Practical Assignment -101 Frontend Development using React js.

Question 1: Create a Dynamic Quiz Application

Design and implement a dynamic quiz application using HTML, CSS, and JavaScript. The application should have the following features:

1. Display a question and multiple choice options on the webpage.
2. Allow the user to select one option as their answer.
3. Provide immediate feedback to the user on whether their answer is correct or incorrect.
4. Keep track of the user's score as they progress through the quiz.
5. Present the next question once the user submits their answer.
6. Display the final score and a message at the end of the quiz.

Consider using the following structure for your code:

- HTML: Create the necessary elements to display the question, options, feedback, score, and end message.
- CSS: Style the webpage to make it visually appealing.
- JavaScript: Implement the logic for displaying questions, validating answers, calculating the score, and transitioning between questions.

Feel free to customize the quiz content and design according to your preferences. Remember to use appropriate HTML tags, CSS properties, and JavaScript functions to achieve the desired functionality.

Question 2: Create a ToDo List Application

Design and implement a ToDo list application using HTML, CSS, and JavaScript. The application should have the following features:

1. Display an input field and a "Add" button to add new tasks.
2. When the user clicks the "Add" button, the task should be added to the list below. 3. Each task should have a checkbox to mark it as completed and a delete button to remove it from the list.
4. Styling the list to differentiate between completed and pending tasks.
5. Display a counter that shows the total number of pending tasks.
6. Optional: Implement a feature to edit existing tasks.

Consider using the following structure for your code:

- HTML: Create the necessary elements to display the input field, buttons, task list, and counter. • CSS: Style the webpage and list items to make them visually appealing and distinguish between completed and pending tasks.
- JavaScript: Implement the logic for adding tasks, marking them as completed, deleting tasks, updating the counter, and handling any additional features you choose to implement.

Feel free to customize the design and functionality of the ToDo list application according to your preferences.

Question 3: Create small website and perform JavaScript for the following.

Q. 1: Create a small website with a button labeled "Generate Random Number". When the button is clicked, display a random number between 1 and 100 in an alert message.

Q. 2: Create a form with an input field where users can enter their email address. Add a button that, when clicked, checks if the entered email address is valid (contains the '@' symbol). If it is valid, display an alert message saying "Email address is valid". Otherwise, display an alert message saying "Invalid email address".

Q. 3: Create a simple image slideshow. Display three images on the webpage, and add two buttons labeled "Previous" and "Next". When the "Previous" button is clicked, display the previous image in the sequence, and when the "Next" button is clicked, display the next image in the sequence.

Q. 4: Create a small calculator with an input field and three buttons labeled "Square", "Square Root", and "Factorial". When the user enters a number and clicks one of the buttons, perform the corresponding operation on the entered number and display the result in an alert message.

Q. 5: Create a dropdown menu with three options: "Red", "Green", and "Blue". When an option is selected, change the background color of the webpage to the corresponding selected color.

Q. 6: Create a stopwatch functionality. Display a timer on the webpage that starts at 00:00:00. Add buttons labeled "Start", "Stop", and "Reset". When the "Start" button is clicked, start the timer. When the "Stop" button is clicked, pause the timer. When the "Reset" button is clicked, reset the timer to 00:00:00.

Q.7: Create an image gallery with three images. When an image is clicked, display the image in a larger size in a modal window.

Q.8: Create a small calculator with two input fields and four buttons: "Add", "Subtract", "Multiply", and "Divide". When a button is clicked, perform the corresponding mathematical operation on the two input values and display the result in an alert message.

Q.9: Create a countdown timer that starts at 10 and counts down to 0. Display the current countdown value on the webpage. When the countdown reaches 0, display an alert message saying "Time's up!".

Q.10: Create a form with two input fields: "Username" and "Password". Add a button that, when clicked, checks if the username is "admin" and the password is "password". If both conditions are true, display an alert message saying "Login Successful". Otherwise, display an alert message saying "Login Failed".

Default initializes default value to constructor.