CC-331L Web Technologies Lab Practical Fall 2023

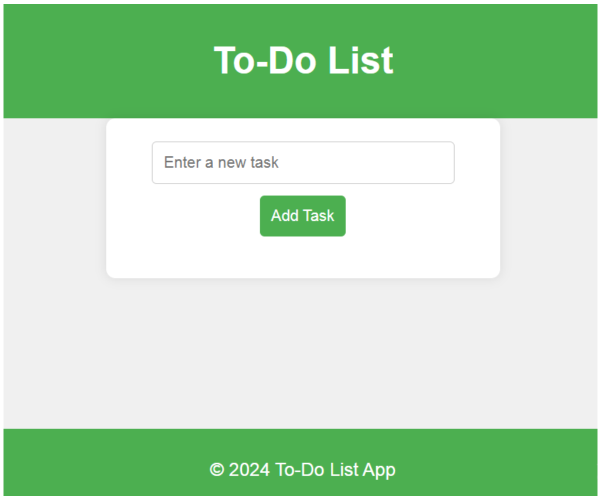
Total Marks: 100 **Web Technologies Laboratory** **Duration: 2 Hours**

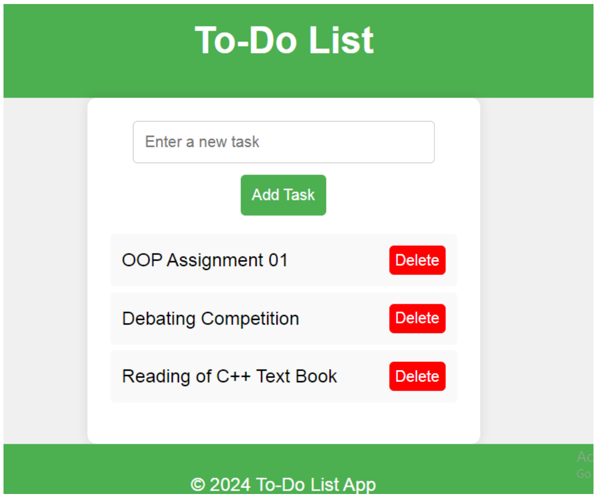
Task 1: Interactive To-Do List Application (1 hour):

Develop a dynamic to-do list application that allows users to add, remove, and mark tasks as completed using HTML, CSS, and JavaScript.

Requirements:

1. **HTML Structure**:
   - Create a basic HTML structure with a header, a main section, and a footer.
   - The main section should contain:
     - An input field to enter a new task.
     - A button to add the task to the list.
     - An unordered list to display the tasks.
2. **CSS Styling**:
   - Style the page to be visually appealing using CSS.
   - Use CSS to highlight completed tasks (e.g., with a strikethrough and a different background color).
   - Add responsive design to ensure the application looks good on both desktop and mobile devices.
3. **JavaScript Functionality**:
   - Implement a function to add a new task to the list when the add button is clicked.
   - Implement a function to remove a task when a delete button next to the task is clicked.
   - Implement a function to mark a task as completed when the task text is clicked.
   - Use local storage to save the tasks so that they persist even when the page is refreshed.
4. **Task Validation**:
   - Validate that the input field is not empty before adding a new task.
   - Display an alert message if the input field is empty when the add button is clicked.
5. **User Feedback**:
   - Provide visual feedback when tasks are added, completed, or deleted (e.g., animations or color changes).





OOP Assignment 01

Debating Competition

Reading of C++ Text Book

Task 2: User Authentication Web Application (1 hour):

Develop a Java web application that allows users to register and authenticate themselves. The application should store user details in a database and validate user credentials during login.

Detailed Requirements

1. **User Registration:**
   - Create a registration form to collect user details:
     1. username,
     2. password,
     3. email, and any other necessary information.
   - Implement a servlet to process the registration form and insert the user details into the database.
2. **User Authentication:**
   - Create a login form for users to submit their username and password.
   - Implement a servlet to authenticate the user by checking the submitted credentials against the data stored in the database.
   - If the credentials match, display a successful login page.
   - If the credentials do not match, display an error message.

Instructions to the Student

1. **Set up the database**:
   - Create the database and the users table as specified.
   - Ensure you have the correct database URL, username, and password in the servlet code.
2. **Implement the registration feature**:
   - Create register.jsp for the registration form.
   - Implement RegisterServlet to handle form submissions and insert user details into the database.
   - Create success and failure pages for registration.
3. **Implement the login feature**:
   - Create login.jsp for the login form.
   - Implement LoginServlet to handle form submissions and authenticate the user against the database.
   - Create success and failure pages for login.
4. **Test your application**:
   - Deploy your application to a servlet container (e.g., Apache Tomcat).
   - Open the application in a web browser and test the registration and login functionalities.
   - Verify that users can register and log in successfully, and appropriate messages are displayed for each scenario.