**Week-7**

**9. ReactJS-HOL**

Getting Started

1. Clone the Repository

git clone https://github.com/yourusername/cricketapp.git

cd cricketapp

2. Install Dependencies

npm install

3. Start the Development Server

npm start

Components

ListofPlayers

Declares an array of 11 players with names and scores.

Uses ES6 map() to display all players.

Filters players with scores below 70 using ES6 arrow functions.

IndianPlayers

Displays Odd and Even indexed players using destructuring logic.

Merges two player arrays (T20players, RanjiTrophyPlayers) using ES6 spread operator.

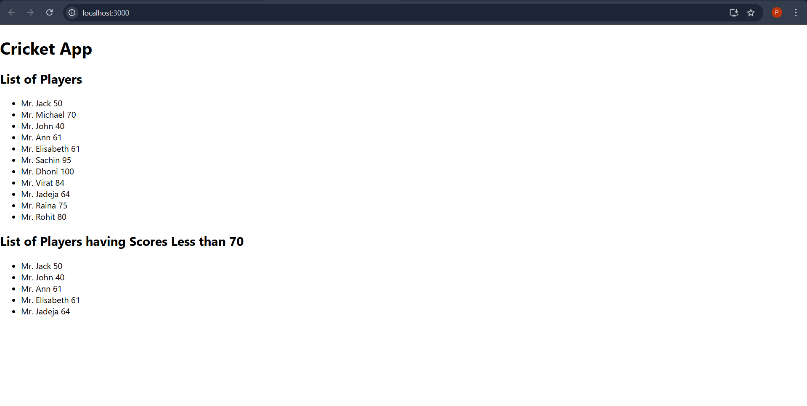
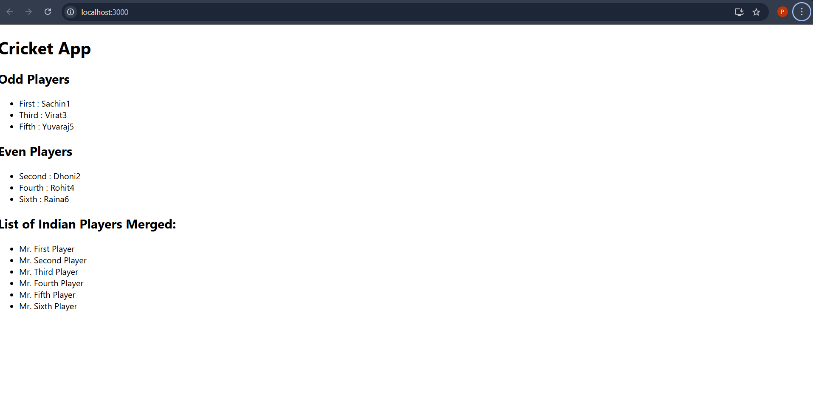
Toggle View

In App.js, toggle between the two components using the flag variable:

const flag = true; // Displays ListofPlayers

// const flag = false; // Displays IndianPlayers

OUTPUT:



**10. ReactJS-HOL**

Getting Started

1. Create the App

npx create-react-app officespacerentalapp

cd officespacerentalapp

2. Add Required Files

Place your office image in the public/ folder: public/office.jpeg

You can use any image and name it office.jpeg or update the filename in App.js.

3. Run the App

npm start

App Structure

src/App.js

Displays a heading (element)

Shows an image (jsxatt)

Renders an object (ItemName) containing office name, rent, and address

Applies conditional class (textRed or textGreen) based on Rent

src/App.css

.textRed {

color: red;

font-weight: bold;

}

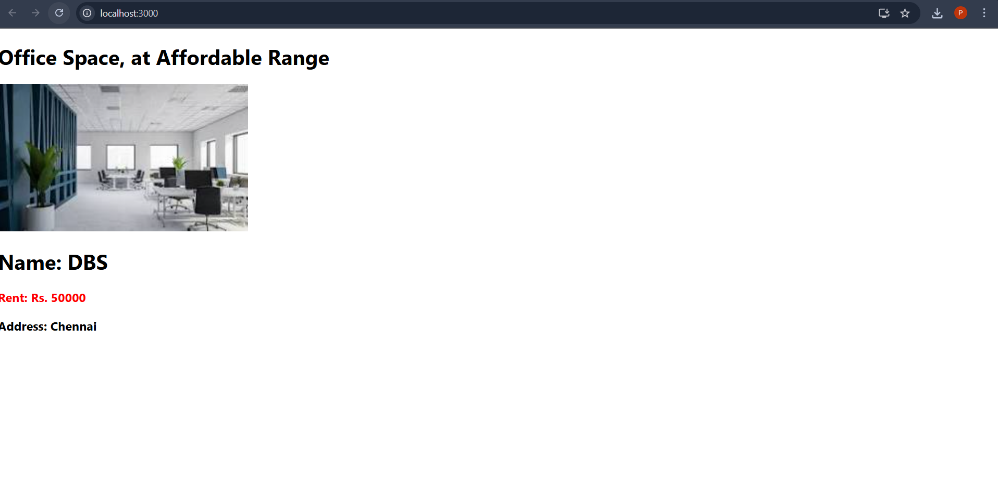
.textGreen {

color: green;

font-weight: bold;

}

OUTPUT:



**11. ReactJS-HOL**

Logic & Implementation

App.js: Handles counter and button actions (all using React synthetic events). Renders the converter.

CurrencyConvertor.js: Controlled form. On submit, if currency is 'Euro', calculates and shows alert; otherwise, shows error.

Setup & Usage

Clone this repo

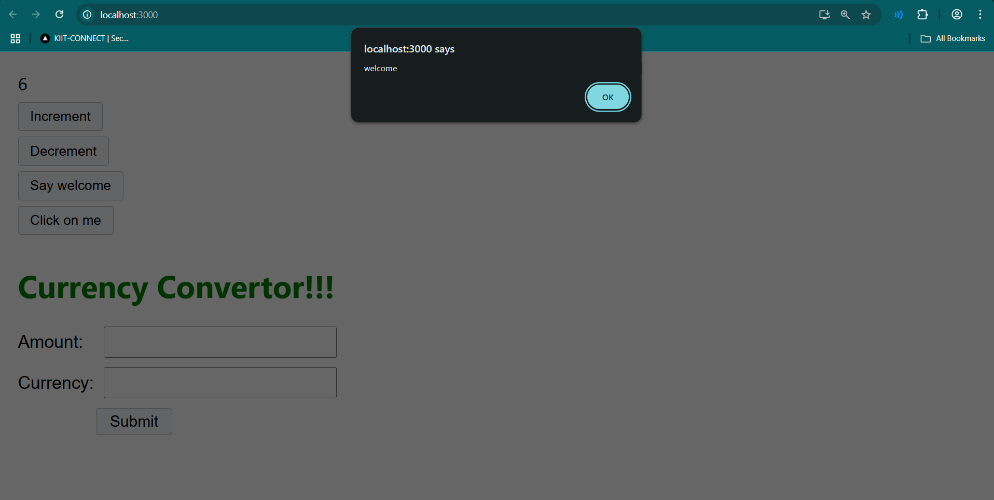
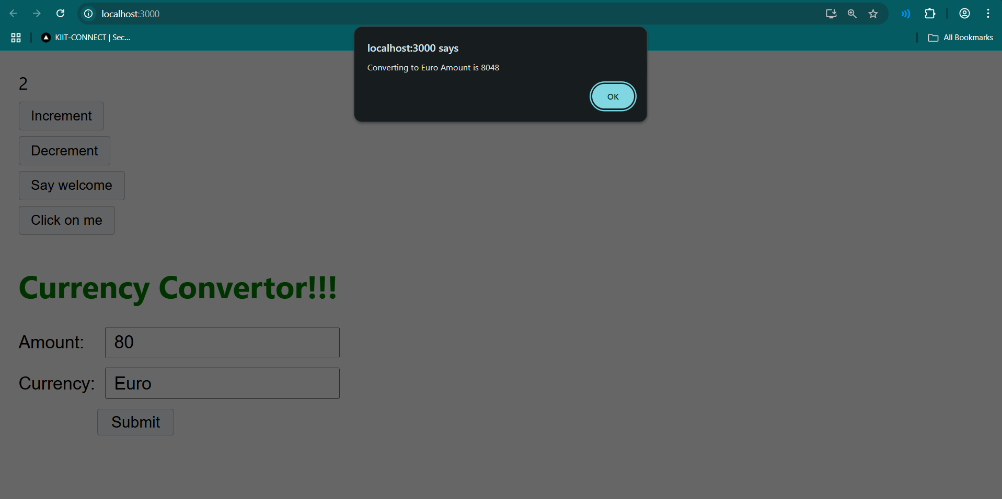
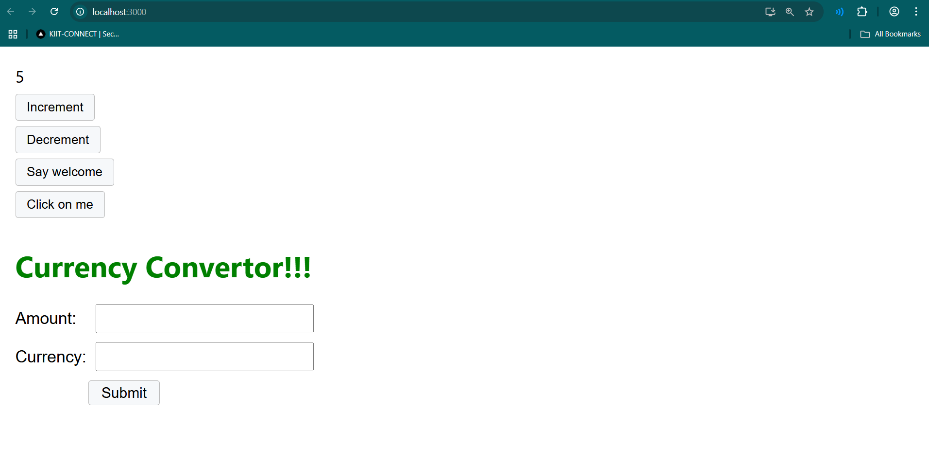
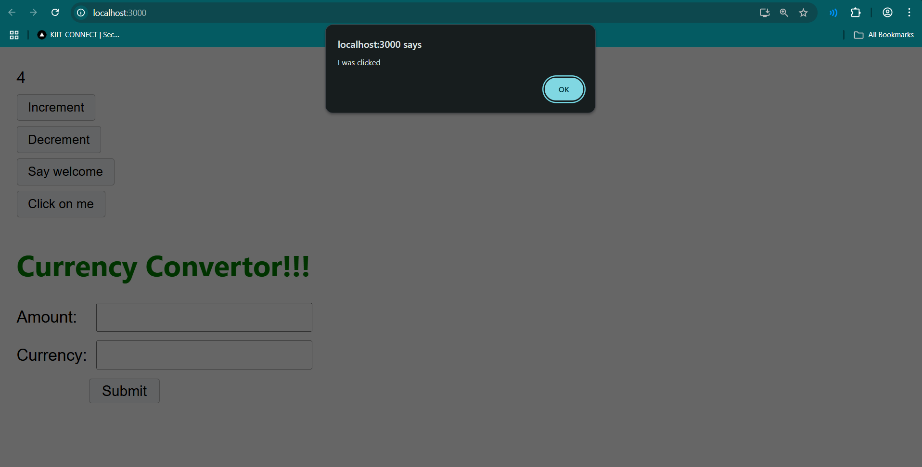
git clone https://github.com/your-username/currency-convertor-app.git && cd currency-convertor-app

Install dependencies

npm install

Run the app

npm start



**12. ReactJS-HOL**

Code Logic Explained

App.js

Maintains isLoggedIn boolean state that tracks authentication status.

Implements event handlers handleLoginClick and handleLogoutClick to update isLoggedIn.

Conditionally renders the greeting message via <Greeting isLoggedIn={isLoggedIn} />.

Conditionally renders either <LoginButton /> or <LogoutButton />.

Imports its components from the /components folder.

Greeting and its subcomponents

Greeting chooses between:

<UserGreeting /> — displays "Welcome back!"

<GuestGreeting /> — displays "Please sign up."

LoginButton & LogoutButton

Simple buttons that accept an onClick prop callback.

Render "Login" or "Logout" text respectively.

Setup & Usage

Prerequisites

Node.js

npm

Installation & Run

Clone or download the repository to your local machine.

Start the development server:

npm start

Using the App

Initially, you see "Please sign up." with a Login button.

Click Login to switch to "Welcome back!" and display the Logout button.

Click Logout to revert to the guest view.

The text and buttons update dynamically based on the login state.



**13. ReactJS-HOL**

Components

The application has three main components:

BookDetails

BlogDetails

CourseDetails

Each component receives or imports static data from a centralized data folder and displays it using .map().

Conditional Rendering Techniques Used

Technique Component Example

map() with key All Displaying lists with unique IDs

JSX variable rendering BookDetails const bookdet = (...)

JSX fragments BlogDetails {blogs.map(blog => (...))}

Setup and Usage

Prerequisites

To run this project locally, make sure you have:

Node.js

npm

Visual Studio Code (or any editor)

Installation & Run

Clone or download the repository to your local machine.

Start the development server:

npm start

