

Exercise 2.5: Django MVT Revisited

Learning Goals

- Add images to the model and display them on the frontend of your application
- Create complex views with access to the model
- Display records with views and templates

Reflection Questions

1. In your own words, explain Django static files and how Django handles them.

In Django, files as images, CSS and JavaScript, are referred to as “static files”. This means they aren’t dynamically generated by the Django server, and they also don’t change while the web application is running. They’re typically provided by the developer during project creation. Django handles them by automatically figuring out in which folder these static files are, assuming they are stored in a specific structure, i.e. in a subfolder called static either in the project or inside an app.

2. Look up the following two Django packages on Django’s official documentation and/or other trusted sources. Write a brief description of each.

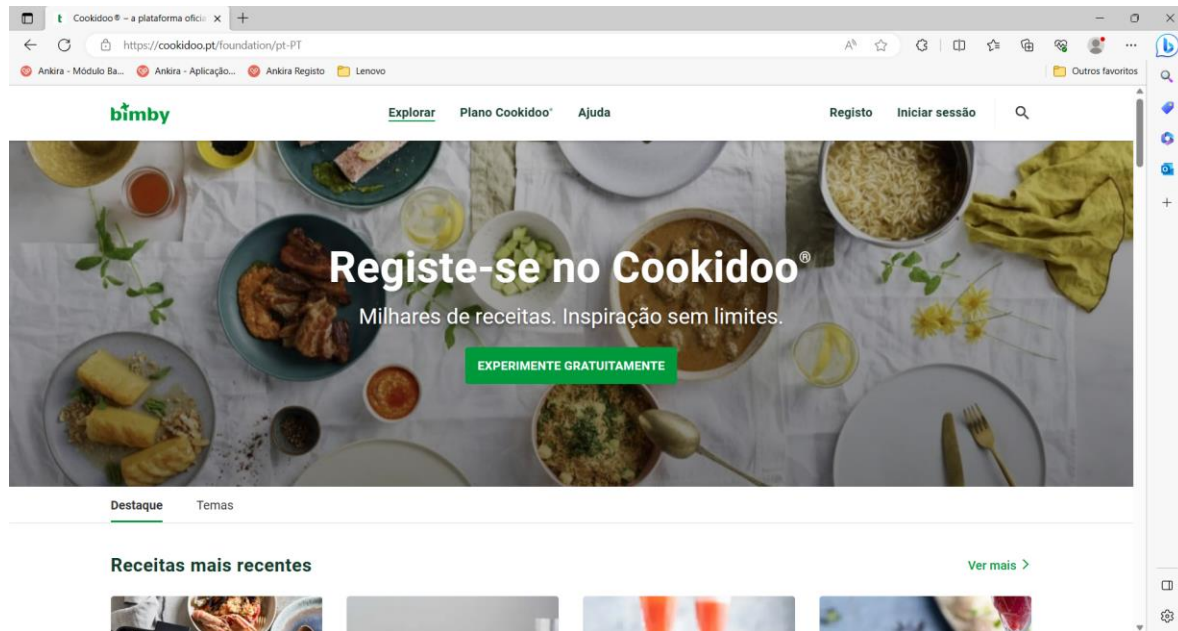
Package	Description
ListView	A page representing a list of objects. While this view is executing, <code>self.object_list</code> will contain the list of objects (usually, but not necessarily a queryset) that the view is operating upon. Helps display data as a list.
DetailView	While this view is executing, <code>self.object</code> will contain the object that the view is operating upon. Helps display data details.

3. You’re now more than halfway through Achievement 2! Take a moment to reflect on your learning in the course so far. How is it going? What’s something you’re proud of so far? Is there

something you're struggling with? What do you need more practice with? You can use these notes to guide your next mentor call.

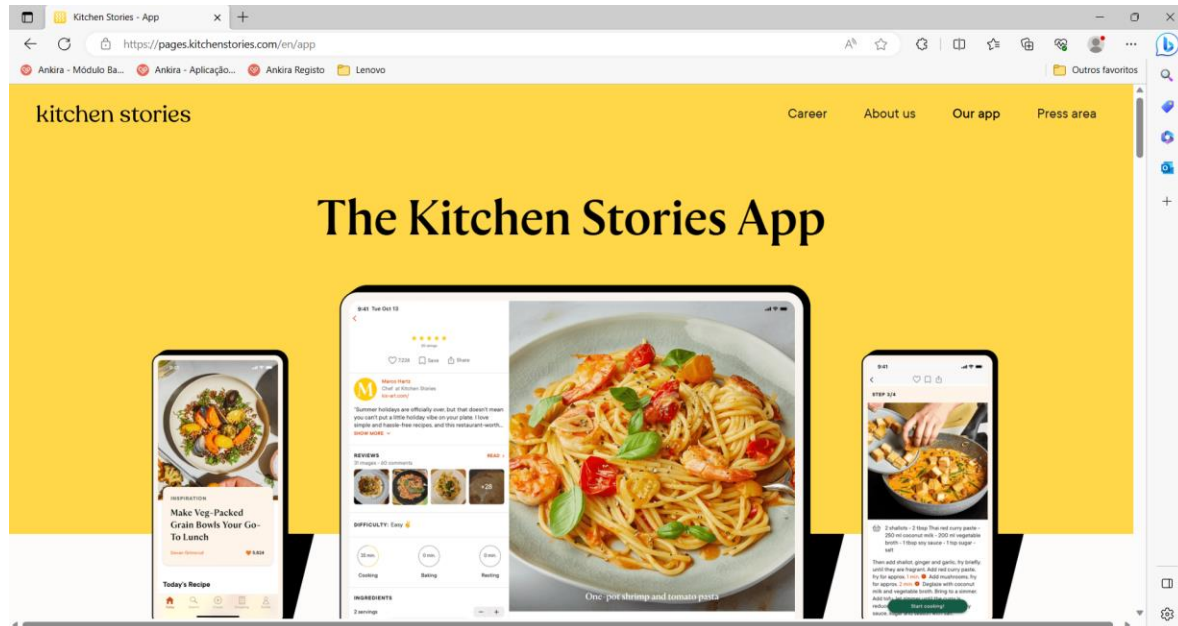
So far I've been able to complete the exercises/tasks. The exercise/task which took me more time to complete was the one about Django Models. I spent a considerable amount of time creating the database blueprint. I think I need more practice with Django Models.

Frontend inspirations



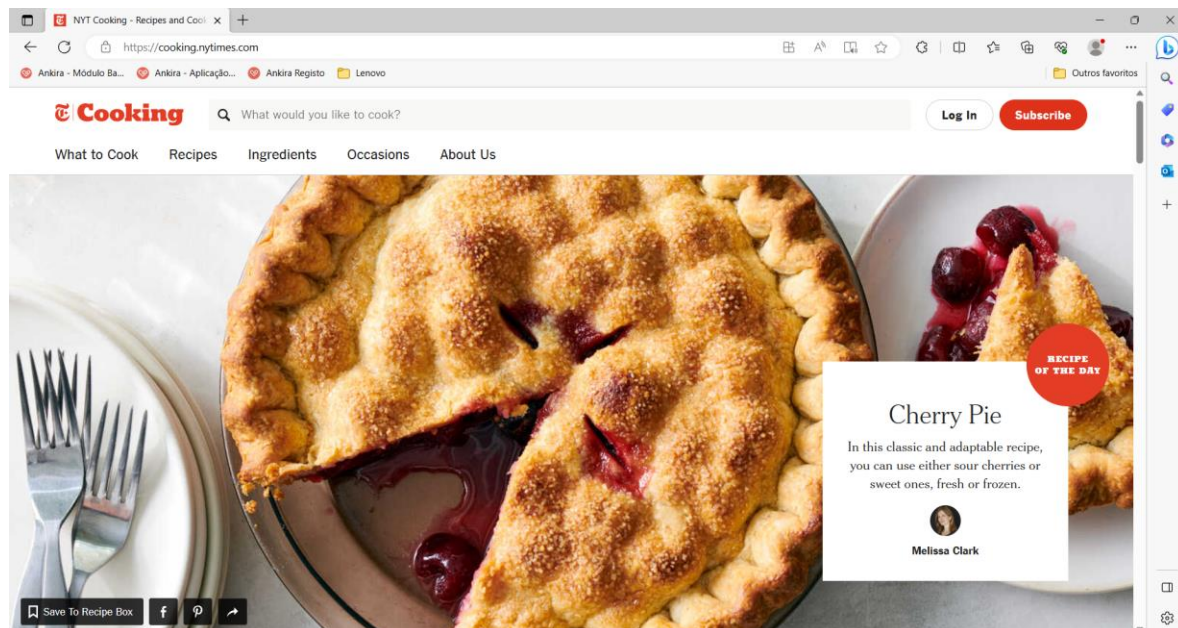
1 Cookidoo® (<https://cookidoo.pt/foundation/pt-PT>)

I like the welcome page with the photo in the background. I also like the fonts, recipe cards and the overall layout of the website (very clean).



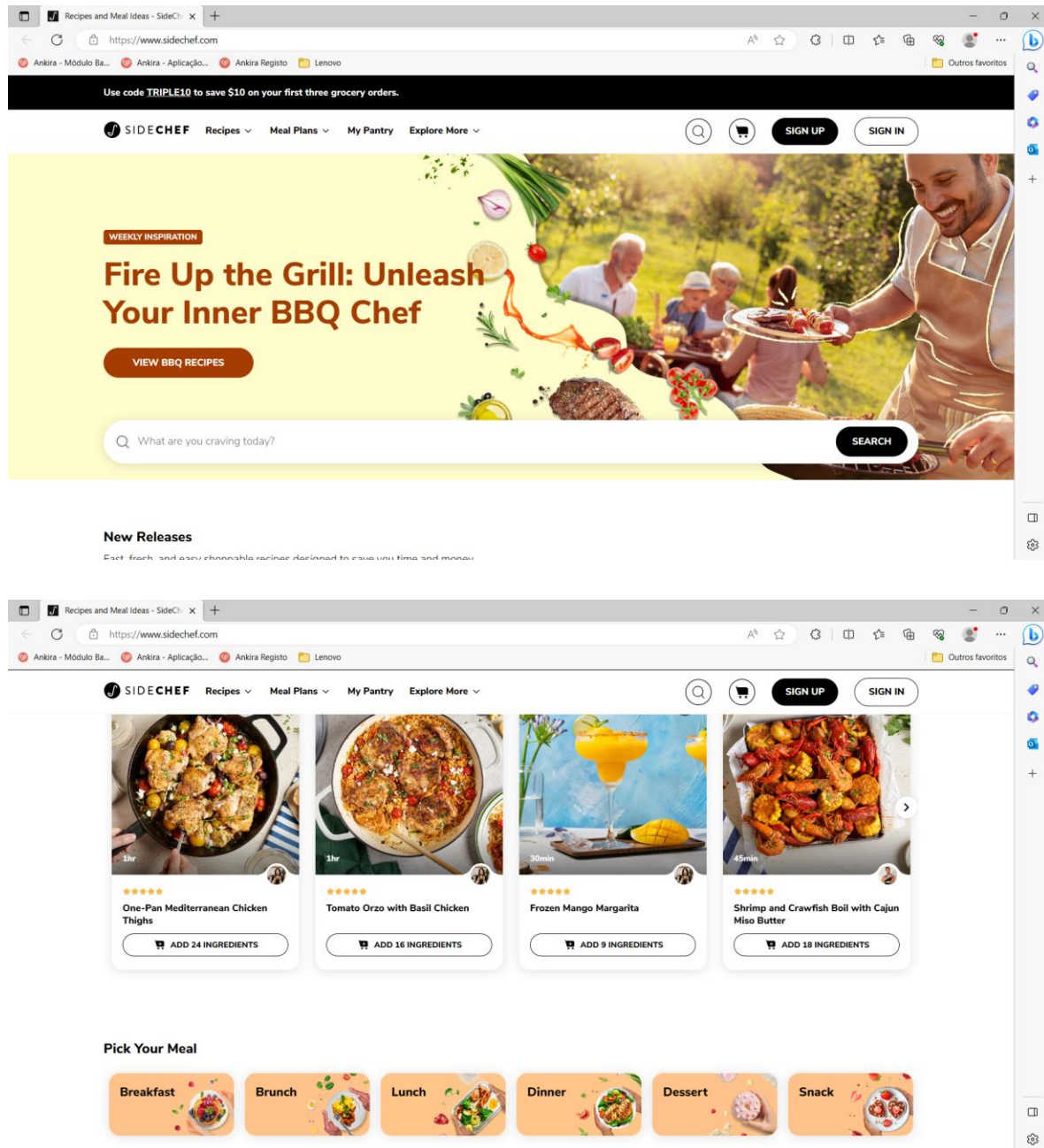
2 Kitchen Stories (<https://pages.kitchenstories.com/en/app>)

I love the design (of the website). I also like the color choice (yellow). We see a lot of green and red in recipe websites. I also like the lettering.



3 NYT Cooking (<https://cooking.nytimes.com/>)

I like the welcome page with the recipe in display with a big photo. The font reminds me of the fonts used in printed press (newspapers, magazines, etc.) which I think it's intentional. I also like the recipe cards and details.



Bonus SideChef (<https://www.sidechef.com/>)

I like the recipe cards and details. Love the images they use for collections and the way they use the same image at the top of the page with the lists of recipes from that collection. I also like the way they display the ingredients with an image of the ingredient on the left side.