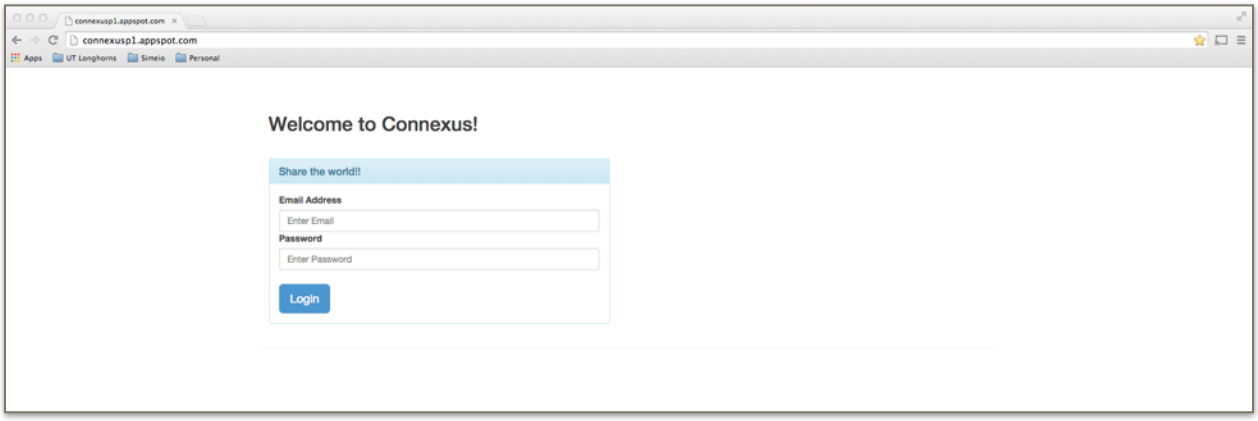
Screenshots of Connexus App (MiniProject Phase 2)

Connexus App URL: http://connexusp2.appspot.com/

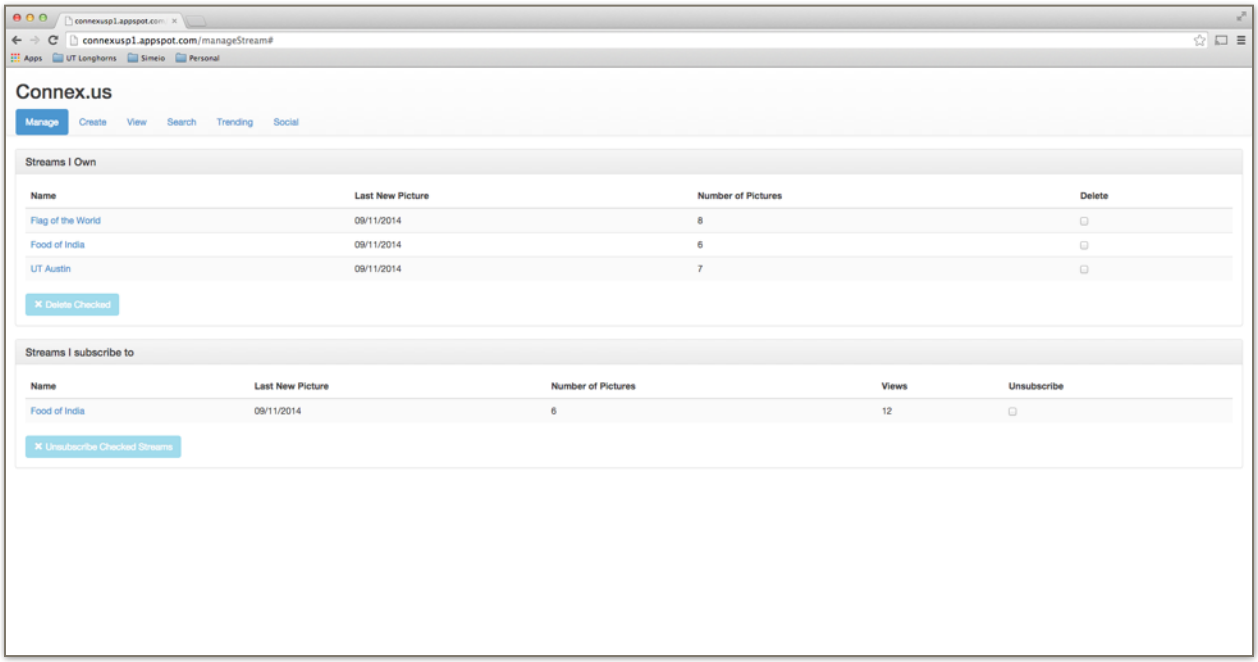
Login Page:

The Login Page shown here is meant only for Display Purposes. This would be the First screen shown to the user on accessing the application. Once the User clicks on the Login button (even without entering any values), they will be redirected to the Google Authentication for logging into the Application



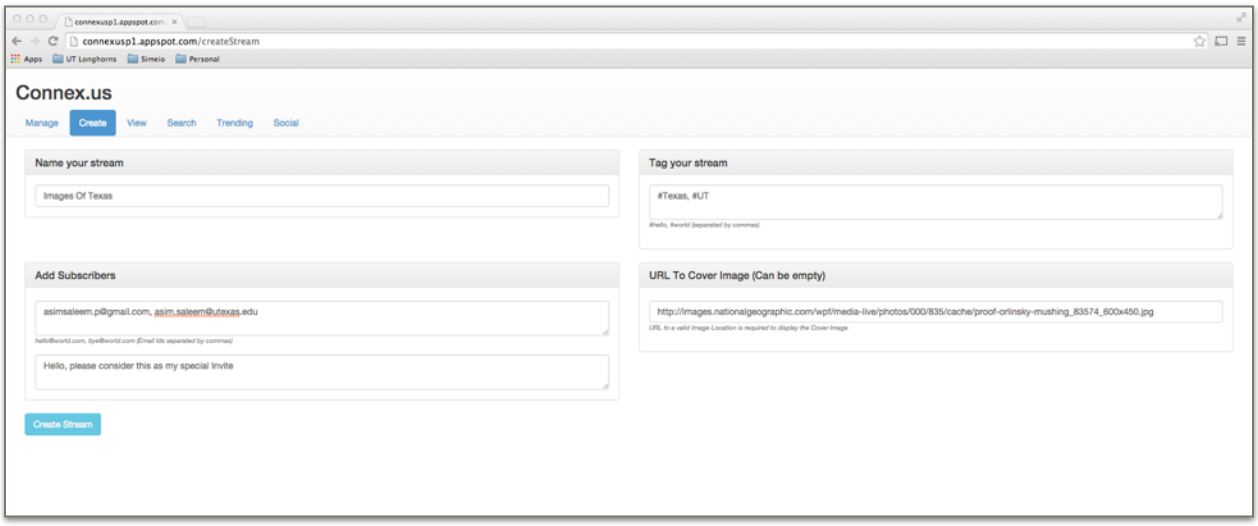
Management Page:

User will redirected to this page, once they are successfully logged in. They can view all the stream they Own or Subscribe to



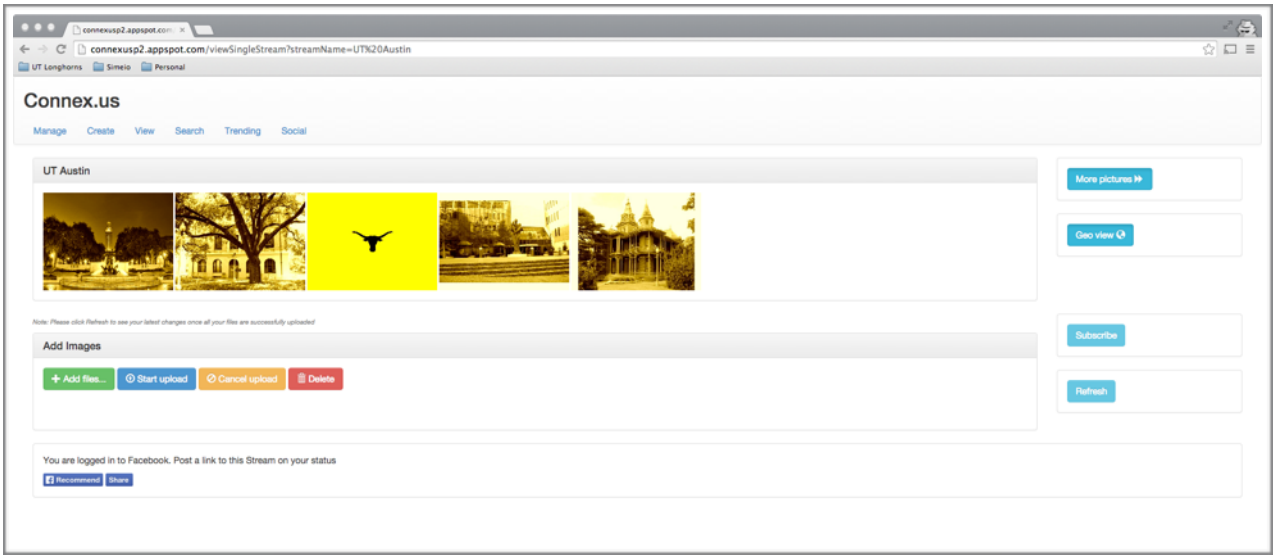
Create Stream Page

User can click on the "**Create**" tab to open this screen and create a new stream and provide the requisite details that is needed



View A Single Stream Page

Users can access this Page by clicking on a stream name in the Manage Stream page or on the search result in the Search Stream page.

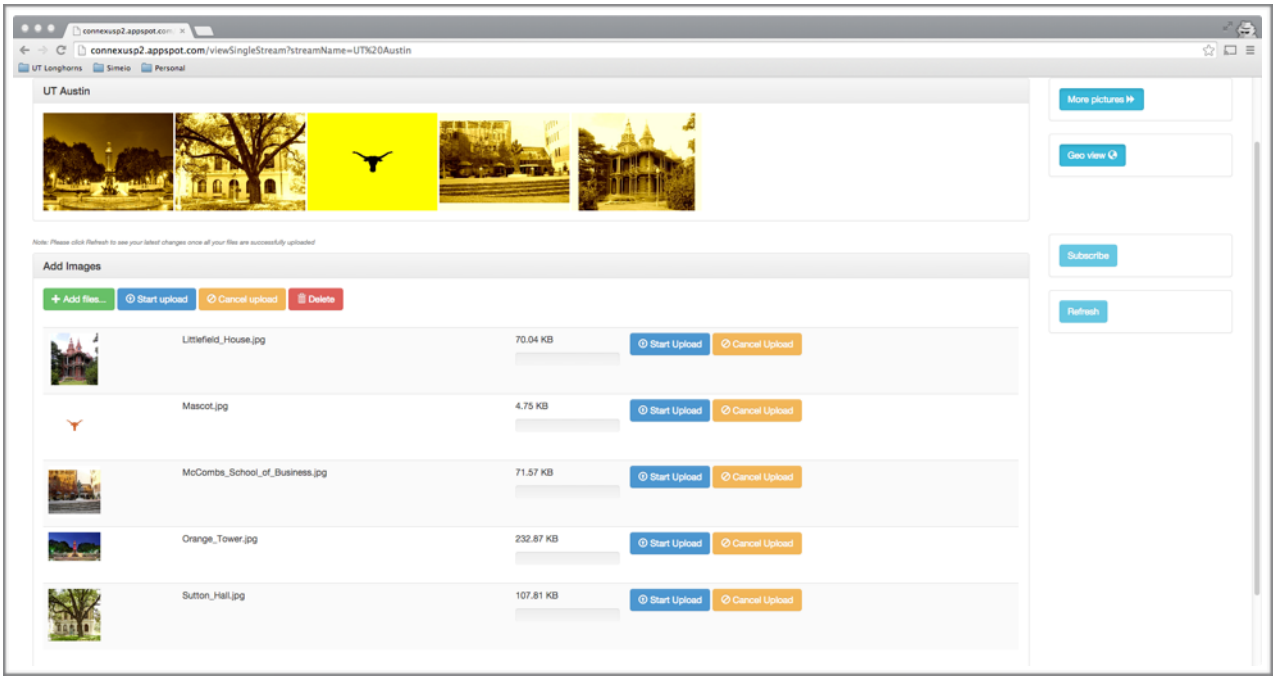


In this version of the Connexus Application, this page has been enhanced with the following features:

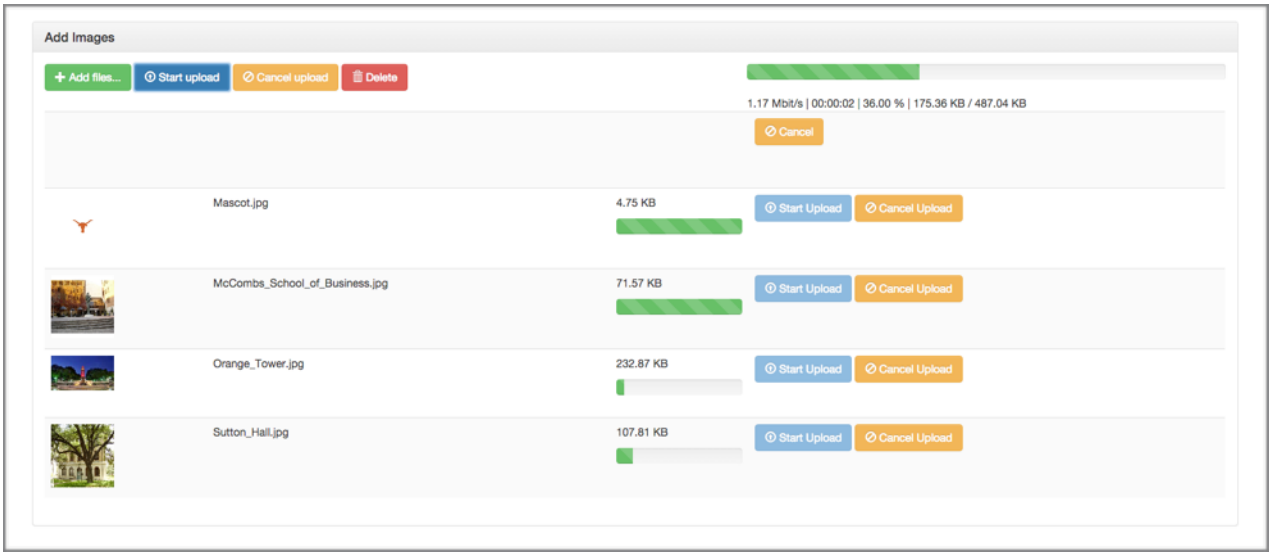
1. A new File upload handler has been added. This handler is capable of uploading multiple images to the server at the same time. It will also show a progress bar indicating the status of the upload process.

2. A new refresh button has been added to the page. This is needed to refresh the screen and view the images that have been uploaded to the stream using the File uploader.
3. A new feature for viewing the images on the map has been added. This feature can be accessed by clicking on the new “Geo View” button added to the View single stream page
4. A Date wise slider option is also provided in the GeoView screen so that the user can filter the images based on a specific date range. By default the date range is set to the maximum value of Current Date and the minimum value of a year less than the current date.

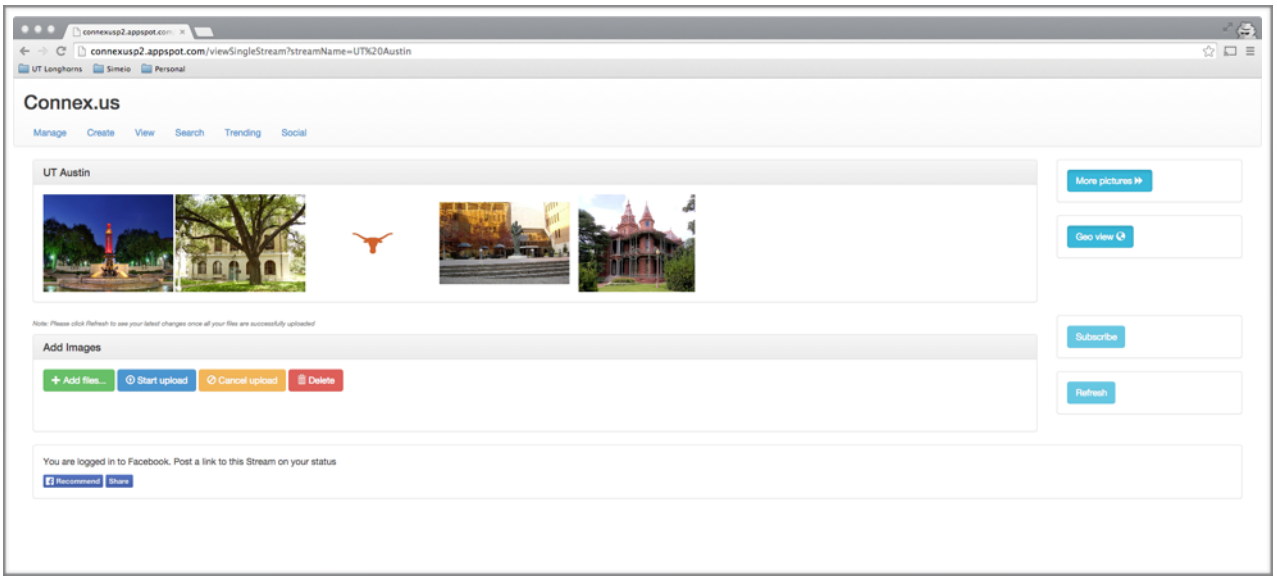
A sample screenshot of how the upload looks when multiple files are added or dragged and dropped into the screen is shown below:



Now when the “Start Upload” button is pressed, the images start getting uploaded to the server and a Progress bar displays the status. A sample screenshot is given below:



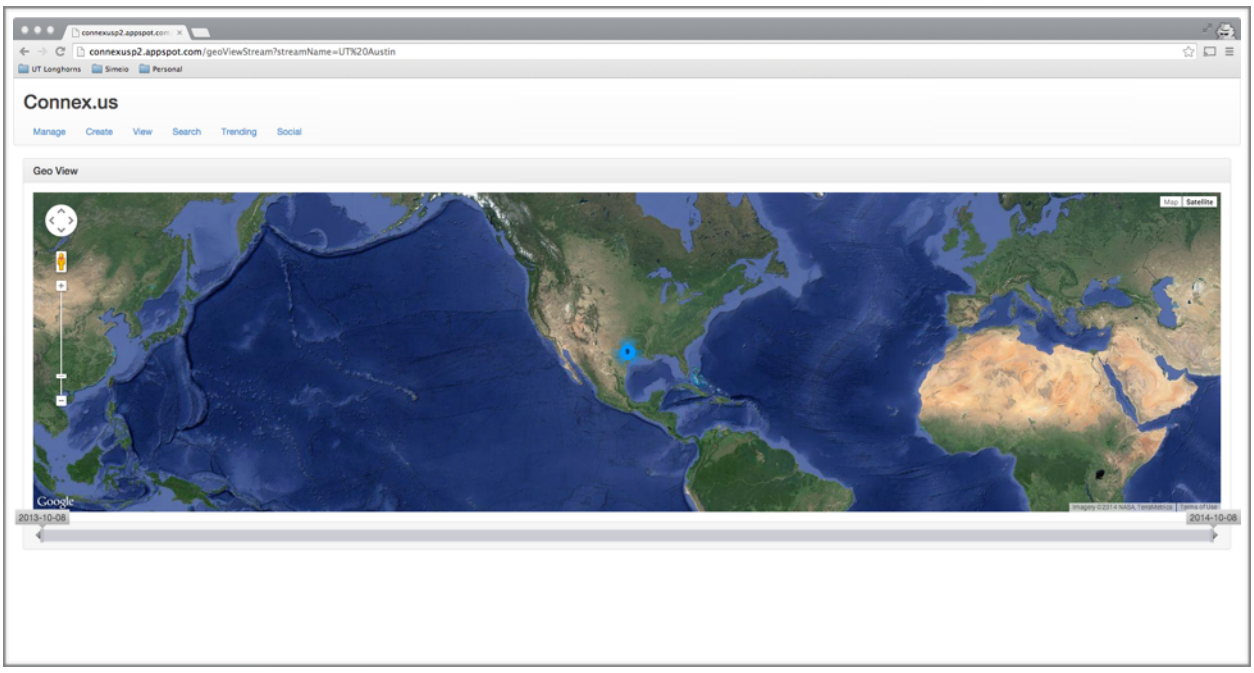
After all the selected files are uploaded, when the Refresh button is clicked, the images can be seen in the same page, as shown below:



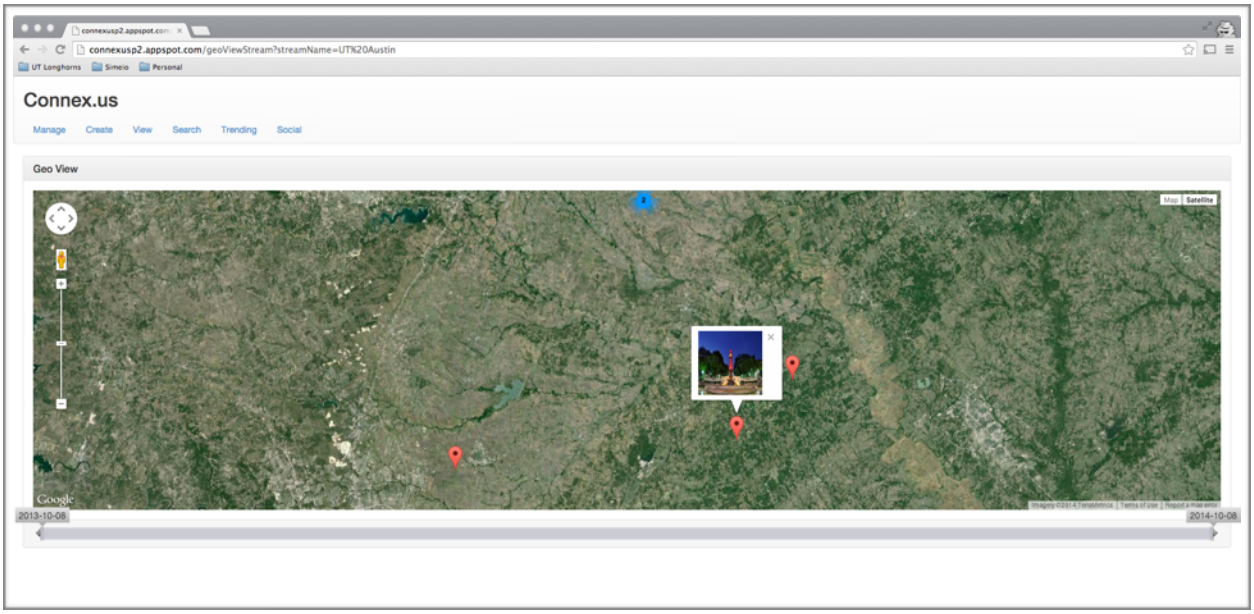
Note: Since the images don't have the location encoded, the browser location is saved as the location of the image.

In order for this to work efficiently, please allow your browser to share your location. User's can then click on the Geo View button to view the images in the embedded locations.

Initially the user is shown the clustered view of the markers pertaining to various image locations. The Clusterer displays a count of the number of images present in the vicinity of that location



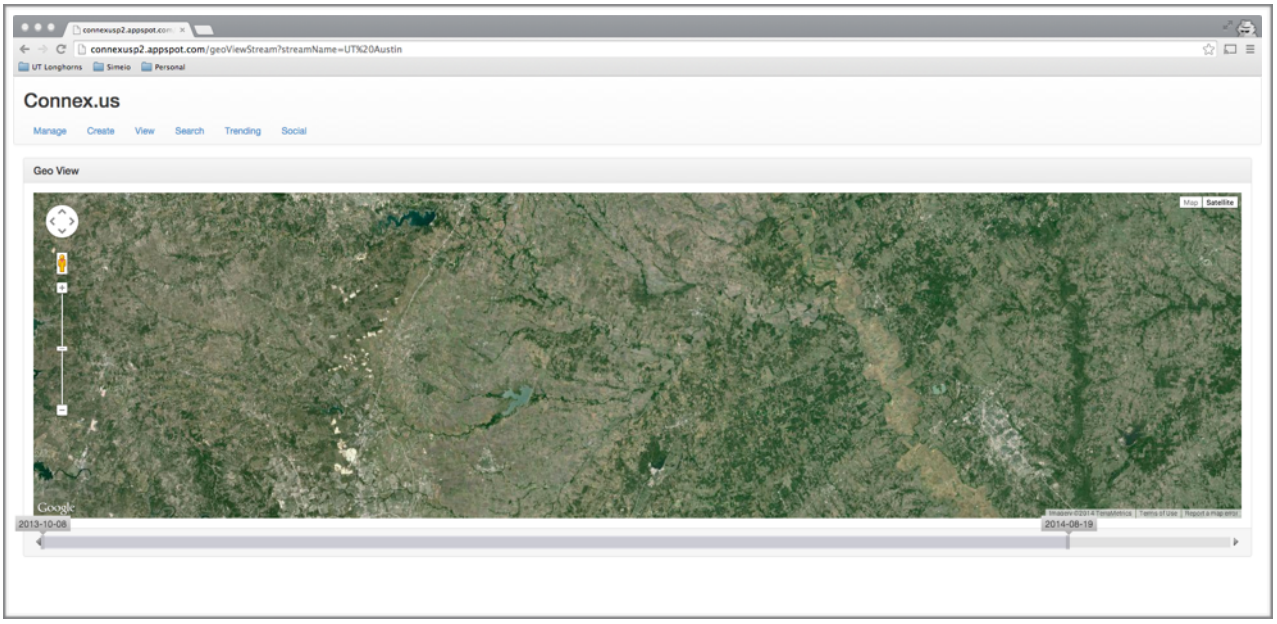
User's can zoom into the map to see the individual markers and on mouseover on top of those markers, the corresponding image will also be shown to the User as shown below:



Also as explained earlier, the Date range slider can be seen in the GeoView stream defaulted to Max value of Current Date and the Min value of 1 year less than the Current Date.

Note: Please note that the third party API provided for Marker Clusterer is buggy. It does not display the Markers if they are all uploaded from the same location, unless either the zoom level is set to the value {maxZoom: 15} or the images are loaded for different locations.

Therefore in order to show the functionality, the current code randomizes the latitude and longitude values by a small percentage for display purposes. This helps us avoid the bug in the MarkerClusterer plugin



In order to verify the operations of a slider, we can change the date to a range where no images were uploaded. It can be noticed that the Clusterer disappears from the Map. This happens because no results are returned for it to display.
The images reappear when the slider is set to a date range value where an image was uploaded.

View All Streams

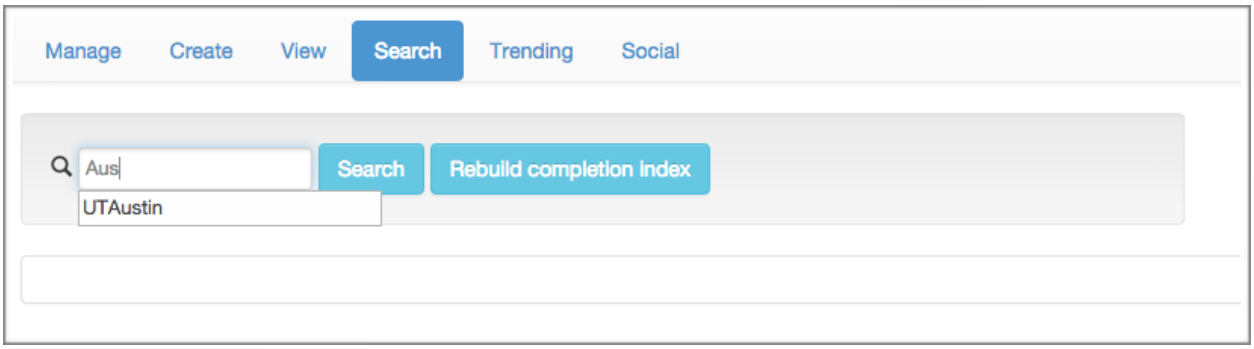
On clicking “View” users will be shown all the Streams that exist in the system. Please note that this is a read only screen and there is no requirement to allow Users to open individual streams from here

Search Streams Page

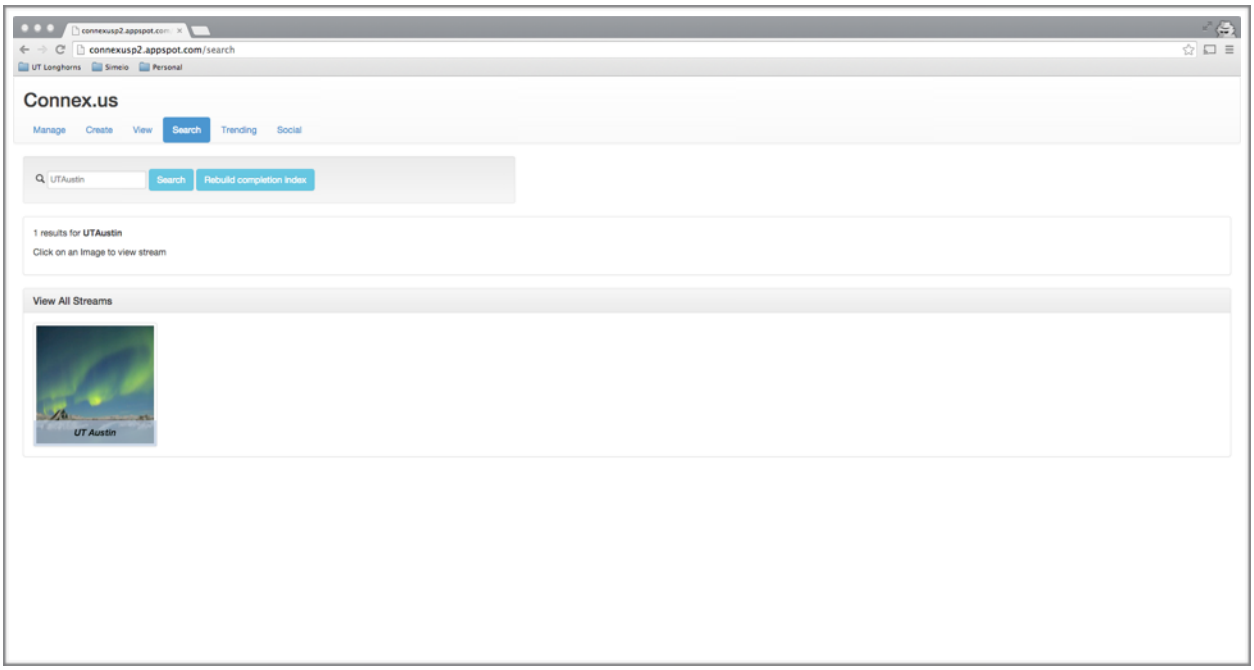
Users are allowed to search for different streams in this page, based on various parameters they provided during the stream creation process. This page has been enhanced further to provide a Jquery Autocomplete feature. Basically when the User starts typing in, the server will retrieve the corresponding matches for the given set of alphabets.

The purpose of the Typeahead feature is to help users with their search by providing them suggestions.

An example of the usage of Autocomplete is shown below:



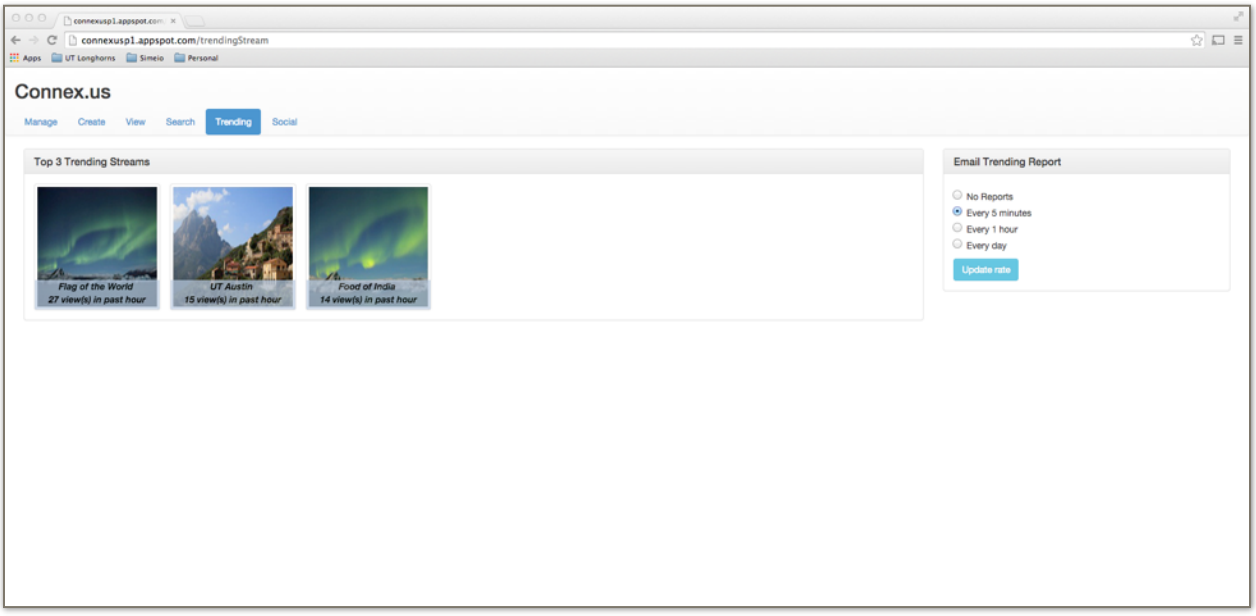
Now when the User selects one of the suggested results and then clicks on Search, it will go ahead and provide the results corresponding to that.



It has to be kept in mind that the search term should match one of the streams that exist in the system. This way we ensure that only relevant streams are suggested and displayed, if selected

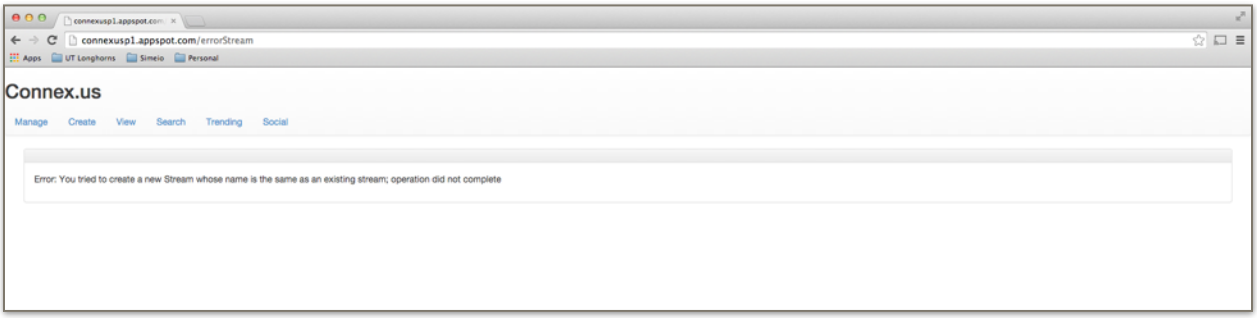
Trending Streams Page

As per the requirement, the images that have been clicked in the past 1 hour will be displayed here. If there are no clicks in more than an hour, this screen would remain empty. This screen also contains the notification options for email intimation of the trending streams



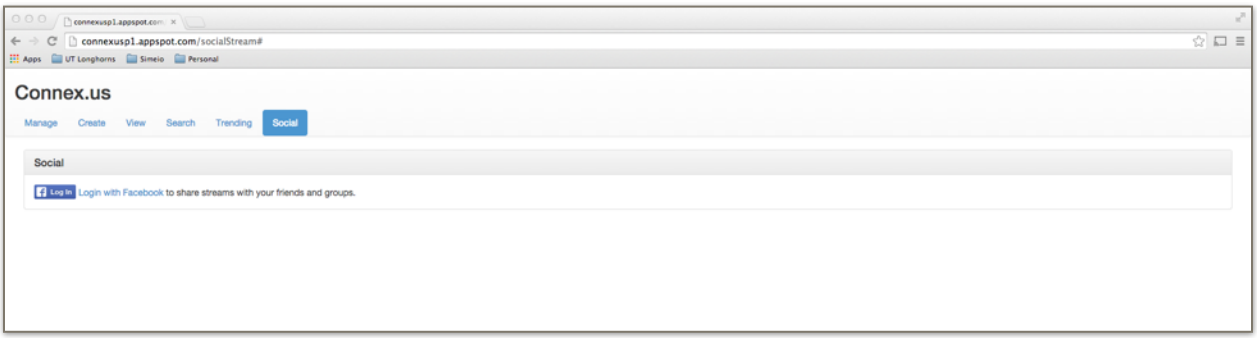
Error Page

If a User tries to create a new Stream with the name of an existing stream, then they would be redirected to this error page



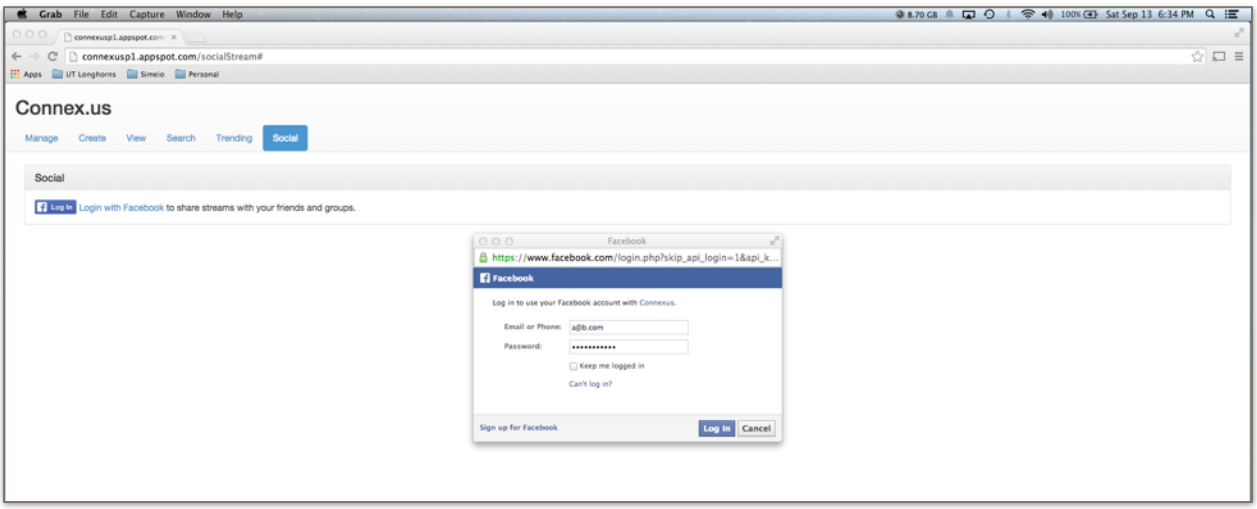
FB Authorization Page

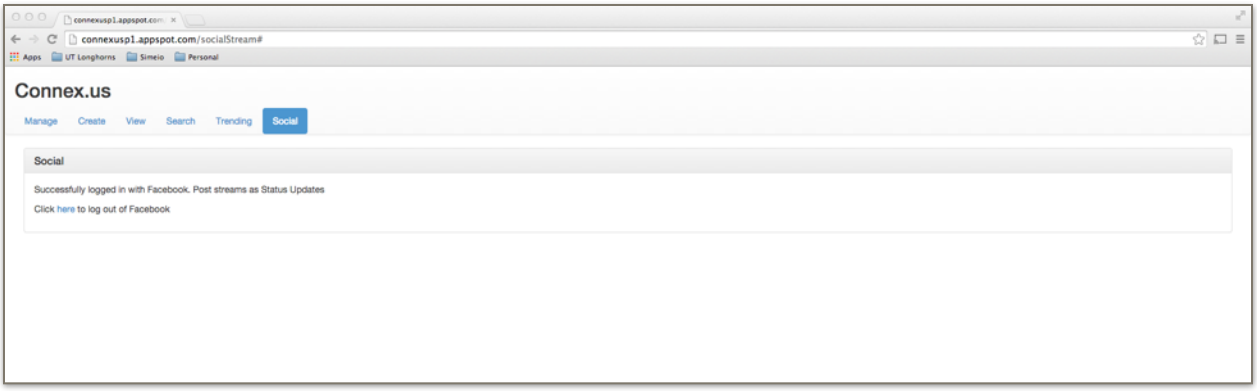
Users also have the option to login to Facebook by clicking on the “Social” tab and logging in using their regular FB credentials



FB Successful Authorization Page

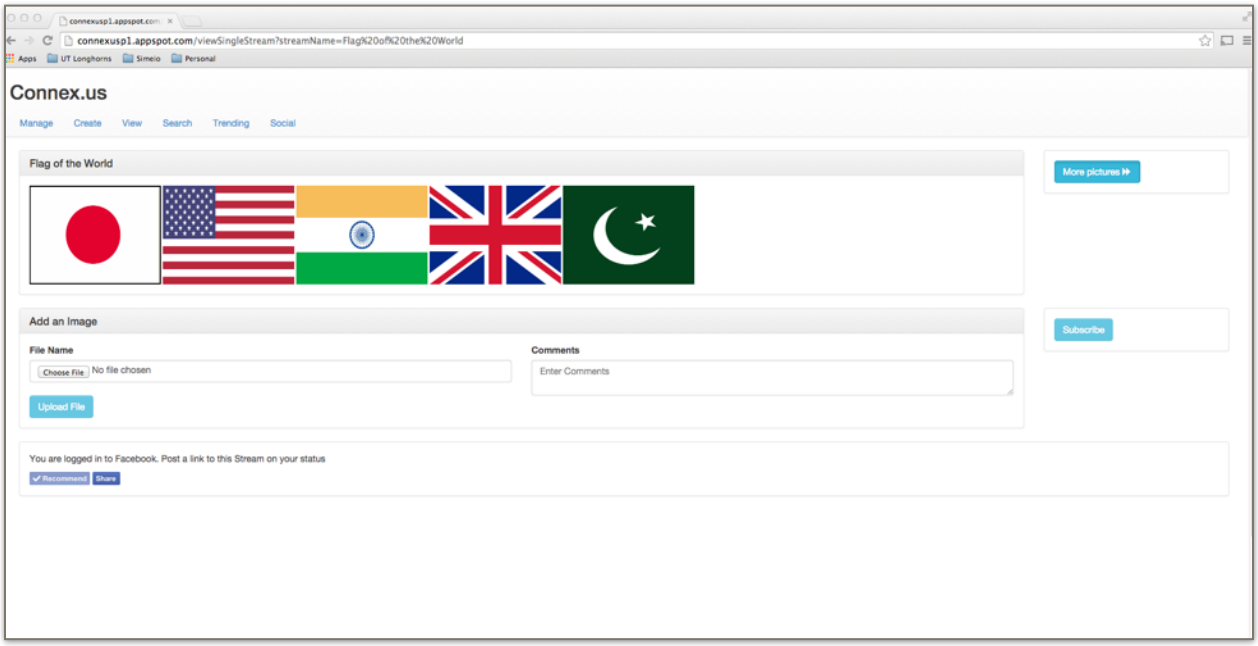
On successful login to FB, the users will be displayed the below message to indicate a successful login





View Stream with FB Post Link Page

If a User is successfully logged in to FB, they will be displayed an option to post the stream link to their FB page when they access the single stream page



Top 5 Things Learnt:

1. One of the things I learnt in this phase of the project was the usage of the plugin for File Upload (http://blueimp.github.io/jQuery-File-Upload/). It was very interesting to see how the code was written to handle the various functionalities like displaying thumbnails, progress bars etc.,
2. Second would be the use of the Jquery Google Map UI plugin(http://jquery-ui-map.googlecode.com/svn/trunk/demos/jquery-google-maps-clustering.html). I found it very informative to see how the markers are set up and displayed to the users
3. The plugin to control the date range functionality was also very interesting, especially how we could conveniently handle the event triggering process once the sliders reaches a final state at that point
4. The Jquery autocomplete feature was a very useful feature to learn, especially how it makes it so easy for the Users to search by providing them search suggestions

5. But the most important thing I learnt in this phase was how to make different plugins made by different people work together. It was a challenge because of really poor documentation and buggy code.

Top 5 Mistakes to avoid going forward:

1. The Jquery file upload plugin had really bad documentation and expected user's to know how to use it as is. This caused me to commit a lot of mistakes and I ended up customizing some operations as I couldn't find proper documentation.
2. Spent a considerable amount of time working on the Jquery Google Map UI. Again due to lack of good documentation, I committed a lot of mistakes on getting it configured correctly.
3. Again this had to do with making mistakes in making the plugin's behave well together. Since they were not as clearly documented, I made a lot of mistakes in my understanding of how they work individually first before combining certain features together
4. Understanding how to retrieve the lat, long values took some time to figure out until I decided to go by the HTML5 geolocation approach. It was very difficult to get good documentation of retrieving location information from EXIF data
5. While displaying images on the Map, the marker cluster would not show the images if the upload location was same. This resulted in countless hours of debugging to find out the mistake i was making. All along it was a bug in the MarkerClusterer code

Top 5 Things that worked Out of the Box:

1. Webapp2 didn't give any issues from the beginning as mentioned earlier. It always routed the requests as expected without any issues
2. GoogleAppEngine Launcher worked flawlessly from Day 1
3. Jinja2 also worked out of the box as I knew the basic constructs from other tools even though I had to override some File Upload plugin related functions because they were using the same syntax as Jinja2.
4. All the Python libraries that were needed worked as desired when used the correct way the first time itself
5. Cron Jobs worked as expected too based on the scheduling though I had to turn off the emailing since I was hitting some Google set quota