MERN Full Stack Project

1. Prerequisites

Ensure you have the following installed:

1. **Node.js**: Download from Node.js.
2. **MySQL**: Install MySQL server and create a database.
3. **Code Editor**: Use Visual Studio Code.

---

2. Backend Setup (Node.js)

Step 1: Create the Backend Directory

1. Open the terminal and run:

```
mkdir backend
cd backend
npm init -y
```

Step 2: Install Required Packages

```
npm install express mysql2 body-parser bcrypt jsonwebtoken cors
```

Step 3: Write the server.js File

Create a file named server.js in the backend folder and add this code:

```
const express = require("express");
const bodyParser = require("body-parser");
const mysql = require("mysql2");
const bcrypt = require("bcrypt");
const cors = require("cors");

const app = express();
app.use(bodyParser.json());
app.use(cors());

const db = mysql.createConnection({
 host: "localhost",
 user: "root",
 password: "", // Replace with your MySQL password
 database: "registration_db",
});

// API to Register a New User
app.post("/api/register", async (req, res) => {
 const { firstName, lastName, mobileNumber, password } = req.body;
 const hashedPassword = await bcrypt.hash(password, 10);

 db.query(
  "INSERT INTO users (first_name, last_name, mobile_number, password) VALUES (?, ?, ?, ?)",
  [firstName, lastName, mobileNumber, hashedPassword],
  (err) => {
   if (err) {
    return res.status(500).json({ error: err.message });
```

```
      }
      res.status(201).json({ message: "User registered successfully" });
    }
  );
});

// API to Login a User
app.post("/api/login", (req, res) => {
  const { mobileNumber, password } = req.body;
  db.query(
    "SELECT * FROM users WHERE mobile_number = ?",
    [mobileNumber],
    async (err, results) => {
      if (err || results.length === 0) {
        return res.status(404).json({ error: "User not found" });
      }
      const user = results[0];
      const isMatch = await bcrypt.compare(password, user.password);
      if (!isMatch) {
        return res.status(401).json({ error: "Invalid credentials" });
      }
      res.status(200).json({ message: `Welcome ${user.first_name} ${user.last_name}` });
    }
  );
});

app.listen(5000, () => console.log("Server running on port 5000"));
```

Step 4: Create the Database

1. Open MySQL and run:

```
CREATE DATABASE registration_db;

USE registration_db;

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    mobile_number VARCHAR(15) UNIQUE,
    password VARCHAR(255),
    created_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_date DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

Step 5: Run the Backend

1. Start the server:
   ```
   node server.js
   ```
   Output: Server running on port 5000.

3. Frontend Setup (React.js)

Step 1: Create the Frontend Directory

1. Open a new terminal window:

```
npx create-react-app frontend
cd frontend
```

Step 2: Install Axios for API Calls

```
npm install axios
```

Step 3: Create a Registration Component

1. Navigate to src/ and create a folder components.
2. Create a file RegistrationPage.js:

```
import React, { useState } from "react";
import axios from "axios";

const RegistrationPage = () => {
  const [formData, setFormData] = useState({
    firstName: "",
    lastName: "",
    mobileNumber: "",
    password: "",
  });

  const handleChange = (e) => {
    setFormData({ ...formData, [e.target.name]: e.target.value });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      const response = await axios.post("http://localhost:5000/api/register", formData);
      alert(response.data.message);
      window.location.href = "/login";
    } catch (err) {
      alert(err.response.data.error || "Registration failed");
    }
  };

  return (
    <div>
      <h2>Register</h2>
      <form onSubmit={handleSubmit}>
        <input name="firstName" placeholder="First Name" onChange={handleChange} required />
        <input name="lastName" placeholder="Last Name" onChange={handleChange} required />
        <input name="mobileNumber" placeholder="Mobile Number" onChange={handleChange} required />
    <input name="password" type="password" placeholder="Password" onChange={handleChange} required />
    <button type="submit">Register</button>
   </form>
  </div>
 );
};

export default RegistrationPage;
```

Step 4: Update App.js

1. Replace the content of App.js:

```
import React from "react";
import RegistrationPage from "./components/RegistrationPage";

const App = () => {
 return (
  <div>
   <RegistrationPage />
  </div>
 );
};

export default App;
```

Step 5: Run the Frontend

1. Start the React app:

```
npm start
```

4. Test the Project

Test Registration

1. Open your browser and visit http://localhost:3000.
2. Fill out the form and click **Register**.
3. Check the database to confirm the new user entry.

Test Login

1. Create a login component following the same steps.
2. Use the /api/login API to verify login functionality.

---

5. Deploy the Project

1. **Frontend**:
   - Deploy to platforms like **Netlify** or **Vercel**.
   - Example for Netlify:

     ```
     npm run build
     netlify deploy
     ```