INTERNSHIP REPORT

WEEK 6 DAY 1

Submitted to:	**Ali Hyder**	Submission Date:	**28thJuly, 2025**
Internship Domain:	**Front Development**	Internship Name:	**ProSensia**
Student Name:	**Yasal Qamar**	Roll No.	**S25031**

React.js Basics

Topics: Introduction to React, create-react-app setup

Objective

To understand how React apps are structured, how JSX is rendered, and how styling is applied using external CSS in a component-based architecture.

Topics Covered:

- Introduction to **React.js**
- Setting up a project using `create-react-app`
- Editing the root component `App.js`
- Rendering JSX directly in `index.js`
- Creating and applying external CSS (`index.css`)

What is React?

React is a **JavaScript library** used to build **user interfaces (UI)**, especially for **web applications**. It helps you create **dynamic, reusable components** that can change based on data without needing to reload the page.
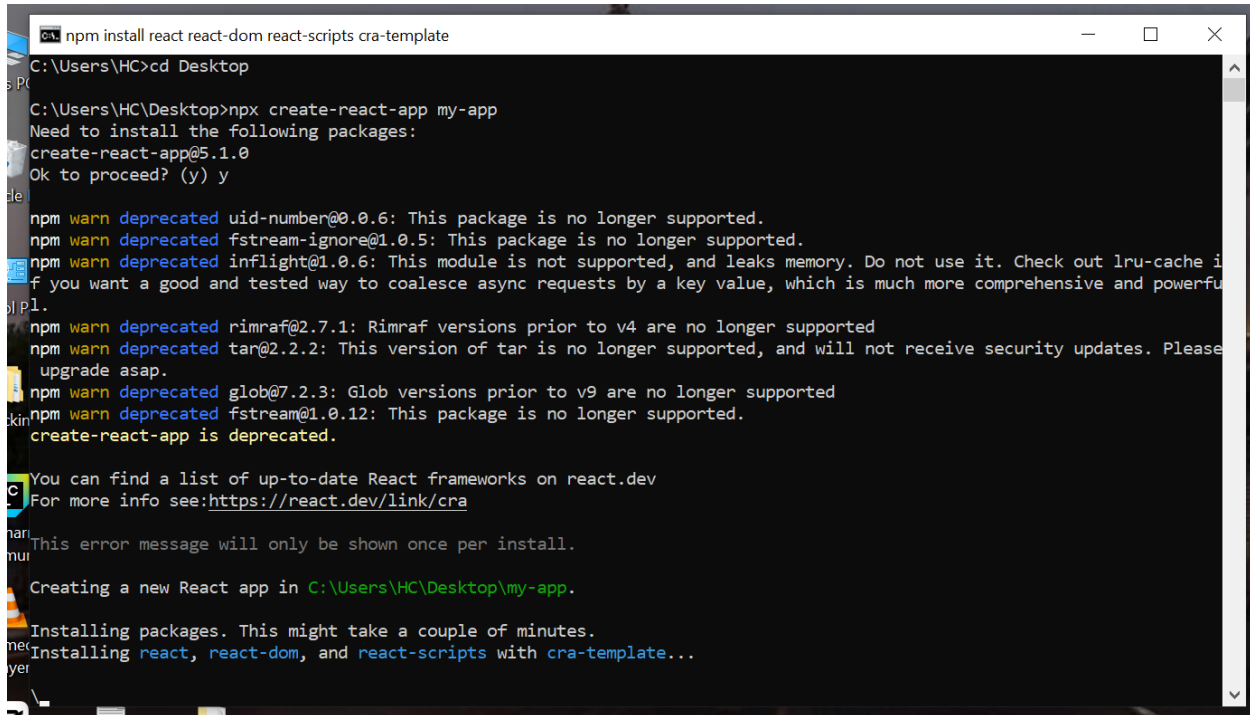
Why Use React?

- **Component-Based**: Break your UI into small, reusable pieces (like buttons, forms, headers).
- **Fast and Efficient**: Uses a virtual DOM to update only the changed parts of the UI.
- **One-Way Data Flow**: Data flows in one direction, making the app easier to debug and manage.
- **Reusable JSX Code**: Write HTML-like code inside JavaScript using JSX.

1. React App Setup:

I created a new React app using the terminal with these commands:

```
npx create-react-app my-app
cd my-app
npm start
```

Command Prompt

Run `npm audit` for details.

Created git commit.

Success! Created my-app at C:\Users\HC\Desktop\my-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd my-app
  npm start

Happy hacking!

C:\Users\HC\Desktop>


```
C:\Users\HC\Desktop>cd my-app

C:\Users\HC\Desktop\my-app>npm start

> my-app@0.1.0 start
> react-scripts start

(node:264) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is dep
recated. Please use the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:264) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is d
eprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```
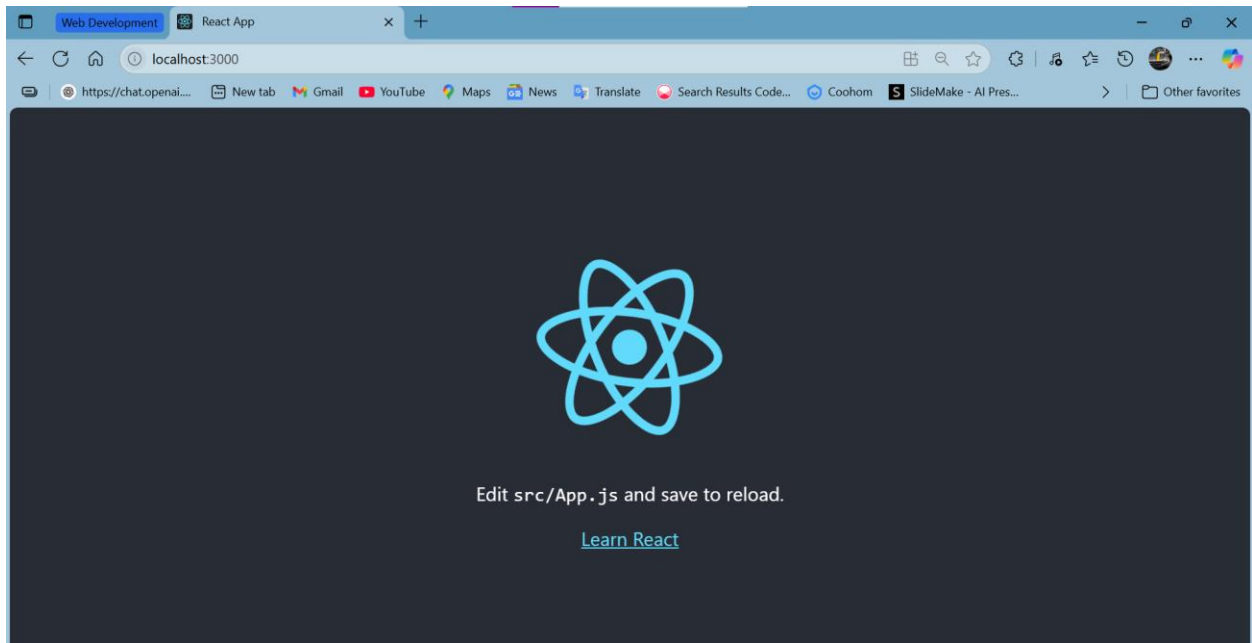
This opened the app at http://localhost:3000 showing the default React welcome screen.
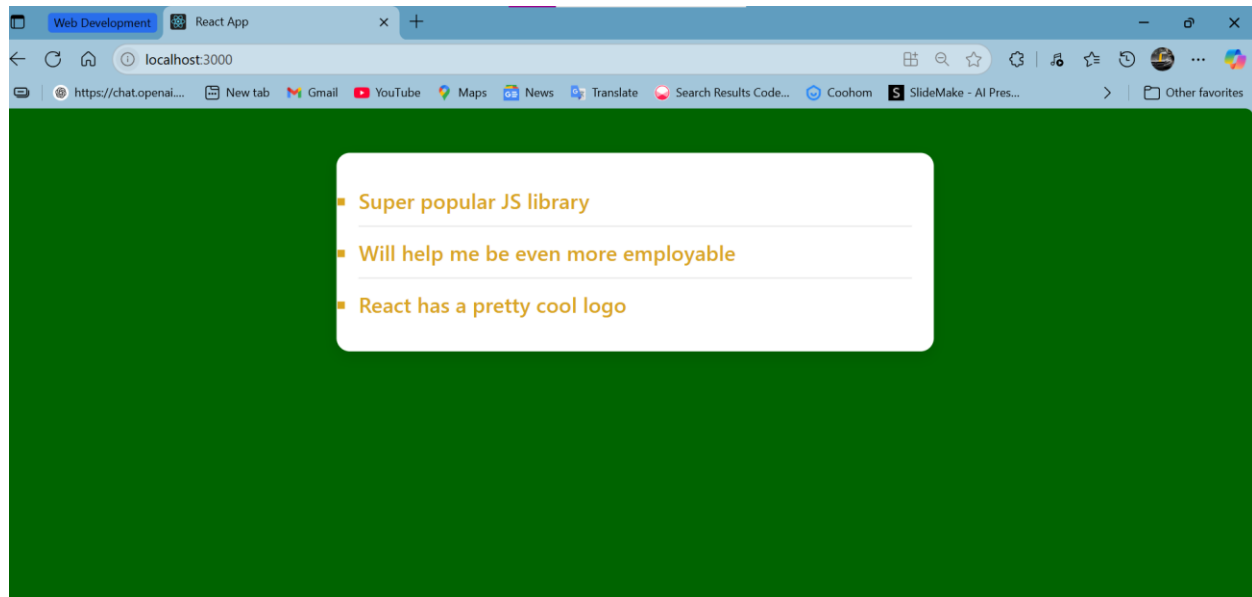
Example JSX:

2. Editing the Root Component (App.js):

I modified App.js to render a custom unordered list using JSX:

```
function App() {
  return (
   <ul>
    <li>Super popular JS library</li>
    <li>Will help me be even more employable</li>
    <li>React has a pretty cool logo</li>
   </ul>
  );
}

export default App;
```
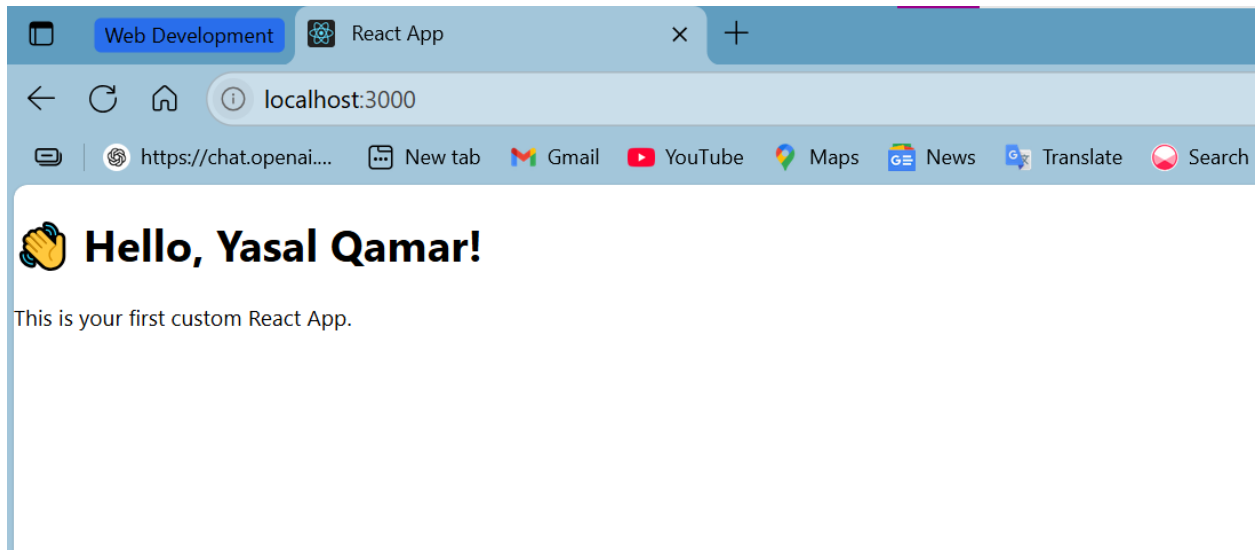
3. Styling with External CSS:

I created a custom styles.css file and added styling to improve the UI. Then I imported it into App.js using:

```
import './styles.css';
```

This gave the list a modern, clean design with padding, background color, borders, and centered layout.

EXAMPLE

```
function App() {
  return (
    <div>
      <h1>👋 Hello, Yasal Qamar!</h1>
      <p>This is your first custom React App.</p>
    </div>
  );
}

export default App;
```

CONCUSLION:

Today I successfully set up and customized a React app, edited components using JSX, and applied external CSS for styling. I now understand the structure of a basic React project and how to render content to the DOM using both App.js and index.js.