

Team: git push --force

Overview:

For our web application, we are creating a website for the UMass Outing Club. One of the functions of the website to manage a database of club members and upcoming trips. It also provides a way to track which members have signed liability waivers and completed the necessary paperwork. This will save club administration a lot of effort as they had been doing this manually with paper files piling up in cabinets before. The website will provide a convenient platform for club members to sign up for trips, and for administrators to manage the club. There have been no changes to the general idea of the website since Project 1.

Team Members:

- | | |
|---------------|-------------------|
| - Susan Pan | - Jeffrey Shao |
| - Caroline Yu | - Nonu Bajaj |
| - Kyle Vedder | - Stefan Kussmaul |

Github Repository:

<https://github.com/desporous/CS326Project>

Design Overview:

Most of our features are functional and user-editable. For trips, the “Add trips” button is only visible to admins. Creating new trips and editing current trips is not currently working, although we made the forms. Deleting a trip is also possible, but only available for admins.

Our user data is modeled by the UserProfile class. Users can now edit their information.

Log-in/out: There is now a user log in and out function so that it’s no longer a default user view

Registration: New users can register an account on the site

Problems:

Switching over to Django’s user authentication system was difficult, because it required splitting user information into a UserProfile model separate from the main User model (separating authentication and business logic). This made architecture slightly more challenging, but worked in the end. There were also some difficulties implementing forms, although these were mostly resolved. Comments were difficult to implement: they required AJAX calls and were rendered in Javascript. Posting new comments does not currently work due to Django CSRF requirements--we hope to have this fixed for the final release. There was also some difficulty in allowing users to view the profile of another user, since the public profile displayed the logged in user’s profile instead of the desired user.

Successes:

We successfully made forms to change information in the database, and we have much of the final functionality UMOC wants completed. Though we had some difficulty registering new users and linking up a new user and user profile for them, we managed to get it running pretty smoothly in the end. The dashboard now also filters out trips that have already ended into a separate “Past Trips” category, making it easy to identify which trips are ongoing. We also managed to get a rudimentary version of threaded comments running, though they are not aesthetically fitting with the rest of the site. Getting log-in functionality on the navbar was also a cool feature we implemented.

Team Choice:

Our work for the team choice functionality of the website is the addition of commenting and notifications.

Individual Write-Ups:

Stefan: I did a lot of the site architecture for this part of the project. This included switching from a static website to a user-authenticated website. That required many template and view changes, and was implemented by creating a new UserProfile model, and giving it a one-to-one relationship with the User authentication model. This was not as clean as I had wanted (ideally, there would only be one model per user, containing all information) but made the most sense, because it separated authentication and business logic.

Another of my projects was getting comments working, which were initially rendered server-side. However, it made more sense to render them in Javascript, which required AJAX and unfortunately didn't end up as clean as I would have hoped. While comments are displayed correctly in threads, it is not currently possible to post new comments.

Nonu: I created the registration and login pages, the forms (and views) for those pages, and the login from navbar functionality. I also made filters (that add classes to forms), used those filters to beautify the profile settings page, and modified the UserProfile model to allow linking between Users (Django's implementation) and UserProfiles (our way of storing other user data, such as date of birth, phone number, etc.).

Jeffrey: Updated the dashboard to filter trips into current and past trips, updated UI for some pages and formatted some pages to utilize bootstrap. Made some minor bug fixes in other areas.

Kyle: [Added password reset forms and templates](#), allowing users to reset their passwords in the same way that locallibrary did. Note that this currently depends on an email server which has not been setup. Converted to [form based profile that takes Django's User object](#), built a [mapping](#) from User to UserProfile and then used said mapping to [update both UserProfile and User data in the database](#).

Caroline: Made Add Trip button and post form. Submitting the form doesn't work right now, because it thinks the form is not valid when I submit it. Added delete button and form to delete a trip. It can be found when clicking on the details of a trip. Then the delete button is under the drop down menu. Made Add Trip and Delete button only available for users with Admin role. Don't know how to make it available for members under the "Leader" group. Will fix posting add trip form and making it only available for "Leaders", not "Admins" in next iteration.

Susan: Added functioning administration template, allowing admins only to view a live list of all users and their administration level outside the django admin site.