MATES Computer Science

Senior Capstone Project Bi-Weekly Progress Report

Project Title	Bit by Bit
Team Members	Anna Pitera
Dates Covered by Report	03/22/24 to 04/18/24
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 JavaScript Canvas 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**

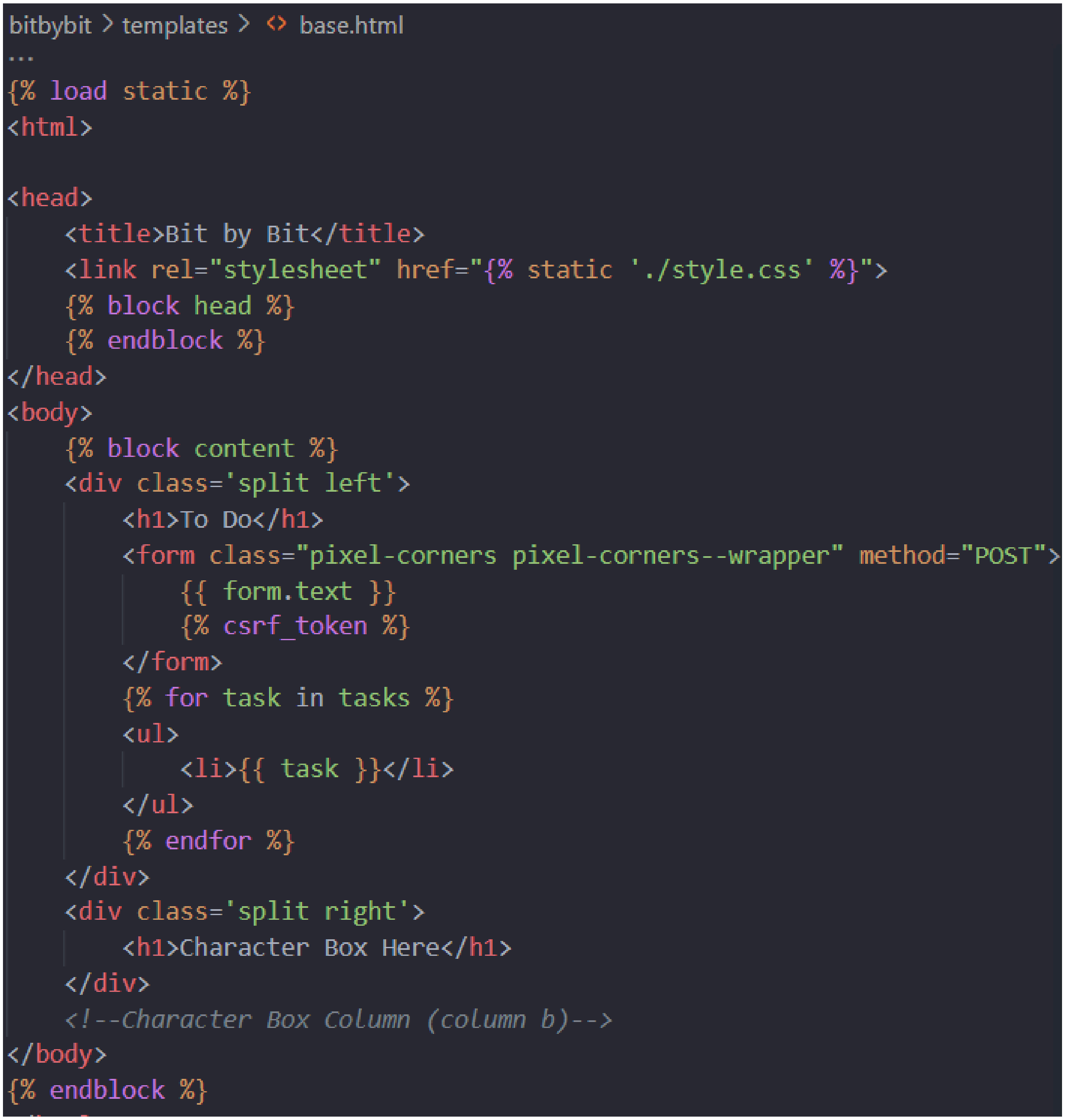
Most of this progress period was spent continuing work on the Django backend. This included working on implementing basic database functionality (CRUD), incorporating text input forms for users to enter task information, and setting up the admin backend controls for the database. Some time was also spent on creating a little bit more of a frontend. This meant making sure the CSS pixelated border styles and fonts were working properly (particularly with the Django input forms as they are a little unique) as well as setting up the split-screen UI (task list and controls on one side, and character box and customization on the other). I also made the decision not to use Phaser anymore because it doesn't really seem to fit the scope of the project; I will be using JavaScript Canvas instead for the game elements of the site. While there wasn't much tangible/visible progress made, I feel like I'm gaining a more practical understanding of Django the more I work with it.

3. **Detailed Progress this Period, separated by Team Member**

As shown in the midterm presentation demo, I got part of the task creation for the database working. I would show some screenshots here, but unfortunately I broke some things while attempting to implement more database functionality, so I kind of went a little bit backwards. Part of the issue that occurred is that one of the main guides/tutorials I was following wasn't really the best or most flexible way to do what I needed it to do. For example, after I finished that guide and had the task creation and viewing working, it all kind of fell apart when I tried adding the task due date part to the input form. It's hard to explain and a little frustrating, but I'm working on fixing it as soon as possible.

Even though I don't really have any pictures of the kind-of-working site, I do have some of that code saved just in case I need to go back to it at any point.

Here's what base.html looked like:

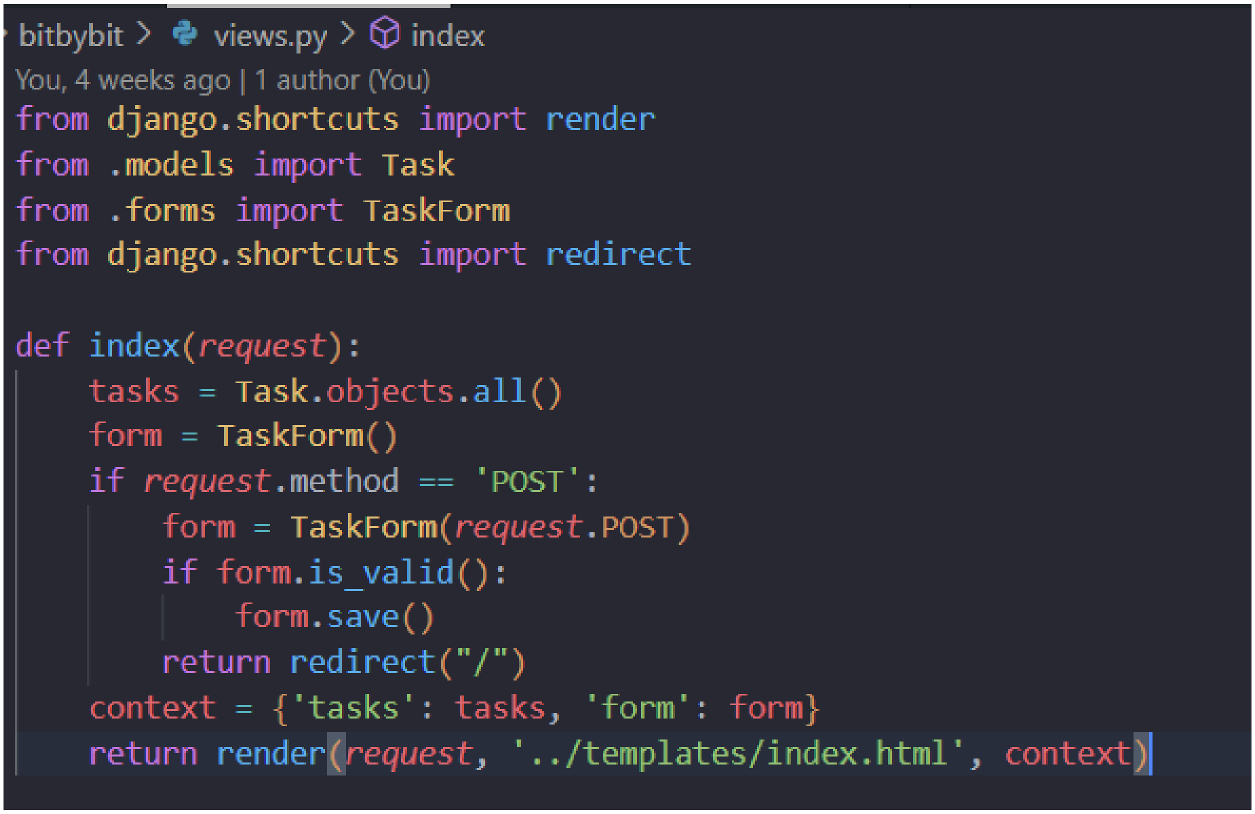


bitbybit > templates > base.html

```
...
{% load static %}
<html>

<head>
    <title>Bit by Bit</title>
    <link rel="stylesheet" href="{% static './style.css' %}">
    {% block head %}
    {% endblock %}
</head>
<body>
    {% block content %}
    <div class='split left'>
        <h1>To Do</h1>
        <form class="pixel-corners pixel-corners--wrapper" method="POST">
            {{ form.text }}
            {% csrf_token %}
        </form>
        {% for task in tasks %}
        <ul>
            <li>{{ task }}</li>
        </ul>
        {% endfor %}
    </div>
    <div class='split right'>
        <h1>Character Box Here</h1>
    </div>
    <!--Character Box Column (column b)-->
</body>
{% endblock %}
```

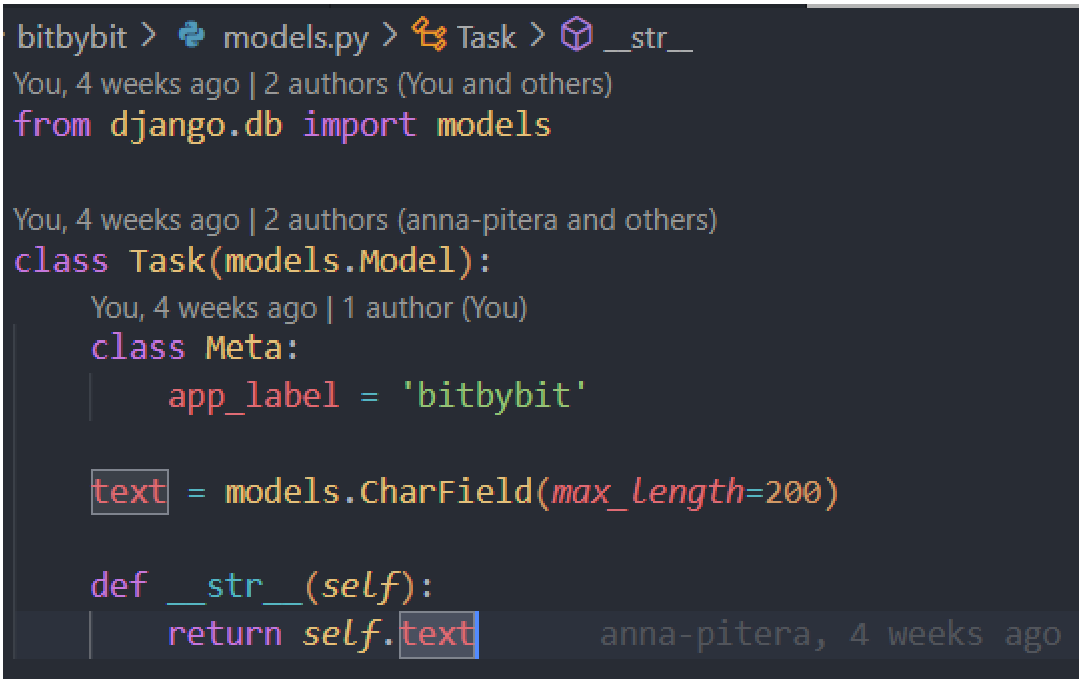
Here’s what views.py looked like:



```
You, 4 weeks ago | 1 author (You)
from django.shortcuts import render
from .models import Task
from .forms import TaskForm
from django.shortcuts import redirect

def index(request):
    tasks = Task.objects.all()
    form = TaskForm()
    if request.method == 'POST':
        form = TaskForm(request.POST)
        if form.is_valid():
            form.save()
        return redirect("/")
    context = {'tasks': tasks, 'form': form}
    return render(request, '../templates/index.html', context)
```

Here’s what models.py looked like:



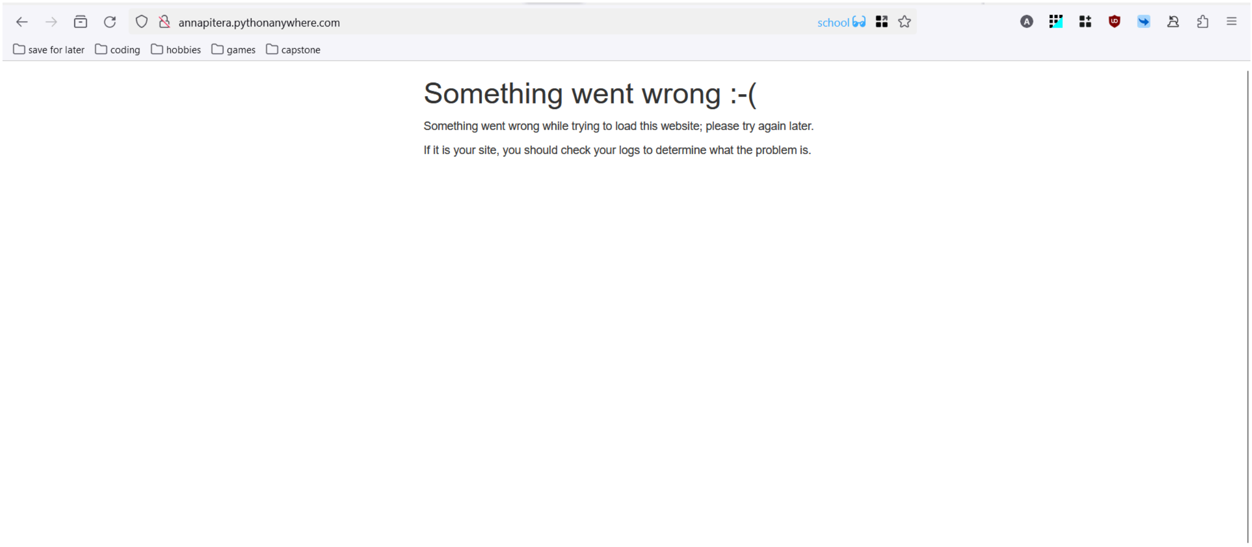
```
You, 4 weeks ago | 2 authors (You and others)
from django.db import models

You, 4 weeks ago | 2 authors (anna-pitera and others)
class Task(models.Model):
    You, 4 weeks ago | 1 author (You)
    class Meta:
        app_label = 'bitbybit'

    text = models.CharField(max_length=200)

    def __str__(self):
        return self.text        anna-pitera, 4 weeks ago
```

For reference, this is what the website looks like right now:



Something went wrong :-(

Something went wrong while trying to load this website; please try again later.

If it is your site, you should check your logs to determine what the problem is.

Cool.

I'll be able to fix it, it's just going to take more effort obviously.

4. Difficulties Encountered this Progress Period

- Django
  - While I have learned a lot about Django compared to the beginning of the semester, I definitely still have a lot more to learn in order to complete the project.
  - The official Django documentation has proven to be really helpful, so I'm going to continue referring back to that. I'm also probably going to try and use Codecademy some more in my free time (preferably not during class, I want to try and get more tangible progress during my class time).
- Git
  - You're going to be so happy to hear this.
  - I had to rebase the repository.
  - Why does this keep happening to me during major comp sci projects? Not sure. I'm terrible with Git, probably.
  - What happened was that I accidentally, very foolishly forgot to add a specific Django settings file to the .gitignore, and that obviously revealed some information that shouldn't be public. It probably wasn't a big deal looking back on it (because who in the world is going to steal the passwords for a website that isn't even functional yet?), but I went back in time and fixed it.
  - It really wasn't that big of a struggle to rebase it though; the struggle came from trying to remember how to do it (because I did it for one of the first semester projects).

5. Updated Trello Board and Discussion

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

Changes:

- Added more links to “Resources” list
- Added a little note to the “Phaser” card in “Resources” list stating that I’m not using Phaser anymore but that I’m keeping the resources there just in case
- Separated “HTML/CSS/JavaScript” card in “Resources” into “HTML/CSS” and “JavaScript/Canvas” to better organize new Canvas resources
- Saved a few Canvas video tutorials
- Added a few mid and high priority tasks to do (also added an automation to automatically sort the list by priority label every time something is added to the list)

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

- First, fix the website so it at least loads.
- Then fix the CRUD functionality with the Django input forms (I have some ideas for how to fix it already).
- Add due dates to database.
- Add due date calendar picker widget to due date text box (I already have resources for this).

7. **Additional Information**

I’m disappointed that the Django stuff isn’t working as smoothly as I’d want it to. It’s also a little frustrating not having that much tangible progress. I guess it’s better to deal with it all now instead of in the last few weeks of senior year, though.

Also, I should mention that there’s not that many commits to the repository because the majority of what I’m doing is easier to do in the PythonAnywhere code editor (because it allows me to update the website quickly and see error logs easier). This is also because, to my knowledge, there’s not really a great way to test Django database stuff in VS Code. The Live Preview extension works well with HTML/CSS/JavaScript, but not Django.