Experiment14: Create a TODO application in react with necessary components and deploy it into github.

Creating a simple TODO application in React involves defining components for adding, displaying, updating, and deleting tasks. Additionally, you can deploy the application to GitHub Pages

Steps to create the Application:

- **NPX:** It is a package runner tool that comes with npm 5.2version, npx is easy to use CLI tools. The npx is used for executing Node packages.

  npx create-react-app todo-react

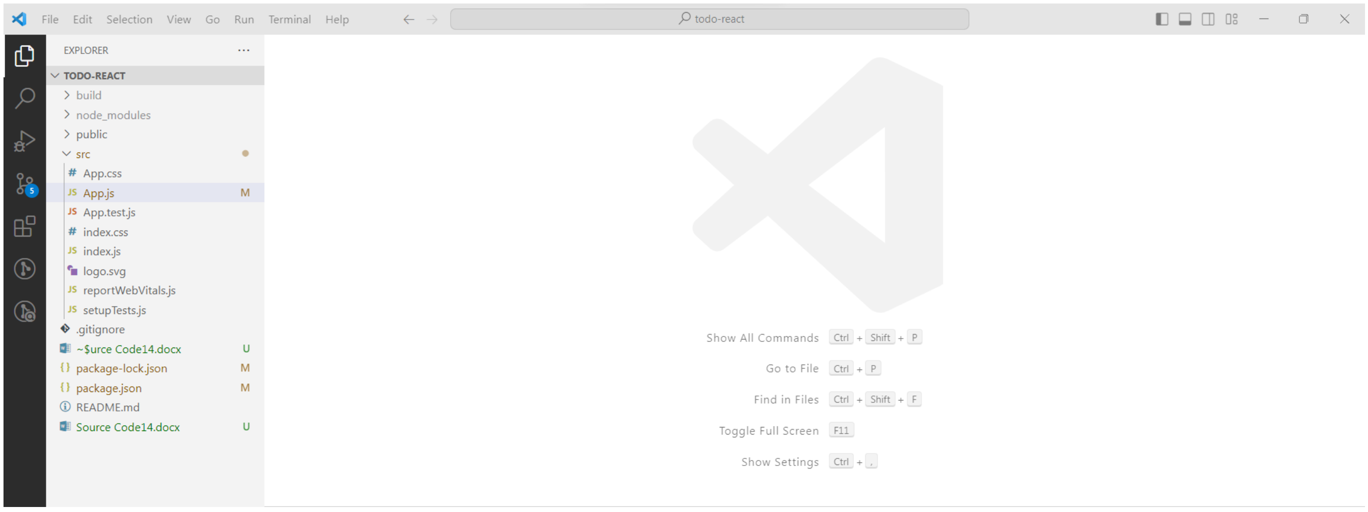
- Now, goto the folder

  cd todo-react

- Install the bootstrap and react-bootstrap module

  npm install bootstrap
  npm install react-bootstrap

After following the above steps, the Folder structure will look like:



```
TODO-REACT
  > build
  > node_modules
  > public
  v src
      App.css
      App.js               M
      App.test.js
      index.css
      index.js
      logo.svg
      reportWebVitals.js
      setupTests.js
    .gitignore
    ~$urce Code14.docx     U
    package-lock.json      M
    package.json           M
    README.md
    Source Code14.docx     U
```

File Edit Selection View Go Run Terminal Help
todo-react
EXPLORER
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Toggle Full Screen F11
Show Settings Ctrl + ,

```
// App.js File
import React, { Component } from "react";
import "bootstrap/dist/css/bootstrap.css";
import Container from "react-bootstrap/Container";
import Row from "react-bootstrap/Row";
import Col from "react-bootstrap/Col";
import Button from "react-bootstrap/Button";
import InputGroup from "react-bootstrap/InputGroup";
import FormControl from "react-bootstrap/FormControl";
import ListGroup from "react-bootstrap/ListGroup";

class App extends Component {
    constructor(props) {
        super(props);

        // Setting up state
        this.state = {
            userInput: "",
            list: [],
        };
    }
```

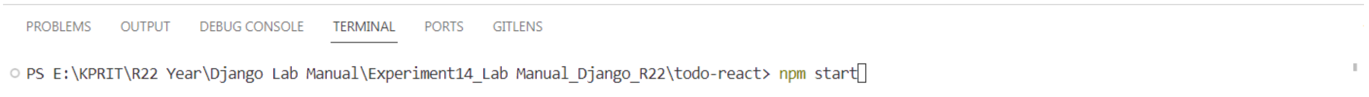
```
// Set a user input value
updateInput(value) {
    this.setState({
        userInput: value,
    });
}

// Add item if user input in not empty
addItem() {
    if (this.state.userInput !== "") {
        const userInput = {
            // Add a random id which is used to delete
            id: Math.random(),

            // Add a user value to list
            value: this.state.userInput,
        };

        // Update list
        const list = [...this.state.list];
        list.push(userInput);

        // reset state
        this.setState({
            list,
            userInput: "",
        });
    }
}

// Function to delete item from list use id to delete
deleteItem(key) {
    const list = [...this.state.list];

    // Filter values and leave value which we need to delete
    const updateList = list.filter((item) => item.id !== key);

    // Update list in state
    this.setState({
        list: updateList,
    });
}

editItem = (index) => {
const todos = [...this.state.list];
const editedTodo = prompt('Edit the todo:');
if (editedTodo !== null && editedTodo.trim() !== '') {
    let updatedTodos = [...todos]
    updatedTodos[index].value= editedTodo
    this.setState({
    list: updatedTodos,
});
}
}

render() {
    return (
        <Container>
            <Row
                style={{
                    display: "flex",
                    justifyContent: "center",
                    alignItems: "center",
                    fontSize: "3rem",
                    fontWeight: "bolder",
                }}
            >
                TODO LIST
            </Row>
            <hr />
            <Row>
                <Col md={{ span: 5, offset: 4 }}>
                    <InputGroup className="mb-3">
                        <FormControl
                            placeholder="add item . . . "
                            size="lg"
                            value={this.state.userInput}
                            onChange={(item) =>
                                this.updateInput(item.target.value)
                            }
                            aria-label="add something"
                            aria-describedby="basic-addon2"
                        />
                        <InputGroup>
                            <Button
                                variant="dark"
                                className="mt-2"
                                onClick={() => this.addItem()}
                            >
                                ADD
                            </Button>
                        </InputGroup>
                    </InputGroup>
                </Col>
            </Row>
            <Row>
                <Col md={{ span: 5, offset: 4 }}>
                    <ListGroup>
                        {/* map over and print items */}
                        {this.state.list.map((item, index) => {
                            return (
                            <div key = {index} >
                                <ListGroup.Item
                                    variant="dark"
                                    action
                                    style={{display:"flex",
                                            justifyContent:'space-between'
                                }}
                                >
                                    {item.value}
                                    <span>
                                    <Button style={{marginRight:"10px"}}
                                    variant = "light"
                                    onClick={() => this.deleteItem(item.id)}>
                                    Delete
                                    </Button>
                                    <Button variant = "light"
                                    onClick={() => this.editItem(index)}>
                                    Edit
                                    </Button>
                                    </span>
                                </ListGroup.Item>
                            </div>
                            );
```

```
                                    })}
                                </ListGroup>
                            </Col>
                        </Row>
                    </Container>
                );
            }
        }

export default App;
```

Steps to run the Application:

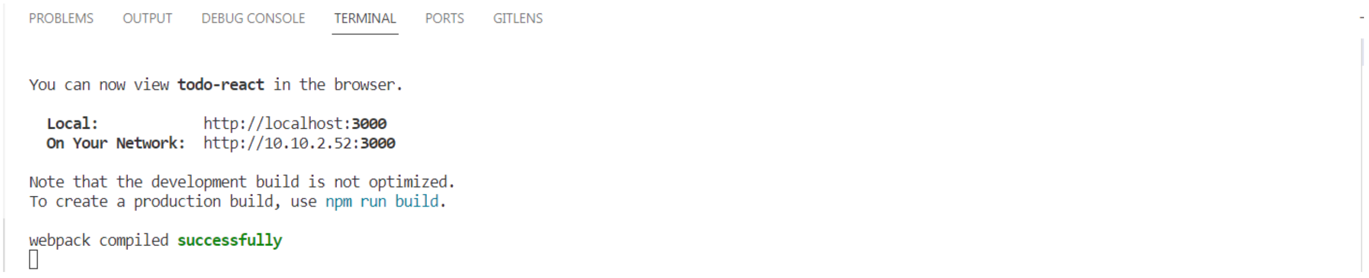
- Type the following command in the terminal:

  npm start

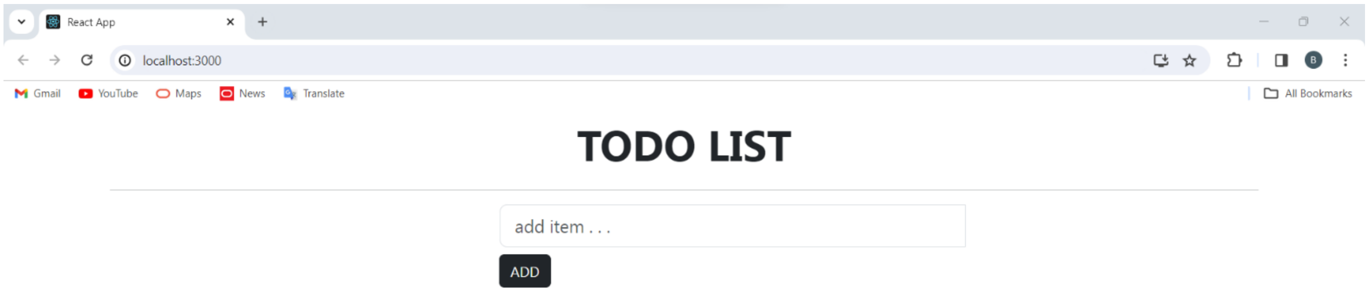


```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   GITLENS

PS E:\KPRIT\R22 Year\Django Lab Manual\Experiment14_Lab Manual_Django_R22\todo-react> npm start
```

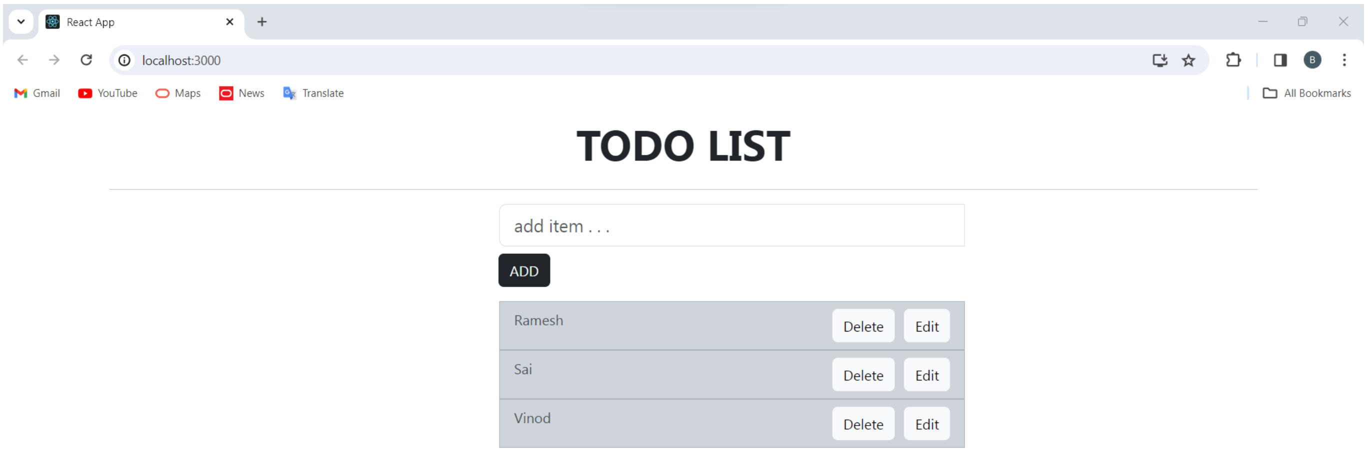
- Type the following URL in the browser: http://localhost:3000/



```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   GITLENS

You can now view todo-react in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.10.2.52:3000

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

webpack compiled successfully
```



After adding the TODO Lists



4. Deploy to GitHub Pages:

Install gh-pages**:**
npm install gh-pages --save-dev

Update package.json**:**
Add the following scripts to the **scripts** section in your **package.json** file:

```
"scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject",
    "predeploy": "npm run build",
    "deploy": "gh-pages -d build"
  }
```

Deploy to GitHub Pages:
npm run deploy

This will build your React app and deploy it to the **gh-pages** branch of your GitHub repository.

5. Access the Deployed App:

Your TODO application will be accessible at **https://your-username.github.io/repo-name**.

Official github link**: https://b-ramesh.github.io/react-todo/**

Make sure to replace **your-username** and **repo-name** with your GitHub username and repository name.

Your-username: b-ramesh.

Repo-name: react-todo