Mayuresh Naidu

HTTP5126 Final Project Proposal

06/04/2025

**cookbook\_db Database Proposal**

## Real World Scenario (2 marks)

Being a student, cooking everyday is like a difficult task. However, what's even more difficult is to decide what to cook. Hence, I have decided to create a food recipe database that will allow me to efficiently organise and manage my recipe collection. I will be the primary user and admin responsible for managing the database.

## Problems (2 marks) & Features (4 marks)

### Problem 1: Recipe Search

I (the user) often find it challenging to quickly find a recipe based on a primary ingredient, I have to manually browse through many recipes to find something.

### Problem 1 - Solution

The solution is to implement a search feature using a Stored Procedure that will allow me to search for recipes by name, ingredients or cuisine. This procedure will query the ***recipes***, ***ingredients*** and ***recipe\_ingredients*** tables.

### Problem 2: Favourite Recipes

After I (the user) try many recipes, I forget which ones were successful. It is difficult to quickly recall my past favourite recipes.

### Problem 2 - Solution

Implement a rating system (on a scale of 1-5) within the ***recipes*** table. This can be done by creating a Database View that displays recipes sorted by their rating in descending order.

## Architecture Description (16 marks)

My database is called **cookbook\_db**.

### Database Tools (3 marks) & Justification (3 marks)

View: **top\_rated\_recipes**

This *View* selects *recipe\_id*, *name*, *rating* and *cuisine* from ***recipes*** ordered by *rating* in descending order.

Trigger: **update\_recipe\_timestamp**

This *Trigger* executes before an update on the ***recipes*** tables. It sets the *last\_updated* to the current time whenever a row in ***recipes*** is updated.

Procedure: **find\_recipes**

This *Procedure* accepts input parameters such as ingredient name or cuisine. It queries ***recipes*** and ***ingredients*** tables using JOINs and WHERE clauses based on the input parameters and returns recipes.

**Justification:**

* The view **top\_rated\_recipes** solves problem-2 by providing an easy way to view top-rated recipes.
* The trigger **update\_recipe\_timestamp** keeps track of when a recipe was last updated which is useful for data maintenance.
* The procedure **find\_recipes** solves Problem-1 by providing an efficient way to search the database using key parameters such as ingredients.

### Database ERD (7 marks) & Justification (2 marks)



#### 

**Justification:** The tables separate different entities such as recipes, ingredients and categories. The tables ***recipe\_ingredients*** and ***recipe\_categories*** are bridge tables that resolve many-to-many relationships:

* ***recipe\_ingredients*** links ***recipes*** and ***ingredients*** (a recipe can use many ingredients, and an ingredient can belong to many recipes).
* ***recipe\_categories*** links ***recipes*** and ***categories*** (a recipe can belong to multiple categories, and a category can contain many recipes).