

## 8. Recipe Sharing Platform

Creating a Recipe Sharing platform using the MEAN stack (MongoDB, Express.js, Angular, Node.js) is a great project. Here's a high-level overview of how to get started and some code snippets to guide you:

### Project Setup and Structure

Set up a new project folder and structure for your Recipe Sharing platform. Install the required Node.js packages and create a basic Angular application.

#### # Create a new Angular application

ng new recipe-sharing

#### - Backend (Node.js & Express.js)

Create the backend of your Recipe Sharing platform using Node.js and Express.js.

### Installation of Packages

Install the necessary packages for Express.js, Mongoose, and other dependencies.

```
npm install express mongoose cors
```

### Setting up Express.js

Create your Express.js server, set up middleware, and handle routes.

- javascript

// **server.js**

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

```
const app = express();
```

// **Middleware**

```
app.use(express.json());
app.use(cors());
```

// **Database connection**

```
mongoose.connect('mongodb://localhost/recipe-sharing', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  useCreateIndex: true,
});
```

## // Routes

```
app.use('/api/recipes', require('./routes/recipes'));
```

```
const port = process.env.PORT || 3000;
```

```
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

## Define Schemas and Models

Create Mongoose schemas and models for recipes. These models will define how data is structured and stored in MongoDB.

- javascript

### // models/Recipe.js

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

```
const recipeSchema = new mongoose.Schema({
```

```
  title: String,
```

```
  description: String,
```

```
  ingredients: [String],
```

```
  directions: [String],
```

```
author: String,  
// Add more fields as needed  
});  
  
module.exports = mongoose.model('Recipe', recipeSchema);
```

## Routes

Define routes for CRUD operations on recipes, including create, read, update, and delete.

### javascript

```
// routes/recipes.js  
  
const express = require('express');  
  
const router = express.Router();  
  
const Recipe = require('../models/Recipe');
```

### **// Create a new recipe**

```
router.post('/create', (req, res) => {  
  
// Validate user authentication (if required)  
  
// Save the recipe to the database  
  
// Return success response  
});
```

// Create similar routes for other operations (read, update, delete)

- **Frontend (Angular)**

Create the frontend of your Recipe Sharing platform using Angular. Design the user interface and implement recipe sharing and search features.

### **Design and UI**

Design the user interface for your Recipe Sharing platform using Angular components, templates, and styles.

### **Recipe Sharing**

Create components and forms for users to share their recipes, including title, description, ingredients, and directions.

- **typescript**

// recipe-share.component.ts

```
import { Component } from '@angular/core';
import { RecipeService } from './recipe.service';
```

```
@Component({
```

```
        selector: 'app-recipe-share',
        templateUrl: './recipe-share.component.html',
    })

export class RecipeShareComponent {
    title: string;
    description: string;
    ingredients: string[];
    directions: string[];

    constructor(private recipeService: RecipeService) {}

    shareRecipe() {
        this.recipeService.shareRecipe(this.title, this.description,
            this.ingredients, this.directions);
    }
}
```

## Recipe Search

Implement components for recipe search functionality, allowing users to search and filter recipes.

## **MongoDB**

Create a MongoDB database to store recipe data.

## **Authentication (Optional)**

Consider implementing user authentication if you want to track and manage user-contributed recipes.

## **Putting It All Together**

Integrate the frontend and backend by making API requests from Angular components to Node.js routes. Ensure that you handle recipe sharing and searching properly.

Building a Recipe Sharing platform is a rewarding project. As you progress, you can consider adding features like user accounts, ratings, comments, and image uploads to enhance the platform's functionality and user experience.