lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

STUDENT LABORATORY MANUAL

OF

IV B.TECH I SEMESTER (R20)

REACT JS FRAMEWORK

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

INSTITUTE

VISION

Producing globally competent and quality technocrats with human values for the holistic needs of industry and society

MISSION

- Creating an outstanding infrastructure and platform for enhancement of skills, knowledge and behaviour of students towards employment and higher studies
- Providing a healthy environment for research, development and entrepreneurship, to meet the expectations of industry and society.
- Transforming the graduates to contribute to the socio-economic development and welfare of the society through value based education.

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

VISION

To excel in the computing arena and to produce globally competent computer science and Information Technology graduates with Ethical and Human values to serve society.

MISSION

- To impart strong theoretical and practical background in computer science and information technology discipline with an emphasis on software development.
- To provide an open environment to the students and faculty that promotes professional growth
- To inculcate the skills necessary to continue their education and research for contribution to society.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates of Computer Science and Information Technology will acquire strong knowledge to analyze, design, and develop computing products and solutions for real-life problems utilizing the latest tools, techniques, and technologies.

PEO2: Graduates of Computer Science and Information Technology shall have interdisciplinary approach, professional attitude and ethics, communication, teamwork skills and leadership capabilities to solve social issues through their Employment, Higher Studies and Research.

PEO3: Graduates will engage in life-long learning and professional development to adapt to dynamically computing environment.

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design & Development: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

PO4: Investigations: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tools: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: Engineer & Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment & Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice

PO9: Individual & Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings

PO10: Communication Skills: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions

PO11: Project mgt. & Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments

PO12: Life Long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Ability to solve contemporary issues utilizing skills

PSO2: To acquire knowledge of the latest tools and technologies to provide technical solutions

PSO3: To qualify in national and international competitive examinations for successful higher studies and
Employment

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

Course Outcomes (CO's)

C417.1	Understand the anatomy of React Java Script.
C417.2	Understand the life cycle methods of React JS.
C417.3	Implement React components for building applications.
C417.4	Implement React hooks for component reusability and monitoring.
C417.5	Implement React rendering for interactive applications.

.

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

SYLLABUS

Subject Code	Subject Name	L	T	P	C
R20CIT-SC4101	React JS Framework (Skill Course)	0	1	2	2

Course Objectives:

- To learn essential, React JS skills for front-end development.
- To explore client-side JavaScript application development and the React library.
- To implement React components, hooks, and state management for building interactive UIs.
- To gain experience with React.js, JSX, HTML, CSS, and JavaScript.
- To create a functional front-end web application using React.

Course Outcomes:

1. Understand the anatomy of React Java Script.
2. Understand the life cycle methods of React JS.
3. Implement React components for building applications.
4. Implement React hooks for component reusability and monitoring.
5. Implement React rendering for interactive applications.

Unit 1: React JS – Introduction to React JS, React vs Angular, React Version History, Anatomy of React Project, Creating and Running React App.
Templating using JSX: Expressions, Operators, Attributes, Fragments.

Learning Outcomes: Student will be able to

- Understand react framework for building applications.(L2)
- Understand the installations of react packages.(L2)
- Create templates in react applications. (L4)

Unit 2:React Core: Props, State, Event Handling, Lists and Keys, Styling, React Life Cycle, Life cycle methods, Mounting Life Cycle,

Learning Outcomes: Student will be able to

- Understand event handling in React. (L2).
- Implement life cycle methods in react.(L4).
- Create props and states in building react apps.(L4)

Unit 3:React Components: Pure Components, memo, Refs, Portals, Higher Order Components (HOC), Context, HTTP requests (POST & GET).

Learning Outcomes: Student will be able to

- Understand http request methods in handling end points. (L2)
- Create components to handle react requests. (L4)
- Create higher order components and refs in react.(L4)

Unit 4: React Hooks: Introduction to Hooks, useState, useEffect, Run Effects, Fetching Data, useContext, useReducer, useCallBack, useMemo, useRef, Custom Hooks

Learning Outcomes: Student will be able to

- Understand react hooks. (L2)
- Create hooks and custom methods for handling components. (L4)
- Implement context and callback methods in hooks. (L4)

Unit 5: React Render: Introduction to Rendering, useState, useReducer, State Immutability, Parent & Child, Memo, Context, useCallBack.

Learning Outcomes: Student will be able to

- Understand the working react rendering. (L2)
- Implement userReducer and context for rendering react apps. (L4)

APPLICATIONS:

- Online web applications
- Financial, banking applications and gateways etc
- Online and Social media applications

TEXT BOOKS:

1. React.js Book: Learning React JavaScript Library From Scratch by Greg Sidelnikov, Learning Curve.
2. React: Quickstart Step-By-Step Guide To Learning React Javascript Library (React.js, Reactjs, Learning React JS, React Javascript, React Programming) by Lionel Lopez

REFERENCE BOOKS:

- Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js, 2nd Edition by Shama Hoque, Packt

COURSE OUTCOMES VS POs MAPPING (DETAILED: HIGH: 3, MEDIUM: 2, LOW: 1)

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
C417.1	3	2	2	1	3	-	-	-	1	-	1	1	2	3	2
C417.2	3	2	1	1	3	-	-	-	1	-	1	1	1	3	3
C417.3	3	1	2	2	3	-	-	-	1	-	1	1	2	3	2
C417.4	3	1	2	2	3	-	-	-	1	-	1	1	2	3	3
C417.5	2	2	1	1	2	-	-	-	1	-	1	1	1	2	2
C417*	3	2	2	2	3	-	-	-	1	-	1	1	2	3	3

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

COURSE OUTCOMES - On successful completion of this course, students should be able

S.NO.	DESCRIPTION	PO (1…12) MAPPING	PSO (1…3) MAPPING
C417.1	**Understand** the anatomy of React Java Script. (LEVEL 2)	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12	PSO1, PSO2, PSO3
C417.2	**Understand** the life cycle methods of React JS. (LEVEL 2)	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12	PSO1, PSO2, PSO3
C417.3	**Implement** React components for building applications. (LEVEL 3)	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12	PSO1, PSO2, PSO3
C417.4	**Implement** React hooks for component reusability and monitoring. (LEVEL 3)	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12	PSO1, PSO2, PSO3
C417.5	**Implement** React rendering for interactive applications. (LEVEL 3)	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12	PSO1, PSO2, PSO3
COURSE OVERALL PO / PSO MAPPING: PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3.			
L1 – Remember, L2 – Understand, L3 – Apply, L4 – Analyze, L5 – Evaluate, L6 – Create			

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

INDEX

5	Create a Stateless function component for Student and display the details along with percentage.	13-15	CO2	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3
6	Create a Multipage React App using page routing with react-router-dom library.	16-20	CO3	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3
7	Build a Counter app using the useState hook.	21-24	CO4	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3
8	Build a Favorite Color component using the useState hook.	25-26	CO4	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3
9	Implement pagination using the useState, useEffect hooks and Fetching Data.	27-30	CO4	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3
10	Build Timer & Counter components using useEffect Hooks without and with Dependency respectively.	31-33	CO4	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3
11	Render a form component with validation using the useState hook.	34-36	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3
12	Create a To-do List app using the useState, useEffect and useContext hooks.	37-39	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3

13.	Create a React App using the Nested Components and useContext hook.	40-41	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3
14.	Create a React App for tracking the state changes by using useRef hook.	42-43	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3
15.	Create a React App for tracking the multiple pieces of the state changes by using the useReducer and Custom State logic.	44-46	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3

Experiments Beyond the Syllabus				
1	Create a React App to demonstrate how to fetch data by using API call.	47-48	CO3	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3
2	Build a Password Generator App using React hooks.	49-54	CO4	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3
3	Build a Real-time Chat App using React hooks.	55-61	CO5	PO1, PO2, PO3, PO4, PO5, PO9, PO11, PO12, PSO1, PSO2, PSO3

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

Instructions to Students

Pre-lab Activities:

- Prepare observation note book which contains the following:
  - Procedure/algorithm/program to solve the problems discussed in the theory class
  - Solutions to the exercises given in the previous lab session
- Refer the topics covered in theory class

In-lab activities:

- Note down errors observed while executing program and remedy for that.
- Note down corrections made to the code during the lab session
- Answer to vivo-voce
- Get the observation corrected
- Note down inferences on the topic covered by the programs executed

Post-lab activities:

- Solve the given exercises
- Devise possible enhancements that can be made to the solved problem to simplify the logic
- Executed programs should be recorded in the lab record and corrected within one week after completion of the experiment.
- After completion of every module, a test will be conducted, and assessment results will have weight in the final internal marks.

General Instructions:

- Student should sign in the log register before accessing the system.
- Student is only responsible for any damage caused to the equipment in the laboratory during his session.
- Usage of pen drives is not allowed in the lab.
- If a problem is observed in any hardware equipment, please report to the lab staff immediately; do no attempt to fix the problem yourself.
- Systems must be shut down properly before leaving the lab.

- Please be considerate of those around you, especially in terms of noise level. While labs are a natural place for conversations regarding programming, kindly keep the volume turned down.

Experiment – 1

AIM:
Installation of NodeJS, ReactJS Libraries, Create ReactJS App, and Run ReactApp.

DESCRIPTION:
The installation of Node.js and npm, the installation of essential React.js libraries, the creation of a new React app using Create React App, and the process of running the newly created React application. It provides step-by-step instructions to help beginners get started with React.js development, from the initial setup to running a sample React app for the first time.

SOLUTION:

Step_1

Download & Install NodeJS from nodejs.org

Step_2

Check the version

```
node -v
```

Step_3

Install React JS Libraries

```
npm install -g create-react-app
```

Step_4

Create a folder reactjs

```
mkdir reactjs
```

Step_5

Change directory

```
cd reactjs
```

Step_6

Create ReactJS App

```
npx create-react-app myapp
```

Step_7

Change directory

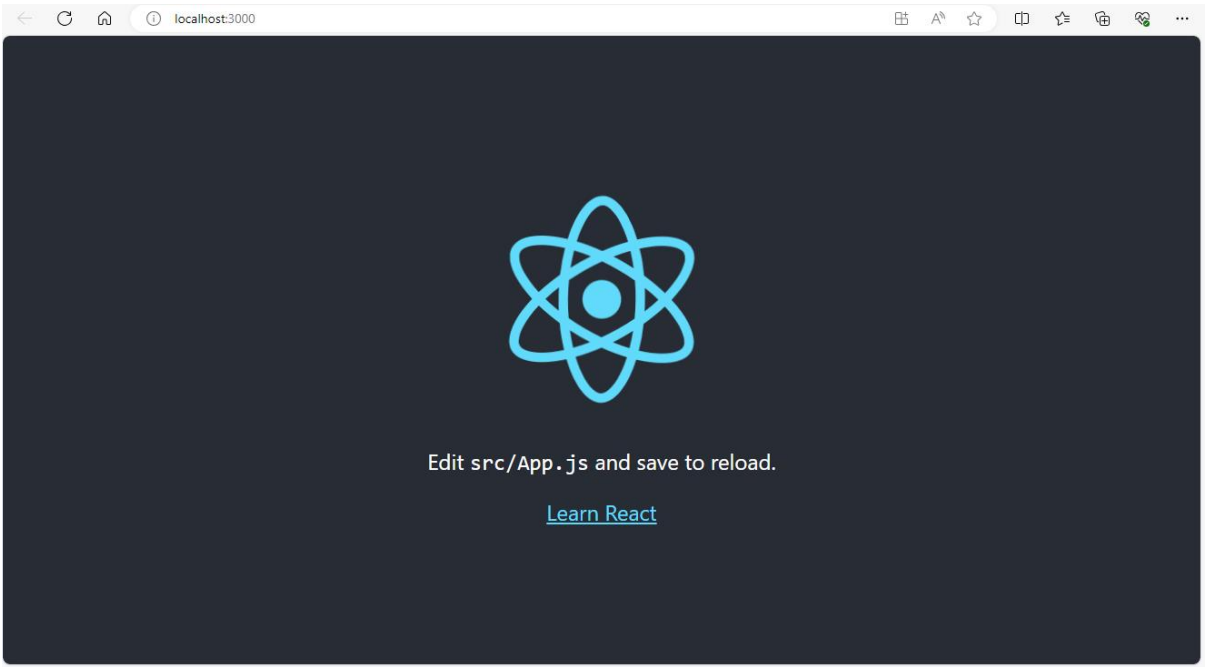
```
cd myapp
```

Step_8

Run App

```
npm start
```

SAMPLE OUTPUT:



Experiment – 2

AIM:
Create a function component for displaying Employee details in a table format using the props.

DESCRIPTION:
A React function component is one of the fundamental building blocks of a React application. It is a JavaScript function that returns JSX (JavaScript XML) to define the structure and content of a user interface component. React function components are used to create reusable, self-contained pieces of a user interface.

Function components can accept input data, known as "props" (short for properties). Props are passed into the component as arguments to the function and allow you to customize the behaviour and appearance of the component.

Install create-react-app by running

npm-create-react-app myapp

After installation,change the directory to the myapp folder by running

Cd myapp

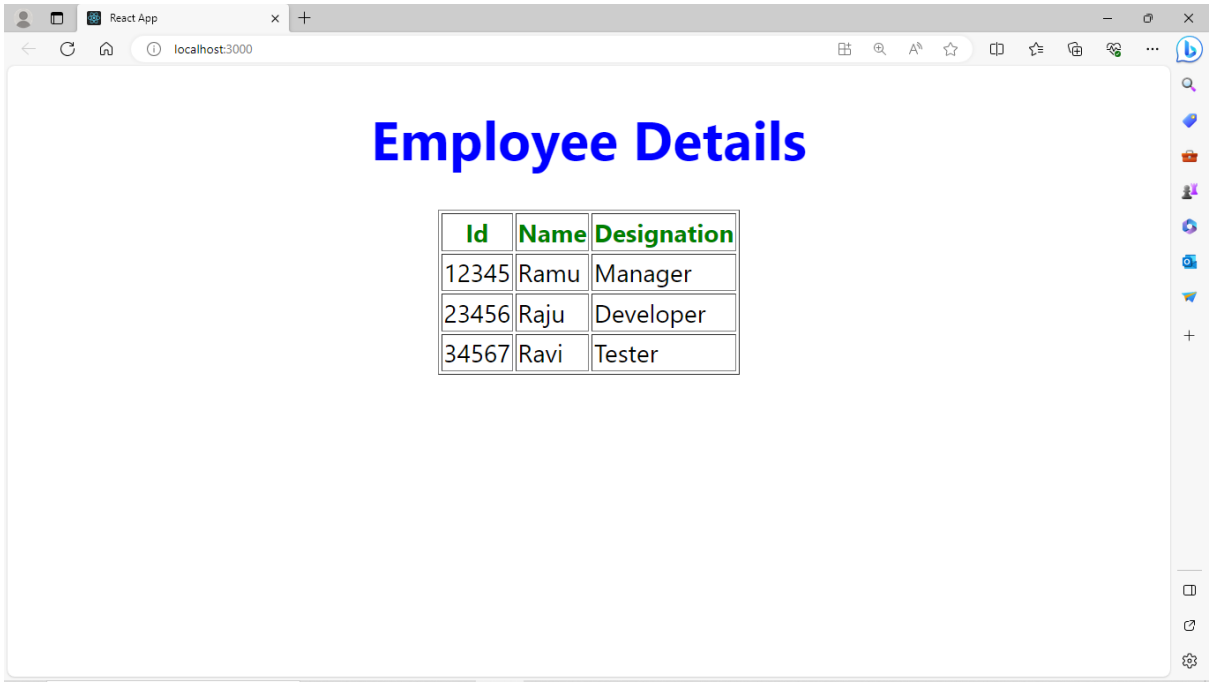
I'musing myapp here but you can call your app anything you want. Run

npm start

To start the development server whichisalwaysavailableatport3000,

i.e.http://localhost:3000.Whentheserverstartsrunningonport3000,youseeaspinner.

SAMPLE OUTPUT:



Experiment – 3

AIM:
Create a Class component for Student and display student details in a table format using theprops.

DESCRIPTION:
A class component must include the extends React.component statement. This statement creates an inheritance to React.Component, and gives your component access to React.Component's functions. The component also requires a render() method, this method returns HTML.

When creating a React component, the component's name must start with an uppercase letter.

Install create-react-app by running

npm-create-react-app my app

After installation, change the directory to the myapp folder by running

cd myapp

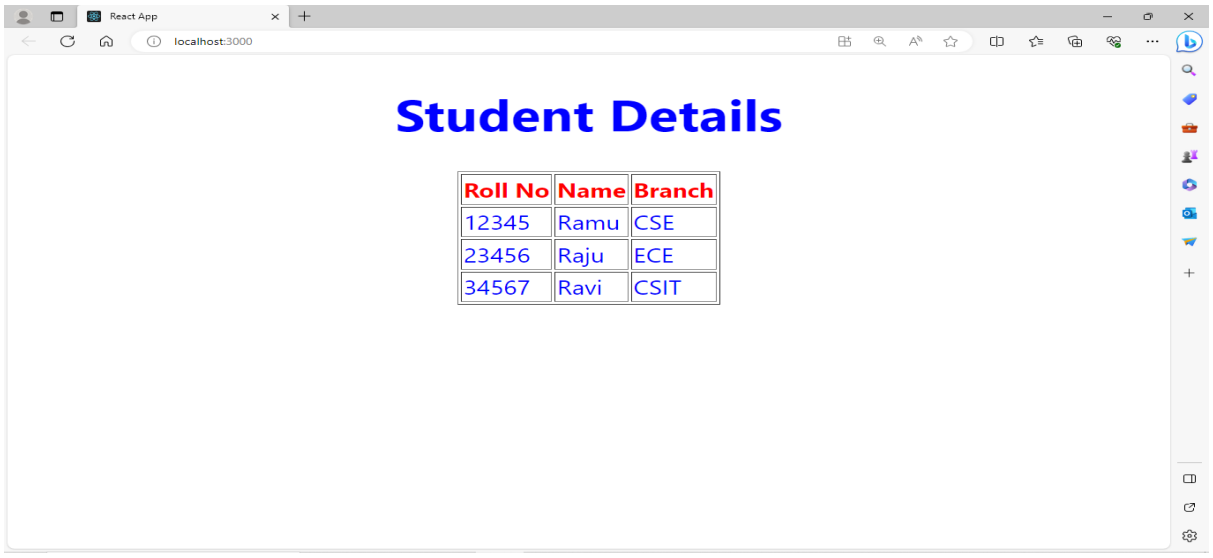
I'm using my app here but you can call your app anything you want. Run

npm start

To start the development server whichisalwaysavailableatport3000,

i.e.http://localhost:3000.Whentheserverstartsrunningonport3000,youseeaspinner.

SAMPLE OUTPUT:



Experiment – 4

AIM:
Create a Stateful Class component for Employee and display the details using tag.

DESCRIPTION:
' ' tag is a fundamental HTML element used for creating a container or a "division" in the user interface. It doesn't have any specific behaviour in React; instead, it's primarily used for structuring and grouping other elements and components.

Install create-react-app by running

npm-create-react-app myapp

After installation, change the directory to the myapp folder by running

cd myapp

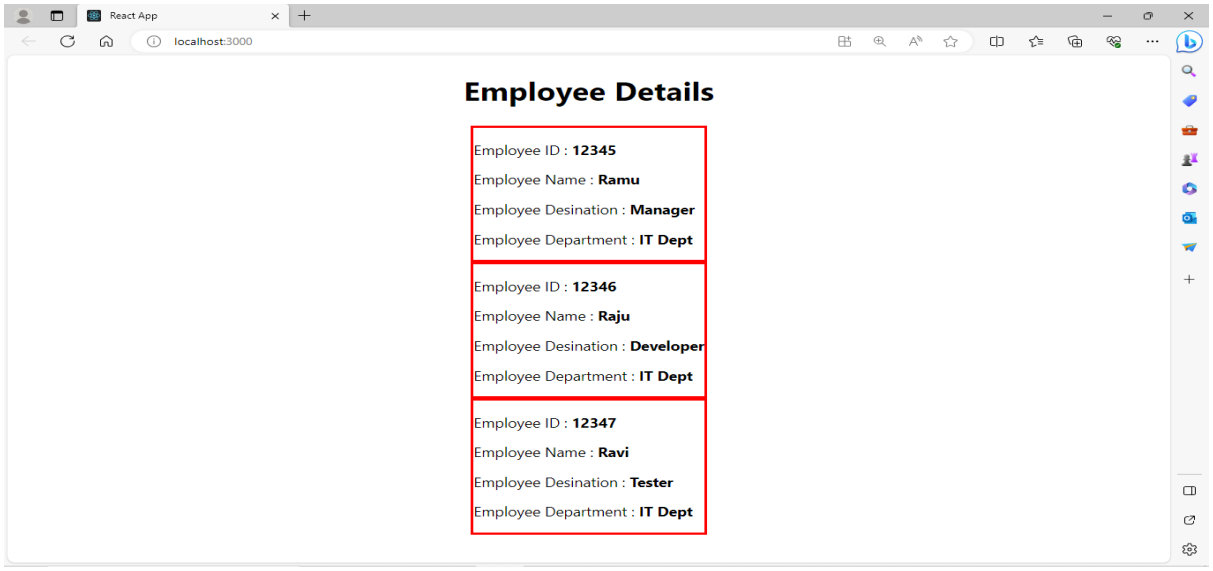
I'm using myapp here but you can call your app anything you want. Run

npm start

To start the development server whichisalwaysavailableatport3000,

i.e.http://localhost:3000.Whentheserverstartsrunningonport3000,youseeaspinner.

SAMPLE OUTPUT:



Experiment – 5

AIM:
Create a Stateless function component for Student and display the details along with percentage.

DESCRIPTION:
A stateless function component is a typical React component that is defined as a function that does not manage any state. There are no constructors needed, no classes to initialize, and no lifecycle hooks to worry about. These functions simply take props as an input and return JSX as an output.

Install create-react-app by running

npm-create-react-app myapp

After installation, change the directory to the myapp folder by running

cdmyapp

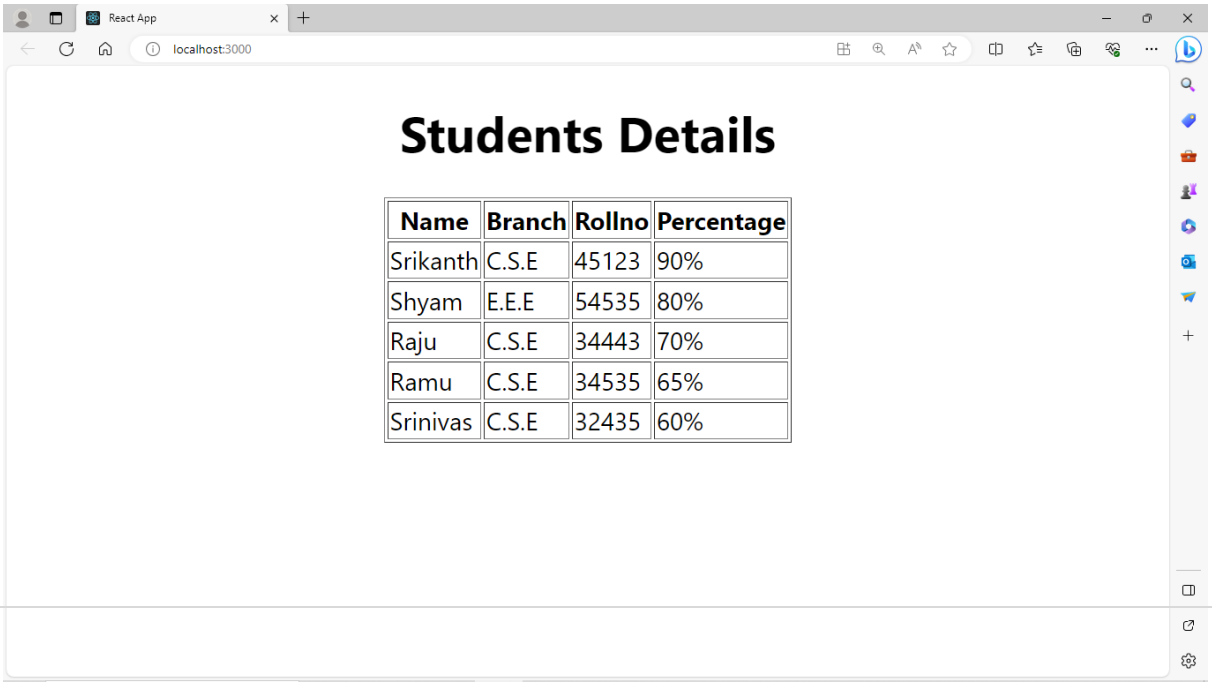
I'm using myapp here but you can call your app anything you want. Run

Nnpm start

To start the development server whichisalwaysavailableatport3000,

i.e.http://localhost:3000.Whentheserverstartsrunningonport3000,youseeaspinner.

SAMPLE OUTPUT:



Experiment – 6

AIM:
Create a Multipage React App using page routing with react-router-dom library

DESCRIPTION:
In single-page applications (SPAs) built with React, routing helps navigate between multiple "pages" without reloading the browser. React Router DOM, a popular library for managing routing, allows developers to set up dynamic and nested routes within an application. This enables users to interact with different views, maintaining a seamless user experience. In this approach, each "page" in the React app corresponds to a component, and routing controls which component displays based on the URL.

A multi-page React application with efficient page routing using the React Router DOM library, ensures smooth navigation across different pages.

SOLUTION:

- Begin by installing `react-router-dom` via npm:
  npm install react-router-dom

- Import the necessary components from `react-router-dom` to configure routing within the app.

- In `App.js`, set up `BrowserRouter`, `Routes`, and `Route` components.

- Each route corresponds to a path and renders a specific component.

Folder Structure
To create an application with multiple page routes, let's first start with the file structure.
Within the src folder, we'll create a folder named pages with several files:
src\pages\:

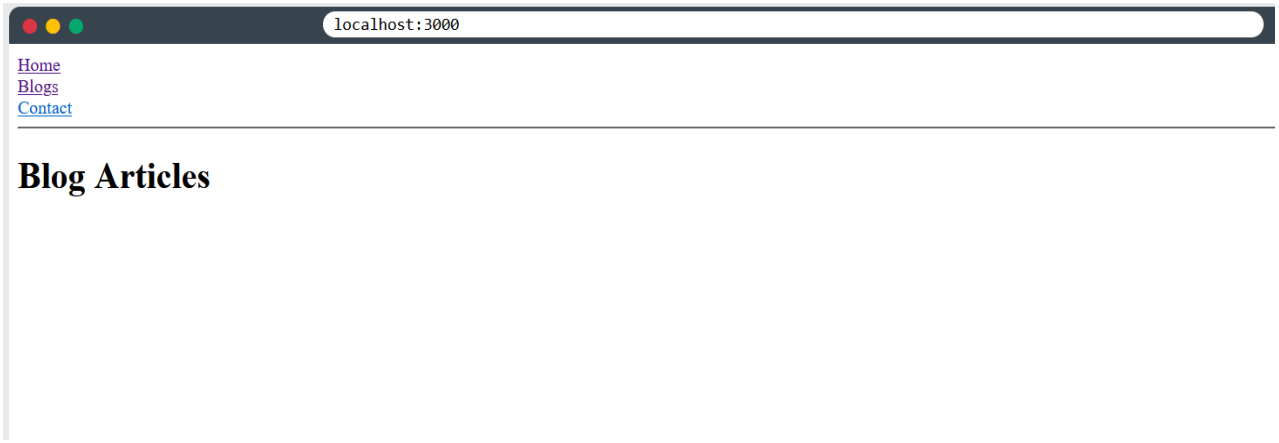
- Layout.js
- Home.js
- Blogs.js
- Contact.js
- NoPage.js

Each file will contain a very basic React component.

Use React Router to route to pages based on URL:

Now we will use our Router in our index.js file.

SAMPLE OUTPUT:



Experiment – 7

AIM:
Build a Counter app using the useState hook

DESCRIPTION:
React.js has become very popular to the extent that almost every frontend developer wants to learn how to use it. To get up and running with React, we have to set up our development environment by installing React and will be using the CLI (Command Line Interface) tool create-react-app, which is very popular in the React ecosystem.

Install create-react-app byrunning

npm-create-react-app myapp

After installation, change the directory to the myapp folder by running

cd myapp

I'm usingmyapp here but you can call your app anything you want. Run

npm start

To start the development server whichisalwaysavailableatport3000,

i.e.http://localhost:3000.Whentheserverstartsrunningonport3000,youseeaspinner.

To start bringing the counter app to life, import the useStatehook from React by typing"import React { useState } from 'react'" and the app.css file by typing "import './app.css'". Declare a function called App and return a div with CounterApp in an h1 tag as shown in the snippet below:

We must export our App component usingES6 modules, that's why you can see

export default App

in the last line of the snippet.

Now, you should have CounterApp shown in anh1 tag in the browser.

To start building the counter app, we have to declare a state with our useState hook. This is normally done by declaring two variables, the state and another to update the state,setState. This is done by using array destructuring and initializing the state to 0.

Right before there turn statement, React allows us to write pure JavaScript, so,we can pass the identifier into the on Clicks then write the functions before the return statement.

To make our little app a little more beautiful, let's add the styles in the snippet below:

SAMPLE OUTPUT:

Experiment – 8

AIM:
Build a FavoriteColor component using the useState hook.

DESCRIPTION:

The useState hook allows us to store and update values within functional components. Here, we use useState to store the user's favorite color and display it dynamically. When the user types in their favorite color, the component updates in real-time to reflect their choice.

SOLUTION:

1. Setup:
   - Import the `useState` hook from React.
   - Initialize a state variable `color` with an empty string, and a function `setColor` to update it.

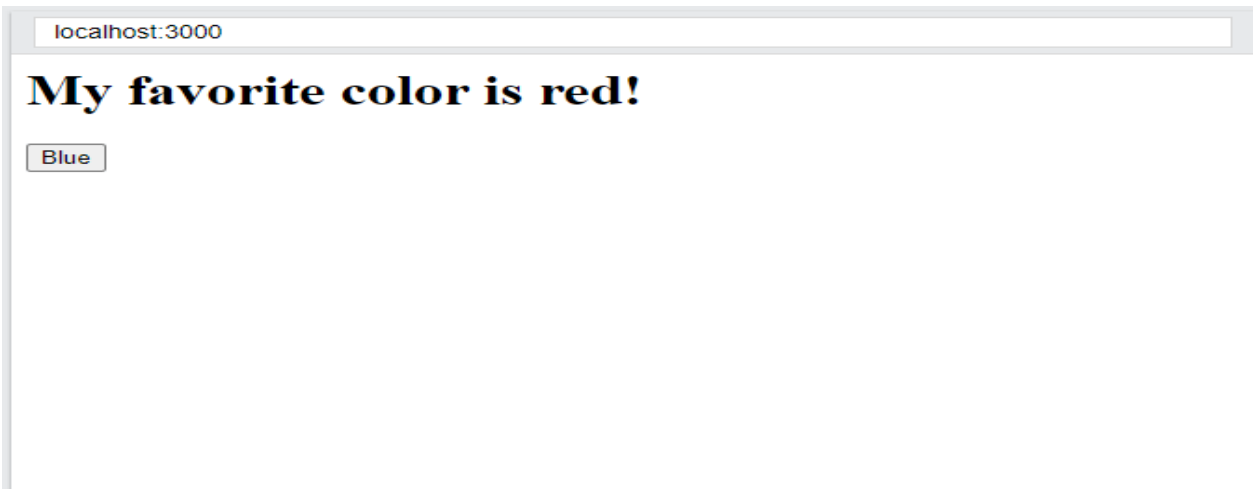
2. Create the Component:
   - The component includes an input field for users to enter their favorite color.
   - A dynamic message displays the color entered.

3. Explanation:
   - `useState` initializes `color` with an empty string. `setColor` is used to update this state.
   - `onChange` Event updates the `color` state each time the input changes.
   - Dynamic Display: The paragraph below the input field reflects the latest `color` value.

4. Usage:
   - Add `<FavoriteColor />` to your main app or another component to render it.

This component is a simple, interactive way to introduce state management with `useState` in React.

SAMPLE OUTPUT:



Experiment – 9

AIM:

Implement pagination using the useState, useEffect hooks and Fetching Data.

DESCRIPTION:

Requirements

- react-paginate
- axios

Create your react project

```
yarn create react-app projectname
```

and install the required modules

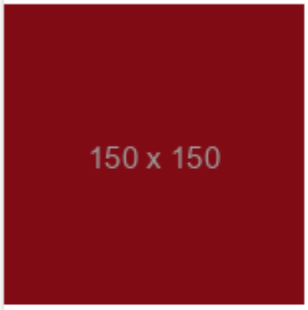
```
yarn addaxios react-paginate
```

and then open your **app.js**file.

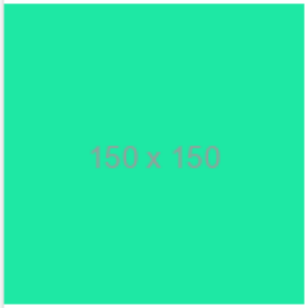
1. We will first import our hooks from the react and also import the required modules.
2. Nowcreateafunctionalcomponentandinsidethatinitializesomestateusing React **useState** hook.
3. Now we will create a **getData** function to load our data from the dummy API usingAxios.Thiswillbeanasynchronousfunctionthatwillfetch5000arraysof images from JSON placeholder API. Paste the below code inside the functional component.
4. Now call that **getData** methodinsideReactuseEffectmethod.
5. Create a method to handle our page click.Paste the below code inside the functional component.
6. Now all you need is to return your **data** state and **ReactPaginate** tag which we previously imported from react-paginate.
7. We also need to pass our **pageCount** state to react paginatepage Count props and **handle PageClick**methodtoonPageChangeprops.

SAMPLE OUTPUT:

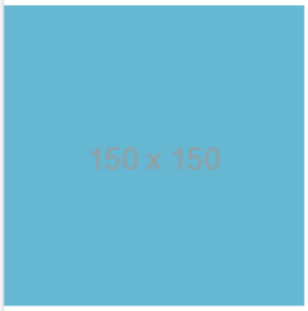
beatae et provident et ut vel



nihil at amet non hic quia qui



mollitia soluta ut rerum eos aliquam consequatur perspiciatis maiores



prev 1 2 3 4 5 6 ... 499 500 next

Experiment – 10

AIM:
Build Timer & Counter components using useEffect Hooks without and with Dependency respectively.

DESCRIPTION:

The useEffect Hook allows you to perform side effects in your components.
Some examples of side effects are: fetching data, directly updating the DOM, and timers.

useEffect accepts two arguments. The second argument is optional.

useEffect(<function>, <dependency>)

Timer Component:

Use setTimeout() to count 1 second after initial render:

EXPLANATION:
It keeps counting even though it should only count once!

useEffect runs on every render. That means that when the count changes, a render happens, which then triggers another effect.

This is not what we want. There are several ways to control when side effects run.

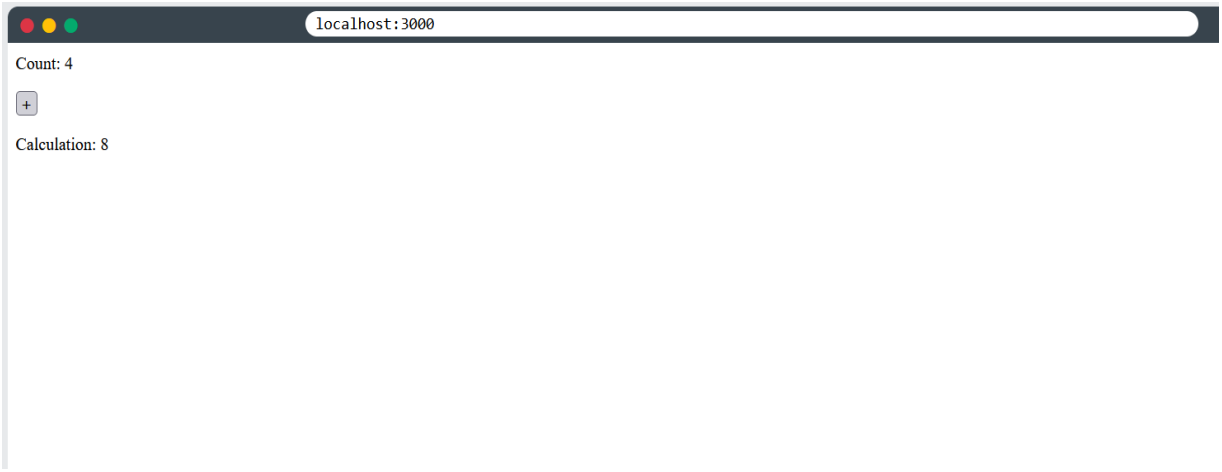
We should always include the second parameter which accepts an array. We can optionally pass dependencies to useEffect in this array.

SAMPLE OUTPUT:

Counter Component:

useEffect Hook that is dependent on a variable.
If the count variable updates, the effect will run again:

SAMPLE OUTPUT:



Experiment – 11

AIM:
Render a form component with validation using the useState hook.

DESCRIPTION:

The use State() is a hook in ReactJs which allows functional component to have a state. We pass the initial state in this function, and it returns us a variable and a function to update that state.

- We have to import the useState() hook from the react

  package. import{ useState } from 'react';
- SyntaxtocreatestateusinguseState()hook:

  const[state,updateState]=useState("InitialValue")

The useState() returns a list with two-element. first is the state itself, and the second is the function to update this state.

Creating React Application:

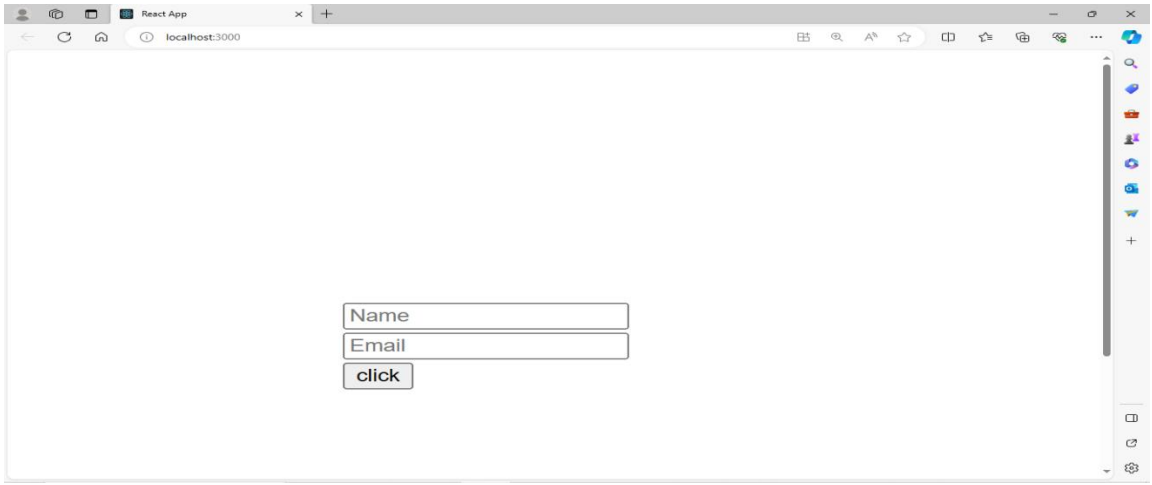
Step 1: Create a React application using the following command: npx create-react-app foldername

Step 2: After creating your project folder, i.e., foldername, move to it using the following command:

cd folder name

Validation of Input value in React allows an error message to be displayed if the user has not filled out the form with the expected value. There are many ways to validate input value with React.

SAMPLE OUTPUT:



Experiment – 12

AIM:
Create a To-do List app using the useState, useEffect and useContext hooks.

DESCRIPTION:

Hooks were introduced in React16.8.They allow the use of state and other React features by using functional components. There are various types of hooks available in React for example **useState**,**useEffect**, and **useContext** among others.FortheTo-doList project, we will only be using the useState hook.

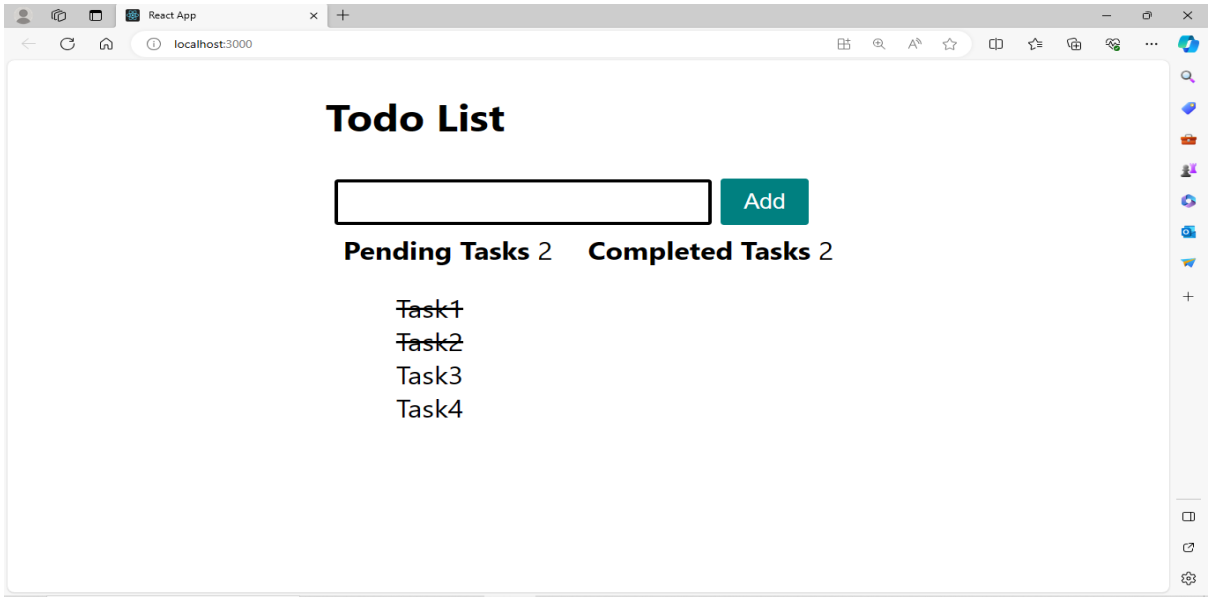
useState: allows adding of state to a functionalcomponent.

Styled-component on the other hand is a popular library that is used to style react applications. It allows writing actual CSS in your JavaScript.

To-DoApp

The To Do App that we are going to build will allow a user to add a task to a list of to-do items. Once the task is added, the user will be ableto mark it as completed once it's done. When you click on atask if it was pending it will be marked as completebycrossing thetask withaline.Therewillbeacountthatwillbedisplayingboththependingandcompletedtasks.

SAMPLE OUTPUT:



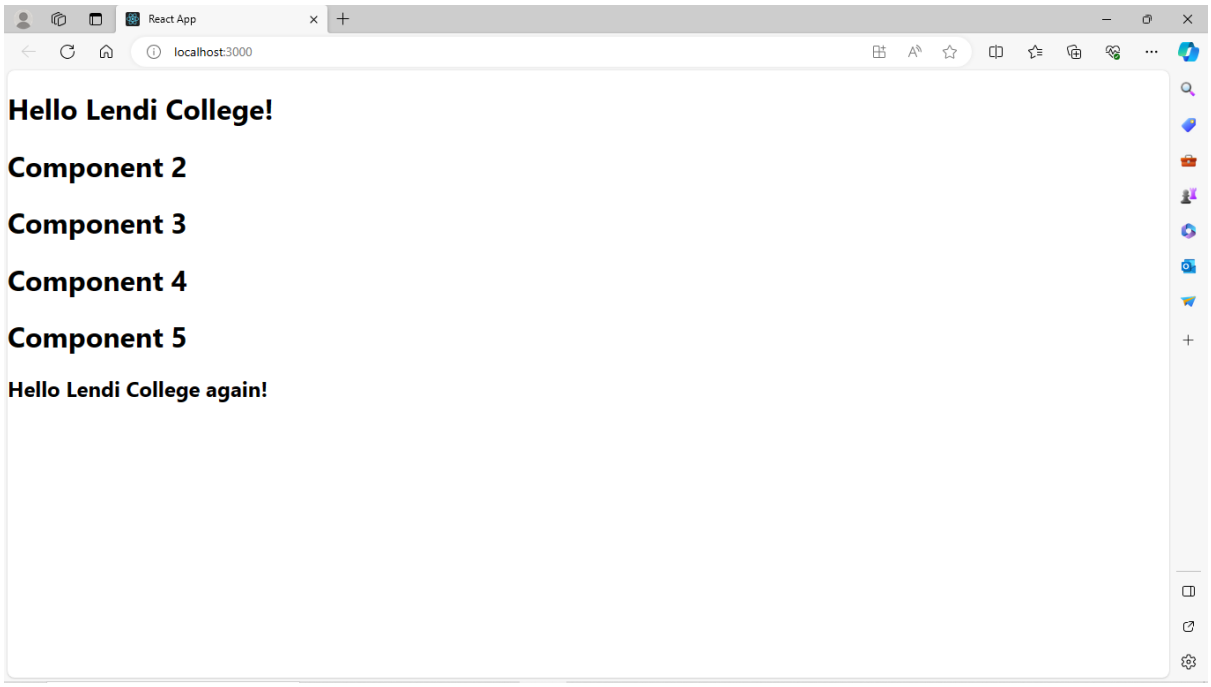
Experiment – 13

AIM:
Create a React App using the Nested Components and useContext hook.

DESCRIPTION:

A component can contain within itself, several more components. This helps in creating more complex design and interaction elements. Render method: In its minimal form, a component must define a render method specifying how it renders to the DOM.

SAMPLE OUTPUT:



Experiment – 14

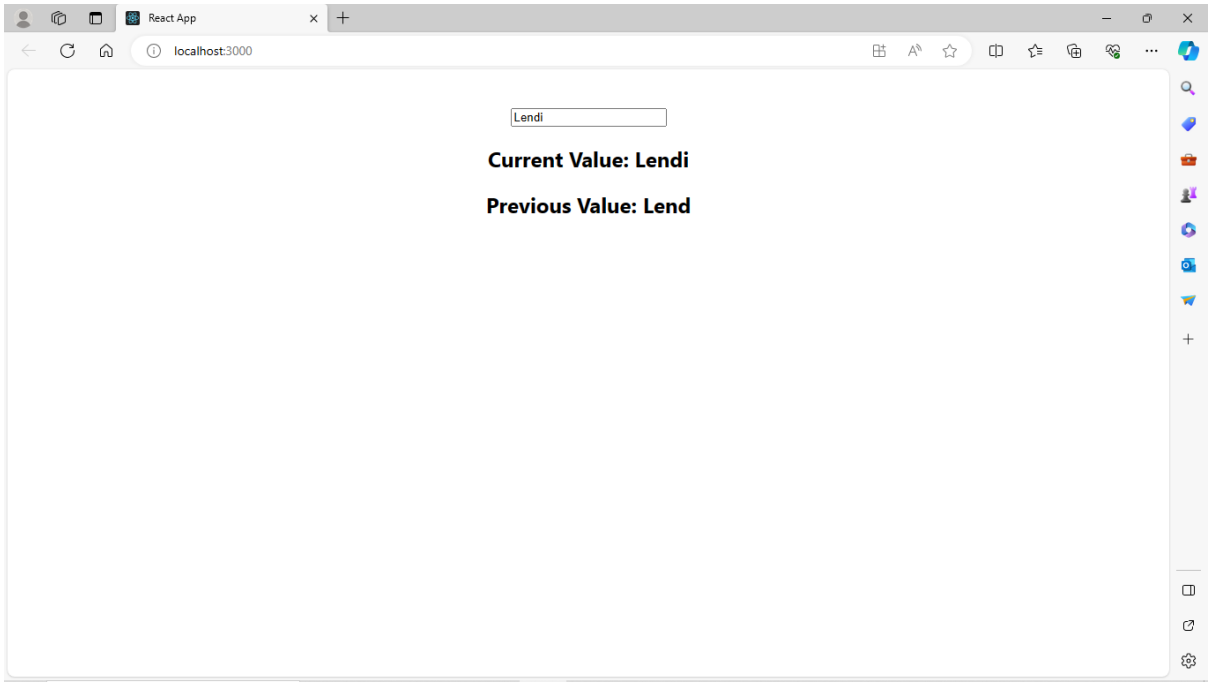
AIM:

Create a React App for tracking the state changes by using useRef hook.

DESCRIPTION:

The useRef Hook allows you to persist values between renders. It can be used to store a mutable value that does not cause a re-render when updated. It can be used to access a DOM element directly.

SAMPLE OUTPUT:



Experiment – 15

AIM:

Create a React App for tracking the multiple pieces of the state changes by using the useReducer and Custom State logic.

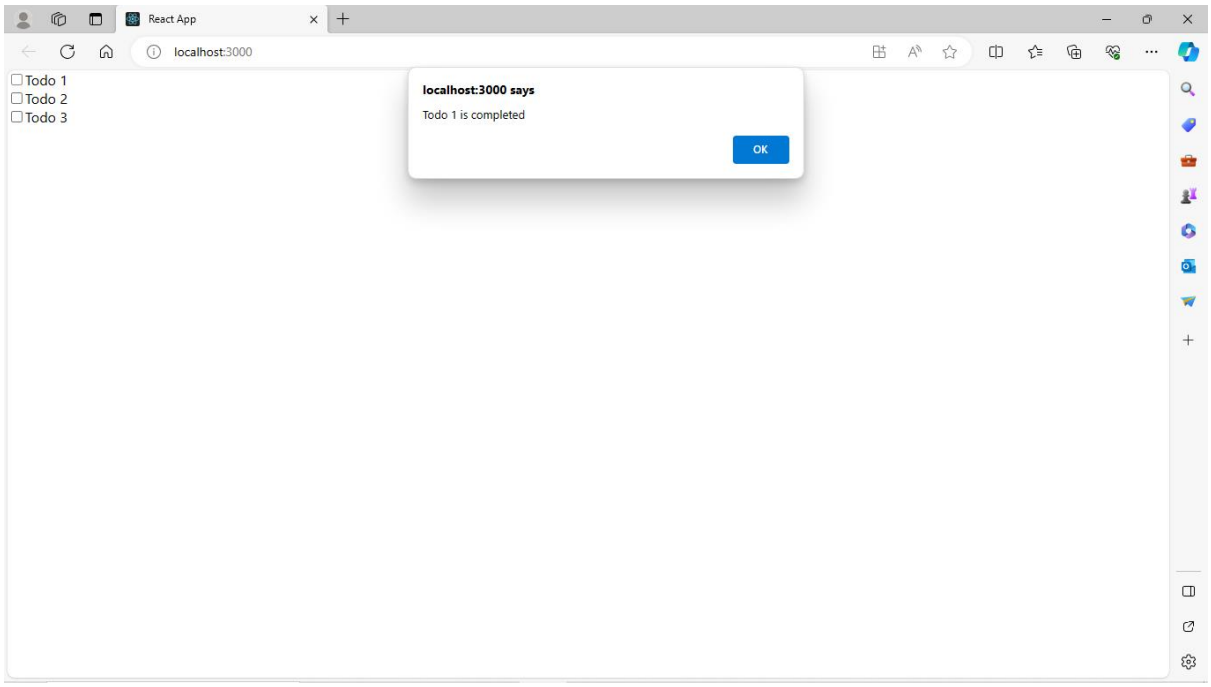
DESCRIPTION:

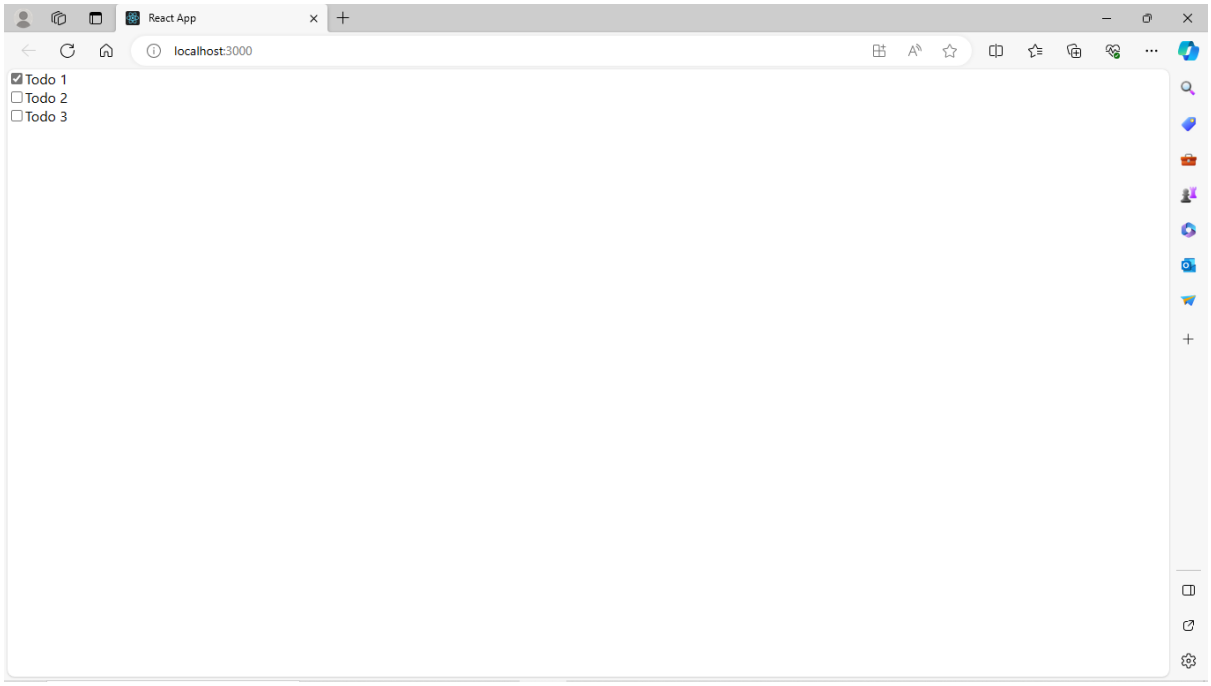
To create a React app for tracking multiple pieces of state changes using use Reducer and custom state logic, you can follow these steps:

Step 1: Set up a new React app using Create React App or your preferred method.

Step 2: Create a custom hook for state management. We'll call it use State With History to track state changes.

SAMPLE OUTPUT:




React App
localhost:3000
Todo 1
Todo 2
Todo 3

Experiment – 16

AIM:
Create a React App to demonstrate how to fetch data by using API call.

DESCRIPTION:

To fetch data from an API in React, you can use the fetch() function, a built-in function for making HTTP requests. Here is an example of how you can use fetch() to get data from an API and set it in the state of your React component. In this example, the fetchData() function is used to fetch the data from the API.
Install create-react-app by running

npm-create-react-app myapp

After installation,change the directory to the myapp folder by running

cd myapp

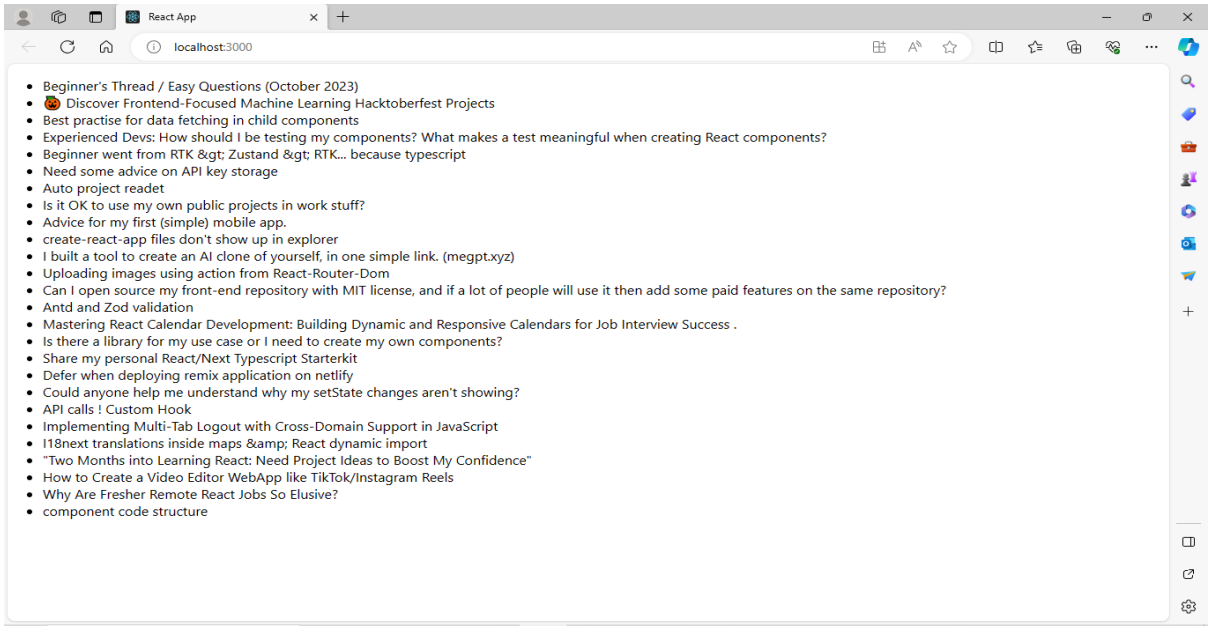
I'm using myapp here but you can call your app anything you want. Run

npm start

To start the development server whichisalwaysavailableatport3000,

i.e.http://localhost:3000.Whentheserverstartsrunningonport3000,youseeaspinner.

SAMPLE OUTPUT:



Experiment – 17

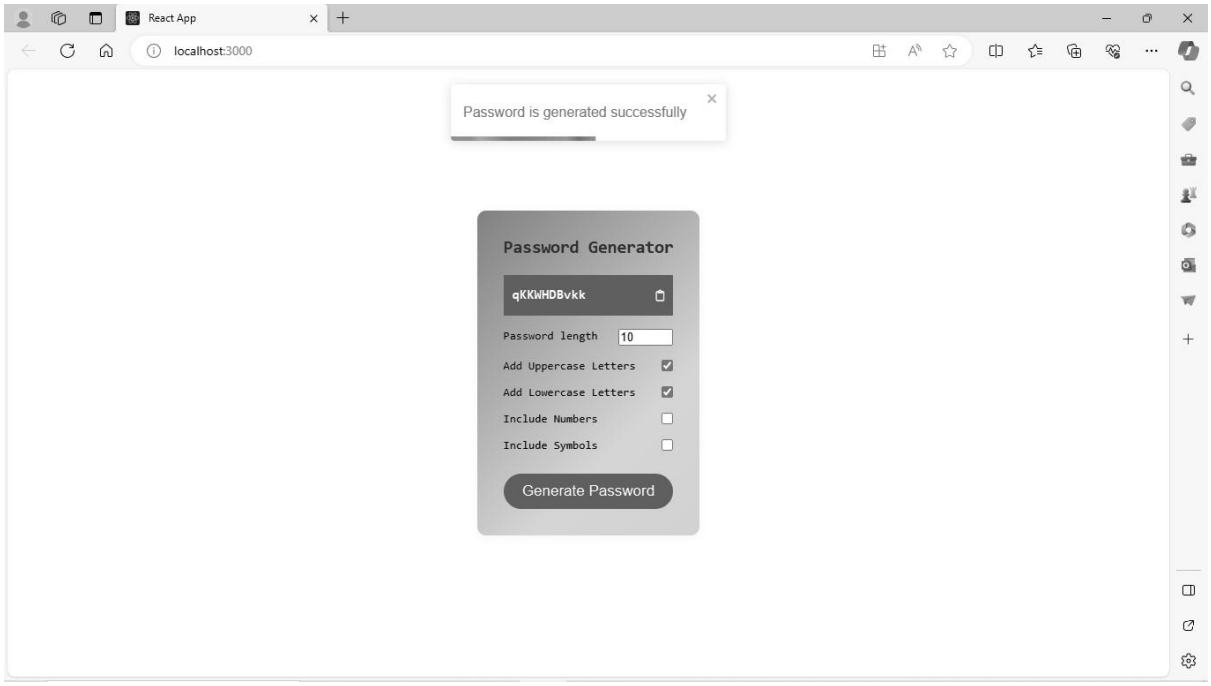
AIM:

Build a Password Generator App using React hooks.

DESCRIPTION:

To generate a random password, the application utilizes the `generate Password` function. This function constructs the password by iterating through selected character sets based on user preferences**.**

SAMPLE OUTPUT:



Experiment – 18

AIM:

Build a real-time chat app using the React hooks.

DESCRIPTION:

Getting setup for work

React requires both Node and npm.

Let's spin up a new project from the Terminal:

npxcreate-react-appsocket-client
cd socket-client

npm start

Now we should be able to navigate to http://localhost:3000 in the browser and get the default welcome page for the project.

Fromhere,we'regoingtobreaktheworkdownbythehookswe'reusing.Thisshouldhelpusunderstandthe hooks as we put them into practical use.

Using the useStatehook:

The first hook we'regoing to use is useState. It allows us to maintain state within our component as opposed to, say, having to write and initialize a class using this.state.Data that remains constant, such as username, is stored in useState variables. This ensures the data remains easily available while requiring a lot less code to write.

The main advantage of useState is that it's automatically reflected in the rendered component whenever we update the state of the app. If we were to use regular variables, they wouldn't be considered as the state of the componentandwouldhavetobepassedaspropstore-renderthecomponent.So,again,we'recutting out alot of work and streamlining things in the process.

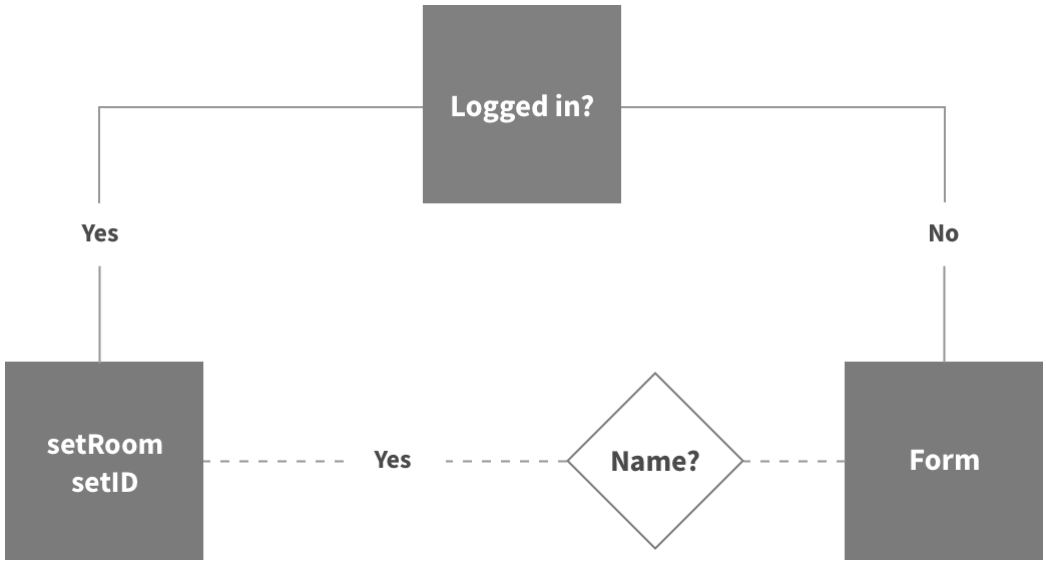
The hook is built right into React, so we can import it with a singleline:

importReact,{useState}from 'react';

Wearegoingtocreateasimplecomponentthatreturns"Hello"iftheuserisalreadyloggedinoralogiformif the user is logged out. We check the id variable for that.

Our form submissions will be handled by a function we're creating called handle Submit. It will check if the Name form field is completed. If it is, we will set the id and room values for that user. Other wise, we'll throwin a message reminding the user that

the Name field is required in order to proceed.



Using the use Effecthook:

WearegoingtousetheuseEffecthooktorunapieceofcodeonlywhentheapplicationloads.Thisensuresthat our code only runs once rather than every time the component re-renders with new data, which is good for performance.

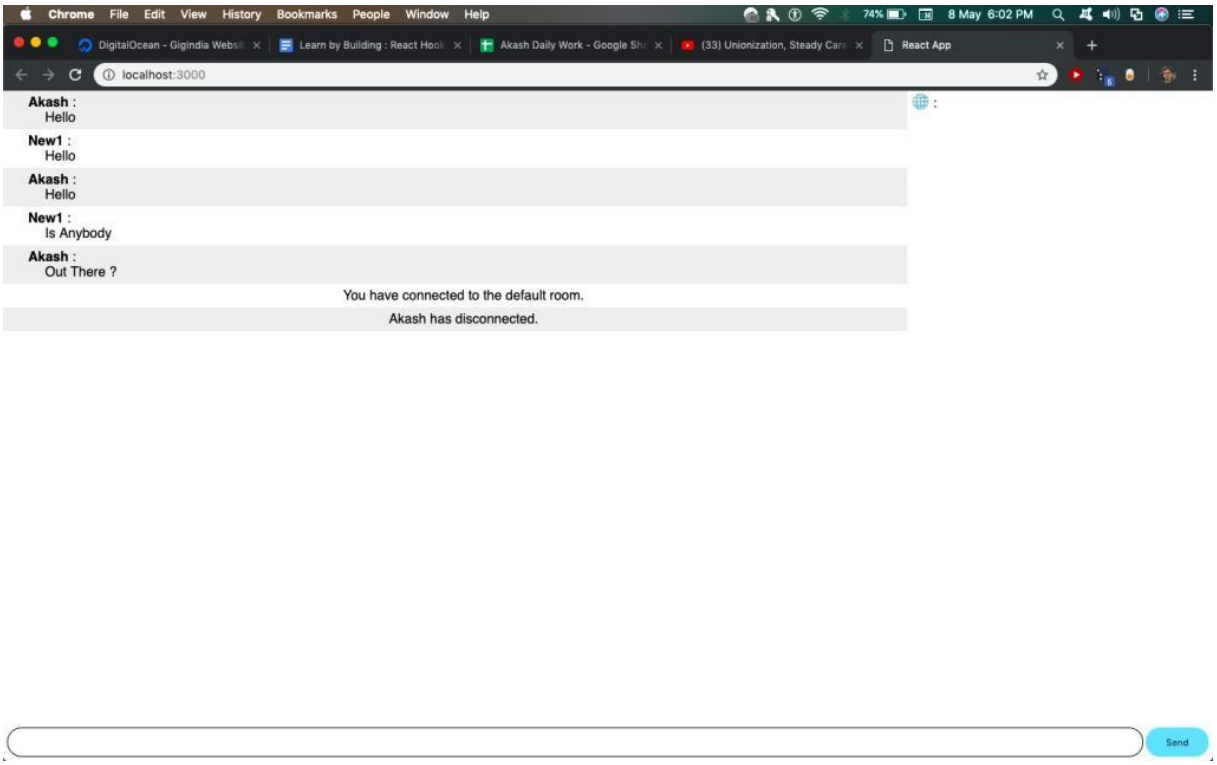
All we need to do to start using thehook is to importit.

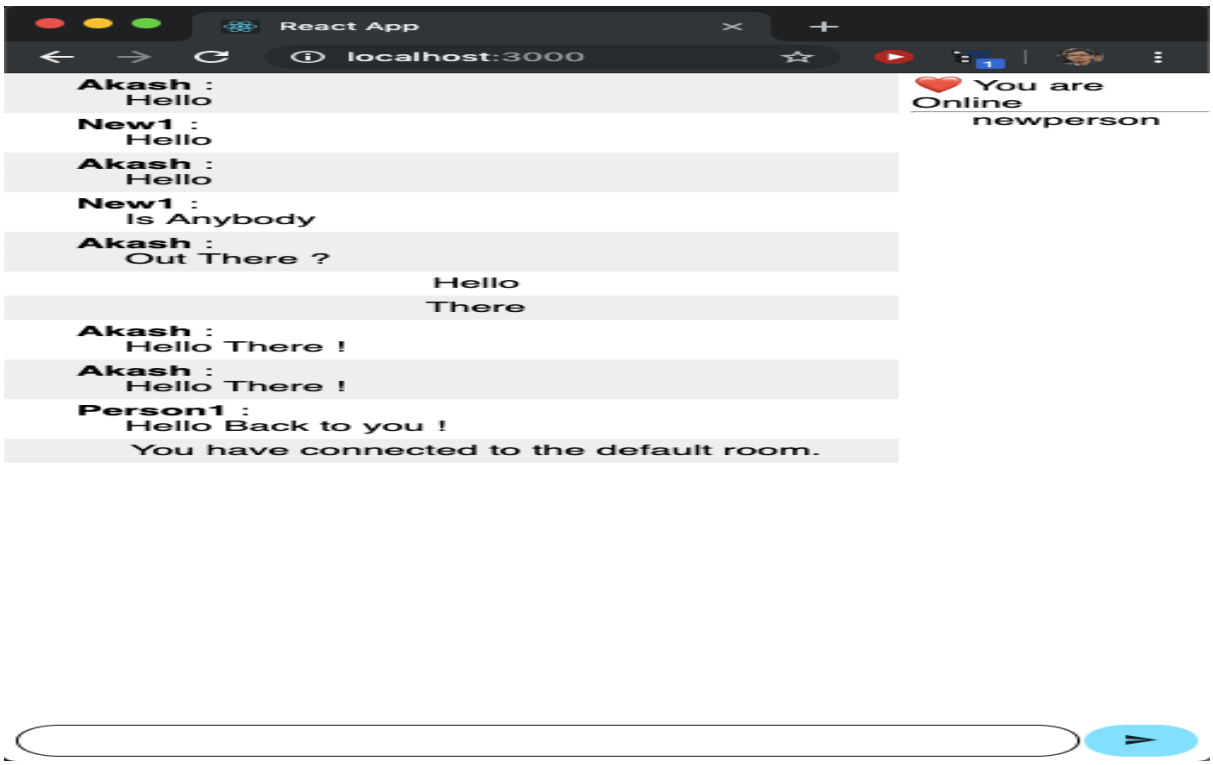
importReact,{useState,useEffect}from'react';

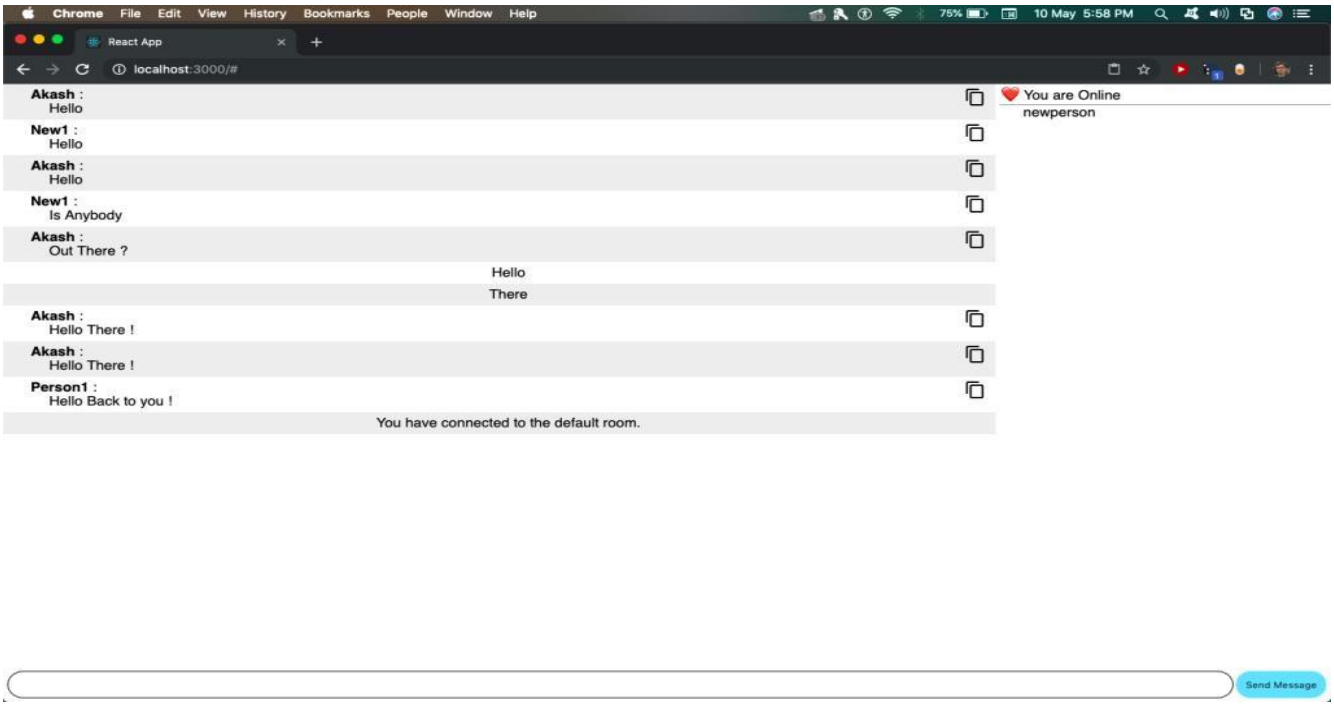
Wewillneedacomponentthatrendersa**message**oran**update**basedonthepresenceorabsence ofa **sende ID** in the array. Being the creative people we are, let's call that component Messages.

Anothertouchwe'llthrowinforgoodmeasureisa"join"messageiftheusernameandroomnamea recorrect. This triggers the rest of the event listeners and we can receive past messages sent in that room along with any updates required.

SAMPLE OUTPUT:






Chrome File Edit View History Bookmarks People Window Help
10 May 5:58 PM
React App
localhost:3000/#
Akash : Hello
New1 : Hello
Akash : Hello
New1 : Is Anybody
Akash : Out There ?
Hello
There
Akash : Hello There !
Akash : Hello There !
Person1 : Hello Back to you !
You have connected to the default room.
You are Online
newperson
Send Message