'Rendezvous' Team Write-Up for Project 2

Team: Etc Etc Etc

Overview

The general design and goal of our web application is still the same as in Project 1, but there have been changes made to the UI of the 'Create a New Project' page as suggested in the feedback from Project 1. We basically changed some of the text fields such that they aligned correctly with the data fields in our data model. We then implemented our data model (diagram PDF can be found in the project folder), created mock data, and represented it in our UI.

Team Members: Leonardo Costa, Samantha Cote, Robin Wu, Matthew Kelley, Ibiyemisi Gbenebor

Github Repository: https://github.com/LOCosta/CS326-Etc-Etc-Etc

Design Overview

Our data model can be seen in diagram form in the PDF in the project folder labeled *rendezvous_data_model_diagram.pdf*. Basically, we have two large classes, *User* and *Project* that use/reference a variety of smaller classes (*Event*, *Location*, *Tags*, etc). These were registered in admin.py and then through views and templates have been displayed in our UI. The profile and project view functions take both a request and a respective id as arguments. The id is then used to fetch the specific profile or project from the database, and generates information pertaining to each (i.e. *user_name*, *proj_created*, *proj_name*, *date_created*, etc.), finally passing the information to their templates to be displayed. The other notable url mappings are those for the index page, the search page, and the project creation page. There are also two url mappings for generic list views for projects and users, although those pages are solely for debug purposes.

Problems/Successes

When implementing one of the templates, we realized we neglected to incorporate certain elements of necessary data in our data model, but this was quickly addressed and resolved. Also, when trying to run our server for the first time, we received errors from models.py which said that we were referencing data fields that did not exist in certain places. Basically, we had referenced classes before they were defined, and so they 'did not exist' when we tried to run our server and make migrations. This problem was easily fixed in models.py and from there we were able to move forward and continue registering and displaying our data model. These problems were minor, and we were able to work through them as a team and continue to make progress on the project.

In terms of successes, there were many. The homework assignments with the LocalLibrary in class were very helpful in preparing us for what needed to be done with our own data model. With some outside research, every team member was able to complete their task with minimal trouble, and problems that we did run into, we fixed as a team! We were also very good about dividing up the work evenly and early on, such that each team member had a manageable amount of work to do.

Individual Write-Ups

Robin Wu

My responsibility was to implement a functioning admin site and to add in mock data into the database. I implemented the admin.py file which registers the classes from Model.py. I defined a few classes to visualize the fields for the users in a more organized manner. Afterwards, I made the migrations and ran the server. Upon logging in, I added in mock data for the classes and their respective fields. Contribution to total: ~15-20%

Samantha Cote

My role in Project 2 was to implement our data model in models.py. First, I created and designed our data model diagram, using Photoshop. This has been saved as a PDF in our project folder and provides a good layout of what our data model looks like in Django. I then implemented the actual data model. I created each of the classes and class members in models.py in order to correctly represent the data which our site will need.

From there, other team members were able to register my classes in admin.py and create the URL mappings and templates/views. My role was primarily defining the data model as well as creating an aesthetically pleasing diagram of it, and then other team members were responsible for presenting/displaying that data in our site. Contribution to total: ~15-20%

Leonardo Costa

My role in the second part of the project was to implement the url mapping for our web application, as well as implement the view and template for the advanced search page. In addition to that, I was the one who initially implemented the skeleton website for our webapp, and I also created the generic template that the other templates extend.

Beyond that, I fixed all of the issues that Django had with rendering our website after everyone had otherwise finished their respective parts of the project. I did so by going through each of the pages and fixing any errors thrown, and then any HTML and styling issues I found. Contribution to total: ~20-25%

Matthew Kelley

In Project 2 I implemented the view functions and templates for both the profile and project pages. This involved fetching individual profiles and projects from the database by id, as well as relevant profile and project information. By using {% if … %} and {% for ... %} blocks, I created templates that dynamically display information on the page, like all projects corresponding to a particular profile, and all profiles corresponding to a particular project, handling for the case in which data may not be available. Contribution to total: ~15-20%

Ibiyemisi Gbenebor

For Project 3, I implemented the view functions and templates for the index and upload new project pages. This involved editing the actual templates for both sites and writing additional code for views.py. I also fixed the ‘upload a new project’ page in terms of fixing the text boxes because there were some problems from the last project. Contribution to total: ~15-20%