**CEN 4010 Principles of Software Engineering**

**Fall 2019**

**Team 4**

**Final Project Portfolio**

**The Freelancers**

**Campus Snapshots**

<http://lamp.cse.fau.edu/~cen4010fal19_g04/CampusSnapshot/login.php>

**Members:**

**Aaron Haim (Scrum Master) – ahaim2016@fau.edu**

**Daniel Fradlin (Front End Lead) – dfradlin2017@fau.edu**

**Richard Urena (Back End Lead) – urenar2016@fau.edu**

**Yiyang Liu (Product Owner) – yliu2015@fau.edu**

**Created December 1, 2019**

**Product Summary:**

*Campus Snapshots* is a web system that people can access and inform users about issues and events around campus. It provides real time snapshots via a map that shows the campus status. If an issue arises, it can allow administration to provide help as needed to those markers located on the map. It can also be used to report events happening on campus such as games, concerts, activities, etc. *Campus Snapshots* is made with the intention of allowing campuses to more easily communicate items in an instantaneous fashion throughout the area. It can be used as an itemized list of problems to solve, a way to better organize what is happening, or as a way to send out notifications about specific problems that cannot be solved at that current moment. It is, in a sense, a one stop shop to know any major developments within a campus.

There are some great advantages to implementing such a system. One major advantage is that the student to administrator ratio is about 22:1. With more eyes, more issues can be reported quickly. It allows for the administration to have extra eyes and stay up to date on the markers. Also, with the ability to upload photos of the reported issues, it will allow for issues to be validated and for there to be no false reporting.

Another great part of this system is that if students decide to host their own events, they can add that event to the site for other students to join if they are interested. FAU doesn't have a club for everything so this system makes it easier for students to find others that share the same hobbies/interests.

**Major Committed Functions**

* Students can create accounts if they do not already have one (Administrators are preregistered)
* Users can login to their account
* Users can create markers (events/issues)
* Users can attach photos to created markers
* Users can add comments to an event
* Users can report markers
* Administrators can update status of issues
* Administrators can remove markers
* Students can sort events by location or most recent

**URL:** <http://lamp.cse.fau.edu/~cen4010fal19_g04/CampusSnapshot/>

**Admin Username:** [admin@fau.edu](mailto:admin@fau.edu) **Password:** administration123

**Milestone Documents:**

**Executive Summary:**

*Campus Snapshots* is a web system that people can access and inform users about issues and events around campus. It provides real time snapshots via a map that shows the campus status. If an issue arises, it can allow administration to provide help as needed to those markers located on the map. It can also be used to report events happening on campus such as games, concerts, activities, etc. *Campus Snapshots* is made with the intention of allowing campuses to more easily communicate items in an instantaneous fashion throughout the area. It can be used as an itemized list of problems to solve, a way to better organize what is happening, or as a way to send out notifications about specific problems that cannot be solved at that current moment. It is, in a sense, a one stop shop to know any major developments within a campus.

**Competitive Analysis:**

|  |  |  |
| --- | --- | --- |
| **WAZE app** | **FAU Mobile App - Maps** | **Campus Snapshots** |
| Traffic update | Search campus buildings | Report campus issues and events |
| Speed Traps | View popular locations | Show status report of instance |
| Accidents |  | Information about all current events/issues |
|  |  | Search campus buildings |
|  |  |  |

There are some great advantages to implementing such a system. One major advantage is that the student to administrator ratio is about 22:1. With more eyes, more issues can be reported quickly. It allows for the administration to have extra eyes and stay up to date on the markers. Also, with the ability to upload photos of the reported issues, it will allow for issues to be validated and for there to be no false reporting.

Another great part of this system is that if students decide to host their own events, they can add that event to the site for other students to join if they are interested. FAU doesn't have a club for everything so this system makes it easier for students to find others that share the same hobbies/interests.

As for the information in the table, there really is not a major competitor in this field since most campuses handle this information on their own. The closest campus app has to be the FAU Mobile App while the closest global app is the WAZE app. Neither of these have any sort of reporting system implemented or have any major way to update user sightings automatically. Our web system allows students to report issues and events to a database instantaneously and updates on the map. You can check on administration marked statuses to view the progress of how the issue is being handled or what event is currently going on. You can also read the information provided by administration and/or students to see what the issue or event going on is. Finally, there is a way to search the campus buildings as the FAU Mobile App does but with the power to see what is specifically happening in that building. This is not a standardized industry allowing for even basic implementation to have massive benefits with the use of students participating in spreading the word.

**Data Definition:**

* *Campus Snapshots*: Name of the product
* Students: An access point attending a campus that can create markers and upload photos with them
* Administration: An access point working at a campus that has the same roles as Students along with the ability to mark status updates on the marker and/or delete them
* Users: Refers to both Students and Administration as a whole
* Campus: The location currently being monitored by the web system
* Web System: A storage database map that holds all markers on a certain campus
* Snapshot: An instance in time where a marker has occurred and has been updated to the web system
* Marker: An issue or event created by the users
  + Location will refer to the closest building to the marker
  + Image upload can be in PNG/JPG format and at least 480p

**Overview, Scenarios, and Use Cases:**

Users create an account. Students log in and can view current events or issues happening around campus. Students may create a new marker and report an issue. Administration then sees the new marker and provides status update on that issue based on their progress. If there is a duplicate reporting then Administration can remove the duplicate report. Students can also create a new marker for an event they are hosting. Students can report any inappropriate posts and provide a reason. Administration has privileges to remove any inappropriate posts if policies are violated. Users can add comments on events. Administration can update the status of any issue similar to comments. Markers are structured like cards that are split into its two categories: Events and Issues. The user interface is simple to allow for maximum understanding. Student and Administration login are split into two separate pages.

**High-Level Functional Requirements:**

1. Users should be able to create an account if they do not already have one.
   1. A valid email address is required to create an account.
   2. A password must be at least 8 characters long. If not, the user will not be allowed to create an account.
2. Users will be identified by their campus provided email.
   1. Any duplicate emails shall not be permitted to create an account.
3. Users should be able to create a marker.
   1. Markers are either specified as events or issues.
   2. Event Name and Time are not for issues as it has no purpose.
   3. Issues Status is not for events as it has no purpose.
4. Users should be able to attach a photo to their reporting and creation of marker.
   1. The picture must be a PNG or JPG file format, all other types will not be taken.
   2. The picture must be at least 480p so the image can have decent quality.
5. Administration should be able to delete/modify markers and comments.
   1. Any deleted markers/comments will be completely removed from the web system.
6. Users should be able to report inappropriate posts.
   1. Students that report inappropriate posts must be reviewed by Administration.
7. Administration should be able to provide status updates of reported issues.
   1. Status updates will either be in the form of comments or its own issue resolver system.
   2. If in the form of comments, previous status updates will be deleted from the system.
8. Users should be able to comment on existing markers.
   1. Comments will be limited to 100 characters.
   2. Students can only post one comment per marker and zero on markers created by that student.

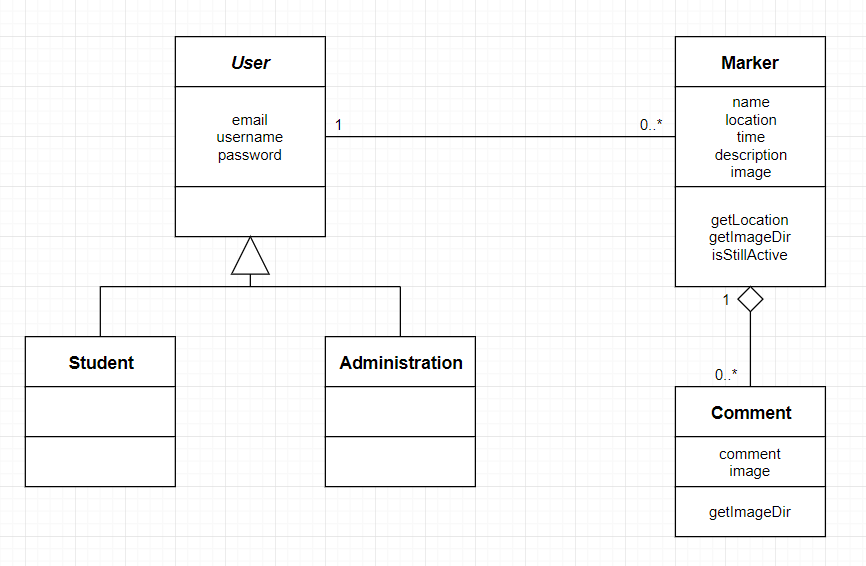
**Non-Functional Requirements:**

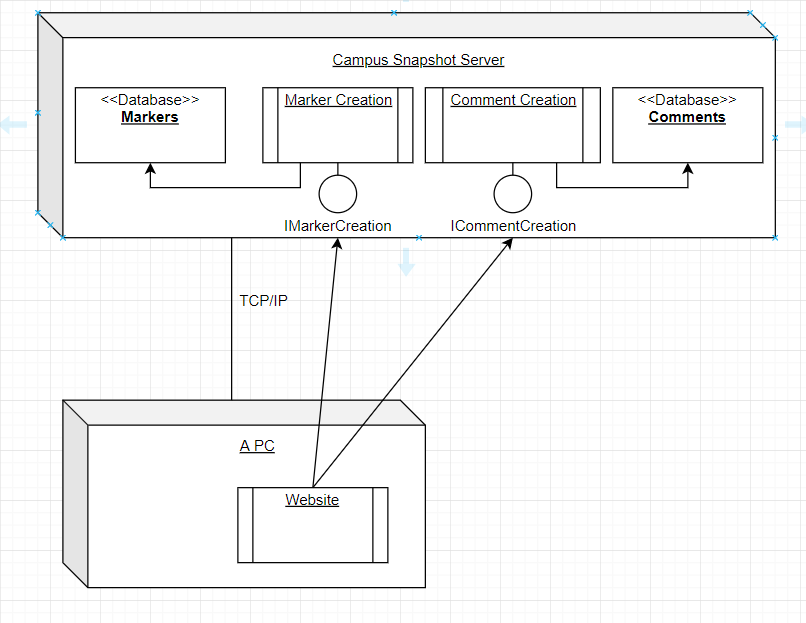
1. JavaScript, SQL, HTML, and CSS will be the main focus languages used to develop *Campus Snapshots*.
2. Users can only have a single account.
3. Users should be able to use the system without instructions.
4. Markers can be sorted/displayed by location on campus.
5. Students should be limited to a specific number of posts per day to avoid spamming the system.

**High-Level System Architecture and Database Organization:**

* Languages
  + JavaScript
  + HTML/CSS
  + SQL
* Tools
  + GitHub
  + Google Docs
  + Trello
  + Discord
* APIs
  + Google Maps API (Dependent on best functionality or just campus map)
  + Custom APIs
    - Marker Creation
      * Creates markers based on User ID
    - Comment Creation
      * Creates comments based on User ID and Marker ID
* Browsers
  + Google Chrome
  + Mozilla Firefox
* Database Organization
  + Users
    - User ID (Unique)
    - Email Address (Unique)
    - Password
    - First and Last Name
  + Markers
    - Email Address (Users)
    - Marker ID
    - Marker Name
    - Marker Location
    - Marker Time (Used for Events)
    - Marker Description
    - Marker Status (Used for Issues)
    - Image Path
    - Reported Amount
  + Comments
    - Comment ID
    - Marker ID (Markers)
    - Email Address (Users; independent of Markers email address)
    - Comment
    - First Name
  + Filtering/Searching
    - Students can only filter markers by location or by most recent
* Images
  + PNG or JPG format
  + At least 480p
  + File System with DB reference format
* Prioritization
  + Most recent events will be prioritized first in the marker display
* User Privileges
  + Students
    - Create markers
    - Upload photos attached to markers
    - Add a comment on marker
    - Report an issue on specific marker
  + Administration
    - All Student privileges
    - Update status on markers
    - Delete markers
* Registration Information
  + Users
    - Valid Campus Email Address
    - Password (8+ characters long)
* Main Information
  + Raw Data
    - Information entered by users
  + Metadata
    - File format and size of images (PNG/JPG; 480p+)
    - Marker Data
      * Event/Issue ID (11-bit integer)
      * Event Name (20 characters)
      * Event/Issue Location (50 characters)
      * Event Time (8 characters; will be switched to date format)
      * Event/Issue Description (255 characters)
      * Event/Issue Image Path (100 characters)
      * Event Reported (1-bit integer)
      * Issue Status (255 characters)
  + Supporting Data
    - User Login Information (email/ password)
    - Location listing

**High-Level UML Diagrams**





**Key Risks To Project**

* **Skill Risks**
  + None at the current moment
* **Schedule Risks**
  + **Potential schedule risks**
    - There is a small potential that some features might not make it into the final product
    - Due to current workflow speeds, however, this issue is very minute and probably will not affect the final product release
* **Technical Risks**
  + No technical risks
* **Teamwork Risks**
  + No teamwork issues
* **Legal/Content Risks**
  + **Google Maps API**
    - **User agreements and Costs**
      * Google Maps API is free to use but does have some user agreements and costs depending on what is used
      * The easiest way around this issue is to just use an image instead of the API, depends on the path taken once we get to that stage of front-end programming

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Risk No. | Description | Likelihood | Impact | Severity | Owner | Mitigating Action | Contingent Action | Progress | Status |
| 1 | Some features (e.g. comments, status updates) might not make it into the final product due to lack of time. | Medium | Low | Small | Richard Urena | Create a basic placeholder implementation that meets the basic requirements to be improved upon if there is enough time. | All documents will be edited accordingly to remove any mention of features or to say they will be added after release to the product. | All features have been added successfully | Closed |
| 2 | Google Maps API is free to use but may have some unfavorable user agreements and/or costs that might not make it usable for this product. | High | Medium | Large | Aaron Haim | Work out all the legal terms and costs that would allow us to use the API in the final product. | Use a static image to represent the map and have dots as holders for locations. Otherwise, remove the feature altogether. | Contingent Actions have been taken to remove the feature. | Closed |

**Team:**

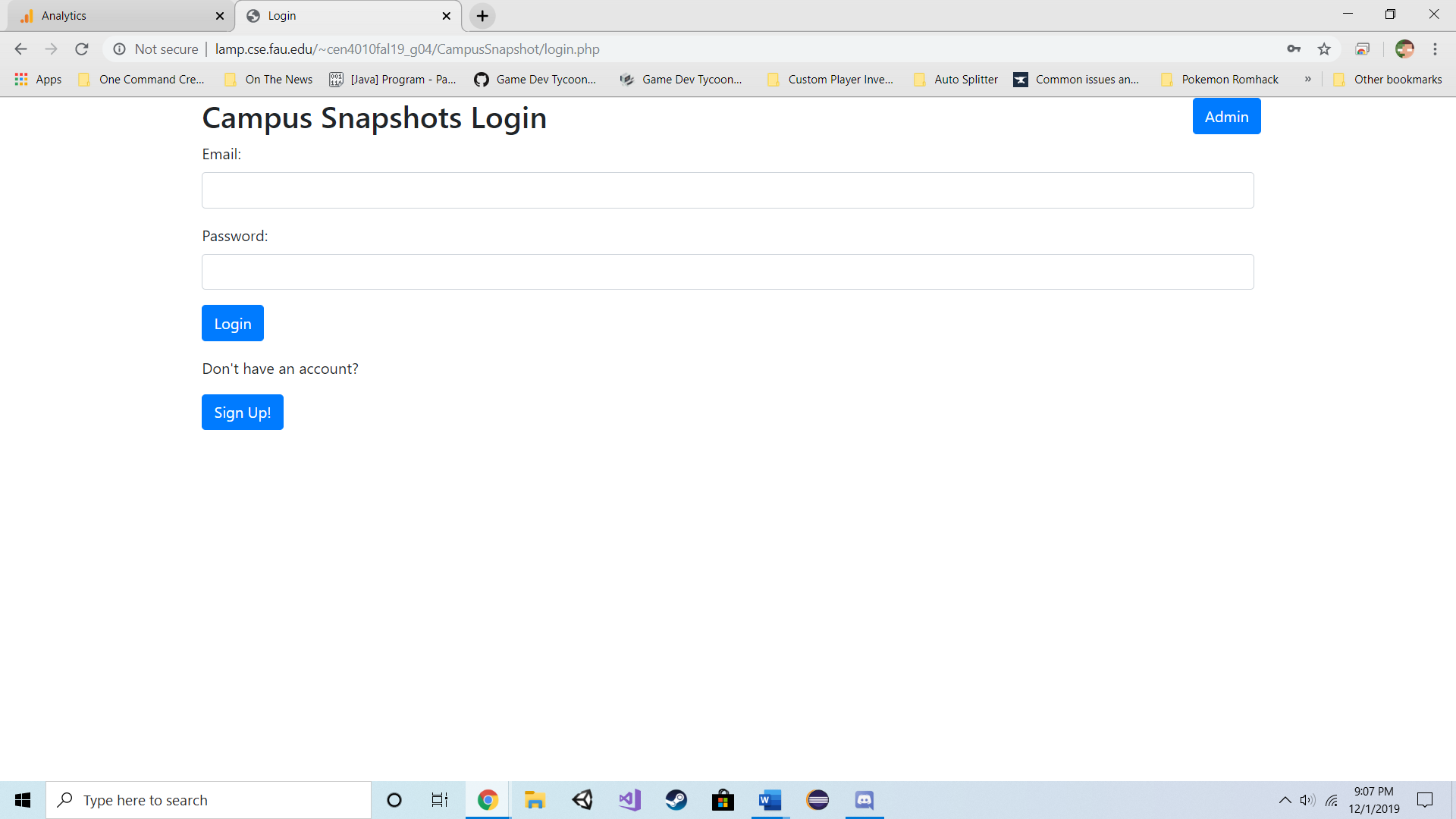
* **Aaron Haim**
  + **Scrum and GitHub Master**
  + **Back End Programmer**
* **Daniel Fradlin**
  + **Front End Lead Programmer**
* **Richard Urena**
  + **Back End Lead Programmer**
* **Yiyang Liu**
  + **Product Owner**
  + **Front End Programmer**

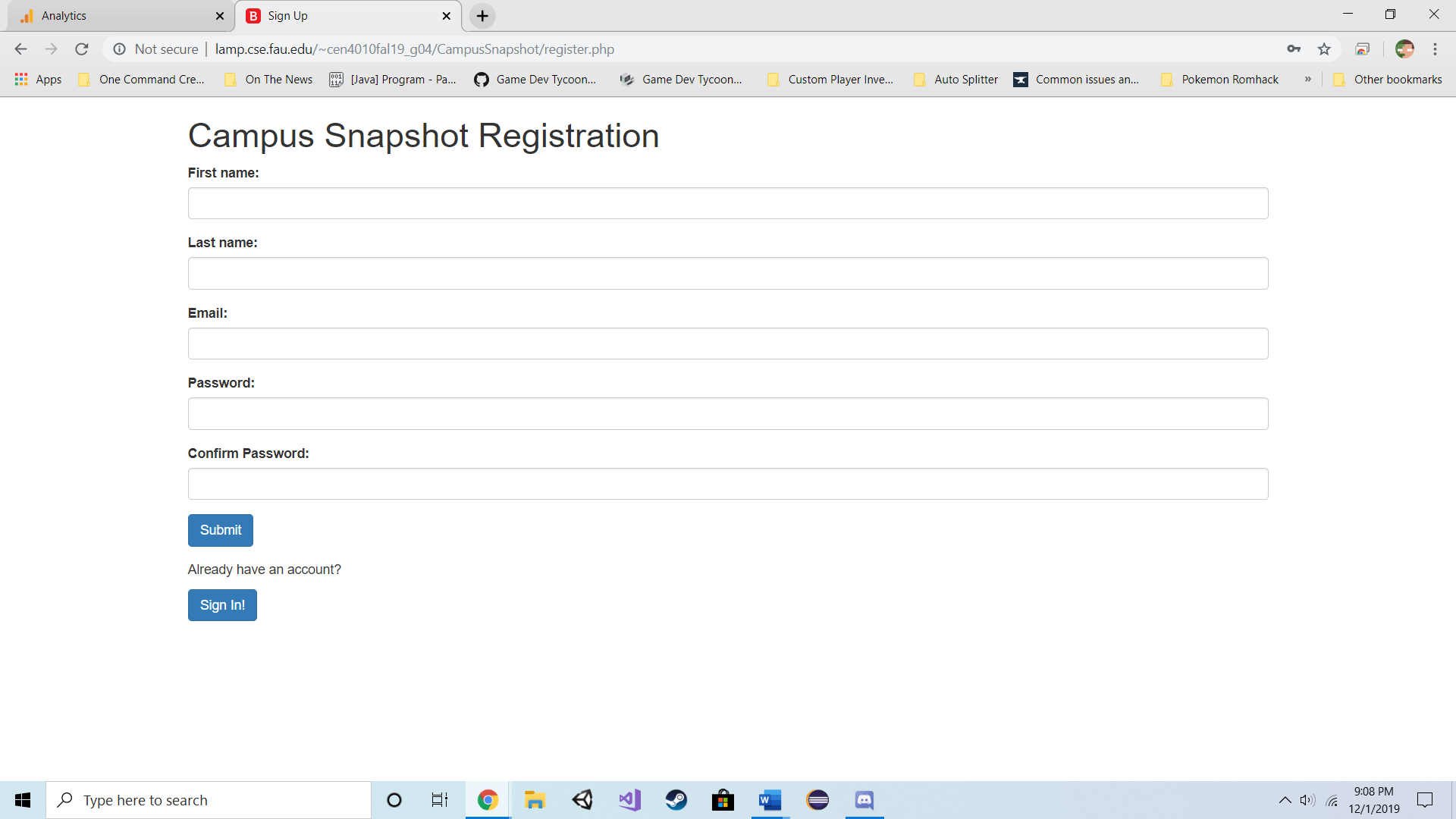
**Checklist:**

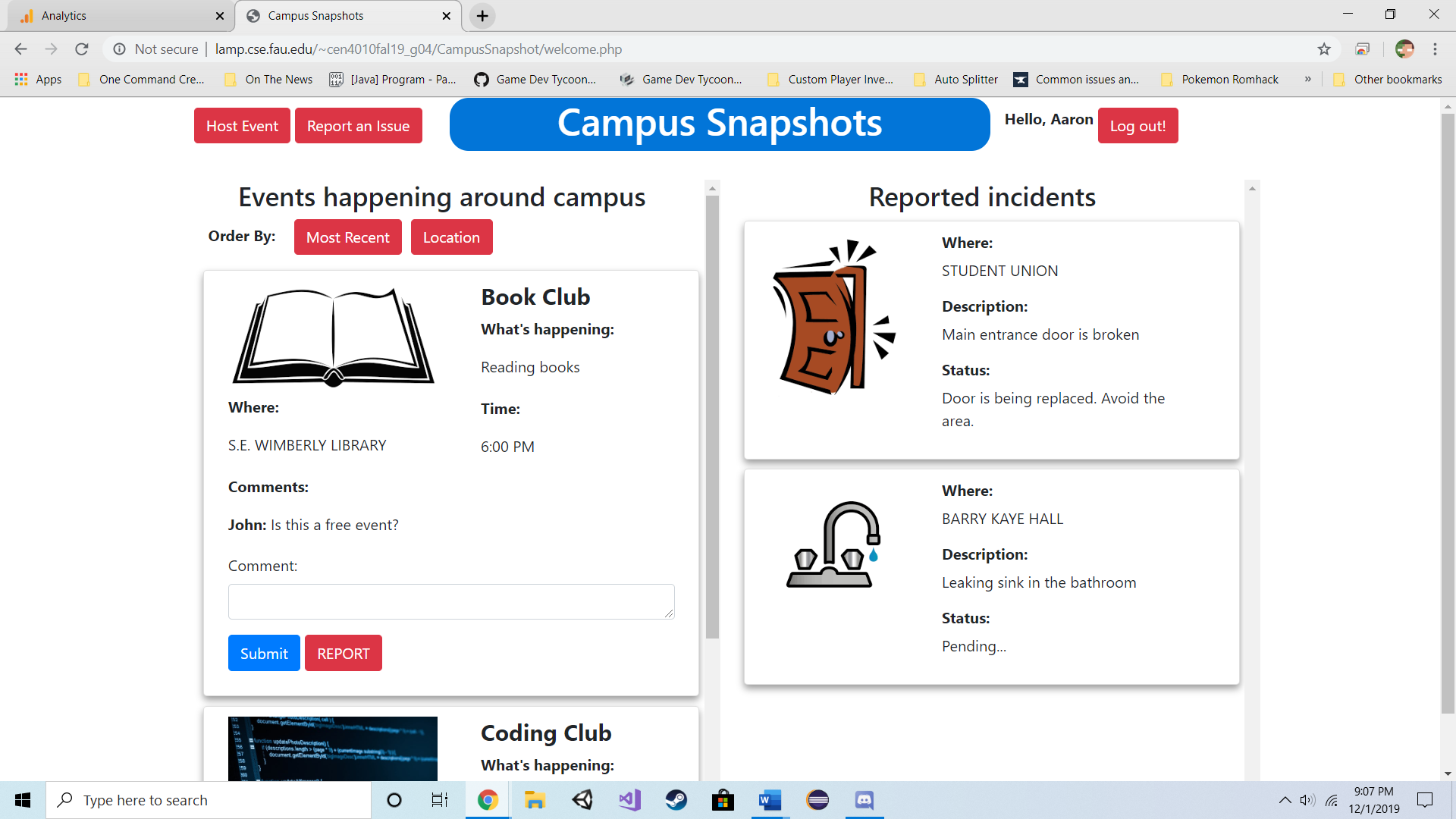
* **Basic Means of Communications - DONE**
* **Time Slot to Meet - DONE**
* **Front and Back End Leads Chosen - DONE**
* **GitHub Master Chosen - DONE**
* **Team Ready/Able to Use Chosen Frameworks - DONE**
* **Member’s Skills Defined and Known - DONE**
* **Team Lead Ensures Members Agree/Understand M1 - DONE**

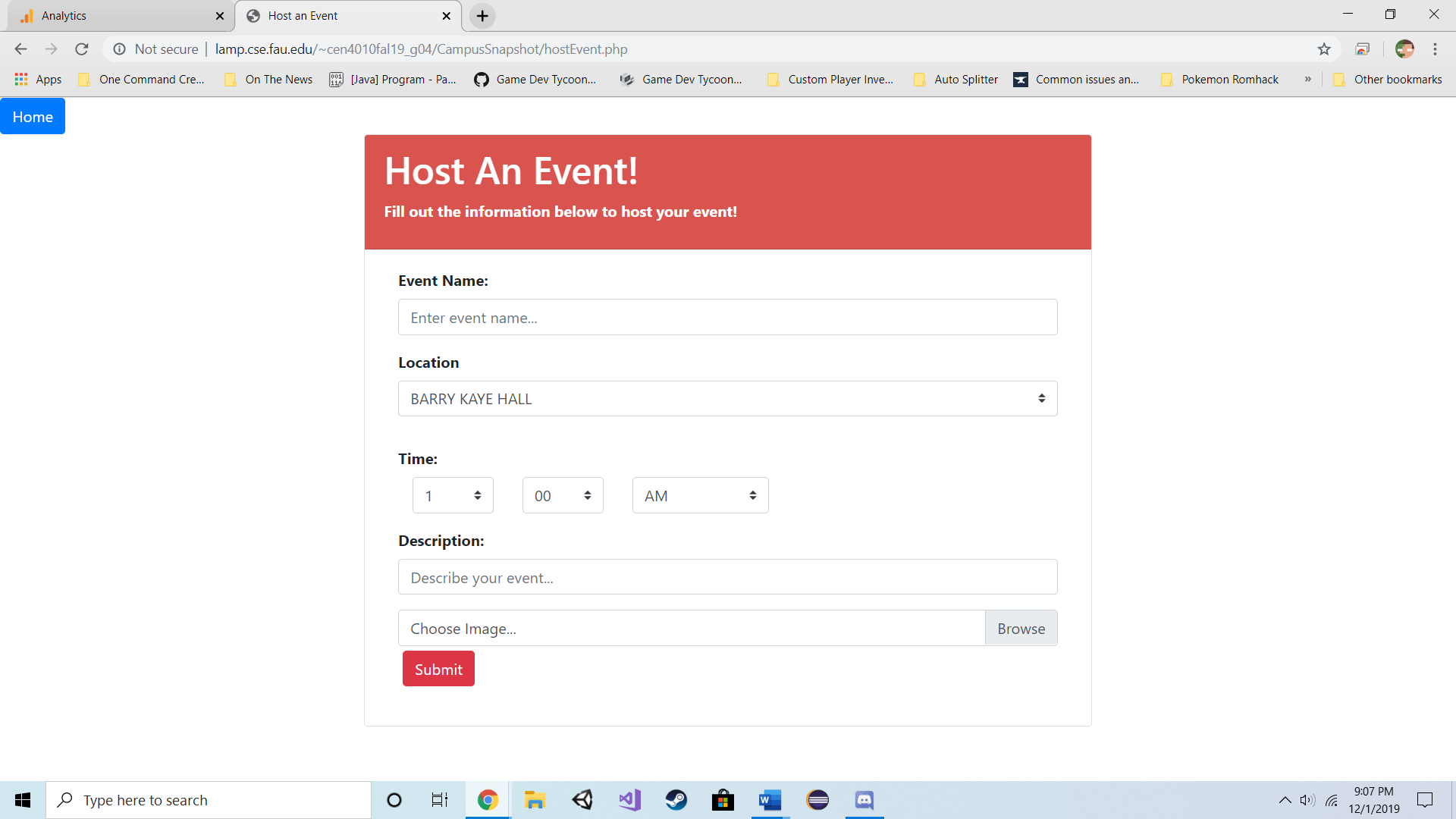
The instructor originally specified that structures should be listed under Architecture. More detail had to be added to the specific features of the document while providing the mitigation strategy to any risks that we might have had. Our original demo did not include the database being shown and our code review did not show any communication between the peer reviewer and the programmer. Finally, the input validation was not clearly understood and misrepresented.

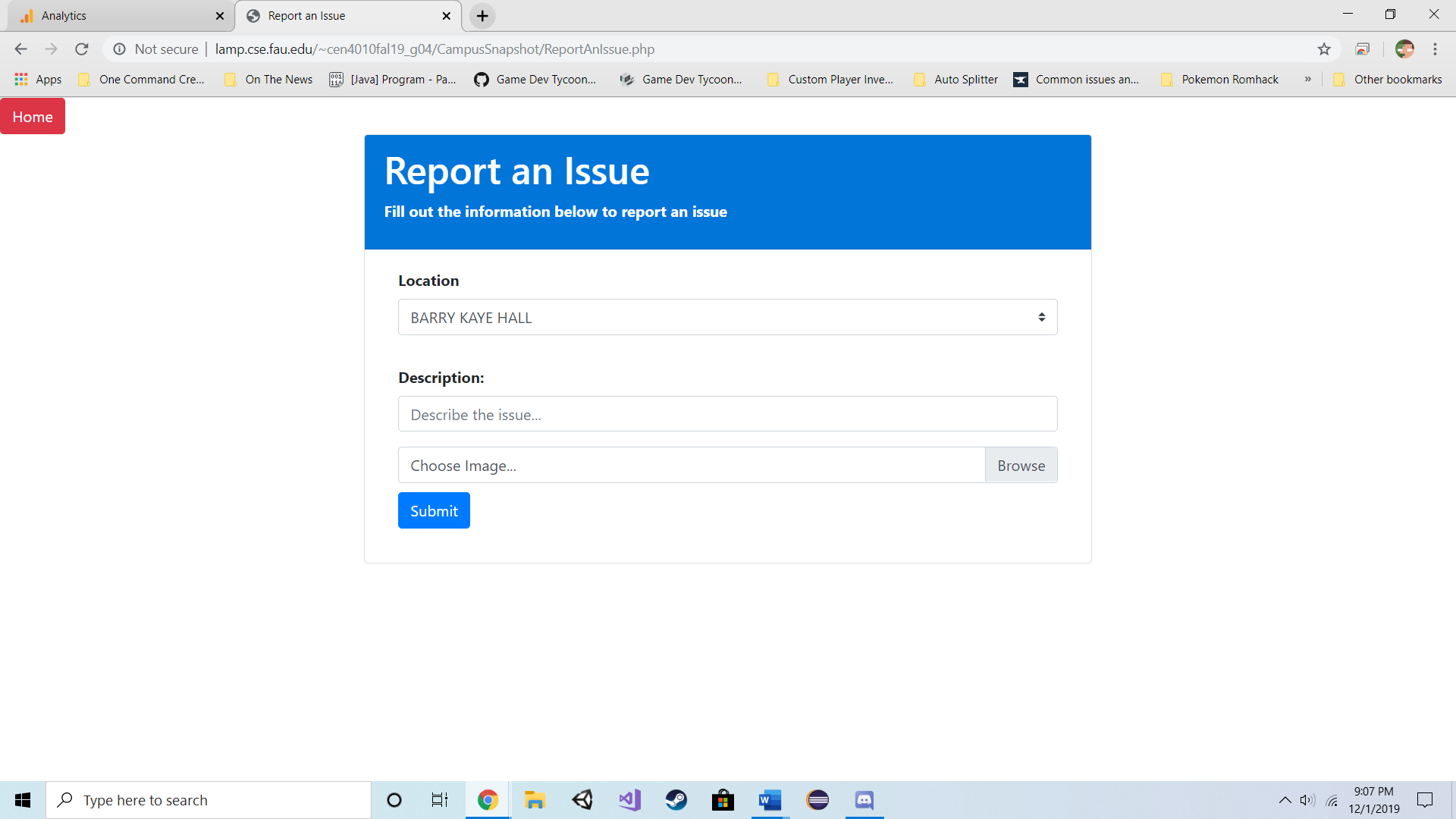
**Screenshots of Final Product**

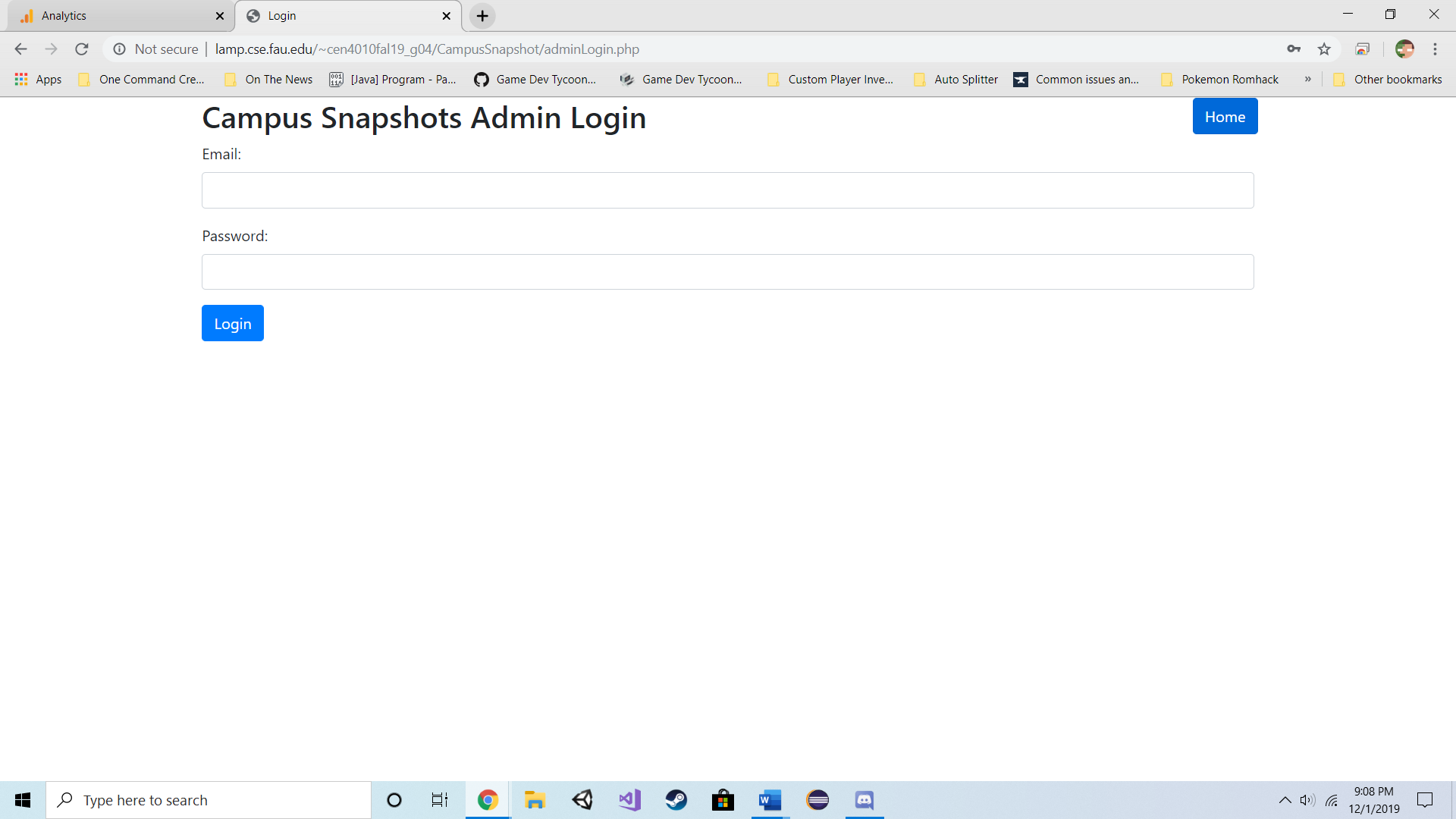


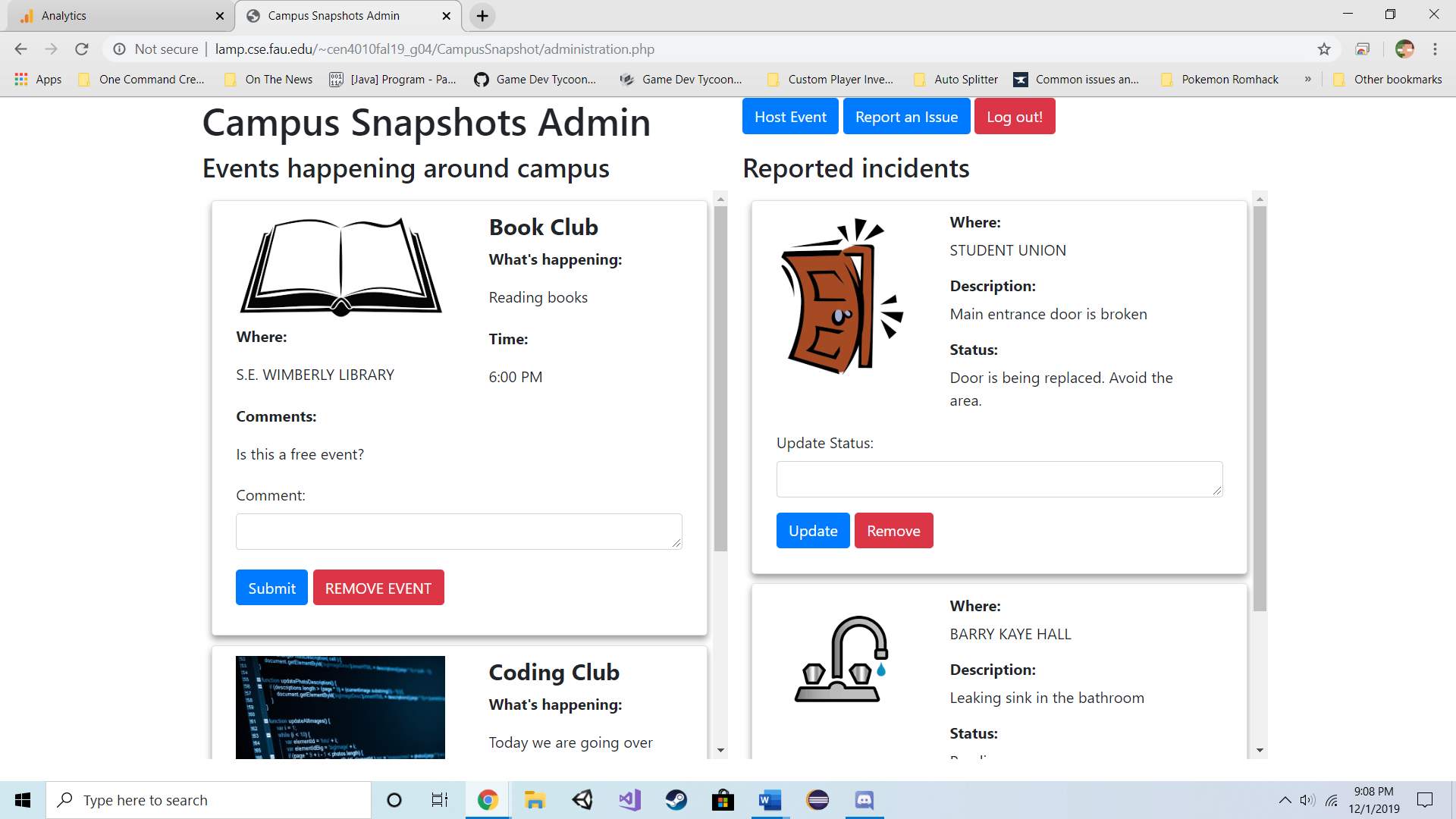




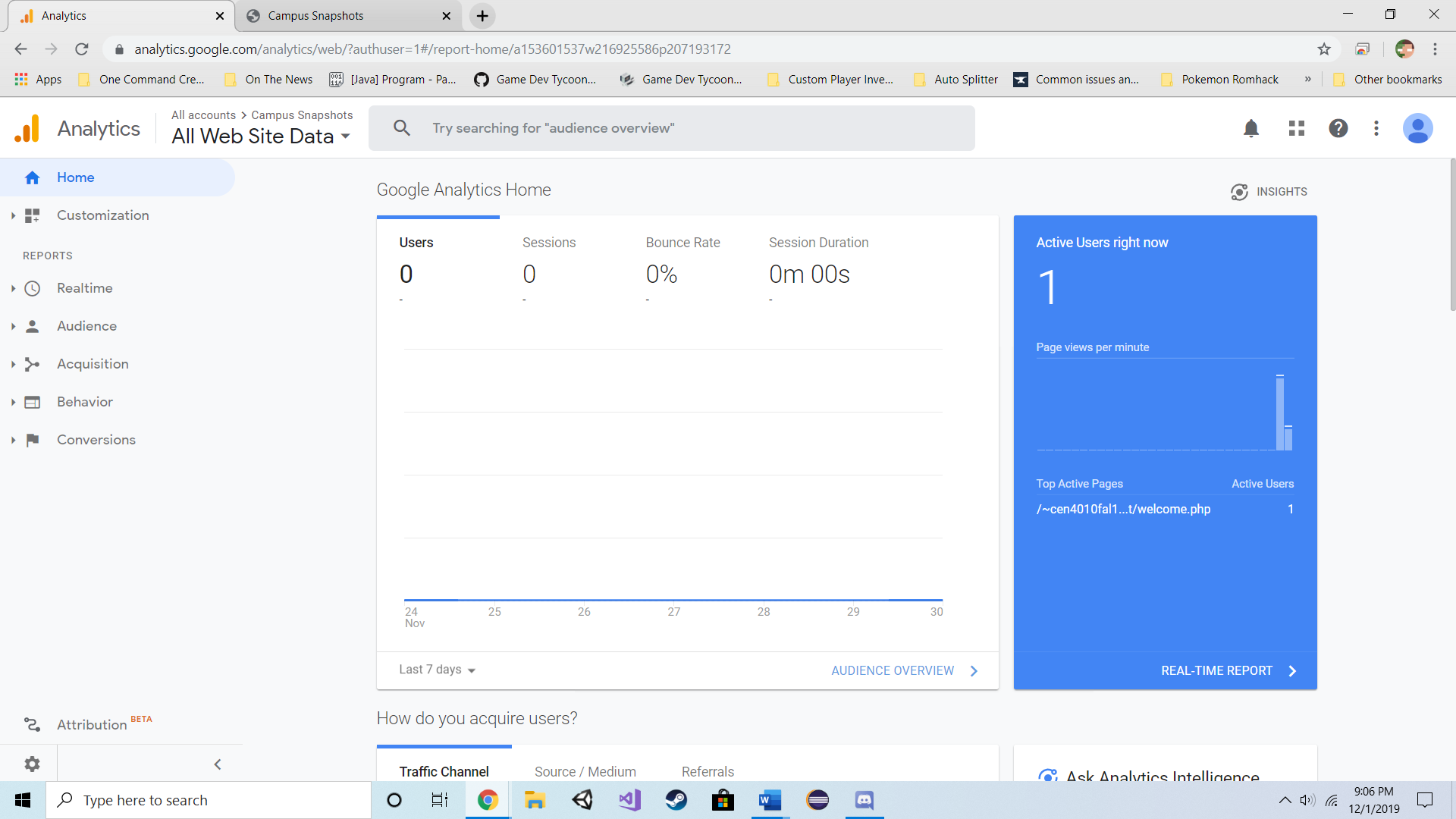


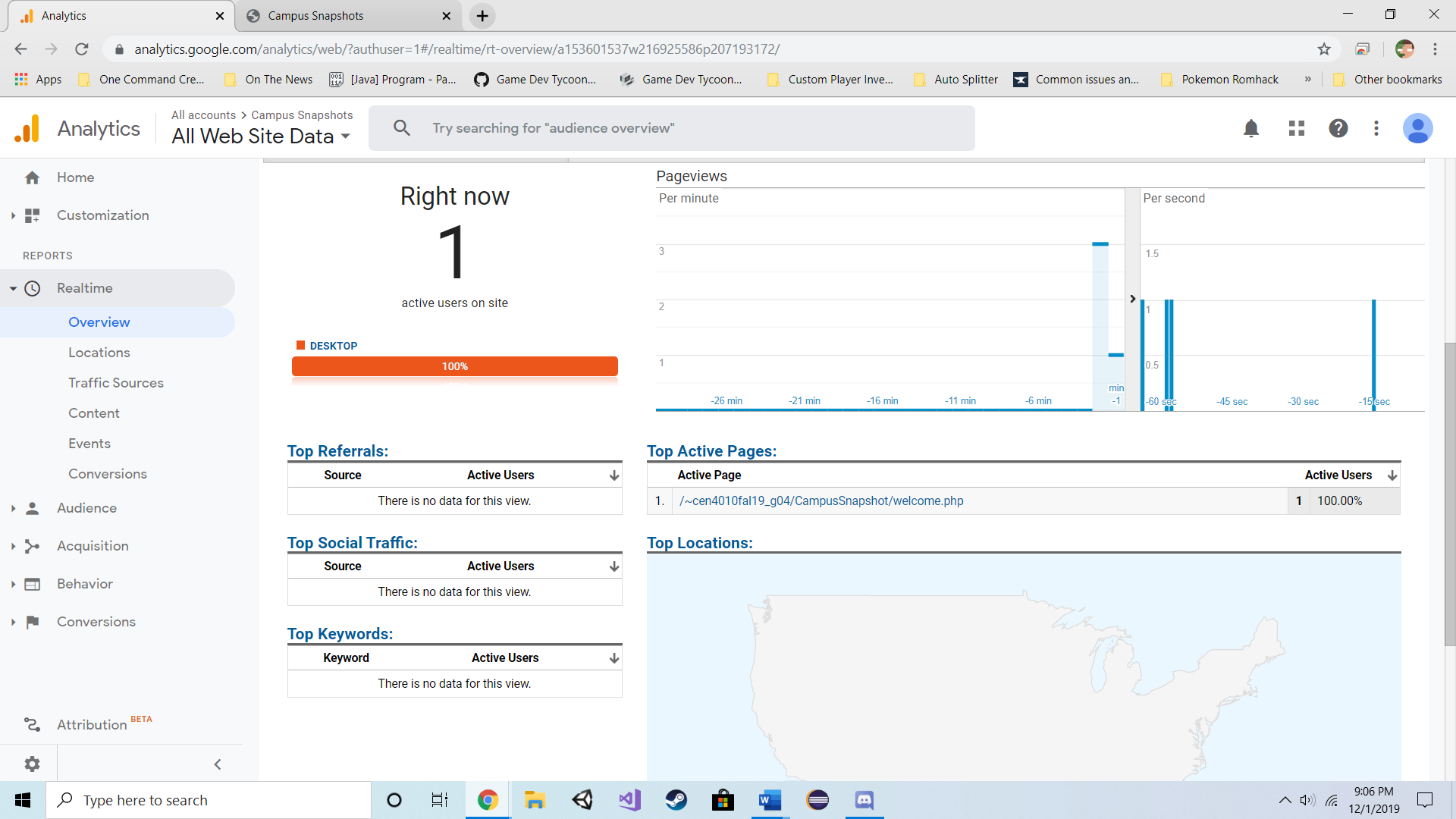






**Google Analytics Plot**





**Team Members Contribution**

* Aaron Haim
  + 25 Points
  + Updated Github and Scrum with project information
  + Wrote base deliverables
  + Reviewed code
  + 11 Github Submissions
* Daniel Fradlin
  + 25 Points
  + Designed the UI implemented in the pages
  + Programmed front-end of the website
  + Cleaned up any bugs in back end
  + Reviewed deliverables
  + 6 Github Submissions
* Yiyang Liu
  + 25 Points
  + Cleaned up databases
  + 0 Github Submissions
* Richard Urena
  + 25 Points
  + Designed the back-end system used on the site
  + Programmed back-end of the website
  + Specified scrum cards needed
  + 1 Github Submission

**Post-Project Analysis**

After finishing the project, there were a couple of things to reflect upon. As the team lead, I took a different role than I usually do. I was the one overseeing, managing, and presenting the product we made rather than actually programming it. Because of that I saw a new perspective to how things are done. I think there are quite a few issues with the product since we didn’t extensively test every possibility imaginable. However, the main issue is how we managed our time originally. At first, we were very good at procrastinating the project to the last minute. Near the end of one of the milestones, we literally turned in the assignment with 10 minutes to spare. Because of that, we finished the main page about a week early and fixed any bugs we encountered as it passed for this milestone. I would say we should have had a more definite schedule that we should’ve stuck to. It would have improved the quality of the assignments we were turning in instead of waiting for review.

The project had quite a few features that were cut due to infeasibility after the major redesign. Although we had a marker system for events and issues, there was originally supposed to be a map that displayed with the location of the markers. However, there was a slight issue due to Google Maps API being difficult to integrate with its potential costs and user agreements to a static image which really did not fit with the style of the site. So, for the last version, we scrapped the map idea altogether. If we had to redo this website, the markers would probably be navigation bars on the side with the map being the center of the screen. One of the other features that we thought about adding in but decided against was a sign-up screen for administrators. Since anyone would have access to administrator credentials if we did it that way, we decided to have admins preregistered in the database so as to prevent this issue.

By the end of this project, I think I learned that the perspective of a project is not limited to the technical side of the idea, but rather how each design fits one another that allows for maximum efficiency and usability or some balance between them. On the technical side, we learned a better understanding of JavaScript and PHP along with how UI interacts with a user. On the management side, we learned how to better schedule our product timing and divide task among ourselves. Finally, the teamwork perspective taught us the skills of each other and how we utilized them to create the final product presented for this class.