**Assignment 7.5.2**

**Prerequisites**

Make sure you have the following installed on your system:

* Node.js
* npm or yarn
* MongoDB

**Step 1: Setting Up the Backend**

**1. Initialize the Backend Project**

Create a new directory for your backend project and initialize it:

mkdir backend

cd backend

npm init -y

**2. Install Dependencies**

Install the necessary dependencies:

npm install express mongoose cors dotenv bcrypt

npm install nodemon --save-dev

**3. Set Up the Server**

Create the following files:

* app.js
* models/list.js,user.js
* routes/auth.js,list.js
* conn/conn.js

**conn/conn.js:**

const mongoose = require("mongoose");

const conn = async (req, res) => {

  try {

    await mongoose

      .connect(

        "mongodb+srv://pradeeppssharma829:<password>@cluster0.mova428.mongodb.net/"

      )

      .then(() => {

        console.log("Connected");

      });

  } catch (error) {

    console.log(error);

  }

};

conn();

**models/list.js:**

const mongoose = require("mongoose");

const listSchema = new mongoose.Schema(

  {

    title: {

      type: String,

      required: true,

    },

    body: {

      type: String,

      required: true,

    },

    user: [

      {

        type: mongoose.Types.ObjectId,

        ref: "User",

      },

    ],

  },

  { timestamps: true }

);

module.exports = mongoose.model("List", listSchema);

**models/user.js:**

const mongoose = require("mongoose");

const userSchema = new mongoose.Schema({

  email: {

    type: String,

    unique: true,

    required: true,

  },

  username: {

    type: String,

    dunique: true,

    required: true,

  },

  password: {

    type: String,

    required: true,

  },

  list: [

    {

      type: mongoose.Types.ObjectId,

      ref: "List",

    },

  ],

});

module.exports = mongoose.model("User", userSchema);

**app.js:**

const express = require("express");

const app = express();

const cors = require("cors");

require("./conn/conn")

const path = require("path");

const auth = require("./routes/auth");

const list = require("./routes/list");

app.use(express.json());

app.use(cors());

app.use("/api/v1", auth);

app.use("/api/v2", list);

app.get("/", (req, res) => {

  app.use(express.static(path.resolve(\_\_dirname, "frontend", "build")));

  res.sendFile(path.resolve(\_\_dirname, "frontend", "build", "index.html"));

});

app.listen(3000, () => {

  console.log("Server Started");

});

**Step 2: Setting Up the Frontend**

**1. Initialize the Frontend Project**

Create a new directory for your frontend project and initialize it:

mkdir frontend

cd frontend

npx create-react-app .

**2. Install Dependencies**

Install Axios for making HTTP requests:

npm install axios

**3. Set Up Components**

Create the following files and directories:

* src/components/about.js
* src/components/home.js
* src/components/Todo.js
* src/App.js

**src/components/home.js:**

import React from "react";

import "./home.css";

const Home = () => {

  return (

    <div className="home d-flex justify-content-center align-items-center">

      <div className="container d-flex justify-content-center align-items-center flex-column ">

        <h1 className="text-center">

          Organize your <br /> work and life, finally.

        </h1>

        <p>

          Become focused, organized, and calm with <br />

          todo app. The World's #1 task manager app.

        </p>

        <button class="home-btn p-2">Make Todo List</button>

      </div>

    </div>

  );

};

export default Home;

**src/components/todo.js:**

import React, { useEffect, useState } from "react";

import "./todo.css";

import TodoCards from "./TodoCards";

import { ToastContainer, toast } from "react-toastify";

import "react-toastify/dist/ReactToastify.css";

import Update from "./Update";

import axios from "axios";

let id = sessionStorage.getItem("id");

let toUpdateArray = [];

const Todo = () => {

  const [Inputs, setInputs] = useState({

    title: "",

    body: "",

  });

  const [Array, setArray] = useState([]);

  const show = () => {

    document.getElementById("textarea").style.display = "block";

  };

  const change = (e) => {

    const { name, value } = e.target;

    setInputs({ ...Inputs, [name]: value });

  };

  const submit = async () => {

    if (Inputs.title === "" || Inputs.body === "") {

      toast.error("Title Or Body Can't Be Empty");

    } else {

      if (id) {

        await axios

          .post(`${window.location.origin}/api/v2/addTask`, {

            title: Inputs.title,

            body: Inputs.body,

            id: id,

          })

          .then((response) => {

            console.log(response);

          });

        setInputs({ title: "", body: "" });

        toast.success("Your Task Is Added");

      } else {

        setArray([...Array, Inputs]);

        setInputs({ title: "", body: "" });

        toast.success("Your Task Is Added");

        toast.error("Your Task Is Not Saved ! Please SignUp");

      }

    }

  };

  const del = async (Cardid) => {

    if (id) {

      await axios

        .delete(`${window.location.origin}/api/v2/deleteTask/${Cardid}`, {

          data: { id: id },

        })

        .then(() => {

          toast.success("Your Task Is Deleted");

        });

    } else {

      toast.error("Please SignUp First");

    }

  };

  const dis = (value) => {

    document.getElementById("todo-update").style.display = value;

  };

  const update = (value) => {

    toUpdateArray = Array[value];

  };

  useEffect(() => {

    if (id) {

      const fetch = async () => {

        await axios

          .get(`${window.location.origin}/api/v2/getTasks/${id}`)

          .then((response) => {

            setArray(response.data.list);

          });

      };

      fetch();

    }

  }, [submit]);

  return (

    <>

      <div className="todo">

        <ToastContainer />

        <div className="todo-main container d-flex justify-content-center align-items-center my-4 flex-column">

          <div className="d-flex flex-column todo-inputs-div w-lg-50 w-100 p-1">

            <input

              type="text"

              placeholder="TITLE"

              className="my-2 p-2 todo-inputs"

              onClick={show}

              name="title"

              value={Inputs.title}

              onChange={change}

            />

            <textarea

              id="textarea"

              type="text"

              placeholder="BODY"

              name="body"

              className=" p-2 todo-inputs"

              value={Inputs.body}

              onChange={change}

            />

          </div>

          <div className=" w-50 w-100 d-flex justify-content-end my-3">

            <button className="home-btn px-2 py-1" onClick={submit}>

              Add

            </button>

          </div>

        </div>

        <div className="todo-body">

          <div className="container-fluid">

            <div className="row ">

              {Array &&

                Array.map((item, index) => (

                  <div

                    className="col-lg-3 col-11 mx-lg-5 mx-3 my-2"

                    key={index}

                  >

                    <TodoCards

                      title={item.title}

                      body={item.body}

                      id={item.\_id}

                      delid={del}

                      display={dis}

                      updateId={index}

                      toBeUpdate={update}

                    />

                  </div>

                ))}

            </div>

          </div>

        </div>

      </div>

      <div className="todo-update " id="todo-update">

        <div className="container update">

          <Update display={dis} update={toUpdateArray} />

        </div>

      </div>

    </>

  );

};

export default Todo;

**src/App.js:**

import React, { useEffect } from "react";

import Navbar from "./components/navbar/Navbar";

import Home from "./components/home/Home";

import Footer from "./components/footer/Footer";

import About from "./components/about/About";

import { BrowserRouter as Router, Routes, Route } from "react-router-dom";

import Signup from "./components/signup/Signup";

import SignIn from "./components/signup/SignIn";

import Todo from "./components/todo/Todo";

import { useDispatch } from "react-redux";

import { authActions } from "./store";

const App = () => {

  const dispatch = useDispatch();

  useEffect(() => {

    const id = sessionStorage.getItem("id");

    if (id) {

      dispatch(authActions.login());

    }

  }, []);

  return (

    <div>

      <Router>

        <Navbar />

        <Routes>

          <Route exact path="/" element={<Home />} />

          <Route path="/about" element={<About />} />

          <Route path="/todo" element={<Todo />} />

          <Route path="/signup" element={<Signup />} />

          <Route path="/signin" element={<SignIn />} />

        </Routes>

      </Router>

      <Footer />

    </div>

  );

};

export default App;

**Step 3: Run the Application**

**1. Start the Backend Server**

In the backend directory, start the backend server:

Nodemon app.js

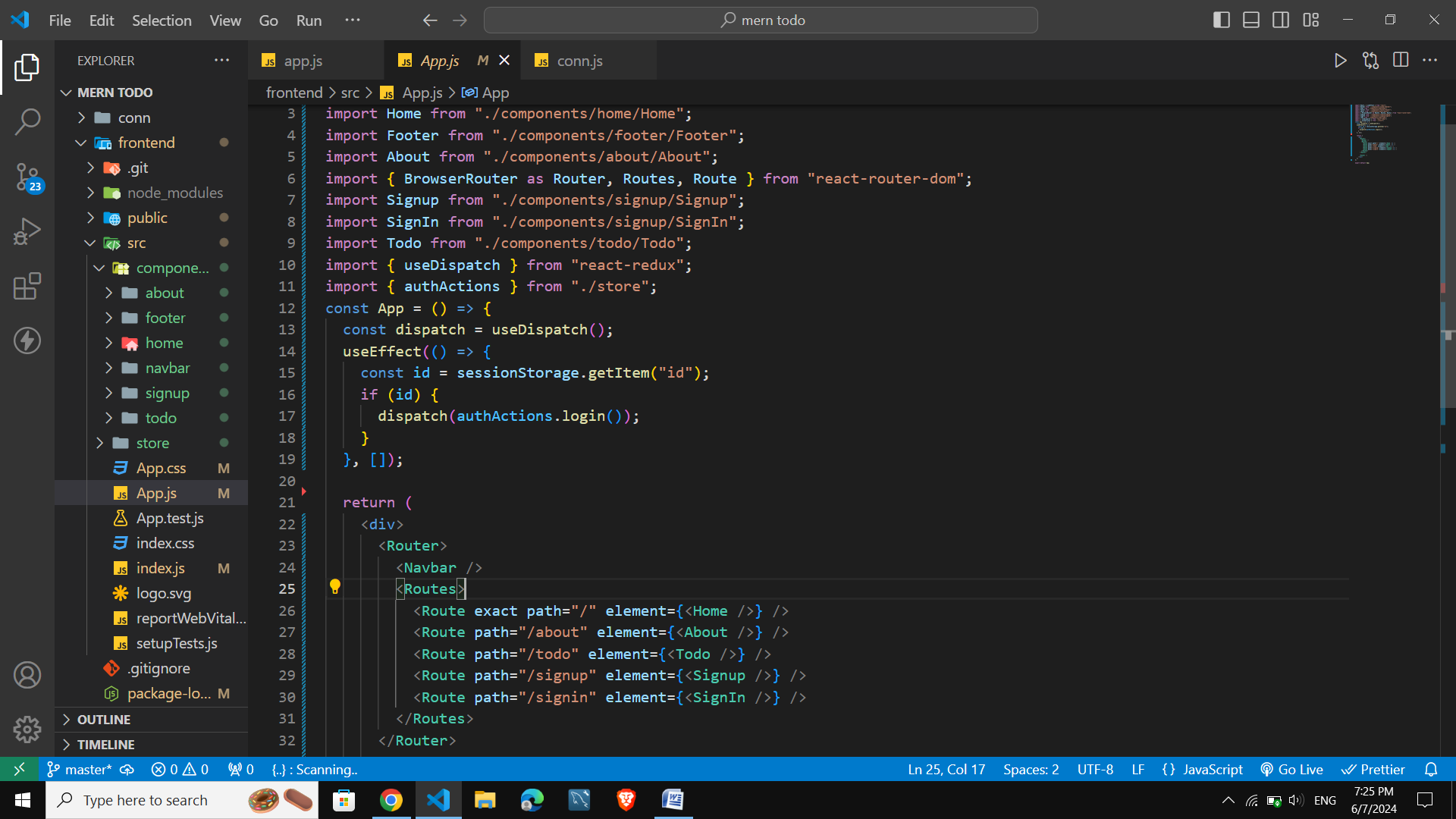
**2. Start the Frontend Server**

In the frontend directory, start the frontend development server:

npm start

**Step 4: Test the Application**

Open your browser and navigate to http://localhost:3000. You should see your Todo application. You can add, delete, and toggle todos.



Snapshots

