San Francisco State University
CSC 317 – Introduction to Web Development Software
Term Project

Introduction

For the final project of the semester, every student will be completing a term project with 1 other student. ***This is a non-negotiable***. The purpose of this project is to bring together all of what you have learned from the previous assignments and incorporate it into a larger project to help everyone gain experience making a web application. For this semester, everyone will be implementing a photo management website that will allow users to post images, search for images and comment on posted images. Similar to the website https://imgur.com/ . We will not recreate the entire website but will focus on the 3 features listed above. However, if students are willing, you are welcome to go above the requirements and do extra features for extra credit. We will be using NodeJS and Express as our server-side technologies.

For most students, this will be one of the larger projects you will have worked on. It is important each student is committed to completing this project by the due date. You are also making a commitment to your team member. As each team progresses through the project, a high-level schedule will be given to help keep teams on track. It is up to the teams themselves to micromanage the allotted time between milestones.

Before any code can be written towards your team's term project, we will need to begin setting up our software stack. A software stack is simply a collection of software used to run a web application. Each team will have a similar software stack unless otherwise approved by the instructor. The software stack will be running on student's laptops as a localhost application. Teams can elect to use a remote server if they like or the VM from assignment one, but this is not a requirement. Our software stack will comprise of the following:

- Linux/Windows/MacOS operating system
    - Up to teams to decide, in some cases there will be two choices if members have different OSes on their laptop.
- NodeJS
- Express JS
- MySQL

The above is mostly referring to the back end of a web application. These are the pieces of software running on a remote server (in our case, our laptops). The other side, front end, will comprise of a mixture of HTML, CSS, and JavaScript. Use of libraries or frameworks like Bootstrap, jQuery, React and the like are allowed. These may not be covered in class if time does not permit. So, it will be on the students to do the research on how to use them. These front-end technologies don't need to be installed like the back-end software. In most cases, its built into most browsers. Then technologies like Bootstrap, React and jQuery will need to either be downloaded and included in your project or linked from a CDN (Content Delivery Network).

During the first milestone, your team will be required to install and configure your web stack, and have it display “Hello World” through your web application.

Requirements

Each team’s application must satisfy the following requirements. Teams will be graded on how many requirements are completed. The base requirements are non-negotiable, but more functionality can be added for extra credit.

- Functional Requirements:
    - Unregistered users must be able create a user account.
    - Registered users must be able to log in.
    - Registered users must be able to log out.
    - Registered users must be able post an image
    - Registered users must be able to search for image posts
    - Registered users must be able to comment on image posts.
    - Unregistered users must be able to search for image posts.
    - Unregistered user must be able to view image posts.

- None-Functional Requirements:
    - User account information must be stored in the application’s database
    - Image Post information must be stored in the application’s database
    - Messages that are made to post images must be stored in the application’s database.
    - Logged in users must stay logged in as they navigate your website until they log out.

- Additional Requirements:
    - Teams must fill out the README.md with the following information:
        - Each Team Members Name
        - Each Team Members SFSU Email
        - Link to Web Application. In most cases, this will be localhost:3000 but in the case your team changes the port, the instructor needs to know.
        - Instructions on how to run your team’s application. This includes installing all dependencies, what command is used to run the application (what is used to start the server-side code) and any additional information needed to run the application
    - Script used to make Database tables. This will be needed to test your application while grading. It may be something you can create later towards the end of the term project.

Milestones

Below is a set of milestones to help guide teams on planning out the implementation of the application. The dates attached to each milestone set an expectation of how far along a team should be when it comes to completing the application. It is important that teams stay on track. Teams are expected to have the items listed in each milestone completed (or very close to completed) by the data attached to each milestone. This may be checked randomly. Teams will have at least **48 Hours** of notice in advance before checks will occur. Repositories will be checked for completion of tasks. Please be sure that code used to complete the features are in the repository before they are checked. **Rechecks will NOT be performed.** If a feature is broken or incomplete this may be ok if enough effort went into its progress.

Milestone One – Sunday November 17th, 2019

For milestone one, each team will setup their web stack and initialize their application and repository to its initial state. We will call this our hello world application. It is required that each team get their application running and pushed to their GitHub repository by the end of milestone one. Please refer to the Express JS slides on ilearn for a sample file structure for your application. You do not need to follow this exactly, but your application's file structure needs to be well thought out. Points will be deducted for sloppy applications. To complete milestone one each team must do the following:

1. Use the link on ilearn to create your team's repository.
    a. ONLY ONE REPOSITORY PER TEAM. DUPLICATES WILL BE DELETED.
2. Follow the instructions in the README.md of the repository.
    a. This will include inviting your team member to the repository
    b. Changing the name of the repository to your given team number.
3. Install all the necessary software so each member can run the application on their laptop. These include node js and mysql. Express can be installed later once your application is created.
4. Create and push the initial structure of your team's application. You may let node make the initial structure for you by using the following commands:

    First create a new folder, the name of the folder should be chosen by your team.

    ```
    mkdir photo-app

    cd photo-app

    npx express-generator --no-view --git

    npm install
    ```

    The above commands will make a default express app with no view engine and a pre-filled git ignore file. Then to run your application you can use the following command:

    ```
    npm start
    ```

    It is required that your team can run your application via nodeJS and when navigating to your applications URL, it displays "Hello World".

Milestone Two – Sunday November 24th, 2019

For milestone two, each team will determine which forms from their earlier assignments will be used for the term project. It is also possible to just start from scratch. Once decided, each team will need to sketch out the look of the remaining pages of the web application. These additional pages include but are not limited to a landing/home page and a page for viewing a specific image post. Once these additional pages are sketched out (this is called wire framing), begin to convert these sketches into HTML and apply CSS. Please note that it is expected that your application has a theme (color wise). This means all the pages should contain a similar look and feel. It should not be the case that going from page to page the colors are very different. These created HTML files and the accompanying CSS files should be stored in the correct folders. The CSS files need to be correctly linked to the HTML file as well. Absolute paths to files **ARE NOT ALLOWED**.

The completed pages need to be pushed to the team's repository.

Milestone Three – Sunday December 1st, 2019

For milestone three, your team will begin to implement the remaining portions of your application. This will include creating routes for the functionality of your application. During this milestone, teams should focus on completing the log in / log out, registration and post image features of their application. This will give your application the following functionality:

- Allow users to enter information to register a new account on your website
- Allow users to log in with their credentials on your website
- Allow users to log out of your website.
- Allow users to post images to your website

The code used to implement the above features need to be pushed to the team's repository.

Note that in order to complete these features, your database and the database tables will need to be created as well.

Milestone Four – Saturday December 14th, 2019

For milestone four, your team will begin to implement the remaining portions of your application. This will include creating routes for the functionality of your application. During this milestone, teams should focus on completing the following remaining features: searching for image posts, viewing an image post and commenting on an image post. This will give your application the following functionality:

- Allow users to search for image posts on your website
- Allow users to view an image post on your website
- Allow users comment on an image post on your website.

The code used to implement the above features need to be pushed to the team's repository.

Please note the above date is the due date of the term project. Please ensure AS A TEAM that ALL REQUIREMENTS ARE MET BY THIS DATE. NO LATE SUBMISSIONS.

Submission

Teams will submit their application to their respective team GitHub repositories. Only the master branch will be graded so if teams use multiple branches, then it is the team's responsibility to ensure that the most up to date and working form of your application is on the master branch.

A project will be marked late if the latest commit's timestamp is past due date posted on ilearn.

Grading

Applications will be graded on the following criteria:

- Usage of GitHub
- Structure of application (how your application is organized)
- Code Quality and cleanliness.
- Completion of Requirements
- Visual Appearance of Application.
- Progress being made at each milestone.