San Francisco State University

Term Project

CSC 317

Anthony John Souza

Contents

Introduction

For the final project of the semester, every student will be completing a term project. 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, all students will be implementing a photo management website that will allow users to post images, search for images and view and comment on individual image posts. Like the website https://imgur.com/ . We will not recreate the entire website but will focus on the 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. A high-level schedule will be given to help keep students on track as they progress through the project. It is up to the students to micromanage the allotted time between milestones.

Before any code can be written towards your term project, we will need to setup our software stack. A software stack is simply a collection of software used to run an application. Each student 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. Students 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
- NodeJS
- Express JS
- MySQL – installation can be postponed until needed.

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 do not need to be installed like the back-end software. In most cases, they are 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, you will be required to install and configure your web stack, and have it display "Hello World" through your web application.

Requirements

Each student's application must satisfy the following requirements. Students 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 to post an image.
    - Registered users must be able to comment on posts.
    - Registered users must be able to search for image posts
    - Unregistered users must be able to search for image posts.
    - Unregistered user must be able to view individual image posts.

- Non-Functional Requirements:
    - User account information must be stored in the application's database
    - User Passwords must be encrypted.
    - Image Post information must be stored in the application's database
    - Logged in users must stay logged in as they navigate your website until they log out.
    - All input received from login and registration must be validation on both client and server side.

- Additional Requirements:
    - Students must fill out the README.md with the following information:
        - Student's Name
        - Student's SFSU Email
        - Link to Web Application. In most cases, this will be localhost:3000 but in the case, you change the port, the instructor needs to know.
        - Instructions on how to run your 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. This is very important, as projects that cannot be built will be penalized according to the non-build penalty outlined in the syllabus.
    - 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. An example process will be shown in a submission video.

Milestones

Below is a set of milestones to help guide students on planning out the implementation of the application. The dates attached to each milestone set an expectation of how far along you should be when it comes to completing the application. It is important that students stay on track. Students are expected to have the items listed in each milestone completed (or very close to completed) by the date attached to each milestone. This may be checked randomly. Students 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. Students are encouraged to commit/push code frequently into their respective repositories. **Failure to keep up with the expected features by the due dates will cause points to be deducted from your final term project score.**

Milestone One

For milestone one, each student 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 student 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 student must do the following:

1. Use the link on ilearn to create your term project repository.
2. Follow the instructions in the README.md of the repository.
3. Install all the necessary software so you can run the application on your laptop. These include node js and mysql (installing MySQL is optional for milestone 1).
4. Clone your term project repository to your computer and run the following commands:

   ```
   cd csc317-termproject-yourusername/application

   npm install
   ```

   Then to run your application you can use the following command:

   ```
   npm start
   ```

   It is required that you can run your application via nodeJS and when navigating to your applications URL, it displays "Hello World, I am {{Student name}}". Please replace {{student name}} with your name.

Milestone Two

For milestone two, each student will need to import their HTML pages from assignments 2, 3, and 4 (code to fetch fake photos can be left out, but can be used as reference later) into their Express JS applications. Each Student will also need to implement the remaining portions of their pages. This will include but is not limited to comment section for posts, navigation bars, banners, footers and so on. The Front-end of your application does not need to be completely done by this milestone but the layout and structure should be close to complete. Please make sure to put the files in the correct folders. You may notice there is no HTML folder in our application. For our term project we are going to be using a front-end framework called Handlebars. Handlebars is a templating language that allows us to easily write dynamic HTML. This is similar to frameworks like React, Vue, Angular, and so on but handlebars is simpler but it is also not as powerful.

Since we are using Handlebars, each student will need to convert their HTML pages into handlebars templates. This process will also require breaking your HTML pages down into small chunks called partials (like the idea of components in React). You can think of partials as small components in your application that are reusable for multiple pages of your application. Banners, nav bars and footers are good examples. Partials (and components in react) allow developers to write code once and reuse it multiple times on different pages of your application (See functions in any language).   See the videos on iLearn for some hints and help do this conversion.

Once imported, 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. The imported HTML, CSS, and JS files should be stored in the correct folders. The CSS and JS files need to be correctly linked to the HTML file as well. Absolute paths to files **ARE NOT ALLOWED**. See the express JS slides for a sample file structure.

The completed pages need to be pushed to your term project repository.

Milestone Three

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

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

The code used to implement the above features need to be pushed to the student’s repository.

Note that to complete these features, your database and the database tables will need to be created as well. Not having Sessions completed for this milestone is OK.

Milestone Four

For milestone four, you will begin to implement the remaining portions of your application. This will include creating routes for the functionality of your application. During this milestone, students 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 create a new image post on your application
- Allow users to search for image posts on your application
- Allow users to view an image post on your application
- Allow registered users to comment on an image post.

The code used to implement the above features need to be pushed to the student's term project repository.

Please note the above date is the due date of the term project. Please ensure that ALL REQUIREMENTS ARE MET BY THIS DATE.

Milestone Five

Given that we are doing remote instruction we are going to do something a little different for Term Project demos and their presentations. Each student will create a short video demoing their term project. A script is outlined below help guide you with what features should be shown when demoing your application.

You may use any form of video recording technology that is available to you. Use of smart phones are OK as well. A simple(ish) free opensource video recording software, https://obsproject.com/ , can be used to record videos. It is available on Windows, MacOS and Linux. This is the software used when making the videos for this course.

When making the video, make sure the highlight all the requirements completed. I would recommend have the database prepopulated with data to help show certain features. You can use the following script(outline) to help you demo the main features.

- Introduce yourself
- Introduce your project
- Create a new user
- Show the newly created user in the database
- Log the new user in.
- Create a new post
- Show the newly created post in the database
- Show the newly created post on the home page
- Search for the new created post.
- Click on the searched for post and view the individual post page
- Comment on the post.
- Show the new created comment in the database.
- Closing remarks. Anything you would like to say about your term project.
- Thank viewers for watching your video.

Please try to keep the demo's runtime between 2 to 10 minutes. Please be careful of your surroundings when recording.

Once the recording is completed go to ilearn and click the "Term Project Demo" glossary in the Term Project Section of ilearn. Create a new entry with the following name (on ilearn, it is the textbox marked as Concept) First-name Last-named Game-demo-name. For example, if I were demoing my project the name would be "Anthony Souza instructor's Photo App or iPA". You can upload your video to ilearn using the Ilearn Video button the definition box.

A video will be posted showing how this can be done as well as a demo of a sample application.

Submission

Students will submit their application to their respective term project GitHub repositories. Only the master branch will be graded so if Students use multiple branches, then it is the student'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.
- Completion of video demoing the project