

System Design Project

The Recipes Index (Name Pending)

## Front Page, Sign up, log in, about us page header, top bar, footer

======================================================================

## Landing Page

The landing page introduces the platform with a few paragraphs explaining its purpose and features. It includes a "Learn More" button, along with Sign In and Sign Up options for account creation and authentication. Users can log in using Google authentication for a seamless experience. The top navigation bar provides access to key sections, including Featured Recipes, Top 10 Lists, Discover More, and an Alphabetical Recipe Index.

**Title bar**

Top title bar with features, Top 10’s, new recipes, Discover, index, ingredients, menu ………. About us, contact us………… all the way to the right of page your profile with 48x48 png upload of your choice, have pre-set 24 .png choices of profile images to upload.

**Header**

Left to right scroll of main page randomized recipes with button to learn more cards with scroll feature **6-8** each time, to populate form database with images, api will grab images from local searches to pull for similar food items based on recipe title and name.

The images can be randomized as well to create a dynamic view of the images for the food items.

**Body**

Scripts and features

Capabilities copy share learn more

Additional suggestion recipe reference

rate/ save / share / print/ star

Nutritional information

Additional recipes of similar items

Selections

**Footer**

**Table of contents** (dynamic)

Contact us

Help page resource

Email

Creators

Additional social links

Domain name System

build , security etc.

unit testing load balancers, …….

=====================================================================

## Profile Page

Users can create profiles and manage their favorites, ratings, and personal recipe submissions.

Recipes added by users are stored in a structured database and automatically indexed for search and discovery. Google authentication enables easy login, while a profile bar allows quick access to user settings.

User Information

Personal details

Saved recipes

Created recipes

Bookmarks

Ratings history

Content Management

Upload recipes

Manage images

Edit profile

====================================================================

## Operations management Page

Admin Features

User management

Content moderation

System analytics

Performance monitoring

====================================================================

## Print page

images , notes, portion size for recipe information, sizing for information serving sizes, textual sizes, placement for image, placement for recipe, cool time, genre information,

Ingredients,instructions notes for dishes, footer bar fit within print minimal for resource info with us.

Image placement

Recipe text formatting

Portion size and serving information

Cooking time and genre details

Ingredients and instructions

Notes section

Minimal footer with essential information

=====================================================================

## Recipe Index & API Integration

The recipe index dynamically fetches food recipes from external APIs and

displays them using interactive cards. The main page features a scrolling section with 20 recipe

cards, showcasing popular dishes. Each card contains a title, image, short description, and rating, with placeholders for additional data. Clicking on a card links to a detailed recipe page, displaying full ingredients, step-by-step instructions, images, and related recipes.

=====================================================================

## Contact page

Contact form

Support information

FAQ section

======================================================================

## Security, user management, admin page

An admin panel provides user management, content moderation, analytics, and system performance monitoring. The back-end is optimized for load balancing, caching, and structured data to improve SEO and responsiveness. The site is fully mobile-optimized, ensuring smooth functionality across different devices.

Admin Page

User account management

Access control

Security settings

Backup and recovery options

======================================================================

## 

## Search index page

The site includes an alphabetical recipe index, dynamically generating a complete list of all

available recipes. Recipes are searchable and filterable by ingredients, cuisine, and dietary preferences.

Advanced search options

Filters (ingredients, cuisine, dietary restrictions)

Search result display

=====================================================================

## Sign in sign up page

User registration form

Login interface

Password recovery

Social media login integration

====================================================================

Ensure mobile-friendly layout

Use media queries or mobile-first approach

Optimize images for faster page load speed

Implement a searchable recipe database

Use high-quality, professional food photography

Create interactive cooking guides

Add meal planning features

Use structured data and schema markup for recipes

Implement JSON-LD schema for better Google rankings

Set canonical meta tags to avoid duplicate content issues

Use descriptive and engaging recipe titles

recipe name, cooking time, servings, main photo, ingredient list, instructions, cooking notes, and helper tools

Use step-by-step photos for instructions

Provide both US and metric measurement options

Implement simple navigation with clear categories

Use the 1:10 rule for category creation (1 category for every 10 recipes)

Ensure recipes are findable within 3 clicks

social media sharing buttons

user interaction through social features

caching mechanisms for faster load times

Use load balancers for better traffic management

Conduct regular unit testing and performance testing

=======================================================================

**FRONTEND THOUGHTS**

======================================================================

HTML5

CSS3 and Tailwind

Javascript

AJAX

=====================================================================

**RESTful API**

=======================================================================

NODE.js + Express

Stateless

Resource oriented

Following the CRUD method

“Create, reuse, update, delete”

======================================================================

**BACKEND THOUGHTS**

======================================================================

NODE.js + Express

Json web token

Helment.js for security

Need a relational schema

======================================================================

**Host domain**

======================================================================

Aws we did this 2x times might as well go for 3rd

Ec2 instance s3 bucket rds database with postgre and we need an api gateway, domain name and etc……

Azure can't go wrong with microsoft

=======================================================================

**Endpoints** **Function**

POST /auth/signup User registration

POST /auth/login User authentication (Google/Facebook/Email)

GET /recipes Fetch all recipes (with filters & pagination)

GET /recipes/popular Fetch trending recipes

GET /recipe/:id Fetch single recipe details

POST /recipe User posts a new recipe

POST /recipe/:id/rate Rate a recipe

POST /recipe/:id/favorite Save recipe to favorites

GET /user/profile Fetch user profile details

PUT /user/profile Update user profile

GET /admin/analytics Fetch system analytics (admin only)

=======================================================================

**PostgreSQL:**

Users Table (id, name, email, profile\_img, auth\_provider)

Recipes Table (id, title, description, image\_url, ingredients, instructions, created\_by, rating)

Ratings Table (id, user\_id, recipe\_id, rating)

Favorites Table (id, user\_id, recipe\_id)

=======================================================================

## CORE NECESSITIES AKA BIBLE OF PROJECT MAKING

======================================================================

We want it scalable

We want to focus on composition instead of inheritance

We want to focus on performance and optimization in our planing

As we build this project recall it will be api driven architecture

Security first so a good place to start would be creating the login and security, including role based access controls

Clean code””””” *keep it simple stupid*:””””””

Can optionally look at mobile first design conceptualization and focus on our development via **vue.js** and build from there

Include **unit testing** as we develop our code and project

Properly document for development, improvements and information.