MATES Computer Science

Senior Capstone Project Bi-Weekly Progress Report

Project Title	Bit by Bit
Team Members	Anna Pitera
Dates Covered by Report	02/24/24 to 03/07/24
Link to Github	https://github.com/anna-pitera/Bit-by-Bit

1. Summary of Project

Overview:

Bit by Bit is a gamified to-do list and habit tracker website that encourages users to complete their daily tasks and maintain good habits by rewarding them with in-game currency, which can be used to purchase customization options and in-game benefits. The website is created with HTML, JavaScript, CSS, and Phaser, a JavaScript library for creating web games. The Django web framework is used along with PythonAnywhere, which allows programmers to host web apps online for free with restrictions that can be removed by purchasing a subscription. To use Bit by Bit, players must create an account with a username and password which stores their progress in a MySQL database, allowing them to close and reopen their progress at any point from any computer. Bit by Bit "resets" every day at midnight so users can check off their habits each day, and users are rewarded for "checking in" each day.

Website Design & Layout:

The design of the website is in a simple, easily reproducible pixel art style, and the main page of the website consists of five key features: the user's character profile box, a to-do list layout, a store where users can spend their in-game currency on customization options and character accessories, a habit tracker display, and a few different number displays. The user's character profile box displays the user's customizable avatar/character and is made using Phaser sprite functionality. The to-do list is a list where users can enter and complete their daily tasks, and task completion rewards the user with currency. The store is a tiled display showing all obtainable items that can be purchased with currency, including character accessories and UI color themes. The habit tracker displays the user's entered habits that they choose to maintain, along with a check box where users can check off whether or not they maintained their daily habits each day. The user's habit "streak,"

(the number of days they have maintained their desired habits in a row), currency amount, and total number of tasks on their to-do list are all shown in various number displays. Simple quality-of-life features are also included, such as the ability to undo the accidental completion of a task.

2. Summary of Progress this Period

The beginning of this progress period was spent going through several Django and PythonAnywhere tutorials to build the basis for backend development. I won't go into too much detail here (see "Difficulties Encountered" section for more info), but I had to repeatedly restart the guides and tutorials to figure out what was going wrong in my specific use cases. I spent about four days doing this, and while that may sound like a lot of time, it was definitely necessary; without having an up-and-running PythonAnywhere site (even if there's barely anything on it right now), I would have nowhere to test features and database mechanics. Put simply, everything is going to be going on PythonAnywhere by the end of the project, so I needed to figure out the basics now in order to test future stuff. After I got the website up and running with the proper static files and no errors, I continued working on the Django Codecademy course, which conveniently works in PythonAnywhere hosting towards the end of the course. That course also shows you how to use databases in Django projects, which is perfect for what I need. Overall, I'm proud of how much I've learned about Django, PythonAnywhere, and even MySQL during this progress period.

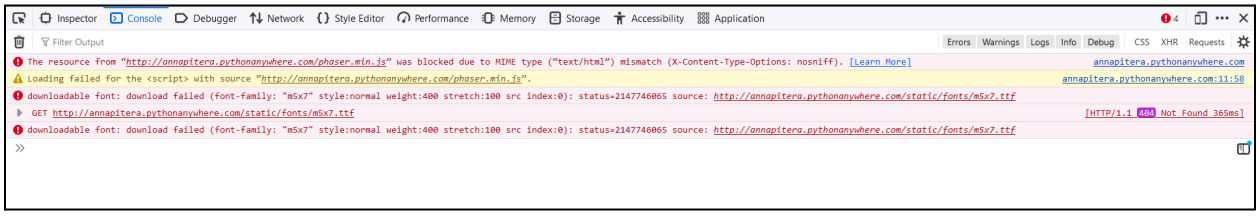
3. Detailed Progress this Period, separated by Team Member

In the previous progress report, I talked about how I adjusted the file structure to cooperate with Django's project structure, and I also talked about how I did that by using Django commands in the VS Code console manually. However, I learned during this progress period that it is much easier to just let PythonAnywhere do some of that work for you. Not only is it easier, it also prevents a bunch of issues with file paths differing between VS Code and PythonAnywhere.

So I followed this tutorial: https://help.pythonanywhere.com/pages/FollowingTheDjangoTutorial (along with the official Django tutorial)

And I ran into a few consistent issues. The main one that I struggled with was connecting static (CSS, JavaScript) files with the HTML code on my PythonAnywhere site. I wish I had taken screenshots while I was encountering these issues, but you're just going to have to trust me that there were a lot of them. There were a few different error messages popping up in the website's console, and sometimes the website wouldn't even start if I really messed something up.

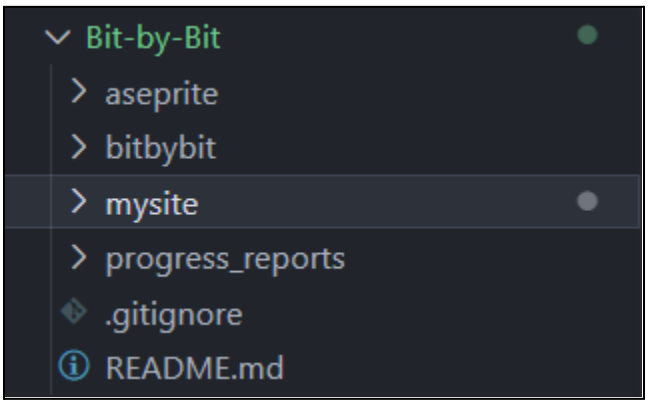
In the end, following the tutorial linked above very closely as well as using Stack Overflow really helped me. This is what the website's console looks like after I fixed the major errors (i.e. the static files errors):



And then this is what the console looks like after figuring out how to fix those last few errors:

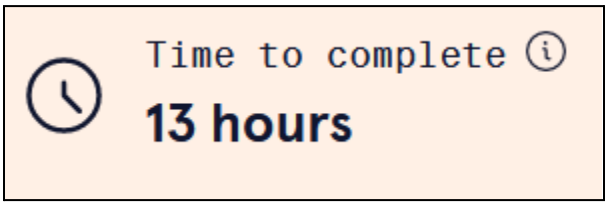
No errors! Perfect! Clean! Beautiful! Stunning! I managed to fix those last few errors after looking on Stack Overflow and finding a post from someone with a pretty similar situation to me. I added that Stack Overflow post link to my Trello resources list as well.

After getting that set up and working, I zipped my PythonAnywhere site folder and moved it into my VS Code workspace as well as my Github repository. So right now, this is what the repository looks like:



“aseprite” is just a temporary folder to store my Aseprite drawing files for the UI elements, “bitbybit” is the old folder from when I tried setting up Django in just VS Code that I’m keeping just in case, and “mysite” is the new PythonAnywhere folder that I’m just going to reupload into VS Code every time I make any changes to the Django files on PythonAnywhere.

The last few days of this progress period were spent continuing to follow the Codecademy Django/PythonAnywhere course. I started the progress period with about 10% of the 13-hour course completed, and now I’m at 35% completed.



In case you were curious, here's the link to the PythonAnywhere site:
http://annapitera.pythonanywhere.com/

The site is up 24/7 and updates whenever I manually reload it as of right now. It doesn't have a custom domain or a frontend yet though.

4. **Difficulties Encountered this Progress Period**
   - Django & PythonAnywhere
     - A lot of this progress period was just trial and error. While there are some great resources online (including the surprisingly well-made official PythonAnywhere guides), there were a few issues that I couldn't wrap my head around and had to figure out on my own.
     - For example, after repeatedly trying to get the static files to work on the website, I kept running into the same errors. I don't have screenshots of them, but one of the main ones kept occurring when I ran the "collectstatic" command to finalize the static files. It's hard to explain, but it kept saying that I had zero changes to collect when in reality there were many new changes to collect. So I looked into it and ended up combining advice from two different Stack Overflow posts. This is what the "settings.py" Django file looked like before that:

```
STATIC_URL = '/static/'
STATIC_ROOT = '/home/annapitera/mysite/static'
```

     - And this is what it looks like with the the new stuff:

```
STATIC_URL = '/static/'
STATIC_ROOT = '/home/annapitera/mysite/static'
STATICFILES_DIRS = (
    '/home/annapitera/mysite/static/node_modules',
    '/home/annapitera/mysite/static/fonts/',
)
```

     - It's the simple, one-line-of-code stuff like that that's really difficult to figure out. But I did it eventually!
   - Github
     - Initially, I wanted to link the PythonAnywhere folders into the Github repository automatically, but I learned that it's actually much easier to just download the folders and reupload them every so often. It also leaves a lot more room for mistakes because, if I mess something up on VS Code, I still have the PythonAnywhere copy easily ready to download.

5. **Updated Trello Board and Discussion**

Link: https://trello.com/b/Lzd1bk2c/bit-by-bit

Changes:
- Added more links to "Resources" list
- Added goals for this upcoming progress period

6. **Tasks to Be Worked on in Next Progress Period**

Creating proof of concept test form with database interaction (most of this is the same/continued from last report's goals, because I honestly spent a lot of time figuring out the Django stuff)

- Continue Codecademy course
- Watch/read tutorials and guides (already have some saved in Trello board and bookmarked in browser)
- Practice by making a test form where user can enter tasks and they get saved in the database
- Update Pythonanywhere website with new site
- Possibly work on username and password login system with MySQL if time permits? (just a very basic login system)

7. **Additional Information**

I didn't get as much done as I wanted this week, mostly due to the difficulties I faced with Django, but also partly due to the fact that I haven't had any time to work on capstone at home these past two weeks because I've been working at my new second job and also working on my research project. I'll definitely have more time once research is done though.