./server/controller/UserController.js

const User = require("../model/UserModel");

exports.newUser = async (req, res) => {

  try {

    // Get user input

    const { name, password,country } = req.body;

    const params = {};

    if (name) {

      params.name = name;

    }

    if (password) {

      params.password = password;

    }

    if (!(params?.name && params?.password && params?.country)) {

      res.status(400).send({ message: "All input is required" });

    }

    // Create user in our database

    const user = await User.create({

      name: params?.name,

      password: params?.password,

      country: params?.country,

    });

    user.save();

    // return new user

    res.status(201).json(user);

  } catch (err) {

    console.log(err);

  }

};

exports.getAllUsers = async (req, res) => {

  try {

    // Validate if user exist in our database

    const user = await User.find({ });

    if (user) {

      res.status(200).json({

        code: 2001,

        data: user,

      });

    }

  } catch (err) {

    return res.status(500).json({

      code: 2002,

      message: "Something went wrong",

    });

  }

};

./server/model/UserModel.js

const mongoose = require("mongoose");

//Object Data Modeling (ODM) library for MongoDB and Node.js.

//It manages relationships between data, provides schema validation,

//and is used to translate between objects in code and the representation

//of those objects in MongoDB.

const userSchema = new mongoose.Schema(

  {

    name: {

      type: String,

      required: true,

    },

    password: {

      type: string,

      required: true,

    },

    country: {

      type: string,

      required: true,

    },

  },

  {

    timestamps: true,

    versionKey: false,

  }

);

const User = mongoose.model("User", userSchema);

module.exports = User;

./server/routes/userRoute.js

const express = require("express");

const router = express.Router();

const userController = require("../controllers/UserController");

router.post("/new", userController.newUser);

router.get("/getAll", userController.getAllUsers);

module.exports = router;

./server/index.js

const express = require("express");

const app = express();

const cors = require("cors");

const mongoose = require("mongoose");

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

const connect = () => {

  mongoose.connect(`mongodb://127.0.0.1:27017/LabExam`, {

    useNewUrlParser: true,

    useUnifiedTopology: true,

  }).then(() => {

    console.log("Successfully connected to database");

  })

  .catch((error) => {

    console.log("database connection failed. exiting now...");

    console.error(error);

    process.exit(1);

  });

};

app.use(express.json());

app.use(cors());

app.use("/user", userRoute);

app.listen(8080, async () => {

  await connect();

  console.log(`Server is connected to port 8080`);

});

./src/pages/Form.js

import React from 'react'

import {Form, Button} from 'react-bootstrap';

import axios from 'axios';

import { useNavigate } from 'react-router-dom';

export default function FormComp() {

  const [name, setName] = React.useState("");

  const [password, setPassword] = React.useState("");

  const [country, setCountry] = React.useState("");

  const history = useNavigate();

  const submitForm = () => {

    axios.post('http://localhost:8080/user/new', {name: name, password: password, country: country}).then((res) => {

      console.log('res', res);

      alert('User created successfully')

      history('/');

    })

  }

  return (

    <div className='row container'>

    <Form className="col-md-8">

      <Form.Group className="mb-3" controlId="exampleForm.ControlInput1">

        <Form.Label>Name</Form.Label>

        <Form.Control value={name} onChange={(e) => setName(e.target.value)} type="name" placeholder="Name" />

      </Form.Group>

      <Form.Group className="mb-3" controlId="exampleForm.ControlInput1">

        <Form.Label>Password</Form.Label>

        <Form.Control value={[password]} onChange={(e) => setPassword(e.target.value)} type = "password" placeholder='\*\*\*\*\*\*\*\*' />

      </Form.Group>

      <Form.Group className="mb-3" controlId="exampleForm.ControlInput1">

        <Form.Label>Country</Form.Label>

        <Form.Control value={[country]} onChange={(e) => setCountry(e.target.value)} type="name" placeholder='Pakistan' />

      </Form.Group>

      <br />

      <br />

      <Form.Group className="mb-3" controlId="formBasicCheckbox">

        <Form.Check type="checkbox" label="I accept terms of service" />

      </Form.Group>

      <Button onClick={submitForm}>Submit</Button>

    </Form>

    </div>

  )

}

./src/App.js

import './App.css';

import React from 'react';

import {

  BrowserRouter,

  Routes,

  Route,

} from "react-router-dom";

import Form from './pages/Form';

function App() {

  return (

      <BrowserRouter>

        <Routes>

          <Route path="/" element={<Form />} />

        </Routes>

      </BrowserRouter>

  );

}

export default App;

./src/index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import 'bootstrap/dist/css/bootstrap.min.css';

const root = ReactDOM.createRoot(document.getElementById('app'));

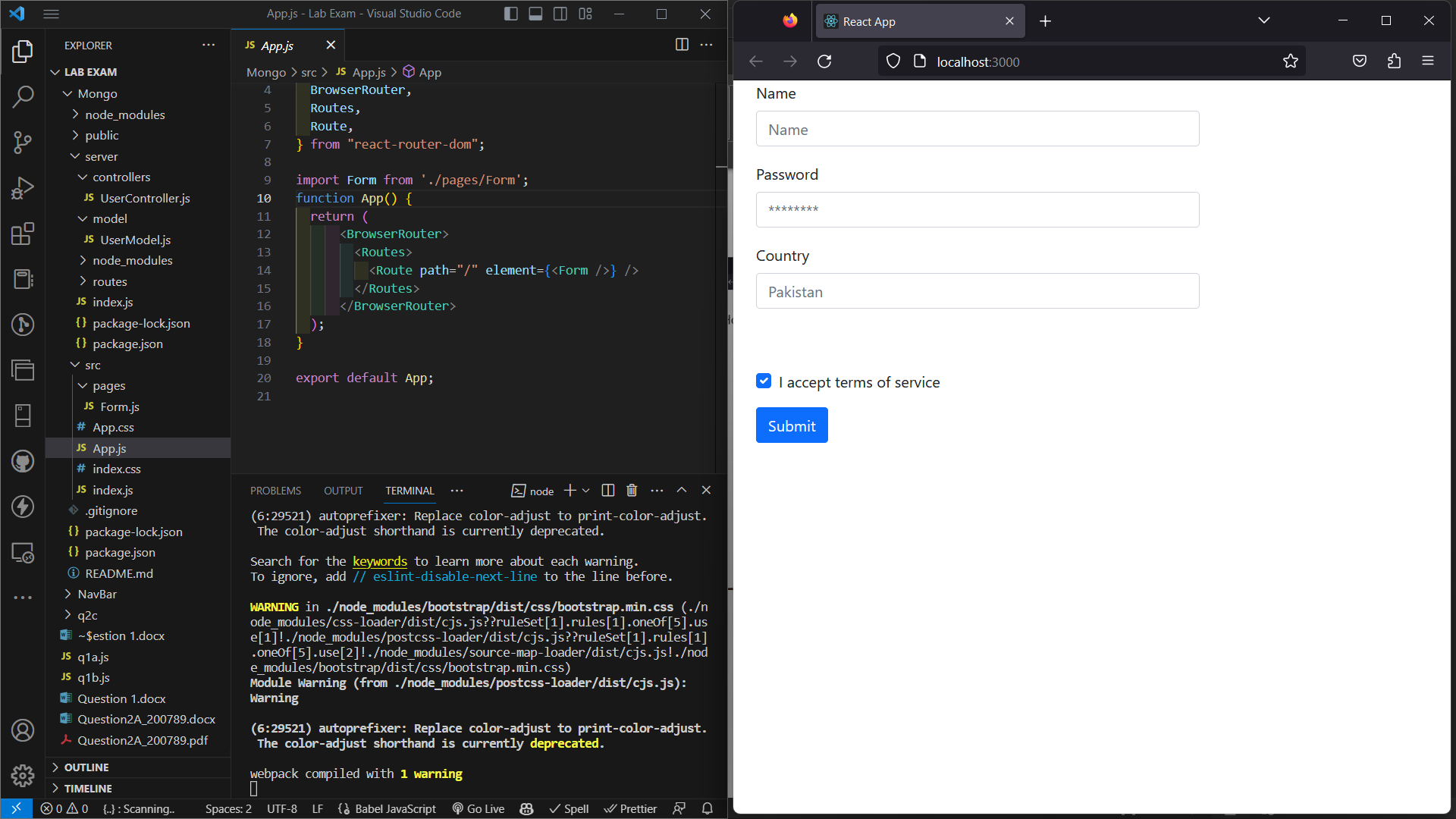
root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);



A screenshot of a computer

Description automatically generated with medium confidence