Assignment 1	
Dr. Keshav Sinha **Total Marks: 100 Marks**	**Course: Database Management Systems** **Course Code: CSEG2072**
Last Date: 03/10/2025	

Marks Distribution Rule (Practical – 25 Marks)

Criteria	Marks	Description
Implementation / Code Correctness	10	HTML/JS code runs without major errors; meets requirements of the question.
Output / Functionality	5	Correct results (e.g., correct scheduling chart, correct resource allocation result, proper diagram rendering).
Design / Presentation	5	Webpage layout neat, properly structured, readable, user-friendly.
Explanation / Documentation	5	Comments in code OR a short write-up explaining approach, logic, and observations.

S.No	Practical Question	Unit	CO	Marks	Assigned To
Q1	**DBMS Fundamentals Visualizer** – Create an HTML/JS page that demonstrates: • **Three-Schema Architecture** (External, Conceptual, Internal) with interactive layers • Difference between **Schema and Instance** using simple form/table examples • Advantages of DBMS over traditional file systems (interactive cards or comparisons).	Unit I	CO1	25	Odd SAP ID
Q2	**Data Models & DBMS Architecture Explorer** – Develop a webpage that explains: • Different **Data Models** (Relational, Hierarchical, Network, Document) with small examples • Visual comparison of **Centralized vs Client–Server Architecture** using flow animations or diagrams.	Unit I	CO1	25	Even SAP ID
Q3	**ER/EER Diagram Builder** – Build a drag-and-drop tool in HTML/JS that allows: • Adding **Entities and Attributes** (simple, composite, derived, multivalued) • Drawing **Relationships** with cardinality (1:1, 1:N, M:N) and participation constraints • Auto-generating **Relational Schema** (tables with PK/FK) from the ER diagram.	Unit II	CO2	25	Odd SAP ID
Q4	**Normalization Assistant** – Create a webpage where students can: • Enter **Attributes and Functional Dependencies** • Compute **Attribute Closures** and **Candidate Keys** • See step-by-step transformation of relations into **1NF → 2NF → 3NF → BCNF** • Highlight **lossless join** and **dependency preservation** checks.	Unit II	CO2	25	Even SAP ID
Q5	**Relational Algebra Visualizer** – Develop an in-browser simulator where: • Users create small **in-memory relations** (tables) • Apply **Relational Algebra operations**: SELECT (σ), PROJECT (π), JOIN (⋈), UNION (∪), DIFFERENCE (−) • Display **result tables dynamically** after each operation.	Unit III	CO1 / CO4	25	Odd SAP ID
Q6	**SQL Query Simulator** – Implement an HTML/JS page that mimics SQL queries using arrays as tables. Features must include: • **Basic queries**: SELECT, WHERE, ORDER BY • **Aggregate queries**: COUNT, SUM, AVG, GROUP BY, HAVING • **Set operations**:	Unit III	CO4	25	Even SAP ID

	UNION, INTERSECT, EXCEPT • **Nested queries** with subselects.				
Q7	**Transaction Management Simulator** – Build a page that: • Shows **Transaction States** (Active, Partially Committed, Committed, Failed, Aborted) with interactive buttons/animations • Demonstrates **ACID properties** with examples • Simulates a **Dirty Read scenario** between two transactions • Provides a demo of **GRANT and REVOKE privileges** on sample data.	Unit III	CO3 / CO4	25	Odd SAP ID
Q8	**Deadlock & Recovery Visualizer** – Create an HTML/JS page that: • Displays a **Process–Resource allocation graph** • Highlights a **Deadlock cycle** visually when it occurs • Implements **Banker's Algorithm** to check safe/unsafe states • Demonstrates **Deadlock recovery** strategies (abort or resource preemption).	Unit III	CO3	25	Even SAP ID