CookEasy

Curated Recipes for the New Chef

A project by The Debuggers

CookEasy is a recipe library that provides tailored recipe recommendations to match a user's skill level, accessible ingredients, and schedule availability for new cooks who are unhappy with current recipe sites which cater to more experienced home cooks.

FEATURES

New chefs will be able to personalize their recipe searches based on various factors related to ease of cooking, such as skill level, available equipment, timeframe — for both food preparation and actual cook time — and ingredient availability. In addition to these features, the site will provide user's options to filter recipes by overall caloric value, ingredients to avoid (due to food sensitivity or taste) and cuisine, enabling users to further curate their recipe recommendations according to dietary needs and preferences.

In the future, this product could include additional search features including:

- Users can search by cost of ingredients match user's budget or skills (e.g. "mincing" or "braising") that a user may be particularly interested in practicing
- Enable users to create profiles to track cooked recipes, earn badges
- Optimize recommendations on more difficult cooking as users learn new skills.

THE DATA

- 1. <a href="https://www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions?select=RAW recipes-and-user-interactions?select=RAW recipes-and-user-interactions.select=RAW recipes-and-user-inter
- 2. https://www.kaggle.com/irkaal/foodcom-recipes-and-reviews

Up until now, both our data sources have csv files which we are going to assess for data discrepancies and once we are okay with them, we would then import them to our schema.

THE TEAM

Kate Caldwell Saurav Shah Clara Jiaoyue Lan Colin Nordquist Linli Mei Yongjiang Yu

PM2 Notes

Tables to create:

- 1. Recipes
- 2. Users
- 3. Ingredients
- 4. Association (Recipes & Ingredients)
- 5. Reviews
- 6. Recommendations
- 7. RecipeBox
- 8. Keywords
- 9. Keywords Association Table
- 10. Ratings Table (Prep Rating, Cook Rating, Cost Rating, Taste Rating)

```
PM2 Create Tables:
CREATE SCHEMA IF NOT EXISTS CookEasyDatabase;
USE CookEasyDatabase;
DROP TABLE IF EXISTS SavedRecipes;
DROP TABLE IF EXISTS Ratings;
DROP TABLE IF EXISTS Recommendation;
DROP TABLE IF EXISTS Reviews;
DROP TABLE IF EXISTS Users:
DROP TABLE IF EXISTS RecipeIngredients;
DROP TABLE IF EXISTS Ingredients;
DROP TABLE IF EXISTS KeywordInRecipe;
DROP TABLE IF EXISTS Keywords;
DROP TABLE IF EXISTS Recipes:
CREATE TABLE Keywords (
      KeywordName VARCHAR(255),
 CONSTRAINT pk_Keywords_KeywordName
                                              # changed, use KeywordName as primary key
             PRIMARY KEY (KeywordName)
);
CREATE TABLE Ingredients (
 IngredientName VARCHAR(255),
 CONSTRAINT pk_Ingredients_IngredientName
                                              # changed, use IngredientName as primary key
             PRIMARY KEY (IngredientName)
);
CREATE TABLE Recipes (
      RecipeID INT,
```

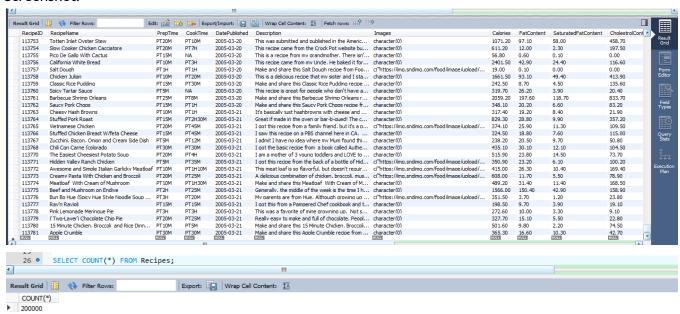
```
RecipeName VARCHAR(255),
 PrepTime VARCHAR(255), #NEED TO UNDERSTAND THIS BETTER
 CookTime VARCHAR(255), #NEED TO UNDERSTAND THIS BETTER
 DatePublished DATE,
 Description LONGTEXT,
                          # change to LONGTEXT because the text is too long
                                                                              by LinLi
 Images LONGTEXT,
                          # change to LONGTEXT because the text is too long
                                                                              by LinLi
 Calories DECIMAL (10, 2),
 FatContent DECIMAL (10, 2),
 SaturatedFatContent DECIMAL (10, 2),
 CholestrolContent DECIMAL (10, 2),
 SodiumContent DECIMAL (10, 2),
 CarbohydrateContent DECIMAL (10, 2),
 FiberContent DECIMAL (10, 2),
 SugarContent DECIMAL (10, 2),
 ProteinContent DECIMAL (10, 2),
 RecipeServings DECIMAL (10, 2),
 RecipeInstructions LONGTEXT,
 CONSTRAINT pk_Recipes_RecipeID
            PRIMARY KEY(RecipeID)
);
CREATE TABLE KeywordInRecipe (
      KeywordRecipeID INT AUTO_INCREMENT,
 KeywordName VARCHAR(255),
 RecipeID INT,
 CONSTRAINT pk KeywordInRecipe KeywordRecipeID
             PRIMARY KEY (KeywordRecipeID),
      CONSTRAINT fk KeywordInRecipe KeywordName
             FOREIGN KEY (KeywordName)
   REFERENCES Keywords(KeywordName)
   ON UPDATE CASCADE ON DELETE CASCADE,
      CONSTRAINT fk KeywordInRecipe RecipeID
             FOREIGN KEY (RecipeID)
   REFERENCES Recipes(RecipeID)
  ON UPDATE CASCADE ON DELETE CASCADE
 );
CREATE TABLE RecipeIngredients (
      RecipeIngredientID INT AUTO_INCREMENT,
 RecipeID INT,
 IngredientName VARCHAR(255),
 Quantity VARCHAR(255),
                                # changed
 CONSTRAINT pk_RecipeIngredients_RecipeIngredientID
             PRIMARY KEY (RecipeIngredientID),
      CONSTRAINT fk_RecipeIngredients_RecipeID
```

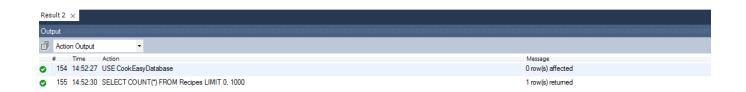
```
FOREIGN KEY (RecipeID)
   REFERENCES Recipes(RecipeID)
   ON UPDATE CASCADE ON DELETE CASCADE,
      CONSTRAINT fk RecipeIngredients IngredientName
             FOREIGN KEY (IngredientName)
   REFERENCES Ingredients(IngredientName)
  ON UPDATE CASCADE ON DELETE CASCADE
 );
CREATE TABLE Users (
 UserName VARCHAR(255),
 FIrstName VARCHAR(255),
 LastName VARCHAR(255),
 Email VARCHAR(255),
 CONSTRAINT pk Users UserName
            PRIMARY KEY(UserName)
);
CREATE TABLE SavedRecipes (
      SavedRecipeID INT AUTO_INCREMENT,
 Recipeld INT,
 UserName VARCHAR(255),
 RecommendedByCookEasy bool,
 CONSTRAINT pk_SavedRecipes_SavedRecipeID
            PRIMARY KEY(SavedRecipeID),
      CONSTRAINT fk_SavedRecipes_RecipeID
            FOREIGN KEY(RecipeID)
   REFERENCES Recipes(RecipeID)
   ON UPDATE CASCADE ON DELETE CASCADE.
      CONSTRAINT fk SavedRecipes UserName
            FOREIGN KEY(UserName)
   REFERENCES Users(UserName)
   ON UPDATE CASCADE ON DELETE CASCADE
);
CREATE TABLE Ratings (
      Ratingld INT,
 Recipeld INT,
 UserName VARCHAR(255),
 Difficulty DECIMAL (2, 1),
 Cost DECIMAL (2, 1),
 Taste DECIMAL (2, 1),
 Time DECIMAL (2, 1),
 CONSTRAINT pk_Ratings_RatingID
            PRIMARY KEY(RatingID),
```

```
CONSTRAINT fk_Ratings_RecipeID
            FOREIGN KEY(RecipeID)
   REFERENCES Recipes(RecipeID)
  ON UPDATE CASCADE ON DELETE CASCADE,
      CONSTRAINT fk_Ratings_UserName
            FOREIGN KEY(UserName)
   REFERENCES Users(UserName)
  ON UPDATE CASCADE ON DELETE SET NULL
);
CREATE TABLE Recommendation (
      RecommendationId INT,
 Recipeld INT,
 UserName VARCHAR(255),
 CONSTRAINT pk_Recommendation_RecommendationID
            PRIMARY KEY (RecommendationID),
      CONSTRAINT fk_Recommendation_RecipeID
            FOREIGN KEY (RecipeID)
   REFERENCES Recipes(RecipeID)
  ON UPDATE CASCADE ON DELETE CASCADE,
      CONSTRAINT fk_Recommendation_UserName
            FOREIGN KEY (UserName)
   REFERENCES Users(UserName)
  ON UPDATE CASCADE ON DELETE SET NULL
);
CREATE TABLE Reviews (
      ReviewId INT,
 UserName VARCHAR(255),
 Recipeld INT,
 Content VARCHAR(255),
 Created TIMESTAMP,
 CONSTRAINT pk Reviews ReviewID
            PRIMARY KEY (ReviewID),
      CONSTRAINT fk_Reviews_UserName
            FOREIGN KEY (UserName)
   REFERENCES Users(UserName)
  ON UPDATE CASCADE ON DELETE SET NULL,
      CONSTRAINT fk_Reviews_RecipeID
            FOREIGN KEY (RecipeID)
   REFERENCES Recipes(RecipeID)
  ON UPDATE CASCADE ON DELETE CASCADE
);
```

```
Insert Data:
# load data for recipes
LOAD DATA INFILE 'path/recipes_0.csv'
       INTO TABLE Recipes
       FIELDS TERMINATED BY "OPTIONALLY ENCLOSED BY ""
       LINES TERMINATED BY '\n'
       IGNORE 1 LINES
       (RecipeID, RecipeName, @dummy, @dummy, CookTime, PrepTime, @dummy, @val1,
 Description, Images, @dummy, @dummy, @dummy, @dummy, @dummy, @dummy, Calories,
 FatContent, SaturatedFatContent, CholestrolContent, SodiumContent, CarbohydrateContent,
 FiberContent, SugarContent, ProteinContent, @valServ, @dummy, RecipeInstructions)
 SET DatePublished = substr(@val1, 1, 10);
# load data for keywords
LOAD DATA INFILE "path/keywords 0.csv"
       INTO TABLE Keywords
       FIELDS TERMINATED BY "
       LINES TERMINATED BY '\n'
 (KeywordName);
# load data for ingredients
LOAD DATA INFILE "path/ingredients 0.csv"
       INTO TABLE Ingredients
       FIELDS TERMINATED BY "
       LINES TERMINATED BY '\n'
 (IngredientName);
# load data for recipe_keywords
LOAD DATA INFILE "path/recipe_keywords_0.csv"
       INTO TABLE KeywordInRecipe
       FIELDS TERMINATED BY "
       LINES TERMINATED BY '\n'
 (RecipeID, KeywordName);
# load data for recipe_ingredients
# recipe_ingredients_0 contains lots of data, if cannot load data directly from recipe_ingredients_0, try
# recipe_ingredients_0_0 to recipe_ingredients_0_7, where I split the data into 7 smaller files
LOAD DATA INFILE "path/recipe_ingredients_0.csv"
       INTO TABLE RecipeIngredients
       FIELDS TERMINATED BY "
       LINES TERMINATED BY '\n'
       IGNORE 1 LINES
 (RecipeID, Quantity, IngredientName);
```

Screenshot:





Datasets:

https://drive.google.com/drive/folders/1BdFIHJfgwyg ikg YONyYufp5aVTn73R?usp=sharing