RouteHunter

A Locally Hosted Rock Climbing Route Search Application

By: Andrew James McNeil

University of Maryland, College Park

ENAE 380 - Section 0203
Final Project Report
Dr. Mumu Xu,
December 14, 2020

Table of Contents

Executive Summary

One of the hardest parts of rock climbing is learning how, and where, to practice the sport. There is loads of data online, inputted by everyday people; yet, for beginner climbers (and even for experienced ones) the sheer quantity of routes can be overwhelming. This area of disconnect led me to develop RouteHunter. The RouteHunter program is a software application that allows the user to easily query top rated climbing routes that are in a given geographic radius, at or below a difficulty the user sets themselves. This removes the confusion and overwhelming nature of online route databases and breaks down many entry barriers to the sport.

Mountainproject.com is the main source, outside of guidebooks, that climbers use to find route information. It is equipped with great search tools to find routes, however, for the inexperienced climbers who do not know any “crags” (cliffs with climbing routes on them) it can be very difficult to know which crag is suitable for their skill level. This is the main reason I decided a more specific search tool for the website is necessary; it allows users to search multiple crags at once - something Mountainproject.com does not have tools for on their website. Once I learned the API can access data solely based on latitude and longitude coordinates, I devised RouteHunter to cycle through the best climbing routes at a given grade.

My main motivation behind this project is my passion for climbing. I wanted to use this Final Project opportunity to create a program I will actually use in my day to day life. I gained help from many of my climbing friends for testing and ideas on how to make the program more user friendly and add features that expand upon this basic idea of data scraping; and I am glad to say I am proud of how well the project turned out. Below is the report for this project,

Thank you, Andrew.

Process and Implementation

I began RouteHunter development by familiarizing myself with how API's work in general. API's are an easy way to pull public data from a company's website through HTTP requests. Mountain Project's API is extremely well documented so learning their keys for an HTTP request was simple. After implementing code to do basic API queries, I then created the user interface through the PySimpleGUI library. This library makes formatting UI's convenient, giving me flexibility to make the actual substance of RouteHunter better. I wanted user's to immediately understand what information they can input, and remove basic limitations such as capitalization, abbreviated state names, and other inputs that are not normally applicable. For example, I did not want to force users to input latitude and longitude coordinates, as that is not easily accessible information, so I mapped city names to their coordinates so users can more easily send query requests. I chose to format the pulled data into a pandas dataframe as I learned it is one of the best formats for large data organization, and was able to edit it as an HTML table through CSS styling, allowing RouteHunter to be aesthetically pleasing.

After the UI was set up, I spent multiple days testing, having friends run tests, and properly debugging the program as much as possible. This by far took the longest as large data manipulation can be extremely sensitive to even the most minor perturbations of inputs. I added details to the program such as printing process messages on what the program is currently doing, and giving users a pop-up message after sending a query. I find small additions like these make a program feel much more thought out and overall creates a better user experience.

Problems and Solutions

My initial proposal for RouteHunter was a website, not a local UI; however, Mountain Project's API is well protected against "bots" making requests. Having another degree of separation from the request, specifically the step of pulling data from a website and not directly from the user's computer, was enough to throw "HTTP 403 Forbidden" errors. I fully created a website using the Flask web development tool, and formatted all of the input boxes I needed, but the API was still blocking the queries. Even after spending hours trying to figure out a work around I decided the website would not be able to be functional. An example of this troubleshooting I did was trying different methods outside of the default Python urllib.request library, Python's Request library, inputting user-agent headings with each request, implementing the core code on a free website builder like Weebly, and more; however, the API was too well protected. This is the point where I began learning about creating local UIs with Python and adding more front-end capabilities for RouteHunter to allow the program to still be functional.

Another problem I encountered was extraneous results from the API. For example, a search in New Hampshire would randomly generate one or two climbing routes in Utah. To combat this, I edited my "city name to geographical coordinates" filter by implementing the geopy library. I was able to use the library's mapped dictionary of city names and their respective coordinates to remove many of these extraneous results. Similarly, the API would return routes that were not within the difficulty range the user inputted. I learned this was caused by faulty data inputs to Mountain Project itself. To remedy this, I manually inserted my own climbing grade difficulty filter, so even if the API incorrectly pulled extraneous routes of grades harder than the user requested, the user will only be shown the correct routes without errors.

Conclusion

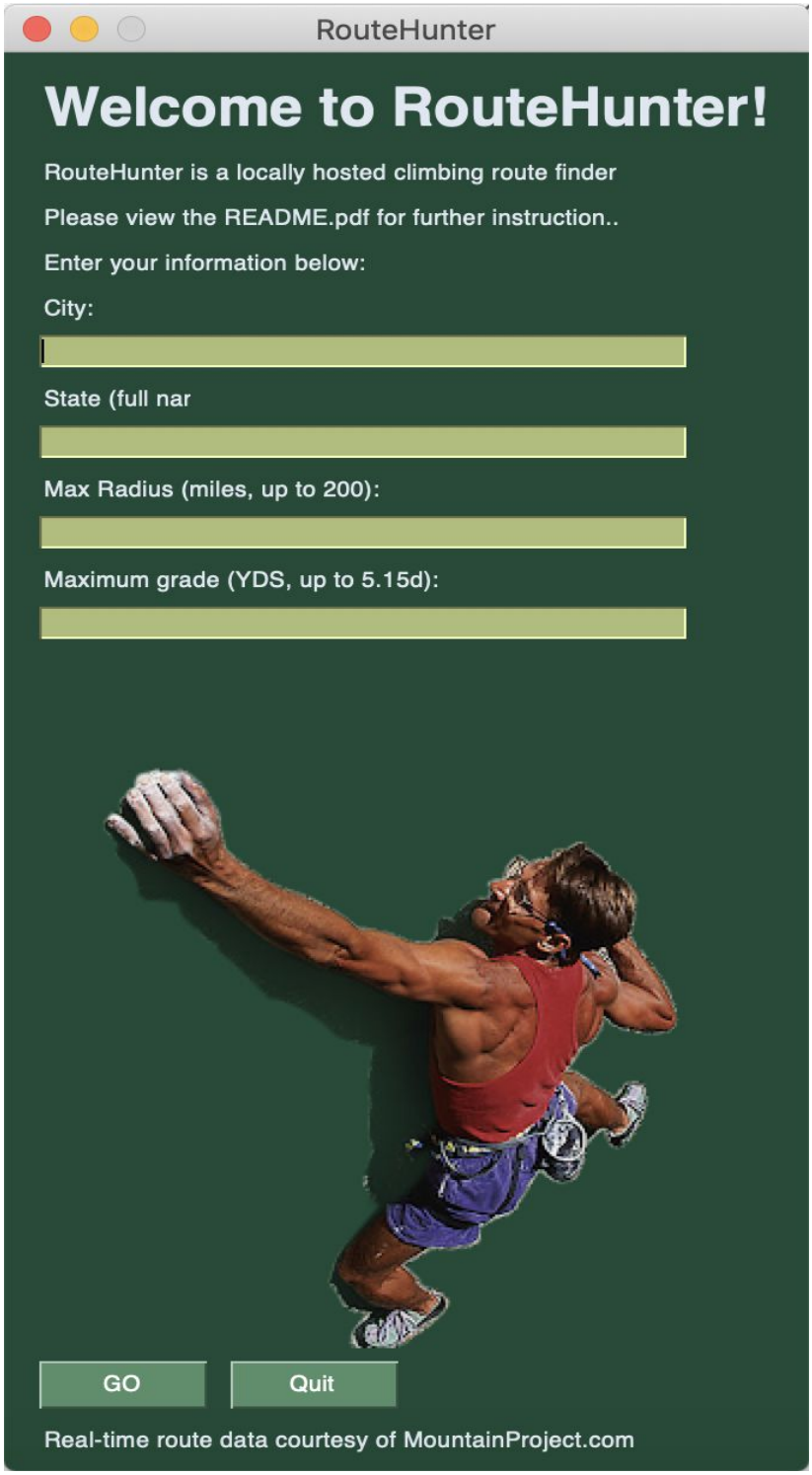
Over the course of this project, I learned many new skills such as API queries, UI creation, large data manipulation, software documentation, web development, and more. In addition, the vast amount of library documentation I read through for this project has equipped me with universal coding skills that will benefit for the entirety of my technical programming career. The failure of making a website taught me extensively about API licensing laws, improving my ethical decision making when handling data, or coding in general.

RouteHunter is a software application I genuinely see myself and my fellow climbers using. The program adequately cuts down planning time for rock climbing trips significantly, and it does it quite elegantly. I feel I have succeeded in breaking down barriers to the sport of rock climbing, even if those who can use RouteHunter are my peers since it is not public.

Thank you for the opportunity to create such a challenging and fun project.

Appendix A

RouteHunter UI



Appendix B

RouteHunter Search Result

Name	Location	Grade	Stars
Anonymous Flake Right	[Virginia, Shenandoah & NW VA Region, Elizabeth Furnace, Buzzard Rocks]	5.8-	4.4
Cornice	[Virginia, DC & Northern VA Region, Great Falls, Cornice]	5.7+	4.4
Strawberry Jam	[Maryland, Rocks State Park, Pinnacle]	5.8+	4.2
Blue Rose	[Maryland, Patapsco Valley State Park, Ilchester]	5.8+	4.1
Breakaway Right	[Maryland, Rocks State Park, Breakaway Wall]	5.9	4.0
Hard Up Direct	[Maryland, Harper's Ferry, Maryland Heights, MD, Sign Wall]	5.8	4.0
Vertical	[Maryland, Rocks State Park, Breakaway Wall]	5.6	4.0
Romeo's Ladder	[Virginia, DC & Northern VA Region, Great Falls, Romeo's Ladder]	5.7	3.9
Herbie's Horror	[Maryland, Carderock, Jungle Cliffs]	5.9	3.9
Breakaway left	[Maryland, Rocks State Park, Breakaway Wall]	5.9	3.9
Sterling's Crack	[Maryland, Carderock, Jungle Cliffs]	5.7	3.9
Snowflake	[Virginia, DC & Northern VA Region, Great Falls, Seclusion]	5.6	3.8
Pure Fun	[Virginia, Shenandoah & NW VA Region, Shenandoah National Park, Old Rag, Summit Crags]	5.7	3.8
Romeo's Left	[Virginia, DC & Northern VA Region, Great Falls, Romeo's Ladder]	5.9	3.8
Jim's Throne	[Pennsylvania, South Central PA, Pond Bank, White Rocks, 11 - Jim's Wall]	5.4	3.8
Triple A	[Maryland, Carderock, Hades Heights]	5.8	3.8
Slab Master	[Pennsylvania, South Central PA, Safe Harbor, Safe Harbor South, Super Slab Sector]	5.9	3.8
Melungian Brotherhood	[Virginia, Shenandoah & NW VA Region, Elizabeth Furnace, Buzzard Rocks]	5.8	3.7
Seven Wishes	[Maryland, Sugarloaf Mountain, Middle Earth]	5.6	3.7
Sand Box Corner	[Virginia, DC & Northern VA Region, Great Falls, Sandbox]	5.4	3.7
Where Eagles Dare	[Pennsylvania, South Central PA, Safe Harbor, Safe Harbor South, Autumn Arch Sector]	5.8+	3.7
Trudie's Terror	[Maryland, Carderock, Hades Heights]	5.6	3.7

Appendix C

Works Cited

“Geopy”. Geopy Authors, 2020. https://github.com/geopy/geopy

“Mountain Project API”. Adventure Projects, Inc. , 2020. https://www.mountainproject.com/data

“Pandas”. NumFOCUS, 2020. https://pandas.pydata.org/

“PySimpleGUI”. The PySimpleGUI Organization, 2020.

https://pysimplegui.readthedocs.io/en/latest/

“Python”. Python Software Foundation, 2020. https://www.python.org/