CEN 4010 Principles of Software Engineering Fall 2019

Milestone 5: Final Project Report and Demonstration

Project Title: Campus Times

Team Name: THE DREAM TECHS
Team/Group Number: 9

Group Members:

Samuel Parker(Scrum Master) - Sparke25@fau.edu

Abdullah Shafique(Front End Lead) - ashafique2014@fau.edu

Donny Stephens II(Back End Lead) - dsteph18@fau.edu

Joshua Vayntrub(Product Owner) - jvayntrub2015@fau.edu

Date: December 2, 2019

http://lamp.cse.fau.edu/~cen4010fal19_g09/project/

History Table:

Date	Checklist
10-28-2019	Database setup
10-30-2019	Updated GUI
11-1-2019	Web Pages Created
11-2-2019	Website Connected to Database
11-12-2019	Rewrite Data Definition

11-12-2019	Prioritized the Functional Requirements
11-12-2019	System architecture and UML diagram updated
11-12-2019	Revised Risks and Mitigation Strategies
11-12-2019	Vertical Prototype Video Demo Created
11-12-2019	Team Member Participation Form Revised
11-30-2019	Updated History Table

Product summary

Campus Times

1. **Create Account:** Users will create an account by entering their Email, Username, Password, and Confirmation Password. After this, the User will select Student, Staff or Teacher. After the account has been created successfully, the user will be directed to the Members page.

2. **Signing In to Existing Account:** Once an account has been created Users will be able to sign in whenever visiting the website. To sign in to Existing Account the User will enter Username and Password, the same Username and Password used to Create Account.

3. **Members Page/Status Update:** Once signed in the user will be directed to the member's page where other members post can be seen. In addition to that feature, Users will also have the option of posting a status update of their own. These updates can range in subject from events happening around campus to complaints about the libraries leaking toilet.

Unique Feature(s): One unique feature of our product will be the ability for users to submit a photo along with their status update. We wanted to add more expression to the member's page by giving the users the option to post pictures along with their

written post.

http://lamp.cse.fau.edu/~cen4010fal19_g09/project/

Summary: Executive

In this project, we will create a web system called Campus Times. By using this system, people (students, teachers & staff) on campus can report issues that need to be addressed as well as campus events. We are developing this project to help students and faculty stay in the know of issues, events, and news. Also, to ensure that the upkeep, the appearance, and maintenance of the campus remains properly maintained.

Our Competitive Analysis

Campus Times	**Competition (FAU)**
1. Report instantaneous issues/events	Only can report concerns NOT events instantly.
2. Real-time Snapshots of campus status to University administration (teachers, faulty)	If need immediate attention, call 9-1-1.
3. Motivate users to upload and report problems and events	Doesn't motivate users to upload and report problems and events
4. Ability to upload pictures along with status update.	Can't upload pictures at all.

The planned advantages for this project page are the instantaneous reports of issues

and events, University administration is notified immediately for issues, the motivation the page gives users to upload and report problems and events, along with the ability to upload pictures with statuses or issues.

Definition of Data

Data definition is comprised of two entities, User and Posts.

The User entity contains four attributes: Username, Password, Email, and Account_Type. All of these attributes will require input on registration of a new user. Afterwards, only username and password will be needed to log onto the website. The account type determines whether the user is marked as a student, faculty, or staff member.

The Posts entity is comprised of five main attributes and the sixth attribute is what connects the tables together. The five main attributes of this entity are ID, Text, IMGpath, Post_Type, and Date. ID is used to signify each users post as a unique post, so when it needs to be deleted, it is deleted via the ID. The Text field is for the user's input text. IMGpath denotes the image file name. Post_Type tells the user looking at the post whether the post is a status or issue post. And the Date is to show when the user made that post. In order to connect this table to the User table, a sixth attribute is part of the table: Username. When a user makes a post, the post will be directly related to the username.

Overview of Use-Cases and Potential Scenarios

The web system called Campus Times will allow students, teachers, staff, and all campus goers to report all news, events, technical breakdowns, sanitary issues, and any other issues that take place on campus instantaneously. A student can use this system to post a need for a tutor, or even offer tutoring services. A teacher can report a toilet overflowing, or an activity organizer can announce a charity concert. Campus statuses are also given to campus staff when they periodically check the system to

monitor campus activity. The system can also be used for security reasons. A student can post suspicious activity, and a campus officer can check the system to irregularities reported by other campus goers.

Sam is a student at FAU. He wishes to organize a fundraising concert to raise money to fund an initiative to bring awareness to social injustice and poverty in inner cities. The concert will consist of local talent that includes students, and any faculty and staff. The concert headers will be determined by having a talent show, and the finalist will be the entertainment for the concert. Sam needs to be able to spread the word for both the talent show and the concert to all campus goers.

Sam sends a message through Canvas to his fellow students in his calculus class on how he can best get the word out to all fellow campus goers of his upcoming talent show and concert. A student messages back to him that he can use a system called Campus Snapshots, which is a site that allows all campus goers to update, in real-time any news and reports of things happening on campus. Campus goers can post pictures associated with the news that they are sharing, that will be accessible to all other campus goers. Sam uses Campus Snapshots to upload content related to his upcoming events to notify campus goers on the details and demographics to get the word out.

A user or a group of users have a photo(s) to be posted on the system. These can be photos taken on a smartphone, or text typed on a post directly to the system. The user must create an account to the system.

The user logs into the system giving their username and password. Once logged in, the campus goer will be redirected to the member’s page that will list a feed of user upload submissions, and the page will also give them the option to type a description of the

event in a text box, along with any associated media. Uploads are listed under the description of the event, along with the poster's username.
Once an event is posted, the system sends an email to the poster confirming the post was successful, and a message is displayed to the poster that their post and event has been uploaded and is now visible to all users.

Possible issues with the system include the following:

The system is self-sustaining, and because 24-hour monitoring cannot be done by the moderator, uploaded content cannot be filtered in real-time. This poses a problem given the real-time nature of the system. Moderation will require other users to report content that is either offensive, abusive, or unsafe for any user of the system.

The system is user-driven, and being so makes it difficult to validate the trustworthiness post that are uploaded by users. Moderators will have to frequently check posts for user violations and compliance issues.

The system administrator may be logged in, flagging posted content that goes against the user agreement.

System state on completion:

The campus goer is signed into the system. The event and associated media have been posted and is currently visible to all users that are signed in. Posted information is visible to the system administrator, who will do nothing if the content of the posted event

is compliant to the user agreement.

System Behavior requirements

1. Create Account: First-time users will be able to create an account and choose their account type based on their status with the school. Different account types include student, faculty, and staff. Each account is associated with a unique username and password.

1.1.1 To create an account, users will be prompted to enter a username, password, email, and account type.

1.2.1 The username and email will be compared to those in the DB, and if there are no matches, the submitted info above will be saved to the database, and the user will be redirected to the member's page.

Priority: 1

2. Sign In with Current Account: Each existing user will be asked to sign into their accounts at the start of the session.

2.1.1 To sign in, user will be prompted to enter their username and password.
2.2.1 The username and password will be compared to those in the DB, and if found, the user will be given a welcome message, and redirected to the members page.
2.3.1 If not found in the DB, the user will be given a message saying the username and password was not found as well as saying if they don't have an account to sign up , and will be redirected to the sign-in page.

Priority: 1

3. Report Issue: Once signed in users will be able to report issues based on what they are experiencing. Issue types include (for example) suspicious activity and technical breakdowns.

3.1.1 Once signed in, the user will be able to view all updates submitted by previous users that are saved on the database.

3.2.1 On the same member page, a user will be able to upload a picture, and associated text.

3.3.2 Once uploaded, the user will be shown a message confirming submission and redirected to the member page where their submission will be displayed along with all other user submissions saved on the database.

Priority: 1

4. Post Status: All users can make status posts.

4.1.2 All users will be able to upload statuses. It is up to the discretion of the user what the status is about based on relevancy. So a student account would post a status about an event, whereas a staff member would post a status about an update to an issue that someone reported

Priority: 1

System Operational requirements

1. The application should run on two latest versions of all major web browsers: Mozilla, Safari, Chrome, IE, Firefox, etc.
2. Data shall be stored in the database on the server
3. Privacy of users shall be protected, all privacy policies will be appropriately communicated to the users.
4. The language used shall be English.
5. The site shall be very easy to use, intuitive (no prior training required).
6. The site shall be attractive and media-rich in appearance.

7. Security of the site shall require users to register and log in (students to decide what functions require registration and login).
8. Effort shall be made to make the site easily searchable by internet search.

System Architecture and DB organization

User will be able to register, log in, and log out. They will consist of a username, password, email, and account type. Posts will be able to be created or deleted (by their respective user) and will contain an ID, text, image path, post type, date, and username. The User and Post is connected together by the username.

The tables in the DB are User and Post. The User table is what will comprise of the users of the website, each with a username (primary and unique), password, email (unique), and account type. The password is hashed using a bcrypt hash function. The account type is what determines whether a user is a student, teacher, or staff. The username and email are unique in that they cannot be used more than once per account.

The Post table will hold the information each user posts. It will have an ID (primary and unique), text, image, post type, date, and username. The ID will be a unique marker to the post so that each post can be identified properly in the DB, especially incase of deletion of the post, the text will be the text body, the image will denote the posted image (if there is one), the post type denotes if it is an issue or status, the date will signify when the post was made, and the username is there to show who is the one who posted the issue/status by connecting it to the User table.

The two tables are connected to each other through the username of each account.

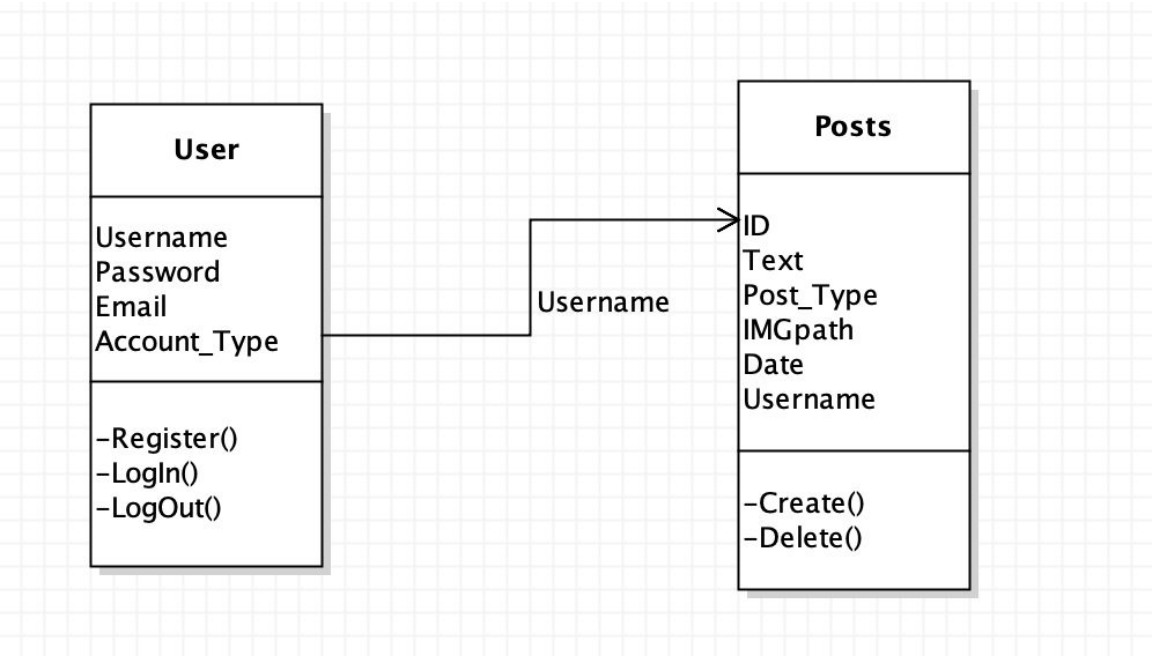
Images that are posted in the issues/status posts will be kept in the database

When creating a user, the code will search to see if the username that is being registered is already as well as the email. If either of those is in use, it won't allow the user to register until they pick a different username/email. If deleting a post, the ID for it needs to be searched for as the unique identifier of which post is being removed.

We will not be creating our own APIs for this project.

The newest posts by users will appear at the top of the screen, like a stack, based on the date. For passwords, passwords will be hashed using the bcrypt hash method.

UML Diagram



```php
if (isset($_POST['registerUser']))
{
    $email = mysqli_real_escape_string($db, $_POST['email']);
    $username = mysqli_real_escape_string($db, $_POST['username']);
    $userPassword = mysqli_real_escape_string($db, $_POST['password']);
    $userConfirmedPassword = mysqli_real_escape_string($db, $_POST['confirmedPassword']);
    $accountType = '';

    $accountTypeButtons = $_POST['accountType'];
    switch($accountTypeButtons)
    {
        case "Student":
        $accountType = "Student";
        break;
        case "Staff":
        $accountType = "Staff";
        break;
        case "Teacher":
        $accountType = "Teacher";
        break;
        default: die("Invalid type!");
    }

    if (empty($email)) { array_push($error, "Email is required."); }
    if (empty($username)) { array_push($error, "Username is required."); }
    if (empty($userPassword)) { array_push($error, "Password is required."); }
    if ($userPassword != $userConfirmedPassword){ array_push($error, "Please make sure that your passwords match."); }

    $user_check_query = "SELECT * FROM User WHERE Email='$email' OR Username='$username' LIMIT 1";
    $result = mysqli_query($db, $user_check_query);
    $user = mysqli_fetch_assoc($result);

    if ($user)
    {
        if ($user['email'] === $email)
        {
            array_push($error, "Email already exists.");
        }

        if ($user['username'] === $username)
        {
            array_push($error, "Username already exists.");
        }
    }

    if (count($error) == 0)
    {
        $password = password_hash($userPassword, PASSWORD_BCRYPT);
        $query = "INSERT INTO User (Username, Password,Email, Account_Type)
              VALUES('$username', '$password', '$email', '$accountType')";
        mysqli_query($db, $query);
        $successMessage = $username . " is now registered as a " . $accountType;
        array_push($info, $successMessage);

        $_SESSION['username'] = $username;
        $_SESSION['success'] = "You are now logged in";
        header('location: member.php');
    }
```

```php
if (isset($_POST['loginUser']))
{
    $username = mysqli_real_escape_string($db, $_POST['username']);
    $password = mysqli_real_escape_string($db, $_POST['password']);

    if (empty($username)) { array_push($error, "Username is required."); }
    if (empty($password)) { array_push($error, "Password is required"); }

    if (count($error) == 0)
    {
        $query = "SELECT * FROM User WHERE Username='$username'";
        $results = mysqli_query($db, $query);

        if (mysqli_num_rows($results) == 1)
        {
            $user = mysqli_fetch_assoc($results);
            if (password_verify($password,$user['Password']))
            {
                $_SESSION['username'] = $username;
                $_SESSION['success'] = "You are now logged in";
                header('location: member.php');
            }
            else
            {
                array_push($error, "Username/password is incorrect.");
            }
        }
        else
        {
            array_push($error, "Username/password is incorrect.");
        }
  }
}
```

```php
if (isset($_POST['upload']))
{
    echo "called";
    $image = $_FILES['image']['name'];
    $image_text = mysqli_real_escape_string($db, $_POST['image_text']);
    $currentuser = $_SESSION['username'];
    $target = "images/".basename($image);
    $postType = '';

    $postTypeButtons = $_POST['postType'];
    switch($postTypeButtons)
    {
        case "Status":
        $postType = "Status";
        break;
        case "Issue":
        $postType = "Issue";
        break;
        default: die("Invalid type!");
    }

    $sql = "INSERT INTO Posts (IMGpath, Text, Username, Post_Type) VALUES ('$image', '$image_text', '$currentuser' , '$postType')";
    mysqli_query($db, $sql);

    if (move_uploaded_file($_FILES['image']['tmp_name'], $target)) { $msg = "Image uploaded successfully"; }
    else { $msg = "Failed to upload image"; }
}
```

```php
if(isset($_POST['delete']) and is_numeric($_POST['delete']))
{
    $postToBeDeletedID = $_POST['delete'];
    $sql = "DELETE FROM Posts WHERE ID = '$postToBeDeletedID'";
    mysqli_query($db, $sql);
}
```

Project Risk Factors

Technical risks

- Risk ID - 1
- Date raised – 10/28/2019
- Risk description - There is a risk that the system will not support different types of image file types, or that the system will not support submissions of one type of file without the other (e.g. The system will not behave correctly is text is submitted without images, or images without text).
- Likelihood - Low
- Impact - High
- Severity - Medium
- Owner – Samuel, Josh, Donny, and Abdullah.
- Mitigating action – research and test PHP code to make sure the system will accept any image file type, and to make sure text and image submissions can be done together and independently.
- Contingent action – Update any PHP code that will hinder the submission of images of certain file types, and update any code that will not allow submission of text/messages and images to be submitted both independently and separately.
- Progress on actions – Update: on 10/28/19, mitigation actions were implemented.
- Status - Closed.

Skills risks

- Risk ID - 2
- Date raised – 10/28/2019
- Risk description - There is a risk that the functionality of the form uploads on the HTML and PHP pages will not work because of the group's lack of knowledge with the PHP code required to post and store videos. If this occurs, it will result in uploads not being visible to users and not being stored in the database, causing the pages not to function correctly.
- Likelihood - Low

- Impact - High
- Severity - Medium
- Owner – Samuel, Josh, Donny, and Abdullah.
- Mitigating action – The group agreed to remove the implementation of code used to upload any media other than pictures and text, and to avoid any functionality that group members are not proficient in.
- Contingent action – remove and avoid any PHP code in the project that the group attempted to use to add any media other than images and texts.
- Progress on actions – Update: on 10/28/19, mitigation actions were implemented.
- Status - Closed.

Schedule risks

- Risk ID - 3
- Date raised – 11/02/2019
- Risk description - There is a risk that there will not be a working Demo and prototype before the due date.
- Likelihood - Low
- Impact - High
- Severity - Medium
- Owner – Samuel and Abdullah.
- Mitigating action – Put more time and effort into the PHP code to make sure that the system can execute the required functions needed for the YouTube demonstrations.
- Contingent action – Write and rewrite PHP and HTML code necessary to make sure that the system will execute and behave according to the requirements.
- Progress on actions – Update: on 11/02/19, mitigation actions were implemented.
- Status - Closed.

Summary of milestone Feedback/Comments

1. Missing information:

- For our System Architecture of milestone three, we submitted this section in the form of text, absent the UML diagram, and with the diagram missing the connections. It was commented that the submission required a UML diagram be included with the diagram

showing the necessary connections.

- In our milestone three submission, there was no hyperlink to the demo of the vertical prototype. It was commented that this was missing from our submission and that a link would be needed to a demonstration of our Vertical prototype that has been uploaded on YouTube.
- The key risks of milestone three were resubmitted in the corrected format according to https://www.stakeholdermap.com/risk/risk-register.html, with the corrected templates in place. It was commented that although the templates were corrected, and the formatting was corrected, the submission lacked the required multitude of risk categories.
- When submitting our milestone four's committed functionality via the project summary, the main functionalities were listed along with the unique features. It was commented that the submission should have included the executive summary.

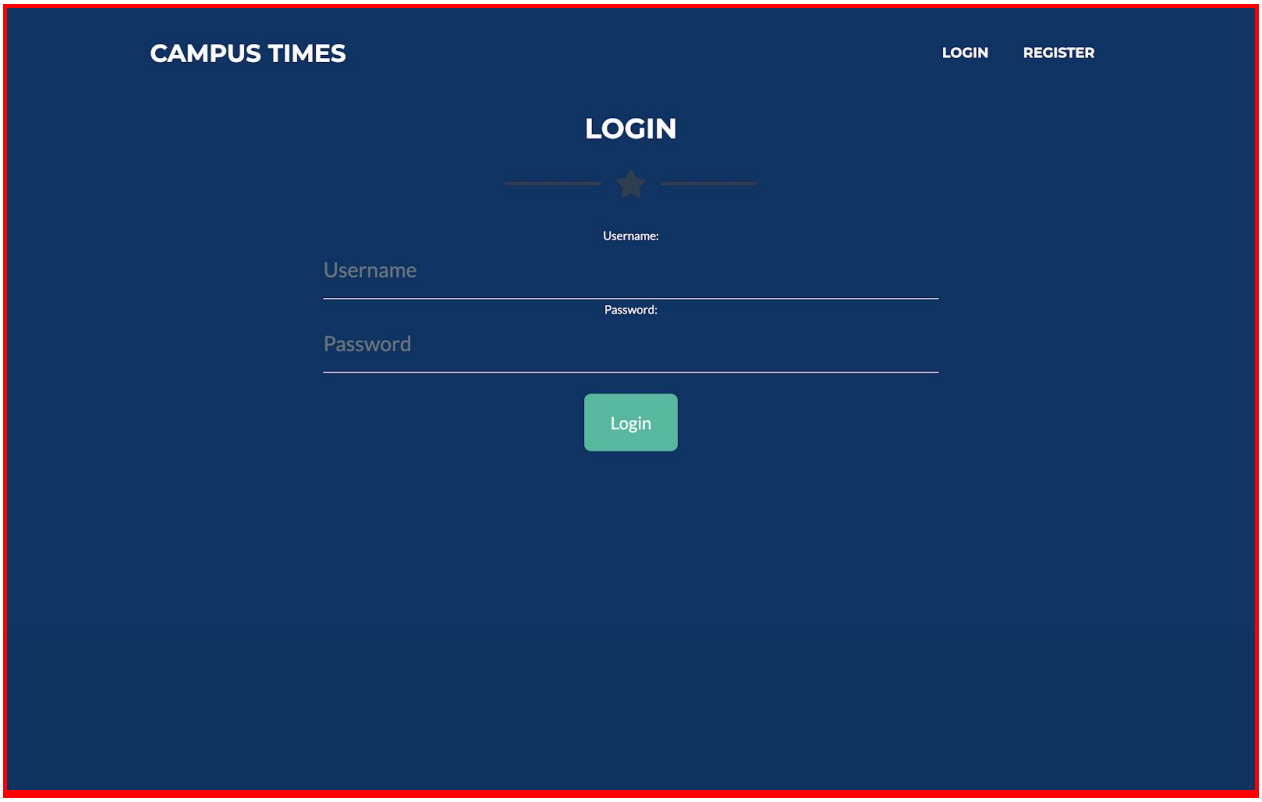
2. Misunderstanding of what to include and how to present:

- For the data definition portion of milestone three, we submitted the data definition in the form of a data structure. It was commented that the required submission was to be in the form of a description of the main terms.
- In the input validation portion of milestone four, under the self-check on best practices portion, the behavior of the way the system would respond to credential inputs. It was commented that the section should have instead included the user submission of file types, and the resulting behavior was from the submission.
- The cover page was submitted with a history of the changes made to the system. It was commented that the changes should be in the form of a table, instead of a sequence, to present a more professional aesthetic.

3. Unprofessional Formatting:

- The functional requirements of milestone three were originally submitted with requirements being listed but lacked any prioritization between the requirements. It was commented that the submission of the functional requirements required that they are prioritized, showing whether the requirement was a must-have, something desired, or something opportunistic.
- The key risk submissions of milestone three were submitted indicating the type of risk but did not give details in the form indicated according to https://www.stakeholdermap.com/risk/risk-register.html. It was commented that the correct template was not used and that the https://www.stakeholdermap.com/risk/risk-register.html should be referenced when identifying the key risks of the project.

Final Product Screenshots



CAMPUS TIMES

LOGOUT

MEMBERS AREA

Choose File No file chosen

Please post your update.

Status Issue

POST

Great Game LastNight Guys! - dsteph18 - Status

testing uploading an image - QAteststudent - Status

CAMPUS TIMES

LOGIN REGISTER

REGISTER

Email:

Email

Username:

Username

Password:

Password

Confirm Password:

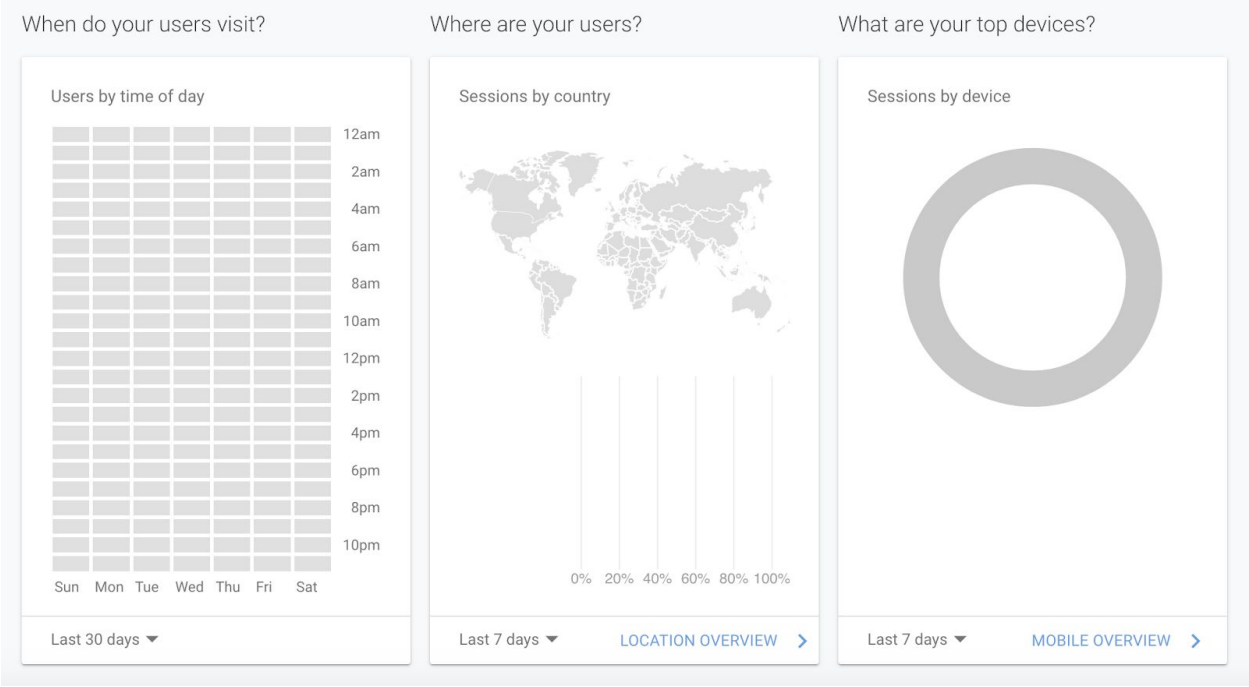
Confirm Password

Account Type:

Student Staff Teacher

Register

Google Analytics



Google Analytics gave us a script to enter into the top of all our pages so they can collect data that we can use to see our target audience. As more and more users use our websites we can see when our users are visiting the website where they are wen they do, and what devices they are using. We can also see what pages they are using the longest and see in real time how many users are on a specific page.

Team Members Contribution

1) **Team member participation form:**

Team Members: Samuel Parker = 25pts, Abdullah Shafique = 25pts,
Donny Stephens = 25pts, Joshua Vayntrub = 25pts.

2) **Team Contribution Description:**

Samuel Parker - Scrum Master

Developed and assisted in (front end) code for website.
Usability test plan developer.
Code Reviewer.

Post-project analysis.
GitHub contributor.
Milestone Documents.

Abdullah Shafique - Front-End Lead

Self-check on best practices for security developer.
YouTube video creator.
Google Analytics Plot.
Post-project analysis.
GitHub contributor.
Screenshots of final product.

Donny Stephens - Back-End Lead.

Title Page.
Competitive Analysis.
Assisted with Project Summary.
Team Members Contribution.
Self-check: Adherence to original Non-functional specs.
Post-project analysis.
GitHub contributor.

Joshua Vayntrub - Product Owner

Developed and assisted in (back-end) code for website.
GitHub Master.
Assisted with Project Summary.
Usability test plan developer.
QA test plan developer.
Self-check on best practices for security.
Post-project analysis.
Milestone Documents.

3) **Number of Submissions each team member made to GitHub:**

Samuel Parker - (2 Submissions), Abdullah Shafique - (3 Submissions), Donny Stephens - (2 Submissions), Joshua Vayntrub - (4 Submissions)

Post-Project Analysis

The main challenges we faced during this project involved our lack of coding knowledge. We had trouble with implementing a feature where the user(s) will be able to upload a video to the member's page. The main issue involved the group's inability to store and post videos due to our lack of knowledge on how to code it using PHP. From this challenge, we have learned that some of the wanted functions will not always be feasible due to time restraints and skill set. Another challenge during this project was communication. As a group, we had some communication issues at first because of our inexperience with working in groups online. Through this challenge, we learned just how important effective and frequent communication is during projects. We learned to lean on one another, ask questions, and be upfront and stern about individual roles and due dates. We also learned each other's strengths and weaknesses which enabled us to make the most out of our project.

Our initial outlook on the project was fairly different from the end product. As we started to get our communication skills in line and were able to organize the group cohesively among us, the plans for the project became clearer overall. We had some interesting ideas as to what we were planning to accomplish from this project, but as we mentioned, we realized certain roadblocks that caused us to reevaluate how we planned to work on the project and what we hoped to accomplish. This was a lesson learned in knowing our limitations. But it was also a way for us to have gained knowledge about our expertise and what we knew we were capable of. This allowed us to pivot our vision for the product and re-adjust our goals, allowing us to create the final product we have now.

Another one of the challenges we have experienced with this project is interpreting the requirements that were set before us. Given that the team members are entry-level developers, a lot of the requirements may have come off as vague, or hard to understand, which led to some of the documentation that was submitted to be incomplete, or incorrectly formatted. A lack of experience in mainstream developing required a lot of the milestones documents have to be resubmitted. We have also struggled with fully comprehending some of the comments issued for revision of our documents. While we've had these complications, it is our opinion that the repetition

through revising the milestone documents will strengthen our skill level for future projects.

Submission

https://youtu.be/JIPjbVLvH0E