B.E / B.Tech. PRACTICAL END SEMESTER EXAMINATIONS, NOV/DEC 2024

Third Semester

CD3281 – NAAN MUDHALVAN

(Regulations 2021)

Time : 3 Hours Answer any one Question Max. Marks 100

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Aim/Principle/Apparatus required/Procedure** | **Tabulation/Circuit/ Program/Drawing** | **Calculation & Results** | **Viva-Voce** | **Record** | **Total** |
| **20** | **30** | **30** | **10** | **10** | **100** |

|  |  |
| --- | --- |
| 1. | You are a fitness coach helping a client track their health. The client weighs 70 kg and is 1.75 meters tall. Implement and Calculate their Body Mass Index (BMI) and determine if it falls within the normal range (18.5 - 24.9).  2. You are developing shopping cart feature for an e-commerce website using React. A user adds the following items to their cart:   * **Item 1:** Price = $20, Quantity = 2 * **Item 2:** Price = $15, Quantity = 1 * **Item 3:** Price = $10, Quantity = 3   Write a React component to calculate and display:   1. The total price of all items in the cart. 2. The breakdown of each item's price (price × quantity).   Include logic to dynamically update the total price if quantities change. |
| 2. | If You develop a library management system. The library has a policy that allows users to borrow up to **5 books** at a time for a period of **14 days**. If a book is returned late, a fine of **$2 per day** is applied.  A user has borrowed the following books:   * **Book 1:** Borrowed 18 days ago * **Book 2:** Borrowed 12 days ago * **Book 3:** Borrowed 15 days ago   Calculate:   1. How many books are overdue? 2. The total fine the user needs to pay.   Design the logic to handle these calculations in a library management system. |
| 3. | Integrate an application the Disney API to provide users with information about their favorite Disney characters.  Using the Disney API:   1. Fetch and display details about a specific character (e.g., "Mickey Mouse") including their name, image, and a brief description. 2. Allow the user to search for other characters and display the results dynamically. 3. Bonus: List all movies or shows the selected character has appeared in, if available.   **Task:** Write a React component that:   * Makes an API call to fetch character data. * Displays the character's details in a user-friendly layout. * Includes a search bar for users to find other Disney characters. |
| 4. | Build an **Advice App** using React.js that fetches random advice from an API (e.g., Advice Slip API).  **Requirements:**   1. When the app loads, it should display a random piece of advice from the API. 2. Include a button labeled **"Get New Advice"** that, when clicked, fetches and displays a new piece of advice. 3. Add a feature to allow users to save their favorite advice to a local list. 4. Display the list of saved advice below the main advice section.   **Bonus:** Use localStorage to persist the saved advice so the list remains intact after a page refresh.  Design the logic and React components to fulfill these requirements. |
| 5. | You are tasked with building a **Shopping Cart** feature for an e-commerce application using React.  **Requirements:**   1. **Cart Items:** Display a list of items in the cart. Each item has a name, price, and quantity. Example:    * Item 1: Name = "T-shirt", Price = $20, Quantity = 2    * Item 2: Name = "Jeans", Price = $40, Quantity = 1 2. **Total Price:** Show the total price of all items in the cart (price × quantity). 3. **Update Quantity:** Allow users to increase or decrease the quantity of an item. Update the total price dynamically. 4. **Remove Items:** Provide a button to remove an item from the cart. 5. **Empty Cart:** Include a button to clear all items from the cart. 6. **Responsive UI:** The cart layout should adjust to different screen sizes.   **Bonus:**   * Show a notification if the cart is empty. * Save the cart data in localStorage so it persists after a page refresh.   Write the React components and logic to implement this functionality |
| 6. | Develop a **To-Do List App** using React.js.  **Requirements:**   1. **Add Tasks:** Allow users to add tasks with a title and optional description. 2. **Mark as Complete:** Provide a checkbox for each task to mark it as completed. Completed tasks should be visually distinct (e.g., crossed out or grayed out). 3. **Delete Tasks:** Include a button to delete individual tasks. 4. **Filter Tasks:** Add filters to view **all tasks**, **completed tasks**, or **pending tasks**. 5. **Persist Data:** Use localStorage to save tasks so they remain available after a page refresh.   **Bonus:**   * Allow users to edit a task's title and description. * Include a feature to set a **due date** for each task and highlight overdue tasks in red. * Add drag-and-drop functionality to reorder tasks.   Write the React components and logic to meet these requirements. |
| 7. | You are developing shopping cart feature for an e-commerce website using React. A user adds the following items to their cart:   * **Item 1:** Price = $20, Quantity = 2 * **Item 2:** Price = $15, Quantity = 1 * **Item 3:** Price = $10, Quantity = 3   Write a React component to calculate and display:   1. The total price of all items in the cart. 2. The breakdown of each item's price (price × quantity).   Include logic to dynamically update the total price if quantities change. |
| 8. | You are building a **Shopping Cart** application using React and Redux for state management.  **Requirements:**   1. **Cart Items State:**    * Manage the cart's state using Redux.    * Each item in the cart should have an ID, name, price, and quantity. 2. **Features:**    * **Add to Cart:** Allow users to add items to the cart.    * **Remove from Cart:** Provide a button to remove an item.    * **Update Quantity:** Users can increase or decrease the quantity of items in the cart.    * **View Total:** Dynamically calculate and display the total price of all items in the cart. 3. **Redux Actions and Reducers:**    * Define actions for adding, removing, and updating cart items.    * Create a reducer to handle state updates for the cart. 4. **UI Components:**    * Display the list of cart items with their name, price, quantity, and subtotal (price × quantity).    * Show the total price at the bottom of the cart.   **Bonus:**   * Use Redux middleware like redux-thunk or redux-saga to simulate fetching product data from an API before adding it to the cart. * Persist the cart state in localStorage or Redux Persist so it remains intact after a page refresh.   Implement the Redux store, actions, reducers, and React components to achieve this functionality.  4o |

**INTERNAL EXAMINER EXTERNAL EXAMINER**