Project Backlog

- Google OAuth login/logout
- **Profile Model**
    - **User**
    - **Picture**
    - **Liked Events**
- **Google maps API to show CMU map**
    - **Default to CMU location**
    - Show event pins on map
    - Pins should show up on event add
- Add events
    - **Event Model:**
        - **User**
        - **Name**
        - **Description**
        - **Longitude**
        - **Latitude**
        - **Specific Location**
        - **Date/ Time**
        - **Tags**
    - Add event with AJAX
    - Create selectable pins for CMU buildings
    - Filter events by tag
    - Liking Events

- Delete event
    - Delete events when time is over
    - Delete events if user who made it wants to

Sprint Backlog

SPRINT 1: Vicky

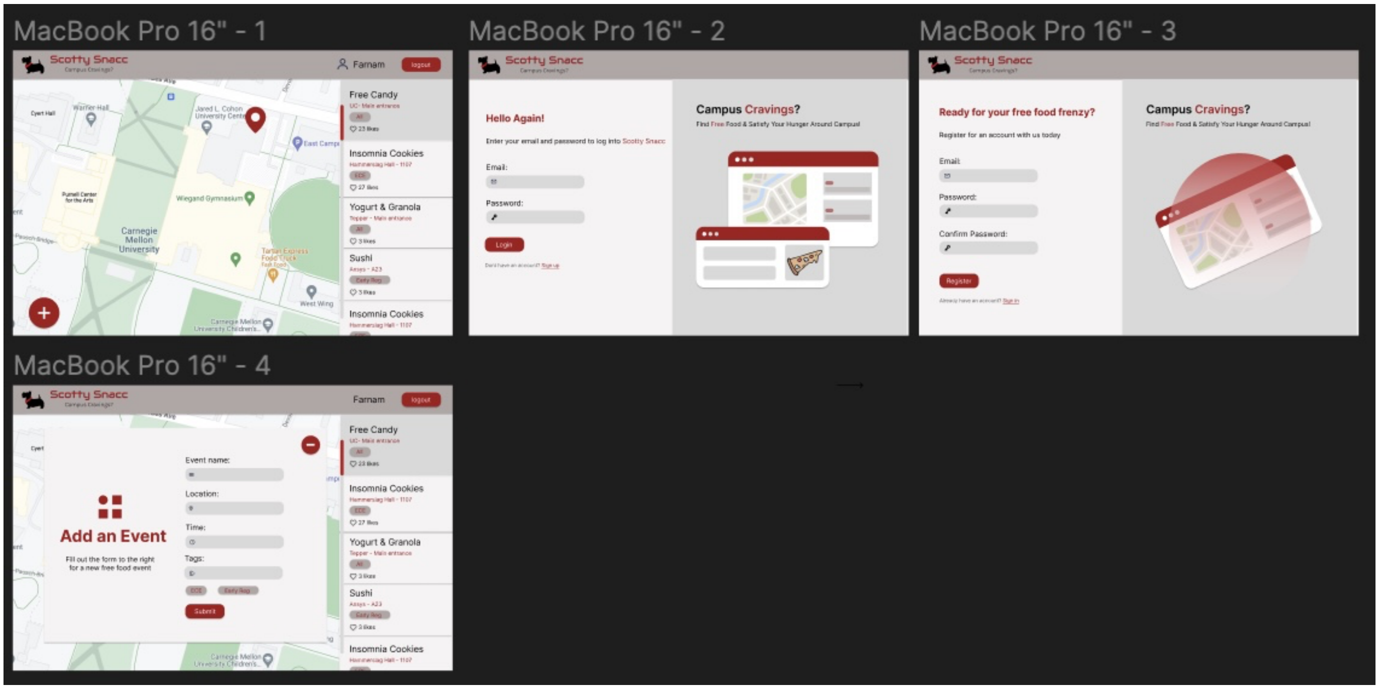
Sprint Goal	Person Assigned	Estimated Time	
Wireframe Drawing with Figma	Erin	3 hrs	✓
Django Setup	Vicky	1 hr	✓
CMU Map Appearance (Google Map API)	Vicky	1 hr	✓
Add Event barebone (AJAX)	Erin	2-3 hrs	
User Model login	Both	1 hrs	✓

MVP DONE!

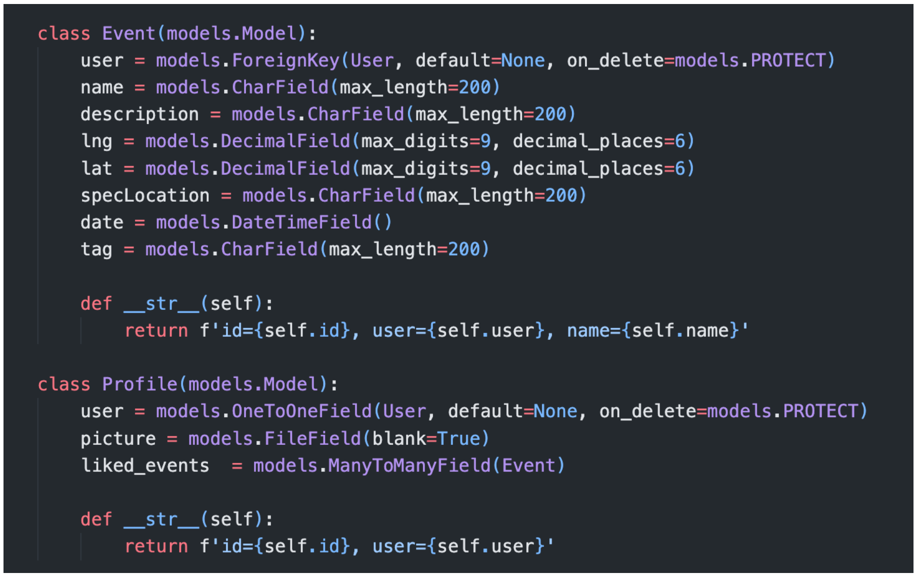
SPRINT 2: Erin

Sprint Goal	Person Assigned	Estimated TIme
UI Aesthetics	Erin	3 hrs
Location Selection with Map	Erin	2 hrs
Tagging Events with Criterias	Vicky	2 hrs
Event Filtering	Vicky	1 hr
Oauth with Google	Vicky	2 hrs
Delete events	Erin	1 hr
Like events	Both	1-2 hr

Mockup



Model



```
class Event(models.Model):
    user = models.ForeignKey(User, default=None, on_delete=models.PROTECT)
    name = models.CharField(max_length=200)
    description = models.CharField(max_length=200)
    lng = models.DecimalField(max_digits=9, decimal_places=6)
    lat = models.DecimalField(max_digits=9, decimal_places=6)
    specLocation = models.CharField(max_length=200)
    date = models.DateTimeField()
    tag = models.CharField(max_length=200)

    def __str__(self):
        return f'id={self.id}, user={self.user}, name={self.name}'

class Profile(models.Model):
    user = models.OneToOneField(User, default=None, on_delete=models.PROTECT)
    picture = models.FileField(blank=True)
    liked_events  = models.ManyToManyField(Event)

    def __str__(self):
        return f'id={self.id}, user={self.user}'
```