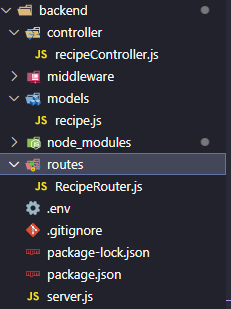
EAD final exam

Submitted by Abdul Rehman BSCS VIII ‘B’ 023-19-0064

Backend

Folder structure



Controller

Code: const Recipe = require('../models/recipe');

// Create a new recipe

*exports*.createRecipe = async (*req*, *res*) => {

  try{

    const {title,ingredients,instructions}=*req*.body;

    Recipe.create({ title,ingredients,instructions })

      .then((*data*) => {

        console.log("Saved Successfully...");

*res*.status(201).send(*data*);

      })

      .catch((*err*) => {

        console.log(*err*);

*res*.send({ error: *err*, msg: "Something went wrong!" });

      });

  }

  catch(err){}

};

// Get all recipes

*exports*.getAllRecipes = async (*req*, *res*) => {

  try {

    const recipes = await Recipe.find();

*res*.json(recipes);

  } catch (error) {

    console.error('Error fetching recipes:', error);

*res*.status(500).json({ error: 'Failed to fetch recipes' });

  }

};

// Get a specific recipe by ID

*exports*.getRecipeById = async (*req*, *res*) => {

  try {

    const recipe = await Recipe.findById(*req*.params.id);

    if (!recipe) {

      return *res*.status(404).json({ error: 'Recipe not found' });

    }

*res*.json(recipe);

  } catch (error) {

    console.error('Error fetching recipe:', error);

*res*.status(500).json({ error: 'Failed to fetch recipe' });

  }

};

// Update a recipe

*exports*.updateRecipe = async (*req*, *res*) => {

  try {

    const { title, ingredients, instructions } = *req*.body;

    const recipe = await Recipe.findByIdAndUpdate(*req*.params.id, {

      title,

      ingredients,

      instructions,

    });

    if (!recipe) {

      return *res*.status(404).json({ error: 'Recipe not found' });

    }

*res*.json(recipe);

  } catch (error) {

    console.error('Error updating recipe:', error);

*res*.status(500).json({ error: 'Failed to update recipe' });

  }

};

// Delete a recipe

*exports*.deleteRecipe = async (*req*, *res*) => {

  try {

    const recipe = await Recipe.findByIdAndDelete(*req*.params.id);

    if (!recipe) {

      return *res*.status(404).json({ error: 'Recipe not found' });

    }

*res*.json({ message: 'Recipe deleted successfully' });

  } catch (error) {

    console.error('Error deleting recipe:', error);

*res*.status(500).json({ error: 'Failed to delete recipe' });

  }

};

Model-code: const *mongoose* = require('mongoose');

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

  title: {

    type: *String*,

    required: true,

  },

  ingredients: {

    type: *String*,

    required: true,

  },

  instructions: {

    type: *String*,

    required: true,

  },

});

const Recipe = *mongoose*.model('Recipe', recipeSchema);

Routes-code: const express = require('express');

const router = express.Router();

const recipeController = require('../controller/recipeController');

// Create a new recipe

router.post('/', recipeController.createRecipe);

// Get all recipes

router.get('/', recipeController.getAllRecipes);

// Get a specific recipe

router.get('/:id', recipeController.getRecipeById);

// Update a recipe

router.put('/:id', recipeController.updateRecipe);

// Delete a recipe

router.delete('/:id', recipeController.deleteRecipe);

*module*.*exports* = router;

Server.js-code: const express = require('express');

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

const cors=require('cors')

const app=express()

const PORT=process.env.PORT | 5000

const routes = require('./routes/RecipeRouter')

require("dotenv").config()

app.use(express.json())

app.use(cors())

*mongoose*.connect('mongodb://127.0.0.1:27017',{dbName:'appdb'}).then(function() {

    console.log('Connected to database');

}).catch(function(*err*) {

    console.log(JSON.stringify(*err*));

});

// Middleware

app.use(express.json());

// Routes

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

// Error handling middleware

app.use((*err*, *req*, *res*, *next*) => {

  console.error('Error:', *err*);

*res*.status(500).json({ error: 'Something went wrong' });

});

// Start the server

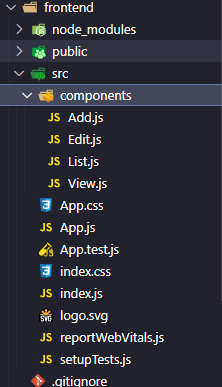
app.listen(PORT, () => {

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

});

Frontend

Folder structure



Components

Add

Look: A screenshot of a recipe

Description automatically generated with medium confidence

Code: import React, { useState } from 'react';

import axios from 'axios';

const Add = () => {

  const [title, setTitle] = useState('');

  const [ingredients, setIngredients] = useState('');

  const [instructions, setInstructions] = useState('');

  const handleSubmit = async (*e*) => {

*e*.preventDefault();

    try {

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

        title,

        ingredients,

        instructions,

      });

      console.log('Recipe added:', response.data);

      // Clear the form after successful submission

      setTitle('');

      setIngredients('');

      setInstructions('');

    } catch (error) {

      console.error('Error adding recipe:', error);

    }

  };

  return (

    <div>

      <h2>Add Recipe</h2>

      <form *onSubmit*={handleSubmit}>

        <label>Title:</label>

        <input *type*="text" *value*={title} *onChange*={(*e*) => setTitle(*e*.target.value)} *required* /><br />

        <label>Ingredients:</label>

        <textarea *value*={ingredients} *onChange*={(*e*) => setIngredients(*e*.target.value)} *required* /><br />

        <label>Instructions:</label>

        <textarea *value*={instructions} *onChange*={(*e*) => setInstructions(*e*.target.value)} *required* /><br />

        <button *type*="submit">Add Recipe</button>

      </form>

    </div>

  );

};

export default Add;

Edit: A screenshot of a recipe

Description automatically generated with medium confidence

Code: import React, { useEffect, useState } from 'react';

import axios from 'axios';

const Edit = ({ *recipeId* }) => {

  const [title, setTitle] = useState('');

  const [ingredients, setIngredients] = useState('');

  const [instructions, setInstructions] = useState('');

  useEffect(() => {

    const fetchRecipe = async () => {

      try {

        const response = await axios.get(`/api/recipes/${*recipeId*}`);

        const { title, ingredients, instructions } = response.data;

        setTitle(title);

        setIngredients(ingredients);

        setInstructions(instructions);

      } catch (error) {

        console.error('Error fetching recipe:', error);

      }

    };

    fetchRecipe();

  }, [*recipeId*]);

  const handleSubmit = async (*e*) => {

*e*.preventDefault();

    try {

      const response = await axios.put(`/api/recipes/${*recipeId*}`, {

        title,

        ingredients,

        instructions,

      });

      console.log('Recipe updated:', response.data);

    } catch (error) {

      console.error('Error updating recipe:', error);

    }

  };

  return (

    <div>

      <h2>Edit Recipe</h2>

      <form *onSubmit*={handleSubmit}>

        <label>Title:</label>

        <input *type*="text" *value*={title} *onChange*={(*e*) => setTitle(e.target.value)} *required* /><br />

        <label>Ingredients:</label>

        <textarea *value*={ingredients} *onChange*={(*e*) => setIngredients(e.target.value)} *required* /><br/>

        <label>Instructions:</label>

        <textarea *value*={instructions} *onChange*={(*e*) => setInstructions(e.target.value)} *required* /><br />

        <button *type*="submit">Update Recipe</button>

      </form>

    </div>

  );

};

export default Edit;

View: A white background with black text

Description automatically generated with low confidence

import React, { useEffect, useState } from 'react';

import axios from 'axios';

const View = ({ *recipeId* }) => {

  const [recipe, setRecipe] = useState(null);

  useEffect(() => {

    const fetchRecipe = async () => {

      try {

        const response = await axios.get(`/api/recipes/${*recipeId*}`);

        setRecipe(response.data);

      } catch (error) {

        console.error('Error fetching recipe:', error);

      }

    };

    fetchRecipe();

  }, [*recipeId*]);

  if (!recipe) {

    return <div>Loading...</div>;

  }

  return (

    <div>

      <h2>{recipe.title}</h2>

      <h3>Ingredients:</h3>

      <p>{recipe.ingredients}</p>

      <h3>Instructions:</h3>

      <p>{recipe.instructions}</p>

    </div>

  );

};

export default View;

List App.js

Code import React from 'react';

import { *BrowserRouter* as Router, Routes, Route } from 'react-router-dom';

import Add from './components/Add';

import View from './components/View';

import Edit from './components/Edit';

import List from './components/List';

const App = () => {

  return (

    <*Router*>

      <div>

        <*Routes*>

          <*Route* *exact* *path*="/" *component*={List} />

          <*Route* *exact* *path*="/recipes/add" *component*={Add} />

          <*Route* *exact* *path*="/recipes/:id" *component*={View} />

          <*Route* *exact* *path*="/recipes/:id/edit" *component*={Edit} />

        </*Routes*>

      </div>

    </*Router*>

    // <div>

    //   <h1>Recipe Management system</h1>

    //   {/\* <List/> \*/}

    //   {/\* <Add/> \*/}

    //   {/\* <Edit/> \*/}

    //   <View/>

    // </div>

    );

};

export default App;