# Charotar University of Science and Technology [CHARUSAT]

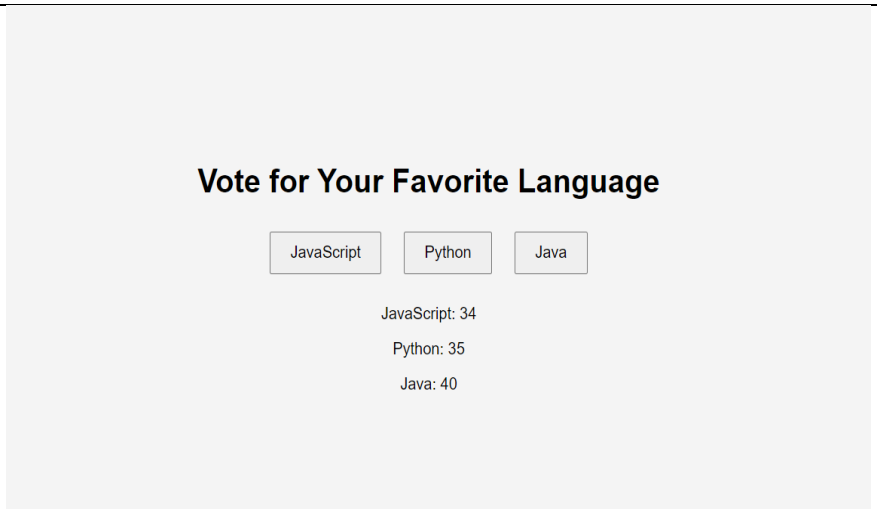
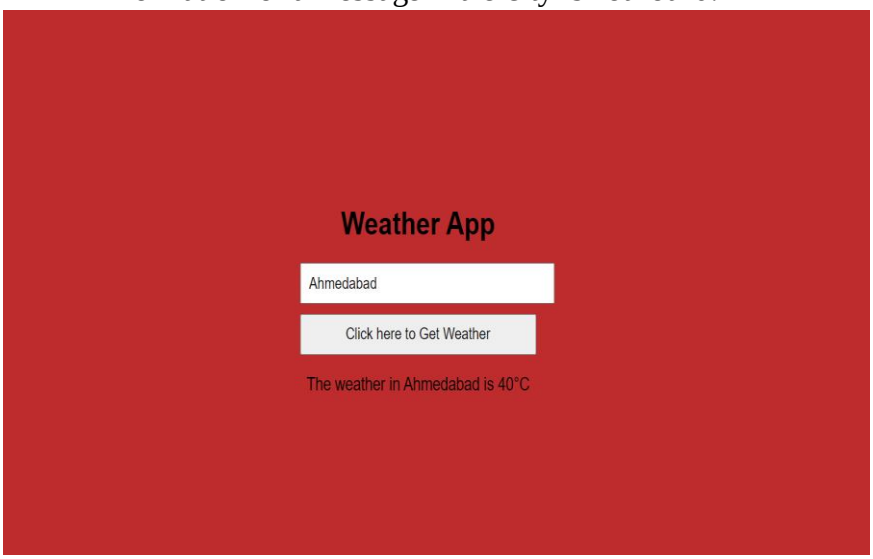
## Chandubhai S. Patel Institute of Technology [CSPIT]

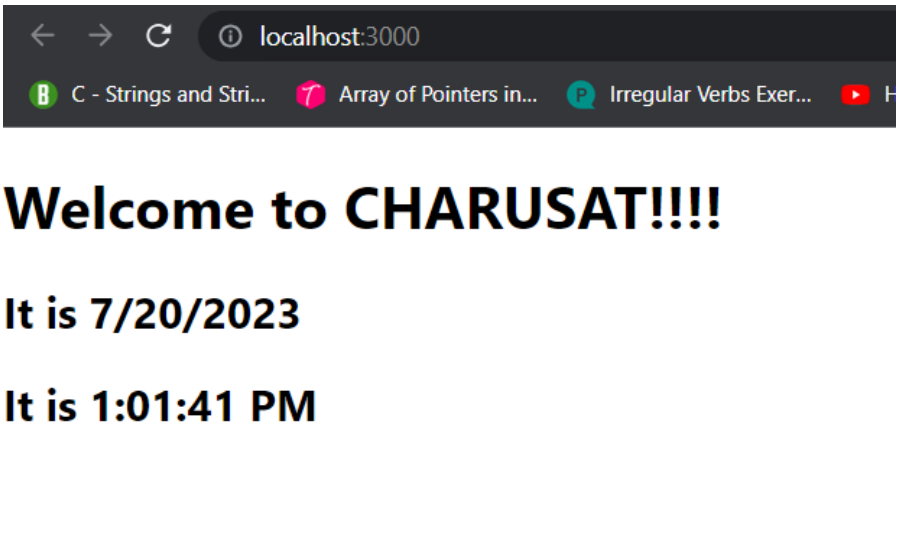
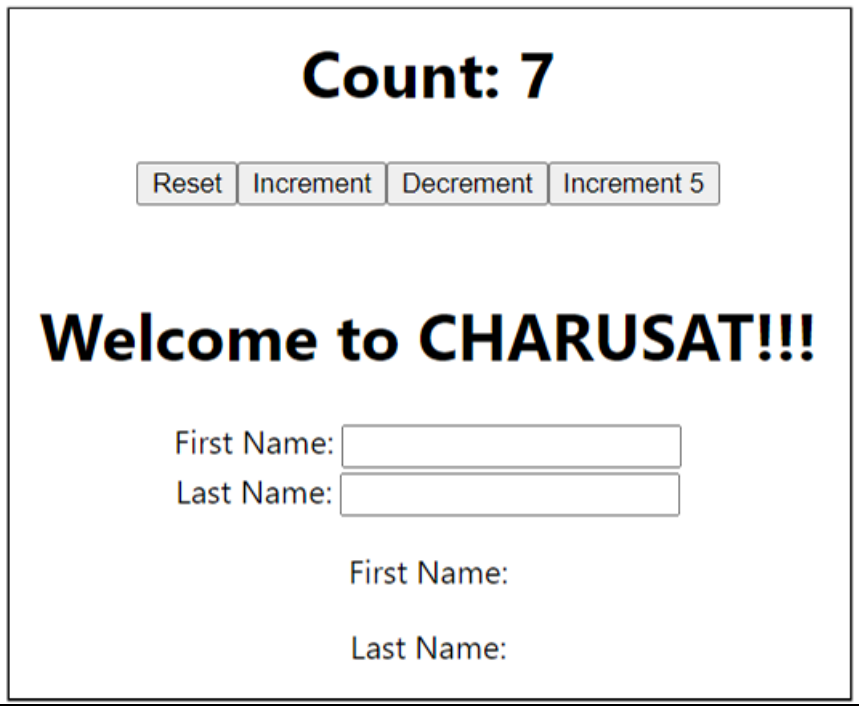
## Department of Computer Science & Engineering

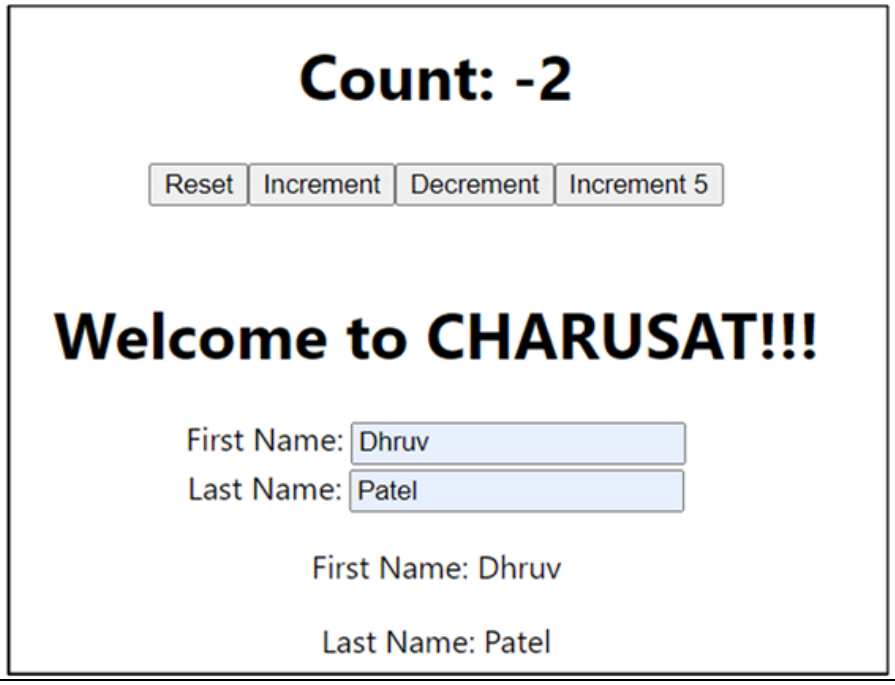
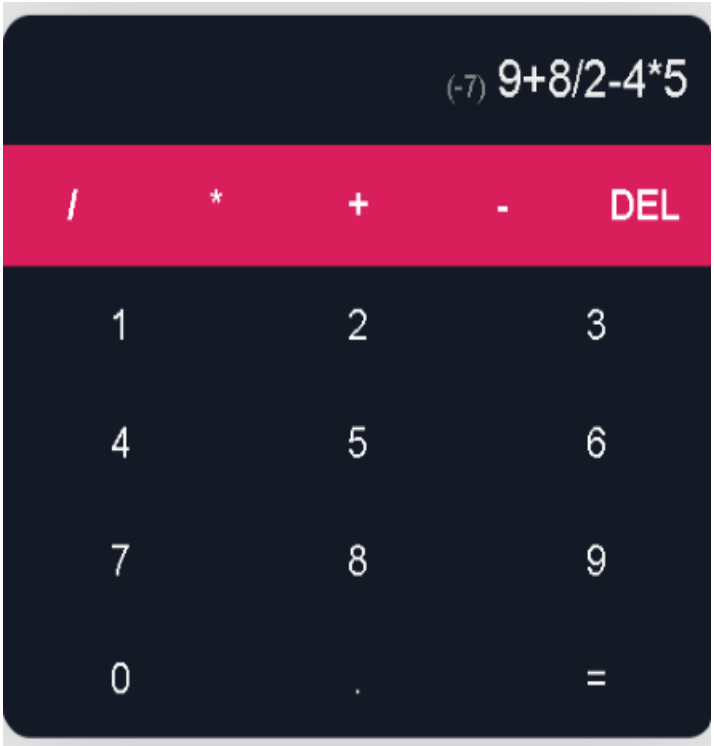
## Practical List

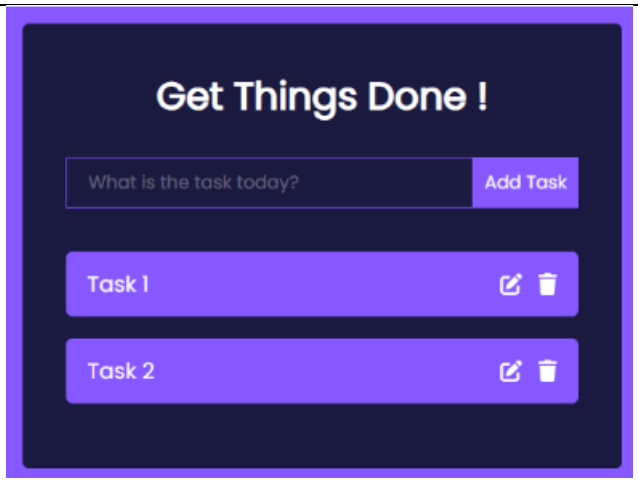
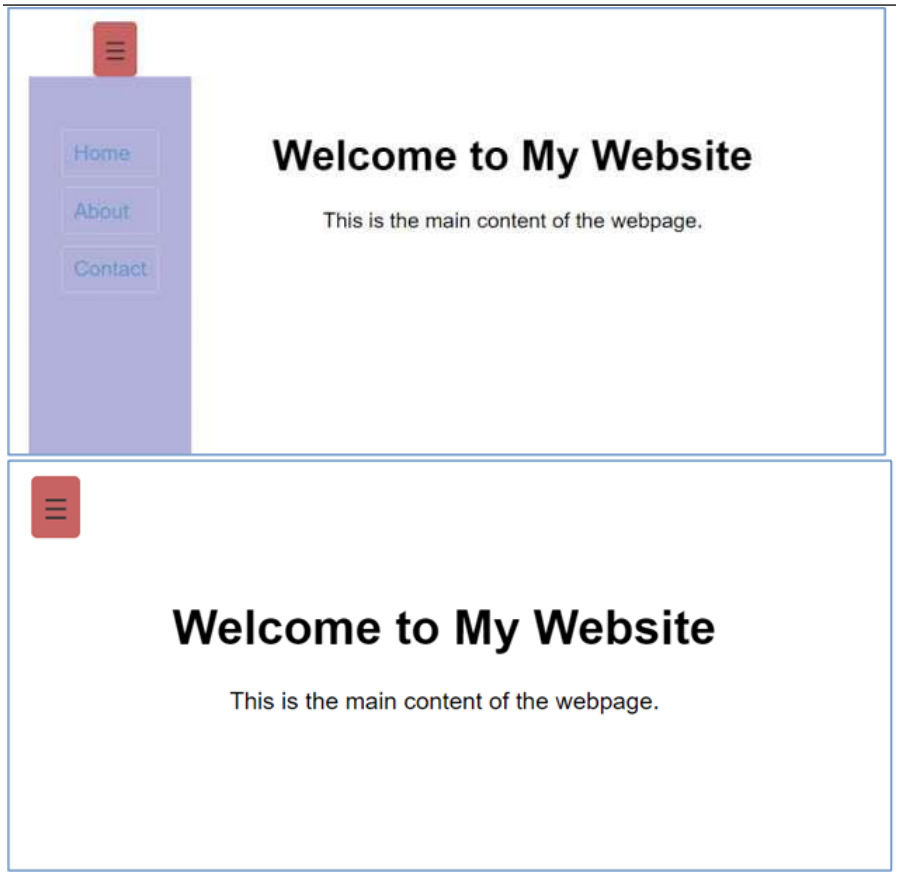
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Subject code | : | CSE304 | Semester | : | 5 | Academic Year | : | 2024-25 |
| Subject name | : | Full Stack Development | | | | | | |

| Sr. No. | Aim | Hrs. | CO |
|---|---|---|---|
| | **ReactJs Tasks** | | |
| 1. | Create a real-time voting system where users can vote on a poll and see the results updated in real-time using only JavaScript, HTML, and CSS.<br>**HTML:** A simple poll interface with buttons to vote and display the results.<br>**CSS:** Styles the poll and results.<br>**JavaScript:**<br>Defines a vote function to update the local votes.<br>Updates the vote counts in the UI.<br>Simulates real-time voting by randomly incrementing votes.<br>**Notes:**<br>• **Votes object:** Keeps track of the current vote counts for each language. It initializes each language with 0 votes.<br>• **vote function:** This function is called when a button is clicked. It increments the vote count for the selected language and calls updateVotes to refresh the displayed vote counts.<br>• **updateVotes function:** Updates the displayed vote counts by setting the text content of the spans in the results section to the current vote counts.<br>• **setInterval:** Simulates real-time updates by randomly incrementing the vote counts every 2 seconds. This mimics votes coming in from other users in real time. | 3 | 1 |

| | | | |
|---|---|---|---|
| |  | | |
| 2. | Create a simple weather application that displays a hardcoded temperature for a given city. This simple weather application demonstrates basic HTML structure for user input, CSS styling for layout and appearance, and JavaScript functionality to handle user interactions and display dynamic content. It provides a foundational example of building an interactive web application using essential front-end technologies.<br>Technologies Used: HTML, CSS, JavaScript (ES6)<br><br>**Notes:**<br>• **weatherData:** This object holds hardcoded weather information for specific cities. In a real-world scenario, this data would typically come from an API.<br>• **addEventListener:** Listens for a click event on the Get Weather button.<br>• **Fetching Weather:** When the button is clicked, it retrieves the city entered by the user, checks if weather data exists in weatherData, and displays the corresponding weather information or a message if the city is not found.<br>  | 3 | 1 |

| | | | |
|---|---|---|---|
| 3. | Create a React page that displays a Welcome message with the current local date and time which changes every second. Use the React JSX concept.<br>Note: Use the setInterval function to update the time.<br>  | 3 | 2 |
| 4. | Create a Counter App that includes the functionality of Increment, Decrement, Reset, and Increment Five.<br>The app also has two text boxes for first and surname names, the contents of which are shown as text on the same page. Utilize the React Hooks idea<br>  | 4 | 2 |

| | | | |
|---|---|---|---|
| |  | | |
| 5. | Create a calculator React app involves several steps, from setting up the React environment to implementing the logic for basic arithmetic operations. Here's a detailed explanation of creating a simple calculator app with addition, subtraction, multiplication, and division capabilities.<br>  | 4 | 2, 3 |
| 6. | Build a To-do List using React Hooks involves creating a user interface that allows users to add, remove, and delete tasks. This can be achieved using React's state management capabilities through hooks like useState and event handling for managing tasks. | 4 | 2, 3 |

| | | | |
|---|---|---|---|
| |  | | |
| 7. | Creating a React Sidebar Navigation Menu involves using React Hooks to manage the state and handle the functionality of the sidebar, such as opening and closing the menu, and navigating between different sections of the webpage.<br>  | 4 | 2, 3 |
| **NodeJs and ExpressJs Tasks** | | | |
| 8. | Develop a simple web-based counter application that increments and decrements a counter value. | 2 | 4 |
| 9. | Write a Program to Demonstrate Express JS Server. | 2 | 4, 5 |
| 10. | Write a Program to Read Text File and Display Content on Server. | 1 | 5 |
| 11. | Create New Application in Express JS with Basic Route. | 1 | 4, 5 |

| | | | |
|---|---|---|---|
| 12. | In Express JS Take 2 Numbers from User and preform arithmetic Operation. | 2 | 5 |
| 13. | Create Sum of 2 Numbers Using post method and EJS in Express. | 2 | 5 |
| 14. | Create File Upload Program in Express JS. Allowed Only 2MB Size File. | 3 | 5 |
| 15. | Create Session Program with Create, Read, Remove Operation. | 3 | 4, 5 |
| 16. | Create Contact Form and on Click Event send Data on your EmailID. | 3 | 5 |
| 17. | CRUD Operation using ExpressJS and MongoDB | 4 | 4, 5 |
| 18. | Create Rest API CRUD using MongoDB and Express JS | 5 | 4, 5 |