

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

Please note that to run project 3 properly, you need to run command: `pip3 install django` and command: `pip install django-money`. If the application prompts you to log in, please use username "compsci326" and password "compsci326". You can also create a new profile page on Django admin page, please use password "compsci326".

Overview: e-Vent is a web application for students and organizations in the 5-college area to post upcoming events. In Project 3, we have not made any significant changes in comparison to our proposal and the mock UI from Project 1, besides creating some forms functionality

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

Git Repo: <https://github.com/aasthan/web-programming>

Design Overview:

- **Authentication, Authorization, and Session Support:** Supports user login/logout functionality as well as tailors the views to display different data depending on which user is logged in. To be more specific, each user has their own **Profile page** (template: profile.html) that displays their basic info, all events created (owner = user) and those saved by currently logged in user (owner = another user). Right now, the profile is only visible to the user themselves, for it contains their privately saved events.

- **User Interaction with Forms:** Implements forms for creating and updating data in e-Vents. Once logged in, user can click on the (+) button on the top right of the navbar to create a new event, whose owner is the user themselves. The form updates the database accordingly to display newly added/modified events or not if they are deleted by the user owner. Please note that only logged in user has the ability to save a particular event. Also, to delete an event, go to `http://127.0.0.1:8000/e_Vent/event/[event-id]/delete`. And to update an event, go to `http://127.0.0.1:8000/e_Vent/event/[event-id]/update`.

Successes:

- Implements "Save" button. User interested in an event can "save" it and it will be saved in their private profile. It is a crucial part of project 3 as it also determines the "Popularity" of an event. In short, if an event has been saved by many different users, it is likely to be popular. The index page uses this piece of information to update "popular events" carousel, which is available to both authenticated/anonymous users.
- Dynamically populates "Saved Events" (events saved by currently logged in user), "Your Events" (events created by currently logged in user), "Related Events" (events sharing one or more tags with currently viewed event) and "Popular Events" (described above)
- Extends Django's USER model to re-implement Profile model. We were able to connect our "Profile" and Django's "User" on Django Admin page. In other words, these 2 models depend and connected to each other. This is a key point in authentication & session support.

Problems:

- We read through Project 3 submission requirements and not sure if we should implement a Signup page where user could create a profile. For now you can only do that in Django admin site.
- Search function still needs to append /search/ manually to display results
- Some contributions division and quality of work among team members.

Team Choice:

- Dependencies/Frameworks: REST framework & AJAX for save button, djmoney for better formatting for event price. We also might include a password reset page for the users
- We also want to turn this application into a more social website: Besides popularity, users will be able to create and individualize their own profile page. They also will be able to Like and Comment on an Event. If we have time, we can even implement a friends network for each user so that interactions and connection is stronger.

TEAM: LESS THAN THREE
APP NAME: e-Vent | INDIVIDUALS CONTRIBUTION

Aastha Niraula - 15%

I worked on extracting similar events based on the event that is currently displayed to the user. I used the `get_object()` method to retrieve the event object from the context and then the `get_tag()` to retrieve the tag associated with the specific event. The tag is then used as a filter in creating the query set of events that is returned by `EventDetailView` class. This list used in the carousel in the `eventDetails.html` page to iterate through related events. I helped populating users for the admin site as well as fixing minor linking issues with the signup/login page.

Vi Nguyen - 30%

I implemented "Save" button (REST framework and AJAX.) Detected issue with Django's `USER` model with our own version of User model (now named Profile) and helped connect the two. Wrote the new class-based view for `YourEventDetail` to populate "Your Event" carousel. Modified database to have more information to populate carousels. Created `.gitignore` file to resolve the problem with database being constantly wiped out. Contributed to **team choice** section. Wrote most of team writeup.

Anh Pham - 30%

Part 1: I did the entire Part 1. I provided authentication support in our application:

- Create mock users and groups.
- Implement user login/logout functionality and render views based on user permission:

Implement login and logout templates. Also, logged out users cannot create event or view profile.

Render nav-bar views based on login/logout status in base generic.

Part 2:

- I connect Django user model and our application profile model.
- Helped Vi with implementing popularity count and display popular events.
- User permission: Logged out user cannot save event. Fix and reformat save button and functionality to save event.
- Implement and fix bugs on profile page and create event page.

Part 3: Contribute partially in Team write-up: Team choice + Application installation requirements.

Shanieka Pereira

I helped to work on the Your Events and Saved Events function of the profile page. These two functions involved displaying events that the user has created and saved, respectively. This need to be done through `views.py`. Lastly I wrote out the draft for the team write up. I believe I contributed 10% to project 3.

Depeng Kong - 10%

I created event form, update event and delete event pages. Modified relevant files in order for this to function properly with the database. Overall I'm estimating my contribution to be around 10% of project 3.

Harry Hu - 10%

I wrote codes for `profile.html` to implement the functions of displaying saved events, and also related supporting codes in `views.py`. In this case, I think I contributed around 10% to the team project 3.