Sure, I can provide you with a step-by-step explanation along with code snippets to achieve this. We'll break down the process into several steps:

1. Setting up the React frontend with a registration form.

2. Creating an Express server to handle API requests.

3. Connecting to MongoDB using Mongoose.

4. Handling POST requests to store registration data.

Let's start:

**Step 1: Setting up the React frontend**

First, create a React application using Create React App:

npx create-react-app registration-form

cd registration-form

Now, let's create a simple registration form component.

// src/components/RegisterForm.js

import React, { useState } from 'react';

const RegisterForm = () => {

const [formData, setFormData] = useState({

username: '',

email: '',

password: ''

});

const handleChange = e => {

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

};

const handleSubmit = e => {

e.preventDefault();

// Send form data to backend

};

return (

<form onSubmit={handleSubmit}>

<input

type="text"

placeholder="Username"

name="username"

value={formData.username}

onChange={handleChange}

/>

<input

type="email"

placeholder="Email"

name="email"

value={formData.email}

onChange={handleChange}

/>

<input

type="password"

placeholder="Password"

name="password"

value={formData.password}

onChange={handleChange}

/>

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

</form>

);

};

export default RegisterForm;

```

**Step 2: Creating an Express server**

Create a new directory for the server:

```bash

mkdir server

cd server

npm init -y

npm install express mongoose body-parser

```

**Now, create an `index.js` file for your server:**

```javascript

// server/index.js

const express = require('express');

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

const mongoose = require('mongoose');

const app = express();

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

**// Middleware**

app.use(bodyParser.json());

**// MongoDB Connection**

mongoose.connect('mongodb://localhost:27017/registration', {

useNewUrlParser: true,

useUnifiedTopology: true

});

const db = mongoose.connection;

db.on('error', console.error.bind(console, 'MongoDB connection error:'));

**// Register Route**

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

// Handle registration logic here

});

**// Start server**

app.listen(PORT, () => {

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

});

``

**Step 3: Connecting to MongoDB using Mongoose**

We'll define a Mongoose schema for our user data. Create a new file `models/User.js`:

```javascript

// server/models/User.js

const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({

username: String,

email: String,

password: String

});

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

module.exports = User;

```

**Step 4: Handling POST requests to store registration data**

Now, let's handle the registration logic in the `/register` route.

```javascript

// server/index.js

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

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

try {

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

// Check if user already exists

const existingUser = await User.findOne({ email });

if (existingUser) {

return res.status(400).json({ message: 'User already exists' });

}

**// Create new user**

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

await newUser.save();

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

} catch (error) {

console.error('Registration failed:', error);

res.status(500).json({ message: 'Registration failed' });

}

});

```

**Step 5: Connecting React frontend to Express backend**

In your React component (`RegisterForm.js`), update the `handleSubmit` function to send data to the backend:

```javascript

// src/components/RegisterForm.js

import React, { useState } from 'react';

import axios from 'axios';

const RegisterForm = () => {

const [formData, setFormData] = useState({

username: '',

email: '',

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/register', formData);

console.log(response.data);

} catch (error) {

console.error('Registration failed:', error.response.data.message);

}

};

return (

<form onSubmit={handleSubmit}>

{/\* Input fields \*/}

</form>

);

};

export default RegisterForm;

```

Conclusion

You now have a complete setup for a registration form in React, with data being sent to a Node.js Express backend and stored in MongoDB using Mongoose. Remember to replace placeholder values with your actual data and adjust configurations according to your environment.

registration-app/

│

├── src/ (React Frontend)

│ ├── components/

│ │ └── RegisterForm.js

│ └── App.js

│

└── server/ (Express Backend)

├── index.js (Entry point for Express server)

├── models/ (Mongoose models)

│ └── User.js

└── package.json (Dependencies and configuration for the server)