* React Bootstrap

npm install react-bootstrap bootstrap

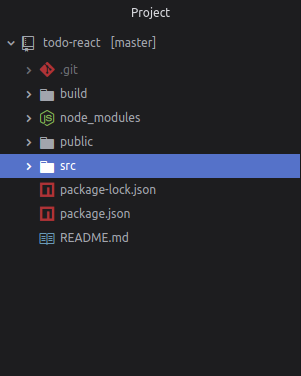
**Basic setup:** Start a project by the following command:

* [NPX:](https://www.geeksforgeeks.org/what-are-the-differences-between-npm-and-npx/) It is a package runner tool that comes with npm 5.2+, npx is easy to use CLI tools. The npx is used for executing Node packages. It greatly simplifies a number of things one of which is checked/run a node package quickly without installing it locally or globally.

npx create-react-app todo-react

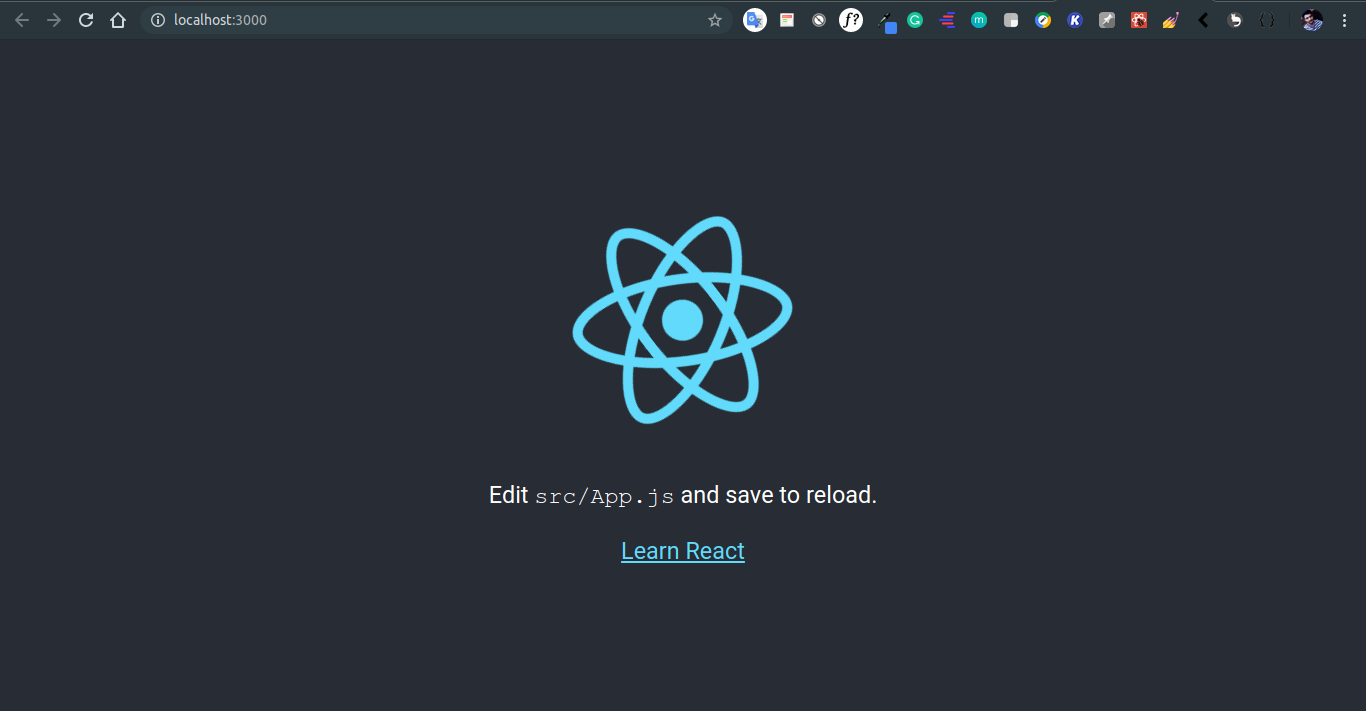
* Now, goto the folder

cd todo-react



* Start the server- Start the server by typing the following command in terminal:

npm start

open **http://localhost:3000/**  


* Change directory to src:

cd src

* Delete every thing inside the directory

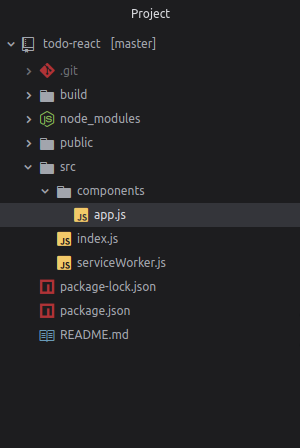
rm \*

* Now create **index.js** file

touch index.js

* This file will render our app to HTML file which is in **public folder**. Also create a folder name components with file name **app.js**

mkdir components && cd components && touch app.js

* **app.js** will contain our To-Do app:  
  
* **Edit index.js file in src:**

|  |
| --- |
| import React from 'react';  import ReactDOM from 'react-dom';  import App from './components/app';  import 'bootstrap/dist/css/bootstrap.min.css';    ReactDOM.render(<App />, document.getElementById('root')); |

* **Edit app.js in components:**

|  |
| --- |
| import React, {Component} from 'react';    // Bootstrap for react  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,      });      }      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.Append>              <Button                variant="dark"                size="lg"                onClick = {()=>this.addItem()}                >                ADD              </Button>            </InputGroup.Append>          </InputGroup>         </Col>     </Row>     <Row>       <Col md={{ span: 5, offset: 4 }}>          <ListGroup>            {/\* map over and print items \*/}           {this.state.list.map(item => {return(                <ListGroup.Item variant="dark" action                onClick = { () => this.deleteItem(item.id) }>                {item.value}              </ListGroup.Item>             )})}          </ListGroup>       </Col>     </Row>       </Container>      );    }  }    export default App; |

* **Save all files and start the server:**

npm start

* **Output:** Open **http://localhost:3000/** in browser:  
  