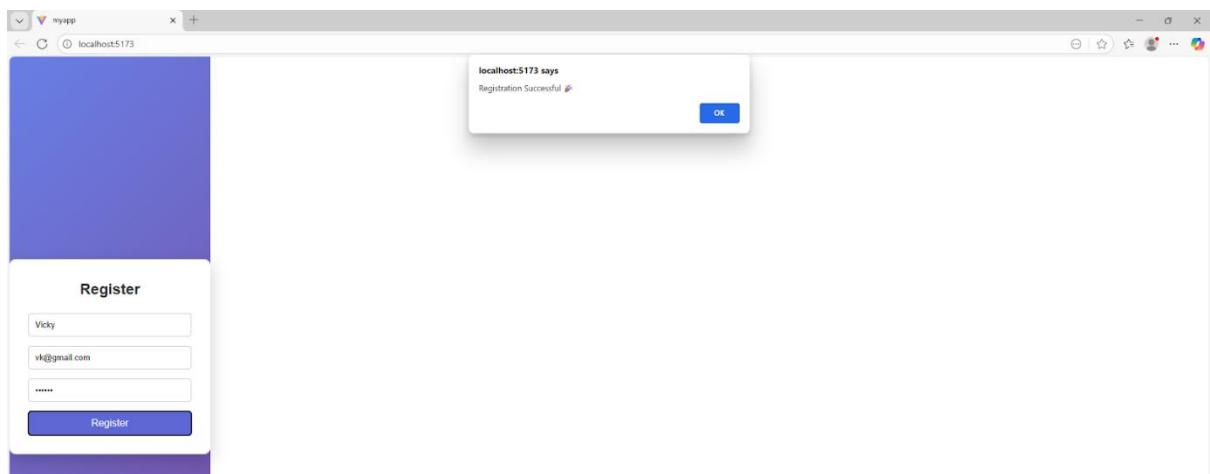


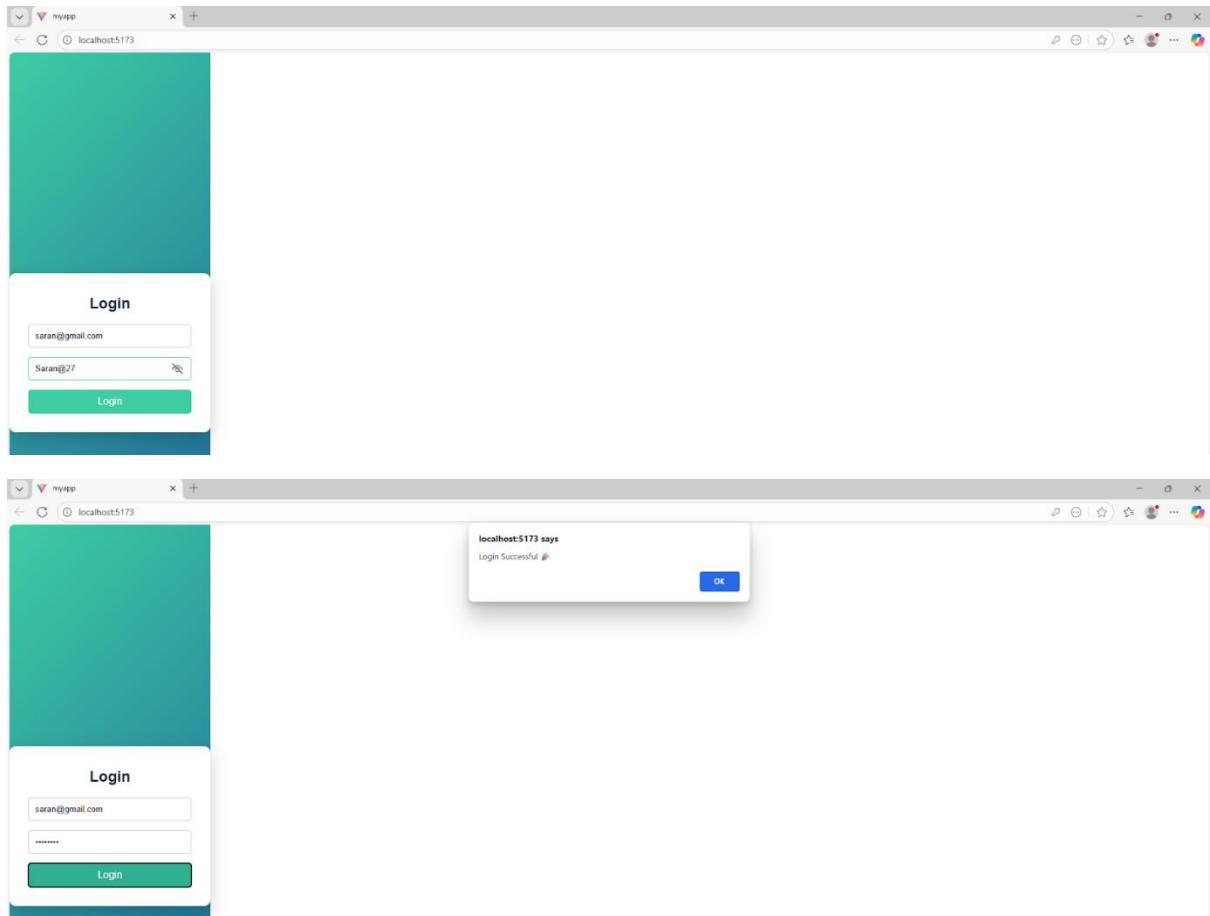
## OUTPUT:

### 1. Create a Registration Form

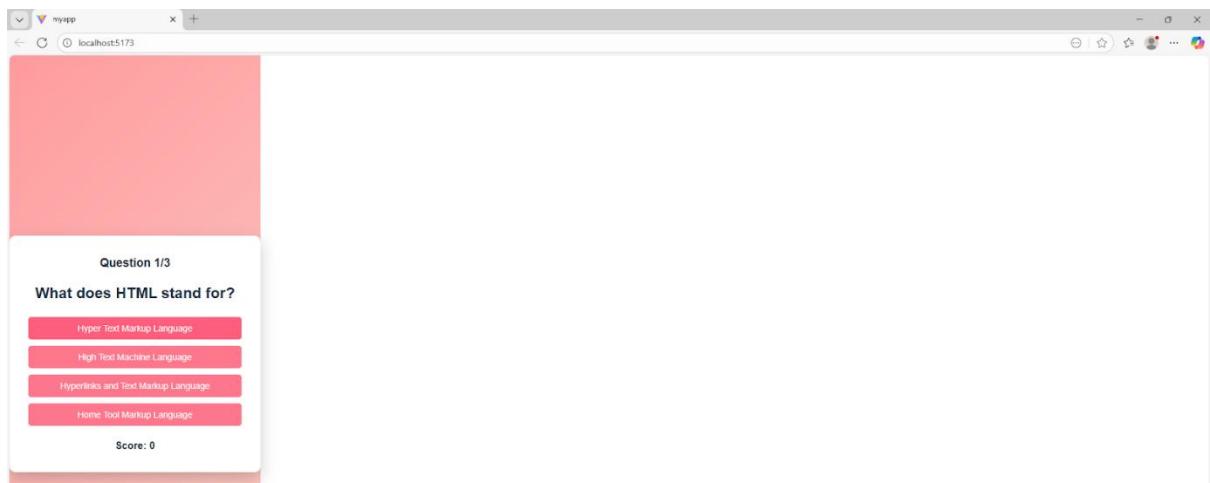
A screenshot of a web browser window titled "myapp" with the URL "localhost:5173". The page has a purple header and footer. In the center, there is a white "Register" form with three input fields: "Vicky" (text), "vk@gmail.com" (text), and "....." (password). Below the inputs is a blue "Register" button.

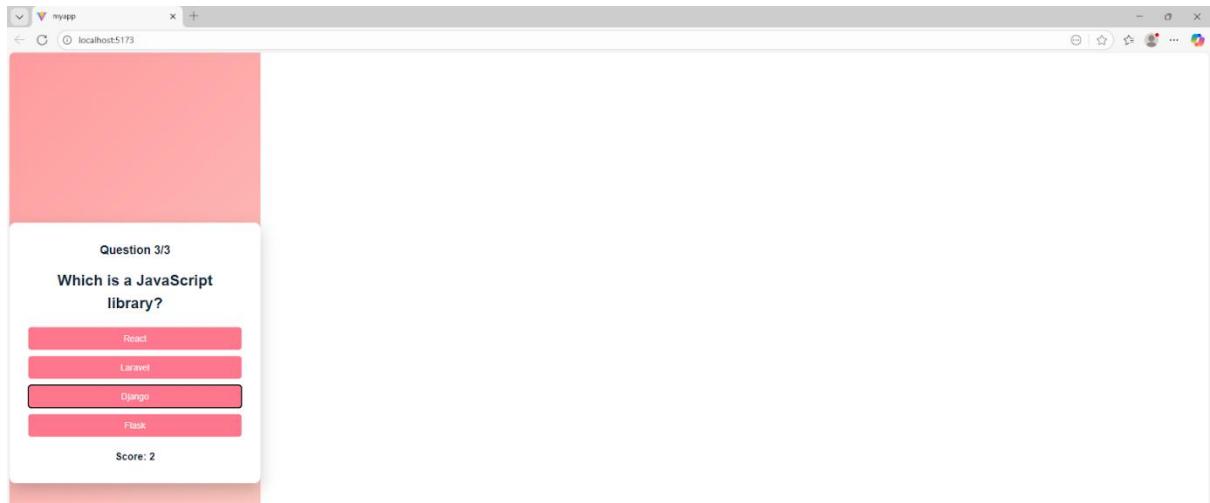
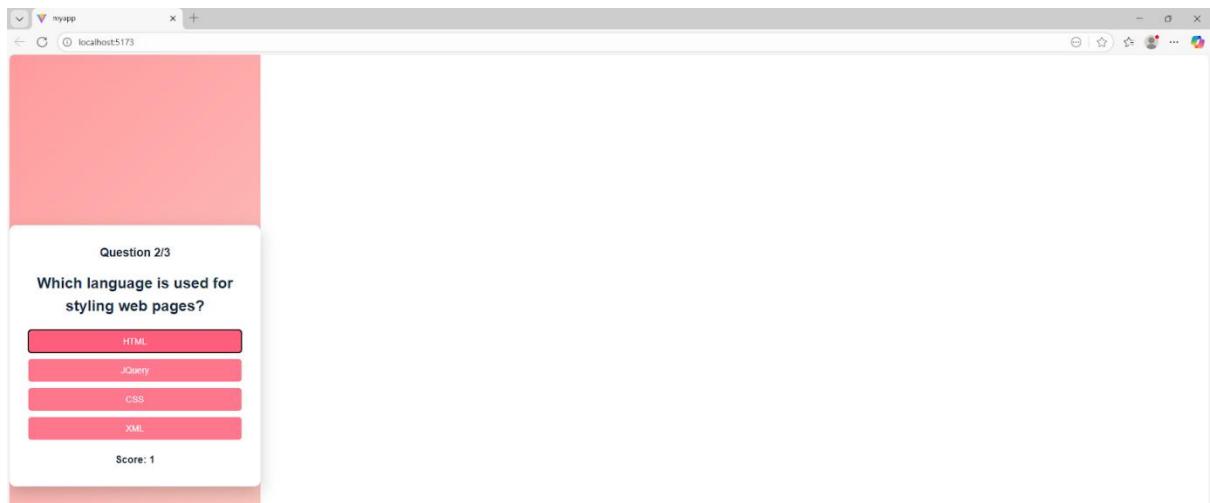


## 2. Create a Login Form with Validation

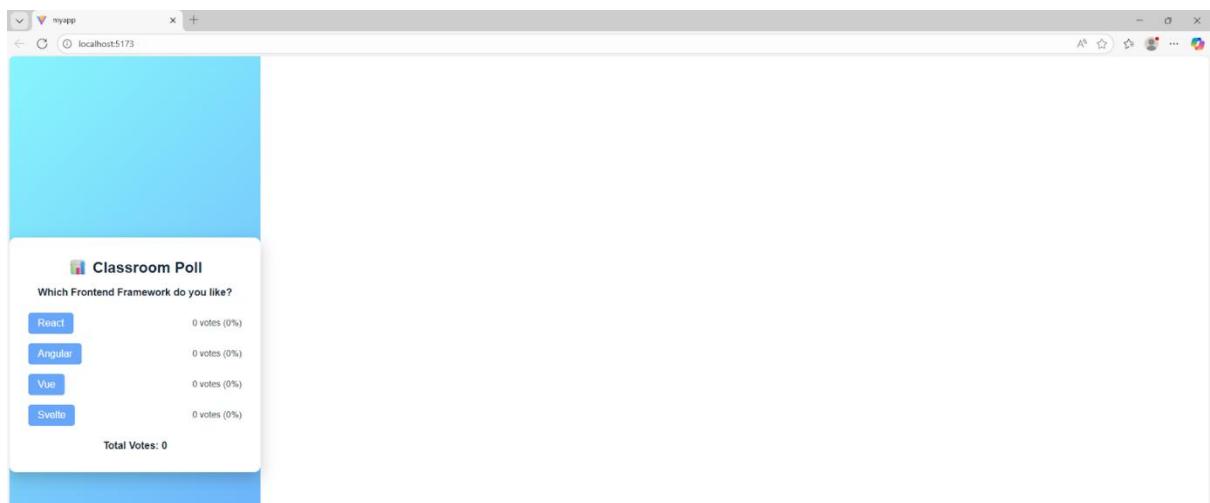


## 3. Design a mini online quiz interface that updates scores dynamically.





#### 4. Implement a real-time polling system for classroom use





## 5. Design a feedback form interface that displays submitted data dynamically on the screen

The screenshot shows a web browser window titled "myapp" with the URL "localhost:5173". The main content area has a light purple background. On the left, there is a white sidebar with a dark purple header containing the text "Feedback Form" and a small icon. Below this, there are two input fields: "Your Name" and "Your Feedback", both currently empty. A blue "Submit" button is located below the feedback field. At the bottom of the sidebar, the text "No feedback yet" is displayed.

The screenshot shows a web browser window titled "myapp" with the URL "localhost:5173". The main content area has a light purple background. On the left, there is a white sidebar with a dark purple header containing the text "Feedback Form" and a small icon. The "Your Name" field now contains the value "Harish". The "Your Feedback" field contains the text "Thank you for such a nice experience..!". A blue "Submit" button is located below the feedback field. At the bottom of the sidebar, the text "No feedback yet" is displayed.

A screenshot of a web browser window titled "myapp" showing a feedback form. The form has fields for "Your Name" and "Your Feedback", and a "Submit" button. Below the form, there are two messages: "Harish" with the text "Thank you for such a nice experience...!", and "Vijay" with the text "Thank you for your greatest support".

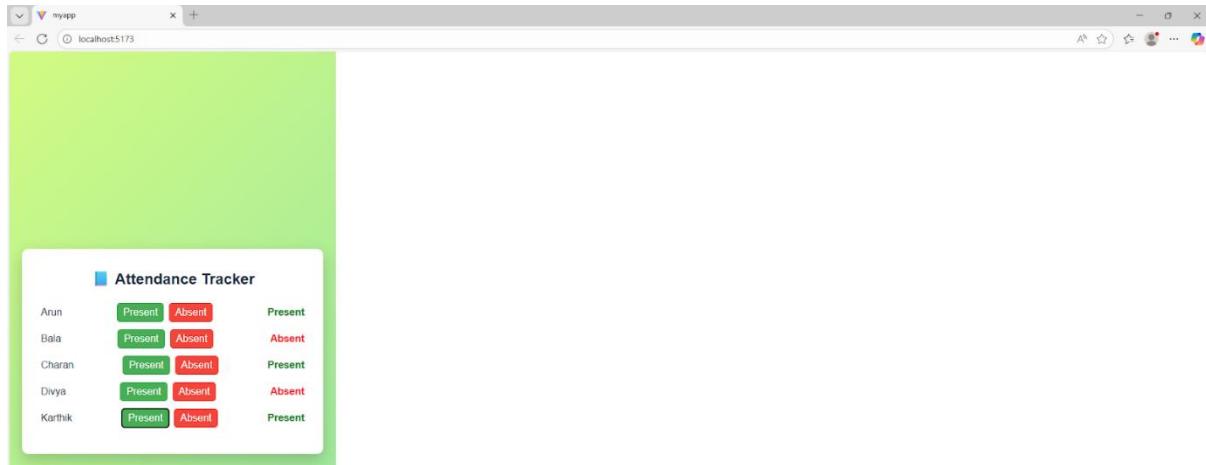
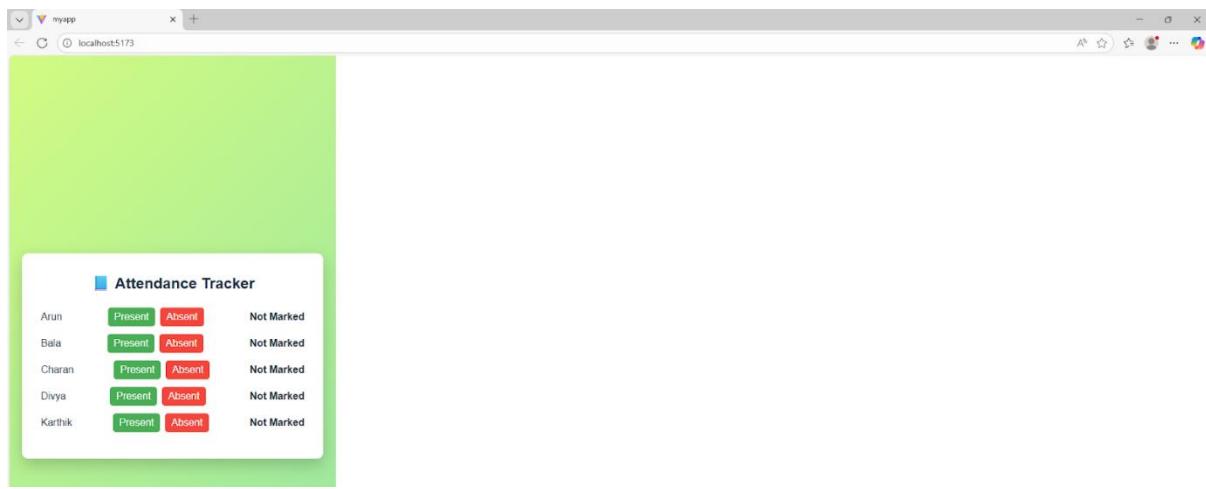
6. Implement a simple course enrollment form that updates the enrolled course list in real time.

A screenshot of a web browser window titled "myapp" showing a course enrollment form. The form includes fields for "Student Name" and "Select Course", and an "Enroll" button. Below the form, a section titled "Enrolled Students" displays the message "No enrollments yet".

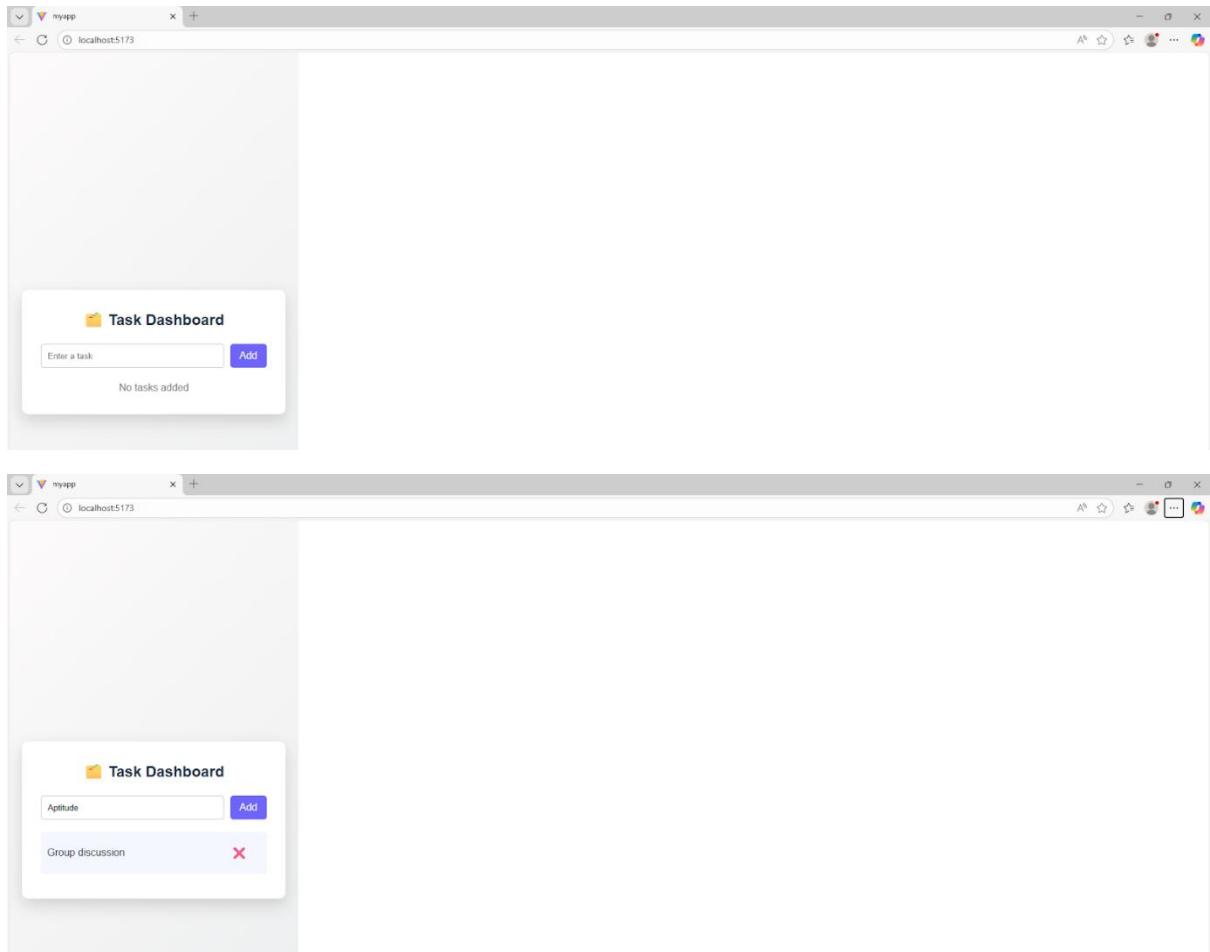
A screenshot of a web browser window titled "myapp" showing the same course enrollment form. The "Student Name" field now contains "Kabilan". The "Select Course" dropdown menu is open, showing options: "React Basics", "JavaScript Fundamentals", "Web Development", "Python Programming", and "Data Structures". The option "React Basics" is highlighted with a dark gray background.



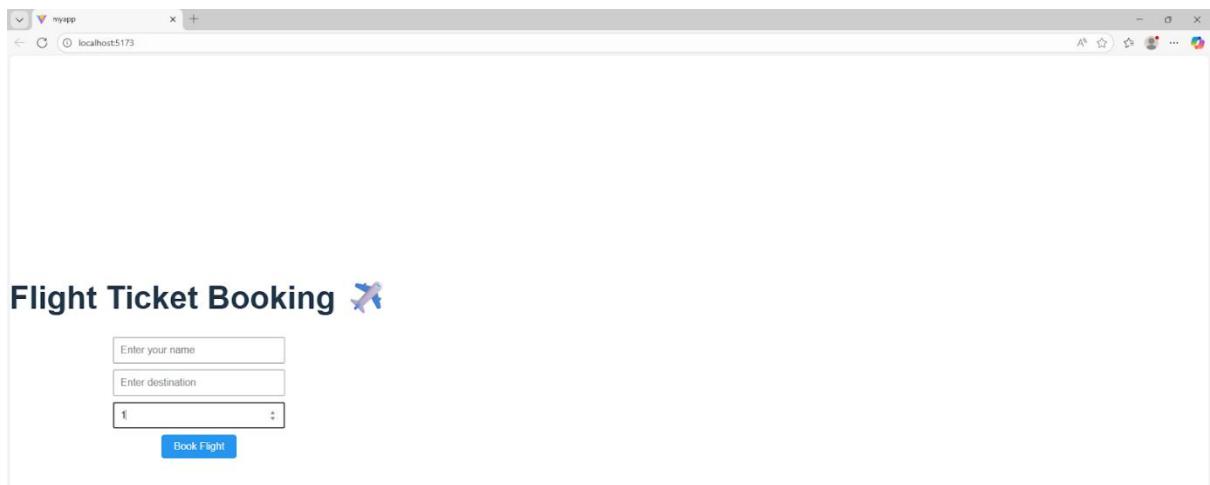
## 7. Simulate an attendance tracker interface that marks students present or absent dynamically



## 8. Develop a simple task list dashboard that allows adding and removing tasks.



## 9. Flight Ticket Booking using JavaScript Dialog Boxes.



Flight Ticket Booking 

Gowtham  
Paris  
2

Book Flight.

## 10. Hotel Reservation System using JavaScript Dialog Boxes

Hotel Reservation System 

Enter your name  
1  
Single - \$50/night

Reserve Room

Hotel Reservation System 

Aravind  
2  
Double - \$90/night  
Single - \$50/night  
Double - \$90/night  
Suite - \$150/night

The image displays three sequential screenshots of a "Hotel Reservation System" application, likely built with Node.js and Express.js, showing the user interface and server-side confirmation messages.

**Screenshot 1:** The application's main form. It contains three input fields: "Name" (Aravind), "Nights" (2), and "Room Type" (Double). Below these is a "Reserve Room" button. A JavaScript dialog box is overlaid, titled "localhost:5173 says", containing the message: "Please confirm your reservation:  
Name: Aravind  
Room Type: Double  
Nights: 2  
Total Price: \$180". It has "OK" and "Cancel" buttons.

**Screenshot 2:** The same form is shown again. A second JavaScript dialog box is overlaid, titled "localhost:5173 says", containing the message: "Reservation confirmed! Enjoy your stay, Aravind!" It has an "OK" button.

**Screenshot 3:** The application's main form is shown once more. A third JavaScript dialog box is overlaid, titled "Reservation Confirmed! ✓", containing the same confirmation message as the previous dialog: "Reservation confirmed! Enjoy your stay, Aravind!". It has an "OK" button. Below this dialog, the form fields and "Reserve Room" button are visible.

## 11. Task 11 Online Cab Booking Application using JavaScript Dialog Boxes

## Online Cab Booking 🚕

Enter your name  
Akash

Pickup location  
Gandhipuram

Drop location  
Singanullur

Car Type  
Mini - \$5/km

Quantity  
1

**Book Cab**

## Online Cab Booking 🚕

Enter your name  
Akash

Pickup location  
Gandhipuram

Drop location  
Singanullur

Car Type  
Mini - \$5/km

Quantity  
1

**Book Cab**

## Online Cab Booking 🚕

Enter your name  
Akash

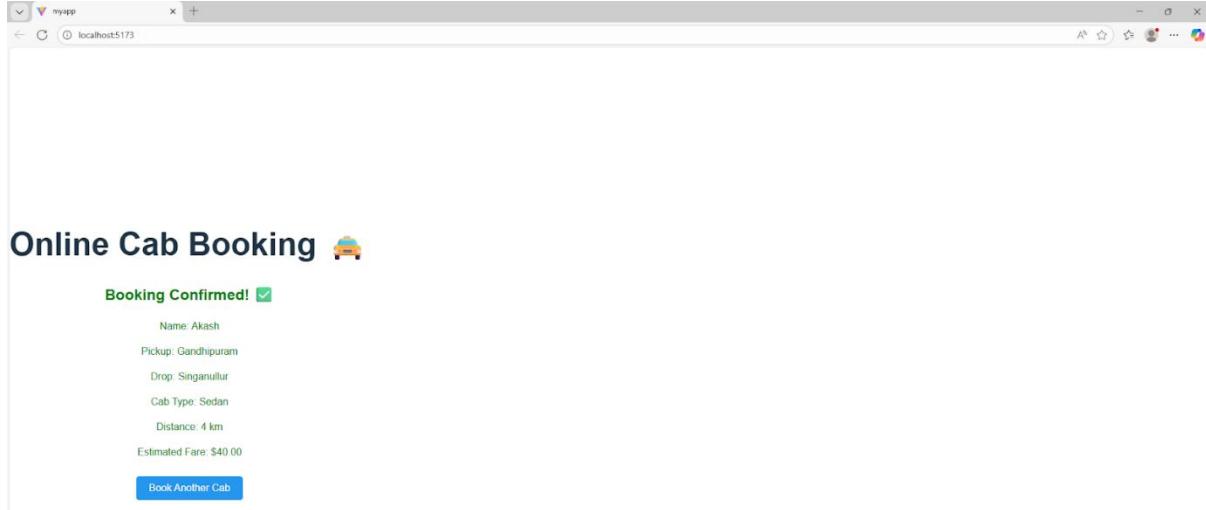
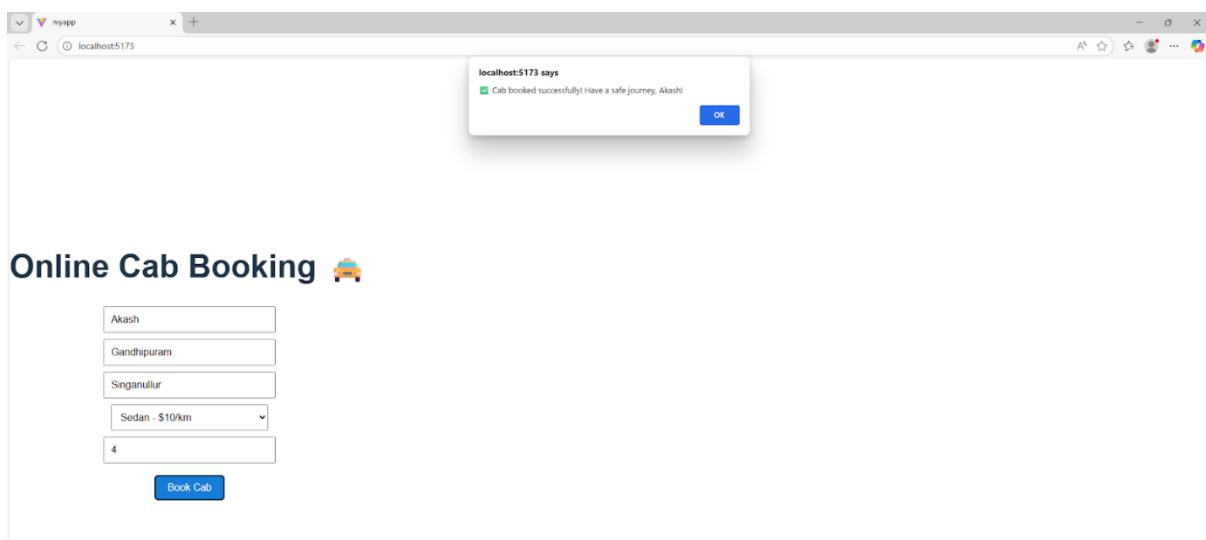
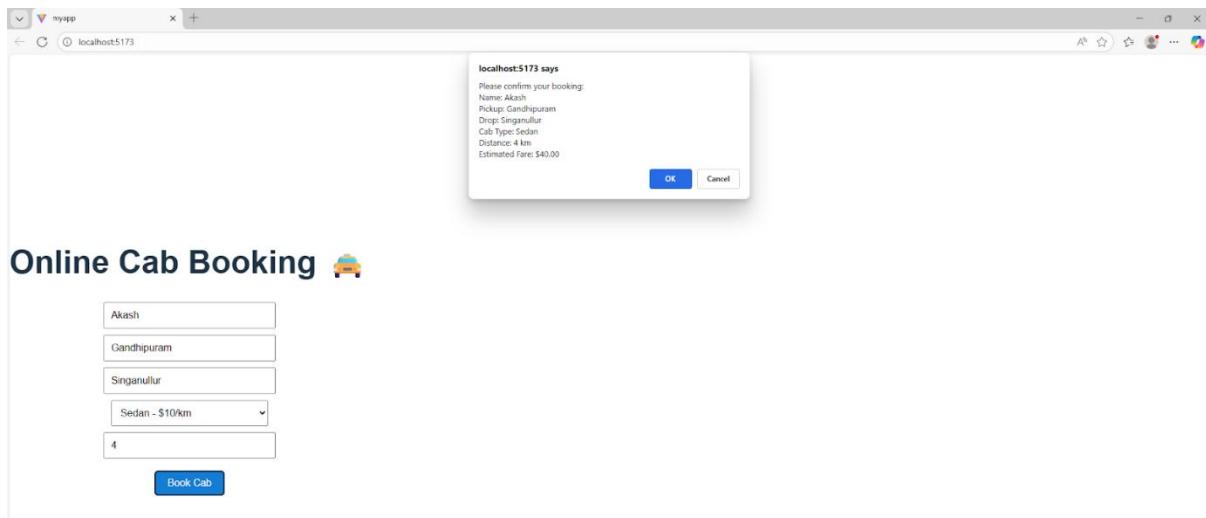
Pickup location  
Gandhipuram

Drop location  
Singanullur

Car Type  
Sedan - \$10/km

Quantity  
4

**Book Cab**



## 12. Task 12 Design an event registration interface with confirmation alerts.

The image consists of three vertically stacked screenshots of a web browser window titled "myapp" at "localhost:5173".

**Screenshot 1:** The first screenshot shows the initial state of the "Event Registration" form. It contains three input fields: "Enter your name" with placeholder "Bala", "Enter your email" with placeholder "bala@gmail.com", and a dropdown menu set to "Workshop". Below the form is a green "Register" button.

**Screenshot 2:** The second screenshot shows the same form after the user has filled in the name and email fields and selected "Workshop" from the dropdown. A confirmation dialog box is overlaid on the page, containing the message: "localhost:5173 says", followed by "please confirm your registration:", "Name: Bala", "Email: bala@gmail.com", and "Event Type: Seminar". The dialog has "OK" and "Cancel" buttons.

**Screenshot 3:** The third screenshot shows the final state of the form after the user has registered. The dropdown menu now shows "Workshop" as the selected option. The "Register" button is no longer visible, indicating the form has been submitted.

The image contains two screenshots of a web browser window titled "myapp" at "localhost:5173".

**Screenshot 1: Event Registration**

This screenshot shows a form for event registration. It includes three input fields: "Name" (Bala), "Email" (bala@gmail.com), and "Event Type" (Seminar). Below the form is a green "Register" button.

**Screenshot 2: Event Registration Confirmation**

This screenshot shows a confirmation message: "Registration Confirmed! ✓". It displays the registered information: Name: Bala, Email: bala@gmail.com, and Event Type: Seminar. At the bottom is a green "Register Another Participant" button.

### 13. Task 13 Implement a simple user profile editing system with form validation.

This screenshot shows a user profile editing form titled "User Profile" with a person icon. The form contains three input fields: "Name" (John Doe), "Email" (john.doe@example.com), and "Phone Number" (1234567890). At the bottom is an orange "Save Profile" button.

The image consists of three vertically stacked screenshots of a web browser window titled "myapp" at "localhost:5173".

**Screenshot 1:** The first screenshot shows the "User Profile" page. It features a header with the title "User Profile" and a small user icon. Below the title is a form with three input fields: "Prabhu" (Name), "Prabhu@example.com" (Email), and "1234567890" (Phone). A "Save Profile" button is located at the bottom right of the form.

**Screenshot 2:** The second screenshot shows a confirmation dialog box from "localhost:5173" stating "Profile updated successfully!" with an "OK" button. This indicates a successful submission of the form.

**Screenshot 3:** The third screenshot shows the "User Profile" page again. It now displays a green success message "Profile Saved! ✅" above the original form. Below this message, the user's details are listed: Name: Prabhu, Email: Prabhu@example.com, and Phone: 1234567890. At the bottom left is an "Edit Profile" button.

14. Task 14 Design a dynamic registration form for an online workshop with live preview.

Online Workshop Registration

Enter your name  
Enter your email  
ReactJS  
Beginner

Register

Live Preview

Name: [Your Name]  
Email: [Your Email]  
Topic: ReactJS  
Level: Beginner

Online Workshop Registration

Vishnu  
Vishnu@gmail.com  
ReactJS  
ReactJS  
NodeJS  
Python  
Data Science

Live Preview

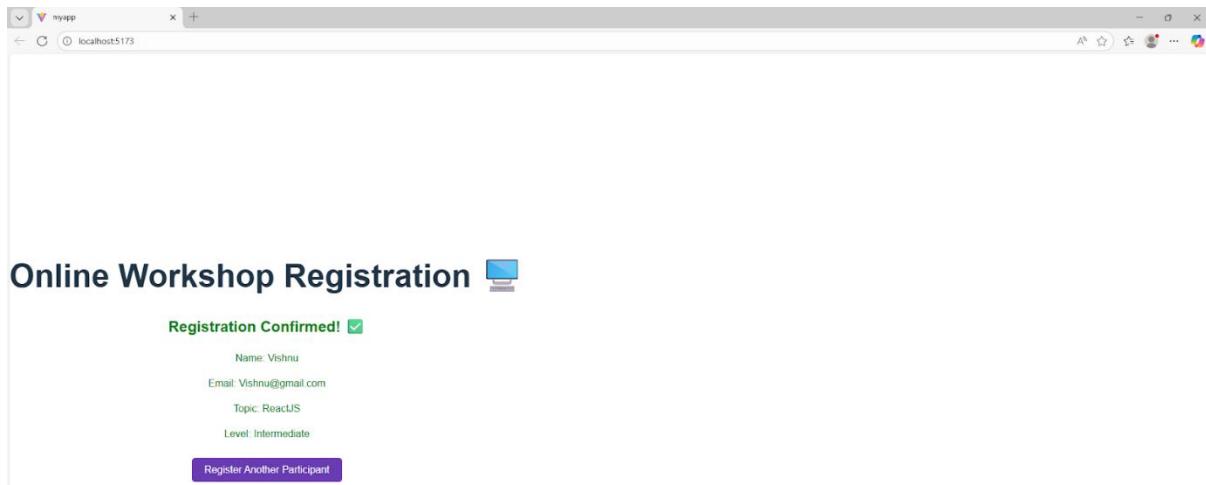
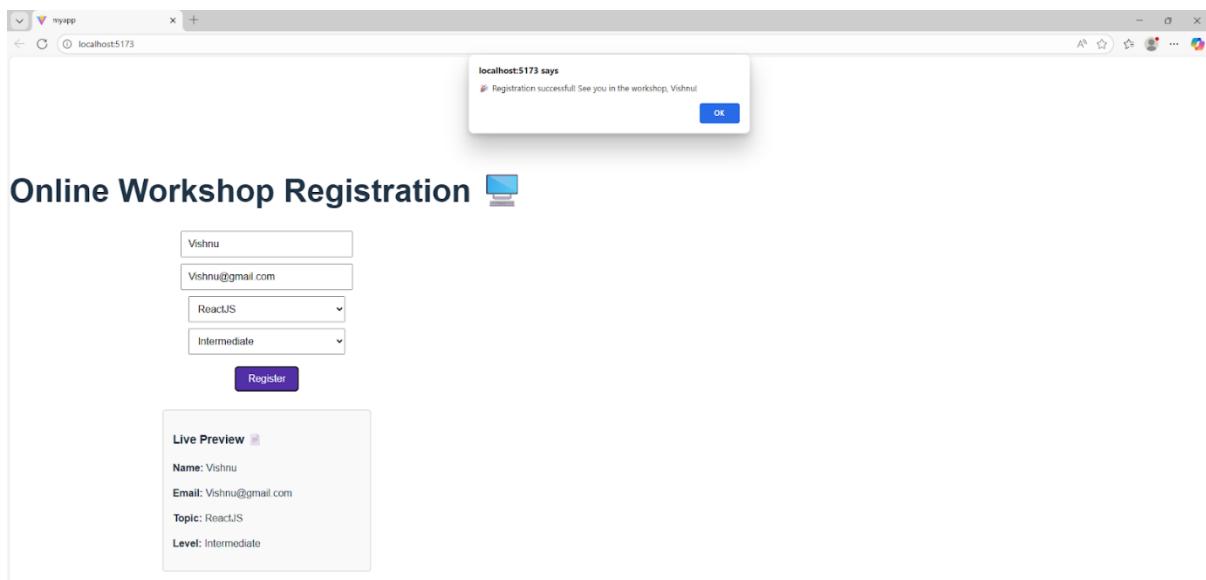
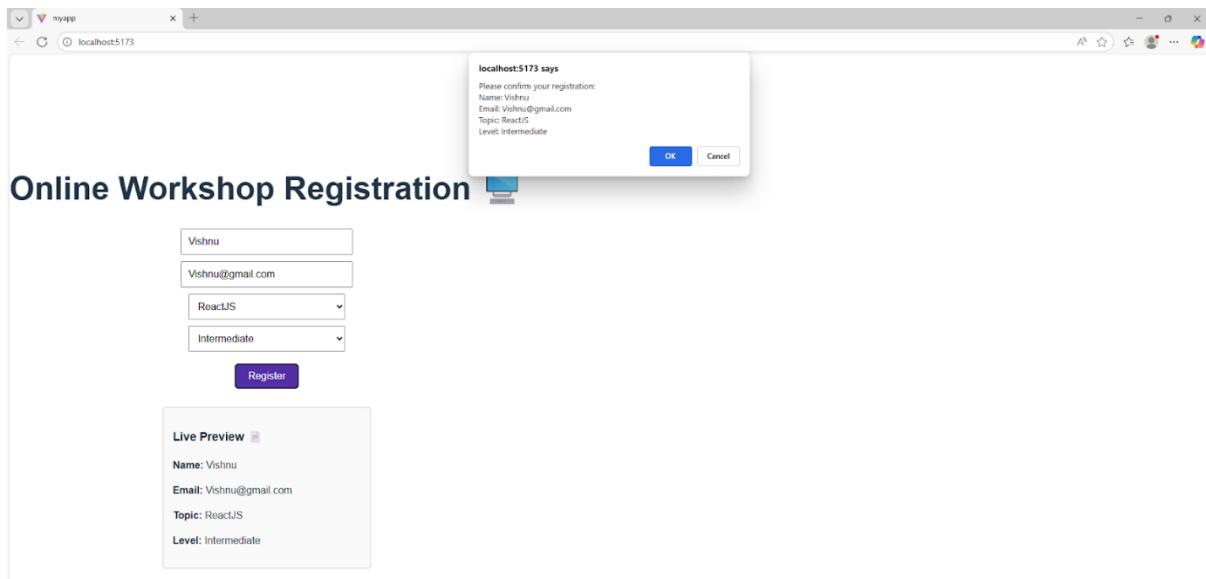
Name: Vishnu  
Email: Vishnu@gmail.com  
Topic: ReactJS  
Level: Beginner

Online Workshop Registration

Vishnu  
Vishnu@gmail.com  
ReactJS  
Beginner  
Beginner  
Intermediate  
Advanced

Live Preview

Name: Vishnu  
Email: Vishnu@gmail.com  
Topic: ReactJS  
Level: Beginner



## 15. 15 Design a login interface that redirects users based on role selection.

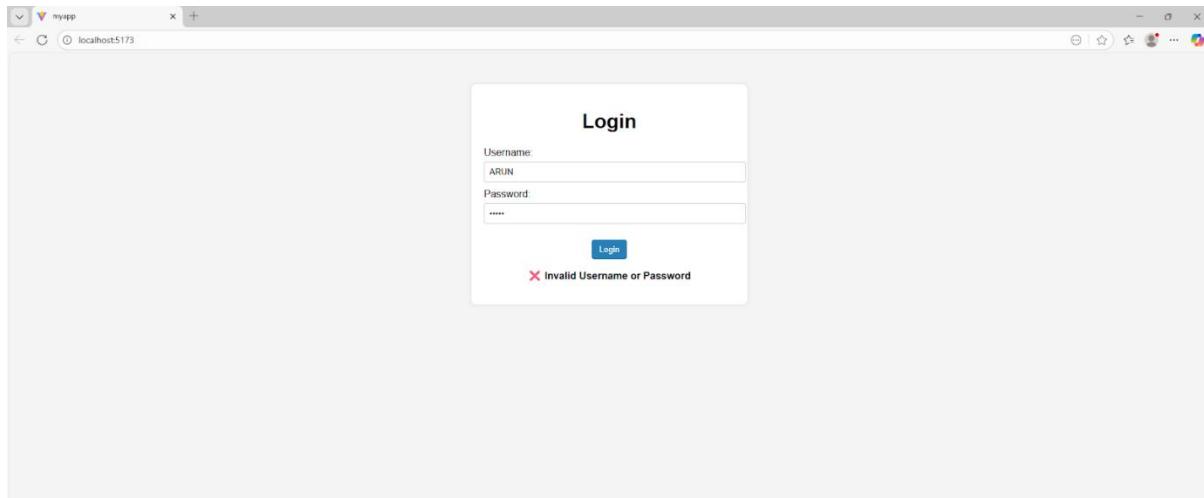
The image consists of three vertically stacked screenshots of a web browser window, likely Google Chrome, displaying a login interface and its subsequent redirection.

**Screenshot 1:** The first screenshot shows the initial "Smart Login" form. It features a title bar with a padlock icon and the text "Smart Login". Below it are two input fields: "Your Name" containing "Scarlet" and "Choose Role" which is currently empty. A blue "Enter" button is at the bottom.

**Screenshot 2:** The second screenshot shows the same form after the user has selected a role. The "Choose Role" dropdown now displays "Admin" and "User", with "Admin" highlighted by a dark grey background. The "Enter" button remains visible.

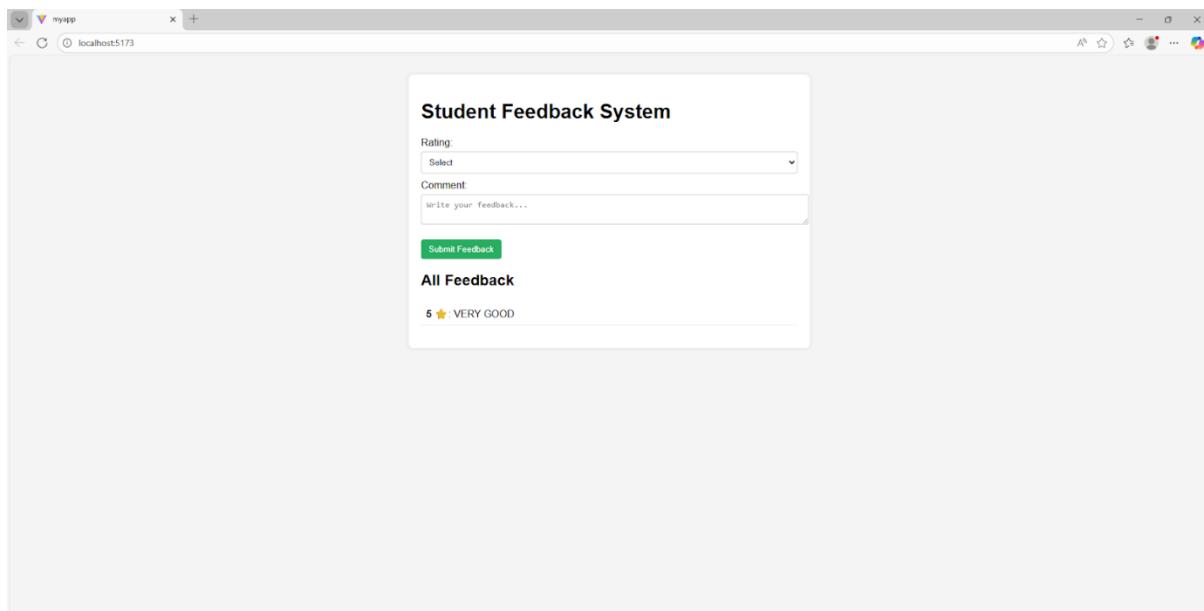
**Screenshot 3:** The third screenshot shows the result of the login. The browser has redirected to a new page with a white header bar. The header bar includes the "app" logo, a search bar with "localhost:5173", and standard browser controls. The main content area displays the message "Welcome Admin Scarlet" in red text, accompanied by a small crown emoji.

## 16. Simulate a basic authentication system with error alerts.



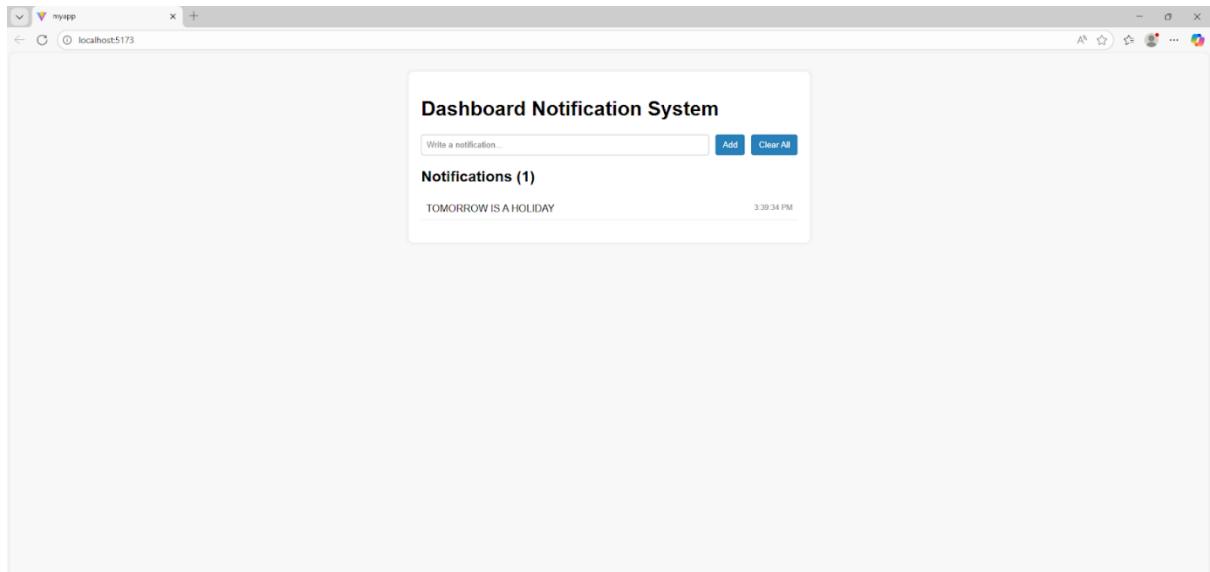
A screenshot of a web browser window titled "myapp" showing a login form. The URL bar indicates the page is at "localhost:5173". The login form has fields for "Username" (containing "ARUN") and "Password" (containing "\*\*\*\*\*"). Below the form is a red error message: "✗ Invalid Username or Password".

## 17. Implement a student feedback system with a rating and comments section.

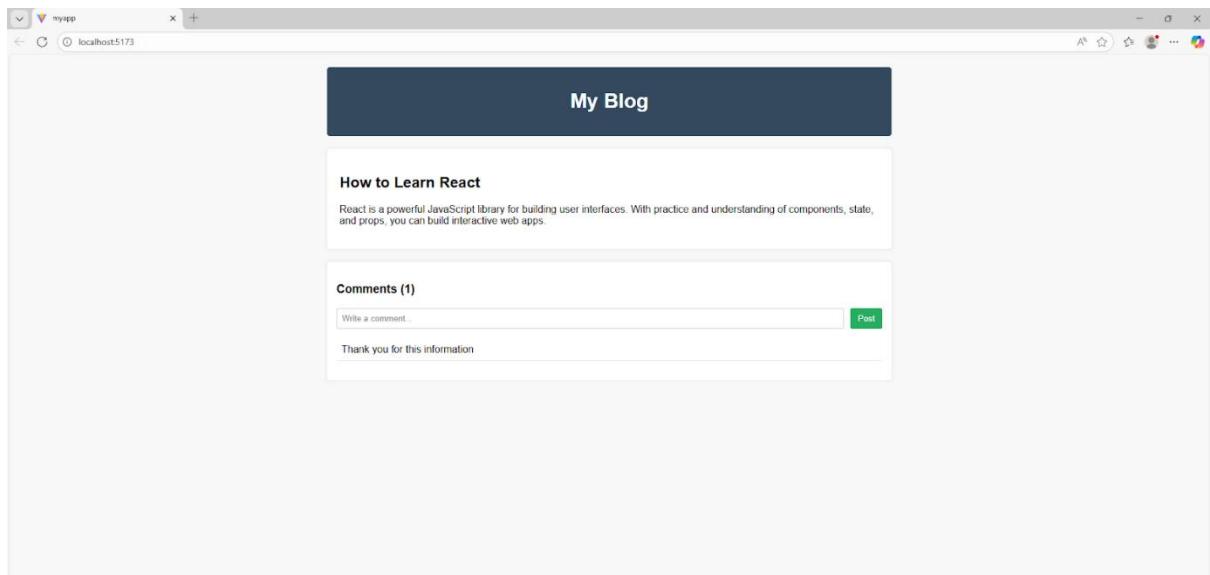


A screenshot of a web browser window titled "myapp" showing a "Student Feedback System". The URL bar indicates the page is at "localhost:5173". The interface includes a "Rating" dropdown menu set to "Select", a "Comment" text area containing "write your feedback...", and a "Submit Feedback" button. Below this is a section titled "All Feedback" showing a single entry: "5 ★ VERY GOOD".

18. Simulate a notification system for a dashboard application.



19. Construct a responsive blog layout with real-time comment update.



20. Simulate a basic e-commerce product listing interface with add-to-cart feature.

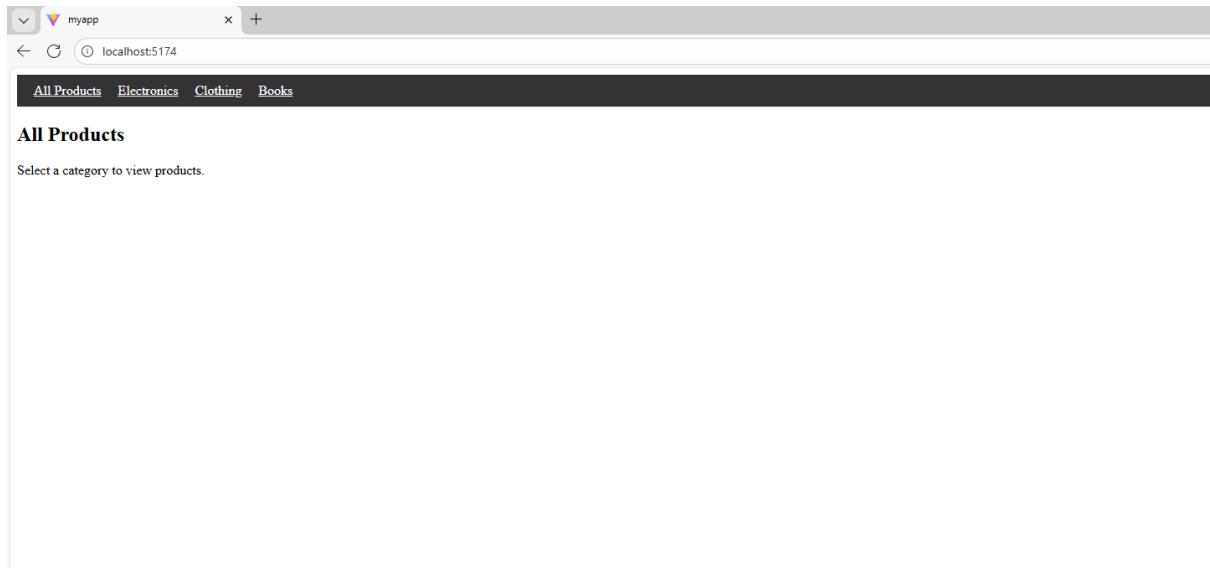
The screenshot shows a web browser window titled "myapp" with the URL "localhost:5173". The page is titled "My E-Commerce Store". It displays a "Products" section with four items: "Laptop" (Price: ₹50000), "Smartphone" (Price: ₹20000), "Headphones" (Price: ₹2000), and "Smartwatch" (Price: ₹5000). Each item has an "Add to Cart" button. Below the products is a "Cart Details" section showing "Smartwatch - ₹5000" and "Smartphone - ₹20000" with a total of ₹25000. The top right corner shows "Cart Items: 2".

21. Design a simple student management dashboard with navigation and data display.

The screenshot shows a web browser window titled "myapp" with the URL "localhost:5173". The page is titled "Student Manager" and features a "Student Management Dashboard". The dashboard includes a navigation bar with links for "Dashboard", "Students", "Courses", and "Logout". Below the navigation is a table with student data:

| ID | Name       | Course  | Grade |
|----|------------|---------|-------|
| 1  | John Doe   | Math    | A     |
| 2  | Jane Smith | Science | B     |
| 3  | Alex Brown | English | A+    |

## 22. Simulate a simple product browsing page with category-based routing.



## 23. Design a multi-page student dashboard with navigation using React Router.

