

MONGODB ATLAS

Package.json

{

  "name": "registration-api",

  "version": "1.0.0",

  "main": "index.js",

  "scripts": {

    "start": "nodemon index.js"

  },

  "keywords": [],

  "author": "",

  "license": "ISC",

  "description": "",

  "dependencies": {

    "body-parser": "^1.20.2",

    "cors": "^2.8.5",

    "express": "^4.19.2",

    "mongoose": "^7.8.1"

  },

  "devDependencies": {

    "nodemon": "^3.1.4"

  }

}

Index.js

const express = require('express');

const bodyParser = require('body-parser');

const mongoose = require('mongoose');

const cors = require('cors');

const app = express();

const port = 3000;

// Middleware to parse JSON data

app.use(bodyParser.json());

app.use(cors());

// MongoDB Atlas connection

const dbURI = 'mongodb+srv://root:root@temp-pro-db.6b2me.mongodb.net/temp-pro-db?retryWrites=true&w=majority';

mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology: true })

  .then(() => console.log('Connected to MongoDB Atlas'))

  .catch((err) => console.error('Error connecting to MongoDB Atlas:', err));

// Define User Schema

const userSchema = new mongoose.Schema({

  username: { type: String, required: true },

  email: { type: String, required: true },

  password: { type: String, required: true }

});

// Create User model

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

// Registration API route

app.post('/register', async (req, res) => {

  const { username, email, password } = req.body;

  if (!username || !email || !password) {

    return res.status(400).json({ message: 'Please provide all required fields' });

  }

  try {

    const newUser = new User({ username, email, password });

    const savedUser = await newUser.save();

    res.status(201).json({ message: 'User registered successfully', userId: savedUser.\_id });

  } catch (err) {

    res.status(500).json({ message: 'Database error', error: err });

  }

});

// Start server

app.listen(port, () => {

  console.log(`Server running on port ${port}`);

});

MYSQL DATABASE

Index.js

const express = require('express');

const bodyParser = require('body-parser');

const mysql = require('mysql2');

const cors = require('cors');

const app = express();

const port = 3000;

// Middleware to parse JSON data

app.use(bodyParser.json());

app.use(cors());

// MySQL Database connection

const db = mysql.createConnection({

  host: 'localhost',

  user: 'root', // Replace with your MySQL username

  password: 'root', // Replace with your MySQL password

  database: 'registration\_db' // Replace with your database name

});

db.connect((err) => {

  if (err) {

    throw err;

  }

  console.log('Connected to MySQL database');

});

// Registration API route

app.post('/register', (req, res) => {

    const { username, email, password } = req.body;

    if (!username || !email || !password) {

      return res.status(400).json({ message: 'Please provide all required fields' });

    }

    const sql = 'INSERT INTO users (username, email, password) VALUES (?, ?, ?)';

    const values = [username, email, password];

    db.query(sql, values, (err, result) => {

      if (err) {

        return res.status(500).json({ message: 'Database error', error: err });

      }

      res.status(201).json({ message: 'User registered successfully', userId: result.insertId });

    });

  });

// Start server

app.listen(port, () => {

  console.log(`Server running on port ${port}`);

});

Package.json

{

  "name": "registration-api",

  "version": "1.0.0",

  "main": "index.js",

  "scripts": {

    "start": "nodemon index.js"

  },

  "keywords": [],

  "author": "",

  "license": "ISC",

  "description": "",

  "dependencies": {

    "body-parser": "^1.20.2",

    "cors": "^2.8.5",

    "express": "^4.19.2",

    "mysql2": "^3.11.0"

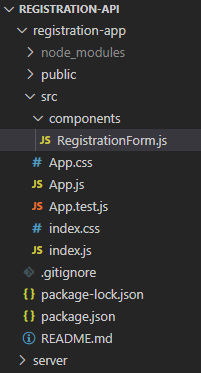
  },

  "devDependencies": {

    "nodemon": "^3.1.4"

  }

}



Package.json

{

  "name": "registration-app",

  "version": "0.1.0",

  "private": true,

  "dependencies": {

    "@testing-library/jest-dom": "^5.17.0",

    "@testing-library/react": "^13.4.0",

    "@testing-library/user-event": "^13.5.0",

    "axios": "^1.7.7",

    "react": "^18.3.1",

    "react-dom": "^18.3.1",

    "react-scripts": "5.0.1",

    "web-vitals": "^2.1.4"

  },

  "scripts": {

    "start": "react-scripts start",

    "build": "react-scripts build",

    "test": "react-scripts test",

    "eject": "react-scripts eject"

  },

  "eslintConfig": {

    "extends": [

      "react-app",

      "react-app/jest"

    ]

  },

  "browserslist": {

    "production": [

      ">0.2%",

      "not dead",

      "not op\_mini all"

    ],

    "development": [

      "last 1 chrome version",

      "last 1 firefox version",

      "last 1 safari version"

    ]

  }

}

RegistrationForm.js

import React, { useState } from 'react';

import axios from 'axios';

const RegistrationForm = () => {

  const [username, setUsername] = useState('');

  const [email, setEmail] = useState('');

  const [password, setPassword] = useState('');

  const [message, setMessage] = useState('');

  const handleSubmit = async (e) => {

    e.preventDefault();

    try {

        const response = await axios.post('http://localhost:3000/register', {

            username,

            email,

            password

          });

      setMessage(response.data.message);

    } catch (error) {

      setMessage('Registration failed. Please try again.');

    }

  };

  return (

    <div>

      <h2>Register</h2>

      <form onSubmit={handleSubmit}>

        <div>

          <label>Username:</label>

          <input

            type="text"

            value={username}

            onChange={(e) => setUsername(e.target.value)}

          />

        </div>

        <div>

          <label>Email:</label>

          <input

            type="email"

            value={email}

            onChange={(e) => setEmail(e.target.value)}

          />

        </div>

        <div>

          <label>Password:</label>

          <input

            type="password"

            value={password}

            onChange={(e) => setPassword(e.target.value)}

          />

        </div>

        <button type="submit">Register</button>

      </form>

      {message && <p>{message}</p>}

    </div>

  );

};

export default RegistrationForm;

App.js

import React from 'react';

import './App.css';

import RegistrationForm from './components/RegistrationForm';

function App() {

  return (

    <div className="App">

      <header className="App-header">

        <h1>Registration</h1>

        <RegistrationForm />

      </header>

    </div>

  );

}

export default App;