CAREERFOUNDRY

Achievement 2 Project Brief: Recipe App (Web Application Version)

Objective

In this project, you'll take your Recipe app from Achievement 1 and use the Django web framework to develop a fully fledged web application with multiple users and an admin panel.

Context

This project focuses on creating a web application using the Django framework. To work with Django, you need an understanding of application design patterns and internal language, which you gained in Achievement 1 when using Python to make a command line Recipe app.

Now, in Achievement 2, it's time for you to rebuild your app using Django. Django has the advantage of being neatly moduralized and developer friendly, while also being powerful enough to run some of the world's most popular websites.

In your project, working with Python-based Django, you'll develop a full-stack web application using the Django development server. You'll then deploy the application using Heroku, with a Postgres database at the backend, HTML, and CSS-based rendered pages at the frontend and Python-based Django as your web application framework.

Your final web application will be dynamic and multi-user, letting users sign up and create their own content. It'll also have statistical dashboards, implementing your new data analytics and data visualization skills. Finally, you'll demonstrate coding best practices by putting your well-tested and well-documented code on GitHub.

The 5 W's

Before you get started, it's crucial you know what you're creating, who for, and why. Here are the details.

1. **Who**? Your finished project will demonstrate your Python and Django skills to your professional network and future employers.

2. **What**? The project consists of you making a fully functional Recipe web application, which can create, read, and modify recipes, and search for recipes based on ingredients. The web application will include an admin panel for data handling and visualization.
3. **When**? You'll host the code for your web application on a GitHub repository so it's ready for viewing. Your application will be hosted on a public web server so that potential employers and professional network have access.
4. **Where**? Your Recipe application will be viewed from your GitHub repository. Viewers (i.e., potential employers and collaborators) will be able to play around with your live Recipe application.
5. **Why**? This project will help you demonstrate in professional settings 1) your understanding of the fundamentals of the Django web framework, and 2) your web application development skills using the Django web framework.

User Goals

Your users should be able to create and modify recipes containing ingredients, cooking time, and a difficulty parameter automatically calculated by the application. Your users should also be able to search for recipes by ingredient.

Key Features

To achieve the user goals, your application will need the following features:

- Allow for user authentication, login, and logout.
- Let users search for recipes according to ingredients.
- Automatically rate each recipe by difficulty level.
- Receive user input and handle errors appropriately.
- Display more details on each recipe if the user asks for that.
- Add user recipes to an SQLite database.
- Include a Django Admin dashboard for working with database entries.
- Show statistics and visualizations based on trends and data analysis.

Technical Requirements

Your finished application should meet the following technical requirements:

- Works on Python 3.6+ installations and Django version 3.
- Handles exceptions or errors that arise during user input, for example, then displays user-friendly error messages.
- Connects to a PostgreSQL database hosted locally on the same system (an SQLite database is needed during the development of your application).
- Provides an easy-to-use interface, supported by simple forms of input and concise, easy-to-follow instructions. Menus containing features like login and logout must be presented neatly—with concise and easy-to-follow prompts.
- Code with proper documentation and automated tests is uploaded on GitHub. A "requirements.txt" file is provided, containing the requisite modules for the project.
- Readme file is provided with instructions on downloading and running the app locally on any machine.

Nice to Have

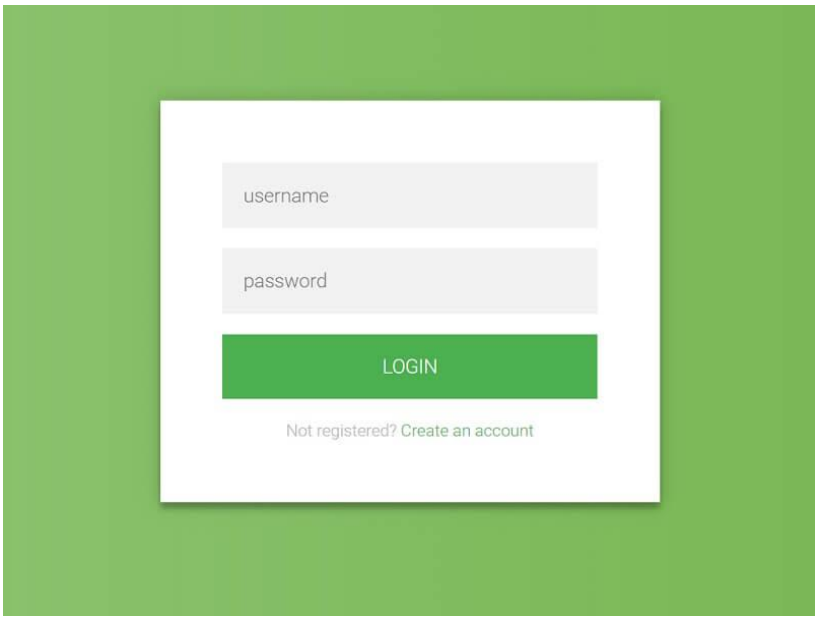
- Makes sure your application's efficiency is maintained when data is being accessed from the database. Your application will access the database to search for recipes and extract data for visualization. However, redundant or repeated database access should be avoided; it can slow down your application. The solution in this case is to have data temporarily stored or memorized by the program for later use. However, if there are large amounts of data, it's advisable that your application doesn't access all of the database data at once. You need to work out how to access data from your database so that your application remains efficient—designing efficient ORM queries is key to this process.

Mockups or Other Assets

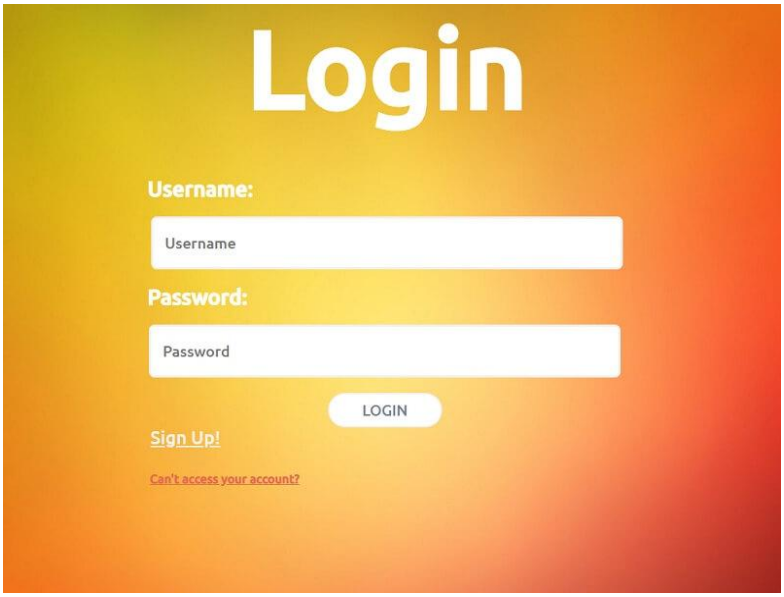
To guide you in your project, here are some mockups showing how to communicate with users in your interface—be it when they're logging in or adding new recipes.

Tip! If you have a UI designer friend, we strongly encourage you to get advice from on the user interface of your Recipe web application. This will not only help you create the best UI possible; it will also give you valuable experience in collaborating with designers, which you'll need to do in future workplaces.

Signup/Login Interface

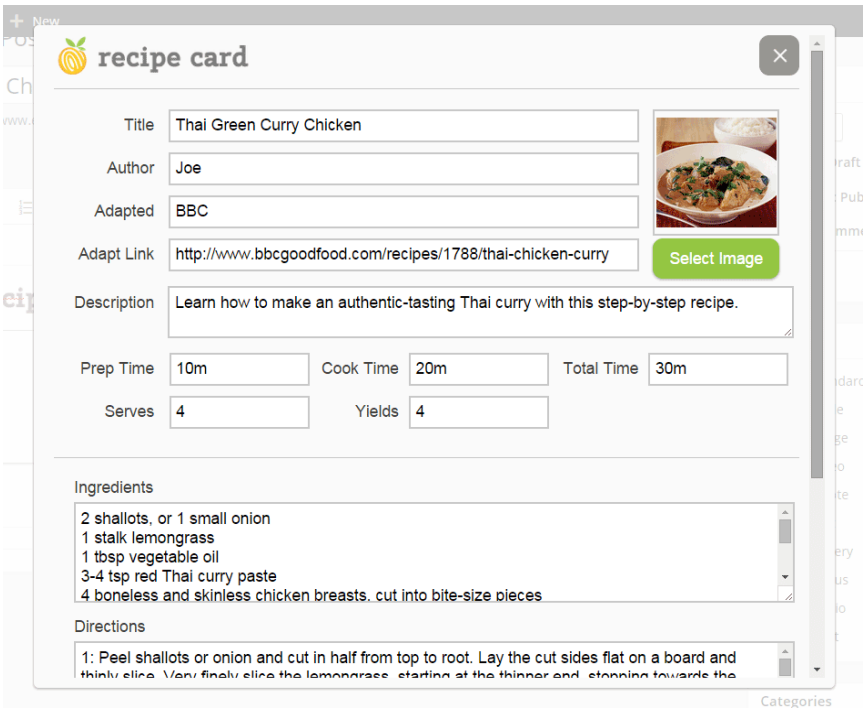


Example 1 (Source: Digital Template Market)



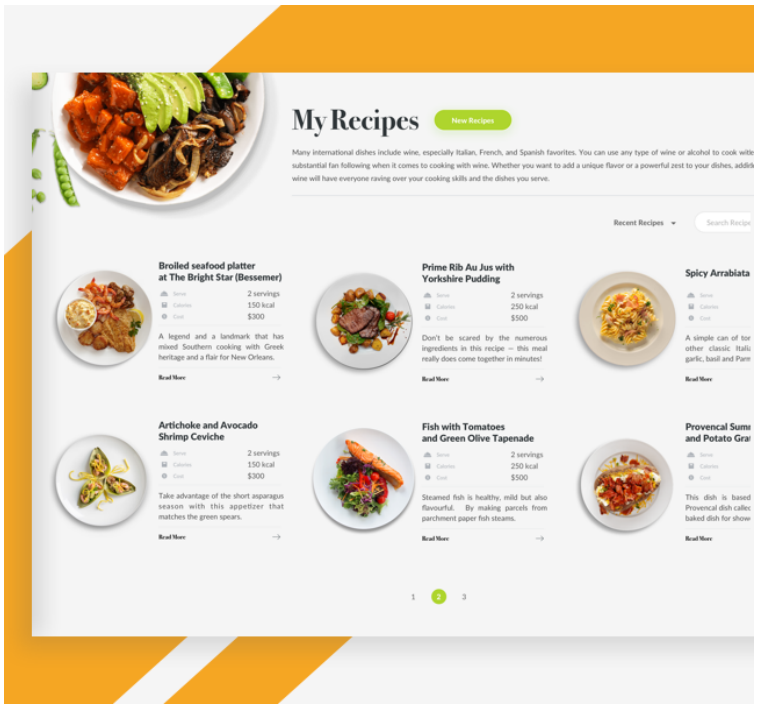
Example 2 (Source: Digital Template Market)

Adding Recipe Card Interface

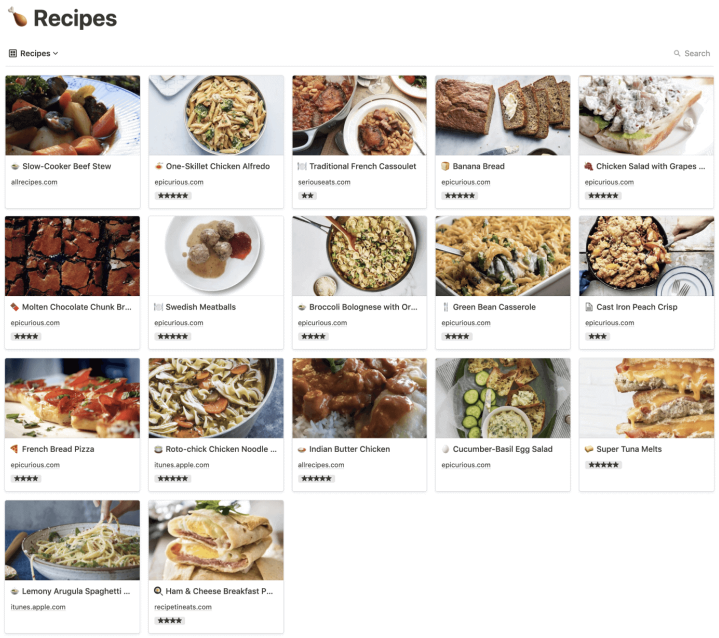


Example 3 (Source: Pagely)

List of Recipes Interface

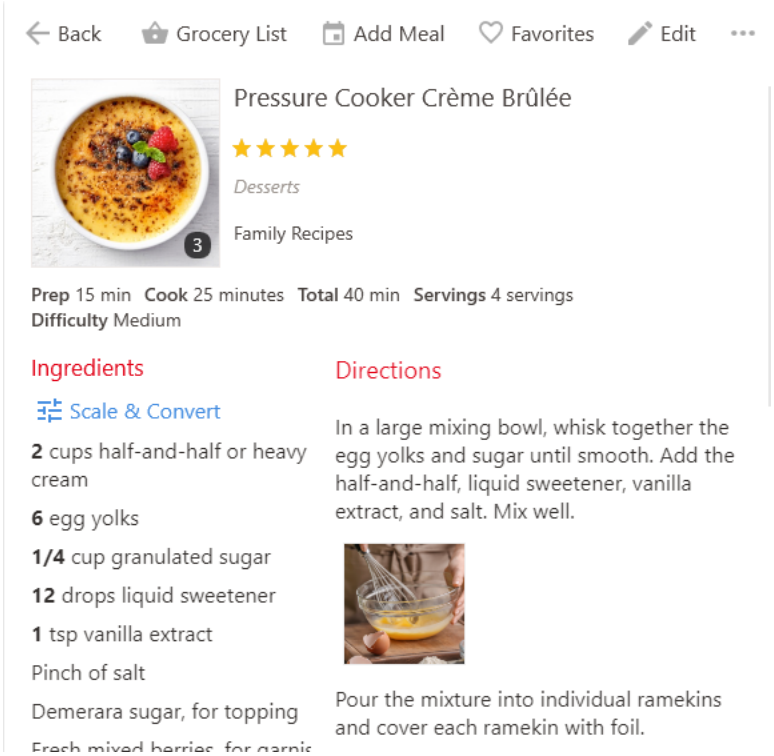


Example 4 (Source: Dribbble)



Example 5 (Source: Notion Everything/Recipe book)

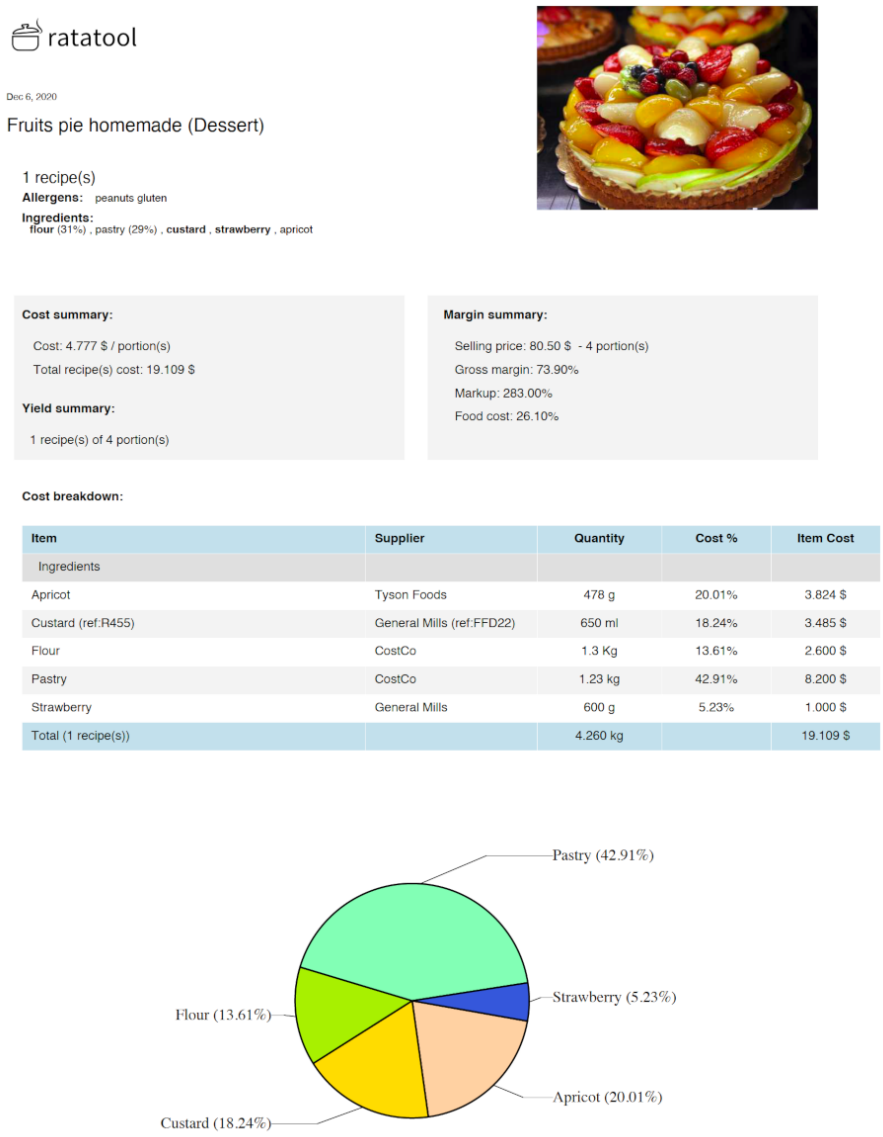
Recipe Details Interface



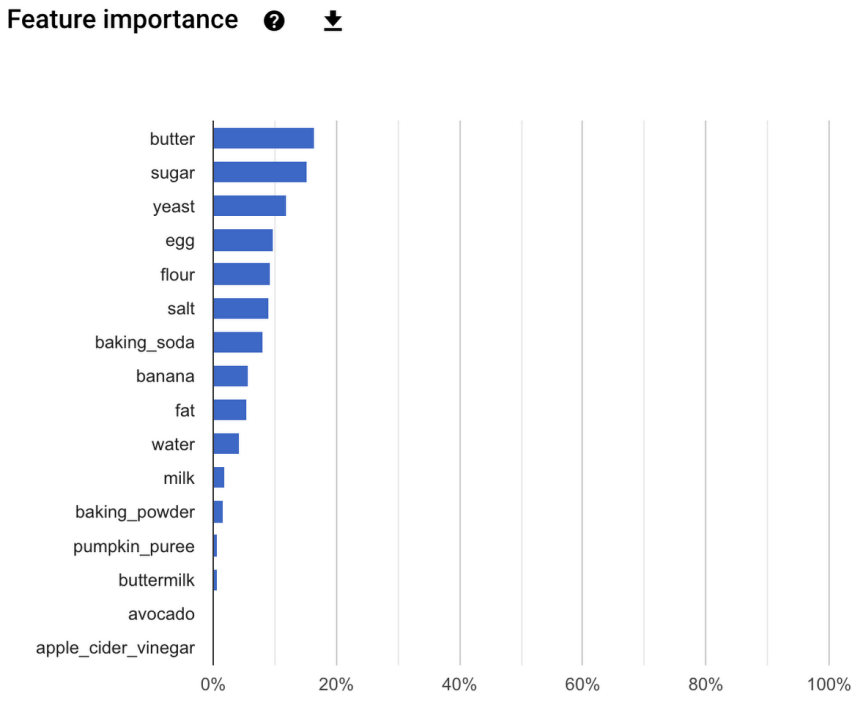
Example 6 (Source: Paprika)

Analytics Dashboard Interface

Your analytics dashboard will contain visualizations on useful site statistics like site visits (using JavaScript), recipes most commonly searched for, calorie content, and ingredients distribution. Here's how your analytics dashboard might look:



Example 7 (Source: ratatool)



Example 7 (Source: Google Cloud)

Your Project Deliverables

Over the course of each Exercise in this Achievement, you'll learn how to implement new features in your web application. And in the task at the end of each Exercise, you'll submit a deliverable that adds the new feature(s) to your application.

Here's a breakdown of the deliverables for this project, Exercise by Exercise.

Exercise 1: Getting Started with Django

- Create and manage virtual environments
- Install Django on macOS, Windows, and Linux

Exercise 2: Django Project Set Up

- Create Django project and apps

- Explore components and settings of Django project
- Create superuser and explore Django admin panel

Exercise 3: Django Models

- Create Django models and database tables
- Register Django models with Django project
- Add database records using Django admin panel
- Start writing automated tests

Exercise 4: Django Views and Templates

- Define views for Django application
- Develop Django templates
- Render web pages in Django application
- Define URLs and routes for the application

Exercise 5: Django MVT Revisited

- Update database tables containing recipe and ingredient data
- Enter recipe data using Django admin panel
- Develop a welcome page for application
- Develop subpages to display recipe information

Exercise 6: User Authentication in Django

- Set up user authentication for application
- Add login and logout features
- Protect Django views

Exercise 7: Data Analysis and Visualization in Django

- Implement search features in the application
- Provide data analytics dashboard

Exercise 8: Deploying a Django Project

- Prepare application code to upload to GitHub
- Package Django application and deploy it on web server

Optional: Advanced Deliverables

Optionally, you can add some additional advanced features to your Recipe application, if you like. This will really help make your final portfolio stand out. Here are some ideas of features you could add:

- You could add more attributes to the Recipe class including instructions and quantities for each ingredient.
- You could add a feature to allow users to share a Recipe page with a fixed (short) URL.
- You could add a feature to allow users to leave comments on a Recipe page.
- You could have the program search for ingredients closest to the user's search terms, in cases where the searched for ingredients aren't present.
- You could use a MySQL database instead of the default SQLite during development.