Put It In Park



Picture from: http://geology.isu.edu

Miles Chandler, Joseph Engelhart, Ella Robertson, Levi Villareal, and Billy Vo

The University of Texas at Austin

10/04/2019

CS373 – Software Engineering

Glenn Downing

Table of Contents

[1 Motivation 3](#_Toc22817990)

[2 User Stories 3](#_Toc22817991)

[2.1 Customer Requests 3](#_Toc22817992)

[2.1.1 User Story 1 3](#_Toc22817993)

[2.1.2 User Story 2 3](#_Toc22817994)

[2.1.3 User Story 3 3](#_Toc22817995)

[2.1.4 User Story 4 3](#_Toc22817996)

[2.1.5 User Story 5 4](#_Toc22817997)

[2.2 Requests to Developer 4](#_Toc22817998)

[2.1.1 Customer Story 1 4](#_Toc22817999)

[2.1.2 Customer Story 2 4](#_Toc22818000)

[2.1.3 Customer Story 3 4](#_Toc22818001)

[2.1.4 Customer Story 4 4](#_Toc22818002)

[2.1.5 Customer Story 5 4](#_Toc22818003)

[3 RESTful API 4](#_Toc22818004)

[4 Models 5](#_Toc22818005)

[4.1 National Parks 5](#_Toc22818006)

[4.1.1 Examples of filtered/sorted attributes 5](#_Toc22818007)

[4.1.2 Examples of searchable attributes 5](#_Toc22818008)

[4.2 Recreational Activities 5](#_Toc22818009)

[4.2.1 Examples of filtered/sorted attributes 5](#_Toc22818010)

[4.2.2 Examples of searchable attributes 6](#_Toc22818011)

[4.3 States 6](#_Toc22818012)

[4.3.1 Examples of filtered/sorted attributes 6](#_Toc22818013)

[4.3.2 Examples of searchable attributes 6](#_Toc22818014)

[5 Tools 6](#_Toc22818015)

[6 Hosting 7](#_Toc22818016)

# 1 Motivation

This project aims to encourage Americans and tourists to go outdoors by highlighting National Parks and recreational activities in their area. This would serve to promote physical fitness and increase awareness of all that national parks have to offer. In doing this, we hope to answer three questions:

1. What national parks are in close proximity to our users?
2. What recreational areas are in close proximity to our users?
3. What types of sites can visitors expect to see in each area and what activities can they participate in?

# 2 User Stories

## 2.1 Customer Requests

### 2.1.1 User Story 1

Include a description within the splash or “Home” page. User story completed; altered the splash page to match specifications.

### 2.1.2 User Story 2

Include profile pictures, commits, and issues of each team member within the “About” page. User story completed; about page data was dynamically pulled from GitLab.

### 2.1.3 User Story 3

Structure our models in a grid or table form for each of their respective pages. User story completed; structured models in a grid format.

### 2.1.4 User Story 4

Include a link to our API documentation designed by Postman. Our API should be capable of returning a list of models, returning a list of models, returning attributes of the instances, and returning detailed list of models. User story completed; added link to API documentation on About page.

### 2.1.5 User Story 5

Design our instance pages as stated. Three instance pages for each of the three models, five attributes from the model page, and one instance-specific media. This was completed by creating the three instance pages, and filling them with static data.

### 2.1.6 User Story 6

Add pagination - It would be very helpful to be able to page the instances grid when clicking on a model, allowing users to see only a portion of the full list of instances for each model.

### 2.1.7 User Story 7

About page improvement - include what tools you are using, motivations for the website, and link to your gitlab repo on About page. Issue resolved; Levi restyled the pages and made the fetching of commits and closed issues work correctly, as well as adding the purpose, and links to the gitlab and postman.

### 2.1.8 User Story 8

### Build and consume RESTful API. Please expand on the existing schema of the Postman RESTful API you designed in the previous phase. As stated earlier, the API must be configured (at minimum) to allow user to request:

1. List of instances of a model with their 5 basic attributes
2. List of instances of a model
3. Detailed attributes of an instance Your API must pull the data from YOUR database hosted on AWS or GCP. Ensure your website is pulling the data from the API dynamically, and no data presented on the site for a model/instance is static.

### 2.1.9 User Story 9

Highlight table row for model pages - When cursor hovers over an instance in model page highlight the table row. Also, make the entire table row clickable instead of just the name of the instance. Implement this functionality for all instances of every model page. Issue resolved; Levi altered the table to a grid format for all model pages (parks, states and rec) and added an animation where the grid item pops up when you hover over it.

### 2.1.10 User Story 10

Edit model pages - It would look nice if the instance pages were presented in a format other than a columnized list. Maybe some more color or a change in layout would look good, and could even help with loading the data dynamically to have set containers. A few changes could make these pages really pop, and would make the website that much more appealing. Levi changed the model pages from tables to grid, and updated the styles of the model and instance pages to make them much more appealing.

## 2.2 Requests to Developer

### 2.1.1 Customer Story 1

As a user, I would like your website to include at least the project/website name in place of "Logo" at the top left corner. This is so I can know clearly what website I'm on and the purpose of the project. The font/styling can stay the same, I would just like the content to clearly reflect the website for now.

### 2.1.2 Customer Story 2

As a user, I would like to see data on three instances for Recipes, Restaurants, and Grocery Items on their respective pages. This is so I could see what kind of instances would be included in each model, and get a preview of the types of information that will be shared in the final website. The data can be displayed in any visual format for this step; I just want it to appear on the page.

### 2.1.3 Customer Story 3

As a customer, I would like the About Page to have stats dynamically derived from GitLab. This is so I can see how productive the developers I've hired are being. The statistics I would like derived from GitLab are total number of commits, total number of issues, and number of unit tests (probably 0 tests so far, but I would still like it to appear on the page).

### 2.1.4 Customer Story 4

As a customer, I would like the home page to display something other than lyrics. Please change the title of the home page to the title of the project, and the body of the web page to give a brief description of the project. This will help the customer to see what goal of the final version of the website is.

### 2.1.5 Customer Story 5

As a customer, I would like the website to have a color theme other than black and white. This is to make the website more aesthetically pleasing to users. In order to begin this shift in color theme, could the nav bar color be changed from black to something else? Any other color will do - choose one that you feel will best fit with the feel of your final website.

# 3 RESTful API

APIs currently utilized:

* National Park Service (NPS) API - Used to pull information about national parks.
* GitLab API - Used to pull members’ profile pictures, amount of commits, and amount of issues.
* Recreation Information Database (RIDB) API - Used to pull information about recreational activities.
* Center for Disease Control - Chronic Data API - Used to pull information about health varying by state.

Our API (PINP API):

Postman documentation: <https://documenter.getpostman.com/view/9011044/SVtR19mz>

Base URL: api.putitinpark.xyz/

* Requests of models return a list of all parks/recreation/states depending on the URL specified in the request
* Requests of instances return specified park/recreation/state given respective id’s as a parameter.
* Requests of models specifying a park/recreation/state id return a list of instances of that model that offer or exist in that park/recreation/state.

# 4 Models

## 4.1 National Parks

The model “National Parks” represents the 61 parks throughout the United States. This model connects to the “States” model as location plays a large part in park visitation, and it connects to the “Recreational Activities” model as National Parks often include outdoor recreation.

### 4.1.1 Examples of filtered/sorted attributes

1. Name
2. Location
3. Park fees
4. Dates open
5. Visitors per year (or ranking)
6. Date founded

### 4.1.2 Examples of searchable attributes

1. Major sites
2. Facilities
3. Campgrounds
4. Events
5. News
6. Wildlife
7. Seasonal climates

## 4.2 Recreational Activities

The model “Recreational Activities” represents the estimated 1000+ recreational activity opportunities throughout the United States. This model connects to the “States” model as different recreational activities are available based upon location, and it connects to the “National Parks” model as National Parks often include outdoor recreation.

### 4.2.1 Examples of filtered/sorted attributes

1. Name
2. Locations
3. Fees
4. Date
5. Permit required (true/false)

### 4.2.2 Examples of searchable attributes

1. Activity type
2. Sponsoring Organization
3. Facilities
4. Events
5. Tours

## 4.3 States

The model “States” represents the 50 states within the United States. This model connects to the “Recreational Activities” model as different recreational activities are available based upon location, and it connects to the “National Parks” model as National Parks are visited heavily based upon their location and are tied to the ecosystem of the land they exist on.

### 4.3.1 Examples of filtered/sorted attributes

1. State name
2. Physical activity by state
3. Number of national parks
4. Number of recreational activities available
5. Population

### 4.3.2 Examples of searchable attributes

1. Native animal species
2. Native plant species
3. Bodies of water
4. Land characteristics
5. Tourist attractions

# 5 Tools

* React JS - JavaScript library, namely using react routing
* Node - JavaScript run-time environment that executes JavaScript code outside of a browser
* Express - Web application framework for Node.js, designed for building web applications and APIs
* GCP - Used to deploy and host the web application
* Bootstrap - free and open-source CSS framework directed at responsive, mobile-first front-end web development
* Postman - Used to document our API and output in HTML format.
* Gitlab - Used to host our code and provide CI/CD environments
* Pixabay - Used to find royalty-free images for the static site
* NameCheap - used to provide website URLs

# 6 Hosting

URL: putitinpark.xyz (and putitinpark.me)  
Website URLs were obtained using NameCheap, and the site itself is hosted using Google App Engine. Details to deploy the app using app engine can be found here <<https://cloud.google.com/appengine/docs/standard/nodejs/building-app/deploying-web-service>>, and are also listed below.

* Clone the repository onto your local machine.
* Navigate to the cloned repository directory.
* Install the [gcloud command-line tool](https://www.google.com/search?q=install+gcloud+command+line&rlz=1C5CHFA_enUS809US809&oq=install+gcloud+com&aqs=chrome.1.69i57j0l4j69i60.6729j0j4&sourceid=chrome&ie=UTF-8).
* Run ‘gcloud app init’ in the repository directory, then select the appropriate project from the list provided (You must be invited to the correct gcloud project).
* When ready to deploy, run ‘gcloud app init’.