Practical Assignment -2 101 Frontend Development using React js.

jQuery Practical Questions:

1. Create a simple HTML form with fields for name, email, and a submit button. Field name and email should have relevant validation. Use jQuery to prevent the form from submitting when the submit button is clicked and instead display an alert with the form data.
2. Create an HTML page that includes two dropdown menus (combos): one for selecting states and another for displaying cities. When the user selects a state from the primary combo, the secondary combo (cities) should dynamically update to display cities relevant to the selected state.
3. Write jQuery code to toggle the visibility of a div element when a button is clicked. The div should start as hidden, and clicking the button should alternate between showing and hiding it.
4. Create a basic image slider using jQuery. Users should be able to click "Next" and "Previous" buttons to navigate between a series of images displayed on a webpage.
5. Implement a simple client-side validation using jQuery. Check if a text input field is empty when a form is submitted. If it's empty, display an error message next to the input field.

AJAX Practical Questions:

6. Develop a webpage that retrieves and displays data from a public JSON API (e.g., a list of books, movies, or weather information) using AJAX. Display this data in a visually appealing manner on the page.
7. Create an autocomplete search feature using AJAX and jQuery. When users start typing a query into a search input field, dynamically fetch and display matching results from an API as dropdown suggestions.
8. Build a real-time chat application using AJAX to send and receive messages without refreshing the page. Messages should be displayed on the page as they are sent and received.
9. Design a "Load More" button that, when clicked, fetches additional content from an API and appends it to an existing list on the webpage. Implement pagination using AJAX.
10. Develop a weather app that allows users to enter a city name and retrieve weather data from a weather API using AJAX. Display the temperature, weather description, and an icon representing the weather condition. Allow users to refresh the weather data without reloading the entire page.