MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. **RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956**

MK University, Patan
Faculty of Engineering Technology,
Department of Data Science Engineering

DIPLOMA (DATA SCIENCE ENGINEERING) SEM-I									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	DDSE101	ENGINEERING MATHEMATICS-I	4	0	4	40	60	100
2	MAJOR	DDSE102	ENGINEERING PHYSICS	4	2	6	90	60	150
3	MAJOR	DDSE103	ENGINEERING CHEMISTRY	4	2	6	90	60	150
4	MAJOR	DDSE104	PROGRAMMING IN C	4	2	6	90	60	150
5	MINOR	DDSE105	WORKSHOP PRACTICE	0	2	2	50	00	50
TOTAL				16	8	24	360	240	600

DIPLOMA (DATA SCIENCE ENGINEERING) SEM-II									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	DDSE201	ENGINEERING MATHEMATICS-II	4	0	4	40	60	100
2	MAJOR	DDSE202	DATA STRUCTURES AND ALGORITHMS	4	2	6	90	60	150
3	MAJOR	DDSE203	DATABASE MANAGEMENT SYSTEMS	4	2	6	90	60	150
4	MINOR	DDSE204	WEB TECHNOLOGIES	4	2	6	90	60	150
5	SEC	DDSE205	COMMUNICATION SKILL	2	0	2	00	50	50
TOTAL				18	6	24	310	290	600

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. **RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956**

DIPLOMA (DATA SCIENCE ENGINEERING) SEM-III									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	DDSE301	STATISTICS FOR DATA SCIENCE	4	2	6	90	60	150
2	MAJOR	DDSE302	OBJECT ORIENTED PROGRAMMING	4	2	6	90	60	150
3	MAJOR	DDSE303	COMPUTER NETWORKS & SECURITY	4	2	6	90	60	150
4	MINOR	DDSE304	INDUSTRIAL VISIT REPORT	0	2	2	50	00	50
5	IKS	DDSE305	IKS-ANNCIENT INDIAN ENGINEERING PRACTICE	0	2	2	50	00	50
TOTAL				12	10	22	370	180	550

DIPLOMA (DATA SCIENCE ENGINEERING) SEM-IV									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	DDSE401	INTRODUCTION TO MACHINE LEARNING	4	2	6	90	60	150
2	MAJOR	DDSE402	DATA VISUALIZATION & BI TOOLS	4	2	6	90	60	150
3	MAJOR	DDSE403	BIG DATA BASICS (HADOOP, SPARK)	4	0	4	40	60	100
4	MINOR	DDSE404	DIGITAL LOGIC SYSTEMS	4	0	4	40	60	100
5	VAC	DDSE405	ENVIRONMENTAL SCIENCE	2	0	2	00	50	50
TOTAL				18	4	22	260	290	550

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. **RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956**

DIPLOMA (DATA SCIENCE ENGINEERING) SEM-V									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	DDSE501	ADVANCED MACHINE LEARNING	4	2	6	90	60	150
2	MAJOR	DDSE502	CLOUD COMPUTING FOR DATA SCIENCE	4	0	4	40	60	100
3	MAJOR	DDSE503	NATURAL LANGUAGE PROCESSING	4	2	6	90	60	150
4	MINOR	DDSE504	OPERATING SYSTEMS	4	0	4	40	60	100
5	SEC	DDSE505	MIN PROJECT	0	2	2	50	00	50
TOTAL				16	6	22	310	240	550

DIPLOMA (DATA SCIENCE ENGINEERING) SEM-VI									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	DDSE601	DEEP LEARNING FUNDAMENTALS	4	2	6	90	60	150
2	MAJOR	DDSE602	IOT FOR DATA SCIENCE	4	2	6	90	60	150
3	MAJOR	DDSE603	ETHICS& GOVERNANCE IN AI/DS	4	2	6	90	60	150
4	MINOR	DDSE604	DIPLOMA PROJECT	0	6	6	150	00	150
TOTAL				12	12	24	420	180	600

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-I

COURSE CODE: DDSE101

COURSE NAME: ENGINEERING MATHEMATICS-I

Course Objectives:

- To develop foundational mathematical skills essential for mechanical engineering applications
- To apply algebraic, trigonometric, and calculus concepts to solve engineering problems
- To interpret and analyze data using statistical methods
- To build problem-solving abilities through applied mathematics
- To prepare students for advanced engineering mathematics in subsequent semesters

Course Outcomes: At the end of the course students shall be able to

CO1	**Solve** algebraic equations and apply them to engineering problems
CO2	**Apply** trigonometric functions to analyze mechanical systems
C03	**Perform** basic differentiation and integration relevant to engineering applications
C04	**Analyze** data using measures of central tendency and dispersion

Unit	Content	Credit	Weightage
I	**Algebra and Trigonometry** **Topics:** • **Algebra:** Quadratic equations, simultaneous linear equations (2 and 3 variables) • Arithmetic and geometric progressions • **Trigonometry:** Trigonometric ratios, identities, compound angles • Heights and distances (engineering applications) • Complex numbers: basics and operations • **Applications:** Simple harmonic motion, projectile motion, force resolution	1	25%

II	**Differential Calculus**
	Topics:
	• Functions, limits, and continuity
	• Derivatives: Standard formulas
	• Rules of differentiation: Product, quotient, chain rule
	• Applications of derivatives:
	  ○ Rate of change (velocity, acceleration)
	  ○ Maxima and minima (optimization problems)
	  ○ Tangents and normals
	• Partial differentiation (introduction)
	• **Applications:** Optimization in design, motion analysis, slope of curves
		1	25%

III	**Integral Calculus** **Topics:** • Indefinite integrals: Standard formulas	1	25%

	• Methods of integration: Substitution, integration by parts
	• Definite integrals and properties
	• Applications of integration:
	  ○ Area under curves
	  ○ Volume of solids of revolution
	  ○ Center of gravity/centroid (basic concepts)
	• **Applications:** Area calculation in engineering drawings, volume of tanks/containers		

IV	**Statistics and Probability** **Topics:** • **Statistics:** Data classification, frequency distribution • Measures of central tendency: Mean, median, mode • Measures of dispersion: Range, standard deviation, variance • Graphical representation: Histogram, frequency polygon, ogive • **Probability:** Basic concepts, addition and multiplication theorems • **Applications:** Quality control, measurement analysis, manufacturing data interpretation	1	25%

Textbooks:

- Primary: *Engineering Mathematics* — NP Bali & Dr. Manish Goyal
- Primary: *A Textbook of Engineering Mathematics* — B.S. Grewal

Reference books:

- *Advanced Engineering Mathematics* — H.K. Das
- *Engineering Mathematics* — D. G. Gupta
- *Basic Technical Mathematics with Calculus* — Allyn J. Washington
- *Mathematics for Mechanical Engineering* — B.V. Ramana

Online Platforms:

1. NPTEL Videos: "Basic Course in Mathematics" for engineering
2. Coursera: "Pre-Calculus" by University of California, Irvine

COURSE CODE: DDSE102

COURSE NAME: ENGINEERING PHYSICS

Course Objectives:

- To understand fundamental physics principles relevant to mechanical engineering
- To apply physics concepts to solve practical engineering problems
- To develop skills in measurement, experimentation, and data analysis
- To correlate theoretical physics with mechanical systems and applications
- To build foundation for advanced engineering courses

Course Outcomes: At the end of the course students shall be able to

CO1	**Apply** mechanics principles to analyze forces, motion, and energy in mechanical systems
CO2	**Explain** thermal physics concepts relevant to heat engines and refrigeration
C03	**Demonstrate** understanding of optics and acoustics in

	engineering contexts
C04	**Perform** measurements using physical instruments and analyze experimental data

Unit	Content	Credit	Weightage
I	**Mechanics & Properties of Matter** **Topics:** • Scalars and vectors, force resolution, moment of force • Laws of motion, friction, work, energy, power • Circular motion, centripetal force • Elasticity: Stress, strain, Hooke's law, Young's modulus • Surface tension and viscosity (basic concepts) • **Applications:** Machine design, material strength, fluid mechanics basics	1	25%
II	**Thermal Physics & Thermodynamics** **Topics:** • Heat and temperature, thermal expansion • Calorimetry, specific heat capacity • Laws of thermodynamics (zeroth, first, second) • Heat transfer: conduction, convection, radiation • Kinetic theory of gases (basic) • **Applications:** Heat engines, refrigeration, insulation materials	1	25%
III	**Waves, Optics & Acoustics** **Topics:** • Simple harmonic motion, wave motion • Sound: characteristics, intensity, Doppler effect • Ultrasonics and applications • Reflection, refraction, lenses, optical instruments • Fiber optics (basic principles) • **Applications:** Machine vibration, NDT, optical measurements, noise control	1	25%
IV	**Modern Physics & Material Science** **Topics:** • Quantum physics basics: photons, matter waves • Lasers: principles, types, applications • Semiconductors: basics • Superconductivity (elementary concepts) • Nanotechnology introduction • **Applications:** Laser machining, sensors, advanced materials	1	25%

Textbooks:

- Primary: *Engineering Physics* — D. R. Khanna & H. N. Srivastava
- Primary: *Engineering Physics* — R. K. Gaur & S. L. Gupta

Reference books:

- *Fundamentals of Physics* — Halliday, Resnick & Walker
- *Concepts of Physics* — H. C. Verma

- *Engineering Physics* — M. N. Avadhanulu & P. G. Kshirsagar
- *Practical Physics* — C. L. Arora

Online Platforms:

- SWAYAM/NPTEL: "Engineering Physics" courses by IITs/NITs

PRACTICAL LIST:

Section A: Mechanics

1. Vernier Calipers & Screw Gauge: Measurement of dimensions of given objects
2. Simple Pendulum: Determination of 'g' and study of laws of pendulum
3. Young's Modulus: By Searle's method or cantilever
4. Coefficient of Friction: Using inclined plane
5. Force Table: Verification of law of parallelogram of forces

Section B: Thermal Physics

6. Specific Heat Capacity: Of solid/liquid using calorimeter
7. Thermal Conductivity: Of good conductor (Searle's apparatus)
8. Mechanical Equivalent of Heat: Using Joule's calorimeter
9. Coefficient of Linear Expansion: Using optical lever

Section C: Waves & Optics

10. Sonometer: Verification of laws of vibrating strings
11. Melde's Experiment: Transverse and longitudinal modes
12. Compound Pendulum: Determination of 'g' and radius of gyration
13. Optical Bench: Focal length of convex lens
14. Prism: Refractive index using spectrometer

Section D: Modern Physics

15. LASER: Determination of wavelength using diffraction grating
16. Photoelectric Effect: Verification of Einstein's equation
17. PN Junction Diode: Characteristics
18. Thermistor: Temperature-resistance characteristics

COURSE CODE: DDSE103

COURSE NAME: ENGINEERING CHEMISTRY

Course Objectives:

- To provide fundamental knowledge of chemistry relevant to mechanical engineering applications
- To understand material properties, corrosion, fuels, and lubricants from chemical perspective
- To develop skills in chemical analysis, quality control, and material testing
- To correlate chemical principles with mechanical systems and manufacturing processes
- To build foundation for materials science, metallurgy, and environmental engineering

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** water treatment processes for industrial applications
CO2	**Analyze** properties of fuels and lubricants used in mechanical systems
C03	**Identify** corrosion mechanisms and prevention methods
C04	**Apply** principles of electrochemistry to batteries and corrosion control

Unit	Content	Credit	Weightage

I	**Water Chemistry & Treatment** **Topics:** • Water impurities: hardness, alkalinity, pH • Water softening methods: lime-soda, ion exchange • Boiler feed water treatment: scale and sludge formation, prevention • Cooling water treatment • Drinking water standards • **Applications:** Boiler operations, cooling systems, industrial water supply	1	25%
II	**Fuels & Combustion** **Topics:** • Classification of fuels: solid, liquid, gaseous • Calorific value determination: bomb calorimeter • Solid fuels: coal analysis (proximate & ultimate) • Liquid fuels: petroleum refining, petrol, diesel, octane/cetane number • Gaseous fuels: LPG, CNG, biogas • Combustion calculations • **Applications:** IC engines, furnaces, power generation	1	25%
III	**Lubricants & Corrosion** **Topics:** • Lubrication: mechanisms, types of lubricants • Properties: viscosity index, flash point, pour point • Additives in lubricants • Corrosion: types, mechanisms (electrochemical) • Factors affecting corrosion • Corrosion prevention methods • **Applications:** Machine maintenance, automotive, industrial equipment	1	25%
IV	**Engineering Materials & Polymers** **Topics:** • Cement: composition, setting and hardening • Refractories: properties, classification • Polymers: addition, condensation, engineering plastics • Composite materials: introduction • Batteries: primary, secondary, fuel cells • **Applications:** Construction materials, polymers in engineering, energy storage	1	25%

Textbooks:

- Primary: *Engineering Chemistry* — Jain & Jain
- Primary: *Engineering Chemistry* — Dr. O. P. Verma

Reference books:

- *A Textbook of Engineering Chemistry* — S. S. Dara & S. S. Umare
- *Engineering Chemistry* — R. P. Mani & K. N. Mishra
- *Chemistry for Engineering Students* — B. S. Jai Prakash & R. Venugopal
- *Applied Chemistry* — H. D. Gesser

Online Platforms:

- SWAYAM/NPTEL: "Engineering Chemistry" courses by IITs
- Khan Academy: Chemistry fundamentals
- MERLOT: Chemistry learning materials

PRACTICAL LIST:

Section A: Water Analysis

1. Determination of Hardness: By EDTA titration method
2. Alkalinity Determination: Using acid-base titration
3. pH Measurement: Using pH meter/universal indicator
4. Chloride Content: By argentometric method

Section B: Fuels & Lubricants

5. Viscosity Measurement: Using Ostwald viscometer/Redwood viscometer
6. Flash Point Determination: Using Abel/Pensky Martens apparatus
7. Calorific Value: Bomb calorimeter demonstration
8. Saponification Value: Of oil sample

Section C: Corrosion & Electrochemistry

9. Corrosion Rate Measurement: Weight loss method
10. Galvanic Series Determination
11. Electroplating: Copper plating on iron
12. EMF Measurement: Of simple galvanic cell

Section D: Materials & Polymers

13. Cement Setting Time: Initial and final setting time
14. Polymer Identification Tests
15. Preparation of Polymer: Phenol-formaldehyde/Bakelite
16. Refractory Properties: Porosity, thermal shock resistance

COURSE CODE: DDSE104

COURSE NAME: PROGRAMMING IN C

Course Objectives:

- To introduce fundamental programming concepts and problem-solving techniques.
- To develop logical thinking and algorithmic reasoning using structured programming.
- To enable students to write, debug, and execute programs in the C language.
- To prepare a strong foundation for advanced programming, data structures, and software development.
- To familiarize students with good programming practices and documentation.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** basic programming concepts, algorithms, and flowchart design.
CO2	**Develop** programs using control structures, loops, and functions.
C03	**Implement** arrays, strings, and structures to handle data.
C04	**Apply** pointers and file handling concepts in real-world applications.

Unit	Content	Credit	Weightage
I	**Introduction to Programming & C Basics**	1	25%

	• Problem-solving approaches: Algorithms, Flowcharts, Pseudocode • Introduction to C: History, features, structure of a C program • C tokens: Keywords, identifiers, constants, operators • Data types: int, float, char, double • Input/output functions: printf(), scanf(), getchar(), putchar() • Operators: Arithmetic, relational, logical, assignment, bitwise • **Applications**: Simple calculator, temperature conversion		
II	**Control Structures & Loops** • Decision making: if, if-else, nested if, switch-case • Loops: while, do-while, for • Loop control statements: break, continue, goto • Nested loops and pattern printing • **Applications**: Menu-driven programs, number series, star patterns	1	25%
III	**Arrays, Strings & Functions** • Arrays: 1D and 2D arrays, initialization, traversal • Strings: Declaration, standard library functions (strlen, strcpy, strcmp) • Functions: Definition, declaration, call by value, call by reference • Recursion: Concept, examples (factorial, Fibonacci) • Storage classes: auto, static, register, extern • **Applications**: Matrix operations, sorting, searching, string manipulation	1	25%
IV	**Pointers, Structures & File Handling** • Pointers: Concept, arithmetic, pointer to array, pointer to function • Structures: Definition, accessing members, array of structures • File handling: Opening, closing, reading, writing files • Dynamic memory allocation: malloc(), calloc(), free() • **Applications**: Student record system, file-based data storage, memory management	1	25%

Textbooks:

- *Let Us C* – Yashwant Kanetkar
- *Programming in ANSI C* – E. Balagurusamy

Reference books:

- *The C Programming Language* – Brian W. Kernighan & Dennis M. Ritchie
- *C: The Complete Reference* – Herbert Schildt
- *Problem Solving and Program Design in C* – Jeri R. Hanly & Elliot B. Koffman
- *Programming with C* – Byron Gottfried

Online Platforms:

NPTEL

Programming in C by Prof. Satyadev Nandakumar
Problem Solving through Programming in C by IIT Kharagpur

Coursera

C for Everyone: Programming Fundamentals by University of California, Santa Cruz

PRACTICAL LIST:

Module I: Basics & Control Structures

1. Write a program to find the sum and average of three numbers.
2. Write a program to swap two numbers (with and without third variable).
3. Write a program to check whether a number is even or odd.
4. Write a program to find the largest among three numbers using nested if-else.
5. Write a program to display the multiplication table of a given number.
6. Write a program to print the Fibonacci series up to n terms.
7. Write a program to check whether a number is prime or not.

Module II: Arrays & Strings

8. Write a program to find the largest and smallest element in an array.
9. Write a program to add two matrices.
10. Write a program to multiply two matrices.
11. Write a program to sort an array using bubble sort.
12. Write a program to search an element in an array using linear search.
13. Write a program to reverse a string without using library functions.
14. Write a program to check whether a string is palindrome or not.

Module III: Functions & Recursion

15. Write a function to calculate factorial of a number using recursion.
16. Write a function to check whether a number is Armstrong or not.
17. Write a program to find GCD of two numbers using recursion.
18. Write a program to swap two numbers using call by reference.
19. Write a program to demonstrate the use of a static variable.

Module IV: Pointers, Structures & Files

20. Write a program to demonstrate pointer arithmetic.
21. Write a program to store student information (roll, name, marks) using structures and display it.
22. Write a program to read from a file and count vowels, consonants, and digits.
23. Write a program to copy contents from one file to another.
24. Write a program to dynamically allocate memory for an array and find its sum.

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-II

COURSE CODE: DDSE201

COURSE NAME: ENGINEERING MATHEMATICS-II

Course Objectives:

- To build advanced mathematical skills for engineering problem-solving
- To apply differential equations to model mechanical systems
- To understand vector algebra and its applications in mechanics
- To develop skills in numerical methods for engineering computations
- To prepare mathematical foundation for subjects like Mechanics, Thermodynamics, and Machine Design

Course Outcomes: At the end of the course students shall be able to

CO1	**Solve** ordinary differential equations relevant to engineering systems
CO2	**Apply** vector algebra to analyze forces and motions in 3D space
C03	**Perform** numerical computations using interpolation, differentiation, and integration methods
C04	**Analyze** data using probability distributions and statistical methods

Unit	Content	Credit	Weightage
I	**Differential Equations** **Topics:** • First order differential equations: variable separable, homogeneous, exact • Linear differential equations of first order • Applications: Newton's law of cooling, growth and decay, simple circuits • Second order linear differential equations with constant coefficients • Complementary function and particular integral methods • **Applications:** Spring-mass systems, electrical circuits, vibration analysis	1	25%
II	**Vector Algebra & 3D Geometry** **Topics:** • Vectors: dot product, cross product, scalar triple product • Vector differentiation • Gradient, divergence, curl (basic concepts) • Lines and planes in 3D space • Direction cosines and ratios • **Applications:** Force analysis, moment of force, work done by force	1	25%
III	**Numerical Methods** **Topics:**	1	25%

	• Solution of algebraic equations: Bisection method, Newton-Raphson method • Interpolation: Newton's forward and backward difference formulas • Numerical differentiation and integration • Trapezoidal rule and Simpson's rules • **Applications:** Root finding in design equations, area calculations, data analysis		
IV	**Probability & Laplace Transforms** **Topics:** • **Probability:** Basic concepts, addition and multiplication theorems • Random variables, probability distributions (Binomial, Poisson, Normal) • Mean, variance, standard deviation • **Laplace Transforms:** Definition, basic transforms • Properties: linearity, shifting, differentiation • Application to differential equations • **Applications:** Quality control, reliability analysis, system dynamics	1	25%

Textbooks:

- Primary: *Higher Engineering Mathematics* — B.S. Grewal
- Primary: *Engineering Mathematics* — NP Bali & Dr. Manish Goyal

Reference books:

- *Advanced Engineering Mathematics* — H.K. Das
- *Numerical Methods* — S.S. Sastry
- *Probability and Statistics for Engineers* — Dr. J. Ravichandran
- *Mathematical Methods* — B.V. Ramana

Online Platforms:

- NPTEL Videos: "Differential Equations for Engineers"
- Khan Academy: Complete probability and statistics
- MIT OCW: "Single Variable Calculus" continuation
  Coursera: "Introduction to Numerical Methods"

COURSE CODE: DDSE202

COURSE NAME: DATA STRUCTURES AND ALGORITHMS

Course Objectives:

- To introduce fundamental data structures and their implementation.
- To develop problem-solving skills using appropriate data structures.
- To analyze algorithm efficiency using time and space complexity.
- To prepare students for advanced topics, machine learning, and software development.
- To enable students to implement and apply data structures in real-world scenarios.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** the concepts of arrays, linked lists, stacks, and queues.
CO2	**Implement** trees and graphs for hierarchical and network data representation.
C03	**Apply** sorting and searching algorithms to solve computational

	problems.
C04	**Analyze** algorithm efficiency and select appropriate data structures for given problems.

Unit	Content	Credit	Weightage
I	**Introduction to Data Structures & Linear Lists** Introduction: Data types, ADTs, time and space complexity - Arrays: 1D, 2D, operations, applications - Linked Lists: Singly, doubly, circular linked lists - Stacks: Array and linked list implementation, applications - Queues: Simple, circular, priority queues - **Applications:** Expression evaluation, job scheduling, memory management	1	25%
II	**Trees & Hierarchical Data Structures** Trees: Terminology, binary trees, traversal (inorder, preorder, postorder) - Binary Search Trees (BST): Insertion, deletion, searching - AVL Trees: Rotations, balancing - Heaps: Min-heap, max-heap, heap operations - **Applications:** File systems, database indexing, priority queues	1	25%
III	**Graphs & Hashing** Graphs: Terminology, representation (adjacency matrix, list) - Graph traversals: BFS, DFS - Hashing: Hash functions, collision resolution (chaining, open addressing) - **Applications:** Social networks, GPS navigation, spell checking	1	25%
IV	**Sorting, Searching & Algorithm Design** Sorting algorithms: Bubble, selection, insertion, merge, quick, heap sort - Searching algorithms: Linear, binary, interpolation search - Algorithm design techniques: Greedy, divide and conquer, dynamic programming (intro) - **Applications:** Data analysis, AI search algorithms, database query optimization	1	25%

Textbooks:

- *Data Structures and Algorithms in C* — Reema Thareja
- *Data Structures Using C* — Aaron M. Tenenbaum

Reference books:

- *Introduction to Algorithms* — Thomas H. Cormen et al.
- *Data Structures and Algorithm Analysis in C* — Mark Allen Weiss
- *The Algorithm Design Manual* — Steven S. Skiena
- *Data Structures Through C* — Yashavant Kanetkar

Online Platforms:

- NPTEL:
  1. *Data Structures and Algorithms* by Prof. Naveen Garg (IIT Delhi)

2. *Programming, Data Structures and Algorithms* by IIT Madras

- Coursera:
  1. *Data Structures and Algorithms* by University of California, San Diego
  2. *Algorithms Specialization* by Stanford University

PRACTICAL LIST:

- Section A: Linear Data Structures
- Implement stack using array and linked list with push, pop, display.
- Implement queue using array and linked list with enqueue, dequeue, display.
- Implement circular queue with basic operations.
- Implement singly linked list with insertion, deletion, traversal.
- Implement doubly linked list with insertion, deletion, display.
- Implement polynomial addition using linked list.
- Section B: Trees & Heaps
- Implement binary tree traversal (inorder, preorder, postorder) recursively.
- Implement binary search tree with insertion, deletion, searching.
- Implement heap (max-heap) with insertion and deletion.
- Implement AVL tree with rotations (LL, RR, LR, RL).
- Implement expression tree from postfix expression and evaluate it.
- Section C: Graphs & Hashing
- Implement graph using adjacency matrix and perform BFS and DFS.
- Implement graph using adjacency list and perform BFS and DFS.
- Implement hashing with separate chaining for collision resolution.
- Implement hashing with linear probing for collision resolution.
- Find shortest path in unweighted graph using BFS.
- Section D: Sorting & Searching Algorithms
- Implement bubble sort, selection sort, and insertion sort.
- Implement merge sort and quick sort.
- Implement heap sort.
- Implement binary search recursively and iteratively.
- Implement linear search and count occurrences in an array.
- Implement a menu-driven program to compare sorting algorithm performance.

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

COURSE CODE: DDSE203

COURSE NAME: DATABASE MANAGEMENT SYSTEMS

Course Objectives:

- To introduce fundamental concepts of databases, data models, and DBMS architecture.
- To develop skills in designing databases using ER modeling and normalization.
- To enable students to write SQL queries for data definition, manipulation, and control.
- To understand transaction management, concurrency control, and database security.
- To prepare students for real-world database applications in AI and data science.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** DBMS architecture, data models, and ER diagrams.
CO2	**Design** normalized relational databases using ER modeling and normalization.
C03	**Write** SQL queries for data definition, manipulation, and retrieval.
C04	**Analyze** transaction management, concurrency control, and database security mechanisms.

Unit	Content	Credit	Weightage
I	**Introduction to DBMS & Data Modeling** Database concepts: Data, database, DBMS, advantages - DBMS architecture: 3-tier architecture, data independence - Data models: Hierarchical, network, relational, ER model - ER diagrams: Entities, attributes, relationships, keys - **Applications:** Banking, inventory, student management systems	1	25%
II	**Relational Database Design & SQL** Relational model: Relations, tuples, attributes, keys - Relational algebra: Operations, selection, projection, join - SQL: DDL, DML, DCL commands - SQL queries: SELECT, WHERE, GROUP BY, HAVING, ORDER BY - Joins: Inner, outer, self, cross joins - **Applications:** Querying real databases, report generation	1	25%
III	**Normalization & Advanced SQL** Normalization: 1NF, 2NF, 3NF, BCNF, functional dependencies - Advanced SQL: Subqueries, views, indexes, sequences - PL/SQL basics: Cursors, procedures, functions, triggers - **Applications:** Data integrity, performance tuning, automation	1	25%
IV	**Transaction Management & Security** Transaction concepts: ACID properties, states - Concurrency control: Locks, two-phase locking, deadlock - Database recovery: Log-based recovery, checkpoints - Database security: Authentication, authorization, encryption - **Applications:** Banking transactions, multi-user systems,	1	25%

	secure databases		

Textbooks:

- *Database System Concepts* — Abraham Silberschatz, Henry F. Korth, S. Sudarshan
- *Fundamentals of Database Systems* — Ramez Elmasri, Shamkant B. Navathe

Reference books:

- *SQL: The Complete Reference* — James R. Groff, Paul N. Weinberg
- *Database Management Systems* — Raghu Ramakrishnan, Johannes Gehrke
- *An Introduction to Database Systems* — C.J. Date
- *Oracle PL/SQL Programming* — Steven Feuerstein

Online Platforms:

- NPTEL:
  1. *Database Management System* by Prof. P.K. Biswas (IIT Kharagpur)
  2. *Introduction to Database Systems* by IIT Madras
- Coursera:
  1. *Databases and SQL for Data Science* by IBM
  2. *SQL for Data Science* by University of California, Davis

PRACTICAL LIST:

Section A: Database Design & ER Modeling

1. Draw ER diagrams for:
   - Library Management System
   - Hospital Management System
   - Online Shopping System
2. Convert ER diagrams into relational schemas.
3. Create tables using DDL commands with primary keys, foreign keys, and constraints.

Section B: SQL Queries – Basic to Intermediate

4. Create a database for Student Management System with tables: Student, Course, Enrollment.
5. Insert sample data into all tables.
6. Write SQL queries to:
   - Retrieve all students enrolled in a specific course.
   - Find the average marks of students.
   - List students with marks above average.
   - Count number of students per course.
7. Implement different types of joins (inner, left, right, full) on the database.

Section C: Advanced SQL & PL/SQL

8. Create views for:
   - Students with marks > 75%
   - Course-wise student count
9. Write SQL queries using:
   - Subqueries (nested, correlated)
   - GROUP BY, HAVING, ORDER BY
10. Create indexes on frequently queried columns.
11. Write a PL/SQL procedure to calculate total marks of a student.
12. Create a trigger to log changes in the Student table.

Section D: Normalization & Transaction Control

13. Take an unnormalized table and normalize it to 3NF.
14. Demonstrate transaction control commands:

- COMMIT, ROLLBACK, SAVEPOINT

15. Simulate concurrent transactions and demonstrate locking.
16. Implement user roles and grant/revoke permissions.
17. Perform backup and recovery operations on a sample database.

COURSE CODE: DDSE204

COURSE NAME: WEB TECHNOLOGIES

Course Objectives:

- To introduce the fundamentals of web technologies, internet protocols, and web architecture.
- To develop skills in designing responsive web pages using HTML, CSS, and JavaScript.
- To enable students to create dynamic web applications using front-end and back-end technologies.
- To understand web APIs, AJAX, and integration with databases.
- To prepare students for building web-based AI applications and interactive dashboards.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** web architecture, internet protocols, and client-server models.
CO2	**Design** responsive and interactive web pages using HTML, CSS, and JavaScript.
C03	**Develop** dynamic web applications using front-end frameworks and back-end scripting.
C04	**Integrate** databases and web APIs to create data-driven web applications.

Unit	Content	Credit	Weightage
I	**Web Fundamentals & HTML5** Internet basics: WWW, HTTP/HTTPS, DNS, web browsers - Web architecture: Client-server model, static vs dynamic websites - HTML5: Structure, tags, forms, multimedia, semantic elements - HTML5 APIs: Geolocation, local storage, canvas basics - **Applications:** Portfolio websites, forms, simple web pages	1	25%
II	**CSS3 & Responsive Web Design** CSS3: Selectors, box model, positioning, flexbox, grid - Responsive design: Media queries, mobile-first approach - CSS frameworks: Bootstrap basics - Animations and transitions - **Applications:** Responsive layouts, navigation bars, styled forms	1	25%
III	**JavaScript & Front-End Development** JavaScript basics: Syntax, DOM manipulation, events - ES6 features: let/const, arrow functions, promises - AJAX and Fetch API for async requests - Front-end frameworks: Introduction to React.js/Vue.js - **Applications:** Interactive UI, form validation, dynamic content loading	1	25%
IV	**Back-End Basics & Web APIs**	1	25%

	Back-end introduction: Node.js, Express.js basics - RESTful APIs: CRUD operations, JSON handling - Database integration: MongoDB/MySQL with Node.js - Authentication: JWT, session-based auth basics - **Applications:** Simple web apps, API integration, user authentication		

Textbooks:

- *Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML and AJAX* — Uttam K. Roy
- *HTML & CSS: Design and Build Websites* — Jon Duckett

Reference books:

- *Eloquent JavaScript* — Marijn Haverbeke
- *Learning PHP, MySQL & JavaScript* — Robin Nixon
- *Node.js Design Patterns* — Mario Casciaro
- *React: Up & Running* — Stoyan Stefanov

Online Platforms:

- NPTEL:
  1. *Web Technologies* by Prof. D. Janakiram (IIT Madras)
  2. *Introduction to Modern Application Development* by IIT Madras
- Coursera:
  1. *HTML, CSS, and JavaScript for Web Developers* by Johns Hopkins University
  2. *Web Design for Everybody* by University of Michigan

PRACTICAL LIST:

Section A: HTML5 & CSS3

1. Create a personal portfolio website using HTML5 and CSS3.
2. Design a responsive registration form with validation using HTML5 form elements.
3. Build a webpage layout using CSS Grid and Flexbox.
4. Implement a navigation bar with dropdown menus using CSS.
5. Create a photo gallery with hover effects and transitions.

Section B: JavaScript & DOM Manipulation

6. Create a calculator using JavaScript.
7. Build a to-do list application with add, edit, delete, and mark-as-done features.
8. Develop a digital clock with date and time display.
9. Implement form validation (email, password, phone) using JavaScript.
10. Create a slideshow/carousel using JavaScript and CSS.

Section C: Responsive Design & Bootstrap

11. Convert a given webpage into a mobile-responsive layout using media queries.
12. Design a landing page using Bootstrap 5 (navbar, cards, modal, forms).
13. Create a responsive dashboard layout with Bootstrap grid system.
14. Build a blog template with Bootstrap components.

Section D: Back-End Integration & APIs

15. Create a simple REST API using Node.js and Express that returns JSON data.
16. Build a weather app using a public API (e.g., OpenWeatherMap) with Fetch/AJAX.
17. Develop a note-taking app with local storage (CRUD operations).

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-III

COURSE CODE: DDSE301

COURSE NAME: STATISTICS FOR DATA SCIENCE

Course Objectives:

- To introduce foundational statistical concepts and their applications in data science.
- To develop skills in descriptive statistics, probability theory, and inferential statistics.
- To enable students to apply statistical methods for data exploration, hypothesis testing, and predictive modeling.
- To prepare students for advanced data science topics such as machine learning and data analytics.
- To foster critical thinking and data-driven decision-making abilities.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** fundamental statistical concepts, measures of central tendency, and dispersion.
CO2	**Apply** probability distributions and sampling techniques to real-world data.
C03	**Perform** hypothesis testing, confidence interval estimation, and regression analysis.
C04	**Analyze** and visualize data using statistical software and interpret results for decision-making.

Unit	Content	Credit	Weightage
I	**Descriptive Statistics & Data Visualization** **Topics:** • Introduction to statistics in data science • Measures of central tendency: mean, median, mode • Measures of dispersion: range, variance, standard deviation, IQR • Skewness and kurtosis • Data visualization: histograms, box plots, scatter plots, bar charts	1	25%
II	**Probability Theory & Distributions** **Topics:** • Basic probability concepts, events, sample space • Conditional probability, Bayes' theorem • Random variables: discrete and continuous • Probability distributions: Binomial, Poisson, Normal, Uniform • Central Limit Theorem	1	25%
III	**Inferential Statistics & Hypothesis Testing** **Topics:** • Sampling techniques and sampling distributions • Estimation: point and interval estimates • Hypothesis testing: null and alternative hypotheses, p-value, significance level • Z-test, t-test (one-sample, two-sample) • Chi-square test for independence	1	25%

IV	**Regression & Correlation Analysis** **Topics:** • Correlation: Pearson, Spearman • Simple linear regression: model, slope, intercept, R-squared • Residual analysis and assumptions • Introduction to multiple linear regression • Model evaluation metrics: MAE, MSE, RMSE	1	25%

Textbooks:

Primary: *Statistics for Data Science* — James, Witten, Hastie, Tibshirani

Primary: *Practical Statistics for Data Scientists* — Peter Bruce & Andrew Bruce

Reference books:

- *Introduction to Probability and Statistics for Engineers and Scientists* — Sheldon M. Ross
- *Naked Statistics: Stripping the Dread from the Data* — Charles Wheelan
- *Statistics in Plain English* — Timothy C. Urdan
- *The Art of Statistics: Learning from Data* — David Spiegelhalter

Online Platforms:

- NPTEL:
  1. *Probability and Statistics* by IIT Kharagpur
  2. *Data Science for Engineers* by IIT Madras
- Coursera:
  1. *Statistics with Python* by University of Michigan
  2. *Data Science Math Skills* by Duke University

PRACTICAL LIST:

- Load a dataset (e.g., student marks) and compute mean, median, mode.
- Calculate variance and standard deviation for a given dataset.
- Plot histogram and box plot to visualize distribution and outliers.
- Create scatter plot to analyze relationship between two variables.
- Simulate coin tosses and die rolls to compute empirical probabilities.
- Implement Bayes' theorem for a simple classification problem
- Generate and visualize Binomial and Poisson distributions.
- Simulate Central Limit Theorem using random sampling.
- Perform one-sample t-test to compare sample mean with population mean.
- Conduct two-sample t-test to compare means of two groups.
- Perform Chi-square test on categorical data (e.g., survey results).
- Calculate confidence intervals for population mean and proportion.
- Calculate Pearson and Spearman correlation coefficients.
- Implement simple linear regression using least squares method.
- Plot regression line and residuals to check assumptions.
- Build a multiple linear regression model and interpret coefficients.

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. **RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956**

COURSE CODE: DDSE302

COURSE NAME: OBJECT ORIENTED PROGRAMMING C++

Course Objectives:

- To introduce the principles and concepts of Object-Oriented Programming (OOP).
- To develop skills in designing, implementing, and testing C++ programs using OOP features.
- To enable students to understand and apply encapsulation, inheritance, polymorphism, and abstraction.
- To prepare a foundation for advanced programming, software development, and system design.
- To foster good programming practices and problem-solving using OOP.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** the fundamentals of OOP and the structure of C++ programs.
CO2	**Implement** classes, objects, constructors, and destructors in C++.
C03	**Apply** inheritance, polymorphism, and operator overloading in program design.
C04	**Develop** C++ programs using file handling, templates, and exception handling.

Unit	Content	Credit	Weightage
I	**Introduction to OOP & C++ Basics** • Introduction to OOP: Concepts, advantages over procedural programming • C++ basics: Structure of a C++ program, iostream, namespace • Data types, operators, and control structures in C++ • Functions: Function overloading, default arguments, inline functions • Introduction to classes and objects (basic) • **Applications**: Simple calculator, menu-driven programs	1	25%
II	**Classes, Objects & Constructors** • Classes and objects: Declaration, definition, access specifiers • Member functions: Inside and outside class definition • Constructors: Default, parameterized, copy constructors • Destructors • this pointer • Friend functions and friend classes • **Applications**: Student record system, bank account management	1	25%
III	**Inheritance & Polymorphism** • Inheritance: Types (single, multiple, multilevel, hierarchical, hybrid)	1	25%

	• Access specifiers in inheritance: public, protected, private • Polymorphism: Compile-time and run-time polymorphism • Function overriding and virtual functions • Abstract classes and pure virtual functions • Operator overloading: Unary and binary operators • **Applications**: Payroll system, shape hierarchy, library management		
IV	**Advanced C++ Features** • File handling in C++: if stream, of stream, f stream • Templates: Function templates and class templates • Exception handling: try, catch, throw • Standard Template Library (STL) basics: vector, list, map • Dynamic memory allocation: new and delete operators • **Applications**: File-based data storage, generic programming, error handling	1	25%

Textbooks:

- *Object-Oriented Programming with C++* – E. Balagurusamy
- *Let Us C++* – Yashwant Kanetkar

Reference books:

- *The C++ Programming Language* – Bjarne Stroustrup
- *C++: The Complete Reference* – Herbert Schildt
- *Object-Oriented Programming in C++* – Robert Lafore
- *Programming: Principles and Practice Using C++* – Bjarne Stroustrup

Online Platforms:

- NPTEL
  - *Programming in C++* by Prof. P. P. Chakraborty (IIT Kharagpur)
  - *Object-Oriented Programming in C++* by IIT Bombay
- Coursera
  - *Object-Oriented Programming in C++* by University of London
  - *C++ For C Programmers* by University of California, Santa Cruz

PRACTICAL LIST:

Module I: C++ Basics & Functions

1. Write a C++ program to find the sum and average of three numbers.
2. Write a program to swap two numbers using call by reference.
3. Write a program to check whether a number is prime or not.
4. Write a program to display the Fibonacci series using recursion.
5. Write a program to demonstrate function overloading (add numbers and concatenate strings).
6. Write a program to find the factorial of a number using inline function.

Module II: Classes & Objects

7. Write a program to create a class Student with data members roll_no, name, marks and member functions to input and display data.
8. Write a program to implement a class Circle with member functions to calculate area and circumference.
9. Write a program to demonstrate the use of constructors (default, parameterized, copy).
10. Write a program to show the use of this pointer.
11. Write a program to demonstrate friend function accessing private members of two different

classes.

Module III: Inheritance & Polymorphism

12. Write a program to implement single inheritance: class Employee (base) and Manager (derived).
13. Write a program to demonstrate multiple inheritance.
14. Write a program to implement function overriding and use of virtual functions.
15. Write a program to create an abstract class Shape with pure virtual function area().
16. Write a program to overload + operator to add two complex numbers.

Module IV: Advanced C++ Features

17. Write a program to read from a file and count the number of words.
18. Write a program to write student records into a file and display them.
19. Write a program to demonstrate function template for swapping two values of any data type.
20. Write a program to implement exception handling for division by zero.

COURSE CODE: DDSE303

COURSE NAME: COMPUTER NETWORKS AND SECURITY

Course Objectives:

- To introduce fundamental concepts of computer networks, architectures, and protocols.
- To develop understanding of data communication, network models, and internetworking.
- To enable students to configure, troubleshoot, and secure basic network setups.
- To introduce cybersecurity principles, threats, and defense mechanisms.
- To prepare students for network-related aspects in data science, IoT, and cloud computing.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** network models, topologies, protocols, and transmission media.
CO2	**Configure** network devices, IP addressing, and basic network services.
C03	**Analyze** network traffic and implement basic security measures.
C04	**Apply** cybersecurity principles to protect networks and data.

Unit	Content	Credit	Weightage

I	**Introduction to Networks & Physical Layer**
	• **Network Fundamentals:**
	  ○ Components: NIC, Hub, Switch, Router, Gateway
	  ○ Network Types: LAN, MAN, WAN, PAN, VPN
	  ○ Topologies: Bus, Star, Ring, Mesh, Hybrid
	• **Network Models:**
	  ○ OSI Model (7 Layers)
	  ○ TCP/IP Model (4 Layers)
	  ○ Comparison and functions of each layer
	• **Transmission Media:**
	  ○ Guided: Twisted Pair, Coaxial, Fiber Optic
	  ○ Unguided: Radio Waves, Microwave, Infrared
		1	25%

	• **Network Devices & Cabling:**
	  ○ Ethernet Cables (Straight-through, Crossover, Rollover)
	  ○ Wireless Standards (802.11 a/b/g/n/ac)		

II	**Data Link Layer & Network Layer**
	• **Data Link Layer Functions:**
	  ○ Framing, Error Detection (CRC), Flow Control
	  ○ MAC Protocols: CSMA/CD, CSMA/CA
	• **Ethernet & Switching:**
	  ○ Ethernet Frames, MAC Addressing
	  ○ VLANs, STP (Spanning Tree Protocol)
	• **Network Layer:**
	  ○ IP Addressing: IPv4 Classes, Subnetting, CIDR
	  ○ IPv6 Basics
	  ○ Routing: Static vs Dynamic (RIP, OSPF basics)
	  ○ ARP, ICMP, DHCP
		1	25%

III	**Transport & Application Layers**
	• **Transport Layer Protocols:**
	  ○ TCP: Connection-oriented, 3-way handshake
	  ○ UDP: Connectionless
	  ○ Ports, Sockets, Multiplexing
	• **Application Layer Protocols:**
	  ○ HTTP/HTTPS, DNS, SMTP, FTP, Telnet, SSH
	  ○ Email Systems, Web Services
	• **Network Services:**
	  ○ Client-Server Architecture
	  ○ Peer-to-Peer Networks
	  ○ Cloud Networking Basics
		1	25%

IV	**Network Security & Cybersecurity**
	• **Security Fundamentals:**
	  ○ CIA Triad (Confidentiality, Integrity, Availability)
	  ○ Threats: Malware, DoS, Phishing, Social Engineering
	• **Network Defense Mechanisms:**
	  ○ Firewalls (Packet Filtering, Stateful)
	  ○ IDS/IPS (Intrusion Detection/Prevention Systems)
	  ○ VPNs, Encryption (SSL/TLS)
	• **Access Control & Authentication:**
	  ○ AAA Model (Authentication, Authorization, Accounting)
	  ○ RADIUS, TACACS+
	  ○ Multi-factor Authentication
	• **Ethical Hacking Basics:**
		1	25%

	○ Footprinting, Scanning, Enumeration ○ Penetration Testing Phases ○ Legal & Ethical Considerations		

Textbooks:

1. Primary: *Data Communications and Networking* — Behrouz A. Forouzan
2. Primary: *Computer Networks* — Andrew S. Tanenbaum

Reference books:

- *Network Security Essentials* — William Stallings
- *TCP/IP Illustrated* — W. Richard Stevens
- *Computer Networking: A Top-Down Approach* — Kurose & Ross
- *Cybersecurity Fundamentals* — Rick Howard

Online Platforms:

- **NPTEL:**
  1. *Computer Networks* by Prof. S. Ghosh (IIT Kharagpur)
  2. *Introduction to Cyber Security* by IIT Madras
- **Coursera:**
  1. *The Bits and Bytes of Computer Networking* (Google)
  2. *Introduction to Cybersecurity Tools & Cyber Attacks* (IBM)

PRACTICAL LIST:

Module 1: Network Fundamentals & Configuration

1. Network Topology Design using Cisco Packet Tracer
2. IP Addressing & Subnetting exercises and configuration
3. Basic Router Configuration (hostname, passwords, interfaces)
4. VLAN Configuration on switches

Module 2: Network Services & Protocols

5. DHCP Server Configuration on router/VM
6. DNS Server Setup and hostname resolution
7. FTP & Web Server Configuration
8. Protocol Analysis using Wireshark (HTTP, DNS, ICMP)

Module 3: Routing & Security Basics

9. Static & Dynamic Routing (RIP) configuration
10. Firewall Configuration (Windows/Linux)
11. Wireless Network Setup with WPA2 security
12. Network Scanning with Nmap

Module 4: Cybersecurity & Defense

13. Password Security & hash analysis
14. VPN Configuration (OpenVPN/Wire Guard)
15. Intrusion Detection setup with Snort (demo)
16. Security Audit of a small network

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-IV

COURSE CODE: DDSE401

COURSE NAME: INTRODUCTION TO MACHINE LEARNING

Course Objectives:

- To introduce fundamental concepts, types, and applications of machine learning.
- To develop skills in data preprocessing, feature engineering, and model evaluation.
- To enable students to implement and evaluate supervised and unsupervised learning algorithms.
- To prepare students for advanced ML topics and real-world AI applications.
- To foster ethical awareness and responsible use of ML systems.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** types of ML, data preprocessing steps, and model evaluation metrics.
CO2	**Implement** supervised learning algorithms for classification and regression.
C03	**Apply** unsupervised learning techniques for clustering and dimensionality reduction.
C04	**Evaluate** model performance and interpret results using real datasets.

Unit	Content	Credit	Weightage
I	**Introduction to ML & Data Preprocessing** What is ML? Types: Supervised, Unsupervised, Reinforcement Learning - ML workflow: Problem definition, data collection, preprocessing, modeling, evaluation - Data preprocessing: Handling missing values, encoding, normalization, train-test split - Feature engineering and selection - **Applications:** Spam detection, recommendation systems, customer segmentation	1	25%
II	**Supervised Learning – Regression & Classification** Linear Regression: Simple and multiple, assumptions, evaluation metrics (MSE, R^2) - Logistic Regression: Binary classification, sigmoid function, confusion matrix - Decision Trees: Splitting criteria, pruning, interpretation - Model evaluation: Accuracy, precision, recall, F1-score, ROC-AUC - **Applications:** House price prediction, customer churn, medical diagnosis	1	25%
III	**Advanced Supervised Learning & Ensemble Methods** k-Nearest Neighbors (k-NN): Distance metrics, k-value selection - Support Vector Machines (SVM): Linear and kernel SVM, hyperparameters	1	25%

	- Ensemble methods: Bagging (Random Forest), Boosting (AdaBoost, Gradient Boosting) - Hyperparameter tuning: Grid Search, Random Search - **Applications:** Image classification, fraud detection, credit scoring		
IV	**Unsupervised Learning & Model Deployment Basics** Clustering: k-Means, Hierarchical clustering, evaluation (silhouette score) - Dimensionality reduction: PCA, t-SNE basics - Introduction to model deployment: Saving models (pickle, joblib), Flask basics - Ethical considerations: Bias, fairness, interpretability in ML - **Applications:** Market segmentation, anomaly detection, visualization	1	25%

Textbooks:

- *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* — Aurélien Géron
- *Introduction to Machine Learning with Python* — Andreas C. Müller & Sarah Guido

Reference books:

- *Pattern Recognition and Machine Learning* — Christopher M. Bishop
- *Machine Learning Yearning* — Andrew Ng
- *The Hundred-Page Machine Learning Book* — Andriy Burkov
- *Python Machine Learning* — Sebastian Raschka & Vahid Mirjalili

Online Platforms:

- NPTEL:
  - *Introduction to Machine Learning* by Prof. Balaraman Ravindran (IIT Madras)
  - *Machine Learning* by IIT Kharagpur
- Coursera:
  - *Machine Learning* by Andrew Ng (Stanford University)
  - *Applied Data Science with Python* by University of Michigan

PRACTICAL LIST:

Section A: Data Preprocessing & EDA

1. Load a dataset (e.g., Titanic, Iris) and perform exploratory data analysis (EDA).
2. Handle missing values and encode categorical variables.
3. Normalize/standardize features and split data into train/test sets.
4. Visualize feature distributions and correlations using seaborn/matplotlib.

Section B: Supervised Learning – Regression & Classification

5. Implement Simple Linear Regression to predict housing prices (Boston Housing dataset).
6. Implement Logistic Regression for binary classification (Titanic survival prediction).
7. Build and visualize a Decision Tree classifier and evaluate using confusion matrix.
8. Compare model performance (accuracy, precision, recall) for different classifiers.

Section C: Advanced Supervised Learning

9. Implement k-NN classifier for Iris dataset and find optimal k using cross-validation.
10. Train an SVM classifier with different kernels (linear, RBF) and compare results.
11. Implement Random Forest classifier and analyze feature importance.
12. Perform hyperparameter tuning using Grid Search CV for any classifier.

Section D: Unsupervised Learning & Mini-Project

13. Apply k-Means clustering for customer segmentation (Mall Customer dataset).
14. Perform PCA for dimensionality reduction and visualize clusters in 2D.
15. Build an end-to-end ML pipeline: from data loading to model deployment using Flask.

Mini-Project: Develop a complete ML model (e.g., spam classifier, sales predictor) with a detailed report.

COURSE CODE: DDSE402

COURSE NAME: DATA VISUALIZATION AND BI TOOLS

Course Objectives:

- To introduce the theory and psychology behind effective visual perception.
- To equip students with skills to use leading visualization tools and libraries.
- To enable creation of dashboards for real-time data exploration.
- To develop the ability to choose appropriate chart types for different data and tasks.
- To foster ethical considerations in data representation.

Course Outcomes: At the end of the course students shall be able to

CO1	Understand fundamental principles and importance of data visualization in data science.
CO2	Design and create static visualizations using Python libraries (Matplotlib, Seaborn).
C03	Develop interactive dashboards and web-based visualizations using Plotly, Dash, or Tableau.
C04	Apply visualization best practices to tell compelling data stories and support decision-making.

Unit	Content	Credit	Weightage
I	**Foundations of Data Visualization** • Introduction: What is data visualization? History and importance. • Human visual perception & cognitive principles (Gestalt principles, pre-attentive attributes). • Data types and visualization tasks: categorical, numerical, temporal, hierarchical, network. • The visualization pipeline: from data to insight. • Principles of effective visual design: color theory, scales, axes, labeling, accessibility. • Common chart types and their appropriate use cases. • Ethical issues: misleading visuals, bias, and transparency.	1	25%
II	**Static Visualization with Python** • Introduction to Python visualization ecosystem: Matplotlib, Seaborn, Pandas plotting. • Matplotlib architecture: object-oriented interface vs. pyplot. • Customizing plots: titles, labels, legends, annotations, themes.	1	25%

	• Statistical plotting with Seaborn: distribution plots, categorical plots, regression plots. • Multi-plot layouts: subplots, grids, and faceting. • Geospatial visualization basics with Geopandas/Plotly. • Exporting visualizations for reports and publications.		
III	**Interactive Visualization & Dashboards** **Topics:** • Introduction to interactive visualization: advantages and tools. • Interactive web-based plots with Plotly and Plotly Express. • Introduction to Dash framework: layout, callbacks, components. • Building a simple dashboard with Dash. • Tableau Public: connecting data, creating worksheets, dashboards, stories. • Dashboard design principles: layout, interactivity, user experience. • Publishing and sharing interactive visualizations.	1	25%
IV	**Advanced Topics & Storytelling** **Topics:** • Visualization for big data: aggregation, sampling, and techniques. • Network graphs and tree maps. • Time-series and animated visualizations. • Data storytelling: crafting narratives with data, combining visuals and text. • Dashboard case studies from business, healthcare, finance. • Critique and redesign of flawed visualizations. • Future trends: AR/VR in visualization, real-time dashboards, AI-driven visualization.	1	25%

Textbooks:

- "Storytelling with Data" – Cole Nussbaumer Knaflic
- "The Visual Display of Quantitative Information" – Edward R. Tufte
- "Python Data Science Handbook" (Ch. 4: Visualization with Matplotlib) – Jake VanderPlas
- "Interactive Data Visualization for the Web" – Scott Murray

Reference books:

- "Fundamentals of Data Visualization" – Claus O. Wilke
- "Data Visualization: A Practical Introduction" – Kieran Healy
- "Plotly Dash Tutorial & Documentation" (online)
- "Tableau Desktop Pocket Reference" – Ryan Sleeper

PRACTICAL LIST:

- Principles & Critique
    1. Find misleading charts in media and redesign them ethically.
- Matplotlib Basics
    1. Create line, bar, scatter, histogram with custom styling.

- Seaborn Statistical Plots
    1. Visualize distributions, box plots, violin plots, and pair plots.
- Multi-Plot Dashboard (Static)
    1. Use subplots to display related datasets in one figure.
- Plotly Interactive Charts
    1. Build interactive scatter, line, and bar charts with hover tools.
- Geospatial Map
    1. Plot data on a map using latitude/longitude (GeoPandas/Plotly).
- Dash Dashboard
    1. Create a two-component dashboard with dropdown/input controls.
- Tableau Workbook
    1. Connect CSV, create calculated fields, build a dashboard with filters.
- Network Graph
    1. Visualize relationships in a small network dataset.
- Time-Series Animation
    1. Animate a time-series plot using Plotly or Matplotlib animation.
- Time-Series Animation
    1. Animate a time-series plot using Plotly or Matplotlib animation.
- Data Story Presentation
    1. Prepare a slide deck with visualizations to tell a story from a given dataset.
- Final Project
    1. Choose a dataset, design and implement a full interactive dashboard with narrative.

COURSE CODE: DDSE403

COURSE NAME: BIGDATA BASICS (HADOOP, SHARK)

Course Objectives:

- To introduce the Big Data paradigm and its relevance in modern data science
- To develop skills in distributed computing using Hadoop ecosystem
- To enable efficient data processing using Apache Spark for analytics
- To provide hands-on experience with cluster computing frameworks
- To prepare students for industry-standard Big Data processing scenarios

Course Outcomes: At the end of the course students shall be able to

CO1	Understand Big Data characteristics, challenges, and ecosystem components
CO2	Design and implement distributed data processing using Hadoop MapReduce
C03	Process and analyze large datasets using Apache Spark (RDDs and Data Frames)
C04	Work with Hadoop Distributed File System (HDFS) for storage operations

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

Unit	Content	Credit	Weightage
I	**Introduction to Big Data & Hadoop Ecosystem** • Understanding Big Data: Volume, Velocity, Variety, Veracity (4Vs) • Evolution from traditional systems to distributed computing • Hadoop Architecture: HDFS, YARN, MapReduce • Hadoop ecosystem overview: Hive, Pig, HBase, Sqoop, Flume • Hadoop cluster architecture: Name Node, Data Node, Resource Manager • Setting up Hadoop: Single node and pseudo-distributed mode • Introduction to cloud Big Data services (AWS EMR, Google Data proc)	1	25%
II	**Hadoop Distributed File System & MapReduce** • **HDFS Architecture:** Blocks, replication, rack awareness • HDFS commands for file operations (CLI and Web UI) • **MapReduce Programming Model:** Mapper, Reducer, Combiner, Partitioner • Developing MapReduce programs in Java/Python (Hadoop Streaming) • Word Count problem and other case studies • Input Formats and Output Formats • Optimization techniques: Compression, speculative execution • Limitations of MapReduce and need for Spark	1	25%
III	**Apache Spark Fundamentals** • Introduction to Apache Spark: Why Spark over MapReduce • Spark architecture: Driver, Executor, Cluster Manager • Spark deployments: Standalone, YARN, Mesos • **Resilient Distributed Datasets (RDDs):** Creation, transformations, actions • **Spark Operations:** Narrow vs wide transformations, persistence • **Spark SQL and Data Frames:** Structured data processing • **Dataset API:** Type-safe distributed collections • **Spark Shell:** Interactive analysis with Scala/Python	1	25%
IV	**Spark Applications & Big Data Tools Integration** • Data processing pipelines with Spark • Connecting Spark with HDFS, Hive, and HBase • **Spark for ETL operations:** Data cleaning, transformation, loading • Introduction to Spark MLlib for machine learning • Real-time processing concepts with Spark Streaming (micro-batching)	1	25%

	• **Performance tuning:** Partitioning, caching, broadcast variables • **Project:** Complete data pipeline from ingestion to analysis • Industry use cases: Clickstream analysis, log processing, recommendation systems		

Textbooks:

- "Hadoop: The Definitive Guide" (4th Edition) – Tom White
- "Learning Spark: Lightning-Fast Data Analytics" (2nd Edition) – Jules Damji et al.
- "Big Data: Principles and Best Practices" – Thomas Erl et al.
- "Spark in Action" – Jean-Georges Perrin

Reference books:

- "Hadoop Operations" – Eric Sammer
- "Advanced Analytics with Spark" – Sandy Ryza et al.
- "Designing Data-Intensive Applications" – Martin Kleppmann
- "Big Data Fundamentals" – Thomas Erl et al.

COURSE CODE: DDSE404

COURSE NAME: DIGITAL LOGIC SYSTEMS

Course Objectives:

- To provide foundation in digital electronics essential for computer architecture understanding
- To develop ability to design and analyze digital circuits for computation
- To introduce hardware description languages for digital system modeling
- To bridge theoretical concepts with practical circuit implementation
- To prepare students for advanced courses in computer organization and embedded systems

Course Outcomes: At the end of the course students shall be able to

CO1	Understand number systems, binary arithmetic, and Boolean algebra fundamentals
CO2	Design and simplify combinational logic circuits using gates and Boolean algebra
C03	Analyze and design sequential logic circuits including flip-flops, registers, and counters
C04	Implement digital circuits using HDL (Hardware Description Language) basics

Unit	Content	Credit	Weightage
I	**Number Systems & Boolean Algebra** • **Number Systems:** Binary, Octal, Hexadecimal, conversions • **Binary Arithmetic:** Addition, subtraction (1's & 2's complement), multiplication • **Codes:** BCD, Gray code, ASCII, parity codes • **Boolean Algebra:** Laws and theorems, De Morgan's theorems • **Logic Gates:** AND, OR, NOT, NAND, NOR, XOR, XNOR - symbols, truth tables	1	25%

	• **Universal gates:** Implementation using NAND/NOR only • **Logic levels, voltage specifications, propagation delay**		

II	**Combinational Logic Design**
	• **Canonical forms:** SOP (Sum of Products) and POS (Product of Sums)
	• **Karnaugh Maps:** 2, 3, 4, and 5 variable K-maps, don't care conditions
	• **Quine-McCluskey minimization method**
	• **Combinational Circuits:**
	  ○ Adders: Half adder, full adder, ripple carry adder
	  ○ Subtractors: Half subtractor, full subtractor
	  ○ Multiplexers (MUX) and Demultiplexers (DEMUX)
	  ○ Encoders and Decoders (BCD to 7-segment)
	  ○ Comparators
	• **Implementation of logic functions using MUX/Decoders**
		1	25%

III	**Sequential Logic Design**
	• **Latches:** SR latch, D latch
	• **Flip-flops:** SR, JK, D, T flip-flops (edge-triggered)
	  ○ Characteristic tables, excitation tables
	  ○ Race-around condition, master-slave configuration
	• **Registers:** Buffer register, shift registers (SISO, SIPO, PISO, PIPO)
	• **Counters:**
	  ○ Asynchronous (ripple) counters
	  ○ Synchronous counters
	  ○ Mod-N counters, up/down counters
	  ○ Design using state diagrams and state tables
	• **Timing diagrams** for sequential circuits
		1	25%

IV	**Memory & Introduction to HDL**
	• **Memory Basics:** ROM, RAM (SRAM, DRAM)
	• **Memory Organization:** Address lines, data lines, capacity calculation
	• **Programmable Logic Devices:** PAL, PLA, FPGA basics
	• **Introduction to Verilog/VHDL:**
	  ○ Basic structure: modules, ports, data types
	  ○ Gate-level modeling
	  ○ Dataflow modeling
	  ○ Simple combinational and sequential circuit examples
	• **Digital System Applications:** Simple ALU design, traffic light controller
	• **Interfacing concepts:** I/O ports, tri-state buffers
		1	25%

Textbooks:

- "Digital Logic and Computer Design" – M. Morris Mano
- "Fundamentals of Digital Logic with Verilog Design" – Stephen Brown & Zvonko Vranesic
- "Digital Electronics: Principles and Applications" – Roger L. Tokheim
- "Digital Circuits and Design" – S. Salivahanan & S. Arivazhagan

Reference books:

- "Digital Design: With an Introduction to Verilog HDL" – M. Morris Mano & Michael D. Ciletti
- "Modern Digital Electronics" – R.P. Jain
- "Introduction to Digital Systems" – John Crisp
- "Digital Logic Design: A Rigorous Approach" – Guy Even & Moti Medina

Online Platforms:

- NPTEL (IIT) Courses – "Digital Circuits" by Prof. Santanu Chattopadhyay
- Coursera – "Introduction to Digital Circuits" by University of Illinois
- edX – "Computation Structures" by MIT

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-V

COURSE CODE: DDSE501

COURSE NAME: ADVANCED MACHINE LEARNING

Course Objectives:

- To transition from classical ML to advanced predictive modeling techniques
- To develop proficiency in deep learning frameworks and architectures
- To implement real-world ML systems like recommenders and NLP applications
- To introduce MLOps for production-ready machine learning
- To cultivate critical thinking for model selection and evaluation

Course Outcomes: At the end of the course students shall be able to

CO1	Implement ensemble methods and hyperparameter optimization techniques
CO2	Design and apply neural networks using TensorFlow/PyTorch for various tasks
C03	Build and evaluate recommendation systems and NLP pipelines
C04	Deploy machine learning models with MLOps practices

Unit	Content	Credit	Weightage

I	**Ensemble Methods & Model Optimization**
	• **Advanced Model Evaluation:** Cross-validation strategies, learning curves, bias-variance tradeoff
	• **Ensemble Learning:**
	  ○ Bagging: Random Forest, Extra Trees
	  ○ Boosting: AdaBoost, Gradient Boosting (XGBoost, LightGBM, CatBoost)
	  ○ Stacking and Voting Classifiers
	• **Hyperparameter Optimization:** Grid Search, Random Search, Bayesian Optimization (Optuna)
	• **Feature Engineering Advanced Techniques:** Target encoding, feature interactions, polynomial features
	• **Class Imbalance Handling:** SMOTE, ADASYN, cost-sensitive learning
	• **Model Persistence:** Pickle, Joblib, ONNX format
		1	25%

II	**Deep Learning Fundamentals** • **Neural Network Basics:** Perceptron, activation functions (ReLU, Sigmoid, Tanh), loss functions • **Deep Learning Frameworks:** TensorFlow/Keras vs PyTorch comparison • **Training Deep Networks:** Backpropagation, optimizers (Adam, RMSprop), batch normalization • **Convolutional Neural Networks (CNNs):** ○ Architecture: Convolution layers, pooling, flattening	1	25%

	○ Transfer Learning: VGG, ResNet, MobileNet ○ Applications: Image classification, object detection basics • **Preventing Overfitting:** Dropout, L2 regularization, data augmentation • **Hardware Considerations:** GPU vs CPU, cloud GPUs (Colab, Kaggle)		
III	**Specialized ML Applications** • **Recommendation Systems:** ○ Collaborative filtering (user-based, item-based) ○ Matrix factorization ○ Content-based filtering ○ Hybrid approaches • **Natural Language Processing:** ○ Text preprocessing pipelines ○ Word embeddings: Word2Vec, GloVe, FastText ○ Recurrent Neural Networks (RNNs) and LSTMs for text ○ Transformer architecture basics (BERT introduction)	1	25%
IV	**MLOps & Model Deployment** • **MLOps Fundamentals:** CI/CD for ML, model registry, experiment tracking • **Model Deployment Patterns:** REST API, batch inference, streaming • **Tools & Frameworks:** ○ MLflow for experiment tracking and model registry ○ FastAPI for creating ML APIs ○ Docker containerization for ML models • **Model Monitoring:** Concept drift, data drift, performance degradation • **Explainable AI (XAI):** SHAP, LIME, feature importance • **Edge Deployment:** TensorFlow Lite, ONNX Runtime for mobile/edge • **Ethical Considerations:** Model fairness, bias detection, privacy	1	25%

Textbooks:

- "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" (3rd Edition) – Aurélien Géron
- "Pattern Recognition and Machine Learning" – Christopher M. Bishop
- "Deep Learning" – Ian Goodfellow, Yoshua Bengio, Aaron Courville
- "Machine Learning Engineering" – Andriy Burkov

Reference books:

- "The Hundred-Page Machine Learning Book" – Andriy Burkov

- "Applied Predictive Modeling" – Max Kuhn & Kjell Johnson
- "Natural Language Processing with Transformers" – Lewis Tunstall et al.
- "Building Machine Learning Powered Applications" – Emmanuel Ameisen

Online Platforms:

- fast.ai – Practical deep learning courses
- DeepLearning.AI Specializations on Coursera
- Udacity Nanodegrees – ML Engineer track
- MLOps Zoomcamp – Free MLOps course

PRACTICAL LIST:

1. **Ensemble Methods Comparison**
   - Implement and compare Random Forest vs XGBoost on structured data
   - Use Optuna for hyperparameter optimization
2. **Advanced Feature Engineering Pipeline**
   - Build a complete pipeline with automated feature selection
   - Handle categorical variables with multiple encoding strategies
3. **CNN from Scratch**
   - Build a CNN for CIFAR-10 image classification
   - Implement data augmentation techniques
4. **Transfer Learning Project**
   - Fine-tune a pre-trained ResNet model for custom image dataset
   - Compare training from scratch vs transfer learning
5. **Neural Network Hyperparameter Tuning**
   - Use Keras Tuner to optimize network architecture
   - Implement early stopping and learning rate scheduling
6. **Movie Recommendation System**
   - Build collaborative filtering on MovieLens dataset
   - Create hybrid recommender with content-based features
7. **Sentiment Analysis with Deep Learning**
   - Implement LSTM for text classification
   - Compare with traditional TF-IDF + ML models
8. **Time Series Forecasting**
   - Use LSTM for stock price/weather prediction
   - Compare with traditional ARIMA models
9. **MLflow Experiment Tracking**
   - Track multiple model experiments with parameters and metrics
   - Register best model to MLflow model registry
10. **FastAPI Model Deployment**
    - Create REST API for model inference
    - Add input validation and documentation
11. **Explainable AI Implementation**
    - Use SHAP to explain complex model predictions
    - Create feature importance visualizations

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

COURSE CODE: DDSE502

COURSE NAME: CLOUD COMPUTING FOR DATA SCIENCE

Course Objectives:

- To introduce cloud computing concepts specifically tailored for data science applications
- To develop hands-on skills with major cloud providers' data services
- To enable deployment of scalable machine learning models and data pipelines
- To teach cost management and security considerations for cloud-based data projects
- To prepare students for cloud certifications and industry roles

Course Outcomes: At the end of the course students shall be able to

CO1	Understand cloud computing models, services, and deployment strategies for data science
CO2	Deploy and manage data science workloads on major cloud platforms (AWS/Azure/GCP)
C03	Implement scalable data pipelines and machine learning models in the cloud
C04	Utilize cloud-native services for data storage, processing, and analytics

Unit	Content	Credit	Weightage
I	**Cloud Computing Fundamentals for Data Science** • **Introduction to Cloud Computing:** Evolution, benefits, and challenges for data science • **Cloud Service Models:** IaaS, PaaS, SaaS, FaaS (Serverless) - which to use when • **Deployment Models:** Public, private, hybrid, multi-cloud strategies • **Major Cloud Providers Overview:** AWS, Azure, Google Cloud Platform - comparison • **Cloud Economics:** Pricing models, TCO, cost optimization for data workloads • **Setting up Cloud Environment:** Accounts, IAM, billing alarms, free tier management • **Cloud Security Basics:** Shared responsibility model, encryption, compliance	1	25%

II	**Cloud Storage & Databases for Data Science**
	• **Cloud Storage Services:**
	  ○ AWS S3 / Azure Blob Storage / Google Cloud Storage
	  ○ Lifecycle policies, versioning, cross-region replication
	• **Cloud Databases:**
	  ○ Relational: AWS RDS, Azure SQL Database, Cloud SQL
	  ○ NoSQL: AWS DynamoDB, Azure Cosmos DB, Firestore
	  ○ Data Warehousing: AWS Redshift, Azure Synapse, BigQuery
		1	25%

	• **Choosing the Right Storage:** Based on data type, access patterns, cost • **Data Lake Architecture:** Building data lakes on cloud storage • **Data Migration Strategies:** On-premises to cloud, cloud-to-cloud • **Hands-on:** Creating storage buckets, loading datasets, querying cloud databases		
III	**Data Processing & Analytics in Cloud** • **Compute Services for Data Science:** o Virtual Machines: AWS EC2, Azure VMs, Compute Engine o Containers: AWS ECS/EKS, Azure AKS, Google GKE o Serverless: AWS Lambda, Azure Functions, Cloud Functions • **Big Data Processing:** o AWS EMR (Hadoop/Spark), Azure HDInsight, Dataproc o Serverless ETL: AWS Glue, Azure Data Factory • **Data Orchestration:** AWS Step Functions, Azure Data Factory pipelines • **Stream Processing:** AWS Kinesis, Azure Stream Analytics • **Cloud Notebooks:** AWS SageMaker Notebooks, Azure Notebooks, Colab Enterprise • **Hands-on:** Running Spark jobs on cloud clusters, building ETL pipelines	1	25%
IV	**Machine Learning & MLOps in Cloud** • **Cloud ML Platforms:** o AWS SageMaker (end-to-end ML platform) o Azure Machine Learning o Google Vertex AI • **ML Pipeline Components:** o Data preparation and feature store o Model training (managed training jobs) o Hyperparameter tuning o Model registry and deployment • **Serverless Inference:** Deploying models as API endpoints • **MLOps in Cloud:** CI/CD pipelines, model monitoring, drift detection	1	25%

Textbooks:

- "Cloud Computing for Science and Engineering" – Ian Foster & Dennis B. Gannon
- "Data Science on AWS" – Chris Fregly & Antje Barth
- "Cloud Native Data Center Networking" – Dinesh Dutt
- "Azure Data Engineering" – Vlad Riscutia

Reference books:

- "Building Data Science Solutions with Azure" – Peter Kossitsky

- "Google Cloud Platform for Data Science" – K.C. Yerneni
- "Data Engineering with AWS" – Gareth Eagar
- "MLOps Engineering on AWS" – Joshua Arvin Lat

Online Platforms:

- AWS Skill Builder – Free digital training including AWS Educate
- Microsoft Learn – Free Azure modules and learning paths
- Google Cloud Skills Boost – Free training and hands-on labs
- Qwiklabs – Hands-on cloud labs (some free credits available)

COURSE CODE: DDSE503

COURSE NAME: NATURAL LANGUAGE PROCESSING

Course Objectives:

- To introduce fundamental concepts, tasks, and applications of natural language processing.
- To develop skills in text preprocessing, feature extraction, and language modeling.
- To enable students to implement and evaluate NLP models for classification, generation, and understanding tasks.
- To prepare students for advanced topics in conversational AI, sentiment analysis, and language translation.
- To foster awareness of ethical considerations and biases in NLP systems.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** NLP pipelines, text preprocessing techniques, and linguistic fundamentals.
CO2	**Implement** traditional NLP methods: tokenization, stemming, TF-IDF, and word embeddings.
C03	**Build** and evaluate NLP models for tasks like sentiment analysis, text classification, and named entity recognition.
C04	**Apply** transformer-based models and sequence-to-sequence architectures for advanced NLP tasks.

Unit	Content	Credit	Weightage
I	**Introduction to Neural Networks** What is NLP? History, applications, and challenges - Linguistic basics: Morphology, syntax, semantics, pragmatics - Text preprocessing: Tokenization, stopword removal, stemming, lemmatization - Regular expressions for text pattern matching - Applications: Chatbots, search engines, document summarization	1	25%
II	**Convolutional Neural Networks (CNNs)** CNN architecture: Convolutional layers, pooling, flattening, fully connected layers - Popular CNN architectures: LeNet, AlexNet, VGG (overview) - Transfer learning and fine-tuning - Data augmentation for image data	1	25%

	- **Applications:** Image classification, object detection, medical imaging		
III	**Recurrent Neural Networks (RNNs) & Sequence Models** RNN basics: Sequential data handling, hidden states, vanishing gradient problem - Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) - Sequence-to-sequence models (Seq2Seq) basics - Time-series forecasting and text generation intro - **Applications:** Sentiment analysis, language modeling, stock prediction	1	25%
IV	**Model Optimization, Regularization & Ethics** Regularization: Dropout, batch normalization, L1/L2 regularization - Hyperparameter tuning: Learning rate, batch size, epochs - Model evaluation: Overfitting vs underfitting, confusion matrix for DL - Ethical considerations: Bias in AI, fairness, interpretability - **Applications:** Model deployment basics, bias detection in models	1	25%

Textbooks:

- *Deep Learning* — Ian Goodfellow, Yoshua Bengio, Aaron Courville
- *Neural Networks and Deep Learning* — Michael Nielsen (Online free book)

Reference books:

- *Deep Learning with Python* — François Chollet
- *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* — Aurélien Géron
- *Practical Deep Learning for Cloud, Mobile, and Edge* — Anirudh Koul et al.
- *Deep Learning for Computer Vision* — Rajalingappaa Shanmugamani

Online Platforms:

- NPTEL:
  1. *Deep Learning* by Prof. Mitesh Khapra (IIT Madras)
  2. *Deep Learning for Computer Vision* by IIT Ropar
- Coursera:
  1. *Deep Learning Specialization* by Andrew Ng
  2. *Introduction to Deep Learning* by MIT

PRACTICAL LIST:

Section A: Neural Networks Basics

1. Implement a perceptron from scratch using NumPy.
2. Build a feedforward neural network using TensorFlow/Keras for MNIST digit classification.
3. Experiment with different activation functions (Sigmoid, ReLU, Tanh) and compare performance.
4. Visualize weight updates and loss curves during training.

Section B: Convolutional Neural Networks (CNNs)

5. Build a CNN using Keras for CIFAR-10 image classification.
6. Implement data augmentation (rotation, zoom, flip) and observe impact on accuracy.
7. Use transfer learning with a pre-trained model (VGG16/ResNet) for a custom image dataset.

8. Visualize CNN feature maps and filters to understand layer activations.

Section C: Recurrent Neural Networks (RNNs)

9. Implement an RNN/LSTM for time-series prediction (e.g., stock prices, temperature).
10. Build a sentiment analysis model using LSTM on IMDB movie reviews.
11. Create a text generation model using RNNs on a small text corpus.
12. Compare performance of Simple RNN vs LSTM vs GRU on a sequence task.

Section D: Optimization & Mini-Project

13. Apply dropout and batch normalization to a CNN and evaluate effect on overfitting.
14. Perform hyperparameter tuning using Keras Tuner or Grid Search.
15. Mini-Project: Develop an end-to-end deep learning application (e.g., facial expression recognition, spam detection using LSTMs).
16. Analyze model bias using a fairness toolkit (e.g., IBM AI Fairness 360 demo).

COURSE CODE: DDSE504

COURSE NAME: OPERATING SYSTEMS

Course Objectives:

- To introduce fundamental concepts, functions, and structures of operating systems.
- To develop understanding of process management, CPU scheduling, and synchronization.
- To enable students to comprehend memory management, file systems, and storage organization.
- To prepare students for understanding system-level programming and modern OS architectures.
- To foster awareness of security, protection, and distributed systems basics.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** OS structures, functions, and types of operating systems.
CO2	**Analyze** process management, scheduling algorithms, and synchronization mechanisms.
C03	**Describe** memory management techniques, paging, segmentation, and virtual memory.
C04	**Understand** file systems, storage management, and basic OS security concepts.

Unit	Content	Credit	Weightage
I	**Introduction to OS & System Structures** • Introduction to OS: Definition, objectives, functions • Types of OS: Batch, multiprogramming, time-sharing, real-time, distributed • OS structure: Monolithic, layered, microkernel, modular • System calls and types • Operating system services • **Applications**: Command-line interface, shell scripting basics	1	25%
II	**Process Management & CPU Scheduling** • Process concept: Process states, PCB, operations • Threads: Multithreading models, benefits	1	25%

	• CPU scheduling: Criteria, algorithms (FCFS, SJF, Priority, Round Robin) • Process synchronization: Critical section, semaphores, mutex • Deadlocks: Conditions, prevention, avoidance, detection, recovery • **Applications**: Task scheduling in OS, multi-threaded applications		
III	**Memory Management** • Memory hierarchy: Registers, cache, RAM, secondary storage • Contiguous memory allocation: Fixed and variable partition • Fragmentation: Internal and external • Paging: Basic concept, page table, TLB • Segmentation • Virtual memory: Demand paging, page replacement algorithms (FIFO, LRU, Optimal) • **Applications**: Memory allocation in programs, virtual memory management	1	25%
IV	**File Systems & OS Security** • File concepts: Attributes, operations, types • Directory structure: Single-level, two-level, tree-structured • File allocation methods: Contiguous, linked, indexed • Disk scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN • Protection and security: Access control, authentication, threats, malware • **Applications**: File management in OS, basic disk management, user authentication	1	25%

Textbooks:

- *Operating System Concepts* – Abraham Silberschatz, Peter B. Galvin, Greg Gagne
- *Operating Systems: A Concept-Based Approach* – D. M. Dhamdhere

Reference books:

- *Modern Operating Systems* – Andrew S. Tanenbaum
- *Operating Systems: Principles and Design* – P. C. P. Bhatt
- *Operating Systems: Internals and Design Principles* – William Stallings
- *Operating Systems* – Achyut S. Godbole & Atul Kahate

Online Platforms:

- NPTEL
  - *Operating Systems* by Prof. P. K. Biswas (IIT Kharagpur)
  - *Introduction to Operating Systems* by IIT Madras
- Coursera
  - *Operating Systems and You: Becoming a Power User* by Google
  - *Introduction to Operating Systems* by University of Colorado

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-VI

COURSE CODE: DDSE601

COURSE NAME: DEEP LEARNING FUNDAMENTALS

Course Objectives:

- To introduce the fundamental concepts, architectures, and mathematical foundations of deep learning.
- To develop proficiency in implementing, training, and evaluating neural networks using modern frameworks.
- To enable students to design and apply convolutional and recurrent neural networks to real-world problems.
- To prepare students for advanced topics in computer vision, natural language processing, and generative AI.
- To foster awareness of ethical considerations, model interpretability, and best practices in deep learning.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** the architecture, components, and training dynamics of artificial neural networks.
CO2	**Implement** and optimize convolutional neural networks (CNNs) for image-based tasks.
C03	**Design** and apply recurrent neural networks (RNNs) and LSTMs for sequential data.
C04	**Evaluate** deep learning models, apply regularization techniques, and understand ethical implications.

Unit	Content	Credit	Weightage
I	**Introduction to Neural Networks** What is Deep Learning? Evolution, key milestones, and applications - Artificial neurons, perceptrons, activation functions (Sigmoid, Tanh, ReLU, Softmax) - Feedforward neural networks: Architecture, forward propagation, loss functions - Backpropagation, gradient descent, optimizers (SGD, Adam, RMSprop) - **Applications:** Digit recognition, binary classification, simple regression	1	25%
II	**Convolutional Neural Networks (CNNs)** CNN fundamentals: Convolutional layers, pooling, padding, stride - CNN architectures: LeNet, AlexNet, VGG, ResNet (overview) - Transfer learning and fine-tuning using pre-trained models - Data augmentation and regularization techniques (Dropout, BatchNorm)	1	25%

	- **Applications:** Image classification, object detection, medical image analysis		
III	**Recurrent Neural Networks (RNNs) & Sequence Models** Introduction to sequential data and RNN architecture - Challenges: Vanishing/exploding gradients - Advanced RNNs: LSTM, GRU, Bidirectional RNNs - Sequence-to-sequence models and attention mechanism (basics) - **Applications:** Time-series forecasting, sentiment analysis, text generation	1	25%
IV	**Model Optimization, Ethics & Deployment** Hyperparameter tuning: Learning rate, batch size, epochs - Regularization: L1/L2, dropout, early stopping, data augmentation - Model evaluation: Overfitting vs underfitting, cross-validation - Ethical AI: Bias in datasets, fairness, transparency, and interpretability - **Applications:** Deploying models with TensorFlow Serving/Flask, bias auditing	1	25%

Textbooks:

- *Deep Learning* — Ian Goodfellow, Yoshua Bengio, Aaron Courville
- *Deep Learning with Python* — François Chollet

Reference books:

- *Neural Networks and Deep Learning: A Textbook* — Charu C. Aggarwal
- *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* — Aurélien Géron
- *Practical Deep Learning for Cloud, Mobile, and Edge* — Anirudh Koul et al.
- *Deep Learning for Computer Vision* — Rajalingappaa Shanmugamani

Online Platforms:

- NPTEL:
  1. *Deep Learning* by Prof. Mitesh Khapra (IIT Madras)
  2. *Deep Learning for Computer Vision* by IIT Ropar
- Coursera:
  1. *Deep Learning Specialization* by Andrew Ng
  2. *Introduction to Deep Learning* by National Research University Higher School of Economics

PRACTICAL LIST:

Neural Networks & Optimization

1. Implement a multi-layer perceptron (MLP) from scratch using NumPy.
2. Build and train a neural network with Keras/TensorFlow for the MNIST dataset.
3. Experiment with different activation functions and optimizers; compare performance.
4. Visualize loss curves, accuracy, and weight distributions during training.

Section B: Convolutional Neural Networks (CNNs)

5. Design and train a CNN for CIFAR-10 image classification.
6. Implement data augmentation techniques (rotation, flip, zoom) and evaluate impact.
7. Use transfer learning (VGG16/ResNet50) on a custom image dataset.

8. Visualize convolutional filters and feature maps to interpret model decisions.
   Section C: Recurrent Neural Networks (RNNs)
9. Build an LSTM model for time-series prediction (e.g., stock prices).
10. Implement a sentiment analysis model using RNNs on the IMDB dataset.
11. Create a character-level text generation model using LSTMs.
12. Compare Simple RNN, LSTM, and GRU on a sequence classification task.
    Section D: Model Tuning & Mini-Project
13. Apply dropout, batch normalization, and L2 regularization to a CNN; measure effect on overfitting.
14. Perform hyperparameter tuning using Keras Tuner or Optuna.
15. Deploy a trained deep learning model using Flask or TensorFlow Serving.
    Mini-Project: Develop an end-to-end deep learning application (e.g., facial emotion recognition, handwritten equation solver, or spam detector with LSTMs).

COURSE CODE: DDSE602

COURSE NAME: IOT FOR DATA SCIENCE

Course Objectives:

- To bridge IoT hardware fundamentals with data science applications
- To develop skills in collecting and processing real-time sensor data
- To apply machine learning techniques to IoT-generated data
- To implement cloud-based IoT data architectures
- To prepare students for IoT analytics roles in Industry 4.0

Course Outcomes: At the end of the course students shall be able to

CO1	Understand IoT architecture, components, and protocols for data acquisition
CO2	Collect, preprocess, and store sensor data from IoT devices
C03	Analyze and visualize time-series IoT data using data science techniques
C04	Implement predictive models on IoT data streams

Unit	Content	Credit	Weightage

I	**IoT Fundamentals & Architecture**
	• **Introduction to IoT:** Definition, evolution, and Industry 4.0 applications
	• **IoT Architecture Layers:** Perception, network, middleware, application
	• **IoT Components:** Sensors, actuators, microcontrollers, gateways
	• **Communication Protocols:**
	  ○ Short-range: Bluetooth, Zigbee, WiFi
	  ○ Long-range: LoRaWAN, NB-IoT, Cellular
	  ○ Messaging: MQTT, CoAP, HTTP for IoT
	• **IoT Platforms Overview:** AWS IoT, Azure IoT, Google Cloud IoT Core
	• **Data Generation Patterns in IoT:** Time-series, event-based, batch
	• **Security Considerations:** Device authentication,
		1	25%

	data encryption		

II	**IoT Data Acquisition & Edge Processing**
	• **Hardware Components:**
	  ○ Microcontrollers: ESP32, Arduino (with sensors: temperature, humidity, motion)
	  ○ Single-board Computers: Raspberry Pi for edge computing
	• **Sensor Data Collection:**
	  ○ Analog vs digital sensors
	  ○ Sampling rates, precision, calibration
	  ○ Noise filtering and signal conditioning basics
	• **Edge Computing Concepts:**
	  ○ Why process at the edge?
	  ○ Lightweight ML on edge devices (TinyML introduction)
	  ○ Data reduction techniques before transmission
		1	25%

III	**IoT Data Processing & Analytics**
	• **Time-Series Data Characteristics:** Seasonality, trends, anomalies
	• **Data Preprocessing for IoT:**
	  ○ Handling missing values in streaming data
	  ○ Outlier detection for sensor data
	  ○ Data normalization and scaling
	• **Stream Processing:**
	  ○ Window-based operations
	  ○ Real-time aggregation
	  ○ Introduction to Apache Kafka/Spark Streaming for IoT
	• **IoT Data Storage:**
	  ○ Time-series databases: InfluxDB, TimescaleDB
	  ○ Optimizing storage for sensor data
	• **Visualization of IoT Data:**
	  ○ Real-time dashboards (Grafana, Plotly Dash)
	  ○ Geospatial visualization of sensor networks
		1	25%

IV	**IoT Machine Learning & Cloud Integration**
	• **ML Applications in IoT:**
	  ○ Predictive maintenance
	  ○ Anomaly detection in sensor data
	  ○ Energy consumption forecasting
	• **Edge vs Cloud ML Trade-offs:**
	  ○ Simple models on edge (decision trees, lightweight neural networks)
	  ○ Complex models in cloud
	• **Cloud IoT Platforms:**
	  ○ AWS IoT Core with SageMaker integration
		1	25%

	○ Azure IoT Hub with ML services ○ End-to-end pipeline implementation		

Textbooks:

- "IoT and Edge Computing for Architects" – Perry Lea
- "Building the Internet of Things with ESP32 and MQTT" – Vedat Ozan Oner
- "Practical Time Series Analysis" – Aileen Nielsen
- "TinyML: Machine Learning with TensorFlow Lite on Arduino" – Pete Warden & Daniel Situnayake

Reference books:

- "IoT Data Analytics with Python" – Pethuru Raj
- "Hands-On Industrial Internet of Things" – Giacomo Veneri
- "Time Series Forecasting in Python" – Marco Peixeiro
- "IoT Systems and Industrial Applications with AI" - Jitendra Kumar Verma

Online Platforms:

- Coursera – "IoT Specialization" by University of Illinois
- edX – "IoT Programming and Big Data" by Curtin University
- Udemy – Practical IoT courses with hardware kits

PRACTICAL LIST:

1. Basic Sensor Setup:
   - Set up temperature/humidity sensor (DHT22) with ESP32
   - Collect data and display on serial monitor
   - *Tools: ESP32, DHT22, Arduino IDE*
2. Wireless Data Transmission:
   - Implement MQTT protocol to send sensor data to local broker
   - Use Python to subscribe and receive data
   - *Tools: ESP32, Mosquitto MQTT broker, Python Paho-MQTT*
3. Multi-Sensor Network:
   - Create network with 3+ sensors (temperature, motion, light)
   - Aggregate data at gateway (Raspberry Pi)
   - *Tools: Multiple sensors, Raspberry Pi, Python*
4. Edge Processing:
   - Implement simple filtering on ESP32 (moving average)
   - Compare raw vs filtered data
   - *Tools: ESP32, MicroPython*
5. Time-Series Database Setup:
   - Install and configure InfluxDB
   - Store sensor data with proper timestamps
   - *Tools: InfluxDB, Python influxdb-client*
6. Real-time Dashboard:
   - Create Grafana dashboard for sensor data visualization
   - Implement alerts for threshold values
   - *Tools: Grafana, InfluxDB*
7. Stream Processing Pipeline:
   - Set up Apache Kafka for IoT data streams
   - Process data with window functions
   - *Tools: Kafka, Python kafka-python*
8. Anomaly Detection:
   - Implement statistical anomaly detection on sensor data
   - Visualize anomalies in dashboard

- *Tools: Python (scikit-learn, pandas), Grafana*

9. Predictive Model for IoT:
   - Train simple regression model for temperature forecasting
   - Deploy model for real-time predictions
   - *Tools: Python (scikit-learn), Flask API*
10. Cloud IoT Integration:
   - Send sensor data to AWS IoT Core/Azure IoT Hub
   - Store data in cloud database (AWS Timestream/Azure Cosmos DB)
   - *Tools: AWS/Azure free tier, Python SDKs*

COURSE CODE: DDSE603

COURSE NAME: ETHICS AND GOVERNANCE IN AI/DS

Course Objectives:

- To introduce fundamental ethical principles, challenges, and societal impacts of artificial intelligence.
- To develop skills in identifying, assessing, and mitigating ethical risks in AI systems.
- To enable students to understand and apply governance frameworks, regulations, and standards for responsible AI.
- To prepare students for roles requiring ethical oversight, compliance, and responsible innovation in AI.
- To foster critical thinking around fairness, transparency, accountability, and inclusivity in AI design and deployment.

Course Outcomes: At the end of the course students shall be able to

CO1	**Explain** core ethical principles, biases, and societal impacts of AI systems.
CO2	**Identify** ethical risks and apply fairness, accountability, and transparency (FAT) frameworks.
C03	**Evaluate** AI systems for compliance with legal, regulatory, and governance standards.
C04	**Design** ethical AI solutions and draft governance policies for responsible deployment.

Unit	Content	Credit	Weightage
I	**Introduction to AI Ethics & Societal Impact** What is AI Ethics? History, need, and core principles - Ethical theories: Utilitarianism, deontology, virtue ethics - Societal impacts: Job displacement, privacy, surveillance, digital divide - Case studies: Biased hiring algorithms, facial recognition misuse - **Applications:** Ethical AI design, public policy, tech accountability	1	25%
II	**Bias, Fairness & Transparency in AI** Types of bias: Data bias, algorithmic bias, human bias - Measuring fairness: Demographic parity, equal opportunity, disparate impact - Explainable AI (XAI): Techniques (LIME, SHAP),	1	25%

	model interpretability - Tools for bias detection: IBM AI Fairness 360, Google What-If Tool - **Applications:** Fair credit scoring, unbiased recruitment, transparent healthcare AI		
III	**AI Governance, Regulations & Standards** Global AI regulations: GDPR (EU), AI Act (EU), Algorithmic Accountability Act (US) - National frameworks: India's AI Strategy, NITI Aayog guidelines - Standards: IEEE Ethically Aligned Design, ISO/IEC 42001 - Corporate governance: Ethics boards, compliance officers, audit trails - **Applications:** Regulatory compliance, corporate AI policy, certification	1	25%
IV	**Responsible AI Deployment & Future Challenges** Responsible innovation: Human-centered design, participatory AI - Ethical deployment: Monitoring, feedback loops, incident response - Future challenges: Autonomous weapons, deepfakes, AGI ethics - Sustainability: Green AI, carbon footprint of AI systems - **Applications:** Ethical AI toolkits, audit reports, sustainability assessments	1	25%

Textbooks:

- *Ethics of Artificial Intelligence and Robotics* — Vincent C. Müller
- *Weapons of Math Destruction* — Cathy O'Neil

Reference books:

- *The Alignment Problem* — Brian Christian
- *Atlas of AI* — Kate Crawford
- *AI Ethics* — Mark Coeckelbergh
- *Practical Fairness* — Aileen Nielsen

Online Platforms:

- NPTEL:
  1. *Ethics in AI* by Prof. Ponnurangam Kumaraguru (IIIT Delhi)
  2. *Responsible AI* by IIT Hyderabad
- Coursera:
  1. *AI Ethics* by University of Helsinki
  2. *Responsible AI* by Google Cloud

PRACTICAL LIST:

Section A: Ethical Analysis & Case Studies

1. Analyze a real-world AI ethics case study (e.g., COMPAS recidivism algorithm) and write a critique.
2. Conduct a stakeholder impact analysis for an AI system (e.g., automated hiring tool).
3. Use ethical frameworks (e.g., EU Ethics Guidelines) to evaluate a given AI application.
4. Debate ethical dilemmas in AI (e.g., trolley problem for autonomous vehicles).

Section B: Bias Detection & Fairness Assessment

5. Detect bias in a dataset using Python (e.g., using aif360 or fairlearn).
6. Apply fairness metrics (demographic parity, equal opportunity) to a classification model.
7. Mitigate bias using pre-processing, in-processing, or post-processing techniques.
8. Use SHAP or LIME to interpret model predictions and identify unfair influences.

Section C: Governance & Compliance Simulation

9. Draft an AI ethics policy for a fictional company deploying facial recognition.
10. Conduct a GDPR compliance checklist for an AI system handling personal data.
11. Role-play an AI ethics board meeting to approve/reject a new AI product.
12. Create an AI incident response plan for a bias-related failure.

Section D: Responsible AI Project & Audit

13. Design a “Responsible AI Checklist” for deployment covering fairness, transparency, privacy, and accountability.
14. Perform an ethical audit of an open-source AI model or dataset.
15. Develop an awareness campaign or educational material on AI ethics for non-technical stakeholders.

Final Project: Create a comprehensive “AI Ethics & Governance Report” for a real or hypothetical AI system, covering risk assessment, mitigation strategies, and compliance recommendations.