Final Deliverable

Group 20

Ashley Guthrie, Aaliyah Smith, Elijah Stall, John White

Table of Contents

1……………………………………………. Version Control

2…………………………………………….. Data Migration

3 ……………………………………Verification and Testing

4 ………………………………………..Maintenance Plan

5……………………………………….. Risk Assessment

7 …………………………………...Disaster Recovery Plan

8 ………………………………..Configuration Management

Version Control

For version control, we are using Git, a distributed version control system, and storing our code on Github to easily manage the any versions.

Data Migration

Since we collectively as a group, decided to no longer use PHP as the programming language for the project and switched to Python, we are using PythonAnywhere to run the code. PythonAnywhere is an “online [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) and [web hosting service](https://en.wikipedia.org/wiki/Web_hosting_service) that is based on [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) .[[](https://en.wikipedia.org/wiki/PythonAnywhere#cite_note-1)Every hour with the Tasks feature on PythonAnywhere, we will download the battery reader’s comma-separated values file from the provided endpoint, store the downloaded file in PythonAnywhere, and transform the file to only show the battery values that are below 5.0 and post the results to the designated endpoint.

Verification and Testing

Before the testing period starts, the finalized code will be run with the Tasks feature on PythonAnywhere. If we receive a 50, we will compare the transformed files to the actual result files and look for differences,fix the code, and re-upload to PythonAnywhere to start back testing.However, if we receive a 100, the files are being transformed correctly and changes to the code will not be needed.

Maintenance Plan

* Daily Maintenance
  + We will check to ensure files are properly uploading once daily.
* Weekly Maintenance
  + We will compare actual data directly from the battery readers to data in our file once a week to ensure the data is correct.
  + We will also compare files before and after processing to ensure ALL points below 5 are being included in the file
* Monthly Maintenance
  + We will work with a team to test the accuracy of the physical devices measuring battery usage at the end of every month

Risk assessment

**Assets -**

* The PythonAnywhere server our code is on - It would not take much time to set another server up if lost and would cost around an half an hour to set back up with backups of code stored elsewhere.
* The code itself - With the knowledge of how to set it up correctly, it would not take long to code everything from scratch again and would cost roughly an hour of time and whatever the electrical costs that hour of working would generate.
* The account that is associated with the server and code - It would not take long to set the server and code back up, roughly half an hour with backups of our code. The account being lost means we would need to pay for a new subscription to PythonAnywhere, which is $12 a month.
* The group member that wrote the python code - If this group member was lost it would be the highest priority to get them back, they are the only one who knows how to code in python and the one who managed to get everything working correctly. So long as there are no issues with the project until the end of the semester we would not need them to replace code, but if issues arise and we need to replace the code then we would be unable to get the code back up and running in time. This group member will also be necessary to complete documentation on the project as they are the one who managed to put together working code and know what issues they ran into. The amount of time and value lost if this group member is lost is tremendous and we will not be able to replace them.
* The group member with the PythonAnywhere account logins - This group member is also highly valued in our group as if we lose them then we lose any way to access the PythonAnywhere account that is associated with our server and code. The value of this member is identical to the PythonAnywhere account itself since we would lose the account.

**Risks -**

* Losing account login information - high risk, high value - Losing the account login information would mean we wouldn't be able to access the server and be able to make any necessary changes to the server itself or any code on the server which could end up being bad for us since any issues that were in the code that slipped by us would not be able to be addressed and could negatively impact our grade on the project.
* Group members being sick/unable to work on the project - high risk, high value - If a group member is sick or otherwise unable to work then we would possibly be missing a part of our project that we would need to have completed to get the best possible grade we can on our project.
* A problem occurs with the code that makes it work in a way that wasn’t intended/it doesn’t work at all - low risk, high value - since the code has already been shown to work and is consistently giving us 100’s even after the daylight savings change to time, it is unlikely that any new issues with the code will pop up. If a new issue pops up it would be damaging to our final grade on the project since the code would no longer work as intended.
* The program downloads a file that is malicious and changes the way the code works in some way - low risk, high value - there is next to no chance of this happening seeing as how our site should only be downloading files from the instructor’s site who would never do that to his students. If the site does download something malicious then our server or code could be compromised and would have to be made again quickly to have the project working for when it will be graded.
* Our server on PythonAnywhere encounters a problem and is unable to continue operating - low risk, low value - If for whatever our server encounters some sort of issue that keeps it from running all that would need to be done is to make another server and upload our code to it which should take less than an hour and will be fairly simple.

**Mitigation -**

* We will reduce the risk of losing our code by having backups on personal laptops and on a google drive that is shared between group members.
* We will reduce the risk of losing access to the server by sharing the login credentials among the group members and by using multifactor authentication along with a strong password.
* We will accept the risk of our original server encountering issues since it would be fairly simple to create a new server and upload our code back to it and it would be PythonAnywhere’s responsibility to keep our server running properly.
* We are going to transfer risks of group members being unable to complete their portion of the project by other group members in the project stepping up and completing their portion.
* We are going to avoid any downloads of malicious files by only downloading files from the instructor’s provided site.

Disaster recovery plan

* **If the PythonAnywhere account was compromised in some way**
  + Make sure any Credit/debit cards associated with the account have not been compromised, if they have been then they need to be terminated
  + Then contact PythonAnywhere and see if they can lock the unauthorized user out and see if they could somehow verify the account is our and hand control of the account back to us
  + If PythonAnywhere will not hand the account back to us then we will need to shut the account down and stop our server if possible
  + While the server is being shut down by the group member who the PythonAnywhere account is associated with, another group member should be creating a new account with stronger security in place
  + Once the new account is made we will need to decide on how the payment will be divided up.
  + Once the new account is made we will need to set the server back up and put the code into it to get everything running again

**- If some malicious code was downloaded or our code stops working as intended**

- We need to stop the server from running as it may download more malicious files/break further and inform the instructor as to what happened

- Once the server is stopped we will need to find what caused the server to start having issues, whether it was a downloaded file or if it was something in the code

- The issue in the code needs to be addressed and the problem fixed

- Testing of the updated code needs to be done to make sure the issue has been resolved

- Deploy the updated code and monitor it closely for the next few weeks to see if the issue persists

**- If there are general server issues/ the server becomes unresponsive**

- We need to find where the issue begins and contact PythonAnywhere to see what they are doing on their end to resolve the issue, and then contact the instructor afterward to inform them of the issues we are experiencing

- We need to prepare to start another server and deploy our code into that one if PythonAnywhere won’t be able fix the problem with their server

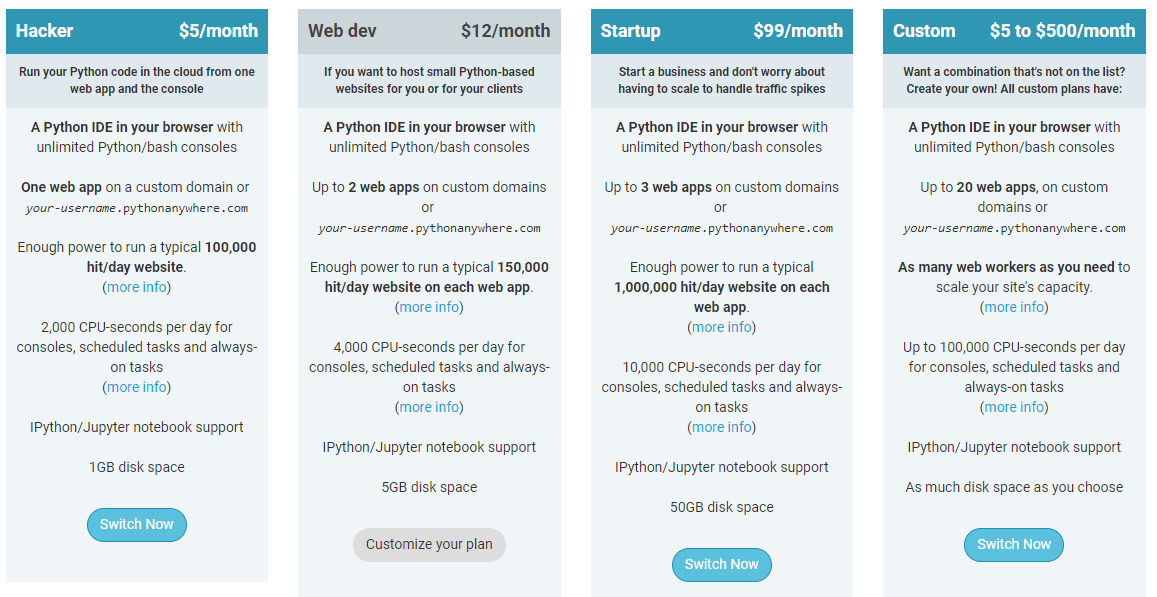
- If the server issue won’t be able to be resolved by PythonAnywhere then we need to deploy a new server and set it up with our code as soon as possible

- If the issues with PythonAnywhere continue or impacted us too greatly we need to consider switching to some other company to host our server to avoid future issues

Configuration Management

For our program we used PythonAnywhere, allowing us to execute the objective with code only. When purchasing a PythonAnywhere package you are given the option to customize the specifics from CPU time per day, Number of web apps, Number of web workers, Number of always-on task, Disk Space, and Postgres disk space.

Here we have listed the basics of our Python program, Web dev $12/month.



Quality Assurance Records

To ensure that the files would be downloaded and transformed correctly, Aaliyah started a test trial of the code by using the Task feature on PythonAnywhere on October 23 at 10:00 PM. Each hour past 10 minutes, the transformed files was saved in the File folder in PythonAnywhere to keep record of the results. Also, to make sure that the files was being POSTED to the designated endpoint the post result was outputted each time the code was executed. Below is a screenshot of the post result being logged on PythonAnywhere.

