Explore Ohio

Alexander Michelbrink, Brian Myers, Forest Lang, Nadine Nam

Problem + Project Overview

Problem to Solve

Lack of a centralized location for local park information that is built and monitored by community members.

Project Overview

- One-stop website for park information.
  - Photos
  - Features
  - Location
  - Reviews
- Built by the Community
  - Submit New Parks and Create Reviews

Technologies Utilized

Frontend

HTML/CSS
Built website visuals, navigation between pages, and informative feedback.

Bootstrap
Easy and consistent formatting.

JavaScript
Built the responsive and dynamic aspects of pages.

Leaflet + OpenStreetMap
Leaflet: Map Page Functionality
OpenStreetMap: Map Page Tiles

Backend

Python
Means of locally hosting the site and supports its core functionality.

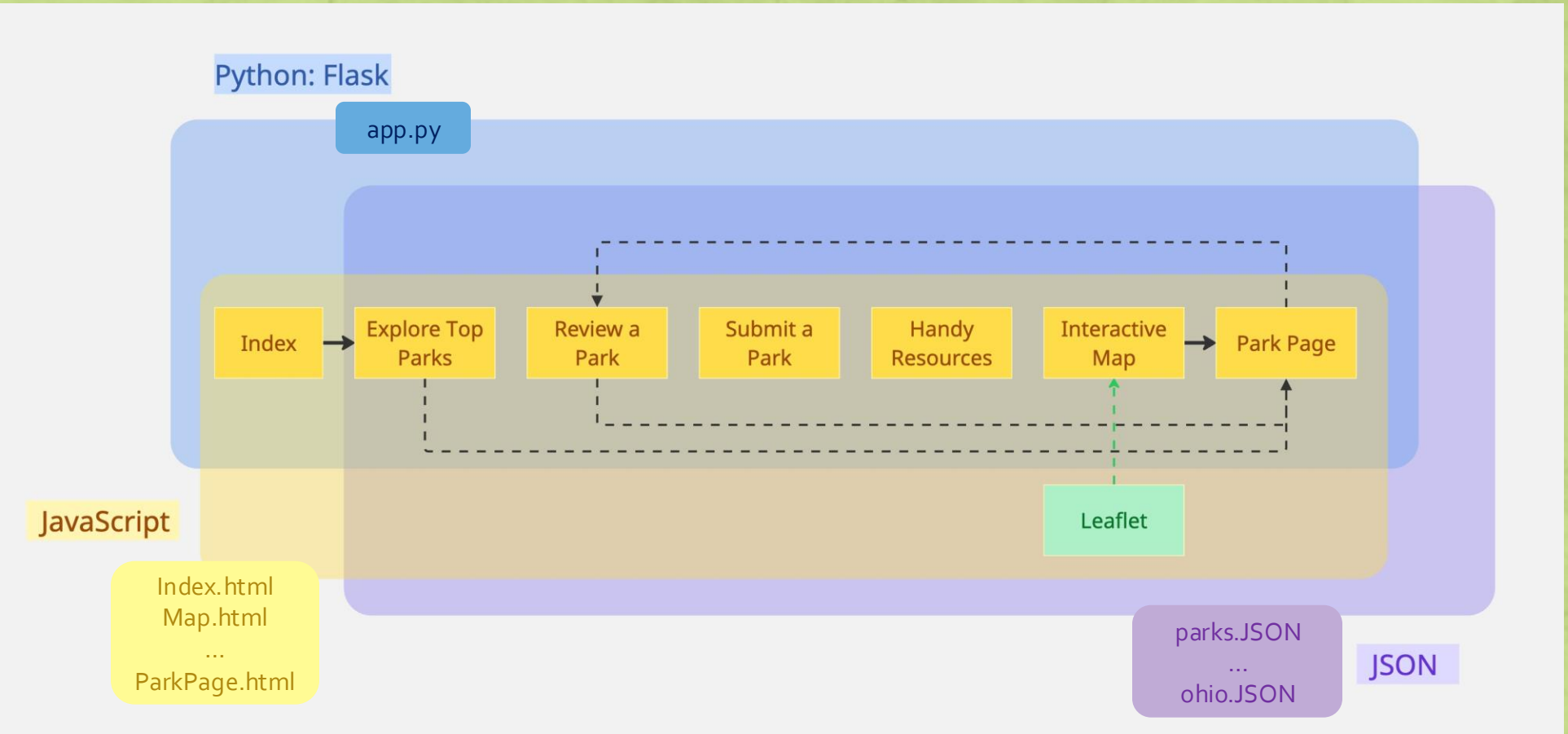
Flask
Handles HTPP requests and allows for server-side logic and updates (the File System).

File System

JSON
Holds most of the data within the website (parks, reviews, tags).

GeoJSON
Holds data necessary for visualizing Ohio on the map page.

Project Architecture

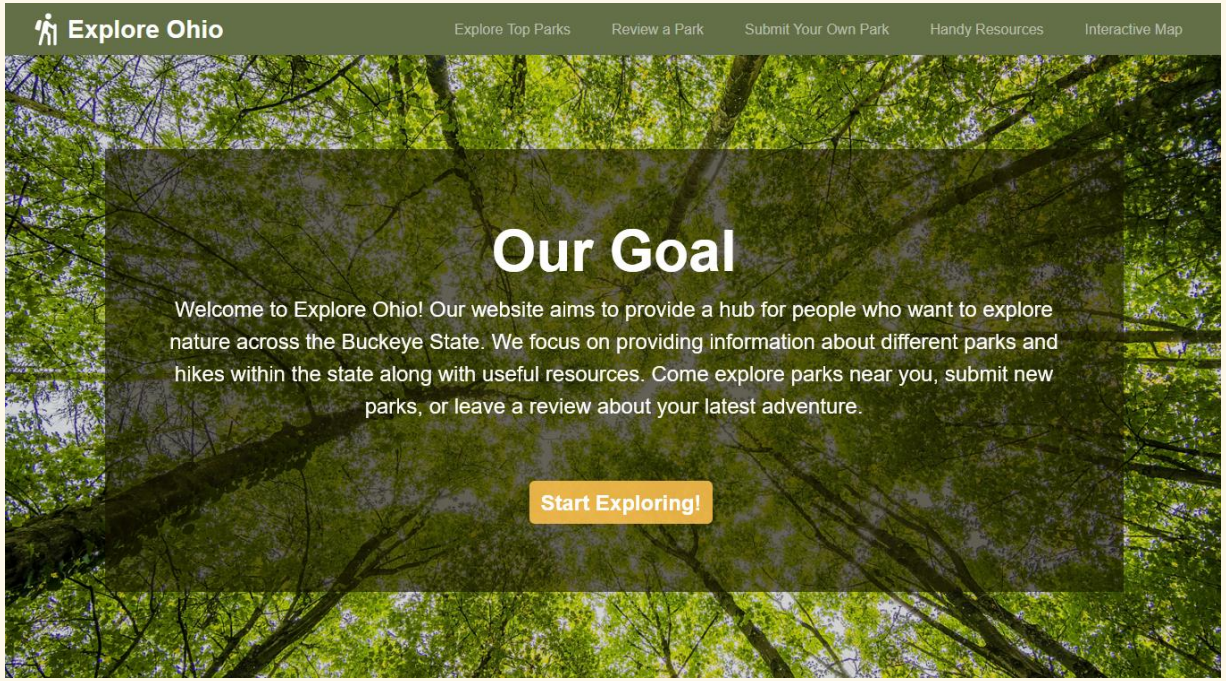


8 Golden Rules

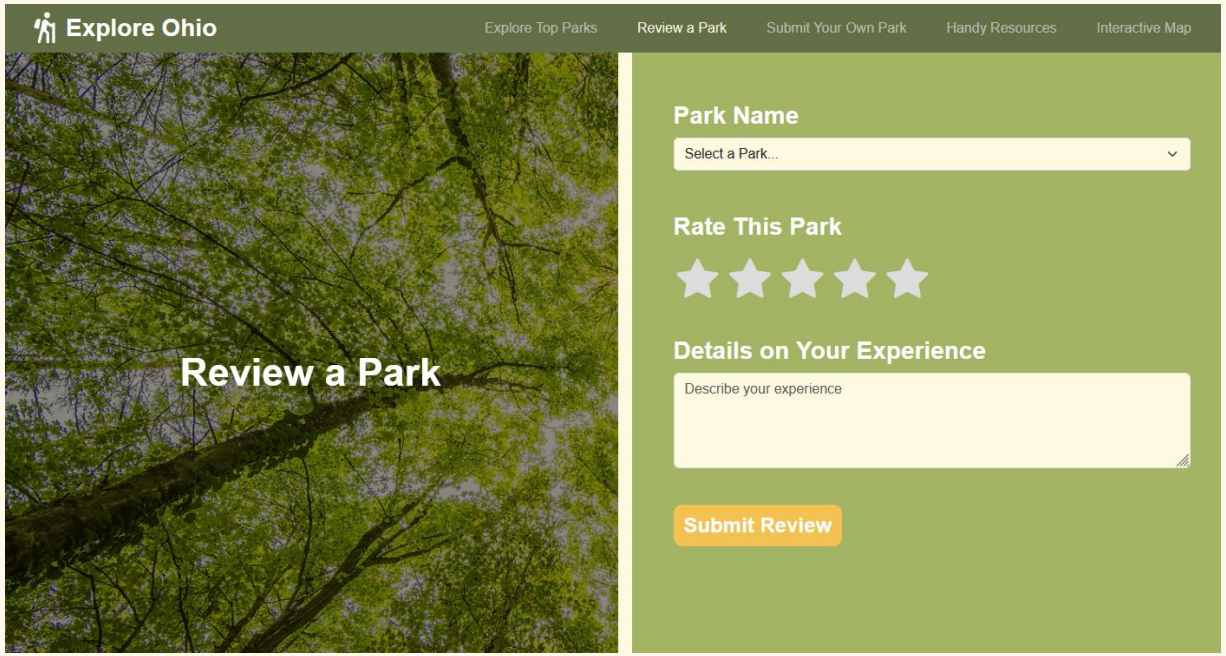
Page by Page

Page #1 - Home Page

- Consistency
  - Several elements are shared across the website
    - Color pallet / fonts
    - Navigation bar
    - Use of yellow for interactable buttons
- Shortcuts
  - Navigation bar allows access to several pages across the website
- Informative Feedback
  - Buttons are highlighted when the user hovers over them



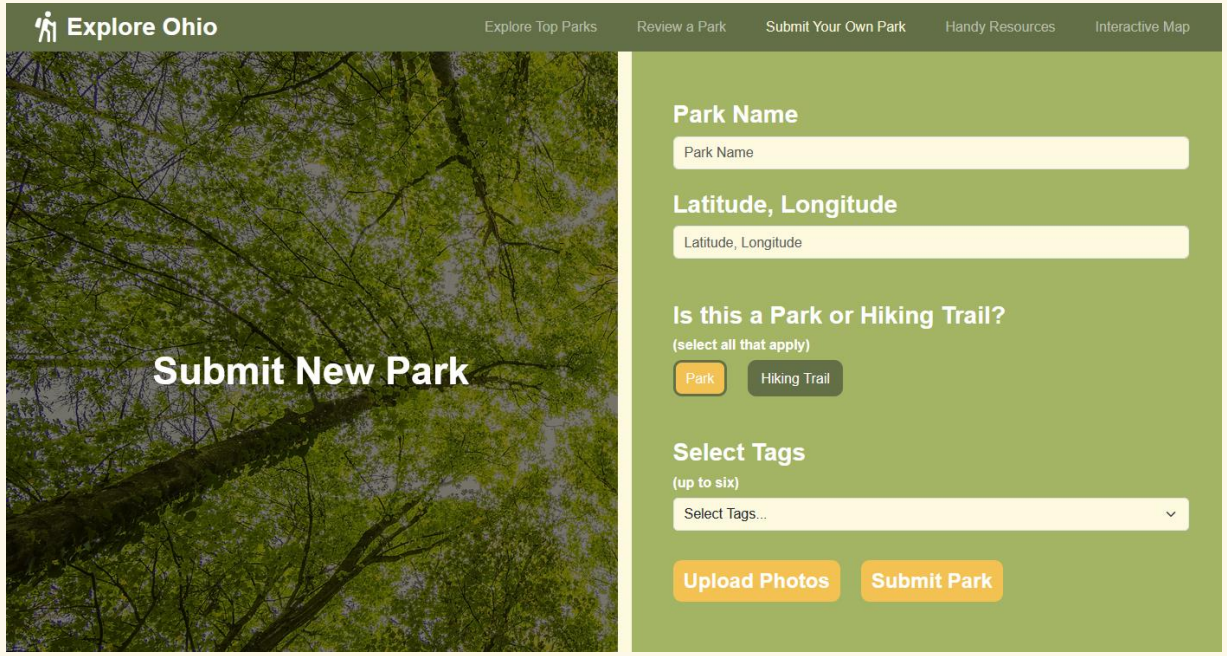
Page #2 - Review a Park



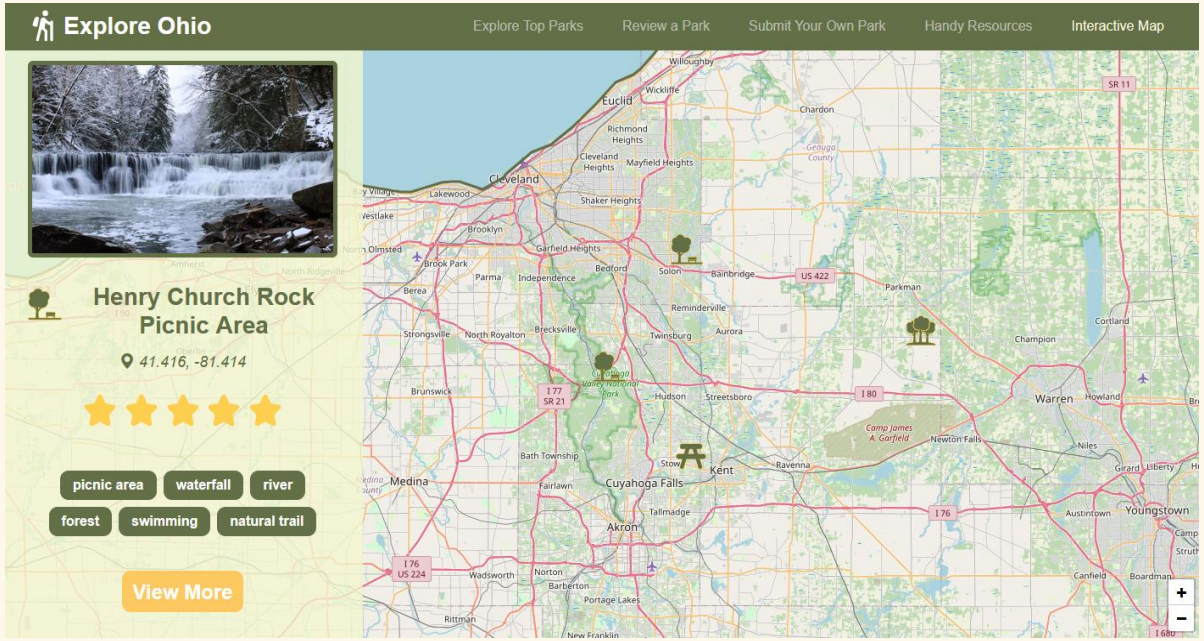
- Informative Feedback
  - Stars react when user clicks or hovers
- Dialog for Closure
  - Submitting a review takes the user to the appropriate Park Page
  - A visual for new Reviews and Ratings
- Easy Action Reversal
  - All inputs can easily be changed before submission
- Internal Locus of Control
  - User creates a review and then it affects the website
- Reduce Short-Term Memory Load
  - Form is only one page
  - The user doesn't need to remember park name (autofill)

Page #3 - Submit New Park

- Consistency
  - Similar setup to the Review a Park page
- Dialog for Closure
  - Submitting a new park takes the user to the new page
- Simple Error Handling
  - If information is not formatted correctly the user will not be able to submit
  - An error message will explain the correct formatting
- Internal Locus of Control
  - User adds new information and then it can be viewed and interacted with
- Reduce Short-Term Memory Load
  - Form is only one page



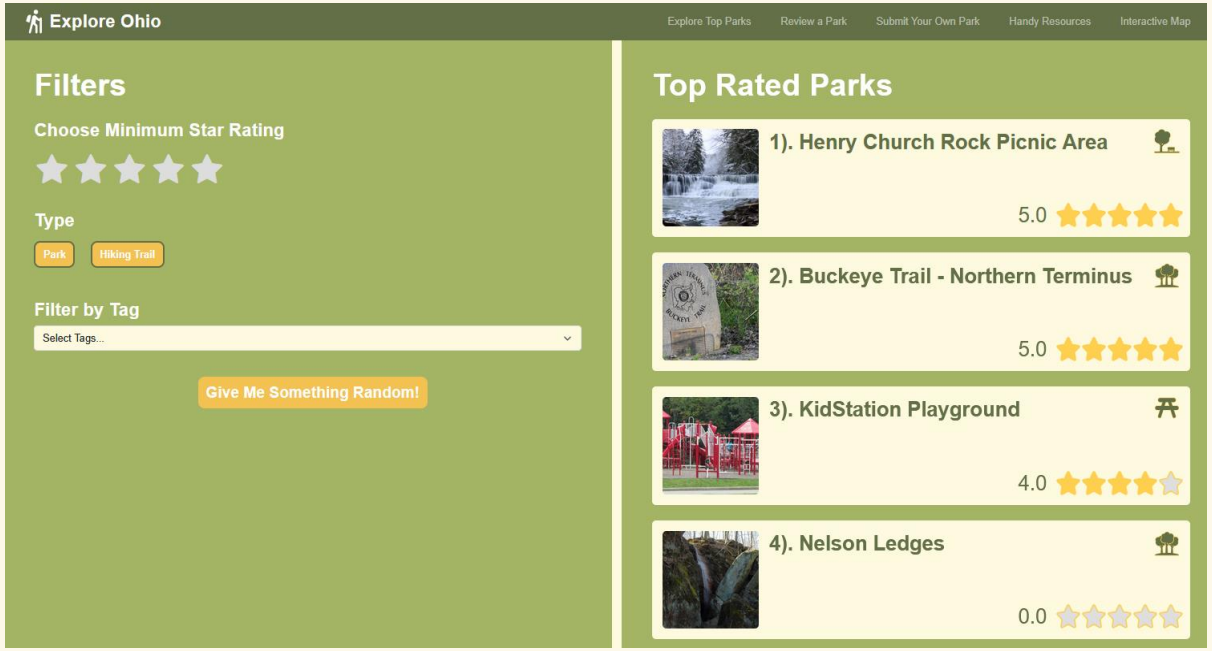
Page #4 - Map



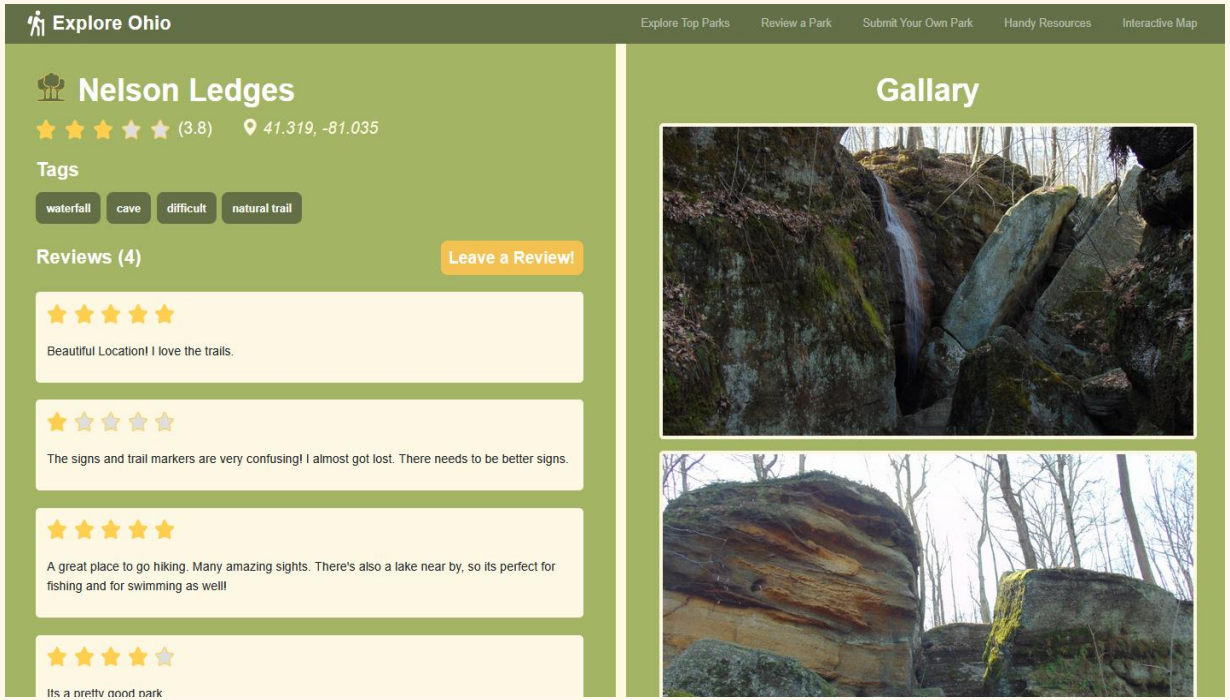
- Shortcuts
  - Users can easily access specific parks from the map
  - Navigation bar continues to be present
- Informative Feedback
  - When park icons are clicked details are displayed on the left
- Internal Locus of Control
  - Users can choose which part of the map they want to look at
  - Users can choose which parks to view

Page #5 - Top Parks

- Shortcuts
  - Random Park button gives quick access to a park the user likely hasn't seen before
- Informative Feedback
  - Filtering by tag immediately affects the results
- Easy Action Reversal
  - Filters can easily be selected or unselected
- Internal Locus of Control
  - Users can filter by type of park or tags
  - Users can choose a minimum rating for the parks they want to see



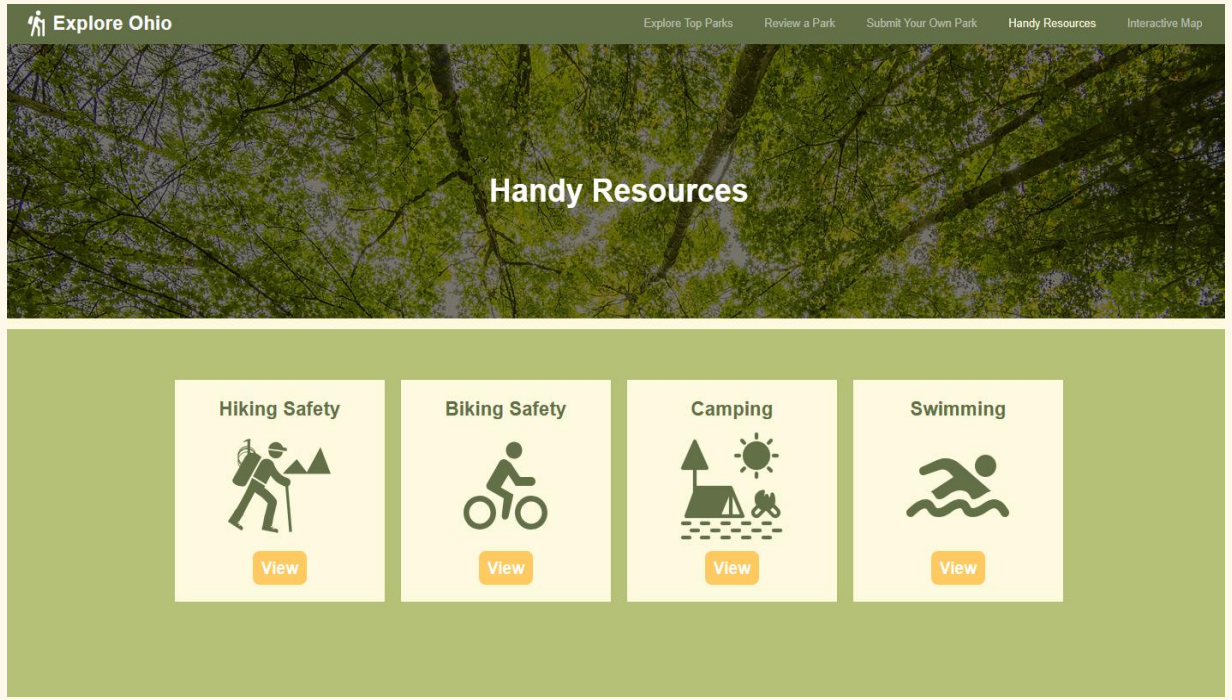
Page #6 - Park Page (example)



- Consistency
  - The same color scheme and font continue to be used across the website
  - Tags continue to have the same format
- Informative Feedback
  - The "Leave a Review" button become highlighted when hovered over

Page #7 - Handy Resources

- Informative Feedback
  - The "View" buttons are highlighted when the user selects them
- Internal Locus of Control
  - The user can choose which specific resources they read about



Project Demo

Localhost: http://localhost:3000/

Conclusion

We are satisfied with our project!

- Achieved our Main Goals
- Website is Visually Appealing
- File System Works Quickly and Accurately

Places for Improvement / Possible Extensions

- Display and store more park information.
  - Maps of trails, trail lengths, size of park, parking info, etc.
- Adding filtering options to the parks map.
  - Including "must be within _ miles of" a location.