

## SP25 Programming Paradigms Group 18 Course Project

Team Member Full Name	NetID
Isabel Ojeda	iojeda
Lang Li	lli28
Phoebe Huang	chuang26
Kyle Phan	kphan2

### Features Implemented for Phase 1

- Feature 1.1
- Feature 1.2
- Feature 1.3

### Persistent Storage Design

We are using SQLite database to persist our data, which is managed by Django's ORM. The database schema is defined in the models.py file, and Django automatically handles the creation and migration of tables. Our database includes the tables shown in Figure 1, which includes a customized user model that inherits from AbstractUser. In the User table, we override the default *first\_name* and *last\_name* fields and introduce a custom *name* field. User has 3 tables: campusmart\_user, campusmart\_user\_groups, and campusmart\_user\_user\_permissions.

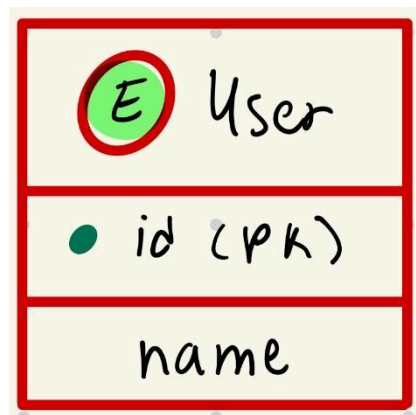
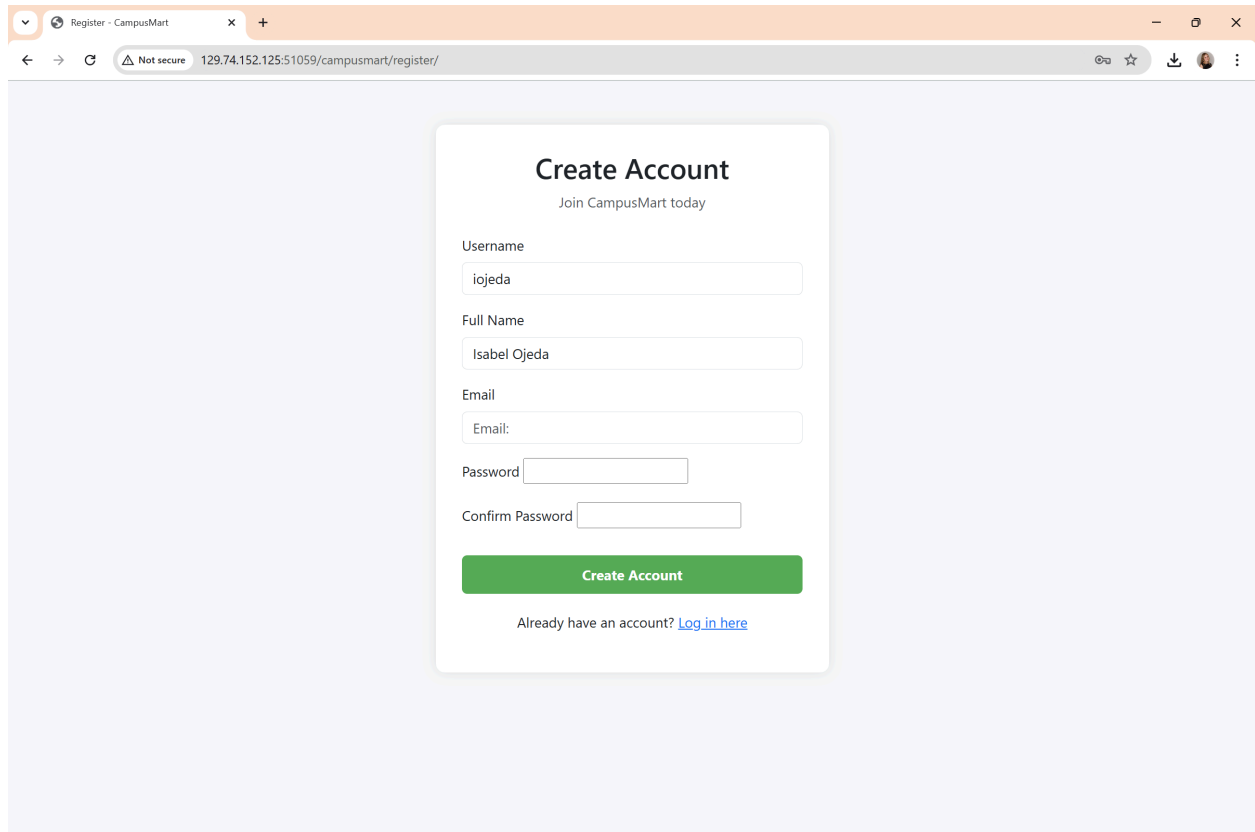


Figure 1 database schema

## Demonstration of the Features Implemented for Phase 1

### Feature 1.1

Figure 2 shows a screenshot for feature 1.1: Create new User Profile. This feature is implemented in register() view in views.py. It allows users to create a new profile by entering their name, username, email, and password. If the form is valid, it saves the new user to the database, immediately logs them in, and redirects to the homepage.

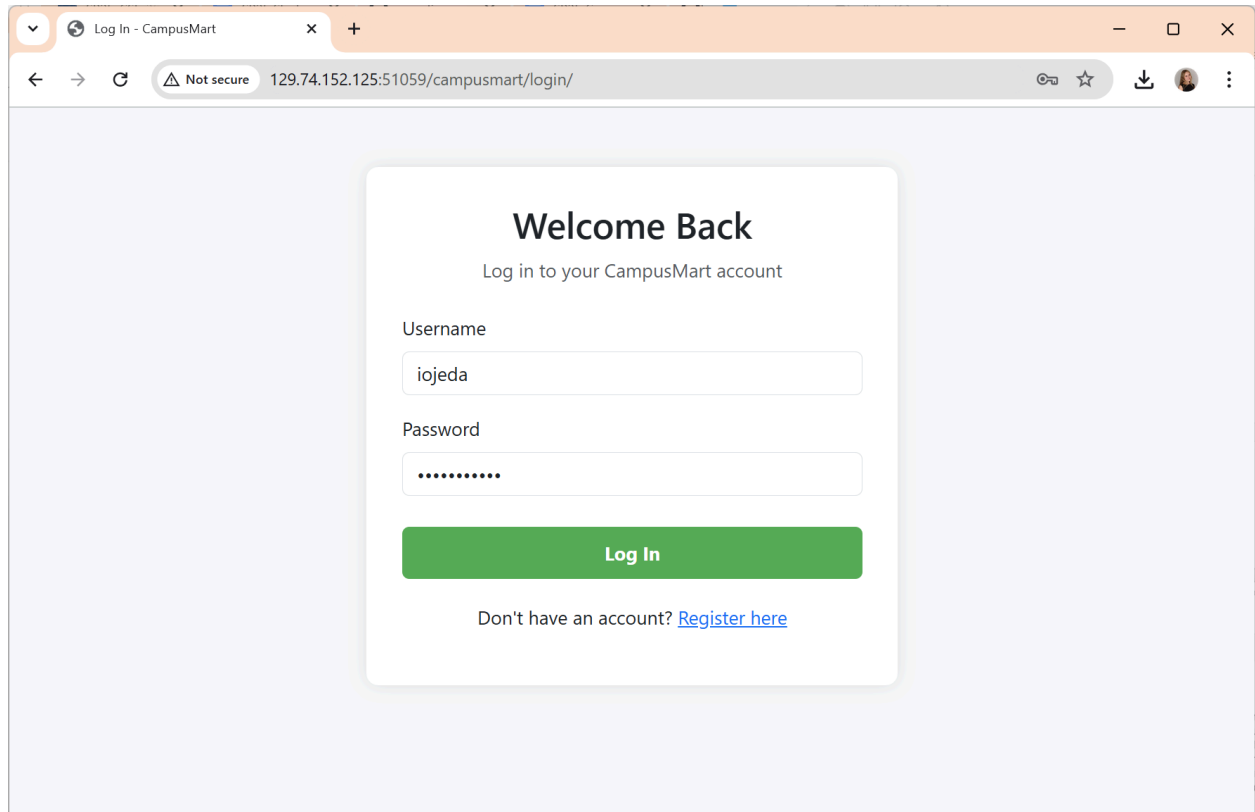


The screenshot displays a web browser window with the title 'Register - CampusMart'. The address bar shows the URL '129.74.152.125:51059/campusmart/register/'. The main content area features a white registration form titled 'Create Account' with the subtitle 'Join CampusMart today'. The form contains the following fields: 'Username' (with the value 'iojeda'), 'Full Name' (with the value 'Isabel Ojeda'), 'Email' (with the label 'Email:'), 'Password', and 'Confirm Password'. A green 'Create Account' button is positioned below the password fields. At the bottom of the form, there is a link that reads 'Already have an account? [Log in here](#)'.

*Figure 2 Screenshot for feature 1.1 showing how to register*

## Feature 1.2

Figure 3 shows a screenshot for feature 1.2: Log-in. The login function uses Django's authentication system in its implementation. If the username and password are valid, the user is logged in. Otherwise, an error message is displayed to the user.

A screenshot of a web browser showing the login page for CampusMart. The browser's address bar displays "Log In - CampusMart" and the URL "129.74.152.125:51059/campusmart/login/". The page features a central white login card with a green border. The card has the heading "Welcome Back" and the subtext "Log in to your CampusMart account". Below this, there are two input fields: "Username" with the text "iojeda" and "Password" with masked characters ".....". A green "Log In" button is positioned below the password field. At the bottom of the card, it says "Don't have an account? [Register here](#)".

Log In - CampusMart

Not secure 129.74.152.125:51059/campusmart/login/

### Welcome Back

Log in to your CampusMart account

Username

Password

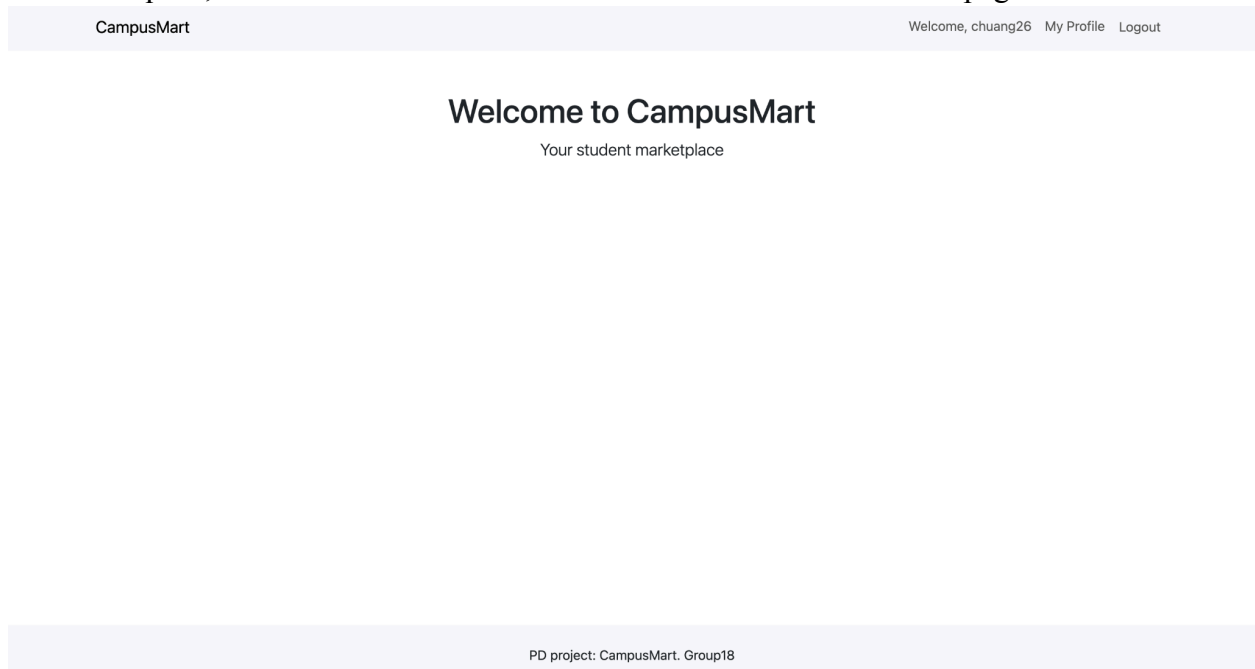
**Log In**

Don't have an account? [Register here](#)

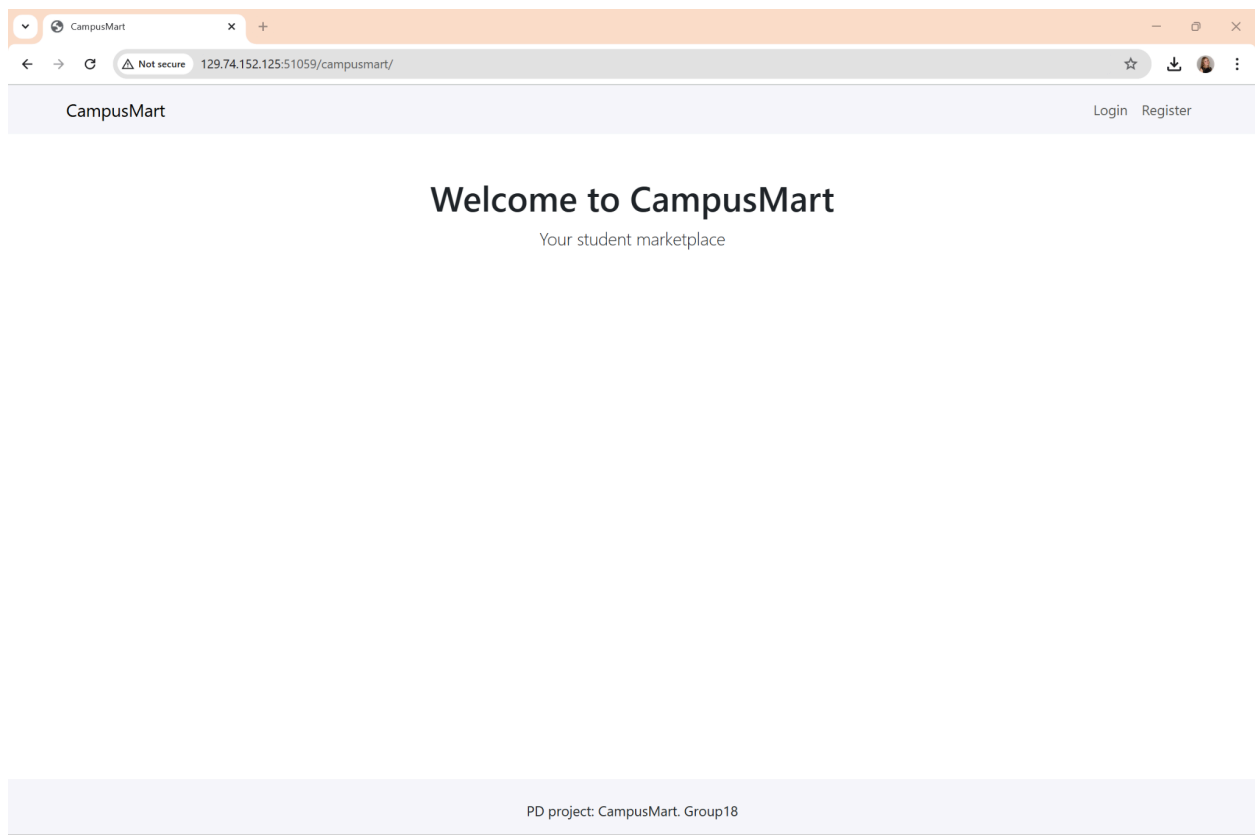
*Figure 3 screenshot for feature 1.2 showing how to login*

### Feature 1.3

Figure 4 shows a screenshot for feature 1.3: Log-out. In this feature, when logout is triggered via a POST request, it clears the user's session and redirects them to the home page.



*Figure 4 screenshot for feature 1.3 showing the logout bottom on the top right*



*Figure 5 screenshot for feature 1.3 showing the user being redirected to the home page after logging out*