# ReactJS Mandatory Hands-on

Name: Yukta Singh

Course: ReactJS Fundamentals

Institution: Cognizant

Assignment Set: Lab 1 to Lab 5

## Lab 1: My First React App

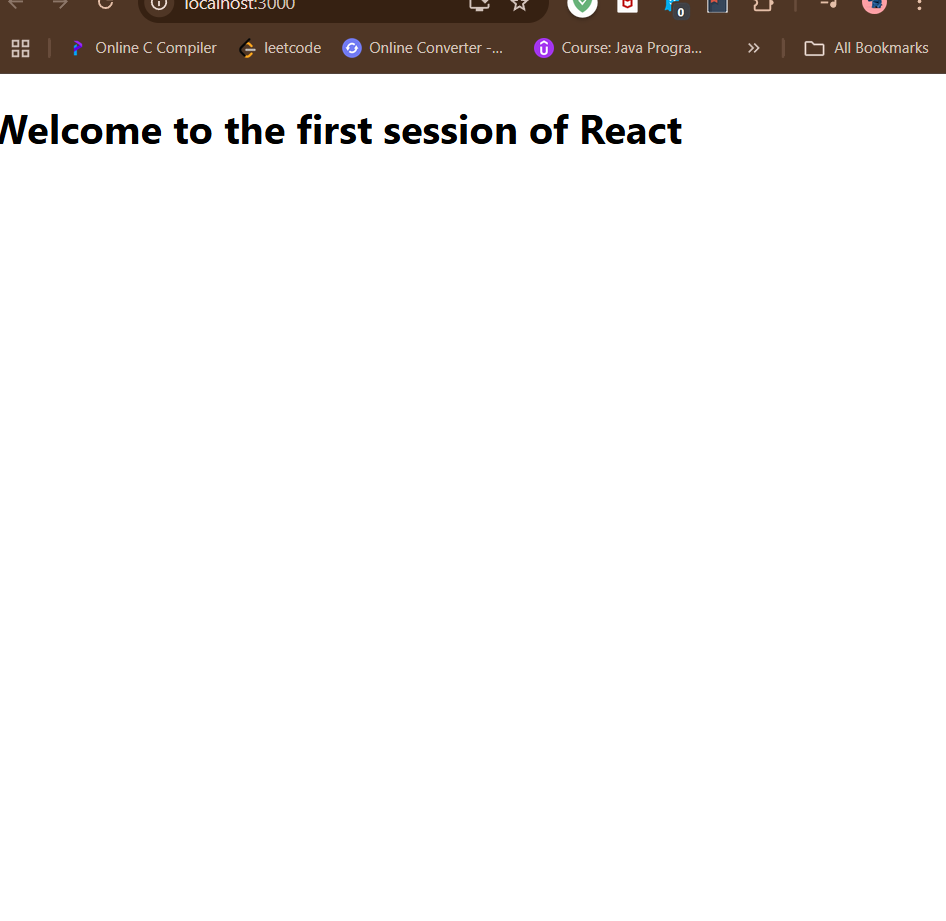
Objective: Set up a React app and print a heading.

Steps Followed:

* Installed Node.js and npm.
* Created the app using: npx create-react-app myfirstreact
* Opened project in VS Code: cd myfirstreact && code .
* Edited App.js to display heading using a React functional component.
* Ran the app using npm start.

Output:

Page displays heading: 'Welcome to the first session of React'



## Lab 2: Student Management Portal with Multiple Components

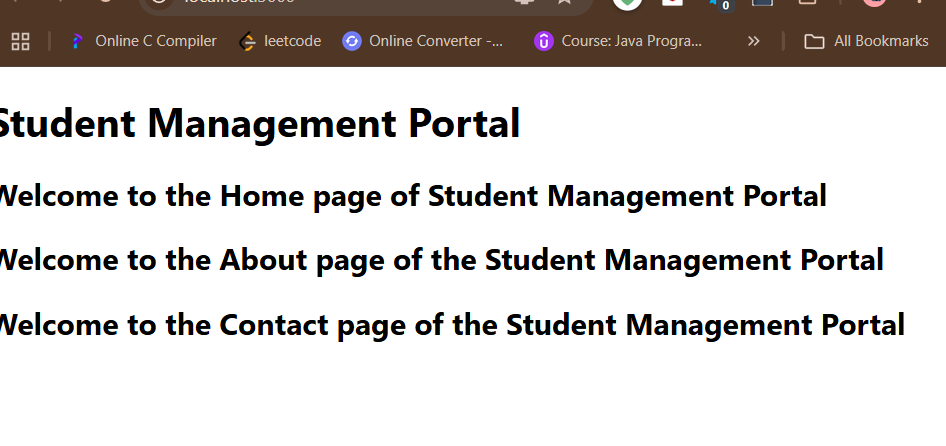
Objective: Create multiple components and display them in App.js.

Steps Followed:

* Created a new app using create-react-app.
* Created Components: Home.js, About.js, Contact.js inside /Components folder.
* Displayed all components inside App.js using JSX.
* Used functional components for simplicity.

Output:

Page displays heading and all three components one after another.



## Lab 3: Score Calculator with Styling

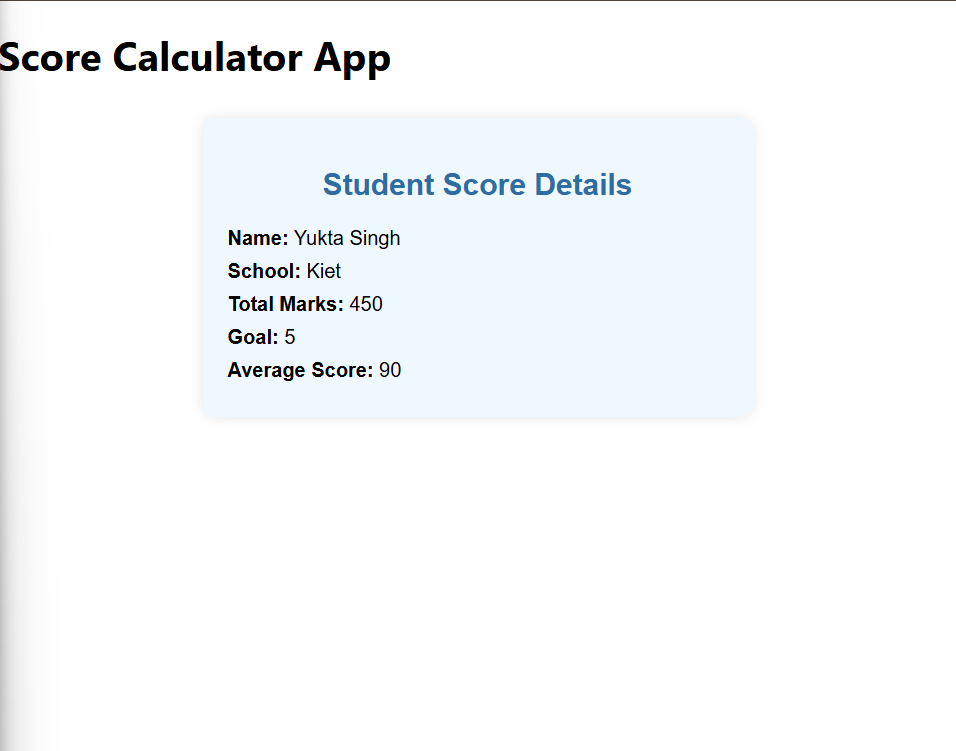
Objective: Create a functional component that calculates and displays a student's average score with CSS styling.

Steps Followed:

* Created a new app and opened it in VS Code.
* Created folders: /Components and /Stylesheets.
* Created CalculateScore.js and added logic to calculate average.
* Styled the component using mystyle.css via CSS classes.
* Imported and displayed the CalculateScore component in App.js.

Output:

Styled card appears showing average score with background, padding and margin.



## Lab 4: Blog App with Lifecycle Methods

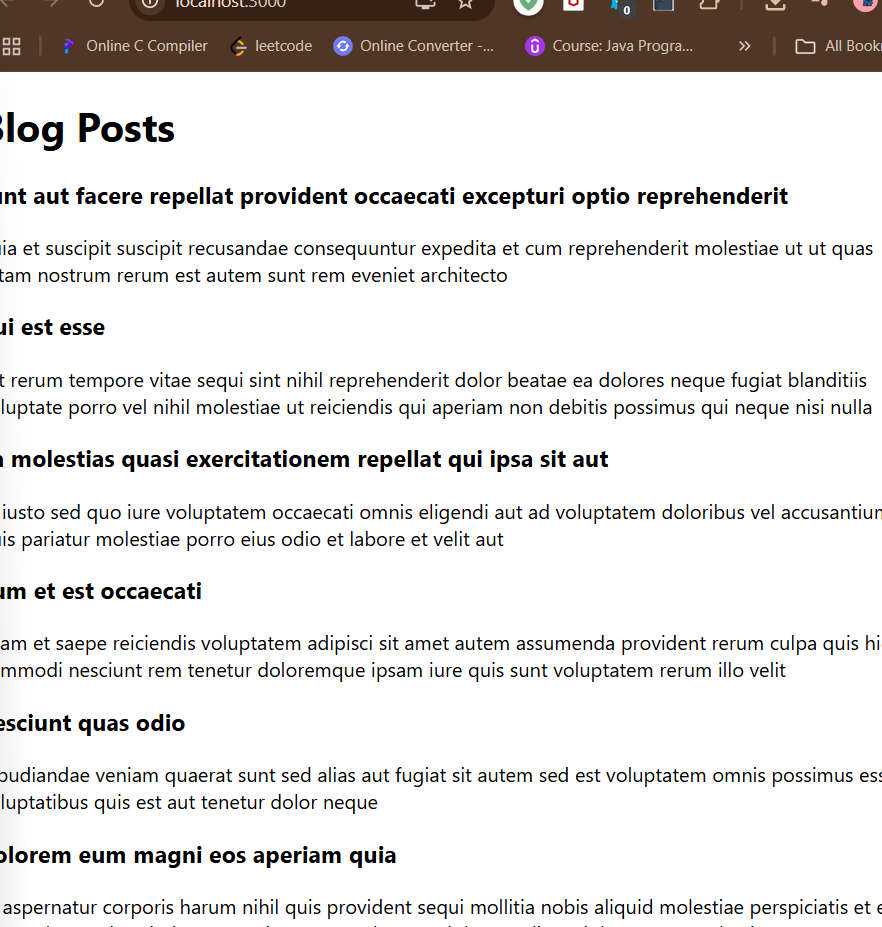
Objective: Use class components with componentDidMount and componentDidCatch to fetch data and handle errors.

Steps Followed:

* Created new React app: blogapp.
* Created Post.js to render each post.
* Created Posts.js to fetch posts from API and render a list using Post.
* Used componentDidMount to load data and componentDidCatch to handle errors.
* Displayed Posts inside App.js.

Output:

Blog app loads data from JSONPlaceholder and displays post titles and content using class components.



## Lab 5: Styling Components with CSS Modules and Inline Styling

Objective: Apply scoped CSS styles using CSS Modules and inline conditional styles based on props.

Steps Followed:

* Opened the given project or created one using create-react-app.
* Created CohortDetails.module.css with .box class and dt tag selector.
* Created CohortDetails.js with inline conditional color styling in <h3>.
* Displayed CohortDetails in App.js and passed mock cohort data.
* Ran the app and verified styling and conditional logic.

Output:

Page displays styled cohort cards. Ongoing cohorts show green heading, others blue.

