

CREATIVE DESIGN & MULTIMEDIA INSTITUTE

React Js Practical Exam

Q1. Below is an exam structure for a React.js task involving the creation of a To-Do list. The task includes implementing basic CRUD operations (Create, Read, Update, Delete) using useState. [20]

Task: To-Do List with CRUD Operations

Requirements:

1. Functionalities:

- Create a web application where users can add, edit, and delete tasks.
- Each task should have a title and a checkbox to mark it as completed.
- Provide a way to filter tasks (e.g., show all tasks, show only completed tasks, show only uncompleted tasks).
- Form Validation

2. User Interface:

- Design a simple and user-friendly interface for the application.
- Use CSS for styling. You can also use a CSS framework like Bootstrap if you're comfortable with it.

3. Component Structure:

- Create at least two React components: one for the task list and another for individual tasks.

4. Functionality Details:

- When a user adds a task, it should appear in the list with an unchecked checkbox.
- The user should be able to mark a task as completed by checking the checkbox.
- The user should be able to edit the title of a task.
- The user should be able to delete a task.

This task is designed to help you get hands-on experience with React.js and familiarize yourself with component structure, state management, and basic interactions. Remember to ask for help or clarification if you encounter any difficulties! Happy coding!

CREATIVE DESIGN & MULTIMEDIA INSTITUTE

React Js Practical Exam

Q2. Develop a currency converter application that allows users to input an amount in one currency and convert it to another. For the sake of this challenge, you can use a hard-coded exchange rate. Take advantage of React state and event handlers to manage the input and conversion calculations. [20]

USD: 1,

EUR: 0.85,

INR: 74.71, // Indian Rupee

PKR: 292.75, // Pakistani Rupee

NPR: 119.53, // Nepalese Rupee

CAD: 1.26, // Canadian Dollar make question for exam.

CREATIVE DESIGN & MULTIMEDIA INSTITUTE

React Js Practical Exam

Q3: Below is a task involving the creation of a Tic-Tac-Toe game using React.js. The task includes implementing the game board, state management with useState, and functionalities for the game. [20]

Task: Tic-Tac-Toe Game

Requirements:

Game Board Component (TicTacToeBoard):

- Create a TicTacToeBoard component that displays a 3x3 grid for the Tic-Tac-Toe game.
- Use useState to manage the state of the board, representing X, O, or an empty cell.

Cell Click Functionality:

- Implement a click event on each cell of the TicTacToeBoard.
- Toggle between X and O when a cell is clicked.
- Ensure that clicking on an already filled cell does not change its value.

Game Status Display:

- Display the current player's turn (X or O).
- If a player wins, display a message indicating the winner.
- If the game ends in a draw, display a draw message.

Reset Button:

- Implement a "Reset" button that resets the game board to its initial state.
- The button should clear the board and display the initial player's turn.

Additional Challenges (Optional):

- Implement the game logic to check for a winning condition.
- Add animations or visual cues for a better user experience.
- Keep track of the game's history and allow players to undo moves.

CREATIVE DESIGN & MULTIMEDIA INSTITUTE

React Js Practical Exam

Q4:Below is an exam structure for a React.js task involving a number puzzle game. This task focuses on creating a grid of numbers from 1 to 9 in a random order, and the goal is to arrange them in ascending order by swapping adjacent numbers. [20]

Task: Number Puzzle Game

Requirements:

Number Puzzle Grid Component (NumberPuzzleGrid):

- Create a NumberPuzzleGrid component that displays a 3x3 grid containing numbers from 1 to 9 in a random order.
- Utilize useState to manage the state of the grid.

Grid Display:

- Display the numbers in a visually appealing format within the grid.
- Each number should be a clickable element.

Swap Numbers Functionality:

- Implement a click event on each number in the grid.
- Clicking on a number should swap its position with the adjacent empty cell (if there's an adjacent empty cell).
- The empty cell should be visually represented.

Winning Condition:

- Implement a winning condition to check if the numbers are arranged in ascending order (1 to 9).
- Display a congratulatory message when the player wins.

Reset Button:

- Create a "Reset" button that shuffles the numbers in the grid to create a new puzzle.
- This button should also clear any winning messages.

CREATIVE DESIGN & MULTIMEDIA INSTITUTE

React Js Practical Exam

Subject: Best of Luck for Your Exam! 🌟

Dear Students,

As you embark on this exam, we want to wish you the very best of luck! 🍀
You've worked hard, prepared diligently, and now it's time to showcase your knowledge and skills.