ANNEXURE-C

Course Curriculum

Of 4th Semester
FYUGP through Design Your Degree Mode

Structure for 4th semester courses offered through "Design Your Degree"

Basket A

Course Category	Course Code	Course Title	Credits	Maximum Marks	Minimum Pass Marks
MJ	DYD-MJ-401	Financial Accounting	04	100	40
MJ	DYD-MJ-402	Human Resource Management	04	100	40
MJ	DYD-MJ-403	Business Environment	04	100	40
MJ	DYD-MJ-404	Capstone Project	04	100	40
MN	DYD-MN-405	System Analysis & Design	04	100	40

Basket B

Course Category	Course Code	Course Title	Credits	Maximum Marks	Minimum Pass Marks
MJ	DYD-MJ-406	Data Structures	04	100	40
MJ	DYD-MJ-407	Computer Networks and Data Communication	04	100	40
MJ	DYD-MJ-408	Database Systems	04	100	40
MJ	DYD-MJ-409	Capstone Project	04	100	40
MN	DYD-MN-410	Linear Algebra	04	100	40

Course Code: DYD-MJ-401 Course Title: Financial Accounting

Credits: 4

Course Objectives

i. Develop basic conceptual knowledge about books of accounts in the double entry system
ii. Document business transactions and record them into accounts
iii. Learn the process of preparing and finalizing books of accounts

Learning Outcomes

i. Understand the purpose and significance of accounting
ii. Gain knowledge of books of accounts and preparation methods
iii. Know the process of finalization of books of account and final accounts
iv. Acquire expertise in accounting software

Unit I Financial Accounting Fundamentals

- Meaning and Scope of Financial Accounting
- Book Keeping and Accounting
- Branches of Accounting
- Principles, Concepts, and Conventions of Accounting
- Cash vs. Accrual system
- Double Entry vs. Single Entry
- Accounting Equation and Terminology.

Classroom Activities

- **Concept Comparison Drill**: Students compare cash vs. accrual system with real-life examples (e.g., shopkeeper vs. corporate firm).
- **Equation Practice**: Quick exercises applying the accounting equation (Assets = Liabilities + Equity).
- **Case Discussion**: Debate on double-entry vs. single-entry systems.

Mini Project

- Case Study on Application of accounting principles to business situations and importance of well organised accounting system to minimise errors and frauds.
- Case study on emerging issues in accounting.

Unit IIAccounting Cycle, Ledger and Errors

- Accounting Cycle: Identification of transactions, Source Documents, Journalizing, Rules of Debit and Credit
- Ledger, posting, balancing, Trial Balance
- Rectification of Errors, Suspense Account, Closure of temporary accounts, Post-closing trial balance, Reversing entries.

Classroom Activities

- **Journalizing Relay**: Teams practice recording transactions into journals quickly and accurately.
- **Trial Balance Puzzle**: Students balance accounts and identify errors.
- **Error Rectification Drill**: Practice correcting common mistakes in ledgers.

Mini Project

- **Real-Life Accounting Application**: Apply accounting principles to a small business scenario (e.g., a café or bookstore), documenting transactions, posting to ledgers, and preparing a trial balance.

Unit III Secondary Books and Depreciation Accounting

- Secondary Books: Cash Book, Petty Cash Book, Bank Reconciliation Statement
- Depreciation Accounting: Depreciation and Amortization, Methods of Depreciation, Inventory Valuation.

Classroom Activities

- **Cash Book Drill**: Students record sample transactions in cash book and petty cash book.
- **Bank Reconciliation Exercise**: Practice reconciling differences between cash book and bank statement.
- **Depreciation Methods Demo**: Groups calculate depreciation using straight-line and diminishing value methods.

Mini Project

- **Bank Reconciliation Statement**: Prepare a reconciliation statement using data from variouseducational institutions in J&K
- **Inventory Valuation Exercise**: Apply FIFO, LIFO, and weighted average methods to a sample dataset.

Unit IV Financial Statements and Accounting of Accruals and Deferrals

- Preparation of Financial Statements: Adjusted Trial Balance, Trading Account, Profit and Loss Account, Balance Sheet . Adjustment entries.

- **Classroom Activities**
- **Accrual vs. Deferral Role Play**: Students act out transactions showing accruals and deferrals.
- **Statement Preparation Lab**: Groups prepare Trading, P&L, and Balance Sheet from a given adjusted trial balance.
- **Comparative Analysis Exercise**: Compare financial statements of two companies to identify differences.

Mini Project

- **J&K Bank vs. Khyber Industries,** Saifco Cements vs TCI Max Cements and similar case studies of business & Industrial establishments in J&K . Collect and study.

Suggested Readings

1. Gupta, R. L., & Radhaswamy. *Advanced accountancy* (Vol. I). Sultan Chand & Sons.
2. Ghosh, T. P.. *Fundamentals of accounting*. Sultan Chand & Sons.
3. Tulsian, P. C. *Accounting for CA foundation students*. Tata McGraw Hill.
4. Tulsian, P. C. *Advanced accountancy* (Vol. I). Tata McGraw Hill.
5. Maheshwari, S. N. . *Financial accounting*. Sultan Chand & Sons.

Online Resources:

1. **GNUCash** – Free open-source accounting software for practice.
2. **LedgerSMB** – Open-source ERP and accounting system.
3. **Mikkel Larsen – Accountants for Good (TEDx)** How accountants can drive sustainability and social impact.

Course Code: DYD-MJ-402

Course Title: Human Resource Management

Credits: 4

Course Objectives

- Develop foundational understanding of the principles, functions, and challenges of Human Resource Management, including its evolution and modern trends.
- Equip students with practical skills in workforce planning, recruitment, selection, training, performance management, and compensation systems.
- Foster analytical and managerial competenciesto design HR policies, evaluate employee satisfaction, and implement effective HR practices in diverse organizational contexts.

Learning Outcomes

- Explain and differentiate key HRM concepts, functions, and practices, including distinctions between Personnel Management and Human Resource Development.
- Apply HR techniques such as job analysis, recruitment strategies, training methods, and performance management systems to real-world organizational scenarios.
- Analyze and evaluate HR policies and practices (e.g., employee satisfaction surveys, HR policy comparisons, PMS effectiveness) to recommend improvements that enhance workforce productivity, loyalty, and cohesiveness.

Unit I: Human Resource Management

- Meaning, Definitions, Characteristics, Objectives, Importance, Functions, Process, Challenges, Recent Trends
- HR Manager – Duties and Responsibilities
- Human Resource Development – Meaning, Definitions, Characteristics, Objectives, Importance, Functions, Process
- Differences between Personnel Management and HRD .

Classroom Activities

- **Concept Mapping**: Students create a mind map of HRM functions and processes.
- **Role Play**: Act out the duties of an HR manager in different scenarios (recruitment, conflict resolution).
- **Debate**: Personnel Management vs. HRD – groups argue differences and relevance in modern organizations.

Mini Project

- **Employee Satisfaction Survey**: Conduct a survey in a chosen organization, analyze results, and present findings with recommendations.

Unit II: Human Resource Planning, Recruitment & Selection

- HR Planning – Meaning, Importance, Benefits, Scope
- Job Analysis, Job Design, Job Description, Job Enrichment, Job Evaluation
- Recruitment – Meaning, Definitions, Sources (Traditional & Modern)
- Selection – Meaning, Definitions, Process
- Identification of five dark qualities in an individual before selection and placement **Project:** Compare HR policies of a public sector bank (SBI) vs. a private sector bank (HDFC).

Classroom Activities

- **Job Description Workshop**: Students draft job descriptions for different roles (bank teller, software engineer).
- **Recruitment Source Comparison**: Groups evaluate traditional vs. modern recruitment sources.
- **Selection Simulation**: Mock interviews and selection exercises to identify candidate qualities.

Mini Project

- **HR Policy Comparison**: Compare HR policies of SBI (public sector) vs. HDFC (private sector), focusing on recruitment, training, and employee benefits.

Unit III: Human Resource Practices

- Induction and Orientation – Meaning, Definitions, Objectives, Purposes
- Training – Meaning, Need, Benefits, Methods, Pros & Cons
- Identification of Training & Development Needs
- HR Development of Managers and Employees
- Performance Management System (PMS) – Meaning, Definitions, Objectives.

Classroom Activities

- **Orientation Design**: Students design an induction program for new employees.
- **Training Methods Demo**: Groups present pros and cons of different training methods (on-the-job, e-learning, workshops).
- **PMS Case Study**: Analyze performance management systems used in Indian companies.

Mini Project

- **PMS Effectiveness Study**: Research examples of PMS adopted by select Indian companies (Infosys, TCS, Reliance) and evaluate their effectiveness.

Unit IV: Compensation and Reward System

- Compensation – Meaning, Definitions, Objectives, Importance
- Wages and Salary
- Perquisites, Fringe Benefits, Bonus, Incentives (Meanings only)
- Performance-based pay, Merit-based pay, Skill-based pay, Competency-based pay, Dual system of payment
- Promotion – Meaning, Definitions, Features, Methods (Seniority vs. Meritocracy) **Project:** Prominent HR staffing solution providers in India and their operations.

Classroom Activities

- **Compensation Structure Exercise**: Students design a compensation package for a sample role.
- **Promotion Debate**: Seniority vs. Meritocracy – groups argue which method is more effective.
- **Reward System Brainstorm**: Discuss innovative reward systems beyond salary (flexible work, recognition programs).

Mini Project

- **HR Staffing Solutions Research**: Identify prominent HR staffing solution providers in India (e.g., TeamLease, Randstad, Adecco) and analyze their operations.

Suggested Readings

1. Aswathappa, K. *Human resource management*. Tata McGraw Hill.
2. Gupta, S. K., & Joshi, R. *Human resource management*.
3. Mamoria, C. B. *Personnel management*.
4. Rao, V. S. P. *Human resource management*.
5. Rama Chandra, K., et al. *Human resource management*. HPH.

Online Resources:

1. **OrangeHRM** – Free open-source HR management software.
2. **Sentrifugo** – Open-source HRMS with modules for performance appraisal, recruitment, and leave management.
3. **MIT OpenCourseWare – Organizational Processes** Covers HRM-related topics like motivation, leadership, and organizational behavior.

Course Code: DYD-MJ-403 Course Title: Business Environment

Credits: 4

Course Objectives

i. Understand components of business environment
ii. Identify internal and external elements affecting business environment
iii. Critically examine recent developments in economic and business policies of the Government
iv. Evaluate and understand best business policies in Indian business environment

Learning Outcomes

i. Identify micro and macro dimensions affecting business
ii. Relate business environment with day-to-day activities
iii. Explain impact of fiscal and monetary policy on business
iv. Evaluate environmental factors in decision-making

Unit I: Business Environment

- Concepts, Components & Importance
- Types: Internal, External, Micro, Macro
- Environmental Scanning, Scope & Characteristics
- Objectives, Uses, Limitations of Environmental Analysis

Classroom Activities

- **Concept Mapping**: Students create a visual map of internal vs. external, micro vs. macro factors.
- **Case Discussion**: Debate on how environmental scanning helps businesses anticipate challenges.
- **Terminology Quiz**: Quick-fire quiz on key terms like environmental analysis, scope, limitations.

Mini Project

- **SWOT Analysis**: Conduct a SWOT analysis of an organization of choice (e.g., Reliance, Infosys, or a local startup) and present findings.

Unit II: Economic, Financial, Industrial & Legal Environment

- Economic Environment: Nature & Structure, Monetary & Fiscal Policies, Economic Planning in India, Economic Reforms of 1991
- Financial Environment: Financial System, Functions of Commercial Banks, Financial Institutions, Stock Exchange, NBFCs, FEMA
- Industrial & Legal Environment: Industrial Policies, Competition Act 2002, Corporate Governance

Classroom Activities

- **Policy Impact Debate**: Groups discuss how fiscal vs. monetary policy impacts business decisions.
- **Stock Exchange Simulation**: Students simulate buying/selling shares to understand financial markets.
- **Legal Case Study**: Analyze the Competition Act 2002 and its impact on Indian businesses.

Mini Project

- **AI in Business Operations**: Research and present how AI is transforming operations in sectors like banking, retail, or manufacturing.

Unit III: Socio-Political Environment

- Social environment: Cultural heritage, social attitudes, foreign culture influence, castes & communities, joint family systems, linguistic & religious groups
- Types of social organization
- Social responsibilities of business

Classroom Activities

- **Cultural Influence Role Play**: Students act out scenarios showing how social attitudes affect business practices.
- **CSR Brainstorm**: Groups design a corporate social responsibility initiative for a company.
- **Startup Challenges Discussion**: Explore how social and political factors affect startups.

Mini Project

- **Macro Environmental Factors for Startups**: Analyze how factors like government policy, social attitudes, and technology trends impact Indian startups.

Unit IV: Political & Global Environment

- Political Environment: Directive Principles of State Policy, Centre–State Relations, Impact on Business
- Global Environment: Global Trends in Business & Management, MNCs – Importance, Advantages, Weaknesses, Foreign Investment

Classroom Activities

- **Directive Principles Case Study**: Discuss how state policies influence business operations.
- **Global Trends Debate**: Groups debate advantages and disadvantages of MNCs in India.
- **Foreign Investment Simulation**: Students role-play as investors deciding whether to invest in India.

Mini Project

- **Ford & Chevrolet Exit Analysis**: Research reasons behind Ford and Chevrolet winding up operations in India, linking them to political and global environmental factors.

Suggested Readings

1. Saleem, S. *Business environment*. Pearson India.
2. Aswathappa, K. *Essentials of business environment*.
3. Adhikari, M. *Economic environment of business*. Excel Books; Sultan Chand.
4. George, A., & Steiner, G. A. *Business, government and society*. Macmillan.
5. Sherlekar, S. A. *Modern business organization and management*.

Online Resources:

1. **Open Knowledge Repository (World Bank)** – Free reports on global business environments.
2. **OECD iLibrary** – Free access to economic and business environment publications.
3. **World Economic Forum** – Talks and discussions on business, government, and society.

Course Code: DYD-MJ-404 Course Title: Capstone Project

Credits: 4

Course Description

The Capstone Project is the culminating academic experience of the semester. It requires students to integrate concepts, theories, and skills acquired from **Financial Accounting, Human Resource Management, Business Environment, and System Analysis & Design** into a comprehensive project. Students will design, develop, and present a solution to a real-world business problem, demonstrating their ability to analyze, plan, implement, and evaluate systems and processes.

Course Objectives

i. Integrate multidisciplinary knowledge into a single, cohesive project.
ii. Apply accounting, HRM, business environment, and system design principles to solve real-world problems.
iii. Develop managerial, analytical, and technical skills through project execution.
iv. Enhance teamwork, communication, and project management abilities.
v. Produce a tangible product or process that reflects professional standards.

Learning Outcomes

By the end of the course, students will be able to:

i. Design and document a complete system integrating **accounting, HR, and business environment analysis**.
ii. Apply **SDLC models** to develop and test a working prototype.
iii. Prepare and analyze **financial statements** and integrate them into system workflows.
iv. Evaluate **HR policies, recruitment, and performance management systems** within the project.
v. Assess **business environment factors** (economic, legal, socio-political, global) influencing the project.
vi. Present a professional-grade project report and defend it through oral presentation.

Project Components

The capstone project must include:

1. **Financial Accounting Component**
   - Preparation of financial records, statements, and reconciliation.
   - Application of accounting principles to project operations.
2. **Human Resource Management Component**
   - HR planning, recruitment, and performance management system design.
   - Employee satisfaction survey or HR policy integration.
3. **Business Environment Component**
   - SWOT analysis of the chosen organization/idea.
   - Impact of economic, legal, and socio-political factors on the project.

4. **System Analysis & Design Component**
   - Requirements engineering (SRS document).
   - System design (architectural, component-level, UI).
   - Implementation, testing, and maintenance of a prototype.

Course Schedule (14 Weeks)

Week	Activity
1–2	Orientation, topic selection, team formation
3–4	Problem identification, environmental scanning, SWOT analysis
5–6	Requirements gathering, preparation of SRS document
7–8	Financial accounting integration (statements, reconciliation)
9–10	HRM integration (policies, recruitment, PMS design)
11	System design (architecture, UI, documentation)
12	Implementation & testing (unit, integration, acceptance)
13	Maintenance plan, configuration management, version control
14	Final report submission & presentation

Deliverables

1. **Project Proposal** (Week 2) – Problem statement, objectives, scope.
2. **SRS Document** (Week 6) – Requirements, use cases, functional/non-functional specifications.
3. **Design Document** (Week 11) – Architecture, UI mockups, component-level design.
4. **Prototype/Working Model** (Week 12) – Demonstration of system functionality.
5. **Final Report** (Week 14) – Comprehensive documentation including accounting, HR, business environment, and system design.
6. **Presentation & Viva** – Oral defense of project before faculty panel.

Suggested Project Themes

- **ERP System for a College/Small Business based in Jammu & Kashmir** (integrating HR, accounting, and environment analysis).
- **HR & Payroll Management System for a tourism & travel agency based in J&K** with accounting and compliance features.
- **E-commerce Platform for Kashmir arts & Crafts** with integrated financial tracking and HR policies.
- **Startup Business Model Analysis based in Jammu & Kashmir** with system design, accounting, and HR planning.

Course Code: DYD-MN-405 Course Title: System Analysis & Design

Credits: 4

Course Objectives

i. Introduce fundamental concepts of software engineering and development practices
ii. Cover software development life cycle models, requirements engineering, design, testing, and maintenance

Learning Outcomes

i. Understand software engineering principles and practices
ii. Apply SDLC models to real-world problems
iii. Perform requirements analysis and specification
iv. Design software systems using appropriate techniques
v. Conduct software testing and maintenance
vi. Use tools and techniques for project management

Unit I – Introduction to Software Engineering

- Definition and nature of software
- Importance of software engineering
- Software process models: Waterfall, Spiral, Incremental, Agile
- Project management: planning, scheduling, tracking

Unit II – Requirements Engineering

- Elicitation, analysis, specification, validation
- Functional and non-functional requirements
- Use case modelling
- Requirements documentation

Unit III – Software Design

- Principles and concepts
- Architectural design
- Component-level design
- User interface design
- Design documentation

Unit IV – Software Testing and Maintainence

- Testing strategies: Unit, Integration, System, Acceptance
- Test case design techniques
- Debugging and testing tools
- Software quality assurance

- Types of Software maintenance
- Maintenance process and models
- Configuration management activities
- Change control and version management

Practicals:

- Create detailed SRS documents (introduction, description, requirements, interfaces, non-functional requirements, appendices)
- Prepare design strategies (architectural, component-level, UI, documentation)
- Present SRS and design documents for review

- Individual assignment: implementation, unit testing, debugging, system integration, acceptance testing, maintenance, documentation
- Maintain project logbook

- Project development in areas such as web, mobile, desktop, database, management systems, e-commerce, educational software, social media, game development
- Steps: problem identification, requirements analysis, design, implementation, testing, maintenance, documentation, presentation
- Submit documentation and present project for assessment

Suggested Readings

1. Sommerville, I. *Software engineering*. Pearson Education.
2. Pressman, R. S. *Software engineering: A practitioner's approach*. McGraw-Hill.
3. Mall, R. *Fundamentals of software engineering*. PHI Learning.
4. Jalote, P. *Software engineering*. Narosa Publishing House.
5. Aggarwal, K. K., & Singh, Y. *Software engineering*. New Age International Publishers.

Online Resources:

1. **StarUML** – Free UML modeling tool for system design.
2. **PlantUML** – Open-source tool for creating UML diagrams from text.
3. **Draw.io** – Free diagramming tool for ER diagrams, flowcharts, and system models.

Basket B

Course Code: DYD-MJ-406 Course Title: Data Structures

Credits: 4

Course Objectives

i. Data structure deals with organizing large amount of data in order to reduce space complexity and time requirement.
ii. This course gives knowledge of Algorithms, different types of data structures

Learning Outcomes

i. Assess how the choice of data structures and algorithm design methods impacts the performance of programs.
ii. Choose the appropriate data structure and algorithm design method for a specified application.
iii. Write programs using object-oriented design principles.
iv. Solve problems using data structures such as linear lists, stacks, queues, hash tables, binary trees, binary search trees and graphs and writing programs for these solutions.

UNIT I: Data Structure Basics

- Basic terminology, Linear and non-Linear Data structures.
- Linear Arrays and implementation; traversing, inserting and deleting.
- Sorting and searching; Linear search and Binary search, selection sort, Bubble sort, Insertion sort, Quick sort and merge sort.

Classroom Activities

- **Terminology Match Game**: Students match terms (e.g., linear, non-linear, traversal) with definitions.
- **Array Simulation**: Use index cards to simulate array operations (insert, delete, traverse).
- **Sorting Race**: Divide students into groups, each group acts out a sorting algorithm (bubble sort, selection sort) with numbers on cards.

Mini Project

- **Student Record Manager**: Implement a program to store student names/IDs in an array, with options to insert, delete, search, and sort records.

UNIT II: Stacks

- Introduction to stacks, static vs. Dynamic implementation of stack
- Stack Operations; create, push, pop and peep, Applications of stack; Polish Notations, Postfix, infix and Prefix Expressions.
- Queues: Linear Queues and their representation in memory, Static vs Dynamic implementation of Linear Queues, Queue operations; create, enqueue and dequeue, doubly ended queue, circular queue, Applications of Queue.

Classroom Activities

- **Stack Role Play**: Students form a line and simulate push/pop operations.
- **Expression Conversion Exercise**: Convert infix expressions to postfix/prefix in small groups.
- **Queue Simulation**: Use chairs arranged in a line to demonstrate enqueue/dequeue operations.

Mini Project

- **Expression Evaluator**: Build a program that evaluates postfix expressions using a stack.
- **Ticket Counter Simulation**: Implement a queue-based system to simulate customers waiting for tickets (linear, circular, and deque versions).

UNIT III: Linked lists and their implementation

- Traversinginserting,and deleting operations in linked lists (singly, doubly and circular).
- Applications of linked lists.

Classroom Activities

- **Human Linked List**: Students act as nodes, holding data and pointing to the next person.
- **Insert/Delete Drill**: Practice inserting and deleting nodes by rearranging the "human linked list."
- **Comparison Discussion**: Groups compare arrays vs. linked lists in terms of memory and flexibility.

Mini Project

- **Music Playlist Manager**: Create a program using a doubly linked list to add, delete, and traverse songs.
- **Circular Linked List Game**: Implement a Josephus problem (elimination game) using circular linked lists.

UNIT IV:Introduction to Trees

- Tree terminology, Binary treeand their representation in memory
- Dynamic implementation of binary tree; Traversing Binary tree, Binary search tree, Application of trees.
- Introduction to Graphs; Graph terminology, types of graphs, Graph Implementation; Static and Dynamic, Applications of Graphs.

Classroom Activities

- **Binary Tree Construction**: Students build a binary tree on the board step by step.
- **Traversal Practice**: Groups perform preorder, inorder, and postorder traversals on a sample tree.
- **Graph Walkthrough**: Use classroom seating as nodes and draw edges to represent connections.

Mini Project

- **Binary Search Tree Dictionary**: Implement a BST to store words and meanings, with search functionality.
- **Campus Map Navigator**: Represent campus locations as a graph and implement BFS/DFS to find paths between buildings.

Suggested Readings

1. Balaguruswamy, E. *Programming in ANSI C*. Tata McGraw Hill.
2. Trebley, R., & Sorenson, B. *An introduction to data structures with applications*. McGraw Hill Kongakusha, 1976.
3. Horowitz, E., & Sahni, S. *Data structures*. SBCS Publication.

Online Resources:

1. **GeeksforGeeks Practice** – Free coding practice on data structures.
2. **LeetCode (Free Tier)** – Hands-on problems to strengthen data structure skills.
3. **CS50 IDE (Harvard)** – Free online coding environment for practicing C-based data structures.
4. **Visualgo.net** – Interactive visualizations of data structures and algorithms.

Course Code: DYD-MJ-407 **Course Title: Computer Networks and Data Communication**

Credits: 4

Course Objectives

i. Acquaint students with the basics of data communication and networking.
ii. Provide a structured approach to explain how networks work from the inside out.
iii. Cover the physical layer of networking, computer hardware, and transmission systems.

Learning Outcomes

i. Understand network communication using the layered concept, OSI, and Internet Model.
ii. Understand various types of transmission media, network devices, and performance evaluation parameters.
iii. Grasp concepts of flow control, error control, and LAN protocols; explain design and algorithms used in physical and data link layers.
iv. Understand LAN working principles, physical/logical addressing, subnetting, and supernetting and learn functions of network management systems and analyze connection establishment and congestion control with respect to TCP protocol.

UNIT I:Introduction to Data Communication

- Data representations
- Data transmission: Terminology (simplex, half duplex, full duplex).
- Transmission modes: Serial and parallel transmission.
- Data and signals: Analog and digital signal, Analog and digital data.
- Introduction to computer networks: Concept, applications, types of networks (PAN, LAN, CAN, MAN, WAN).
- Network topologies (p2p, bus, star, ring, mesh, tree, hybrid).
- Networking/internetworking devices (router, switch, hub, repeater, gateway).

Classroom Activities

- **Signal Role Play**: Students act out simplex, half-duplex, and full-duplex communication.
- **Topology Mapping**: Groups draw and compare different network topologies (star, bus, ring, mesh).
- **Device Match Game**: Match networking devices (router, switch, hub, repeater) with their functions.

Mini Project

- **Network Topology Simulator**: Create a small simulation (using software like Cisco Packet Tracer or Python scripts) to demonstrate different topologies.
- **Device Catalog**: Build a digital catalog explaining the role of each networking device with diagrams.

UNIT II: Transmission Impairments: attenuation, distortion, noise

- Data rate limits: Nyquist bitrate, Shannon's capacity.
- Transmission media and characteristics:
  - Guided: twisted pair, coaxial cable, optical fiber.
  - Unguided: terrestrial microwave, satellite, microwave, radio.
- Modulation techniques: ASK, FSK, PSK, Quadrature.
- Pulse code modulation (PCM), Pulse amplitude modulation (PAM).

Classroom Activities

- **Noise Demonstration**: Use audio clips to simulate attenuation, distortion, and noise.
- **Cable Comparison**: Hands-on inspection of twisted pair, coaxial, and fiber optic cables.
- **Modulation Sketches**: Students draw ASK, FSK, and PSK waveforms on the board.

Mini Project

- **Media Performance Comparison**: Measure and compare data transfer speeds using different media (LAN cable vs. Wi-Fi).
- **Modulation Demo**: Write a simple program to simulate digital modulation techniques (ASK/FSK).

UNIT III: OSI Reference Model and its Seven Layers

- TCP/IP model and its comparison with OSI reference model.
- IP addressing (static and dynamic), IP address classes, classful IP addressing.
- Connection-oriented vs. connectionless services.
- TCP/UDP.

Classroom Activities

- **Layered Skit**: Students act as different OSI layers passing data packets.
- **IP Addressing Drill**: Practice classifying IP addresses into classes (A, B, C).
- **TCP vs UDP Debate**: Groups debate reliability vs. speed in real-world applications.

Mini Project

- **IP Address Calculator**: Build a tool that calculates subnet masks and ranges for given IP addresses.
- **Protocol Analyzer**: Use Wireshark to capture packets and identify TCP vs UDP traffic.

UNIT IVPoint-to-point or Switched Networks and Packet-switched Networks

- Concept of diagrams.
- Principles of congestion.
- Inter/intra-domain routing: RIP, OSPF, BGP.
- Network troubleshooting.

Classroom Activities

- **Routing Table Exercise**: Students manually construct routing tables for small networks.
- **Congestion Simulation**: Demonstrate congestion by sending multiple requests to a server.
- **Troubleshooting Lab**: Practice diagnosing network issues using ping, traceroute, and nslookup.

Mini Project

- **Routing Protocol Demo**: Simulate RIP or OSPF routing using Packet Tracer.

Congestion Control Experiment: Implement a simple TCP congestion control simulation in Python.

Suggested Readings

- Forouzan, B. A., *Data Communication and Networking*, McGraw-Hill.
- Tanenbaum, A. S., *Computer Networks*, Prentice Hall.
- Stallings, W., *Computer Networking with Internet Protocols and Technology*, Prentice Hall of India.
- Douglas Comer, *Internetworking with TCP/IP, Volume 1*, Pearson.

Online Resources:

1. DEV Community – Free Networking Tutorials Collection
2. Wireshark Packet Analyzer

Course Code: DYD-MJ-408 Course Title: Database Systems

Credits: 4

Course Objectives

i. Understand the purpose, components, and applications of database systems and their role in modern data management.
ii. Apply data integrity rules and normalization techniques to design and optimize relational database, and utilize SQL and NoSQL query languages to perform data manipulation, querying, and management tasks effectively.

Learning Outcomes

i. Explain the fundamental components of a database system and their functions in managing and organizing data.
ii. Apply normalization techniques to create efficient and well-structured relational database schemas.
iii. Write and execute complex SQL queries and NoSQL commands to manipulate and retrieve data.
iv. Apply transaction management principles and concurrency control methods to handle data integrity and synchronization.

UNIT I:Introduction to Databases

- Purpose of database systems.
- Components of database systems.
- Applications of DBMS.
- Three-tier DBMS architecture.
- Data independence, database schema, instance.
- Data modeling, Entity-Relationship model, Relational model.

Classroom Activities

- **DBMS vs File System Debate**: Groups discuss advantages of DBMS over traditional file systems.
- **ER Diagram Workshop**: Students design ER diagrams for simple scenarios (library, hospital, e-commerce).
- **Schema vs Instance Drill**: Quick exercises distinguishing schema definitions from actual data instances.

Mini Project

- **Student Information System ER Model**: Design an ER diagram and relational schema for managing student records.
- **Library Database Schema**: Create a relational model for books, authors, and borrowers.

UNIT II: Relational Database Design

- Data integrity rules.
- Functional dependency.
- Normalization: 1NF, 2NF, 3NF, BCNF.
- Multi-valued dependencies.
- Pitfalls in relational database design.
- De-normalization.

Classroom Activities

- **Normalization Game**: Students take a poorly designed table and progressively normalize it to 3NF.
- **Functional Dependency Puzzle**: Identify functional dependencies in sample datasets.
- **Pitfall Discussion**: Analyze examples of bad database designs and propose fixes.

Mini Project

- **Course Registration Database**: Design and normalize a schema for course enrollment.
- **Hospital Management Schema**: Apply normalization to patient, doctor, and appointment records.

UNIT III: Query Language

Relational algebra.

- Introduction to DDL, DML, DCL, TCL.
- Integrity constraints, database keys.
- SQL basic operations, aggregate functions.
- SQL joins, views, subqueries.
- Introduction to NoSQL query languages: MongoDB Query Language, Cassandra Query Language (CQL).

Classroom Activities

- **SQL Query Race**: Teams compete to write correct queries for given problems.
- **Join Practice Lab**: Students experiment with INNER, LEFT, RIGHT, and FULL joins on sample tables.

- **NoSQL Demo**: Hands-on with MongoDB or Cassandra to compare with SQL.

Mini Project

- **E-commerce Database Queries**: Write SQL queries for product search, order history, and customer details.
- **NoSQL Blog Database**: Implement a blog system using MongoDB with collections for posts, users, and comments.

UNIT IV: Transaction Management and Concurrency Control

- Transaction system concepts.
- Desirable properties of transactions: ACID properties.
- Schedules.
- Serializability and concurrency control.
- Lock-based concurrency control (2PL, deadlocks).
- Time stamping methods.
- Optimistic methods.
- Database recovery management.

Classroom Activities

- **ACID Property Role Play**: Students act out transactions showing atomicity, consistency, isolation, durability.
- **Deadlock Simulation**: Groups simulate resource locking and identify deadlock scenarios.
- **Recovery Drill**: Discuss recovery techniques after simulated system crashes.

Mini Project

- **Banking Transaction System**: Implement transactions with rollback and commit operations in SQL.
- **Concurrent Ticket Booking System**: Simulate multiple users booking tickets with concurrency control.

Suggested Readings

1. Elmasri, R., & Navathe, S. B. *Fundamentals of database systems* (7th ed.). Pearson, 2015.
2. Silberschatz, A., Korth, H., & Sudarshan, S. *Database system concepts* (7th ed.). McGraw-Hill, 2020.
3. Korth, H. F., & Silberschatz, A. *Database system concepts* (7th ed.). McGraw-Hill Education, 2020.
4. Date, C. J. *An introduction to database systems* (8th ed.). Pearson, 2003.
5. Ramakrishnan, R., & Gehrke, J. *Database management systems* (3rd ed.). McGraw-Hill, 2002.

6. Rob, P., & Coronel, C. *Database systems: Design, implementation, & management* (13th ed.). Cengage Learning, 2018.

Online Resources:

1. **MySQL** – Widely used open-source relational database system.
2. **PostgreSQL** – Advanced open-source database with strong community support.
3. **SQLite** – Lightweight, file-based database system for practice.
4. **MongoDB (Community Edition)** – Free NoSQL database for exploring non-relational systems.

Course Code: DYD-MJ-409 **Course Title: Capstone Project**

Credits: 4

Course Description

The Capstone Project is the culminating academic experience of the semester. It requires students to integrate concepts, theories, and skills acquired from **Data Structures, Linear Algebra, Computer Networks, and DBMS** into a comprehensive project. Students will design, develop, and present a solution to a real-world computing problem, demonstrating their ability to analyze, plan, implement, test, and maintain systems.

Course Objectives

1. Integrate multidisciplinary knowledge into a cohesive project.
2. Apply data structures, linear algebra, networking, and database principles to solve real-world problems.
3. Develop analytical, technical, and managerial skills through project execution.
4. Enhance teamwork, communication, and project management abilities.
5. Produce a tangible product or process that reflects professional standards.

Learning Outcomes

By the end of the course, students will be able to:

- Design and document a complete system integrating **data structures, linear algebra, networking, and database concepts**.
- Apply **SDLC models** to develop and test a working prototype.
- Implement efficient **data structures and algorithms** for problem-solving.
- Use **linear algebra techniques** for computations in graphics, ML, or optimization.
- Apply **networking principles** for communication and data transfer.
- Design and query **databases (SQL/NoSQL)** for data management.
- Present a professional-grade project report and defend it through oral presentation.

Project Components

The capstone project must include:

1. **Data Structures Component**
   - Use of arrays, linked lists, stacks, queues, trees, or graphs in the project.
   - Implementation of algorithms for searching, sorting, or traversal.
2. **Linear Algebra Component**
   - Application of matrices, vector spaces, or eigenvalues in computations.
   - Examples: graphics transformations, recommendation systems, or ML models.
3. **Computer Networks Component**

- Incorporation of communication protocols, addressing, or client-server architecture.
- Demonstration of data transmission or network simulation.

4. **Database Management System Component**
   - Design of relational schema and normalization.
   - Implementation of SQL queries, joins, and transactions.
   - Optional: NoSQL integration for scalability.

Course Schedule (14 Weeks)

Week	Activity
1–2	Orientation, topic selection, team formation
3–4	Problem identification, environmental scanning, feasibility study
5–6	Requirements gathering, preparation of SRS document
7–8	Data structures and linear algebra integration into system design
9–10	Database schema design, SQL/NoSQL implementation
11	Networking integration (client-server, communication protocols)
12	System design (architecture, UI, documentation)
13	Implementation & testing (unit, integration, acceptance)
14	Final report submission & presentation

Deliverables

1. **Project Proposal** (Week 2) – Problem statement, objectives, scope.
2. **SRS Document** (Week 6) – Requirements, use cases, functional/non-functional specifications.
3. **Design Document** (Week 12) – Architecture, UI mockups, component-level design.
4. **Prototype/Working Model** (Week 13) – Demonstration of system functionality.
5. **Final Report** (Week 14) – Comprehensive documentation including data structures, linear algebra, networking, and DBMS integration.
6. **Presentation & Viva** – Oral defense of project before faculty panel/ external experts.

Course Code: DYD-MN-410 Course Title: Linear Algebra

Credits: 4

Course Objectives

i. Understand the fundamental concepts of linear algebra and its application in computer science.
ii. Learn matrix operations, vector spaces, and their use in solving systems of linear equations.
iii. Explore eigen values, eigen vectors, and their relevance in data science and machine learning.
iv. Apply linear transformations and decompositions in practical scenarios, such as graphics and computer vision.

Learning Outcomes

i. Understand vectors, matrices, and systems of linear equations.
ii. Analyze vector spaces and subspaces.
iii. Solve problems using matrix operations and eigen values.
iv. Apply linear algebra techniques to real-world scenarios.

UNIT I: Introduction to Linear Algebra

- Scalars, vectors, and matrices: Definitions, notation, and basic operations.
- Matrix operations: Addition, scalar multiplication, matrix multiplication, transposition.
- Systems of linear equations: Row reduction, echelon forms, Gaussian elimination.
- Determinants and matrix inverses: Definition, properties, computation.
- Application: Solving linear systems using inverses.

Classroom Activities

- **Matrix Bingo**: Students solve quick matrix addition/multiplication problems to mark off answers on a bingo card.
- **Gaussian Elimination Relay**: Teams compete to reduce systems of equations to echelon form step by step.
- **Determinant Puzzle**: Small groups compute determinants of 2×2 and 3×3 matrices using different methods.

Mini Project

- **System Solver App**: Write a program that takes a system of linear equations and solves it using Gaussian elimination or matrix inverses.
- **Cryptography with Matrices**: Encode and decode simple messages using matrix multiplication.

UNIT II: Vector Spaces and Subspaces

- Definitions of vector spaces and subspaces.
- Linear combinations and span.
- Basis and dimension.
- Rank-Nullity Theorem: Row space, column space, null space.
- Linear independence and dependence, with applications in error detection, coding theory, computer graphics.

Classroom Activities

- **Vector Space Sorting**: Provide sets of vectors; students decide if they form a subspace.
- **Basis Hunt**: Groups find bases for given vector sets and compare dimensions.
- **Rank-Nullity Role Play**: Students act as vectors in row space, column space, and null space to visualize the theorem.

Mini Project

- **Error Detection Demo**: Implement a simple parity-check code using linear independence concepts.
- **Graphics Transformation**: Show how vector spaces apply in 2D graphics by scaling/rotating shapes using matrices.

UNIT III: Eigen values, Eigen vectors, and Diagonalization

- Eigen values and eigenvectors: Definitions, calculation methods, characteristic polynomial.
- Diagonalization: Criteria and applications.
- Applications in machine learning and data science: PCA for dimensionality reduction, eigen values in Markov models, Google's Page Rank algorithm.

Classroom Activities

- **Eigenvalue Treasure Hunt**: Students calculate eigenvalues/eigenvectors for small matrices and match them to applications.
- **PCA Demo with Data**: Use a small dataset (e.g., student scores) and manually compute the first principal component.
- **Google PageRank Simulation**: Groups simulate a small web graph and compute rankings using eigenvalues.

Mini Project

- **PCA for Image Compression**: Implement a simple program that reduces image dimensions using eigen decomposition.

- **Markov Chain Model**: Build a small probability transition matrix and analyze steady states using eigenvalues.

UNIT IV: Linear Transformations and Matrix Factorization

- Linear transformations: Mapping between vector spaces, matrix representation, kernel and image.
- Matrix factorizations: LU decomposition, QR decomposition, Singular Value Decomposition (SVD).
- Applications in computer science: Graphics transformations, machine learning algorithms (neural networks, recommendation systems), robotics.

Classroom Activities

- **Transformation Gallery Walk**: Students apply transformations (rotation, scaling, reflection) to shapes on graph paper.
- **LU Decomposition Race**: Teams break down matrices into L and U factors step by step.
- **SVD Visualization**: Show how SVD works by decomposing a small matrix and interpreting singular values.

Mini Project

- **Graphics Engine Demo**: Implement a program that applies linear transformations to move/rotate objects in 2D space.
- **Recommendation System Prototype**: Use matrix factorization (basic SVD) to suggest items to users based on preferences.
- **Robotics Path Planning**: Represent robot movements as transformations and simulate simple navigation.

Suggested Readings

1. Lay, D. C. *Linear algebra and its applications* (6th ed.). Pearson, 2021.
2. Strang, G. *Introduction to linear algebra* (5th ed.). Wellesley-Cambridge Press, 2023.
3. Anton, H., & Rorres, C. *Elementary linear algebra* (12th ed.). Wiley, 2022.
4. Kumaresan, S. *Linear algebra* (2nd ed.). Prentice-Hall of India, 2017.
5. Lipschutz, S., & Lipson, M. *Schaum's outline of linear algebra* (6th ed.). McGraw-Hill Education, 2020.
6. Kreyszig, E. *Advanced engineering mathematics* (11th ed.). Wiley India, 2020

Online Resources:

1. **GNU Octave** – Free open-source alternative to MATLAB for matrix computations.
2. **SageMath** – Open-source mathematics software with strong linear algebra support.
3. **NumPy (Python Library)** – Widely used for matrix and vector operations.
4. **GeoGebra** – Free interactive tool for visualizing vectors and transformations.
5. **'3Blue1Brown' Youtube Channel** – Essence of Linear Algebra Visual, intuitive explanations of vectors, matrices, and transformations.