

# Django Recipe App — Portfolio Case Study

*A full-stack recipe management platform built with Django.*

## 🌟 Project Overview

 **Live Demo:** <https://django-recipe-app-6t2d.onrender.com/>

### Tech Stack:

-  Python
-  Django
-  HTML, CSS, JavaScript
-  REST API
-  PostgreSQL
-  GitHub Actions (CI/CD)
-  Render (Deployment)

The Django Recipe App is a full-stack web application that allows users to securely create, manage, and organize their personal recipes. It demonstrates my ability to build a complete application from backend logic to frontend UI and deployment.

## 🧩 Problem Statement

Many people store recipes across notebooks, screenshots, and random notes. This makes it difficult to organize and retrieve them.

The goal was to build a **centralized, secure, and user-friendly** platform where users can:

-  Save recipes
-  Edit or delete them
-  Search quickly
-  Keep their data private
-  Access from any device

## My Role

I worked as the **Full-Stack Developer**, responsible for:

-  Designing database models
-  Implementing authentication
-  Building CRUD functionality
-  Creating responsive UI with HTML/CSS
-  Testing and debugging
-  Deploying using Render + GitHub Actions

## ★ Key Features

### User Authentication

- Secure login & registration
- Password hashing
- User-specific recipe storage

### Recipe Management (CRUD)

- Add, edit, delete recipes
- View detailed recipe pages
- Organized recipe dashboard

### Search Functionality

- Search recipes by name
- Fast and intuitive filtering

### Responsive UI

- Built with custom HTML & CSS
- Clean, minimal, mobile-friendly design

## ⚙️ Admin Panel

- Manage users
- Manage recipes
- View database entries

## ⚙️ Technical Implementation

### Backend (Django)

- Django MVC architecture
- Models for recipes & user relationships
- Views for CRUD operations
- REST API endpoints for recipe data
- Form validation & error handling

### Frontend

- HTML templates with Django Template Language
- Custom CSS styling
- JavaScript for interactivity

### Database

- PostgreSQL
- Relational structure with foreign keys
- Secure user-recipe mapping

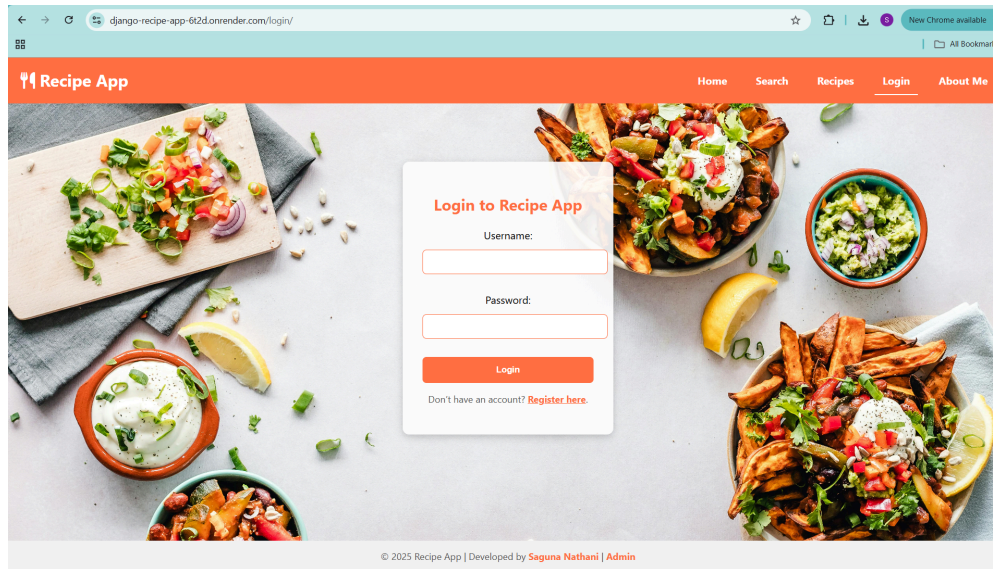
### CI/CD & Deployment

- GitHub Actions for automated testing & deployment
- Render for hosting
- Environment variables for security
- WhiteNoise for static file handling

## Screenshots

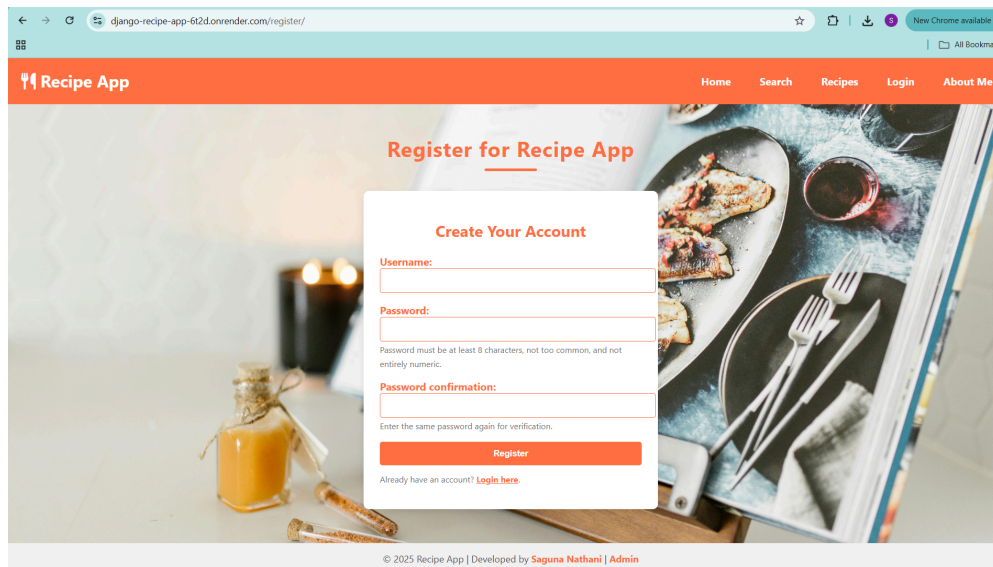
### Login View

A clean login page allowing users to securely access their recipe dashboard.



### Register View

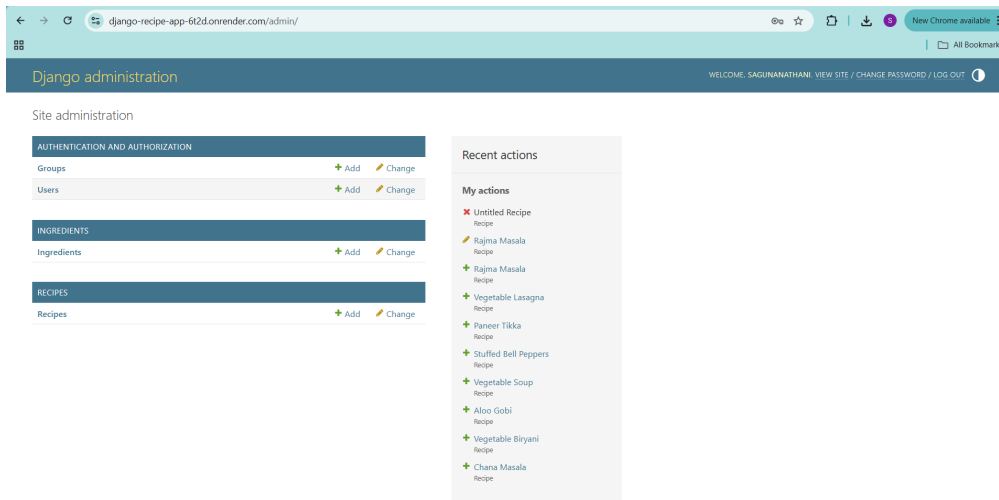
Simple registration form for new users.





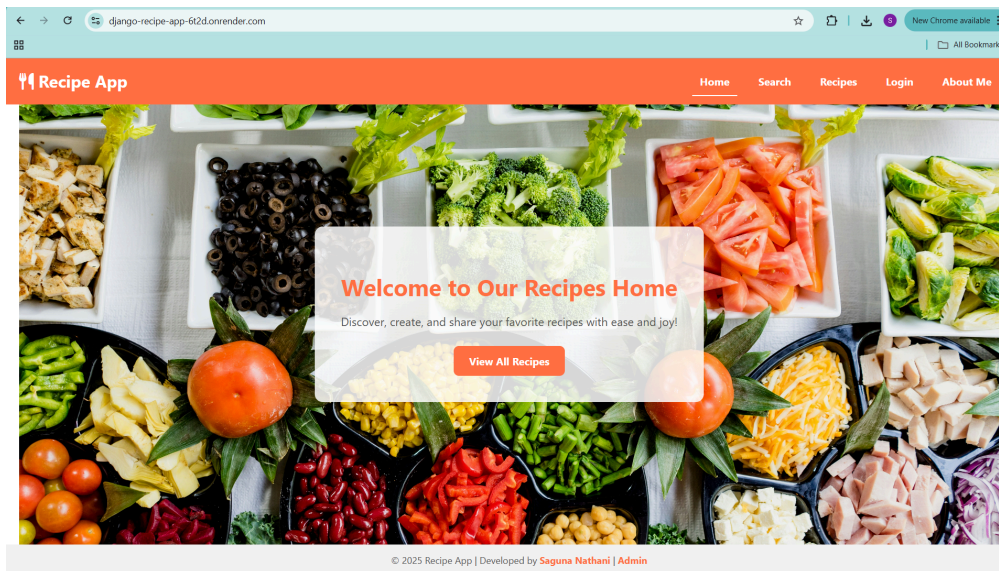
## ⚙️ Admin Panel

Django admin interface for managing users and recipes.



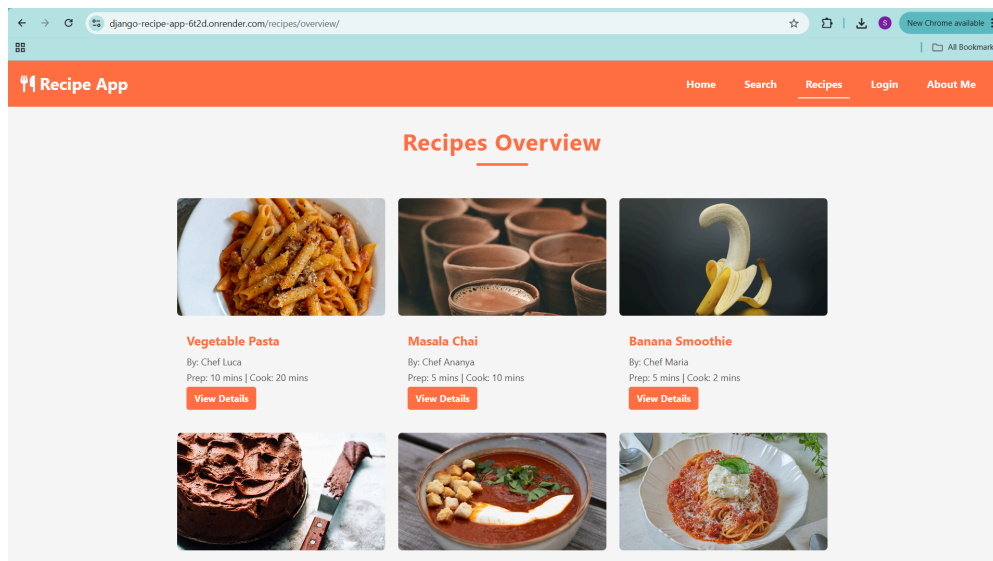
## 🏠 Home View

Landing page after login with navigation options.



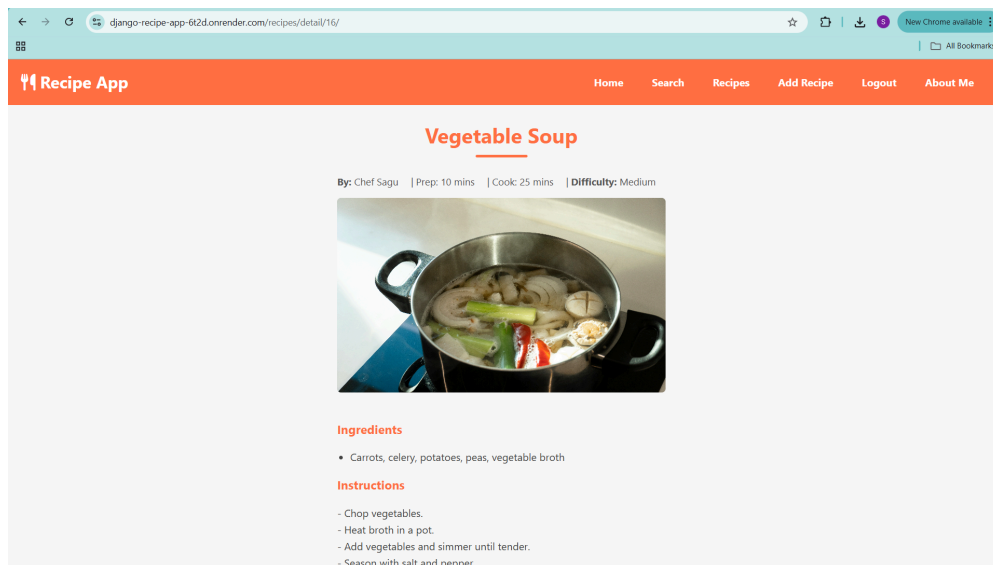
## Recipe Overview (All Recipes)

Displays all recipes created by the logged-in user.



## Recipe Details View

Shows full recipe details including ingredients and instructions.



7

## + Add Recipe View

Form for adding new recipes with validation.

The screenshot shows a web browser at the URL `django-recipe-app-6t2d.onrender.com/recipes/add/`. The page has an orange header with the logo 'Recipe App' and navigation links: Home, Search, Recipes, Add Recipe (active), Logout, and About Me. The main heading is 'Add a New Recipe'. The form contains the following fields:

- Title:** Text input with placeholder 'Untitled Recipe'.
- Author Name:** Text input with placeholder 'Unknown'.
- Description:** Large text area.
- Prep Time (mins):** Text input with placeholder '0'.
- Cook Time (mins):** Text input with placeholder '0'.
- Serves:** Text input with placeholder '1'.
- Ingredients:** Text area with placeholder 'No ingredients listed'.
- Instructions:** Text area with placeholder 'No instructions yet'.

## 🔍 Search View

Search bar to filter recipes by name.

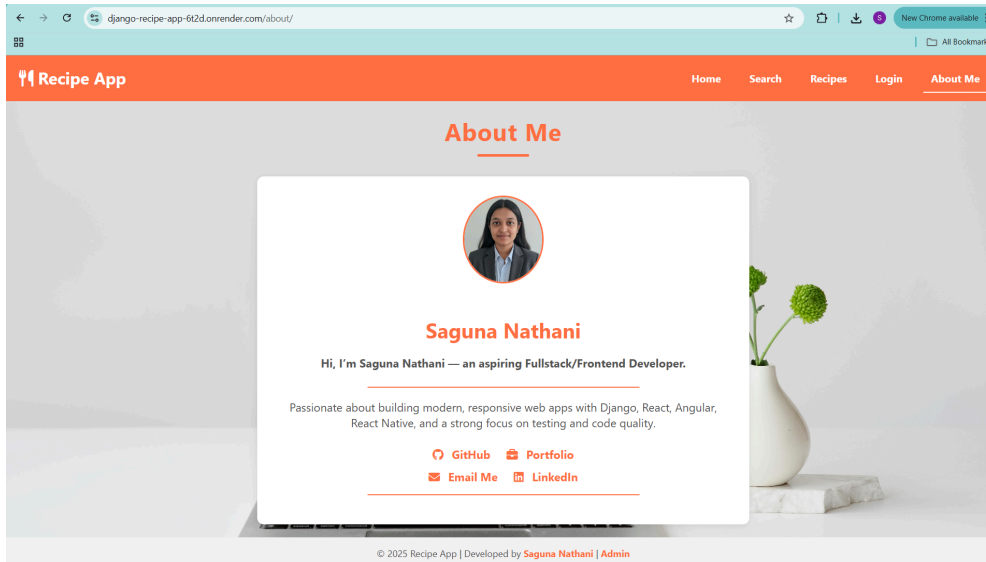
The screenshot shows a web browser at the URL `django-recipe-app-6t2d.onrender.com/recipes/search/?query=pasta&ingredients=8&max_time=`. The page has an orange header with the logo 'Recipe App' and navigation links: Home, Search (active), Recipes, Add Recipe, Logout, and About Me. The main heading is 'Search Your Perfect Recipes'. The search form includes:

- Recipe Name:** Text input with placeholder 'pasta'.
- Ingredients:** Text input with placeholder 'e.g., Milk, Tomatoes, Garlic, Sugar'.
- Max Cooking Time (minutes):** Text input with placeholder 'Maximum minutes'.
- Show all recipes:** Button with a magnifying glass icon.
- Search & Analyze:** Orange button.

Below the form, the section 'Search Results' is visible, showing three recipe images: a plate of pasta, a bowl of soup, and a glass of juice.

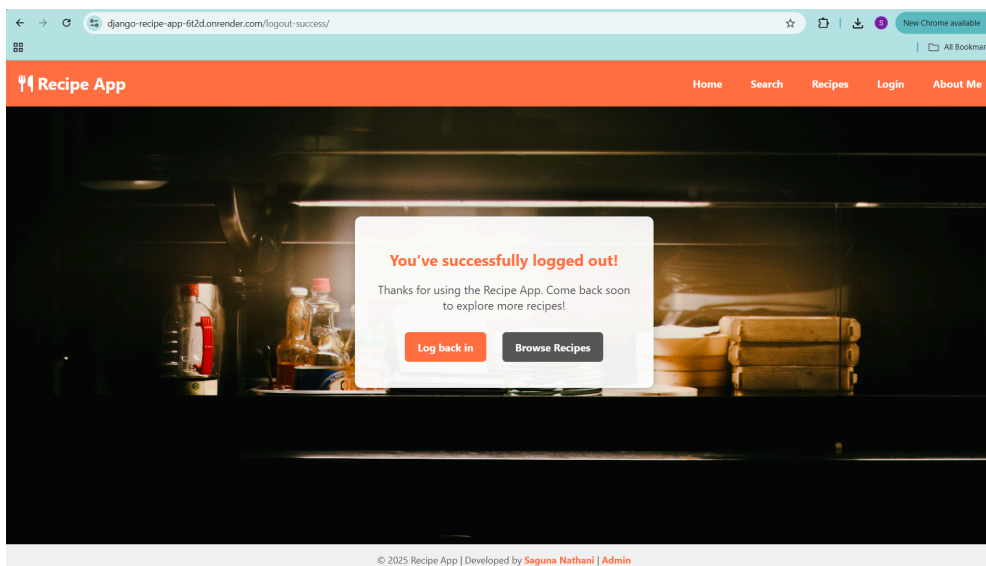
## About View

Displays information about the application, including its purpose, features, and how users can benefit from using the Recipe App.



## Logout View

Logs the user out and redirects to the login page.



## Challenges & Solutions

### ♦ Challenge: User-specific data isolation

**Solution:** Used Django's foreign key relationships and filtered queries by `request.user`.

### ♦ Challenge: Deployment static files issue

**Solution:** Configured WhiteNoise and updated production settings.

### ♦ Challenge: Clean UI without Bootstrap

**Solution:** Designed custom CSS components and layout structure.

## Outcome

The final application is:

- ✓ Fully functional
- ✓ Secure
- ✓ Responsive
- ✓ Easy to use
- ✓ Deployed with CI/CD

It demonstrates my ability to build a complete full-stack application independently.

## What I Learned

- Django ORM & model relationships
- REST API integration
- Deployment workflow with Render
- CI/CD automation using GitHub Actions
- Writing clean, maintainable code
- Improving UI/UX with custom CSS

## Future Enhancements

- Recipe categories & tags
- Image upload for recipes
- Public recipe sharing
- API endpoints for mobile apps

## Project Links

Live App: <https://django-recipe-app-6t2d.onrender.com/>

GitHub Repository: <https://github.com/sagunanathani/django-recipe-app>

## Wrapping Up

Thank you for taking the time to explore my Django Recipe App project. This application reflects my passion for building clean, functional, and user-focused web experiences using modern full-stack technologies. I look forward to creating more projects that combine thoughtful design, strong backend architecture, and smooth user interactions.