Etc-Etc-Etc

Rendezvous

Fall 2017

Overview

Rendezvous is a site that allows users to find and join projects that they would be interested in collaborating on. It is essentially a way for passionate people to find other passionate people to collaborate with. Users can post projects that they would like to work with people on, designating certain desired skill sets and such, and other users can join the project. This site extends across many disciplines, as these projects can range from tech projects to performances and everything in between.

Team Members

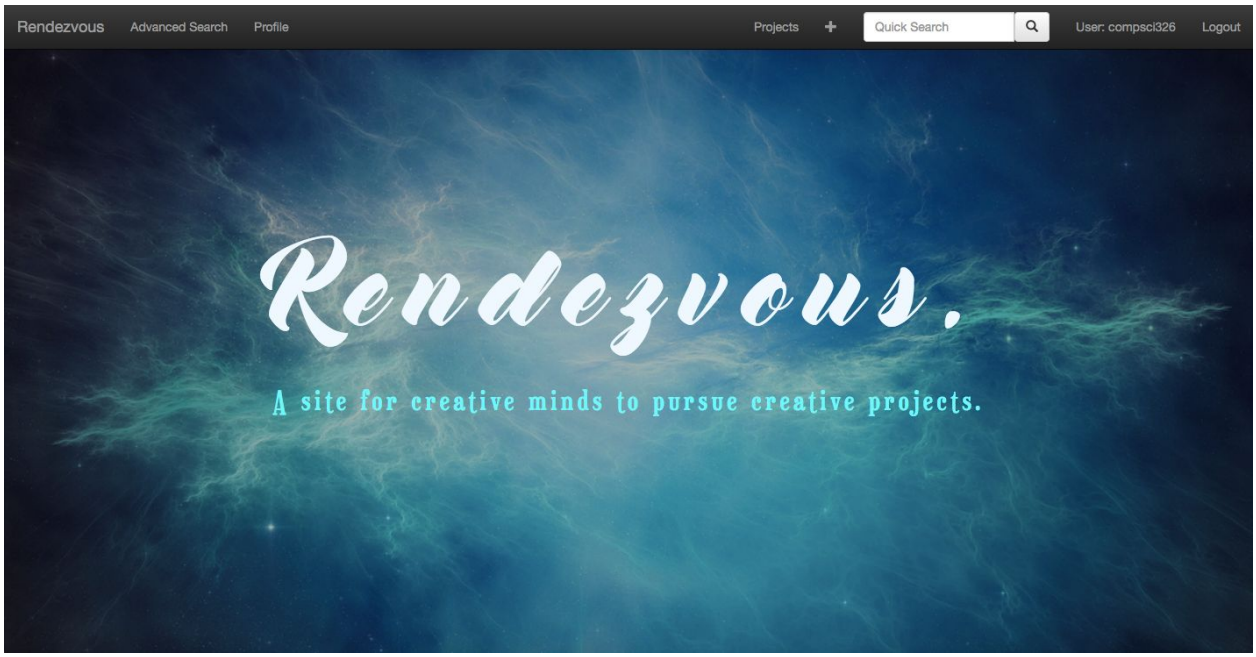
Leonardo Costa, Samantha Cote, Robin Wu, Matthew Kelley, Ibiyemisi Gbenebor

Github Repository

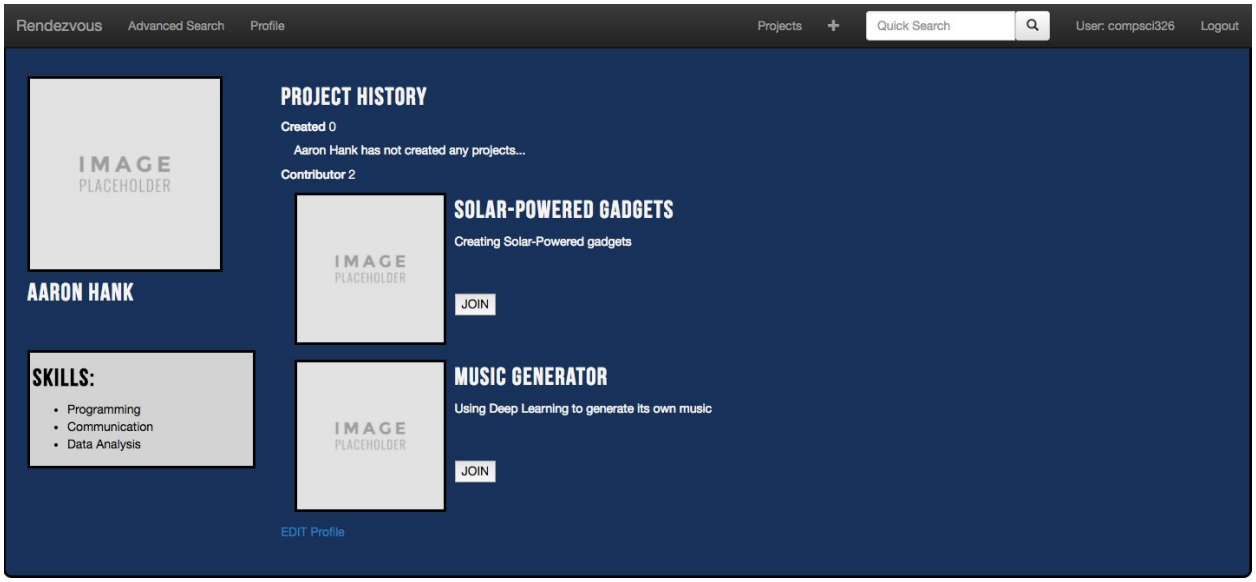
https://github.com/LOCosta/CS326-Etc-Etc-Etc

User Interface

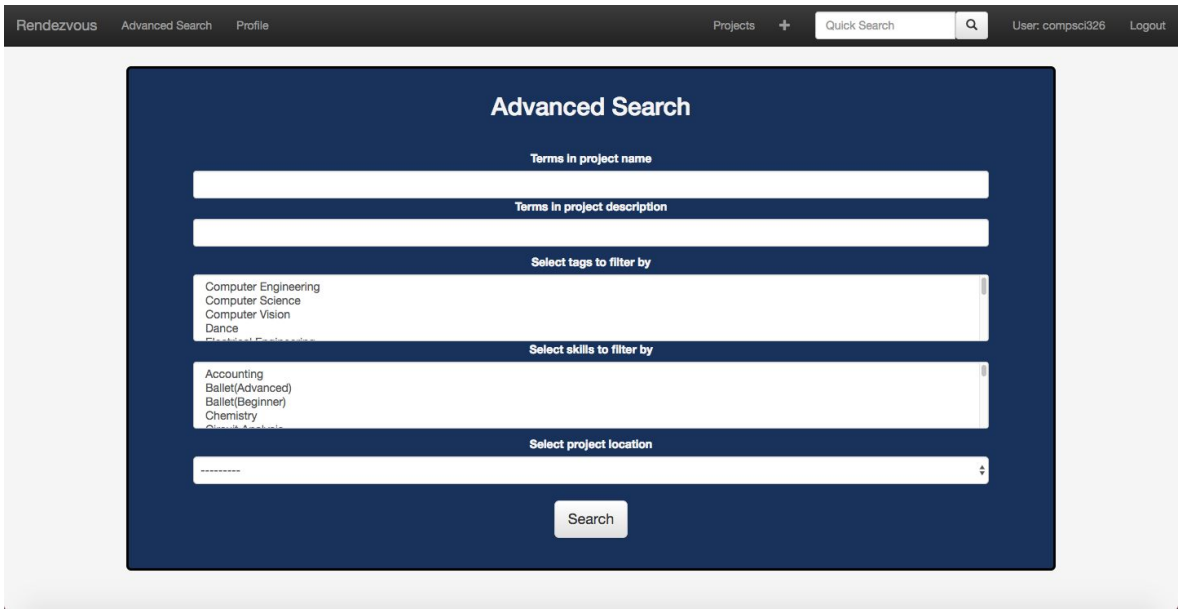
The ***index page*** is the first page the user sees. From here, they can use the navbar to either login or view and search for projects.



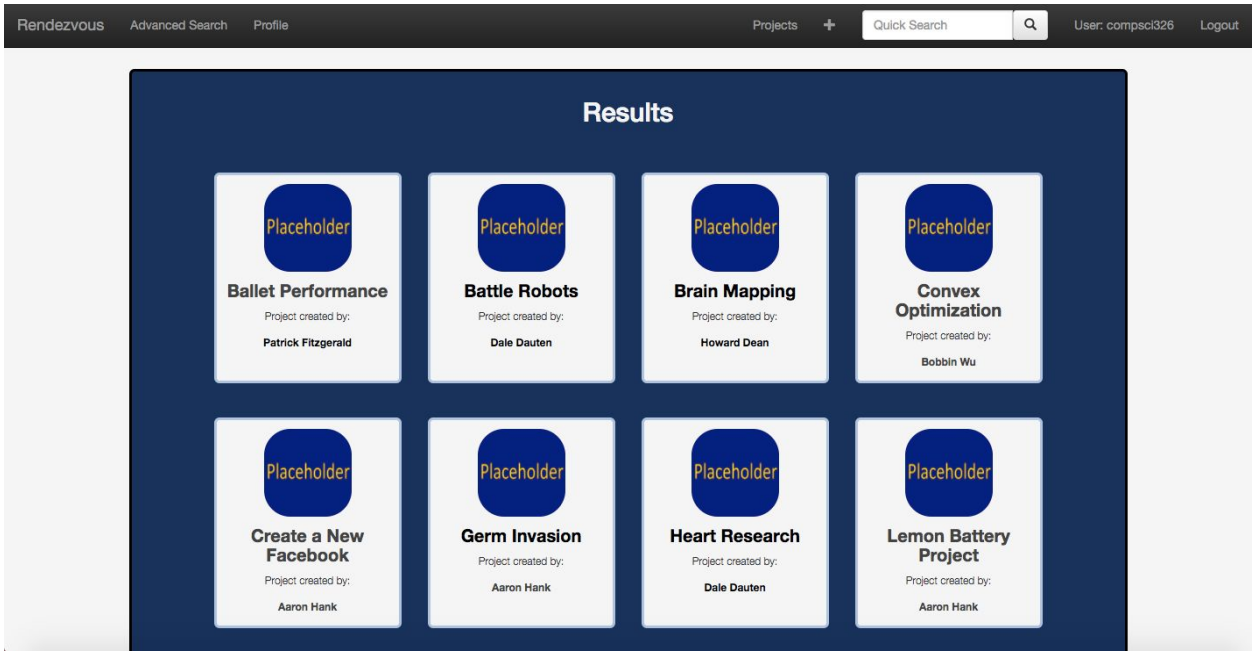
The ***profile page*** appears after a user has logged in. From here they can update the information about themselves (name, email, skills, etc) and they can see all of the projects that they have created and collaborated on.



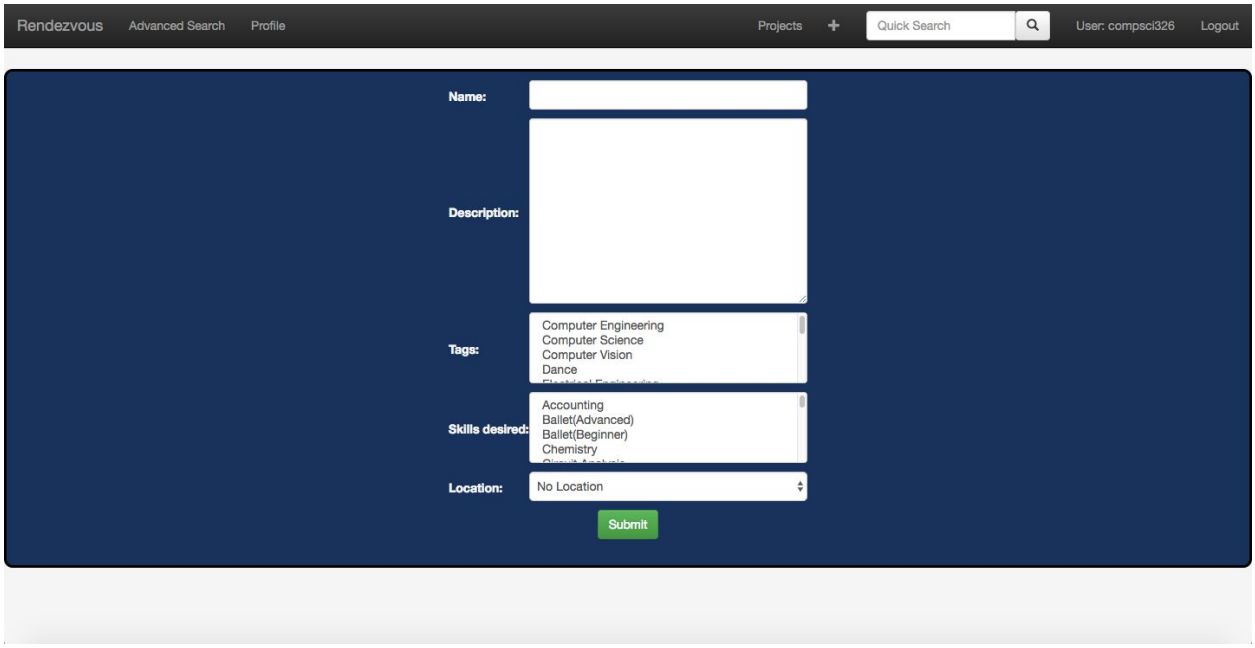
The ***search page*** allows users to search for relevant projects by keyword, title, or even distance. The search page is split into two parts, the form and the actual results page. The form is what allows users to input their search terms/filter by certain tags/skills/location. After they hit the Search button on the form, they are taken to the results page, which shows the projects matching their search query, and is paginated (if there are enough results to require more than one page).



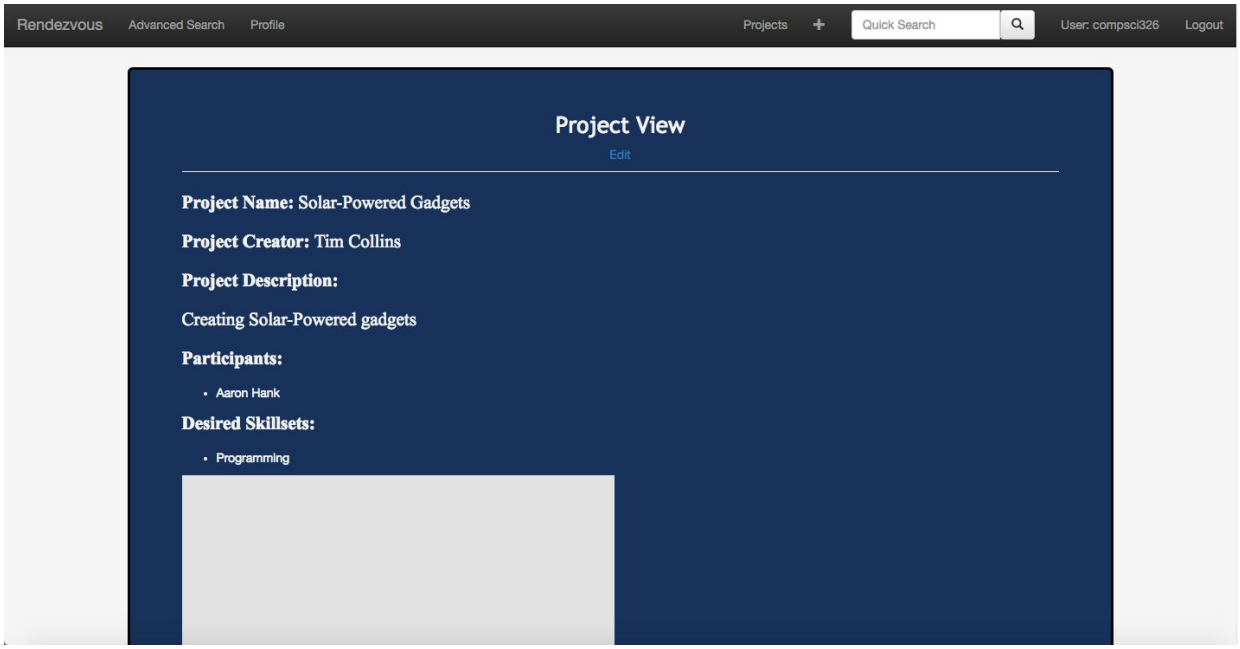
The ***search results page*** is a list of all of the projects that align with the things you searched for. From here, you can click on individual projects that you want to read more about or that you want to potentially join.



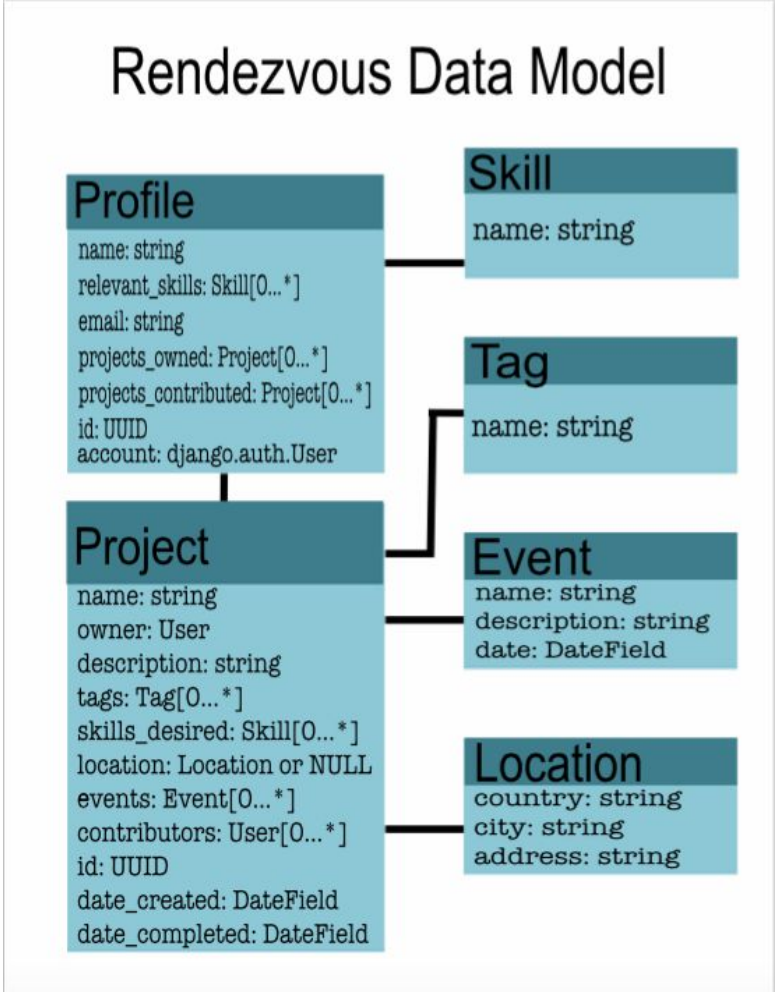
The ***Create New Project page*** allows a user to submit a new project idea. They can fill in relevant information about the types of people they're looking for and a description of what they are hoping to do.



The ***project view page*** is the page the user sees when they have clicked on a specific project that they have found. From here, they can see all of the information about the project, and they can click JOIN at the bottom to officially join the project and become a collaborator.



Data Model



Profiles are specific users of the website. They can create Projects, or they can join Projects. A Project's owner is a Profile and a Project's contributors is a list of Profiles. Each Profile has an associated list of Skills and Projects also have a relevant list of Skills. Projects have a list of Tags, which are essentially keywords for describing it. They can have Events if a Project is to occur at a specific time. And they have a Location so users can search for Projects close to them.

URL Routes/Mapping

```
urlpatterns = [
    url(r'^$', views.index, name='index'),
    url(r'create-project/', views.ProjectCreate.as_view(), name='create-project'),
    url(r'project/(?P<id>[\d\w]+)/$', views.project, name='view-project'),
    url(r'project/(?P<pk>[\d\w]+)/update/$', views.ProjectUpdate.as_view(), name='update-project'),
    url(r'project/$', views.ProjectListView.as_view(), name='project-list'),
    url(r'user/$', views.ProfileListView.as_view(), name='user-list'),
    url(r'user/(?P<id>[\d\w]+)/$', views.profile, name='view-user-profile'),
    url(r'advanced-search$', views.search_form_request, name='advanced-search',),
    url(r'advanced-search/results', views.search_results, name='search-results'),
    url(r'user/?(?P<pk>[-\w]+)/update/$', views.ProfileUpdate.as_view(), name='edit-profile-user')
]
```

We have created URLs that route the user to different pages of the site with their own respective views. This includes URLs for: viewing all registered users, viewing all created projects, each unique user and project, update forms for each unique user and project, home page, project creation page, advanced search form and a separate URL for search results. The URLs that involve mapping to unique users or projects make use of regular expressions to match all different possible primary keys and IDs. The project creation view cannot be accessed unless the user is authenticated and logged in.

Authentication/Authorization

Users are able to log into our site via the Login link that appears on the right side of the navbar. After they are logged in, users are able to edit any projects that belong to them, and are also able to edit their own profile. Additionally, we have some users (including the compsci326 user) in an administrator-like group that are able to edit any project or profile, even if it is not their own. Logged in users are also able to view the Create a Project page (via the + glyphicon in the navbar that appears when logged in), and are able to log out via the Logout button in the navbar.

Team Choice

For our team choice, we implemented the ability for users to search projects based on description, name, desired skills, tags, and location. Initially we planned to use a trigram-based search, which would allow us to show results similar to what was searched for, instead of exactly what was searched for. The main problem with that approach was that Django only supports trigram-based search for PostgreSQL, and changing the database backend to it wasn't going to work. As a result, we settled for a regular search function with a few things added in (ability to AND together terms, and ability to enclose terms in quotation marks to combine them into one term).

The 'advanced-search' and 'search-results' urls are both a part of the search function, as are the views search_form_request and search_results. As far as the implementation, the search works by taking the parameters from the form, encoding them via Django's UrlDict.urlencode() function, and adding them as an argument to the url. The view for search then extracts those parameters from the GET request and filters based on them. One notable thing is that none of the form fields are actually required, and searching with a search form simply returns all projects.

Conclusion

Rendezvous was a fun project that was both challenging and rewarding. Working alongside the homeworks to implement a site was helpful, because the homeworks did not give us everything we needed, and instead made us figure out much for ourselves. This was a project that emphasized teamwork, and though we ran into some bumps along the road, in the end, the site ended up being a great team effort. Tech issues came up along the way (usually they were Git-related), as they tend to, but we tried to help one another in the process to overcome those.

Overall, while definitely not an easy project, we found it very rewarding and learned much about how the development of a web app is carried out. We think our site would be one that people would appreciate and use, and it was fun to put our creative minds to the test to come up with an idea that we thought was both marketable and useful. Fun project, fun team, and fun experience overall.