PROJECT DOCUMENTATION

COOK BOOK-YOUR VIRTUAL KITCHEN ASSISTANT

INTRODUCTION

- **PROJECT TITLE :** cookbook - your virtual kitchen assistant
- **TEAM ID:** NM2025TMID37158
- **TEAM LEADER:**Deebikha.S(deebikhasharmila@gmail.com)
- **TEAM MEMBERS:**Poojaa Sree.V(poojaswager07@gmail.com)
  Vinodinii.B(vinodinii024@gmail.com)
  Swetha.G(swethaswetha8445@gmail.com)

PROJECT OVERVIEW

- PURPOSE: The purpose of the cookbook project is to create a well-structured and user-friendly platform that brings together a variety of recipes in one accessible source. It aims to preserve traditional cooking methods while also introducing modern recipes for users of all ages. By combining clear instructions, ingredients, and step-by-step procedures, the project makes cooking easier for beginners and helps experienced cooks experiment with new dishes.
- Another important purpose is to encourage healthy eating habits by offering recipes that are nutritious, affordable, and practical for everyday life. The project also seeks to build a digital collection that reduces the dependency on

scattered or unreliable sources, making it a reliable reference for households, students, and food enthusiasts.

- Furthermore, the cookbook project promotes cultural exchange by including recipes from different regions, allowing users to explore diverse cuisines. It not only supports learning and creativity in the kitchen but also serves as a long-term resource that can be continuously updated with new ideas.

- In essence, the purpose is to provide an engaging, educational, and convenient guide to cooking that fosters confidence, curiosity, and joy in preparing food.

- FEATURES: The cookbook project is designed with a set of practical and innovative features that make it a valuable resource for users. One of the key features is recipe categorization, where dishes are grouped into sections such as breakfast, lunch, dinner, desserts, snacks, and beverages. This allows users to easily find what they need without confusion.

- Another important feature is the step-by-step instructions, which guide users clearly through the cooking process. Each recipe includes details such as ingredients, cooking time, preparation method, and serving size. This makes the project suitable for both beginners and experienced cooks.

- The project also provides search and filter options, enabling users to quickly locate recipes based on ingredients, cuisine type, or dietary preferences. This ensures flexibility and convenience for users with different needs.

- A notable feature is the inclusion of nutritional information, which promotes healthy eating by giving insights into calories, proteins, fats, and other nutrients. This helps users make informed food choices.

- Additionally, the project supports interactive updates, meaning new recipes can be added, edited, or shared over time, keeping the cookbook dynamic and up to date.

- To make it engaging, the cookbook includes visual support, such as images or screenshots of dishes, enhancing the learning experience. Cultural diversity is another feature, as the project includes recipes from various cuisines, encouraging exploration of global flavors.

- Overall, the cookbook project is not just a recipe collection but an educational and interactive platform that promotes creativity, healthy living, and cultural appreciation through food.

- ARCHITECTURE
  1. Frontend (Presentation Layer):

- The frontend is developed using JavaScript, providing an interactive and user-friendly interface. It allows users to browse, search, and filter recipes, view details such as ingredients and preparation steps, and access images or screenshots of dishes.

2. Backend (Application Layer):

- The backend is built with Node.js, which handles the logic, data processing, and communication between the user interface and the database. It manages recipe requests, stores user inputs, and ensures secure interactions. The backend also supports scalability so that more recipes and features can be added in the future.

3. Database (Data Layer):

- The database uses MongoDB, where all recipe details, categories, nutritional values, and user information are stored. It ensures efficient data storage, retrieval, and updates. MongoDB's flexibility supports structured and unstructured data, making it suitable for managing diverse recipe content.

This layered architecture ensures modularity, scalability, and maintainability, allowing smooth performance and easy updates. The integration of these layers provides a seamless cooking experience for users.

- SETUP INSTRUCTION:

- **Install Required Software:**
  Download and install Node.js (for backend).
  Install MongoDB (for database).

Make sure you have a code editor (e.g.,VS Code) and browsers.

- **Download/Clone Project Files:**
  Get the cookbook project source code from the repository or ZIP file.
  Extract it into a working folder.

- **Install Dependencies**
  Open a terminal in the project folder.
  Run : npm install
  This will install all necessary packages mentioned in package.json.

- **Configure Database**
  Start MongoDB service.
  Create a database (e.g., cookbookDB).
  Update the database connection URL inside the project's configuration file.

- **Run the Backend Server**
  In the terminal,

- run : node server.js
  The server will start and connect to MongoDB.

- **Launch the Frontend**
  Open the index.html file in a browser or run the frontend with a local server if needed.

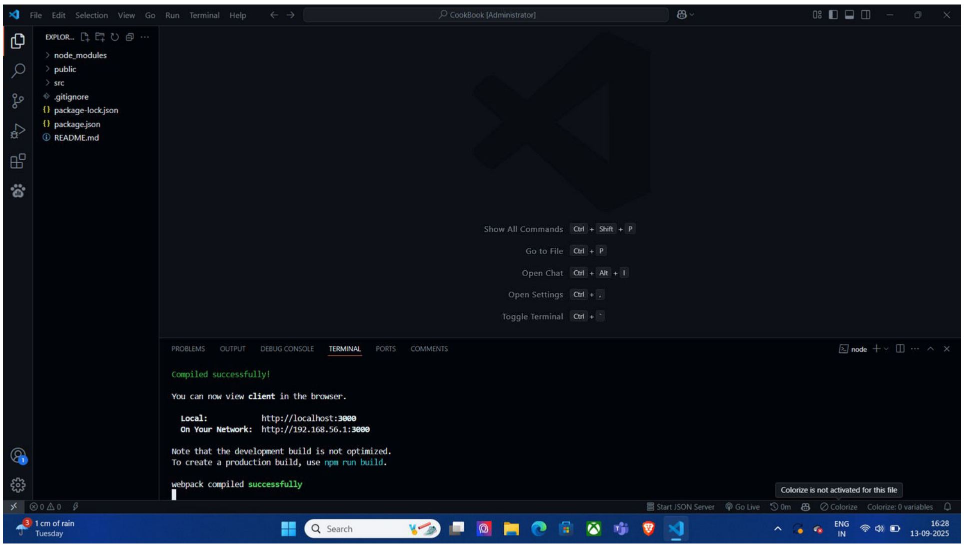
- **Test the Application**
  Search, view, and add sample recipes to confirm everything is working properly.

- **SOURCE CODE:**

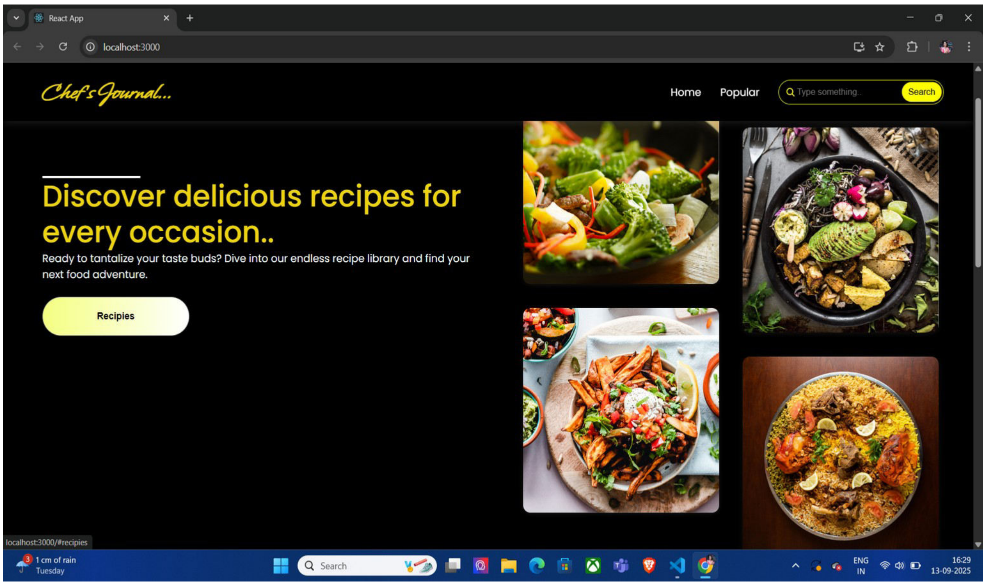
```
cookbook-project/
│
├── backend/
│   ├── server.js
│   ├── routes/
│   │   └── recipeRoutes.js
│   ├── models/
│   │   └── recipeModel.js
│   └── config/
│       └── db.js
│
├── frontend/
│   ├── index.html
│   ├── style.css
│   └── script.js
│
├── package.json
└── README.mdconst mongoose =
require("mongoose");

const connectDB = async () => {
  try {
    await
mongoose.connect("mongodb://localhost:27017/cookboo
kDB", {
      useNewUrlParser: true,
      useUnifiedTopology: true,
    });
    console.log("MongoDB Connected...");
  } catch (error) {
    console.error("Database connection failed:",
error.message);
    process.exit(1);
  }
};

module.exports = connectDB;const mongoose =
require("mongoose");

const recipeSchema = new mongoose.Schema({
  title: { type: String, required: true },
  ingredients: [String],
  instructions: String,
  category: String,
  createdAt: { type: Date, default: Date.now },
});

module.exports = mongoose.model("Recipe",
recipeSchema);const express = require("express");
const Recipe = require("../models/recipeModel");

const router = express.Router();

// Get all recipes
router.get("/", async (req, res) => {
  const recipes = await Recipe.find();
  res.json(recipes);
});
```

```
// Add new recipe
router.post("/", async (req, res) => {
  const newRecipe = new Recipe(req.body);
  await newRecipe.save();
  res.json(newRecipe);
});

module.exports = router;const express =
require("express");
const connectDB = require("./config/db");
const recipeRoutes = require("./routes/recipeRoutes");
const cors = require("cors");

const app = express();
connectDB();

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

// API routes
app.use("/api/recipes", recipeRoutes);

const PORT = 5000;
app.listen(PORT, () => console.log(Server running on
port ${PORT}));<!DOCTYPE html>
<html>
<head>
  <title>Cookbook Project</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Cookbook Recipes</h1>
  <div id="recipes"></div>
  <script src="script.js"></script>
</body>
</html>body {
  font-family: Arial, sans-serif;
  margin: 20px;
}

h1 {
  color: #2c3e50;
}

.recipe {
  border: 1px solid #ccc;
  padding: 10px;
  margin: 10px;
}async function fetchRecipes() {
  const response = await
fetch("http://localhost:5000/api/recipes");
  const recipes = await response.json();

  const container = document.getElementById("recipes");
  container.innerHTML = "";

  recipes.forEach(recipe => {
    const div = document.createElement("div");
    div.className = "recipe";
    div.innerHTML = `<h3>${recipe.title}</h3>
                    <p><b>Ingredients:</b> $
{recipe.ingredients.join(", ")}</p>
                    <p><b>Instructions:</b> $
{recipe.instructions}</p>`;
    container.appendChild(div);
  });
}

fetchRecipes();
```

SCREENSHORTS:

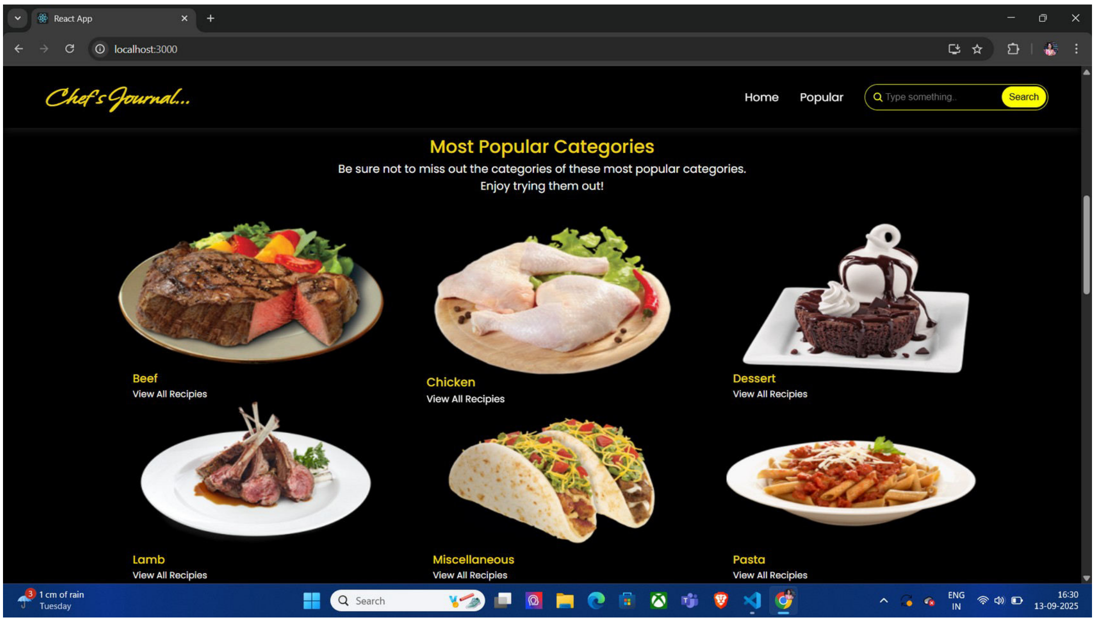


```
Compiled successfully!

You can now view client in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



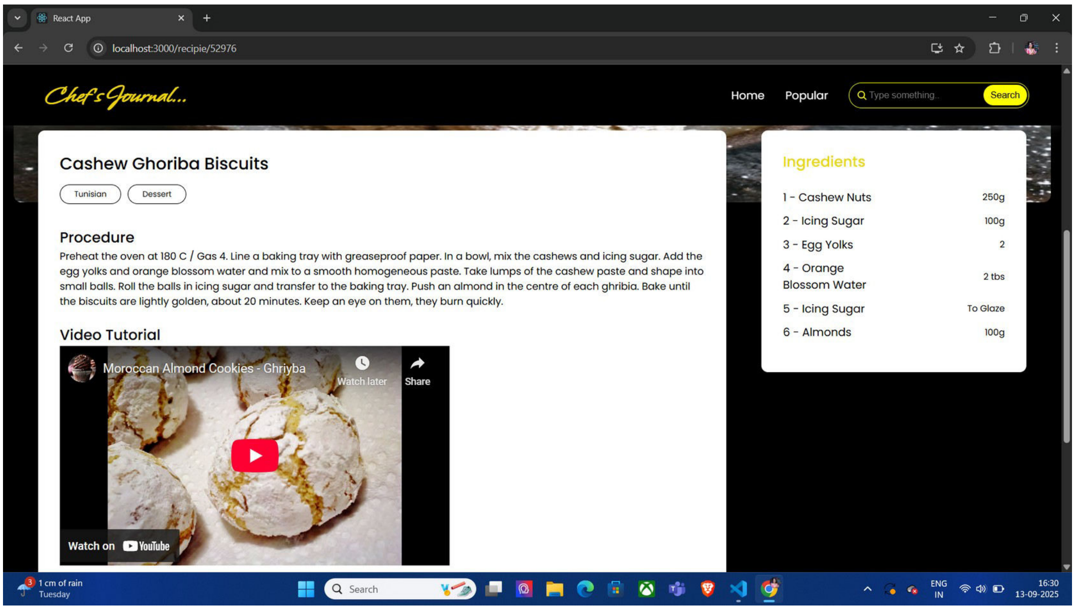
Discover delicious recipes for every occasion..

Ready to tantalize your taste buds? Dive into our endless recipe library and find your next food adventure.



Most Popular Categories

Be sure not to miss out the categories of these most popular categories.
Enjoy trying them out!



Cashew Ghoriba Biscuits

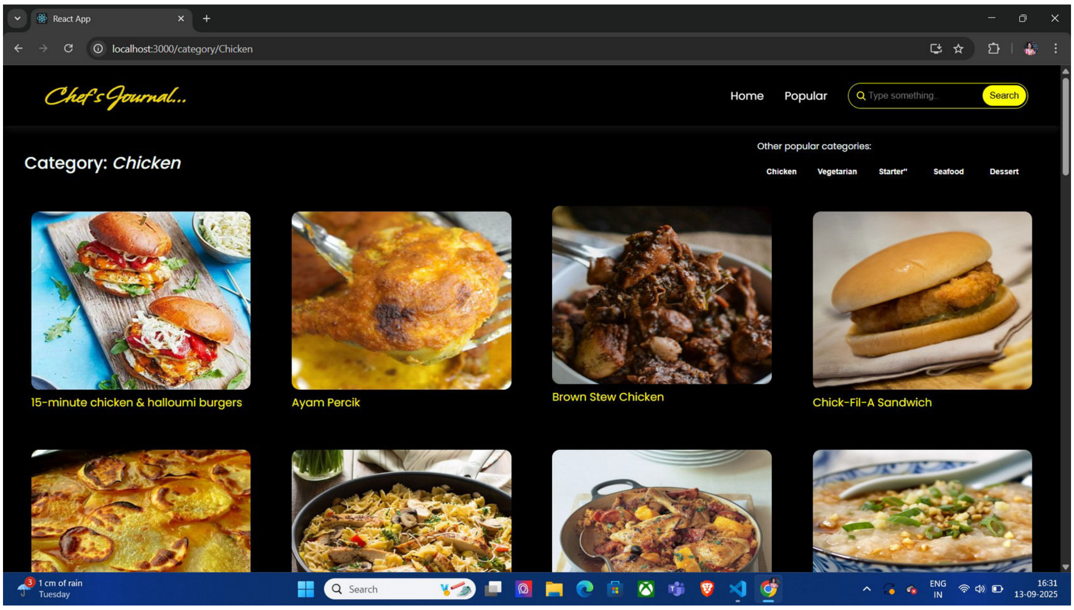
Procedure

Preheat the oven at 180 C / Gas 4. Line a baking tray with greaseproof paper. In a bowl, mix the cashews and icing sugar. Add the egg yolks and orange blossom water and mix to a smooth homogeneous paste. Take lumps of the cashew paste and shape into small balls. Roll the balls in icing sugar and transfer to the baking tray. Push an almond in the centre of each ghribia. Bake until the biscuits are lightly golden, about 20 minutes. Keep an eye on them, they burn quickly.

Video Tutorial

Ingredients

1 - Cashew Nuts	250g
2 - Icing Sugar	100g
3 - Egg Yolks	2
4 - Orange Blossom Water	2 tbs
5 - Icing Sugar	To Glaze
6 - Almonds	100g



Category: *Chicken*

Other popular categories:

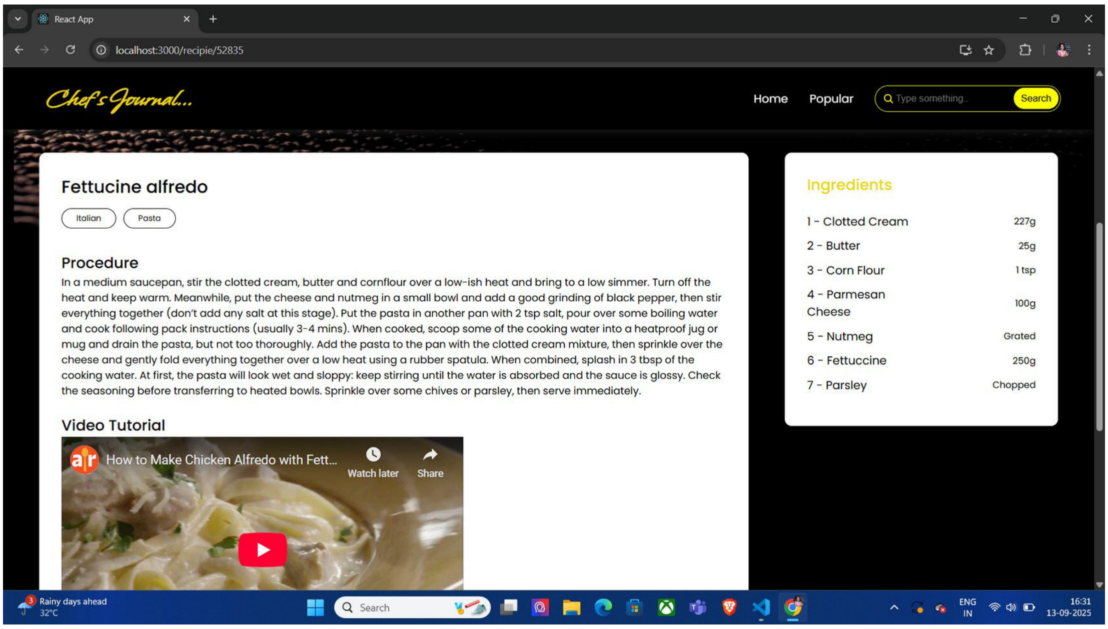
Chicken Vegetarian Starter" Seafood Dessert

15-minute chicken & halloumi burgers

Ayam Percik

Brown Stew Chicken

Chick-Fil-A Sandwich



Fettucine alfredo

Italian Pasta

Procedure

In a medium saucepan, stir the clotted cream, butter and cornflour over a low-ish heat and bring to a low simmer. Turn off the heat and keep warm. Meanwhile, put the cheese and nutmeg in a small bowl and add a good grinding of black pepper, then stir everything together (don't add any salt at this stage). Put the pasta in another pan with 2 tsp salt, pour over some boiling water and cook following pack instructions (usually 3-4 mins). When cooked, scoop some of the cooking water into a heatproof jug or mug and drain the pasta, but not too thoroughly. Add the pasta to the pan with the clotted cream mixture, then sprinkle over the cheese and gently fold everything together over a low heat using a rubber spatula. When combined, splash in 3 tbsp of the cooking water. At first, the pasta will look wet and sloppy: keep stirring until the water is absorbed and the sauce is glossy. Check the seasoning before transferring to heated bowls. Sprinkle over some chives or parsley, then serve immediately.

Video Tutorial

How to Make Chicken Alfredo with Fett...

Ingredients

1 - Clotted Cream	227g
2 - Butter	25g
3 - Corn Flour	1 tsp
4 - Parmesan Cheese	100g
5 - Nutmeg	Grated
6 - Fettuccine	250g
7 - Parsley	Chopped

- **DEMO LINK:**

  https://drive.google.com/file/d/1mXkFPYJizTtqdmjWP13tlQ9v19fMcnpP/view?usp=drivesdk

  https://cook-s-journal.vercel.app/