Exe 9. Build a Registration Form in React integrated with Express & MongoDB.

1. Project Structure

```
registration-app/
│
├── backend/
│     ├── models/
│     │     └── User.js
│     ├── routes/
│     │     └── userRoutes.js
│     ├── server.js
│     └── config/
│           └── db.js
│
└── frontend/
      ├── src/
      │     ├── components/
      │     │     └── Register.js
      │     ├── App.js
      │     └── index.js
```

---

2. Backend Setup (Express + MongoDB)

Step 1: Initialize backend

```
mkdir backend
cd backend
npm init -y
npm install express mongoose cors body-parser
npm install nodemon --save-dev
```

Step 2: Create MongoDB Connection (`config/db.js`)

```
const mongoose = require('mongoose');

const connectDB = async () => {
  try {
    await mongoose.connect('mongodb://127.0.0.1:27017/registrationDB');
    console.log('MongoDB Connected');
  } catch (error) {
    console.error(error);
    process.exit(1);
  }
};

module.exports = connectDB;
```

Step 3: Create User Model (`models/User.js`)

```
const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({
  name: String,
  email: String,
  password: String
});

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

Step 4: Create Routes (`routes/userRoutes.js`)

```
const express = require('express');
const router = express.Router();
const User = require('../models/User');

// Register API
router.post('/register', async (req, res) => {
  try {
    const { name, email, password } = req.body;

    const newUser = new User({ name, email, password });
    await newUser.save();

    res.status(201).json({ message: 'User Registered Successfully' });
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
});

module.exports = router;
```

Step 5: Create Server (`server.js`)

```
const express = require('express');
const cors = require('cors');
const connectDB = require('./config/db');

const app = express();

// Middleware
app.use(cors());
app.use(express.json());

// DB Connection
connectDB();

// Routes
app.use('/api/users', require('./routes/userRoutes'));
```

```
// Server
const PORT = 5000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

Step 6: Run Backend

```
npx nodemon server.js
```

---

3. Frontend Setup (React)

Step 1: Create React App

```
npx create-react-app frontend
cd frontend
npm install axios
```

Step 2: Create Registration Form (`components/Register.js`)

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

function Register() {
  const [formData, setFormData] = useState({
    name: '',
    email: '',
    password: ''
  });

  const { name, email, password } = formData;

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

  const onSubmit = async (e) => {
    e.preventDefault();

    try {
      const res = await axios.post(
        'http://localhost:5000/api/users/register',
        formData
      );
      alert(res.data.message);
    } catch (err) {
      alert('Error registering user');
    }
  };
```

```
  return (
    <div>
      <h2>Register</h2>
      <form onSubmit={onSubmit}>
        <input
          type="text"
          name="name"
          placeholder="Enter Name"
          value={name}
          onChange={onChange}
        />
        <br /><br />

        <input
          type="email"
          name="email"
          placeholder="Enter Email"
          value={email}
          onChange={onChange}
        />
        <br /><br />

        <input
          type="password"
          name="password"
          placeholder="Enter Password"
          value={password}
          onChange={onChange}
        />
        <br /><br />

        <button type="submit">Register</button>
      </form>
    </div>
  );
}

export default Register;
```

Step 3: Update `App.js`

```
import React from 'react';
import Register from './components/Register';

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

export default App;
```

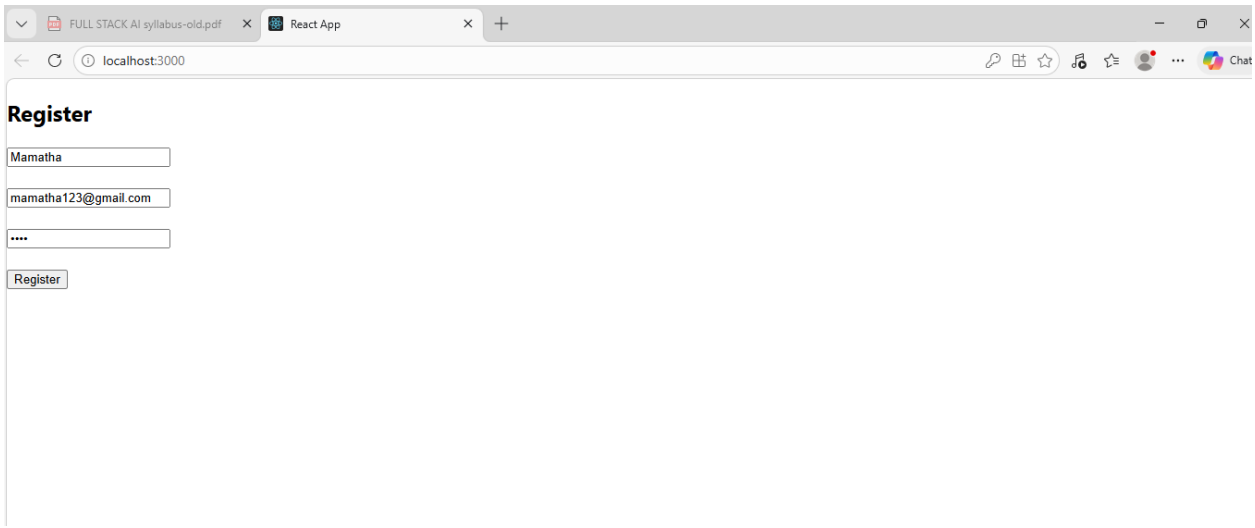
Step 4: Run Frontend

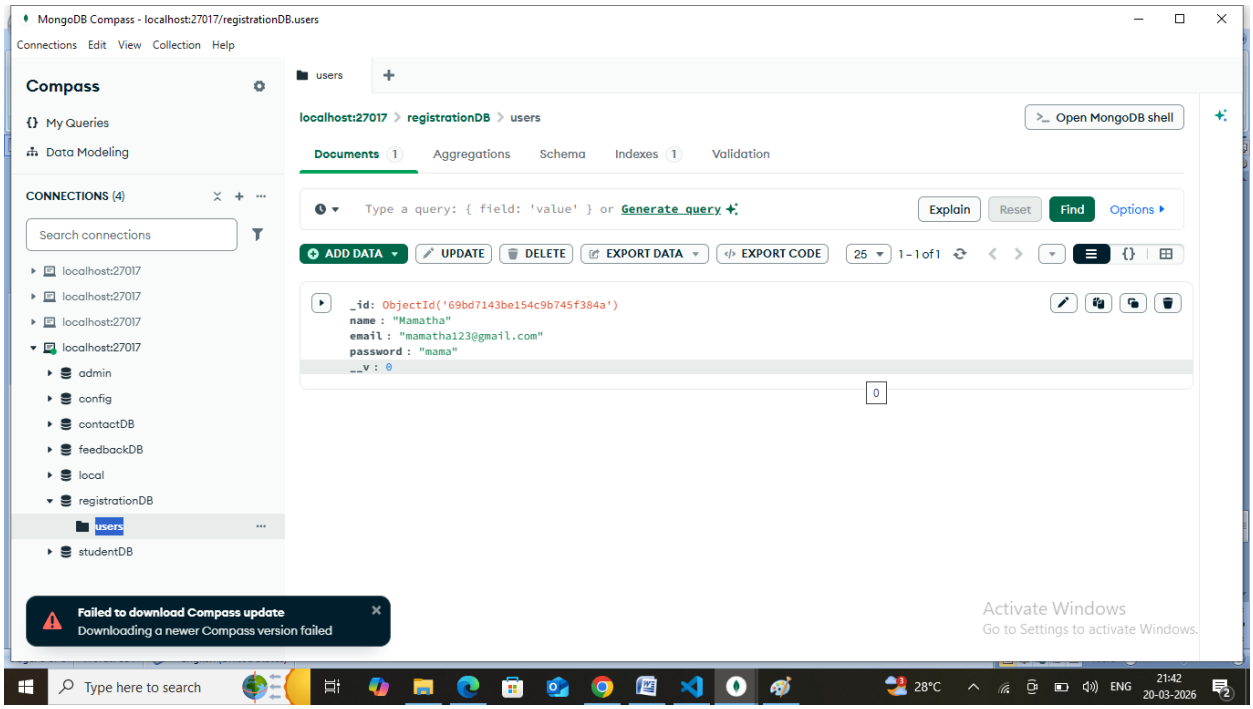
```
npm start
```

---

4. Final Flow

1. User fills form in React
2. React sends POST request using Axios
3. Express receives request
4. Data is saved in MongoDB
5. Response sent back → shown in alert




MongoDB Compass - localhost:27017/registrationDB.users
Connections Edit View Collection Help
Compass
My Queries
Data Modeling
CONNECTIONS (4)
Search connections
localhost:27017
localhost:27017
localhost:27017
localhost:27017
admin
config
contactDB
feedbackDB
local
registrationDB
users
studentDB
users
localhost:27017 > registrationDB > users
Open MongoDB shell
Documents 1
Aggregations
Schema
Indexes 1
Validation
Type a query: { field: 'value' } or Generate query
Explain
Reset
Find
Options
ADD DATA
UPDATE
DELETE
EXPORT DATA
EXPORT CODE
25
1 - 1 of 1
_id: ObjectId('69bd7143be154c9b745f384a')
name : "Mamatha"
email : "mamatha123@gmail.com"
password : "mama"
__v : 0
0
Failed to download Compass update
Downloading a newer Compass version failed
Activate Windows
Go to Settings to activate Windows.
Type here to search
28°C
ENG
21:42
20-03-2026