TIKKACHU
NINTINDER

Overview

Our application takes the powerful matchmaking properties of the dating app *Tinder* and targets a whole new user base: gamers. Nintinder helps bring gamers together and make the most out of their gaming experience. Users will set up a profile that captures their primary interests and what they are looking for. Using this information, Nintinder will bring forward the best users who fit the description. Ultimately, users will be able to decide if they are interested in connecting with those suggested users or not. Nintinder will also feature a social feed where players can post status updates, share game results, post about gaming events, and much more. For all of your gaming needs, Nintinder is the one-stop solution for the optimal gaming experience.

Team Members

Johan Pereira, Raymond Liu, Daniel Zhang, Kush Patel, Oliver Pang, and Bhishma Pant

Github Repository

https://github.com/Philosobyte/nintinder

Design Overview

Login/Logout: We used a simplistic interface for both our log-in and log-out, keeping our red and white theme cohesive throughout our website from the sidebar to our headers.

Tailored Views: Because much of our last project included tailored views for a randomly selected user selected when loading the page, all we needed to do was complete that and link it to the current django user. For example, we had matches tailored but not the games on the profile page.

Forms: We created Django forms for our Settings Page to allow users to update their personal information, namely e-mail, first name, last name, and date of birth. We also created Django forms on our Profile page to allow Users to add a Game through a dropdown and create an Interest model in their database.

Problems/Successes

Successes: When we meet, we were very productive, with great communication throughout. Everyone was willing to drop other work to focus on the project. Development was done with increasingly better git etiquette, and both problems and successes were shared and discussed among the team.

Problems: Previously we had our own User model, so before we could do anything we needed to merge that and the django auth model for users. This in turn also required repopulating the database for any model depending on it, and soaked up quite a bit of time. Time-wise we were not on top of our game this time, we were rushed to complete the project a few days before the deadline due to a wrongly perceived due date.

Team Choice

Our team is considering incorporating password reset into the website through email. This, like in other applications, will be used to recover a lost password through the provided email on the account.

Part 4: Individual Write Up

(16%) Johan Pereira

As the main database dude I looked into hooking up our model design to the Django User model. In the process I worked with Ollie to refactor out the parts of the User model that are exclusive to Nintinder (not Django) into a new model called Profile (that retains a one-to-one relationship with Django's User). With respect to the authentication functionality of the project, I was able to configure and solve issues regarding redirections. For the settings page, I tried to help out as best I could but really I ended up learning a lot more from Raymond's success and his overall prowess. Finally for overall housekeeping, I helped out with almost everything relating to merging/branching in our github repository.

(16%) Raymond Liu

I made three alternate colorings of the logo to be used throughout the site, especially for the login and logout pages. I spent most of the time learning how to use forms and Ajax - I got Matches forms half-finished, before our priorities shifted and I finished the Settings forms.

(16%) Daniel Zhang

My main job for this project was to handle anything related to CSS. However, the my main contribution to this project is that I designed the login and logout pages. In addition, to that I help troubleshoot and fix any bugs that arise during git merge.

(16%) Kush Patel

For this project, my contributions involved incorporating the Django Forms into our Settings page. This included updating our previous code to use the Django Form instead. Most of my time was spent learning more in depth about forms and during that I attempted several implementations and created a foundation of code. Our final implementation, however, was done by Raymond.

(16%) Oliver Pang

As I was in charge of populating the database, I worked on migrating our user to django, with Johan, and repopulating the database to adjust. We played with multiple ways to merge the user models, and eventually settled on a OneToOne. Besides that I worked on miscellaneous tasks including groups, logout, the writeup, and general testing/help for the other groups.

(16%) Bhishma (Gaurab) Pant

I was in charge of a lot of the url mapping and views linking from the database to the front-end once again. I used the log-in page and log-out page created by Dan and adapted it to the Django template inside registration to basically utilize the database updates created by Oliver and Johan, by using Django's pre-existing User model I was able to easily allow for log-in. I was also responsible for some quick cosmetic changes like adding placeholder to the input forms in the log-in page (which took way longer than I'd care to admit!).