Proyecto 3: Pizza

Objetivos

- Ponte más cómodo con Django.
- Adquiera experiencia con el diseño de bases de datos relacionales.

Visión general

En este proyecto, creará una aplicación web para manejar los pedidos en línea de un restaurante de pizza. Los usuarios podrán navegar por el menú del restaurante, agregar artículos a su carrito y enviar sus pedidos. Mientras tanto, los propietarios del restaurante podrán agregar y actualizar elementos del menú, y ver los pedidos que se han realizado.

Hitos

Le recomendamos que intente cumplir los siguientes hitos en orden:

- Complete los pasos Menú, Agregar elementos y Registro / Iniciar sesión / Cerrar sesión.
- Complete los pasos del carrito de compras y la realización de un pedido.
- Complete los pasos de visualización de órdenes y toque personal.

Empezando

Python y Django

Al igual que con los Proyectos 1 y 2, asegúrese de tener una copia de Python 3.6 o superior instalada en su máquina. También deberás instalarlo `pip` . Si descargó Python del sitio web de Python, es probable que ya lo haya `pip` instalado (puede verificarlo ejecutando `pip` en una ventana de terminal). Si no lo tiene instalado, asegúrese de instalarlo antes de continuar.

Para ejecutar esta aplicación Django:

1. Descargue el `project3` código de distribución de https://cdn.cs50.net/web/2020/x/projects/3/project3.zip y descomprímalo.
2. En una ventana de terminal, navegue a su `project3` directorio. Tenga en cuenta que este es el directorio para un proyecto llamado Django `pizza` , dentro del cual hay una aplicación ya creada para usted llamada `orders` .
3. Run `pip3 install -r requirements.txt` in your terminal window to make sure that all of the necessary Python packages (Django, in this instance) are installed.

4. Run `python manage.py runserver` to start up your Django application.
5. If you navigate to the URL provided by Django, you should see the text `"Project 3: TODO"` !

Requirements

Alright, it's time to actually build your web application! Here are the requirements:

- **Menu**: Your web application should support all of the available menu items for Pinnochio's Pizza & Subs (a popular pizza place in Cambridge). It's up to you, based on analyzing the menu and the various types of possible ordered items (small vs. large, toppings, additions, etc.) to decide how to construct your models to best represent the information. Add your models to `orders/models.py` , make the necessary migration files, and apply those migrations.
- **Adding Items**: Using Django Admin, site administrators (restaurant owners) should be able to add, update, and remove items on the menu. Add all of the items from the Pinnochio's menu into your database using either the Admin UI or by running Python commands in Django's shell.
- **Registration, Login, Logout**: Site users (customers) should be able to register for your web application with a username, password, first name, last name, and email address. Customers should then be able to log in and log out of your website.
- **Shopping Cart**: Once logged in, users should see a representation of the restaurant's menu, where they can add items (along with toppings or extras, if appropriate) to their virtual "shopping cart." The contents of the shopping should be saved even if a user closes the window, or logs out and logs back in again.
- **Placing an Order**: Once there is at least one item in a user's shopping cart, they should be able to place an order, whereby the user is asked to confirm the items in the shopping cart, and the total (no need to worry about tax!) before placing an order.
- **Viewing Orders**: Site administrators should have access to a page where they can view any orders that have already been placed.
- **Personal Touch**: Add at least one additional feature of your choosing to the web application. Possibilities include: allowing site administrators to mark orders as complete and allowing users to see the status of their pending or completed orders, integrating with the Stripe API to allow users to actually use a credit card to make a purchase during checkout, or supporting sending users a confirmation email once their purchase is complete. If you need to use any credentials (like passwords or API credentials) for your personal touch, be sure not to store any credentials in your source code, better to use environment variables!
- In `README.md` , include a short writeup describing your project, what's contained in each file you created or modified, and (optionally) any other additional information the staff should know about your project. Also, include a description of your personal touch and what you chose to add to the project.
- If you've added any Python packages that need to be installed in order to run your web application, be sure to add them to `requirements.txt` !

Beyond these requirements, the design, look, and feel of the website are up to you! You're also welcome to add additional features to your website, so long as you meet the requirements laid out in the above specification!

Hints

- Unlike in Project 1, you shouldn't need to build your application's entire login and authentication system yourself. Feel free to use Django's built-in users and authentication system to simplify the process of logging users in and out.
- Before diving into writing your models, you'll likely want to think carefully about the different types of menu items and how best to organize them. Some questions to consider include: how should you represent the different prices for large and small versions of the same dish? Where do toppings fit into your model for pizzas, and how do you calculate the ultimate price of a pizza? How will you make the custom add-ons for the subs work?

FAQs

What is a "Special" pizza?

It's up to you to decide what a "special" pizza means, and to implement it accordingly. It could be one particular set of toppings, allowing up to 5 different types of toppings, or something else entirely!

How to Submit

IMPORTANT: *On 1 July 2020, CS50W will update to a new version and this version of this project will no longer be accepted for credit. If you haven't completed the entire course by that date, that's okay! If you ultimately receive a passing grade on this project, we will consider it equivalent in scope and content to Project 4 in the new version of the course, and within two weeks after 1 July 2020 your gradebook at https://cs50.me/cs50w will update to reflect that you have received credit for having completed Project 4.*

1. If you haven't already done so, visit this link, log in with your GitHub account, and click **Authorize cs50**. Then, check the box indicating that you'd like to grant course staff access to your submissions, and click **Join course**.
2. If you've installed `submit50`, execute

   ```
   submit50 web50/projects/2020/x/3
   ```

   Otherwise, using Git, push your work to `https://github.com/me50/USERNAME.git`, where `USERNAME` is your GitHub username, on a branch called `web50/projects/2020/x/3`.

3. Record a 1- to 5-minute screencast in which you demonstrate your app’s functionality and/or walk viewers through your code. Upload that video to YouTube (as unlisted or public, but not private) or somewhere else.
4. Submit this form.

You can then go to https://cs50.me/cs50w to view your progress!