	ITER, SIKSHA 'O' ANUSANDHAN (Deemed to be University)		Minor Project
Branch	CSIT	Programme	**B.Tech**
Course Name	PROGRAMMING PROJECTS WITH JAVASCRIPT(PPWJ)	Semester	5th
Course Code	CSE 3838	Academic Year	2025-2026

Minor Project

Learning Level (LL)	**L1**: Remembering	**L3:** Applying	**L5**: Evaluating
	L2: Understanding	**L4**: Analysing	**L6**: Creating

Q's	Questions	COs	LL
1	Develop a Memory Card Matching Game using HTML, CSS, and JavaScript. The game consists of cards placed face down. The player clicks two cards at a time to find matching pairs. If the cards match, they remain visible; otherwise, they flip back. **Step 1:** Create the HTML Structure • Create an HTML file named index.html. • Add a heading with the title "Memory Card Matching Game". • Create a container to hold the cards. • Add multiple card elements using div tags. Display: Number of attempts Match status message Add a Reset Game button. **Step 2:** Apply CSS Styling • Use an external CSS file. • Arrange the cards in a grid layout. • Give each card: • Fixed width and height • Border and background color • Style the front and back of cards. • Add a flip animation using CSS transitions. Step 3: Access DOM Elements • Use JavaScript to select:	CO1,CO2, CO3,CO4, CO5,CO6	L3,L4

- All card elements
- Attempt counter
- Message display

Initialize variables to store:

- Selected cards
- Attempt count
- Matched pairs

Step 4: Assign Data to Cards

- Assign a matching value or symbol to each card using data-* attributes.
- Ensure that each value appears exactly twice.
- Shuffle the cards using JavaScript logic.
- Step 5: Handle Mouse Click Event
- Add a click event listener to each card.

On click:

- Flip the card
- Store the selected card
- Prevent clicking the same card twice.

Step 6: Match Comparison Logic

When two cards are selected:

- Compare their assigned values

If the cards match:

- Keep them flipped
- Increase matched pair count

If they do not match:

- Flip them back after a short delay
- Increase the attempt counter for every pair selection.

Step 7: Display Game Status

	• Update the attempt count dynamically using DOM manipulation. • Display a success message when all pairs are matched. • Disable further card clicks after the game is completed. **Step 8:** Reset the Game • Clear all variables. • Shuffle the cards again. • Reset attempts and messages. • Restore cards to face-down state. Expected Output • Cards appear in a grid layout • Cards flip on mouse click • Matching cards stay open • Non-matching cards flip back • Attempts are counted • Success message appears when all pairs are matched		

Note:

1. Minor Project carries a weightage of 20 **marks out of 100**
2. All course outcomes were covered.

Course Outcomes	CO1	Understand Javascript syntax, programming environment, and design functional programs using variables, data types, and operators.
	CO2	Apply Javascript arrays, logic statements, and loops to develop programs that solve real-world problems.
	CO3	Implement Javascript functions and apply OOP concepts such as classes and inheritance to create modular, reusable code.
	CO4	Utilize built-in Javascript methods for arrays, strings, numbers, math, and dates to manipulate and process data effectively.
	CO5	Develop interactive web elements using the DOM and dynamic element manipulation using DOM.
	CO6	Design dynamic and interactive event-driven web applications.