Zunaira Abdul Aziz
BSSE23058

Part 2- Subjective

Question 4: (CLO-1 – 4 Marks)
Scenario:
You are developing a dashboard for a school management system. The Home Page should have:

- A header with the school name.
- A responsive navigation bar with links to Home, Students, and Contact Us.
- Use HTML5 semantic elements and Bootstrap for layout.
- Add a simple CSS hover effect on navigation links.

Task:
Code this page using HTML, CSS, and Bootstrap only.

Question 5: (CLO-2 – 8 Marks)
Scenario:
In a task management system, create a student form using HTML, CSS, and JavaScript:

- Form fields: Name, Email, Phone, Task Description.
- Validation:
  - Email format check.
  - Phone number must be exactly 11 digits.
  - All fields required.
- Show inline error messages below each field on invalid input.
- Show a success message on valid submission.

Task:
Code this form (frontend only).

Question 6: (CLO-3 – 12 Marks)
Scenario:
Build a React.js Employee Directory app:

- Create an EmployeeCard component showing Name, Email, Department.
- Add a "Delete" button to remove an employee from the list dynamically.
- Manage data using React state only (no backend).

Task:
Code the React component and state logic.

Question 7: (CLO-3 – 12 Marks)
Scenario:
In a Product Catalog React app:

- Create a search bar to filter products by name as the user types.
- Display products in a table format.
- Use static array data stored in React state.

Task:
Write the React component implementing search and display.

Question 8: (CLO-3 – 12 Marks)
Scenario:
Build a MongoDB inventory system:

- Create a Mongoose schema for Product with fields:
  - name (string, required)
  - price (number, required)
  - quantity (number, required)
  - category (string, enum: ['Electronics', 'Clothing', 'Food'])
  - createdAt (date, defaults to current date)

Task:
Write the complete Mongoose schema.

Question 9: (CLO-2 – 8 Marks)
Scenario:
Develop an Express.js API for a library system:

- Create a POST route /addBook accepting JSON: { title, author, year }.
- Validate title and author as required fields.
- Return success or error message in JSON.

Task:
Write Express.js route code.

Question 10: (CLO-3 – 12 Marks)
Scenario:
Create a User Management API with Express.js and MongoDB:

- Create a GET route to fetch all users from the MongoDB users collection.
- Return the list as a JSON response.

Task:
Write the Express route using Mongoose.

Question 11: (CLO-3 – 12 Marks)
Scenario:
Develop a Student Management System using HTML, CSS, Express.js, and MongoDB:

- Create a form page to add a student with fields: name, email, roll number.
- On form submission, save data to MongoDB using Express.js.
- Display a success message after saving.

Task:
Write the complete code for the form and Express backend.

Question 12: (CLO-3 – 5 Marks) - Viva

Code + Report Submission Instructions

1- Recommended approach for coded submission (Zip all)

```
FinalExam_12345/
        Q1_HTML_CSS_Bootstrap/
        Q2_Form_Validation_JS/
        Q3_React_EmployeeDirectory/
        Q4_React_ProductSearch/
        Q5_MongoDB_Schema/
        Q6_Express_AddBook_API/
        Q7_Express_GetUsers_API/
        Q8_StudentManagement_FullStac
```

2- Submit your project report as a Microsoft Word document (.docx) file.

3- The file should be well-organized, properly formatted, and named as: YourName_RollNumber.docx

4- Example: AliKhan_12345.docx

5- Include clear screenshots for each question, along with captions, to help explain what you have done.