Solution Analysis for GreenLab

Submitted by:

Whitcher, Zacharyah (Web analyst)

&

Landry, Christophe (Data analyst)

Preface

The following is a proposition regarding GreenLab’s needs for consultants in order to update, upgrade, and maintain their web service.

In this document, you will find a breakdown of the needs of GreenLab, the tools our consultant firm deem appropriate to use to fulfil those needs, some of the ways our team is planning deploy those tools, and the reason behind why we deem them to optimal for GreenLab.

Table of Content

Preface……………………………1

Table of Content…………………1

Technologies used………………2

CSV to Web Browser Graph...…3

Data Processes………………….3

Technologies Used

|  |  |
| --- | --- |
| Tool | Usage |
| Python | Simple coding language used for the base of some of the best data plotting tools. |
| Pandas | The primary tools for data manipulation with Python. |
| Seaborn | Used to plot the data created using Pandas. |
| Matplotlib | Used to style the plot using Seaborn. |
| JavaScript | Can be used for both the front and back end of the site. |
| ExpressJS | Used to build the backend of the web application, with good implementation of API, middleware, and request. |
| NodeJS | Allows for flexible and scalable applications and works well with data heavy infrastructure. |
| ReactJS | Great at building front ends for dynamic applications such as the client requests. |
| Bootstrap5 | Allows rapid Front End progression as it provides a plethora of design templates for forms, buttons, navigation, etc... |
| MongoDB | NoSQL allows for a flexible schema of the user data, which is ideal for web infrastructure. |
| AWS | Cloud platform with a large number of services, giving the client a one shop stops for their web deployment. |
| AWS Lambda | Only charges the client for each request the site receives. |
| AWS API Gateway | Allows the development of different path on the site as requested by the client. |
| AWS Redshift | Allows data requests to be processed faster than competitors, and ease of use and accessibility for new admins and end-users. |

CSV to Web Browser Graph

1. Upload the CSV file to an Amazon S3 bucket.
2. Write a Python script that will read the CSV file from the S3.
3. Perform required data transformations, writing the processed data to a file or database.
4. Using a Python graphing library, create a graph and save it as an image or HTML file.
5. Upload the Graph to an Amazon S3 bucket.
6. Create a web page using HTML and JavaScript that retrieves the graph file from the S3 bucket.

Dasta Processing Recommendation

Python > Excel

|  |  |  |
| --- | --- | --- |
|  | Python | Excel |
| Flexibility | As a vast array of free addons such as NumPy, pandas, and seaborn. | Limited to the tools in the software provided by Microsoft suite. |
| Automation | Can custom and default formulas and functions to manipulate the data automatically. | Able to create custom and default formulas to manipulate the data automatically. |
| Connectivity | Can be implemented into servers to respond to requests. | Cannot be implemented into servers to respond to requests. |
| Maintenance | Requires a specialised python data engineer if maintenance is needed. | Easier maintenance, dependent on Microsoft suite of updates. |
| Scalability | Can work on and manipulate large Data sets | Has limitations on the amount of data the software can work on. |
| Style | As large open-source libraries of graphs and plots types and styles. | Limited to the software library to customise the graphs and plots |