Sprint 1 Backlog

Backend

- Configure Docker development environment
- Configure Django project
- Configure GeoDjango setup
- Create basic Landmark model with the following fields:
  - Name
  - Location (Latitude & Longitude)
  - Address
  - Short Description
  - Long Description
  - IsPublished (Controls if point is actually shown)
- Create LandmarkPhoto model for media associated with landmarks with the following fields:
  - Landmark (Foreign Key, associates photo with a specific landmark)
  - Image
  - Caption
  - Alt Text
  - Sort Order
- Create REST API using Django REST Framework (DRF) for queries / communication with application frontend with the following endpoints:
  - Full List: Lists all Landmarks, with their names, locations, and short descriptions for placing map pins on first load

    - Nearby: Returns the names, locations, short descriptions, and first images (if any) for locations within a given radius of a location for powering nearby landmark interaction
    - Detail View: Returns the name, location, short & long description, address, and all associated photos for a specific location, used when user “clicks into” any specific landmark
- Host backend on a publicly accessible server for access by the frontend on non-development devices

Frontend

- Create Leaflet map page and mount it through Django template routing
- Implement a full-screen responsive map layout with custom CSS
- Work with teammates to connect fronted to Django API
- Parse API response and map fields (id, name, lat, long, short_description, cover_photo_url)
- Create Leaflet markers dynamically from API data
- Implement popup rendering for landmarks
- Implement live location tracking