Naos Internship Website (Process)

Introduction: (Define what needed to be made)

During my journey as a Python Engineer, I was recently given the opportunity to intern for a company named “NAOS Blockchain Capital”. The company centers around the management of Crypto Currency positions and investments for applicants. This Crypto Currency management company is predominately based in Mexico, and allows individuals to participate in a portfolio of virtual assets, and in this way, gain exposure to the market around the technology in a simple way.

As the Software Engineer Intern, I was tasked with the goal of creating a new, refined “Know Your Customer” website. The main functionalities of the site needed to include the following:

- A working Login, Logout, and Register Form. (Needed to be translatable between English and Spanish via buttons.)
- Multiple Form pages (6 in total) that also allowed for image uploading, as well as conditional questions that only showed if other options were selected. (Needed to be translatable between English and Spanish via buttons.)
- A database that stores the User’s Form information.
- A way to Automate the creation of “Completion Forms” which includes all of the given User’s Form information.
- Relatively smooth access for Users to edit information if need be, or reopen their personal data-sheet (that connects to the main database) for further modification.

This is the way in which I approached this project, and just some of the techniques I chose to use with the given objectives.

Part 1: (The Front End)

- HTML
- CSS
- JavaScript
- Bootstrap

To begin, I chose to use Flask as the framework to build the site. Using a base HTML, I was able to create the basic pages by extending the same styles (CSS) and menu options for almost all pages. I then created the form pages, including the Login, Register, and questionnaire pages with HTML for arrangement, CSS for any other styling effects, and JavaScript for each page's functionalities. I also used a couple of unique JavaScript “Script Tags” for the functions that allowed the conditional questions to work.

Part 2: (The Back End)

- Flask (framework)
- Google Drive API
- Google Sheets API
- Flask SQLAlchemy
- Sqlite3 Database

This is where things got interesting. As mentioned previously, Flask was the framework that was chosen for this project. One of the main reasons I chose to use the Flask framework was because it is a light framework. It is fairly simple for others who may not be programmers or coders to understand and read in the future, if needing to make changes or edit the code.

Now, what about saving the data? I chose to save it in multiple ways: One way was saving and updating it to a text file (For record keeping and safely keeping track of information), and the next way was using a FlaskSQLAlchemy database configuration on the Server hosting the Flask Site. As I created my data objects for both methods, conveniently, I was able to add both methods of updating User information to the designated text file, and updating user information to the Sqlite3 FlaskSQLAlchemy database.

The FlaskSQLAlchemy database was primarily for authenticating Users by saving coordinating Usernames and Passwords into the provided sections within the tables.

The text file information acted as a viewable record keeper, for both Login information and User imputed Data.

But how would the user information get transferred from the form to the designated template format? (The idea is to create an automated process so no information needs to be manually entered.)

Using the “Gspread” module and “oauth2client” modules, I utilized the Google Drive API, as well as the Google Sheets API. The program worked as follows: First, the function would open up the template sheet designated for an individual or company, depending on which option was selected. If this was the User’s first time creating a KYC, the parameter’s of “First Name”, “Last Name”, and “Email” would dictate the title of the information sheet. Furthermore, if they had already completed a KYC multipage form, they could access, change, or edit the information by simply inputting the same information for the three designated parameters mentioned earlier. This will select the data-sheet with the unique title (created by the parameters) and allow them to change or update any information without creating a duplicate unnecessarily.

As for the image uploading, I used the enctype="multipart/form-data" within the HTML form code, then connected and initialized an upload folder within the rest of the files located within this project. The back-end for the image uploading was based on a non-conditional system, so that any image could be uploaded in any order.

Part 3 (Functionality)

The overall functionality of the site turned out to be a major success. Not only does it fully automate the creation of google data sheets for each User that fills out the multipage form, it also allows for Users to return to the site and update any information in the future if need be. The Simplicity of the Flask Framework also helps anyone who needs to edit the code or change the paths for any future updates. The site as a whole is easy to use, and can be understood by many.

- Functional Login
    - Saved Usernames and Hashed passwords.
- Data storage
    - Writes to file for record keeping.
    - Connects with Google drive API for updating.
    - Connects to Sheets API for writing data to edit data in specific positions.
    - Creates a copy of the designated template for each user.
- User sessions
    - Allows for Users to Login and Logout with ease.

Conclusion:

This project was very enjoyable to work on, as it pushed me to find different ways of solving a problem, and allowed for creativity in the process. Not only was I able to better

utilize Python while automating, but this project also allowed for the practice of different skills and languages such as HTML, CSS, and JavaScript. It gave me the opportunity to better my skills in using databases, and provided new options for data storage for the future.

I appreciate Naos Blockchain Capital giving me this opportunity to demonstrate myself. With this automation project completed, I hope to refine my Web Development skills as a whole and continue to better my techniques for creating and developing with Python.