CS324 – Web Application Development

June 30, 2025 *Final Exam* *Maximum Marks: 50*

Q1. Consider a scenario where you are developing a web application that includes a feature to create multiple, independent counters on a single page. For instance, clicking one button increments "Counter A" and clicking another button increments "Counter B". Each counter must maintain its own state without interfering with the others. Write JavaScript code which uses closures to create these independent counters. Also write code to demonstrate the usage of the closures. [CLO 2] [5]

Q2. JavaScript provides two primary mechanisms for creating objects: constructor function and the class syntax. Imagine you are tasked with creating a User object constructor that has properties like username and email, and a method login() that prints a welcome message. Implement the User class using a constructor function and using the class syntax. [CLO 3] [5]

Q3. You are required to build a dynamic product filtering interface. When the page loads, you use the fetch API to get an array of product objects from a server (https://example.com/api/v2/products). Each object has properties like name, category, and price. You must then render these products on the page. The interface also includes a set of checkboxes for categories (e.g., "Electronics", "Apparel").

a) Write code to dynamically create and insert HTML elements for each product into the DOM after the data is fetched.

b) When a user checks or unchecks a category checkbox, the displayed list of products should update in real-time to show only products from the selected categories. Use JavaScript array functions (.filter(), .forEach()) to handle this logic. [CLO 3] [10]

Q4. You are tasked with building a simple back-end service using Node.js and Express that manages a collection of "notes" stored in a JSON file on the server (notes.json). The front-end, built with Vite and vanilla JavaScript, will interact with this service. [CLO 4] [10]

a) Explain the fundamental difference between the Node.js runtime environment and the browser environment. In the context of your Express server, describe why you must use asynchronous methods (e.g., fs.readFile, fs.writeFile) for file I/O operations on notes.json rather than their synchronous counterparts. What are the risks of using synchronous file operations in a server environment?

b) Conceptually outline how you would set up an Express server to handle a GET request to /api/notes (to read and return all notes) and a POST request to /api/notes (to add a new note). Describe the flow of data:

i. How the front-end would use the fetch API to make the POST request, including setting the correct headers and sending the new note data in the request body?

ii. How the Express server would receive the request, parse the JSON body, read the notes.json file, add the new note, and then write the updated data back to the file?

iii. How the server would then send an appropriate response back to the client?