

## Implementation Plan: Governor Nelson State Park Recreation Web App

\*diagram at the bottom

### **Phase 1: Data Collection & Preparation (Week 1)**

- Visit Governor Nelson State Park to collect primary data using ArcGIS Field Maps (I have used this before for my own data collection, and it's very useful and integrated!)
- Capture coordinates and descriptive attributes for:
  - Trails
  - Restrooms
  - Picnic areas
  - Parking lots
  - Initial scenic points
- Organize data in shapefile or GeoJSON format.
- Set up PostgreSQL/PostGIS database with spatial extensions enabled.
- Prepare base schema for each layer using SQL.
- Begin prelim testing of data imports into PostGIS.

### **Phase 2: Frontend Development (Week 2)**

- Build responsive HTML/CSS layout using Bootstrap. Am going the web app route for accessibility. It would not make sense to develop an app for people to install from an App Store for a single park.
- Initialize ArcGIS JavaScript API map and load all static data layers.
- Add popups for park features (restrooms, picnic areas, scenic, etc.).
- Build UI for user-submitted scenic locations:
  - Submission form with validation. Using Survey123, I have done something similar to this before with a 576 project.
  - Point placement on map

- Add controls for device location, layer toggles, and basic search tools.
- 

### **Phase 3: Backend Development & User Submission Integration (Week 3)**

- Build Flask server to handle:
    - Scenic point submissions (insert into PostGIS with validation)
    - Queries for spatial filtering (e.g., trail by distance, elevation possibly?)
    - Optional: elevation analysis (if data available via API or DEM). Will do some digging for this data.
  - Connect frontend submission form to backend via AJAX or Fetch API.
  - Secure routes and sanitize inputs to protect DB integrity.
  - Begin integration testing across devices (mobile/desktop). (over the weekend)
- 

### **Phase 4: Demo, Testing & Final Edits (Week 4)**

- Conduct full walkthrough of user stories to ensure all functions work:
  - Find scenic areas
  - Identify potential running routes with elevation/length info
  - Use search, location, and popups
  - Submit scenic points successfully
- Make UI tweaks based on usability testing.
- Record 5-minute demo video.
- Finalize written report, ER diagram, and logical schema.

Pretty simple linear design, not sure that it needs to be any more complex than this?

