, Selecting a Live Response Tool, Live Data Collection on Unix-Based Systems, Forensic Duplication: Forensic Image Formats, Partition Image, Logical Image, Image Integrity. Traditional Duplication Network Evidence: The Case for Network Monitoring, Types of Network Monitoring, Event-Based Alert Monitoring, statistical Modelling

Course Outcomes: At the end of the course, the student should be able to:

CO 1: gain Cybersecurity skills.
CO 2: understand the methodologies and techniques used for penetrating
CO 3: acquire skills to work in the Cybersecurity field as a Cybersecurity Analyst
CO 4: learn key steps in the forensic process and important data to collect
CO 5: identify security vulnerabilities and weaknesses in the target applications

Textbooks:

1. Cybersecurity Incident Response: How to Contain, Eradicate, and Recover from Incidents, Eric C. Thompson, Apress, USA, 2018
2. Incident Response & Computer Forensics, Jason T. Luttgens Matthew Pepe, Third Edition McGraw-Hill Education, 2014.

References:

1. Linux Revealed: Mastering the Penetration Testing, Raphael Hertzog, Jim O'Gorman, Kali Offsec Press, 2017.

Course Code: 22CS453

ROBOTIC PROCESS AUTOMATION

(Professional Elective - V)
(Common to CSE, CSE(AI&ML) & IT)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1: To introduce the functional elements of Robotic Process Automation
2: To impart knowledge on the concepts of RPA
3: To understand the development the Bot
4: To educate on various RPA vendors
5: To understand the data preparation and process mining

Unit I - Introduction to Robotic Process Automation

RPA Foundations: Introduction to RPA, RPA Skills-web technologies, Agile, databases etc.

Process Methodologies: Lean, Six Sigma Roles and Levels, Applying Lean and Six Sigma to RPA.

Unit II – Planning

The Preliminaries, ROI for RPA, RPA Use Cases, Security, Monitoring, and Deployment, The Design, Next-Generation Technologies, RPA Solution Architect, RPA Supervisor, Change Management

Unit III -Bot Development

Preliminaries, Installation of UiPath, Flowcharts and Sequences, Log, Message, Variables, Loops and Conditionals, Debug, Common UiPath Functions, The UiPath Orchestrator, Bot Development.

Unit IV -Data Preparation

Types of Data, Big Data, The Issues with Big Data, The Data Process, Types of Algorithms.

Process Mining - Old Way Vs. Process Mining, How Process Mining Works, The Future of Process Mining.

Unit V - RPA Vendors

UiPath, IQ Bot, Bot Store, Blue Prism, EdgeVerve, PEGA.
Deployment and Monitoring- Testing, Going into Production, Monitoring, Security, Scaling.

Course Outcomes: At the end of the course, student should be able to :

CO 1 : Understand the concepts of RPA and RPA skills lean six sigma.
CO 2 : Design of RPA use cases and security, change management.
CO 3 : Implement the concepts of BOT
CO 4 : Understand the data preparation algorithms and process mining.
CO 5 : Develop and demonstrate an application of RPA with different vendors.

Textbooks:

1. The Robotic Process Automation Handbook – A Guide to implementing RPA systems, Tom Taulli, Apress, Springer Science,2020.
2. Learning Robotic Process Automation, Alok Mani Tripathi, Packet publishing Ltd,2018.

References:

1. Robotic Process Automation for dummies, Steve Kaelbel, John Wiley and sons, 2018.
2. Designing BOTS, Amir shevat, O'Reilly, 2017.

Course Code: 22IT451

DESIGN PATTERNS

(Professional Elective-V)
(Common to IT, CSE, CSE (AI&ML, DS & CS))

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Prerequisites:

1. Object Oriented Programming through Java
2. Software Engineering

Course Objectives:

1: To make the students understand the basic concepts of Design patterns.
2: To understand the various Design patterns.
3: To understand the importance of design patterns for development of a reusable product.

Unit I – Introduction

What Is a Design Pattern? Design Patterns in Smalltalk MVC, Describing Design Patterns, The Catalog of Design Patterns, Organizing the Catalog, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.

Unit II – A Case Study

Designing a Document Editor: Design Problems, Document Structure, Formatting, Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations Spelling Checking and Hyphenation, Summary What to Expect from Design Patterns.

Unit III – Creational Patterns

Abstract Factory, Builder, Factory Method, Prototype, Singleton, Discussion of Creational Patterns.

Unit IV – Structural Patterns

Adapter, Bridge and Composite, Decorator, façade, Flyweight, Proxy.

Unit V – Behavioral Patterns

Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor, Discussion of Behavioral Patterns.

Course outcomes: At the end of the course, the student will be able to

CO 1 : Appreciate the basic concepts of design patterns and able to know how to select and use the design patterns
CO 2 : Identify the design pattern in the existing code and use of creational patterns
CO 3 : Apply and use the structural patterns
CO 4 : Identify and use the behavioral patterns
CO 5 : Find and catalog patterns in the object-oriented software

Textbooks:

1. Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Addison-Wesley, 1995.
2. Java™ Design Patterns: A Tutorial, James W. Cooper, Addison Wesley, 2000.

References:

1. Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML, Mark Grand, Volume 2, Wiley DreamTech.
2. Patterns in Java, Mark Grand, Volume 2, Wiley DreamTech, 2008.
3. Java Enterprise Design Patterns, Mark Grand, Wiley DreamTech, 2006.

Course Code: 22IT452

GENERATIVE AI

(Professional Elective-V)
(Common to IT, CSE, CSE (AI&ML, DS & CS))

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Prerequisites: Machine learning

Course Objectives:

1 : To understand how Large Language Models (LLMs) work
2 : How to integrate generative AI into your personal and professional work
3 : Societal impacts of generative AI

Unit I - Introduction to Generative AI

Artificial Intelligence is Nothing New, Generative AI- A New Approach to AI, Human Equivalents of Predictive and Generative AI, Generative Artificial Intelligence Applications, Generative AI Models

Unit II - Introduction to LLMs

Introduction to Modern Natural Language Processing, Birth of Large Language Models, Explosion of Large Language Models, Applications of Large Language Models, Limitations and Risks of Large Language Models

Unit III - Data Privacy and Safety

Training of LLMS: Emergent Properties of LLMs, Considerations, Strategies for Improving Generations from a Safety Perspective, Data Policies and Regulations

Unit IV - Content Creation and Machine Augmented Work

Rise of Synthetic Media Generative AI in creative workflows, Intellectual Property(IP) in the LLM era, Professional Applications, Personal Usage, Detection of Machine generation, economic consequences

Unit V - Risks, Practices and Ethics of Generative AI

Prospective Developments of LLMs, Social and Technical risks of LLMs, Best practices of Responsible Use, Ethics-Informed AI regulations, Towards AI Governance Framework,

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Define Generative AI and describe various Generative AI applications.
CO 2 : Explore the applications, limitations, and risks of LLMs
CO 3 : Understand how the LLMs are trained and emergent properties of LLMs
CO 4 : Leverage LLMs in professional and personal settings
CO 5 : Explore and formalize the best practices for responsible use of generative AI models

Textbooks:

1. Generative Ai - The Future of Everything, Sharad Gandhi, Christian Ehl, 2023
2. Introduction to Generative AI: An ethical, societal, and legal overview, Numa Dhamani and Maggie Engler, Manning Publications,2023.

References:

1. Generative AI Models, Jovan Pehcevski, Arcler Press, 2024
2. Artificial Intelligence & Generative AI for Beginners: The Complete Guide, David M.Patel , GD Publishing, 2023