Date:09-05-2024 IP CAT1 & 2 Assignment Vaishal Raj
3122215001119

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
Computer Science and Engineering

CAT 1 & CAT2 – Assignments - Regulations – R2021

Course Code	**UCS1601**	Course Name	Internet Programming				
Course Type	**Theory**	Course Category	Professional Core (PC)	**L**	**T**	**P**	**C**
				3	**0**	**0**	**3**
Regulation	**R-2021**		Academic Year	**2023-24 (Even)**			
Degree and Branch	**B.E. Computer Science & Engineering**		Batch	**2021-25**			
Semester		**VI**	Faculty Name	**Mr. S. Raghavendra Kumar** **Dr. B. Prabavathy**			
Department Offering the Course			**Computer Science and Engineering**				

Deadline: 09.05.2024

Design of Recipe Finder Full Stack Web Application

Problem Description

Design a full stack Recipe Finder web application which allows users to search for recipes based on the name of the recipe. The app can fetch data from various recipe APIs, display recipes, and provide additional details such as cooking instructions, nutritional information, and user reviews. Prepare a report containing the design, code, output snapshots, best practices used and learning outcomes.
[CO1, CO2, CO4, CO5, K5, 1.4.1, 2.1.2, 2.2.3, 3.2.1, 10.1.2, 13.3.1]

Here are some key features to consider:

- **Recipe Search:** Incorporate a search feature that empowers users to discover recipes by entering keyword.
- **Filtering and Sorting:** Provide options to filter and sort recipes based on criteria such as cooking time, difficulty level, and cuisine
- **Recipe Details:** Display detailed information about each recipe, including ingredients and step-by-step instructions along with the image.
- **Saved Recipes:** Allow users to save their favorite recipes for future reference. Provide a personal recipe collection where users can manage and organize their saved recipes

Design the following:

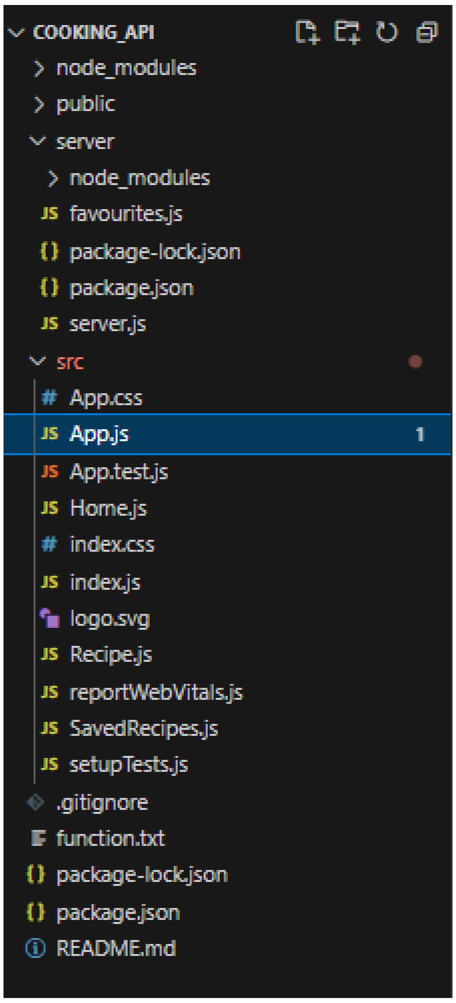
- Most appealing layout template
- Schema of the necessary MongoDB collections
- Necessary endpoints, controllers, collections, and components as a sequence diagram

Do the following operations:

- Sending appropriate GET http requests from front end, to the endpoint in the node server.

Recipe Finder Application using React, React hook, MongoDb, Axios and Express

Project Structure:



```
COOKING_API
  node_modules
  public
  server
    node_modules
    favourites.js
    package-lock.json
    package.json
    server.js
  src
    App.css
    App.js                1
    App.test.js
    Home.js
    index.css
    index.js
    logo.svg
    Recipe.js
    reportWebVitals.js
    SavedRecipes.js
    setupTests.js
  .gitignore
  function.txt
  package-lock.json
  package.json
  README.md
```

CLIENT

App.js

```
import './App.css';
import { useState, useEffect } from "react";
import axios from "axios";
import { BrowserRouter, NavLink, useNavigate } from "react-router-dom";

function App() {
  const [foodlist, setFoodList] = useState([]);
  const [filtered, setFiltered] = useState([]);
  const [currfood, setCurrFood] = useState("");
  const [savedRecipes, setSavedRecipes] = useState([]);

  const navigate = useNavigate();

  useEffect(() => {
    axios.get("http://localhost:3001/getFood")
      .then((response) => {
        setFoodList(response.data);
        setFiltered(response.data);
      })
      .catch((err) => console.log(err));

      axios.get("http://localhost:3001/getSavedRecipes")
      .then((response) => {
        setSavedRecipes(response.data);
      })
      .catch((err) => console.log(err));

  }, []);


  useEffect(() => {
    filterFood(currfood);
  }, [currfood]);

  const filterFood = (searchTerm) => {
    const filteredFood = foodlist.filter(food =>
      food.recipeName.toLowerCase().includes(searchTerm.toLowerCase()) ||
      food.recipeType.toLowerCase().includes(searchTerm.toLowerCase()) ||
      food.recipeArea.toLowerCase().includes(searchTerm.toLowerCase())
    );
    setFiltered(filteredFood);
  };

  const sortRecipes = (letter) => {
    const sortedFood = filtered ===[] ?
      filtered.filter(food =>
        food.recipeName.toLowerCase().startsWith(letter.toLowerCase())
      ) :
      foodlist.filter(food =>
        food.recipeName.toLowerCase().startsWith(letter.toLowerCase())
      );
    console.log(sortedFood);
    setFiltered(sortedFood);
  };


  const alphabets=[]
  for(let i=65;i<91;i++){
    alphabets.push(String.fromCharCode(i));
  }
  console.log(alphabets);
  const saveRecipe = (id) => {
    const recipeToSave = foodlist.find((food) => food._id === id);
    setSavedRecipes([...savedRecipes, recipeToSave]);

    alert("Recipe added to favourites");
    console.log(typeof(recipeToSave));

    fetch("http://localhost:3001/addSavedRecipe", {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(recipeToSave)
    })
    .then((res) => {
      if (!res.ok) throw new Error("Unable to add");
      return res.json();
    })
    .catch(err => console.log(err));
  };
  console.log("saved food items :",savedRecipes);

  return (
    <div className="App">
      <h1>Cooking API</h1>
      <input type='text' onChange={(e) => setCurrFood(e.target.value)}
placeholder='Search by name ,type or area name :' />

      <NavLink to={"/saved-recipes"}>
          <button style={{transform:"scale(1.2"}}>Show ⭐</button>
      </NavLink>

      <div id="display-food">
        {filtered.map((food) => (
          <div className="food-items" key={food._id}>
```

```
            <img src={food.thumbnail} height={150} width={150} alt="Food
Thumbnail" />
            <h3>{food.recipeName}</h3>
            <p>Type: {food.recipeType}</p>
            <p>Area: {food.recipeArea}</p>
            <NavLink to={`/recipe/${food._id}`}>
              <button>Get Recipe</button>
            </NavLink>
            <button onClick={()=>{saveRecipe(food._id)}}>⭑</button>
          </div>
        ))}
      </div>
      <div className='footer'>
        {
          alphabets.map((alpha,i)=>{
            return <button key={i}
onClick={()=>{sortRecipes(alpha)}}>{alpha}</button>
          })
        }
      </div>
    </div>
  );
}

export default App;
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import { BrowserRouter, Route, Routes } from 'react-router-dom';
import App from './App';
import Recipe from './Recipe';
import SavedRecipes from './SavedRecipes';

ReactDOM.render(
  <BrowserRouter>
    <Routes>
      <Route path="/" element={<App />} />
      <Route path="/recipe/:id" element={<Recipe />} />
      <Route path="/saved-recipes" element={<SavedRecipes />} />
    </Routes>
  </BrowserRouter>,
  document.getElementById('root')
);
```

Recipe.js

```
import React, { useEffect, useState } from 'react';
import { useParams} from 'react-router-dom'; // Import useParams hook to
access URL parameters
import axios from "axios";
//import { Link } from 'react-router-dom';
function Recipe(){
    const [recipe, setRecipe] = useState(null);
    const { id } = useParams();

    useEffect(() => {
        axios.get(`http://localhost:3001/showRecipe/${id}`).then(
            (response) => {
                setRecipe(response.data);
                console.log(recipe)
            },
        ).catch(
            (error) => {
                console.log(error);
            }
        )


    }, [id]);

    return (
        <div className="recipe" style={{textAlign:'center'}}>
            {/* <div className="navbar">
                <ul style={{listStyle:"none"}}>
                <li><Link to="#ingredients">Ingredients</Link></li>
                <li><Link to="#instructions">Instructions</Link></li>
                </ul>
            </div> */}
            <h2>Recipe Details</h2>
            {recipe ? (
                <div className='recipe_item'>
                    <h3>{recipe.recipeName}</h3>
                    <p>Recipe Type: {recipe.recipeType}</p>
                    <p>Recipe Area: {recipe.recipeArea}</p>
                    <div id='ingredients'>
                        <h4>Ingredients</h4>
                        <ul>
                            {recipe.ingredients.map((ingredient, index) => (
                            <li key={index}>{ingredient}</li>
                            ))}
                        </ul>
                    </div>
                    <div id='instructions'>
                        <h4>Instructions</h4>
                        <ol>
                            {recipe.instructions.map((instruction, index) => (
                            <li key={index}>{instruction}</li>
                            ))}
                        </ol>
                    </div>
                    <img src={recipe.thumbnail} alt="Food Thumbnail" />
                </div>
            ) : (
                <p>Loading...</p>
            )}
        </div>
    );
};

export default Recipe;
```

App.css

```
body {
  background-color: lightblue;
}

.App {
  text-align: center;
  text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.5);
  text-transform: capitalize;
}
h1{
  font-size:100px;
  color: #fff;
  text-shadow: 2px 5px 5px #000000;
  font-family: cursive;
  transition: transform 1s ease;
  animation: rotateheading 4s infinite;
}

@keyframes rotateheading {
  0% {
    transform:  translateX(0);
  }
  25% {
    transform: translateX(50px);
  }
  50% {
    transform: translateX(-50px);
  }
  75% {
    transform: translateX(50px);
  }
  100%{
    transform: translateX(0);
  }
}

.food-items {
  border: 1px solid #ddd;
  margin: 10px;
  padding: 10px;
  width: 220px;
  background-color: #fff;
  border-radius: 8px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  transition: transform 1s ease;
}

.food-items:hover {
  transform: translateY(-5px);
}

input[type="text"] {
  width: 70%;
  padding: 8px;
  border: 1px solid #ddd;
  border-radius: 4px;
  margin-bottom: 20px;
  margin-right: 50px;
}

.food-items img {
  display: block;
  margin: 0 auto;
  margin-bottom: 10px;
  border-radius: 4px;
}
p{
  font-size: 20px;
}
.food-items img:hover{
  animation: rotateImg 2s infinite;
}

.food-items h3 {
```

```css
  font-size: 20px;
  margin-bottom: 5px;
}

#display-food {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
}

.recipe {
  background-color: #f9f9f9;
  padding: 20px;
  font-family:Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
  font-size: 2em;
  animation: colorChange 2s infinite;
}

.recipe_item {
  text-align: left;
}

.recipe h2 {
  font-size: 24px;
  margin-bottom: 20px;
  font-weight: bold;
  text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.5);
  text-transform: uppercase;
  text-decoration: underline;
  font-size: 2em;

}

.recipe p {
  font-size: 30px;
  margin-bottom: 10px;
}

.recipe h4 {
  font-size: 30px;
  margin-bottom: 10px;
}

.recipe img {
  display: block;
  margin: auto;
  margin-top: 20px;
  border-radius: 4px;
  height: 500px;
  width: 500px;
```

```
}


.navbar{
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 20px;
}

button{
  margin-right: 10px;
  cursor: pointer;
}

button:hover{
  background-color: lightcyan;
}


@keyframes colorChange {
  0%{
    background-color: lightcoral;
  }
  50%{
    background-color: lightcyan;
  }
  100%{
    background-color: lightskyblue;
  }
}

@keyframes rotateImg{
  0%{
    transform: rotate(0deg);}
  50%{
    transform: rotate(-45deg);
  }
  75%{
    transform: rotate(45deg);
  }
  100%{
    transform: rotate(0deg);
  }
}
```

SavedRecipes.js

```
import React from 'react';
import './App.css';
import { useState, useEffect } from "react";
import axios from "axios";
import { NavLink, useNavigate } from "react-router-dom";


function SavedRecipes() {
    const pageReload=()=>{
        window.location.reload();
    }
    const [savedRecipes, setSavedRecipes] = useState([]);

    useEffect(() => {

        axios.get("http://localhost:3001/getSavedRecipes")
        .then((response) => {
            setSavedRecipes(response.data);
        })
        .catch((err) => console.log(err));
        console.log("reloaded");

    }, []);

    const removeSavedFood=(id)=>{
        const obj={_id:id};
        console.log(id);
        console.log(obj);
        fetch("http://localhost:3001/removeSavedRecipes",{method:'PUT',headers
:{'Content-Type':'application/json'},body:JSON.stringify(obj)})
        .then((result)=>{
            if(!result.ok) throw new Error("Unable to delete")
                return result.json();
        }).catch(err=>{
            console.log(err);
        })
        pageReload();
    }
    return (
        <div className='App'>
        <h2>Saved Recipes</h2>
        <div id="display-food">
            {savedRecipes.map((food) => (
            <div className="food-items" key={food._id}>
                <img src={food.thumbnail} height={150} width={150} alt="Food
Thumbnail" />
                <h3>{food.recipeName}</h3>
                <p>Type: {food.recipeType}</p>
                <p>Area: {food.recipeArea}</p>
                <NavLink to={`/recipe/${food._id}`}>
```

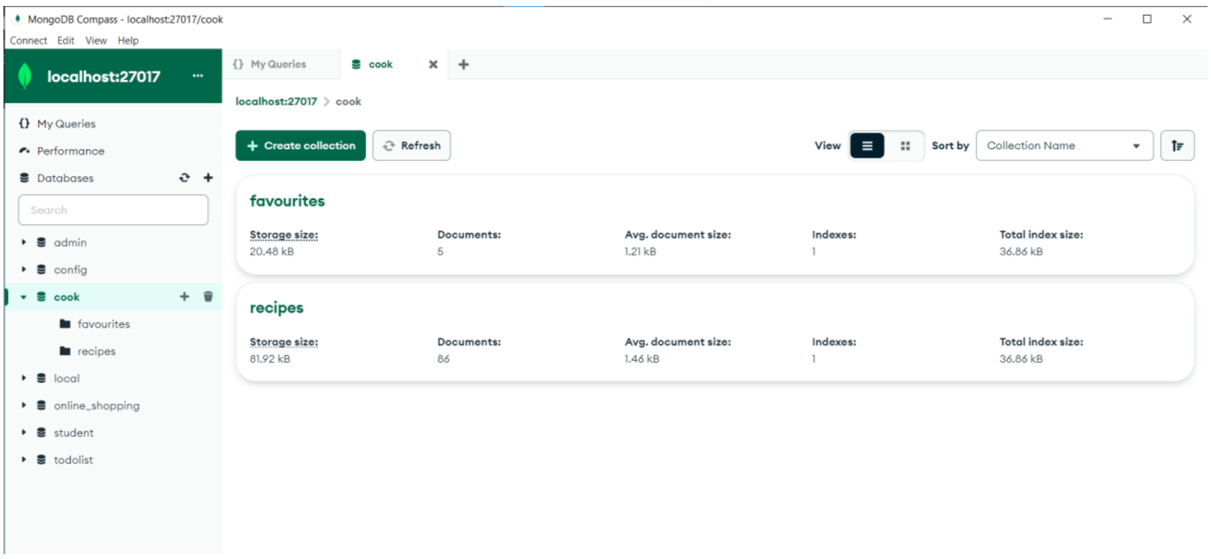
```
                    <button>Get Recipe</button>
                </NavLink>
                <button onClick={()=>{removeSavedFood(food._id)}}>Remove
⭐</button>
            </div>
            ))}
        </div>
    </div>
    );
}

export default SavedRecipes;
```

SERVER

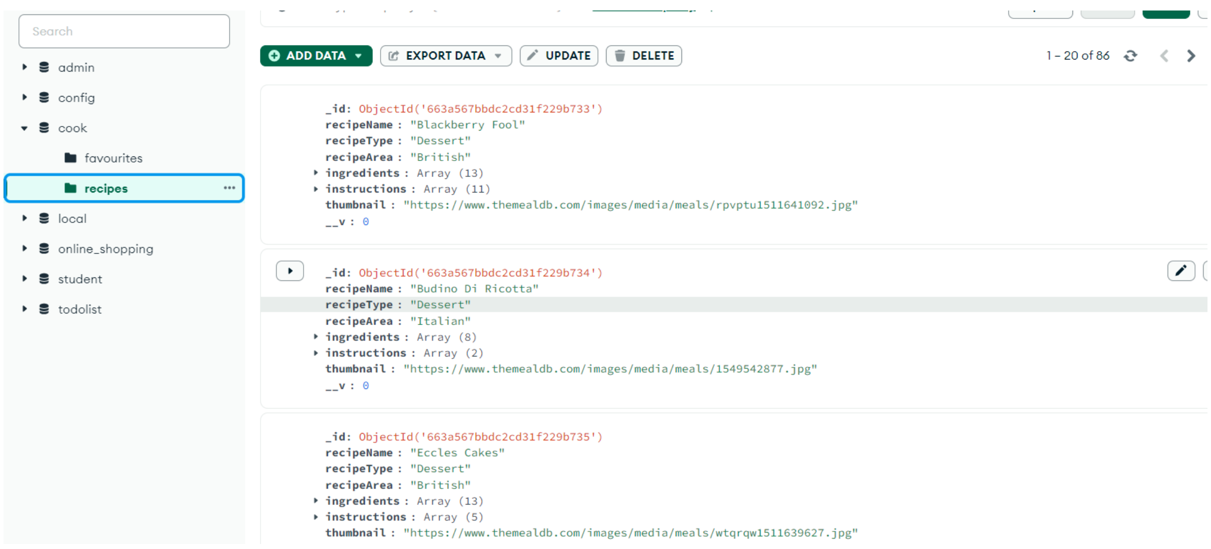
Server.js

```
const express=require("express");
const mongoose=require("mongoose");
const cors=require("cors");
const app=express();
app.use(express.json());
app.use(cors());
mongoose.connect("mongodb://localhost:27017/cook");

mongoose.connection.on("connected", () => {
    console.log("Connected to MongoDB");
});


const UserSchema=new mongoose.Schema({
    recipeName:{
        type:String,
        required:true
    },
    recipeType:{
        type:String,
        required:true
    },
    recipeArea:{
        type:String,
        required:true
    },
    ingredients:{
        type:[String],
        required:true
    },
    instructions:{
        type:[String],
        required:true
    },
    thumbnail:{
        type:String,
        required:true
    },
    __v:{
        type:Number,
        default:0
    }
})

const UserModel=mongoose.model("recipes",UserSchema);
const SavedFoodModel=mongoose.model("favourites",UserSchema); //for favourites

app.get("/getFood", (req, res) => {
    UserModel.find({}).then((food) => {
        res.json(food);
    }).catch(err => {
        console.error(err);
        res.json(err);
    });
});

app.get("/showRecipe/:id",(req,res)=>{
    const {id}=req.params;
    UserModel.findById({_id:id}).then(
        (recipe)=>{
            res.json(recipe);
        }
    ).catch(
        (err)=>{
            console.log(err);
            res.json(err);
        }
    )
})


//for saved recipes------------------------------------------------------------
---------------

app.get("/getSavedRecipes", (req, res) => {
    SavedFoodModel.find({}).then((food) => {
        res.json(food);
    }).catch(err => {
        console.error(err);
        res.json(err);
    });
});


app.post("/addSavedRecipe", (req, res) => {
    const item = req.body;
    console.log("server side saved -------------------------------------------
----------------:",item);
    const newSavedFood = new SavedFoodModel(item);
    newSavedFood.save()
        .then(result => {
            console.log("saved result :", result);
            return res.json(result);
        })
        .catch(err => {
            console.error("Error saving recipe:", err);
            return res.status(500).json({ error: "Unable to save recipe" });
        });
});

app.put("/removeSavedRecipes", async (req, res) => {
    console.log(req.body);
    console.log(typeof(req.body));
    const { _id } = req.body;
    const result = await SavedFoodModel.deleteOne({ _id: _id });
    if (!result) {
        return res.json({error:"Item not found!!"})
    }
    return res.json({message : "Item deleted successfully!!"});

});

app.listen(3001,()=>{
    console.log("server is running on port 3001");
})
```

MONGODB



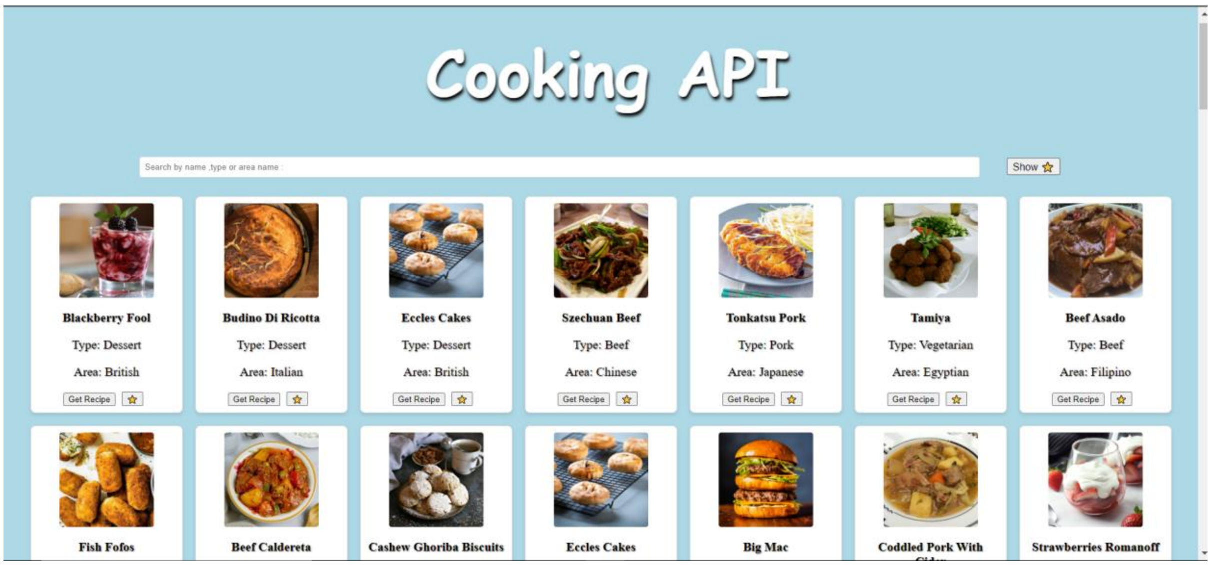
	Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
favourites	20.48 kB	5	1.21 kB	1	36.86 kB
recipes	81.92 kB	86	1.46 kB	1	36.86 kB



```
_id: ObjectId('663a567bbdc2cd31f229b733')
recipeName : "Blackberry Fool"
recipeType : "Dessert"
recipeArea : "British"
ingredients : Array (13)
instructions : Array (11)
thumbnail : "https://www.themealdb.com/images/media/meals/rpvptu1511641092.jpg"
__v : 0

_id: ObjectId('663a567bbdc2cd31f229b734')
recipeName : "Budino Di Ricotta"
recipeType : "Dessert"
recipeArea : "Italian"
ingredients : Array (8)
instructions : Array (2)
thumbnail : "https://www.themealdb.com/images/media/meals/1549542877.jpg"
__v : 0

_id: ObjectId('663a567bbdc2cd31f229b735')
recipeName : "Eccles Cakes"
recipeType : "Dessert"
recipeArea : "British"
ingredients : Array (13)
instructions : Array (5)
thumbnail : "https://www.themealdb.com/images/media/meals/wtqrqw1511639627.jpg"
```

Date:09-05-2024 IP CAT1 & 2 Assignment Vaishal Raj 3122215001119

OUTPUT:-



Blackberry Fool
Type: Dessert
Area: British

Budino Di Ricotta
Type: Dessert
Area: Italian

Eccles Cakes
Type: Dessert
Area: British

Szechuan Beef
Type: Beef
Area: Chinese

Tonkatsu Pork
Type: Pork
Area: Japanese

Tamiya
Type: Vegetarian
Area: Egyptian

Beef Asado
Type: Beef
Area: Filipino

Fish Fofos

Beef Caldereta

Cashew Ghoriba Biscuits

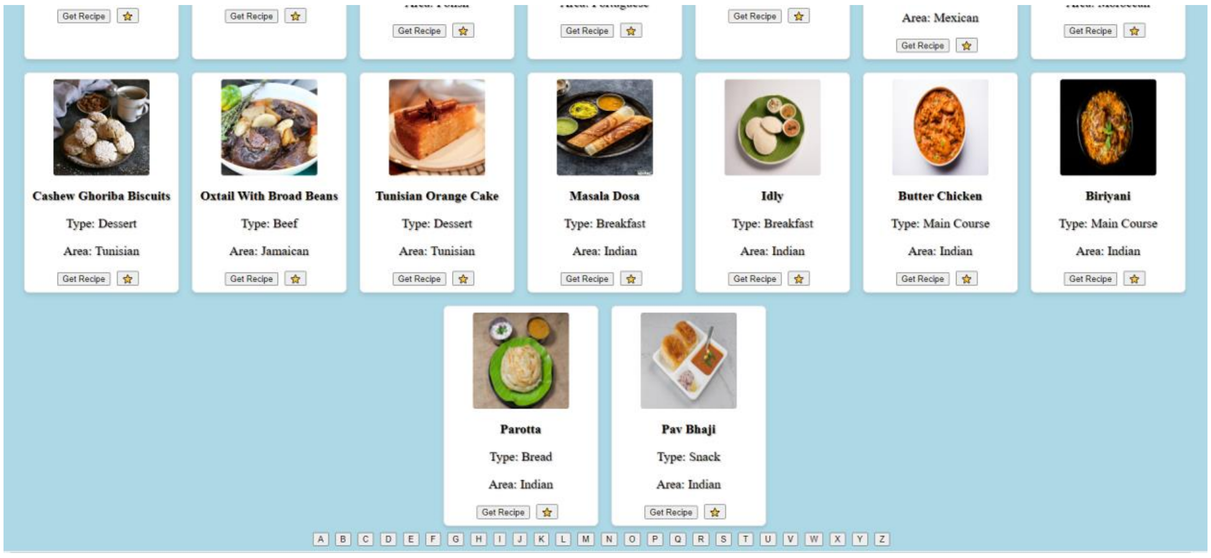
Eccles Cakes

Big Mac

Coddled Pork With

Strawberries Romanoff

Buttons for sorting



Cashew Ghoriba Biscuits
Type: Dessert
Area: Tunisian

Oxtail With Broad Beans
Type: Beef
Area: Jamaican

Tunisian Orange Cake
Type: Dessert
Area: Tunisian

Masala Dosa
Type: Breakfast
Area: Indian

Idly
Type: Breakfast
Area: Indian

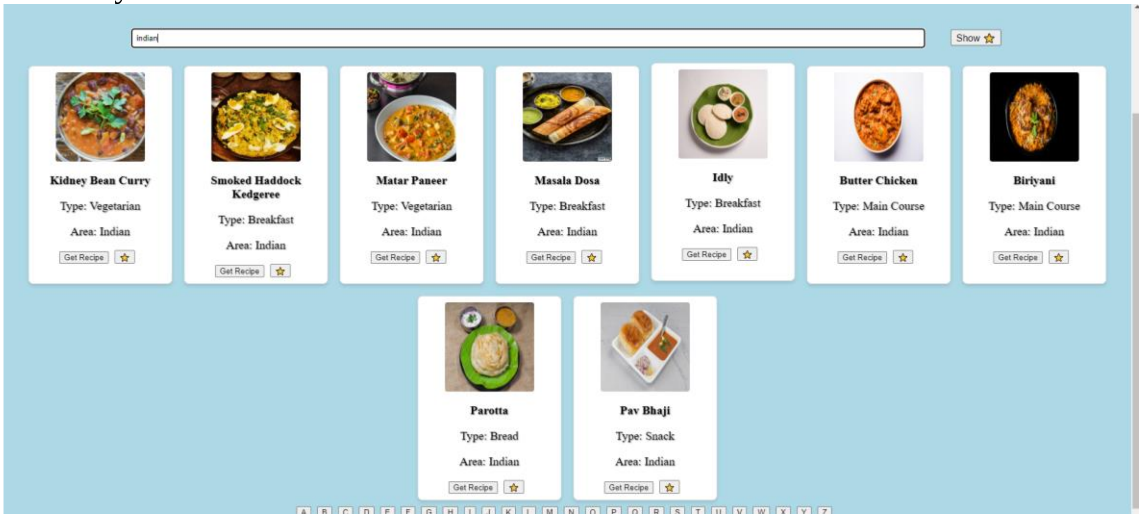
Butter Chicken
Type: Main Course
Area: Indian

Biriyani
Type: Main Course
Area: Indian

Parotta
Type: Bread
Area: Indian

Pav Bhaji
Type: Snack
Area: Indian

Search by area name :



Kidney Bean Curry
Type: Vegetarian
Area: Indian

Smoked Haddock Kedgeree
Type: Breakfast
Area: Indian

Matar Paneer
Type: Vegetarian
Area: Indian

Masala Dosa
Type: Breakfast
Area: Indian

Idly
Type: Breakfast
Area: Indian

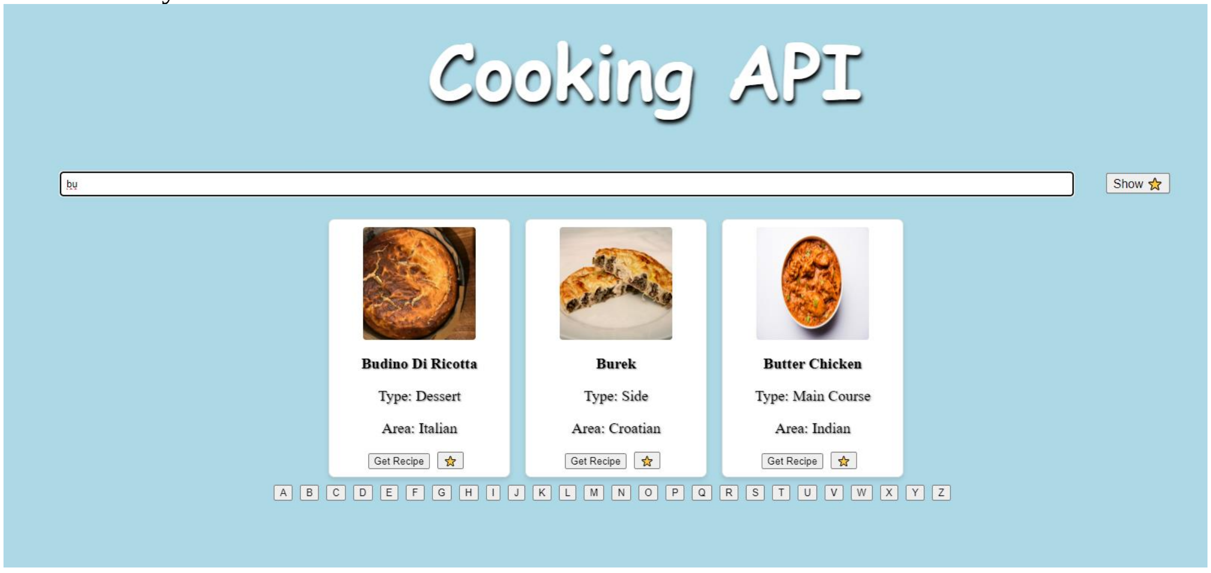
Butter Chicken
Type: Main Course
Area: Indian

Biriyani
Type: Main Course
Area: Indian

Parotta
Type: Bread
Area: Indian

Pav Bhaji
Type: Snack
Area: Indian

Auto filter by search item

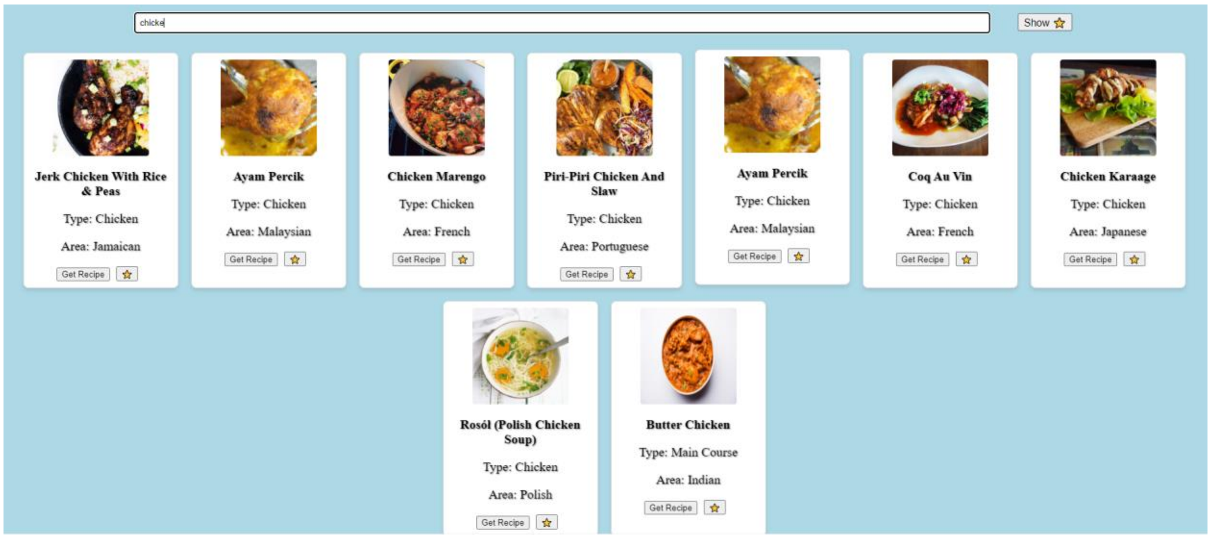


Budino Di Ricotta
Type: Dessert
Area: Italian

Burek
Type: Side
Area: Croatian

Butter Chicken
Type: Main Course
Area: Indian

Search by food type :



Jerk Chicken With Rice & Peas
Type: Chicken
Area: Jamaican

Ayam Percik
Type: Chicken
Area: Malaysian

Chicken Marengo
Type: Chicken
Area: French

Piri-Piri Chicken And Slaw
Type: Chicken
Area: Portuguese

Ayam Percik
Type: Chicken
Area: Malaysian

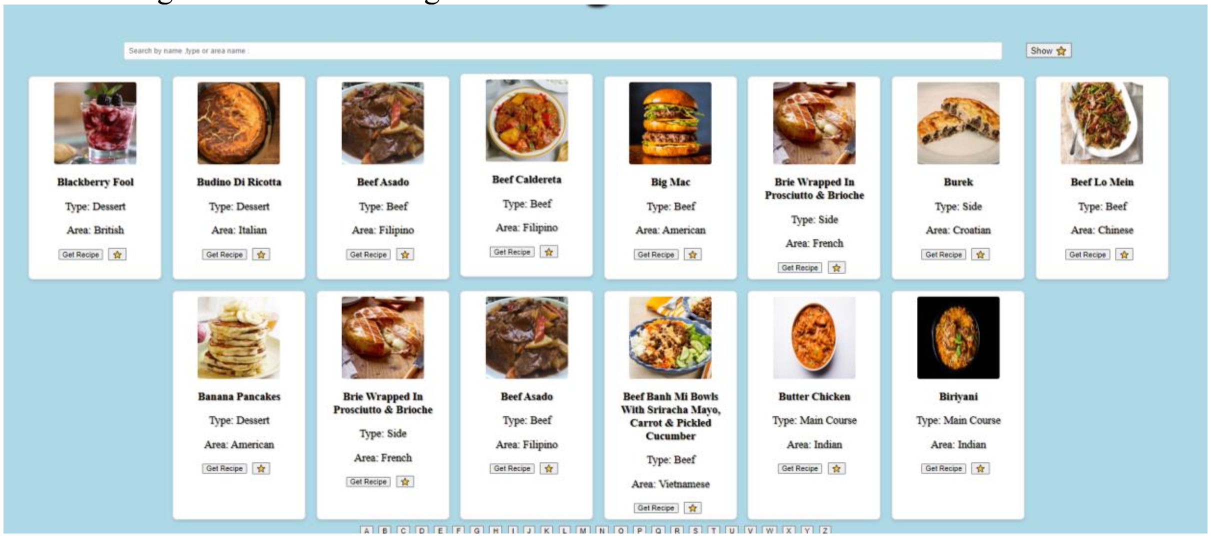
Coq Au Vin
Type: Chicken
Area: French

Chicken Karaage
Type: Chicken
Area: Japanese

Rosół (Polish Chicken Soup)
Type: Chicken
Area: Polish

Butter Chicken
Type: Main Course
Area: Indian

Food starting with letter ‘B’ using bottom buttons



Blackberry Fool
Type: Dessert
Area: British

Budino Di Ricotta
Type: Dessert
Area: Italian

Beef Asado
Type: Beef
Area: Filipino

Beef Caldereta
Type: Beef
Area: Filipino

Big Mac
Type: Beef
Area: American

Brie Wrapped In Prosciutto & Brioche
Type: Side
Area: French

Burek
Type: Side
Area: Croatian

Beef Lo Mein
Type: Beef
Area: Chinese

Banana Pancakes
Type: Dessert
Area: American

Brie Wrapped In Prosciutto & Brioche
Type: Side
Area: French

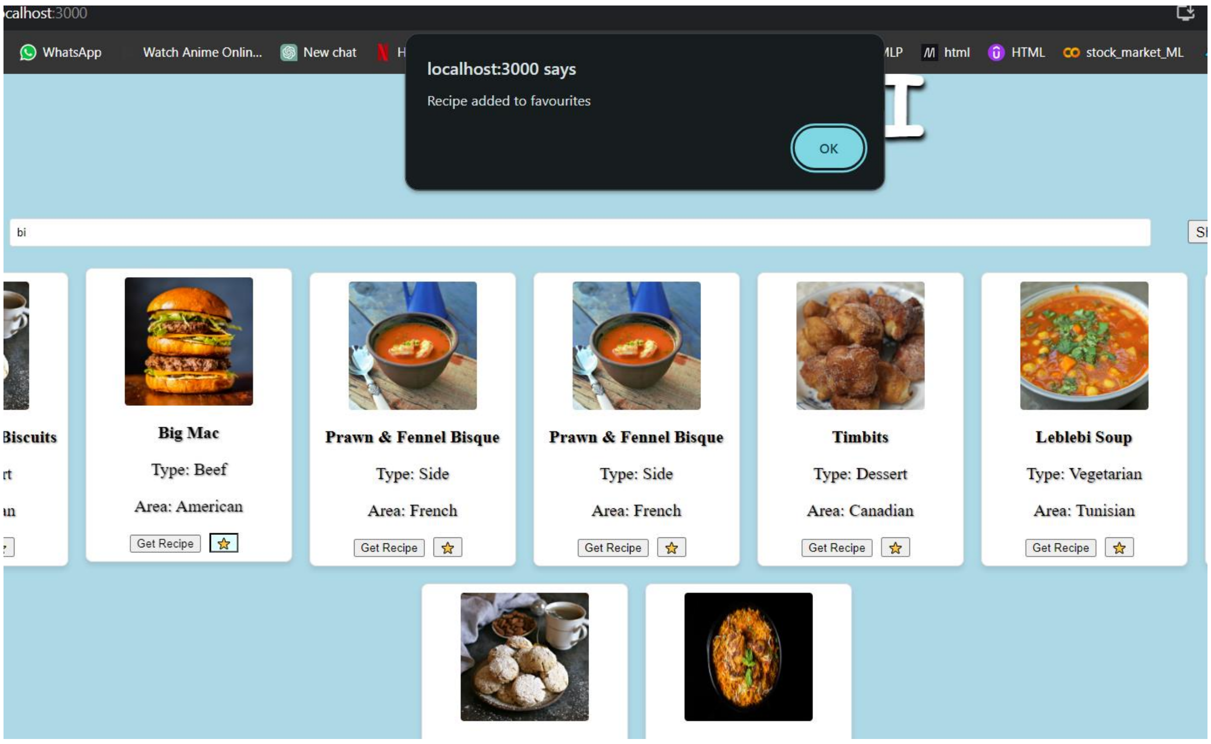
Beef Asado
Type: Beef
Area: Filipino

Beef Banh Mi Bowls With Sriracha Mayo, Carrot & Pickled Cucumber
Type: Beef
Area: Vietnamese

Butter Chicken
Type: Main Course
Area: Indian

Biriyani
Type: Main Course
Area: Indian

Adding to saved recipes



localhost:3000 says
Recipe added to favourites
OK

bi

Biscuits

Big Mac
Type: Beef
Area: American
Get Recipe

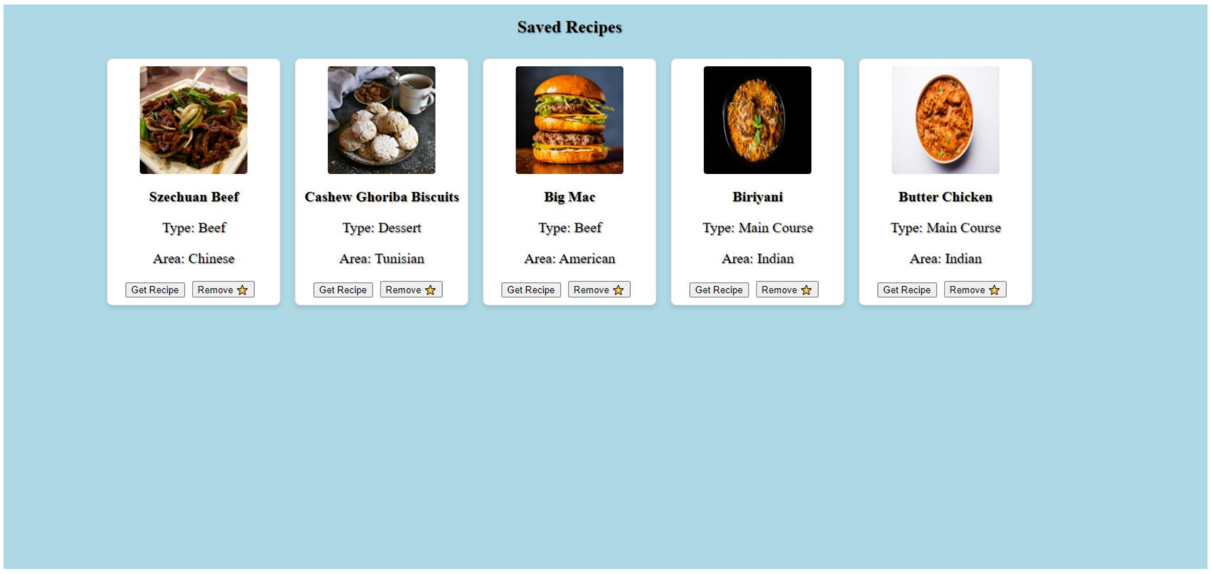
Prawn & Fennel Bisque
Type: Side
Area: French
Get Recipe

Prawn & Fennel Bisque
Type: Side
Area: French
Get Recipe

Timbits
Type: Dessert
Area: Canadian
Get Recipe

Leblebi Soup
Type: Vegetarian
Area: Tunisian
Get Recipe

Saved Recipes:



Saved Recipes

Szechuan Beef
Type: Beef
Area: Chinese
Get Recipe Remove

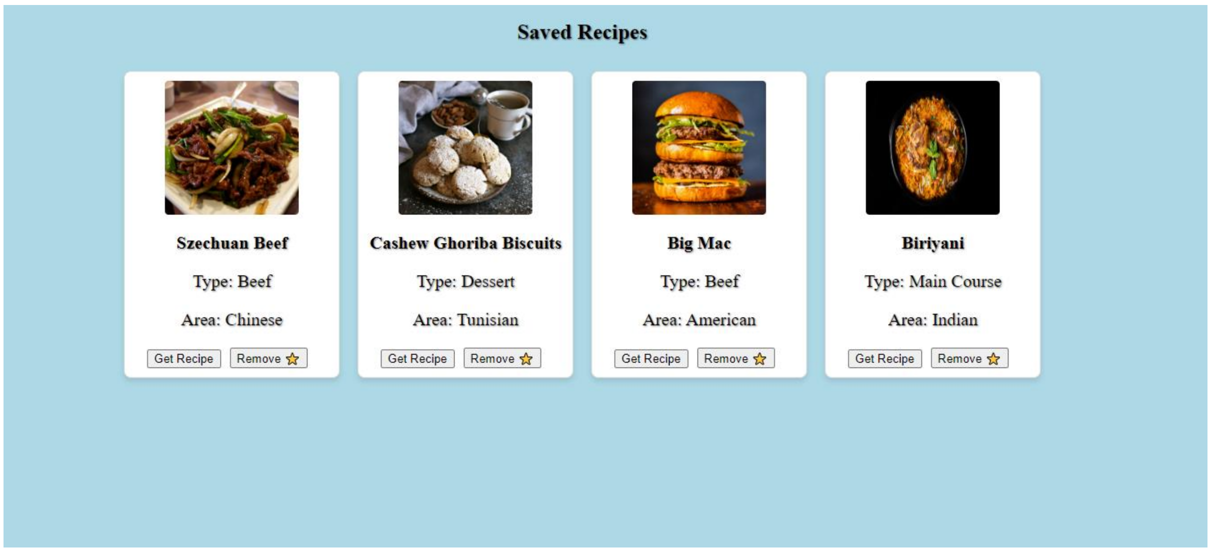
Cashew Ghoriba Biscuits
Type: Dessert
Area: Tunisian
Get Recipe Remove

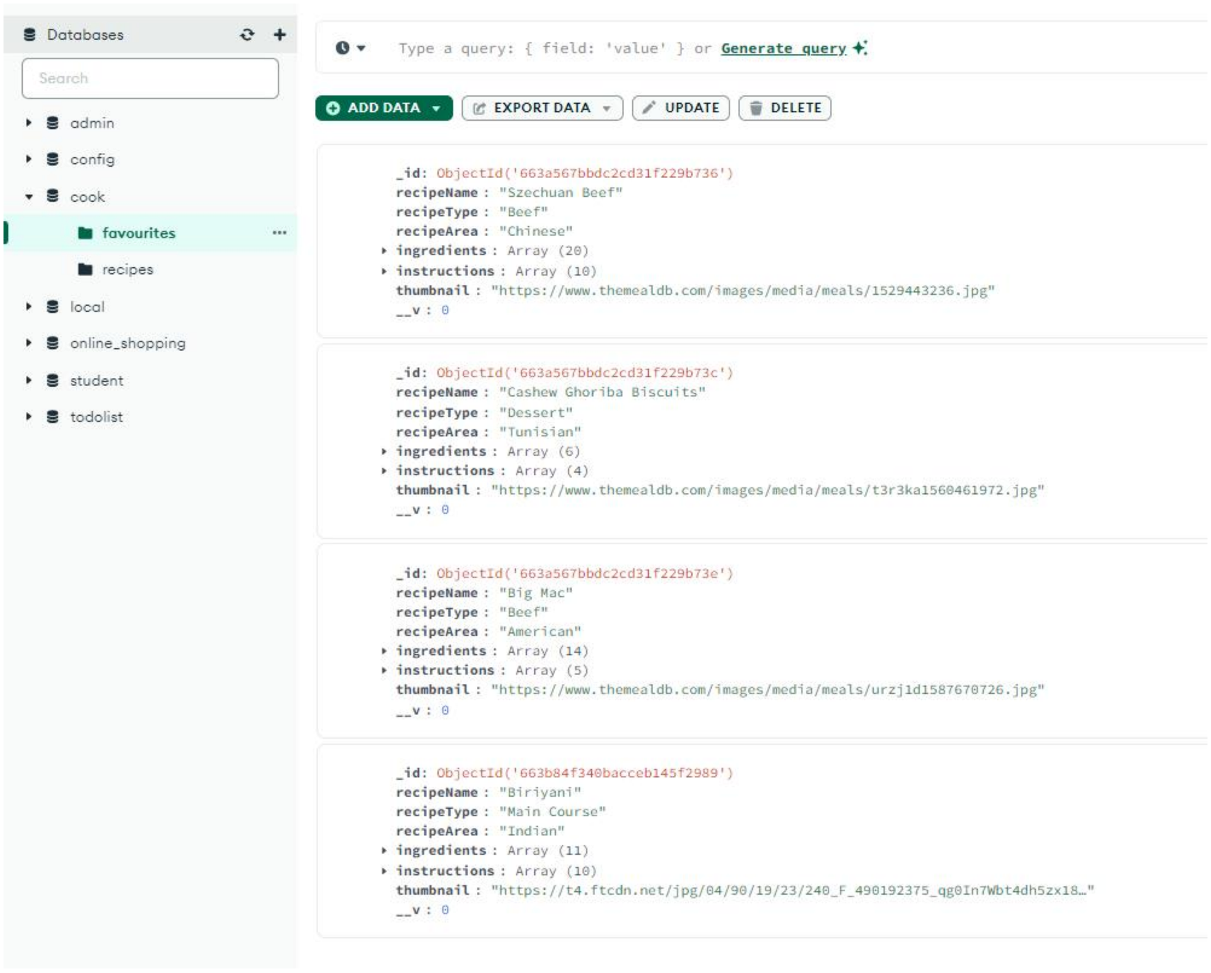
Big Mac
Type: Beef
Area: American
Get Recipe Remove

Biriyani
Type: Main Course
Area: Indian
Get Recipe Remove

Butter Chicken
Type: Main Course
Area: Indian
Get Recipe Remove

Removing Butter Chicken from saved recipes

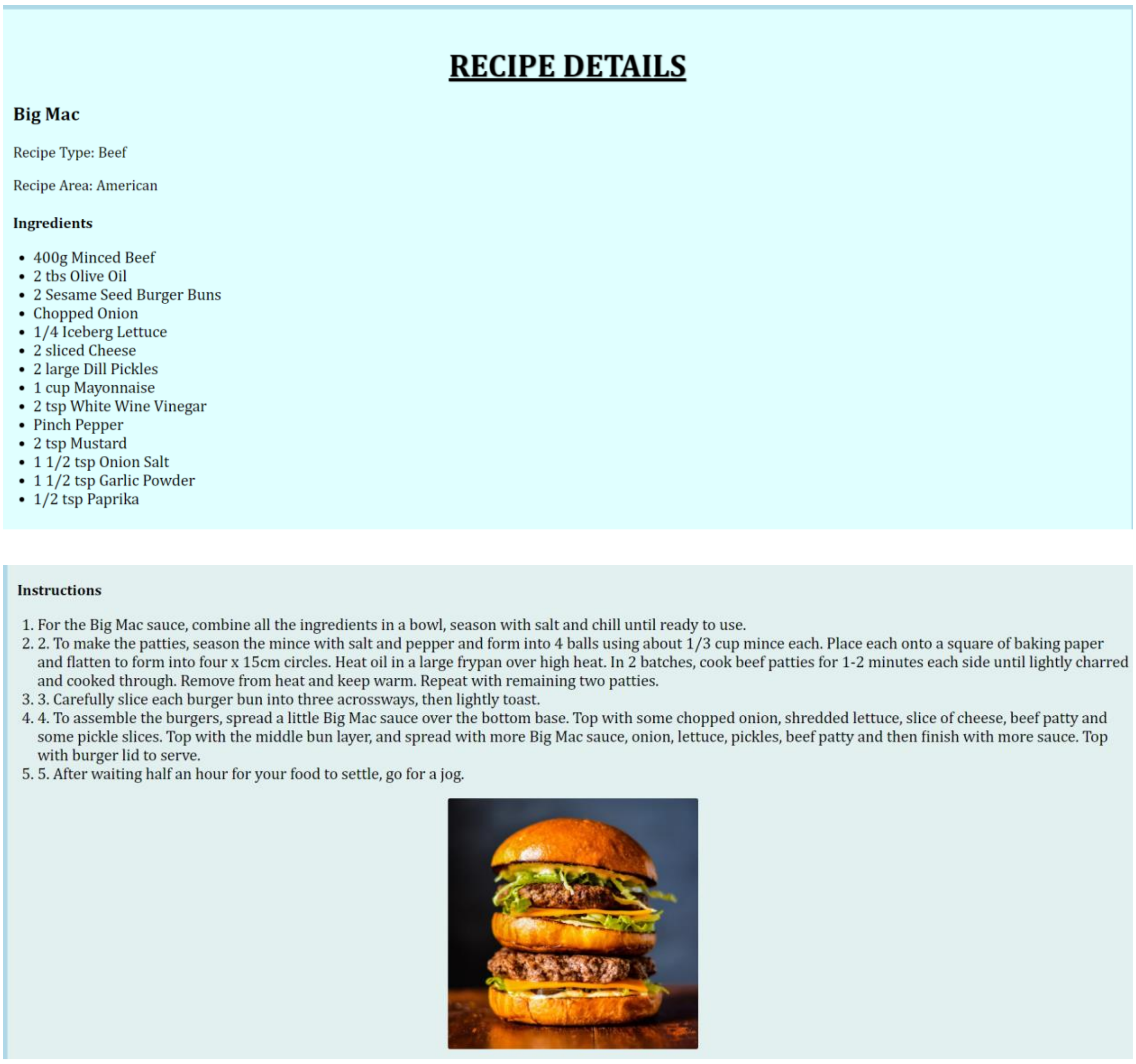




```
_id: ObjectId('663a567bbdc2cd31f229b736')
recipeName : "Szechuan Beef"
recipeType : "Beef"
recipeArea : "Chinese"
ingredients : Array (20)
instructions : Array (10)
thumbnail : "https://www.themealdb.com/images/media/meals/1529443236.jpg"
__v : 0

_id: ObjectId('663a567bbdc2cd31f229b73c')
recipeName : "Cashew Ghoriba Biscuits"
recipeType : "Dessert"
recipeArea : "Tunisian"
ingredients : Array (6)
instructions : Array (4)
thumbnail : "https://www.themealdb.com/images/media/meals/t3r3ka1560461972.jpg"
__v : 0

_id: ObjectId('663a567bbdc2cd31f229b73e')
recipeName : "Big Mac"
recipeType : "Beef"
recipeArea : "American"
ingredients : Array (14)
instructions : Array (5)
thumbnail : "https://www.themealdb.com/images/media/meals/urzj1d1587670726.jpg"
__v : 0

_id: ObjectId('663b84f340bacceb145f2989')
recipeName : "Biriyani"
recipeType : "Main Course"
recipeArea : "Indian"
ingredients : Array (11)
instructions : Array (10)
thumbnail : "https://t4.ftcdn.net/jpg/04/90/19/23/240_F_490192375_qg0In7Wbt4dh5zx18…"
__v : 0
```

Date:09-05-2024 IP CAT1 & 2 Assignment Vaishal Raj 3122215001119

Get Recipe functionality



RECIPE DETAILS

Big Mac

Recipe Type: Beef

Recipe Area: American

Ingredients

- 400g Minced Beef
- 2 tbs Olive Oil
- 2 Sesame Seed Burger Buns
- Chopped Onion
- 1/4 Iceberg Lettuce
- 2 sliced Cheese
- 2 large Dill Pickles
- 1 cup Mayonnaise
- 2 tsp White Wine Vinegar
- Pinch Pepper
- 2 tsp Mustard
- 1 1/2 tsp Onion Salt
- 1 1/2 tsp Garlic Powder
- 1/2 tsp Paprika

Instructions

1. For the Big Mac sauce, combine all the ingredients in a bowl, season with salt and chill until ready to use.
2. 2. To make the patties, season the mince with salt and pepper and form into 4 balls using about 1/3 cup mince each. Place each onto a square of baking paper and flatten to form into four x 15cm circles. Heat oil in a large frypan over high heat. In 2 batches, cook beef patties for 1-2 minutes each side until lightly charred and cooked through. Remove from heat and keep warm. Repeat with remaining two patties.
3. 3. Carefully slice each burger bun into three acrossways, then lightly toast.
4. 4. To assemble the burgers, spread a little Big Mac sauce over the bottom base. Top with some chopped onion, shredded lettuce, slice of cheese, beef patty and some pickle slices. Top with the middle bun layer, and spread with more Big Mac sauce, onion, lettuce, pickles, beef patty and then finish with more sauce. Top with burger lid to serve.
5. 5. After waiting half an hour for your food to settle, go for a jog.