

TasteTrek: Explore, Discover, and Create Delicious Recipes

Nikolina Petrovic, Stefan Teofilovski, Davor Gjurcinovski

Introduction

Welcome to TasteTrek, your ultimate companion for culinary exploration and discovery. TasteTrek is not just a recipe app; it's a culinary journey designed to ignite your passion for cooking, inspire your creativity in the kitchen, and elevate your dining experiences to new heights. With a rich array of features, intuitive user interface, and seamless integration of cutting-edge technologies, TasteTrek empowers food enthusiasts of all levels to embark on a flavorful adventure like never before.

Features Overview

TasteTrek offers a comprehensive suite of features to cater to your every culinary need:

1. **Recipe Search and Filtering:** Effortlessly search for recipes based on your preferences, including ingredients, calories, proteins, and food type. With advanced filtering options, you can find the perfect recipe to suit your taste buds and dietary requirements.
2. **Favorite Recipes:** Save your most loved recipes for quick access and convenient meal planning. TasteTrek provides a dedicated screen accessible from the footer, allowing you to manage and organize your favorite dishes with ease. Whether it's Grandma's secret lasagna recipe or a trendy new vegan dish, your favorites are just a tap away.
3. **Edit Recipe:** Tailor recipes to your taste preferences by editing ingredients, instructions, or any other details. TasteTrek empowers users to customize recipes according to their unique culinary preferences and dietary restrictions. Whether you want to add a personal touch to a classic recipe or adapt it to accommodate allergies or dietary preferences, TasteTrek makes it easy and convenient.
4. **Add Recipe with Camera:** Capture and add recipes using your phone's camera functionality. Simply snap a photo of your culinary creation or choose a photo from your gallery, fill out the recipe form, and boom the recipe is saved to our app.
5. **Location-Based Recommendations:** Utilize location services to discover recipes popular in your area or find local ingredients for your next culinary masterpiece. TasteTrek leverages geolocation data to provide personalized recipe recommendations based on your location, ensuring that you can always find inspiration wherever you are.
6. **User Authentication:** Securely log in and register users using a Node.js web service. TasteTrek ensures the privacy and security of user accounts with robust authentication mechanisms, protecting sensitive information from unauthorized access. With encrypted authentication tokens and stringent security measures, you can trust TasteTrek to keep your data safe and secure.
7. **Stateful Widgets:** Implement stateful widgets throughout the app to manage and update the user interface dynamically. By utilizing stateful widgets, TasteTrek delivers a responsive and

interactive user experience, enhancing usability and engagement. Whether it's updating the recipe list in real-time or displaying interactive elements based on user actions, TasteTrek's stateful widgets ensure a seamless and immersive user experience.

8. **Custom UI Elements:** Enjoy a visually appealing and intuitive user interface with custom-designed UI elements. TasteTrek's interface is meticulously crafted to provide a seamless and delightful user experience, enhancing usability and aesthetic appeal. From elegant navigation bars to interactive buttons and stunning imagery, every aspect of TasteTrek's UI is designed to delight and inspire.

Design Patterns and Principles

TasteTrek adheres to industry-standard design patterns and principles to ensure scalability, maintainability, and performance:

- **Model-View-Controller (MVC):** The app architecture follows the MVC pattern, separating concerns into models, views, and controllers for better organization and maintainability. By adhering to MVC principles, TasteTrek achieves a clear separation of concerns and promotes code reusability and modularity.
- **Single Responsibility Principle (SRP):** Each class and component within TasteTrek adheres to the SRP, ensuring that each component has a single responsibility and is focused on performing one task effectively. By following the SRP, TasteTrek enhances code clarity, maintainability, and testability.
- **Dependency Injection:** TasteTrek utilizes dependency injection to decouple dependencies and improve testability and modularity. By decoupling components and providing dependencies externally, TasteTrek achieves greater flexibility, scalability, and maintainability in its codebase.

Technologies Used

TasteTrek leverages a variety of technologies and services to deliver a seamless and immersive user experience:

- **Flutter:** The app is built using Flutter, Google's UI toolkit for building natively compiled applications for mobile, web, and desktop platforms from a single codebase. Flutter enables TasteTrek to deliver a smooth, responsive, and visually appealing user interface across multiple devices and platforms.
- **Node.js:** TasteTrek's backend services are powered by Node.js, providing a scalable, efficient, and versatile platform for building web services and APIs. Node.js enables TasteTrek to handle user authentication, recipe management, and data processing tasks seamlessly, ensuring high performance and reliability.
- **HTTP Requests:** TasteTrek communicates with the Node.js backend using HTTP requests to perform various actions, including user authentication, recipe retrieval, and data manipulation. By utilizing HTTP requests, TasteTrek establishes a secure and efficient communication channel between the frontend and backend components, enabling seamless data exchange and interaction.

- **Flutter Secure Storage:** TasteTrek employs Flutter Secure Storage to securely store authentication tokens and sensitive user data, ensuring data privacy and security. Flutter Secure Storage provides a robust and reliable solution for storing encrypted data locally on the device, protecting user information from unauthorized access and tampering.

Conclusion

TasteTrek represents the pinnacle of culinary innovation and exploration, offering a comprehensive suite of features to delight and inspire food enthusiasts worldwide. Whether you're seeking new recipes, organizing your favorites, or creating culinary masterpieces, TasteTrek is your indispensable companion on the journey to culinary excellence. Download TasteTrek today and embark on a flavorful adventure like never before!

For more information and updates, visit our GitHub repository to access the complete source code and documentation.

Note: *TasteTrek is continuously evolving, with new features and updates being added regularly to enhance your cooking experience. Stay tuned for exciting new developments and culinary discoveries!*

Widgets and Screens Overview

LoginScreen Widget

- **Description:** The **LoginScreen** widget presents the user with a login form where they can enter their username and password to access the app.
- **Backend API Calls:** The login functionality interacts with the Node.js backend API to authenticate users. It makes a POST request to the **/api/users/login** endpoint with the user's credentials.
- **Additional Features:** Includes a link to the registration screen for new users who haven't registered yet.

RegisterScreen Widget

- **Description:** The **RegisterScreen** widget allows new users to create an account by providing their personal information, including first name, last name, email, username, and password.
- **Backend API Calls:** Registers users through a POST request to the **/api/users/register** endpoint of the Node.js backend. The user's information is sent as JSON data in the request body.
- **Additional Features:** Includes validation for form fields and redirects users to the login screen upon successful registration.

RecipesScreen Widget

- **Description:** The **RecipesScreen** widget serves as the home screen of the app, displaying a grid of recipes that users can browse and explore.

- **Backend API Calls:** Fetches recipes from the backend API using a GET request to the `/api/recipes/getAllRecipes` endpoint.
- **Additional Features:** Supports filtering recipes based on user preferences, such as calories, proteins, and food type. Uses frontend filtering for categories and recipe names.

ManageProfileScreen Widget

- **Description:** The **ManageProfileScreen** widget allows users to view and update their profile information, including first name, last name, email, username, and password.
- **Backend API Calls:** Similar to the registration screen, updates user information through a POST request to the `/api/users/updateProfile` endpoint of the backend API.
- **Additional Features:** Provides validation for form fields and ensures data integrity when updating user profiles.

Header Widget

- **Description:** The **Header** widget represents the app's header, containing the app logo and navigation options.
- **Functionality:** Implements user authentication controls, such as toggling favorites and accessing user profile options.
- **Backend API Calls:** No direct backend API calls; however, it interacts with authentication-related endpoints for user logout functionality.

Footer Widget

- **Description:** The **Footer** widget serves as the app's footer, providing quick access to essential navigation options.
- **Functionality:** Includes buttons for navigating to the home page, favorites page, adding a new recipe, logging out, and accessing the user profile.
- **Backend API Calls:** Executes backend API calls for user logout functionality through the logout button.

AddRecipeWidget

- **Description:** The **AddRecipeWidget** allows users to create and submit new recipes to the app's database. It includes fields for entering recipe details such as title, description, ingredients, instructions, and an option to upload a photo of the recipe.
- **Backend API Calls:** Performs a POST request to the `/api/recipes/addRecipe` endpoint on the backend to add the new recipe to the database. This request includes the recipe details and image data, if provided.

EditRecipeWidget

- **Description:** The **EditRecipeWidget** enables users to modify existing recipes stored in the app. It presents a form similar to the **AddRecipeWidget** with pre-filled fields containing the current recipe details.
- **Backend API Calls:** Utilizes a POST request to the **/api/recipes/editRecipe** endpoint on the backend to update the recipe information in the database. This request includes the modified recipe details.

ViewRecipeWidget

- **Description:** The **ViewRecipeWidget** displays detailed information about a specific recipe, including its title, description, ingredients, instructions, and an accompanying photo.
- **Backend API Calls:** Fetches the recipe details from the backend using a GET request to the **/api/recipes/getRecipeById** endpoint, passing the recipe ID as a parameter.

QuestionsWidget

- **Description:** The **QuestionsWidget** presents users with a series of questions to refine their recipe search criteria. It allows users to specify preferences such as desired calorie range, protein content, food type, and any dietary restrictions.
- **Implementation:** Facilitates frontend filtering of recipe search results based on the user's responses to the questions. The widget dynamically adjusts the displayed recipes according to the selected criteria, providing a tailored browsing experience.

Utilized Services and Functionalities

Location Services

- **Functionality:** Utilizes device location services to provide location-based recommendations for recipes and ingredients.
- **Implementation:** Accesses device location data to customize recipe suggestions based on the user's geographic location.

Camera Service

- **Functionality:** Allows users to add recipe images by capturing photos using the device's camera.
- **Implementation:** Integrates with the device's camera functionality to capture and upload images directly within the app.

Camera Roll Service

- **Functionality:** Enables users to select images from their device's photo gallery to add to recipe entries.
- **Implementation:** Allows users to browse and select images stored in their device's photo gallery for use within the app.

Backend API Calls

- **Endpoints:** Utilizes various endpoints on the Node.js backend for user authentication (login, register, logout), recipe management (add, edit, get all recipes), and user profile management (update profile).
- **HTTP Methods:** Uses HTTP POST and GET methods to interact with backend endpoints, passing relevant data such as user credentials, recipe details, and profile information.

Filter Functionality

- **Description:** Provides users with the ability to filter recipes based on specific criteria, including calories, proteins, and food type.
- **Implementation:** Employs frontend filtering mechanisms to refine recipe search results, enhancing user experience and usability.