

## 6.3HD - Design Document

### 1. Introduction

This document outlines the design and development of a single-page web app for searching cocktails based on names, categories, alcoholic volumes, ingredients, and other functionalities. The frameworks used for this project are VueJS and Bootstrap. This document is a comprehensive application overview, including the requirements, design, implementation, and deployment. The web app will enable users to search for cocktails and view detailed information about each cocktail, including ingredients, instructions for mixing, and sample images. The app's front end will be built using VueJS, with the grid design and layout strapped together using Bootstrap. The backend will be powered using [TheCocktailDB](#) API, an open and free-of-charge database of drinks and cocktails worldwide. In addition, the Axios library will also be used to make HTTP requests in our browser environment. The document will further detail the application's structure, design, and implementation.

### 2. Requirements

#### A. Functional Requirements

The app contains five main functionalities: generate a random cocktail; search cocktail by name, ingredients, alcoholic volumes, categories, and glassware; view the detailed instructions, images, and ingredients of each cocktail; look up ingredients; view ingredient details.

1. Generate a random cocktail: A new and random cocktail featured as a drink of the day.
2. Search cocktails by name, ingredients, alcoholic volumes, categories, and glassware: Each filter can be used independently or together to find the user's best drink.
3. View a cocktail's details: The image, name, category, alcoholic volume, glassware, instructions for mixing, and ingredient will be presented in a comprehensive card view that can be used to look up other drinks with the same components.
4. View all ingredients: All ingredients used for mixing the cocktails will be presented in a search view, with each one shown in a card with the image, name, and will re-route you to the Ingredient Detail view on click.
5. View an ingredient's details: The image, name, type, alcoholic by volume (ABV), and description will be shown in a card view similar to the Cocktail Detail view.

#### B. Non-functional Requirements

The app contains three non-functional requirements: responsiveness, usability, and accessibility.

1. Responsiveness: The website automatically adjusts its layout and content to fit different screen sizes and devices, whether the user is accessing the website using a desktop, a phone, or a tablet.

2. Usability: The app was designed with a clear and intuitive user interface, making it easy for users to search for cocktails, view detailed information about each cocktail, and save their favourite recipes.
3. Accessibility: The accessibility of the app can be interpreted from the high-contrast display of all pages and interactive elements.

### 3. Design Architecture

1. Front-end:
  - a. VueJS: Power the app using custom and reusable templates on the view.
  - b. Vue Router: Navigate between different views in our single-page application.
  - c. Bootstrap: Create a grid structure to style the app in a logical and user-friendly interface.
  - d. Axios: Promise-based HTTP client for the browser and node.js.
2. Back-end: [TheCocktailDB API](#): An open, crowd-sourced database of drinks and cocktails worldwide.

### 4. Implementation

The main structure includes:

src/

```
|— apis
|   |— CocktailAPI.js: Helper functions to send requests and receive responses from API
|— App.vue
|— assets
|   |— cocktail.png
|   |— logo.png
|— components
|   |— AlcoholicFilter.vue: Drop-down menu to filter search results
|   |— AlcoholicSearch.vue: Buttons for the AlcoholicView
|   |— CategoryFilter.vue: Drop-down menu to filter search results
|   |— CategorySearch.vue: Buttons for the CategoryView
|   |— CocktailPreview.vue: Template for SearchView to pass in data and display
|   |— GlassFilter.vue: Drop-down menu to filter search results
|   |— GlassSearch.vue: Buttons for the GlassView
|   |— IngredientFilter.vue: Drop-down menu to filter search results
|   |— IngredientPreview.vue: Template for IngredientView to pass in data and display
|— main.js: Import necessary modules
|— router
|   |— index.js: Router
|— views
|   |— AboutView.vue: About page
|   |— AlcoholicView.vue: View to select alcoholic options
|   |— CategoryView.vue: View to select category options
|   |— CocktailDetail.vue: View detailed information of the selected cocktail
```

- GlassView.vue: View to select glassware options
- HomeView.vue: Home page
- IngredientDetail.vue: View detailed information of the selected ingredient
- IngredientView.vue: List all ingredients that are used in cocktails
- SearchView.vue: Display search results from other's views passed in parameters

## 5. Screenshots of the web app

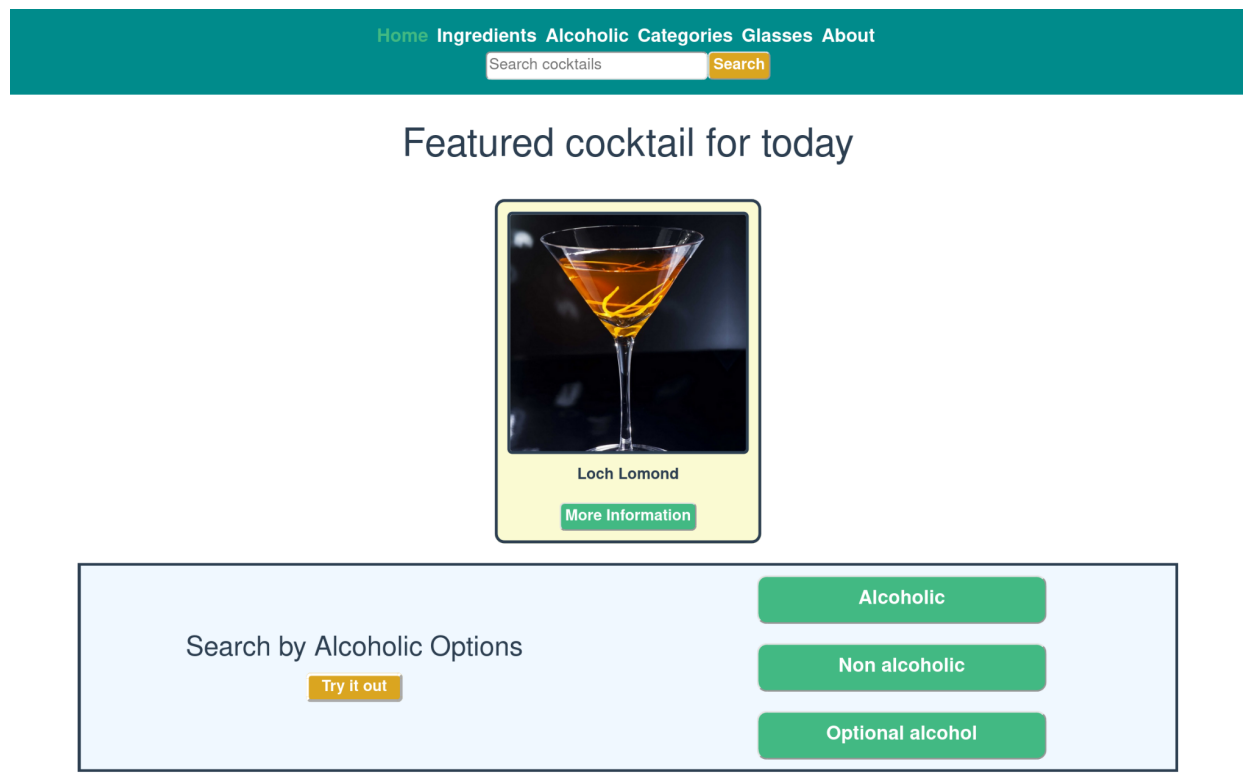


Figure 1: Home Page

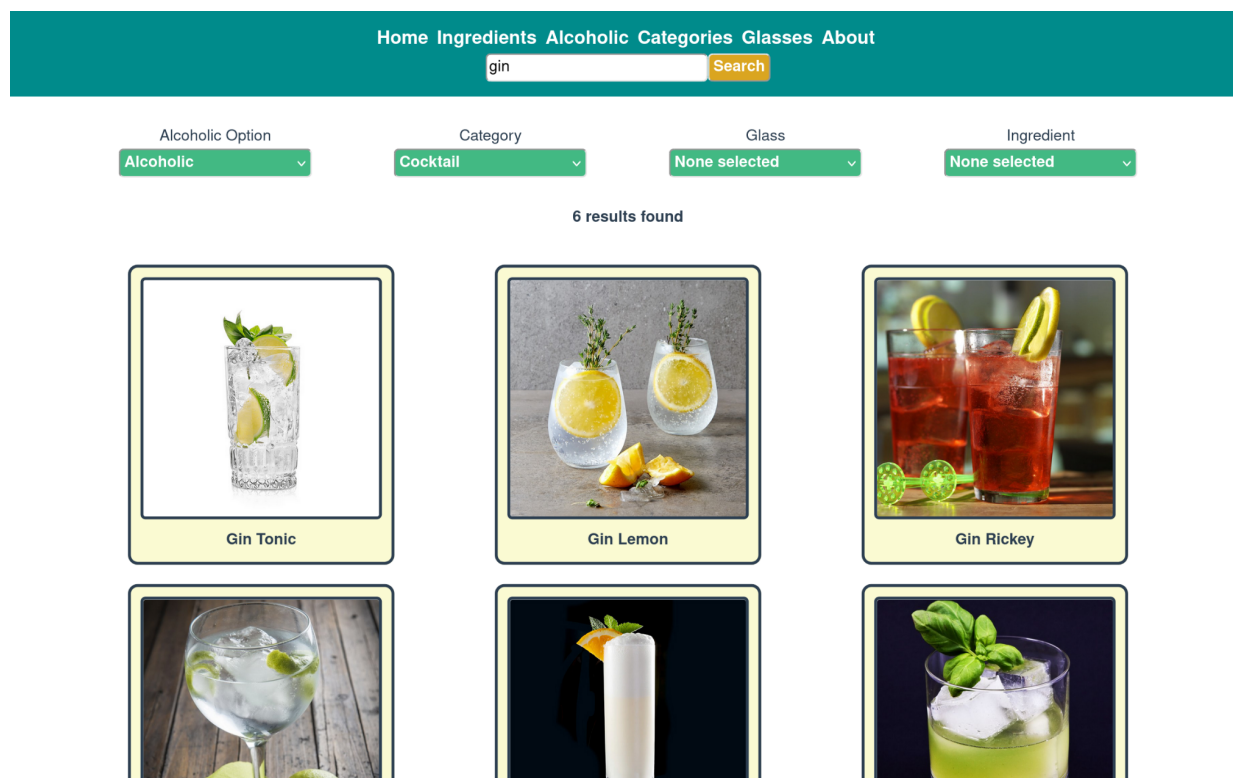


Figure 2: Search Page

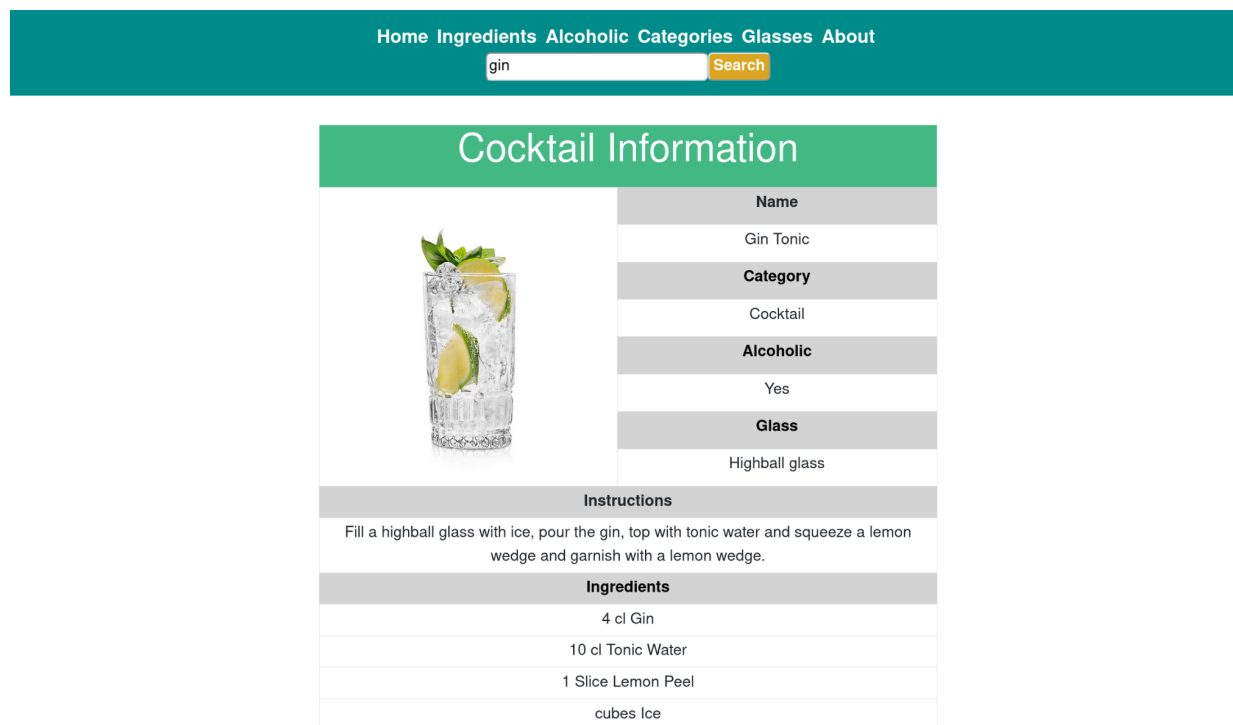


Figure 3: Cocktail Information Page

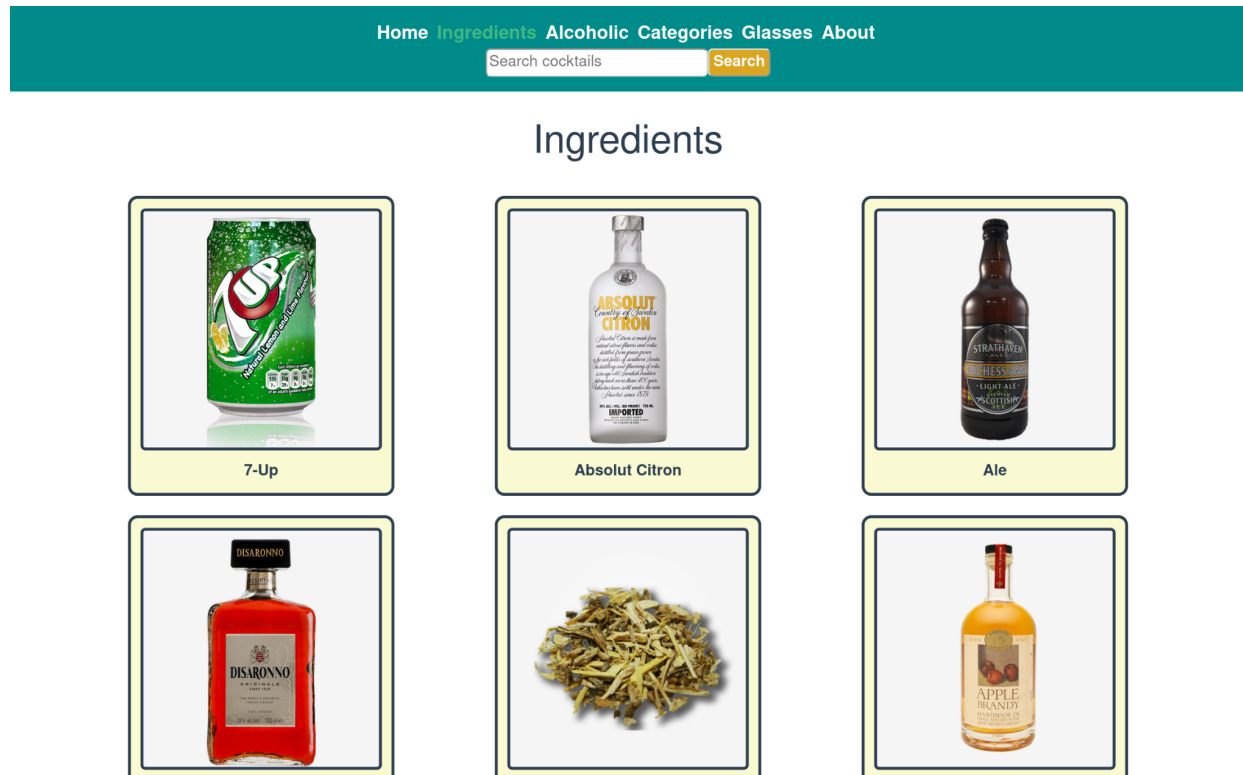


Figure 3: Ingredient Page

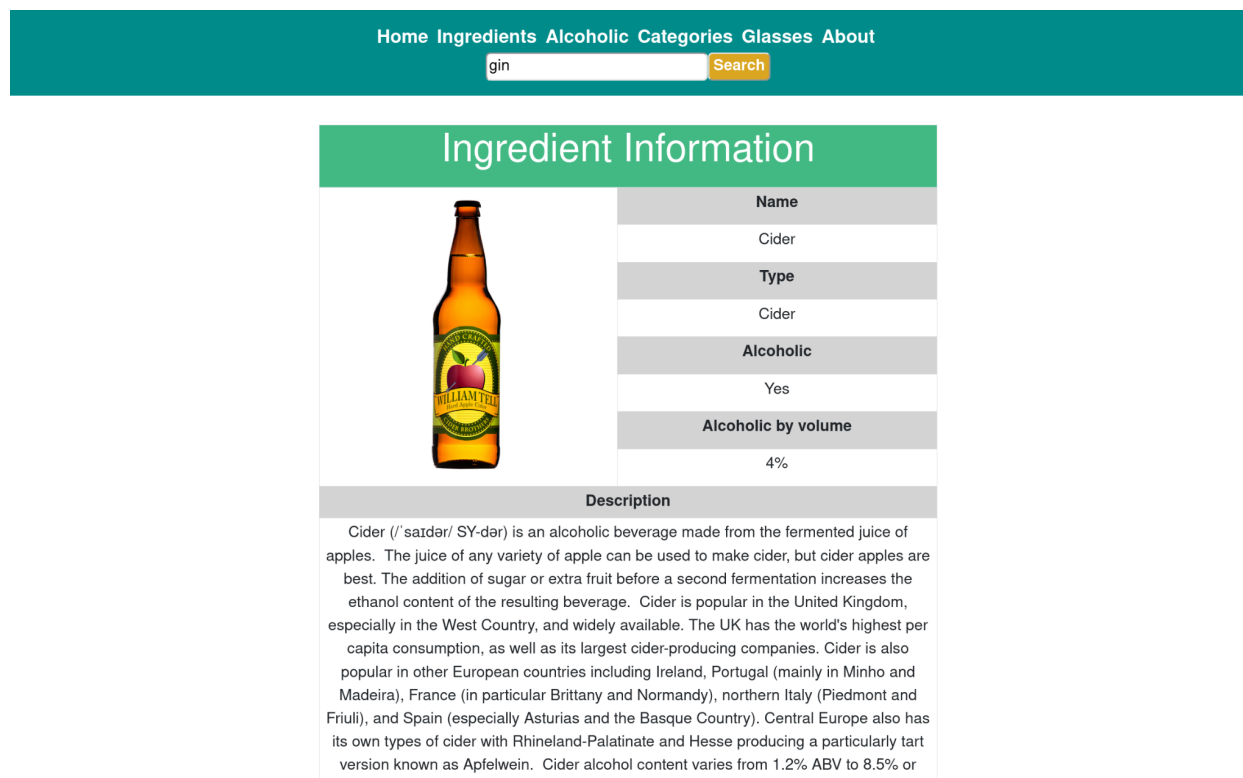


Figure 4: Ingredient Information Page

## Alcoholic Options

- Alcoholic
- Non alcoholic
- Optional alcohol

Figure 5: Alcoholic Option Page

## Glasses

- Balloon Glass
- Beer Glass
- Beer mug
- Beer pilsner
- Brandy snifter
- Champagne flute
- Cocktail glass
- Coffee mug
- Collins glass

Figure 6: Glass Page

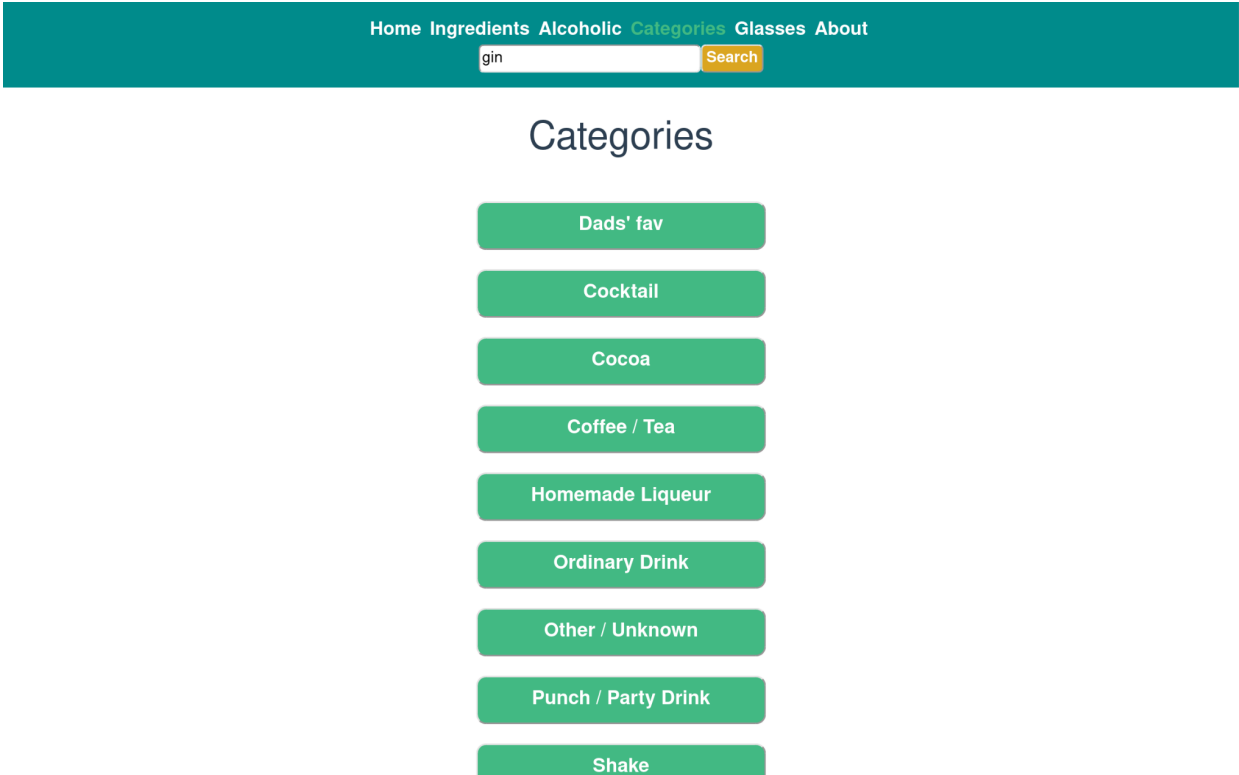


Figure 7: Category Page