Team Member Full Name	NetID
Ethan Little	**elittle2**
Derick Shi	**dhsi2**
Michael Egan	**megan3**
Aaron Wang	**awang27**

Features Implemented for Phase 1

- Create New User Profile (Feature 1.1)
- Log-in (Feature 1.2)
- Log-out (Feature 1.3)
- Create Listings (Feature 2.1)

Persistent Storage Design

We are using SQLite database to persist our data. Our database includes the tables shown in *Figure 1*. Since we are utilizing the default Django authentication system, there are many additional tables and attributes that our program does not use, so those are not displayed. For our purposes, the database contains 4 tables:

1. **auth_user** – the built-in table provided by Django to store user information.
2. **campusmart_listingcounter** – tracks how many listings each user posts per day.
3. **campusmart_listing** – stores individual listings. Each listing is linked to a user via a foreign key.
4. **campusmart_listingimage** – stores images associated with listings. Each image is linked to a specific listing.

Instead of creating separate tables per user or listing, Django uses foreign key relationships to associate data.

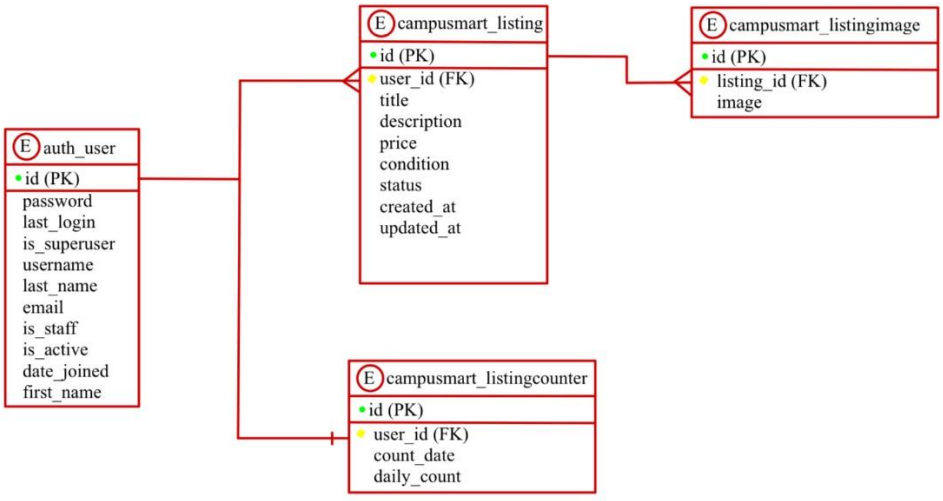

Figure 1 database schema

Demonstration of the Features Implemented for Phase 1

Feature 1.1

Figure 2 shows a screenshot for the sign-up page. When the user clicks the “Sign Up” button, a new user is created in the database based on the fields that are input. It utilizes the default Django authentication system, which automatically checks for invalid input such as usernames already in the database, weak passwords, conflicting passwords, and empty fields.

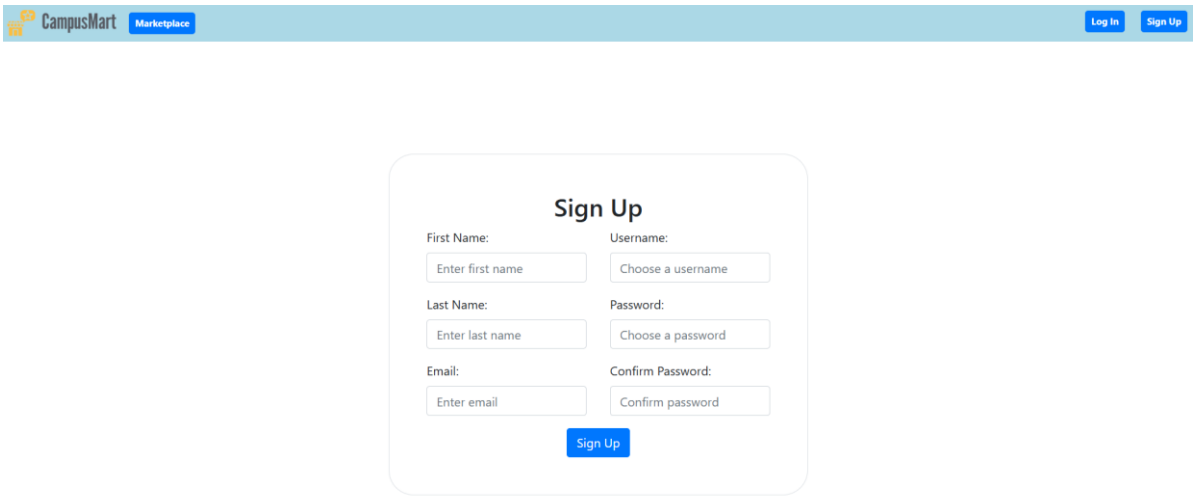

Figure 2 screenshot for feature 1.1

Feature 1.2

Figure 3 shows a screenshot for the log-in page. When the user clicks the “Log In” button, the input fields are checked to see if they match the username and password of any user in the database. If they do, the user is authenticated and the navbar displays a “Hello, <username>!” message next to the “Log Out” button (as show later in *Figure 4*) and the user is redirected to a temporary “Home” page. Once again, this utilizes the default Django authentication system which checks for invalid password/username and displays an appropriate message.

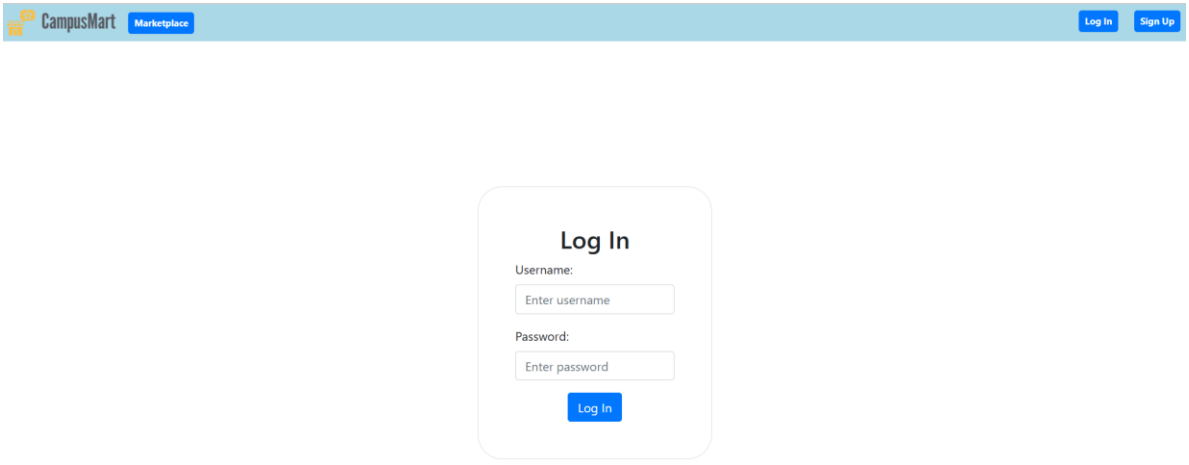

Figure 3 screenshot for feature 1.2

Feature 1.3

Figure 4 shows a screenshot for our "Log Out" feature. It is a button in the top right corner of the screen that is only displayed when a user is logged in. When clicked, the user is unauthenticated and sent back to the temporary "Home" page. Meanwhile the "Log In" and "Sign Up" buttons are displayed again on the navigation bar.

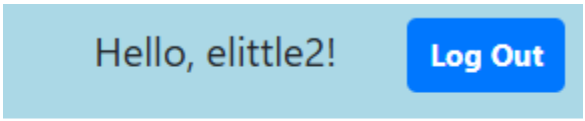

Figure 4 screenshot for feature 1.3

Create Listings (Feature 2.1)

Figure 5 shows a screenshot for our main market homepage. This shows the created listings for our user in the marketplace as well as the main attributes of its name, price, and its image. It also shows the corresponding buttons to bring up other pages to change the attributes for each listing like to view, edit, or delete.

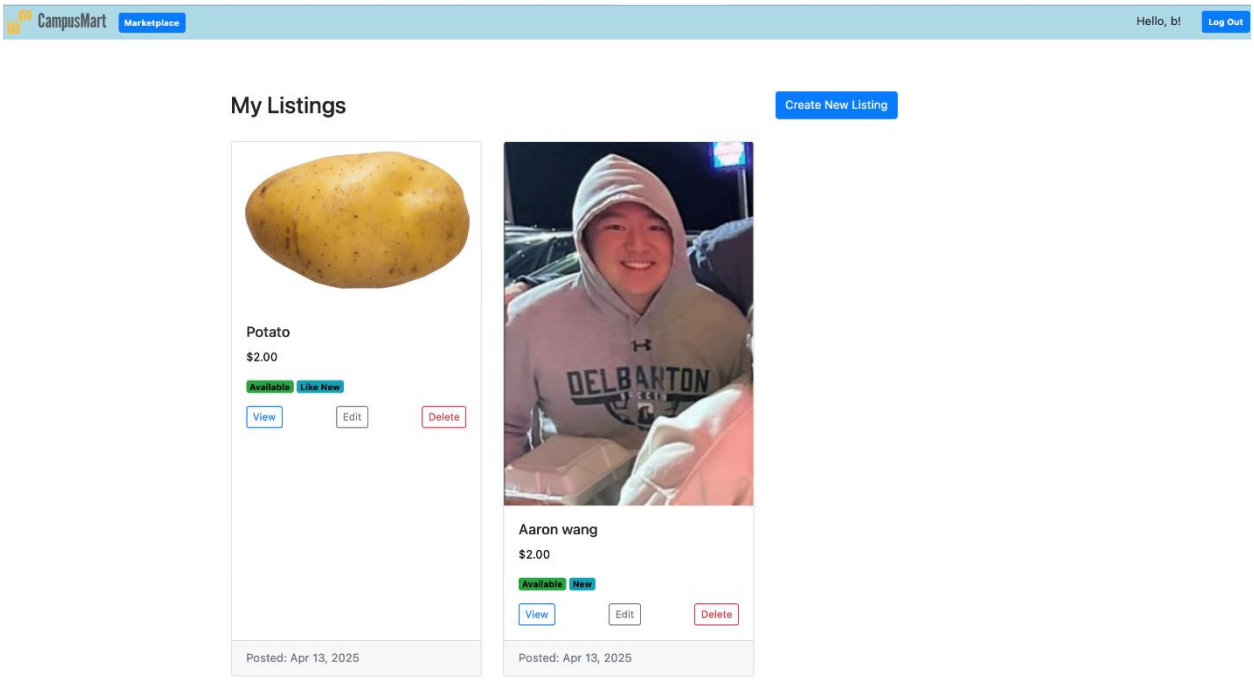

Figure 5 screenshot for feature 2.1

If a user clicks the "Create New Listing" button, it will bring up the listing form template and prompt them to enter the appropriate information to create another item to sell. This would create another entry in our **campusmart_listing** table in our database that stores the information needed for the listing. This could be seen in *Figure 6*.

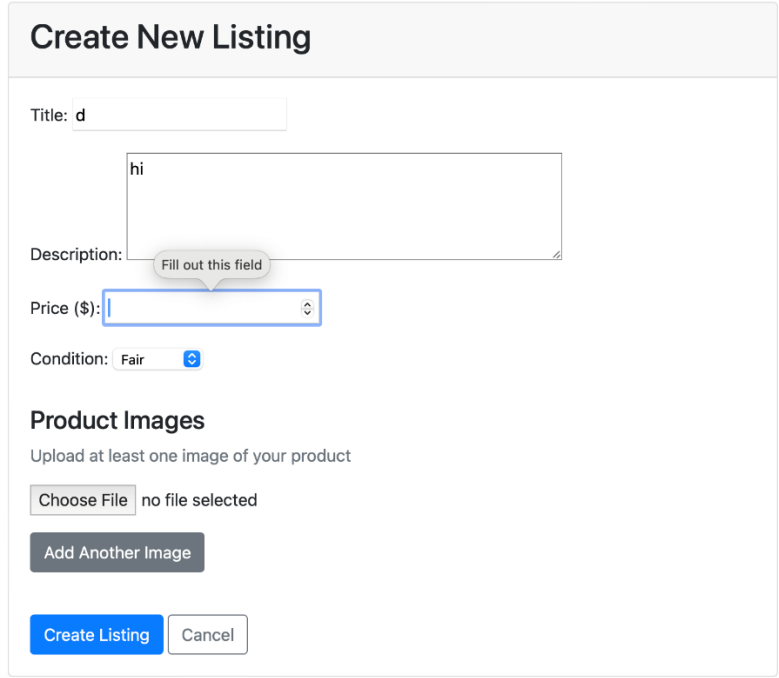

Figure 6 screenshot for feature 2.1

We also implemented the necessary safeguard to make sure that all items created would have the necessary information and you could only create 3 listings per day for a user or it prevents you from creating more. *Figure 6* shows a "fill out this field" notification while *Figure 7* shows a pop up that prevents you from creating more than 3 listings per day.

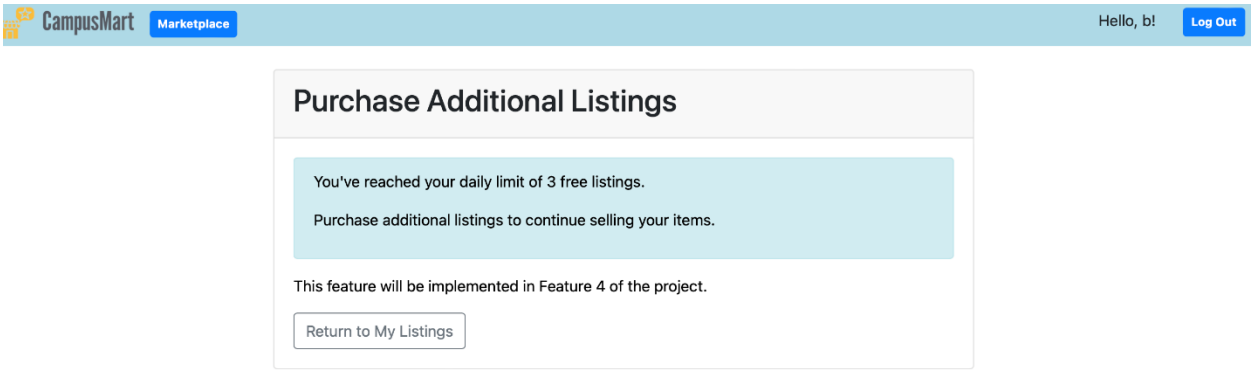

Figure 7 screenshot for feature 2.1