My-app

**Step 2: Set Up Node.js Backend**

1. **Initialize Node.js Project:**

mkdir server

cd server

npm init -y

npm install express mongoose cors

2.Create db.config.js:

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

// server/config/db.config.js

module.exports = {

url: 'mongodb://localhost:27017/test'

};

// server/config/db.config.js

module.exports = {

url: 'YOUR\_MONGODB\_ATLAS\_CONNECTION\_STRING'

};

3. Set Up Server:

// server/server.js

const express = require('express');

const mongoose = require('mongoose');

const cors = require('cors');

const dbConfig = require('./config/db.config');

const app = express();

app.use(cors());

app.use(express.json());

mongoose.connect(dbConfig.url, {

useNewUrlParser: true,

useUnifiedTopology: true

}).then(() => {

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

}).catch(err => {

console.log("Could not connect to MongoDB.", err);

process.exit();

});

require('./routes/user.routes')(app);

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => {

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

});

4. Create User Model:

// server/models/user.model.js

const mongoose = require('mongoose');

const UserSchema = mongoose.Schema({

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

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

password: { type: String, required: true }

}, {

timestamps: true

});

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

5. Create User Controller:

// server/controllers/user.controller.js

const User = require('../models/user.model');

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

if (!req.body.username || !req.body.email || !req.body.password) {

return res.status(400).send({ message: "All fields are required." });

}

const user = new User({

username: req.body.username,

email: req.body.email,

password: req.body.password

});

try {

const savedUser = await user.save();

res.status(201).send(savedUser);

} catch (err) {

res.status(500).send({ message: err.message || "Some error occurred while creating the User." });

}

};

6. Set Up Routes:

// server/routes/user.routes.js

module.exports = (app) => {

const users = require('../controllers/user.controller');

app.post('/api/register', users.register);

};

**Step 3: Set Up React.js Frontend**

1.Initialize React Project:

npx create-react-app client

cd client

npm install axios

2. Create Registration Component:

// client/src/components/Register.js

import React, { useState } from 'react';

import axios from 'axios';

const Register = () => {

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

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

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

const handleRegister = async (e) => {

e.preventDefault();

try {

const response = await axios.post('http://localhost:5000/api/register', {

username,

email,

password

});

console.log('User registered:', response.data);

} catch (error) {

console.error('There was an error registering the user!', error);

}

};

return (

<form onSubmit={handleRegister}>

<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>

);

};

export default Register;

3. Update App.js to Include Registration Component:

// client/src/App.js

import React from 'react';

import Register from './components/Register';

function App() {

return (

<div className="App">

<h1>Register</h1>

<Register />

</div>

);

}

export default App;

4. Run React Application:

cd client

npm start

### Running the Application

1. **Start Node.js Server:**

**cd server**

**node server.js**

2.Start React Application:

**cd client**

**npm start**