Recipe Sharing API

WITH JJE

Presentation by JJE

Recipe Sharing API

The Recipe Sharing API Project offers a streamlined platform for sharing recipes via a RESTful API. Users can easily create, discover, and manage recipes with features like CRUD operations and authentication. Leveraging modern technologies ensures efficiency and reliability.

We chose Node.js and Express.js to build our recipe API. Our API follows the RESTful way, which means we use standard rules for how our web addresses work and how to talk to our server. We set up different paths for different actions you can do with our recipes, like getting all the recipes or adding a new one. We check the data you send us to make sure it's good and safe using Joi, and if something goes wrong, we tell you so you know what's happening. Our API can listen on different doors, either the one you tell us or a default one if you don't specify. We look for recipes even if you don't type perfectly, making searching easier for you. In short, we've made our recipe API simple, safe, and helpful, ensuring clear communication and ease of use.

Project Purpose

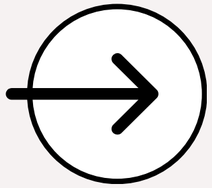
↘ PURPOSE

We're developing an online platform with an API where users can effortlessly share and access recipes, transforming cooking and meal planning into a collaborative and enjoyable experience.

↘ TARGET

Our platform is for people who love cooking, and sharing recipes. It's a place where everyone can share recipes and discover new ones, making cooking more fun for everyone.

Key Features



CRUD OPERATIONS

Users can Create, Read, Update, and Delete recipes.

SEARCH FUNCTIONALITY

Enable users to search for recipes by title and ID.

Technical Implementation

↘ DEPENDENCIES

- express: A Node.js web application framework used for creating RESTful APIs.
- joi: A schema description language and data validator for JavaScript.

↘ DATA

typeOfRecipes:
An array containing objects representing different recipes. Each recipe object has properties like id, title, ingredients, and instructions.

↘ ROUTES HANDLERS

- Each route has a corresponding handler function that performs the required operations.
- For example, to add a new recipe, it validates the incoming request body using Joi, then creates a new recipe object and adds it to the typeOfRecipes array.

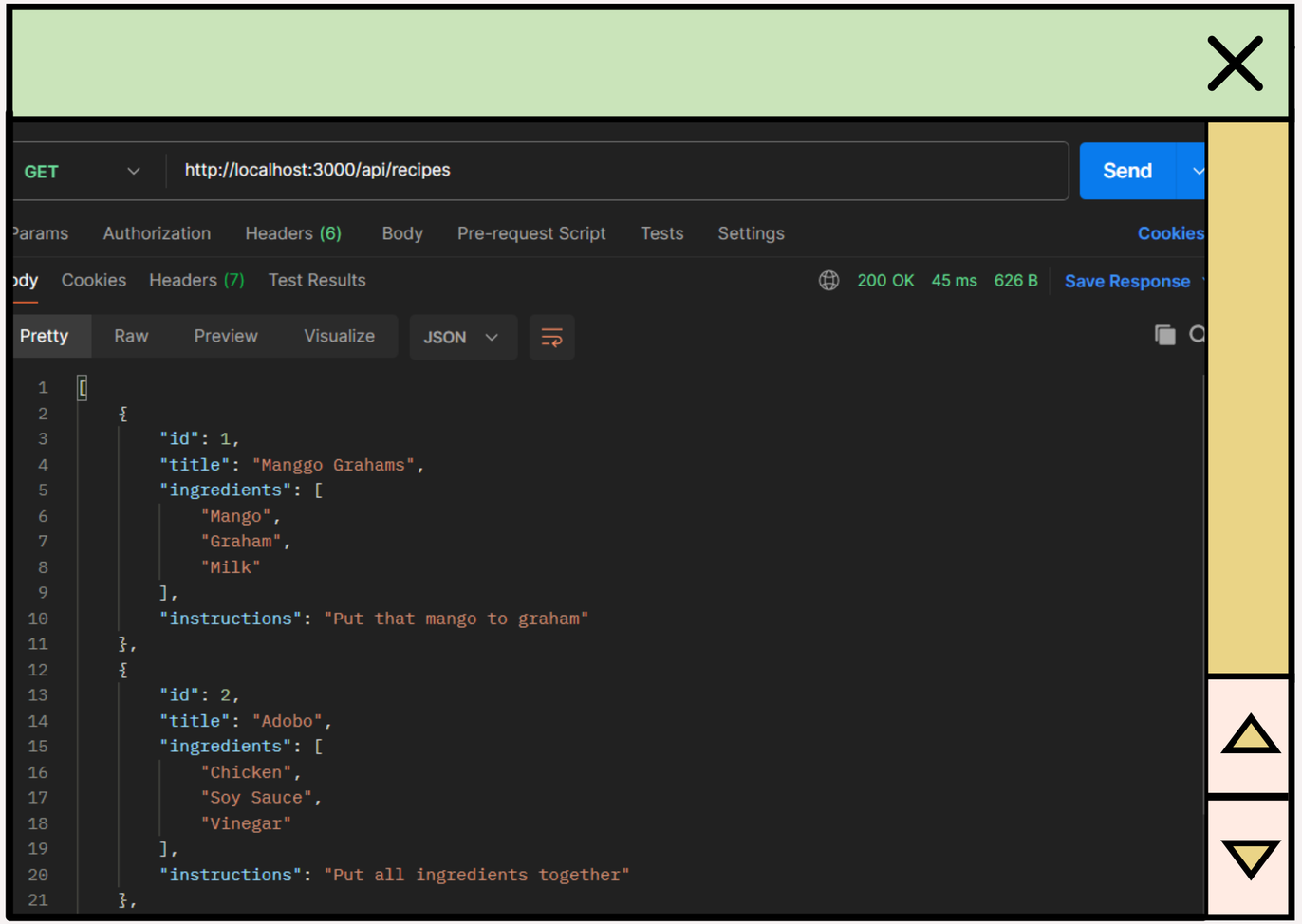
↘ VALIDATION

- Basic validation is implemented using conditional statements.
- For example, in the POST endpoint, it checks if the title is provided and has a minimum length of 3 characters.

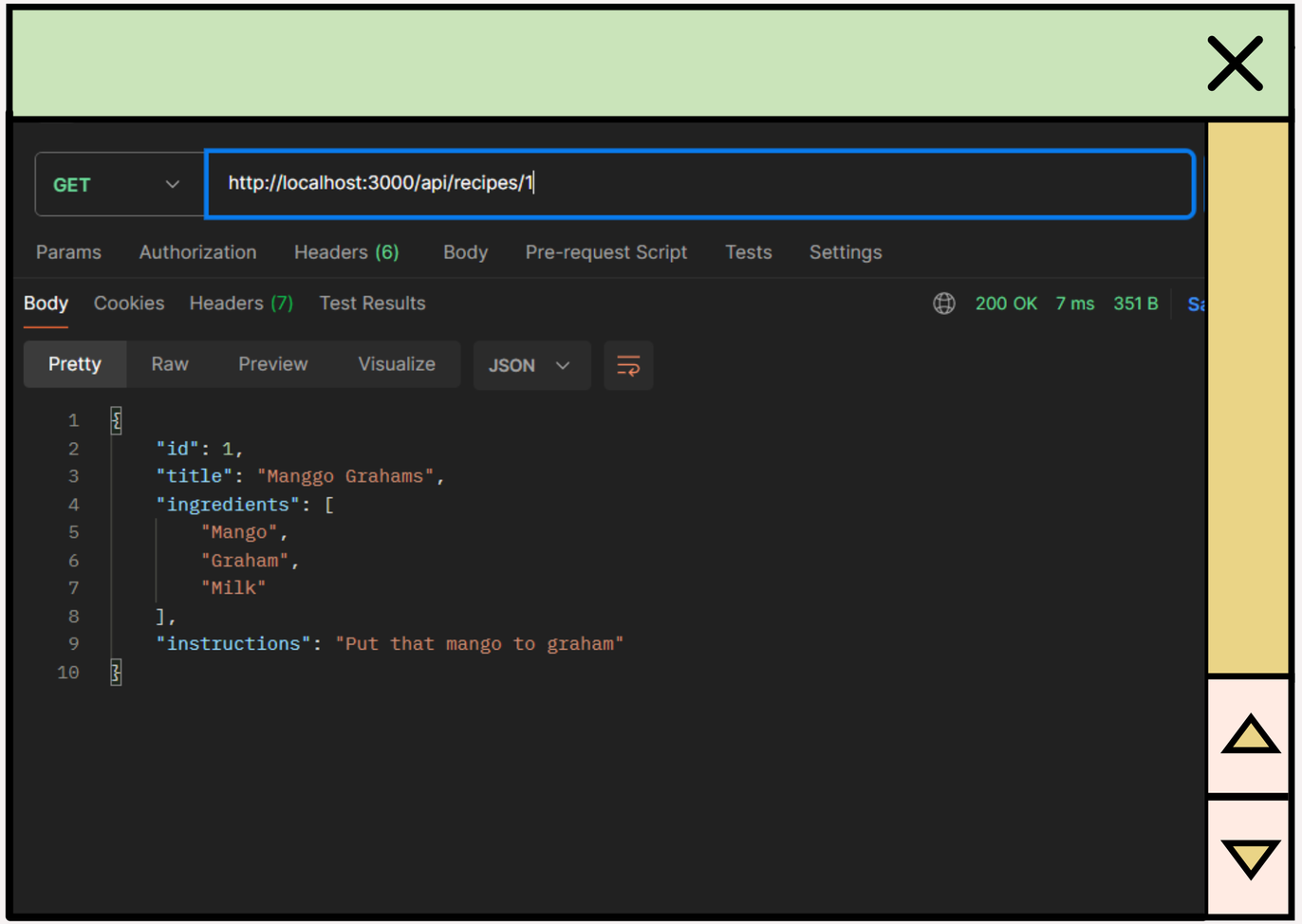
↘ SERVER INITIALIZATION

- Sets the port number for the server (either from environment variable or defaults to 3000).
- Starts the server to listen on the specified port.
- Outputs a message to the console indicating the server is running.

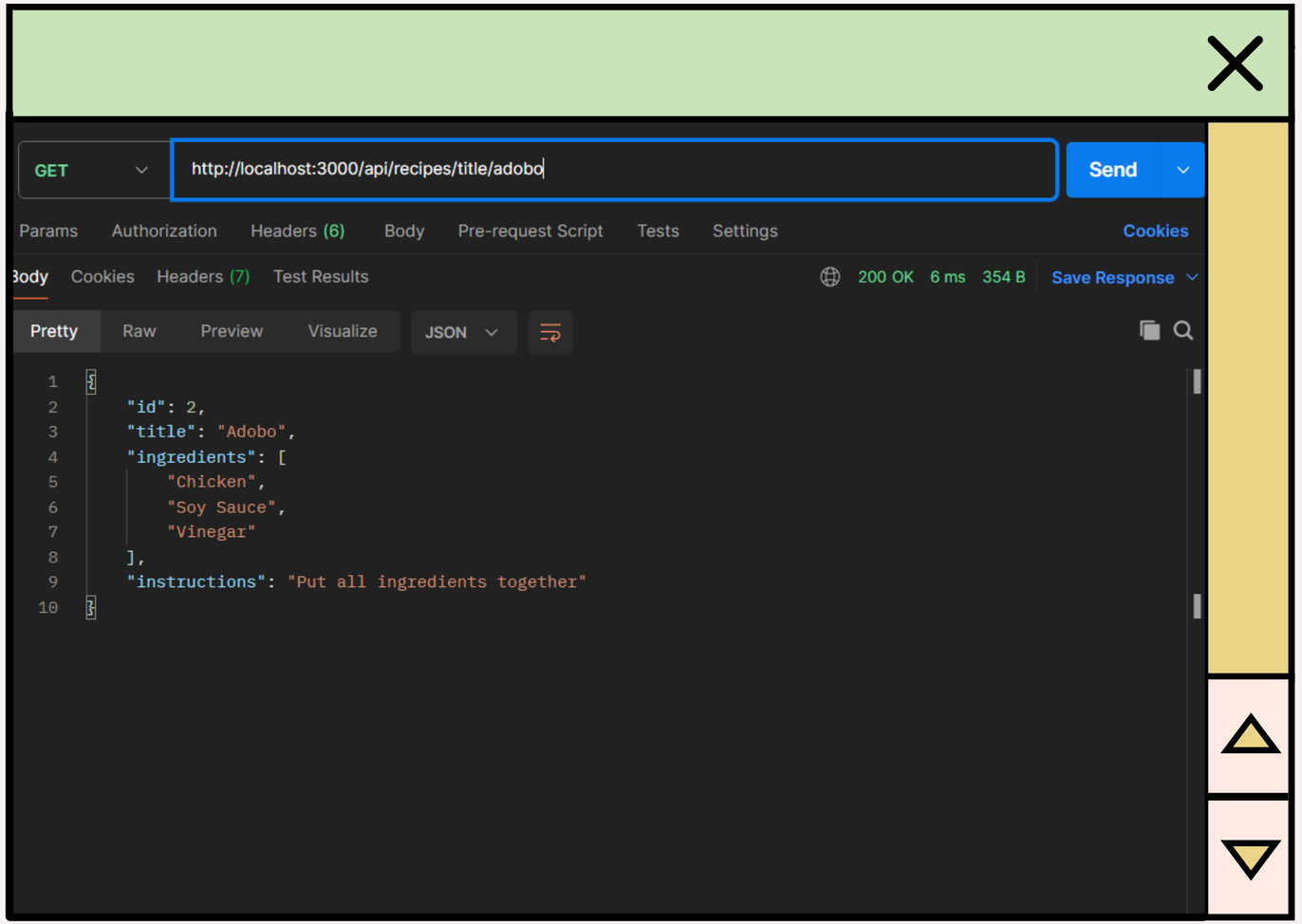
Get all recipes



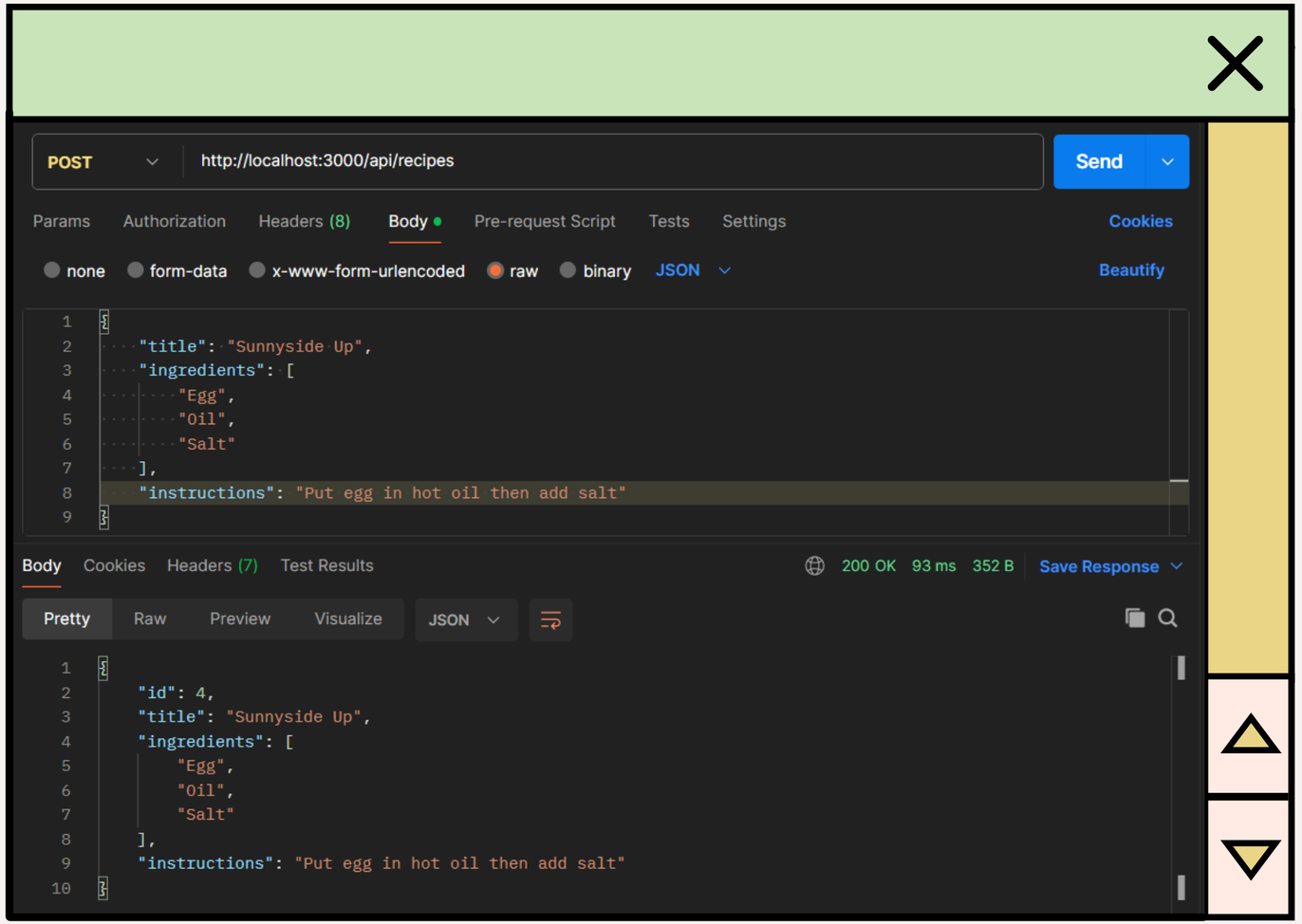
Get recipe by ID



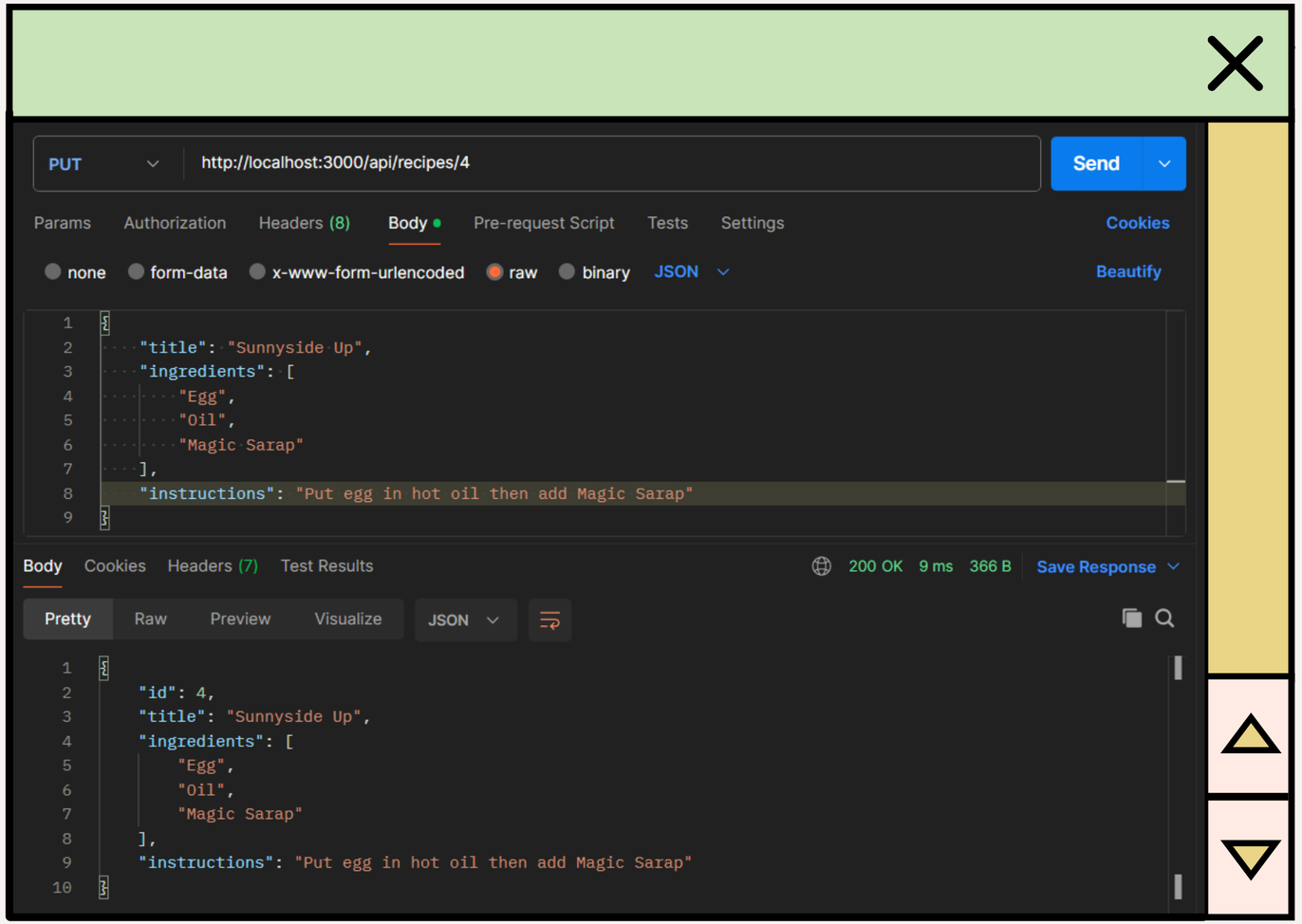
Get recipe by title



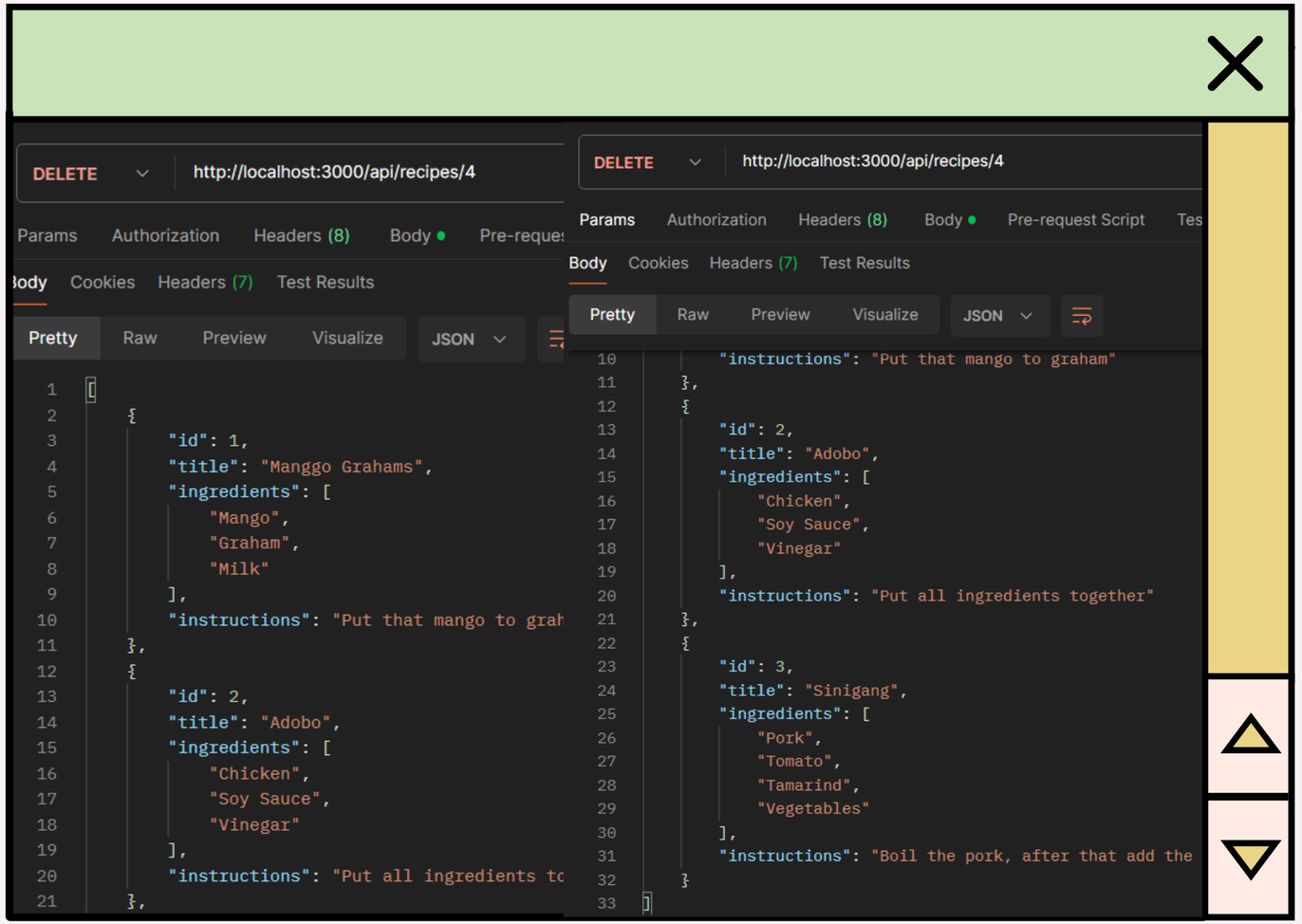
Add new recipe



Edit a recipe



Delete a recipe



Unique Features

CASE-INSENSITIVE SEARCH

Our Recipe Sharing API offers a unique case-insensitive search feature, ensuring a smoother and more user-friendly experience by allowing users to search for recipes without worrying about capitalization.

ERROR HANDLING

Our Recipe Sharing API prioritizes user experience by implementing robust error handling, which provides clear and informative messages to users in case of errors, aiding them in understanding and resolving issues with their requests, thus ensuring a smoother and less frustrating experience overall.

SIMPLE DATA VALIDATION

In our Recipe Sharing API, we use a tool called Joi to check that the information you send us is right, which helps keep everything safe and accurate, so you can trust the recipes you find and share.