G-Tech
ZooMaps
Spring 2018

Part 2: Team Write Up

- **Overview:**
  ZooMaps is a web application in which users can see public events happening on the UMass campus. Users can view a map of the campus and join events or create and add them. They can also comment on and rate events. Locations on campus are also open to comments and ratings. From a user's account page, they can view the events they're attending and which events and locations they have liked and commented on.

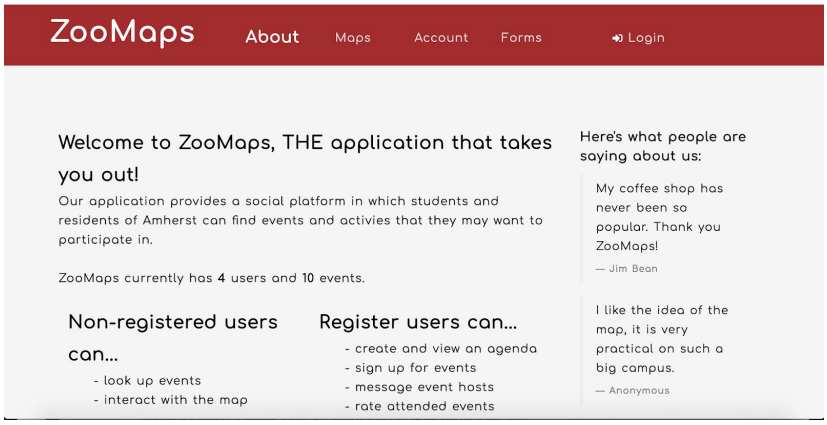
  ZooMaps is innovative in that it provides a public localized directory of events in an area. In this case, that area is the UMass campus. The university provides a similar tool in Campus Pulse, but Campus Pulse is lacking in functionality. Particularly, it does not place events on a map as our application does. To get directions, users have to look up the campus map or look up the directions themselves. ZooMaps solves this problem by merging these features together into one application.

- **Team Members:** Miles Black, Jordan Chen, James Curry, Mégane Michaud

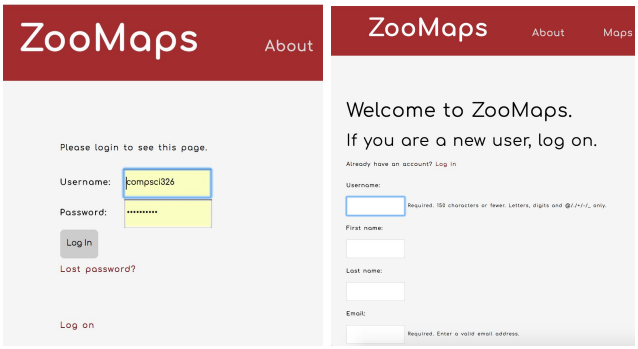
- **Github Repository:** https://github.com/m3gan3/ZooMaps

- **User Interface:**

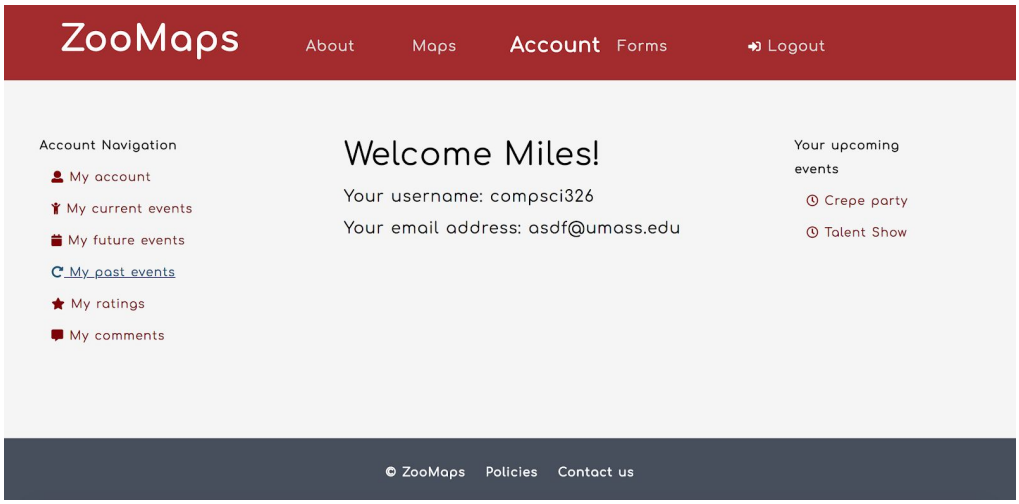
Any user, logged in or not, can access the index page, which gives information about the application and the different interactions possible.



The user can log in, or log on for the first time.



A user logged in has access to their account, and their different information. From their “Account” page, a user can see their username and email address, their current events, events they plan to attend, events they have attended in the past, ratings they’ve given to events, and comments they have made on events.



Users, logged in or not, can see a detailed view of the events, with the map, as well as their messages and their ratings. An event's page has a start and end date, a description, related tags, a list of attendees, an average rating, and comments. Logged in users are able to modify their ratings, add comments, and attend or unattend an event.

A logged in user can rate and comment events, or create an event.

- **Data Model:**

Our application uses different models: User, Event, Tag, CommentEvent and MessageEvent. Most of the interactions occur between a User and an Event. A User can leave a RatingEvent or a MessageEvent on an event. RatingEvents have a “rating” attribute between -1 and 1. A MessageEvent contains a comment field for an Event. Users are linked to an Event by attending it. Lastly, Events have different Tags associated with them.

- **URL Routes/Mappings:**

URL	PURPOSE	LOG IN REQUIREMENT
/	Index page, entitled About	
account/	Display the account of the user	User needs to be logged in
events/	List of all the events and a map	
event/<int:pk>	Detailed view of an event	
contact/	Contact page	
event/create	Form to create an event	User needs to be logged in
event/<int:pk>/rate	Form to rate an event	User needs to be logged in
event/<int:pk>/message	Form to message an event	User needs to be logged in
event/<int:pk>/attend	Form to attend an event	User needs to be logged in
event/<int:pk>/unattend	Form to unattend an event	User needs to be logged in
account/messages	List of all the messages by the user	User needs to be logged in
account/ratings	List of all the ratings by the user	User needs to be logged in
account/future	List of the future events of the user	User needs to be logged in
account/past	List of the past events of the user	User needs to be logged in
account/current	List of the current events of the user	User needs to be logged in
event/<int:pk>/messages	List of the messages about an event	
event/<int:pk>/ratings	List of the ratings about an event	
events/future	List of all the future events	
events/ongoing	List of all the ongoing events	
events/best_rated	List of all the best rated events	
logon/	To register as a new user	User cannot be logged in

- **Authentication/Authorization:**
  We were originally going to make our own user model but realized sticking with what Django provided made it a lot easier for us. Our application allows users to register as new users, and then to log in and have access to their account. Once a user is logged in, they can add events, as well as attend/unattend them, or post ratings or comments. Users have access to their different upcoming events, as well as past. They can also access their messages, or their ratings, which they can modify.

- **Team Choice:**
  We decided to use the Google Maps API. Our application displays a map that allows users to easily discover and locate user-created ZooMaps events in their area through a map. We thought this made our app stand out from just being a basic list, and emphasized the local features we built. We also creatively used the single character labels that the API allows you to add to markers by adding emojis, which makes it much easier to find events compared to a bunch of generic pins. Users can also drop pins on the map to easily add locations for new events, which makes for a far better user experience than calculating precise latitude and longitude coordinates on your own. The map makes our app more engaging and easier to use than a simple list of events, and also gives us good experience with working with third party APIs and managing API keys.

- **Conclusion:**
  Overall, we learned a lot and made a cool project that we were excited to use. None of us had any prior experience with Django, so learning all of the specific oddities around it took some time. Passing the correct data to views and making sure our models were working was especially challenging. We now feel much more comfortable exploring other web technologies, and have talked about building more projects in the future.

Individual Write Up

- Miles Black: I worked on improving the general design of the website's front end. Part of what I did on the design includes making the website scalable to different devices, adding icons, adjusting fonts, correcting text, positioning of content, and making the website look generally more pleasing. I also did work on the ratings feature and displaying algorithms for current/past/future events. Aside from that, small logic/syntax fixes in the models and views file. I think I did 30% of the work for this part of the project.

- Jordan Chen: I added linting functionality to the ./run.py script, answered tech support questions when teammates encountered errors, wrote a suite of unit tests for the models, and made the README more extensive. I also spell-checked and grammar-checked parts of the documents and templates. I believe I contributed about 15% of the work.

- James Curry: I spent time figuring out how we were going to implement the map on the main page. We had talked about different ways to store location data, but decided placing markers of Google Maps would make the most sense. This required changing the templates as well as the views in order to get the right data. I also added a location picker when adding an event to make that process easier, and a map on each event page to make the location more clear. I felt like I contributed 30% of the work.

- Mégane Michaud: I created the best rated view, url and template, to display the best events to the users. I also helped James with the json file of the best rated events, to create the pins on the Map, because the format was different than the other views. I also created a view to have new users register. I made some changes to our models, such as deleting our first User Model. Lastly, I made sure that the detailed template of an event would only offer the user to attend if the event was in the future, which required me to modify the models. I think that I contributed 25% of the work.