Agenda

- Class-based components, state and state manipulation
- Handling Different events
- Conditional Rendering
- Forms in react

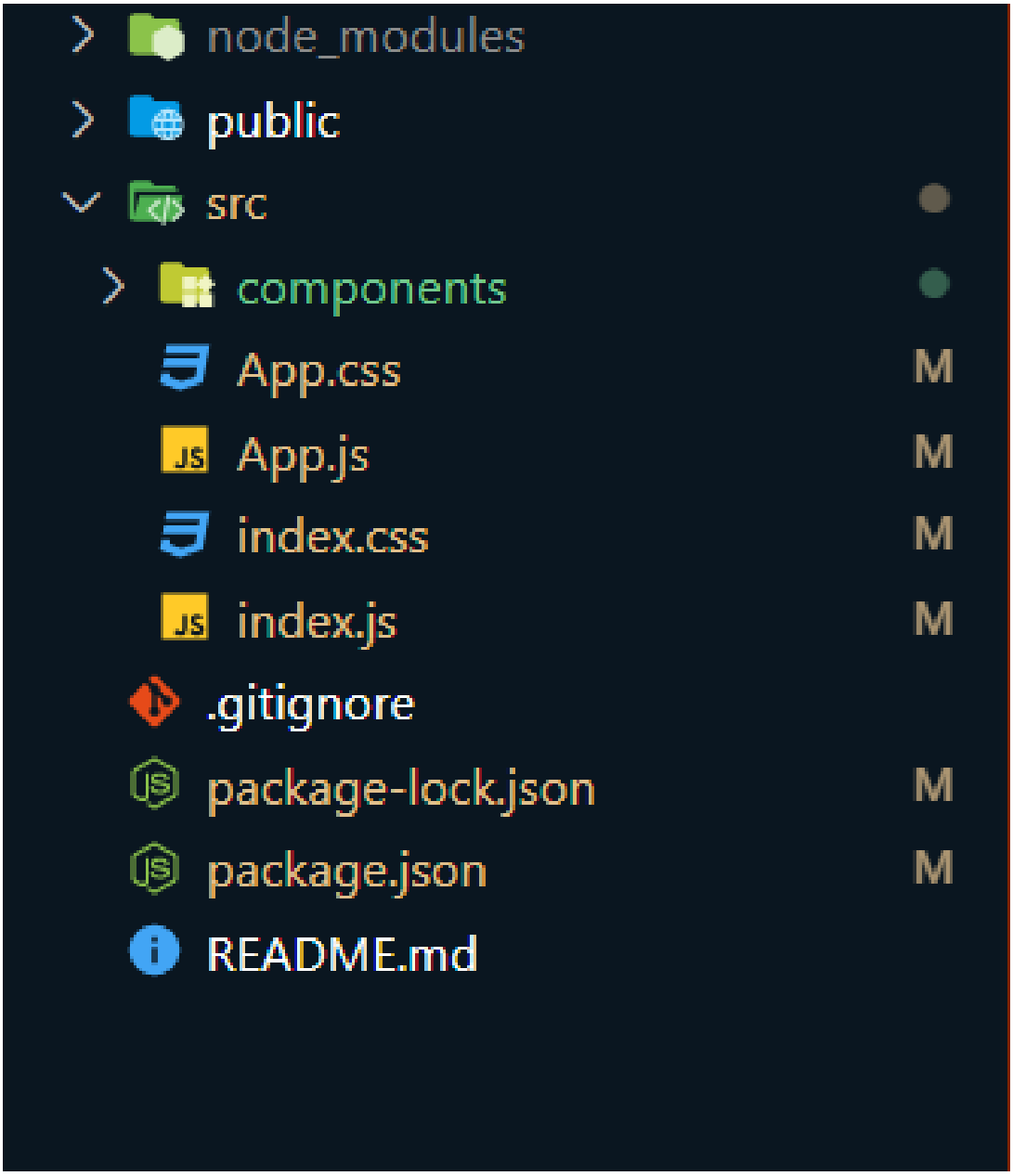
Lets Begin!

First, make sure you have the latest version of NodeJS installed. After that, we will install Create React App toolchain

Open a new terminal and type the following command

```
npx create-react-app react-todo-app
```

After deleting unnecessary files from our app our directory will look something like this:



After that run application by running the following command

```
npm start
```

In our ToDo application, we will use Bootstrap for a nice user interface. Run the following to install bootstrap

```
npm i bootstrap
```

To include Bootstrap in the application, we need open index.js file then import the bootstrap.css as shown below. Also we will be using the modal bootstrap so we will include the corresponsding js file for modal as well

```
import "bootstrap/dist/css/bootstrap.css";
import "bootstrap/js/dist/modal";
import ReactDOM from "react-dom/client";
import App from "./App";
import "./index.css";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<App />);
```

In file App.js file there is a functional Component with the following code

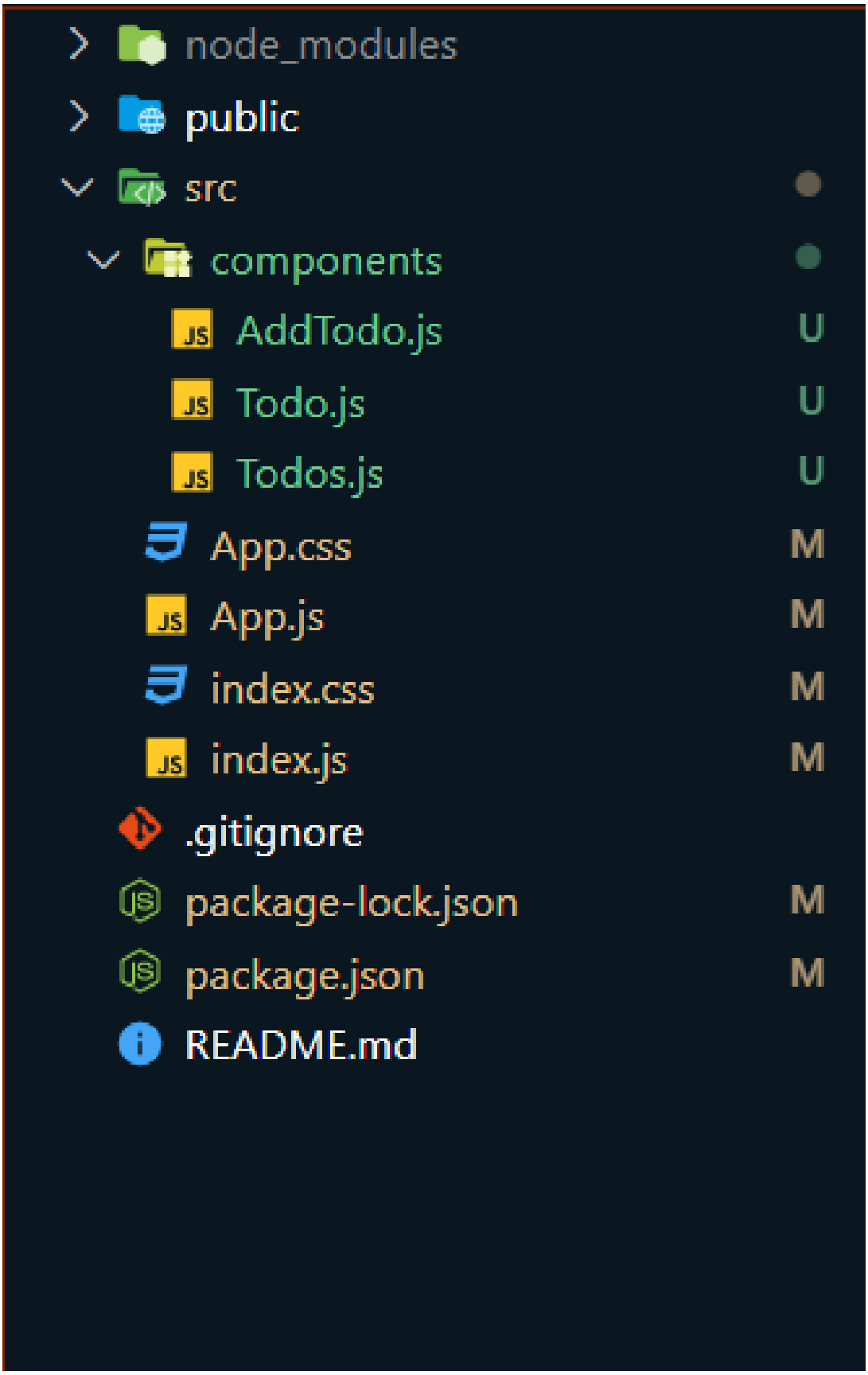
```
import "./App.css";

function App() {
    return <div></div>;
}

export default App;
```

For this project we will convert this into a Class component and also add our code

```
import React, { Component } from 'react';
import './App.css';
import Todos from './components/todos';

class App extends Component {

  render() {
    return (
      <div className="container">
        <h1 className="text-center">ToDo App in ReactJS</h1>
        <Todos />
      </div>
    );
  }
}

export default App;
```

Here we have another Component Todos, which is another Class Component to show List of Todos using Todo and one more Cla Component AddTodo to show Add Todo input with a button.

Let's create these three new Class Components (Todos, Todo & AddTodo) under components folder at "~components". After adding these componen our app directory will look like this:



Todos.js

In Todos class there is state property for adding dynamic content in Component. state is a special property in React component it is basically an object keep data which we will use in our class component. a state is a private object only accessible to the component itself.

In above Class Component *Todos*, we have added some methods which are defined as arrow notation as we are going to access these methods fro other Components like *Todo* and *AddTodo*.

getTime() is getting used to adding an ID to ToDo, this is without arrow notation as we are using this method locally.

handleDone, handleDelete and **addNewTodo, editTodo** are method whose references are getting passed as Component attributes, which will b called from their respective Components.

We have also included our modal to edit todo in this component.

```
import { Component } from "react";
import AddTodo from "./AddTodo";
import Todo from "./Todo";

export default class Todos extends Component {
    //Component state with default values
    state = {
        addTodoValue: "",
        editTodo: {},
        todos: localStorage.getItem("todos")
            ? JSON.parse(localStorage.getItem("todos"))
            : [],
    };

    //Local helper method to get date
    getTime() {
        let d = new Date();
        var n = d.getTime();
        return n;
    }

    //method called from Todo component
    handleDelete = todo => {
        const todos = this.state.todos.filter(t => {
            return t.id !== todo.id;
        });
        this.setState({ todos });
        localStorage.setItem("todos", JSON.stringify(todos));
    };

    handleDone = todo => {
        const todos = [...this.state.todos];
        todos.map(t => {
            if (t.id === todo.id) {
                t.isDone = !t.isDone;
            }
            return t;
        });
        this.setState({ todos });
        localStorage.setItem("todos", JSON.stringify(todos));
    };

    //method called from AddTodo component
    addNewTodo = value => {
        if (value) {
            const todos = [...this.state.todos];
            todos.push({
                id: this.getTime(),
                value: value,
                isDone: false,
            });
            this.setState({ addTodoValue: "", todos });
            localStorage.setItem("todos", JSON.stringify(todos));
        } else {
            alert("Please add a value");
        }
    };

    editTodo = todo => {
        const todos = [...this.state.todos];
        todos.map(t => {
            if (t.id === todo.id) {
                t.value = todo.value;
```

```
        }
        return t;
    });
    this.setState({ editTodo: {} });
    localStorage.setItem("todos", JSON.stringify(todos));
};

setEditValue = todo => {
    this.setState({
        editTodo: todo,
    });
};

render() {
    return (
        <div>
            {this.state.todos?.length <= 0 && (
                <div className='alert alert-info text-center' role='alert'>
                    <b>No Todos Added</b>
                </div>
            )}
            <table className='table'>
                <tbody>
                    {this.state.todos.map((todo, index) => (
                        <tr key={todo.id}>
                            <Todo
                                index={index + 1}
                                todo={todo}
                                fooDelete={this.handleDelete}
                                fooDoneDone={this.handleDone}
                                fooEdit={this.setEditValue}
                            />
                        </tr>
                    ))}
                    <tr>
                        <td colSpan='4' className='text-center'>
                            <AddTodo
                                fooAddTodo={this.addNewTodo}
                                addTodoValue={this.state.addTodoValue}
                            />
                        </td>
                    </tr>
                </tbody>
            </table>
            <div className='modal fade' id='exampleModal'>
                <div className='modal-dialog'>
                    <div className='modal-content'>
                        <div className='modal-header'>
                            <h5 className='modal-title' id='exampleModalLabel'>
                                Update Todo Value
                            </h5>
                            <button
                                type='button'
                                className='btn-close'
                                data-bs-dismiss='modal'
                                aria-label='Close'></button>
                        </div>
                        <div className='modal-body'>
                            <form
                                onSubmit={e => {
                                    e.preventDefault();
                                    this.editTodo(this.state.editTodo);
                                }}>
                                <div className='mb-3'>
                                    <label htmlFor='recipient-name' className='col-form-label'>
```

```
                                    Value:
                                </label>
                                {this.state.editTodo?.value && (
                                    <input
                                        type='text'
                                        className='form-control'
                                        value={this.state.editTodo.value}
                                        onChange={e =>
                                            this.setState({
                                                ...this.state,
                                                editTodo: {
                                                    ...this.state.editTodo,
                                                    value: e.target.value,
                                                },
                                            })
                                        }
                                    />
                                )}
                            </div>
                            <div className='modal-footer'>
                                <button
                                    type='button'
                                    className='btn btn-secondary'
                                    data-bs-dismiss='modal'>
                                    Close
                                </button>
                                <button
                                    type='submit'
                                    className='btn btn-primary'
                                    data-bs-dismiss='modal'>
                                    Update
                                </button>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    );
  }
}
```

Todo.js

Todo Class will represent a single Todo in the list and having methods **fooDoneDone** (*check/ uncheck event handler*) and **fooDelete** (delete butto event handler)

```
import React, { Component } from "react";

class Todo extends Component {
    render() {
        return (
            <React.Fragment>
                <td style={{ width: 10 }} className='text-center'>
                    {this.props.index}
                </td>
                <td style={{ width: 15 }} className='text-center'>
                    <input
                        type='checkbox'
                        defaultChecked={this.props.todo.isDone}
                        onChange={() => this.props.fooDoneDone(this.props.todo)}
                    />
                </td>
                <td>{this.renderTodo()}</td>
```

```
                <td style={{ width: 100 }} className='text-center'>
                    <button
                        data-bs-toggle='modal'
                        data-bs-target='#exampleModal'
                        type='button'
                        className='btn btn-warning btn-sm'
                        onClick={() => this.props.fooEdit(this.props.todo)}>
                        Edit
                    </button>
                </td>
                <td style={{ width: 100 }} className='text-center'>
                    <button
                        onClick={() => this.props.fooDelete(this.props.todo)}
                        className='btn btn-danger btn-sm'>
                        Delete
                    </button>
                </td>
            </React.Fragment>
        );
    }

    renderTodo() {
        if (this.props.todo.isDone) return <s>{this.props.todo.value}</s>;
        else return this.props.todo.value;
    }
}

export default Todo;
```

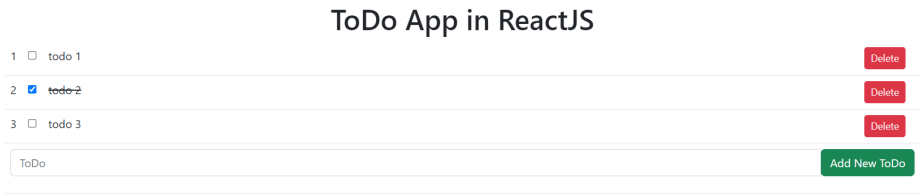
AddTodo.js

In AddTodo class we have **handleChange** method also in arrow notation as we need to call the **setState** method to replicate changes in view. setSta will be undefined if we don't have arrow method.

```
import { Component } from "react";

class AddTodo extends Component {
    state = {
        defaultValue: "",
        value: this.props.addTodoValue,
    };

    handleChange = e => {
        //Updating local component state
        this.setState({
            value: e.target.value,
        });
    };

    clearInput = () => {
        //Updating local component state
        this.setState({ value: "" });
    };

    addTodo = e => {
        e.preventDefault();
        //Call method reference in Todos component using props
        this.props.fooAddTodo(this.state.value);
        this.clearInput();
    };

    render() {
        return (
            <form onSubmit={this.addTodo}>
```

```
                <div className='input-group mb-3'>
                    <input
                        type='text'
                        className='form-control'
                        id='todoValue'
                        placeholder='ToDo'
                        value={this.state.value}
                        onChange={this.handleChange}
                    />
                    <div className='input-group-append'>
                        <button
                            className='btn btn-success'
                            type='submit'
                            id='button-addon2'>
                            Add New ToDo
                        </button>
                    </div>
                </div>
            </form>
        );
    }
}

export default AddTodo;
```

Final View:



Thank You !