Final Project

CEN 4031 Advanced Program Development

Angel Plata, Tinesha Erskine, and Justin Edwards

Fall 2021

**Instructions:**

**Submitting the Final Project**

Each team will submit a final project on the due date (See the dropbox for the due date). Teams will submit a word document containing a group section and an individual section. The submission should also include screenshots of the running application as well as a link to the Github repository.

**Group Section**

The group section contains a description of the project, what it does, how it is used, etc. It should also contain explanations on how the project demonstrates the team's understanding and (hopefully) mastery of the subject matter.

The group section should contain the following information:

* Team members and their roles/assignments. This should include the project and the group paper.
* A general description of the project.
* Initial requirements of the project. (What you planned to do). Identify any requirement changes that were made during design or implementation.
* Explain how the project demonstrates your understanding and mastery of the topic. For example, for the Data Access Layer, this section would describe the framework chosen, and what patterns were used. Explain how this contributed to the quality of the project. (In other words, you didn't just tack it on at the end. :^)
* Collection of status reports and other documents describing the design and implementation process. These should include major decisions and reasons behind them. (You don't need to include detailed bug fixes unless they are germain to major decisions.)
* User documentation. How to run the program, what files to use for input, etc. This should be enough that I can exercise your program.
* An evaluation of the final project. Include what you did right and what you did wrong. Describe what you would have liked to accomplish if you had more time, and what you would do differently if you had to do it again.

**Individual Section**

The individual sections should contain a section written by each individual student. The purpose of this section is to convince me that you understand the subject matter. This section should include an evaluation of the project from the student's perspective. (This may or may not agree with the group evaluation.)

This section should also include a discussion of what was learned in the course. This should not be a rehash of the lecture notes or the textbook. (i.e., not something you could have purchased or someone else could have written for you). It should contain what insights you gained from the course, especially pertaining to the group project or real-life experiences. It should not be a list of facts that you learned, but how it helped you with your project, your job, etc. How could what you learned help you to do the project better if you were to do it over again?

**Group Section**

Team members

Our team members were Angel Plata, Tinesha Erskine, and Justin Edwards. Angel took on the design role. Tinesha took on the presentation level logic role. Justin was our team leader and held the role of the domain layer developer. This included the logic to read and write the data layer.

General Description

For this project we created a stopwatch application that records the amount of time that passes between pressing the start and stop buttons. When the start button is clicked the website continually shows the current elapsed time. When the stop button is clicked, the start and stop times are saved to the database and the elapsed time is calculated and shown from most recent to oldest stored elapsed data.

Requirements

Our stack was:

Presentation Layer – HTML/CSS/JS (JQuery)

Application Layer – Python (Flask, SQLAlchemy)

Data Access Layer – PostgreSQL

We planned to make a stopwatch application based around a web server. This just interested each of us, and we each had skills to apply between Python for the server, PostgreSQL for the database, Javascript for the presentation layer logic, and html/css for the display. We initially planned to only show the start time and then when the stop button was pressed have it display the elapsed time. After we were able to get that to work, we decided to challenge ourselves and have the live elapsed time from the moment the button is clicked. This didn’t require any additional frameworks but it did require some research on the setInterval() method.

Our Understanding

PostgreSQL was chosen for the database and SQLalchemy was the framework to interact between the db and the domain layer. these weren't necessarily better or worse than any other database or framework, we were just already familiar with them and they were plenty enough to meet project requirements. This allows for quick and scalable storage.

Flask was the chosen server framework due to its ease of use, especially for a small project like this. Most of our time in this project was figuring out javascript logic. All that Flask and Python had to do was present the pages and pass data to/from the db.

We use JQuery several times throughout our javascript file for the presentation layer due to its simplicity. It is also what allows us to make the necessary POST and GET calls to exchange data with the server.

Status Reports

Text

Description automatically generated

Text

Description automatically generated

Design and Implementation Documentation

A picture containing text

Description automatically generated

A picture containing diagram

Description automatically generated

User documentation.

How to run the program:

1. Fill in database connection details within data\_connection.py. We used PostgreSQL.
2. Run 'python App.py'.
3. Follow url provided by Flask in CLI.
4. Press 'Start' to start timer.
5. Press 'Stop' to stop timer and update the history list.

Final Project Evaluation

An evaluation of the final project. Include what you did right and what you did wrong. Describe what you would have liked to accomplish if you had more time, and what you would do differently if you had to do it again.

I think our final product turned out great. I think implementing a lap feature and also displaying historic lap times for each entry would make a good challenge and addition to the project.

- Justin Edwards

One thing that I did wrong initially was converting the Unix Epoch time by multiplying it by 1000 to get it in milliseconds and then using the new Date(), .getHours(), .getMinutes(), and .getSeconds() functions. This caused the time to be a bit off. We ended up doing the classic mathematical conversion by dividing the Unix Epoch time to get the hour, minutes, and seconds. If we had more time, I think I would want to add sound to the timer. A sound for when the start and stop button is clicked and maybe a ticker sound for each second. I think this would provide a different type of feedback to the user so they will use another sense to know when the timer is starting, stopping, and currently going. I don’t think I would have done anything differently. I liked that it was a new stack for me. Also, I loved that I had teammates that were willing to challenge themselves, each other, and be creative with the timer.

-Tinesha Erskine

**Individual Section**