

## LA8 – Building a REST API with Express, Node, and MongoDB

- Rohit Arrunachalam IOT B (21110433)

Frontend Code

App.jsx

```
import React, { useState } from "react";
import axios from "axios"
const RegisterPage = () => {
  const [fullName, setFullName] = useState("");
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const handleFullNameChange = (e) => {
    setFullName(e.target.value);
  };

  const handleEmailChange = (e) => {
    setEmail(e.target.value);
  };

  const handlePasswordChange = (e) => {
    setPassword(e.target.value);
  };

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

    const userData = {
      fullName: fullName,
      email: email,
      password: password,
    };

    try {
```

```

    const response = await
axios.post("http://localhost:5000/api/register", userData, {
  headers: {
    "Content-Type": "application/json",
  },
});

    console.log("Registration Successful:",
response.data);

    setFullName("");
    setEmail("");
    setPassword("");
  } catch (error) {
    console.error("Registration Error:", error);
  }
};

return (
  <div className="w-screen h-screen flex items-center
justify-center bg-gray-50">
    <div className="max-w-md w-full p-4 space-y-4 bg-white
rounded-lg shadow-md">
      <h2 className="text-2xl font-semibold text-
center">Register</h2>
      <form onSubmit={handleSubmit}>
        <div>
          <label htmlFor="fullName" className="block text-
gray-700">
            Full Name
          </label>
          <input
            type="text"
            id="fullName"
            name="fullName"
            value={fullName}
            onChange={handleFullNameChange}
            className="w-full px-4 py-2 mt-2 text-gray-700
bg-gray-200 rounded"

```

```
        required
      />
    </div>
    <div>
      <label htmlFor="email" className="block text-
bg-gray-700">
        Email
      </label>
      <input
        type="email"
        id="email"
        name="email"
        value={email}
        onChange={handleEmailChange}
        className="w-full px-4 py-2 mt-2 text-gray-700
bg-gray-200 rounded"
        required
      />
    </div>
    <div>
      <label htmlFor="password" className="block text-
bg-gray-700">
        Password
      </label>
      <input
        type="password"
        id="password"
        name="password"
        value={password}
        onChange={handlePasswordChange}
        className="w-full px-4 py-2 mt-2 text-gray-700
bg-gray-200 rounded"
        required
      />
    </div>
    <button
      type="submit"
```

```

        className="w-full px-4 py-2 mt-4 text-white bg-
blue-500 rounded hover:bg-blue-600 focus:bg-blue-700"
      >
        Register
      </button>
    </form>
  </div>
</div>
);
};

export default RegisterPage;

```

Backend Code

server.js

```

const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors');
const MongoClient = require('mongodb').MongoClient;

const app = express();
const port = process.env.PORT || 5000;

const corsOptions = {
  origin: 'http://127.0.0.1:5173',
  methods: 'GET,HEAD,PUT,PATCH,POST,DELETE',
  credentials: true,
  optionsSuccessStatus: 204,
};

app.use(cors(corsOptions));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.static('public'));

const mongoURI = 'mongodb://localhost:27017/webtech';

```

```
const client = new MongoClient(mongoURI, { useNewUrlParser:
true, useUnifiedTopology: true });

client.connect()
  .then(() => {
    console.log('Connected to MongoDB');

app.post('/api/register', async (req, res) => {
  try {
    const { fullName, email, password } = req.body;
    console.log(req.body.fullName)
    const db = client.db('webtech');
    const usersCollection = db.collection('lab8');

    await usersCollection.insertOne({ fullName, email,
password });

    res.status(201).json({ message: 'Registration
successful!' });
  } catch (error) {
    console.error('Error during registration:', error);
    res.status(500).json({ error: 'Internal Server Error'
});
  }
});

app.get('/api/hello', (req, res) => {
  res.json({ message: 'Hello, world!' });
});

app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
).catch(err => {
  console.error('Error connecting to MongoDB:', err);
```

});

## Output Screenshots

