TEAM: LESS THAN THREE
APP NAME: e-Vent | TEAM WRITE-UP

https://github.com/aasthan/web-programming

Names: Vi Nguyen, Shanieka Pereira, Anh Chau Pham, Aastha Niraula, Depeng Kong, Harry Hu

Overview: Our application is called e-Vent. With e-Vent, students and organizations from the 5 college area can post events that are going on at their campuses. Our project so far has followed the proposal with the exception of a couple changes: the app no longer provides the option to rsvp through e-Vent, however registered users still have the option to "save" an event they are interested in using the "save" button provided on the Event Details page. We also created a search.html page.

Design Overview: For the basic template of our app, we decided to implement a search bar, a navigation bar as well as a footer and a "carousel" to view posted events. The models that we have implemented in Django include: Location, Tags, Price, Popularity, Event, and User. Our important URLs include the index, event-details, profile, event form, search and browse-vent.

The index page is 127.0.0.1:8000/e_Vent/ : This leads to our main page index.html

127.0.0.1:8000/e_Vent/event/<int:pk>': This path leads to eventDetails.html which gives out details of the event as well as the location and related events

127.0.0.1:8000/e_Vent/profile/ : This path leads to profile.html which is the the user profile for registered users that displays the user's created and saved events

127.0.0.1:8000/e_Vent/form/ :This path leads form.html where users can create and post their events

127.0.0.1:8000/e_Vent/'browseEvent/: This path leads to browseEvent.html which is a page to browse current and upcoming events

127.0.0.1:8000/e_Vent/login/: This path leads to login.html which is a page for registered users to login

127.0.0.1:8000/e_Vent/signup/: This pages leads to signUp.html where visitors can create an `account

Problems/Successes: One of the problems we ran into when working on project 2 was when we were trying to populate the database. When multiple group members were trying to populate the database, they were finding that the events they implemented were deleted and replaced by the next person who populated the database. This issue was resolved by pushing our changes on Github after running the server. Another challenge our team faced was implementing the Google Maps function. Our group also came across issues when we were working on the project was getting the "popularity" function to work, this stemmed from our choice to make popularity a foreign key. We successfully populated the Homepage with data from the database.

TEAM: LESS THAN THREE
APP NAME: e-Vent | INDIVIDUAL WRITE-UP

https://github.com/aasthan/web-programming

Vi Nguyen

In project 2, for part 1, I discussed the design with the team and made the UML design. I finished most of part 1 (from step 1 to 4), which consists of creating a model for each object (in models.py), registering them with Django (in admin.py) and creating a superuser. For part 2, I set up the proper directory for our application and added the needed css/js files. I shared the work of the base_generic template with Shanieka and helped debugging with some of her parts. I created a template for index page, did some research and wrote a few additional methods to display things to match our mock UI format, and implemented the URL mapping + views for several pages. I also helped implemented the search function in our application. Other than that, I mostly communicated with other team members and helped them to debug and write the team write-up for part 3.

I would say I've contributed about ~30% of the work to this project.

Anh Pham - Contribution: 25-30%

- For Part 0: I start up the project by setting up the Django skeleton for our website.
- For Part 1:
    - I fix bugs and refactor the event model in models.py .. In addition, I also populate some data in the database (images, url) so that the website does not throw any 404 errors.
    - For Advanced Configuration, I add list and detail views to the Event model to customize the admin site views so that it's easier to view data.
- For Part 2:
    - I create the template for eventDetails, link that to the database and hook the template with urls and views so that the site renders correctly. I used Google Maps API to connect to the event's location in the database, and display that on the view.
    - I also contribute in code-refactoring and fixing bugs in other pages such as views, urls, login, signup, profile.
- I aid Shanieka with Part 3 (Team Write-up) and created and keep track a To-do list for the whole team.

Aastha Niraula:

In Part 1 of the project, I worked on creating the basic admin site for e_Vent that would be used later for populating data into the database. I focused on user, location, tag and price and customized their interface to display filters and to make it easier to populate and update the database using the django admin site. I also updated the README file for the project with the basic information so that detailed instruction could be added later.
In Part 2 of the project, I focused on editing views.py to create filtered queries that would retrieve accurate data from the database and display it using the templates. I created the template for browseEvent.html and linked it to the url and view class. I believe I contributed around 18-20% of the work.

Depeng Kong:
In Part 1 of the project, I help created around half of the different events to enrich database as part of the preparation for our team in order to continue Part 2 with a set of pre-loaded data to test with, while kept a copy of the database and saved a potential data loss. In Part 2 I invested and created search page as well as the new event form page and modified relevant files in base_generic.py, urls.py and views.py accordingly. Overall I think I contributed about 15% of the work.

Shanieka Pereira
In Part 1 I populated events for the database. I added an estimated 15 events, which was time consuming as there were many models that needed to be populated. Populating the database also gave the the chance to see if there were any issues in our code that were affecting the database (ex. Issues with the popularity model and issues when populating the database as stated in the team write up) . In Part 2 I wrote code for the Profile page (profile.html) as well as the base code needed for the page in views.py and hooked up the url (urls.py) I also helped in creating the base_generic template. Lastly, I wrote the Part 3 team write up portion. I believe I contributed 15% of the work.

Harry Hu:
In Part 1 of the project, I started adding events for the database, and added 6 events in UMass and MHC. I also investigated in the correct relationships we should use in the database for module price and module popularity. I deleted the error genres and gathered other events in the common genre.
In Part 2 of the project, I wrote code for the Login page and Sign up page(login.html and signup.html), and had a design theme for them. I also wrote related base code in view.py and urls.py. I believe I contributed ~15% of the work.