CS Capstone Technology Review

December 7, 2018

Auction Hunter

Prepared for

Ryan Kalb

Prepared by

Group 4
Auction Hunter
Alexander Jacobson

Abstract

Auction Hunter is designed to be a website the collects information about salvage car auctions to help users find the right purchase. It tracks past auctions and alerts users of up coming auctions that have cars that meet their criteria. Auction Hunter estimates the value of the car based on base auctions and photos of the damage. Using machine learning and access the auction's photos Auction Hunter tries to estimate the cost of the damage to help users make an educated purchase.

Contents

1 INTRODUCTION

1.1 Auction Hunter Background

Auction Hunter has two main pieces that work independently of each other: the web crawler / data processing portion and the web front end. The web crawler continually searches auction sites for data and inputs into a shared database. The web front end then allows Auction Hunter users to interact with its data and set alerts for auctions. This technology review will discuss programming language options for the website, which hosting solution Auction Hunter will use, and what continuous integration software will be used to tie the front and the back-end together.

1.2 Programming Languages

Auction Hunter does not need to use the same programming language for the front and backend because web technologies and trends tend to be outliers from general programming trends. Front end technologies also have different sized stacks ranging from basic HTML generation all the way through full stack deployments with front end and backend components linked together. Within a given language there are also various frameworks Auction Hunter could use to generate a front end. For example, JavaScript has React as well as Angular.

1.3 Hosting Options

The main trade-offs between hosting options revolve around ease of use versus level of control and adaptability. There are hundreds of different options when it comes to hosting a web service and most of them strive for lowest cost and ease of deployment. These two factors are always at odds with control and adaptability because the easier it is to deploy the less control the provider is going to give you. For example, the easiest solution is to pay a provider to host static HTML or even PHP scripts. From start to functioning website is only the time it takes to enter credit card details. The issue with simplicity is that all advanced features are completely abstracted away from the user. Auction Hunter needs to have automatic jobs run constantly which is just not possible with turn-key solutions.

1.4 Continuous Integration Systems

Auction Hunter is a complex project which will require a myriad of different configurations and operations to be performed. A continuous integration system is used to aid developers in maintaining a large code base with lots of different options. The system's job is to take code from a repository and get it running for development, testing, or even final production deployments. Continuous integration systems can also be extended to include automated testing and other more advanced features to help developers spend more time coding and less time trying to get the code to run.

2 WEBSITE PROGRAMMING LANGUAGE

2.1 Criteria

Picking the programming language is one of the first steps when starting a new project. Auction Hunter has pretty simple criteria for its front-end language. First and foremost, it must be either already known to the developers working on the project or at least straightforward for them to learn. The speed of development and relevancy to the type of project are the next most important. Given the limited amount of time and number of features Auction Hunter requires, faster to write is better than faster to run. Since this project will be used by less than a thousand people (best case) it does not need to be optimized for speed, it needs to be just be working and stable. This eliminates most statically typed languages as they require too much code and memory management to be viable for rapid development.

2.2 JavaScript

JavaScript is an obvious choice being the only language that runs natively in the browser, meaning that any other programming language will either need to be transpiled into JavaScript or supplemented with it. JavaScript is dynamically typed and most developers have at one point written some basic scripts in it. JavaScript is easy to get started with, but can be difficult after getting into more advanced areas due to the looser syntax when compared to other dynamically typed languages [1]. The two main web frameworks for JavaScript are AngularJS and React. Angular is a full suite of software tools including a compiler and package manager while React allows the developer to use whatever tools they want. They are similar with the main trade-off being an all-in-one solution versus a more piecemeal approach.

2.3 Python

Python is dynamically typed and is extremely easy to learn. All the team members of Auction Hunter have already learned Python so it would be a great choice for this project. The downside of Python is that it cannot run natively in a browser and thus must still rely on JavaScript to do client-side operations. One common approach is to use Python or some server-side language to handle API calls to the database and use JavaScript to help render the website on the client side. This is most likely the approach that the Auction Hunter team will take as it allows the use of Python for rapid and powerful development while making it easy to write and manage the user interface in JavaScript. Python's two main web frameworks are Django, which is usually used for larger scale projects, and Pyramid, a more compact but not quite as all-encompassing framework.

2.4 PHP

PHP is a server-side language used since nearly the beginning of the web. Like Python it suffers from the problem of not being able to run natively in the browser and relies on sending JavaScript files to the client to run. Do to PHPs age it has countless web frameworks all of with have different styles and features. Although much effort has been spent to remedy this, PHP still has a reputation of being difficult and confusing to use because of the inconsistency of its language design [2]. The benefit however, is PHP is the go-to standard if you are trying to get a simple site working quickly. It runs on every hosting provider and its basics are simple to learn. Only one of the Auction Hunter developers has used PHP before and therefore is not as great of an option as Python.

3 Hosting Solution

3.1 Criteria

Auction Hunter is a website and thus needs to be hosted on an always online server accessible from the internet. The main criteria for hosting, in order of importance, are cost, flexibility, easy of use, and scalability. The different ends of the hosting spectrum go from shared hosting, which is the cheapest and easiest to use but the least flexible, all the way to self-hosting which can be the most expensive and most difficult to use but offers the greatest flexibility.

3.2 Shared Hosting

Shared hosting is the simplest way to host a website, but it only allows static web pages with simple backend infrastructure. Shared hosting would not be a great option for Auction Hunter because it needs to do more than just serve basic web pages. It requires scripts to run in the background and crawl websites for data without any user intervention. While shared hosting is the easiest to setup and use, it is too simplified for the more complicated needs of Auction Hunter.

3.3 Cloud Hosting

Amazon's AWS platform and Googles Cloud Hosting are the two front runners of cloud hosting. Both would allow Auction Hunter tons of flexibility and a relatively small monetary cost. However, one of the group members already has their own dedicated server online 24/7 and can host Auction Hunter for free while it is in development. In the longer term, a cloud provider will be a better option as the project will not rely on one single person to pay the bill. In the future, after most of the development has been completed, the customer for Auction Hunter may wish to deploy their own version of Auction Hunter on a cloud provider. In this case AWS is a better fit because it would require only minimal amounts of configuration changes to get Auction Hunter running on it. Google's platform runs Python projects directly but it requires the project to be in a specific format [3]. This means it would require some code changes to get Auction Hunter running on Google's platform.

3.4 Self-Hosting

Self-hosting is the most complicated to setup and maintain, but it is also completely cost free if Auction Hunter can piggyback on a server that is already always online. Self-hosting offers even more flexibility then any cloud offering, but it would not be cost effective to setup a server just for Auction Hunter. The Auction Hunter team has access to an always online server and since the option is available, Auction Hunter will most likely go the self-hosting route because it is free when compared to a monthly cost for using a cloud provider.

4 Continuous Integration Systems

4.1 Criteria

Most continuous integration systems are very similar to one another and choosing one is more dependent on the programming languages it is compatible with rather than any other factor. JavaScript and Python are both interpreted languages meaning they are compiled as they are run instead of ahead of time. Since we are using an interpreted language that does not require a discrete compilation step, picking which piece of continuous integration software to use is mostly a matter of personal preference and language / code repository compatibility. What language and what repository Auction Hunter uses will drive the choice of continuous integration software.

4.2 Jenkins

Jenkins is one of the most common choices with continuous integration software, because of its vast compatibility and impressive flexibility. Jenkins has an extensive array of plugins that expand its compatibility even further. Jenkins' main draws are that it is completely free and very easy to setup and use. Several of the Auction Hunter team members have also used Jenkins before which would reduce the time to setup and maintain it.

4.3 Travis CI

Travis CI is very similar to Jenkins in its compatibility and ease of use. Travis CI's plugin library is smaller than Jenkins' is, but Travis CI has compatibility with most products and technologies that the Auction Hunter team is aware of. One notable benefit of Travis CI over Jenkins is its seamless integration with Github. If the Auction Hunter team chooses to host the project's code on Github then Travis CI would be a good option because of how easy it is to use Travis CI with Github.

4.4 Gitlab

Jenkins and Travis CI are both compatible with Github and Bitbucket, but GitLab has its own continuous integration software built in and therefore is only compatible with itself. Any of these options would work for Auction Hunter as they all support Python and JavaScript. However, since Auction Hunter will already need to host one of the continuous integration systems it might be easier to just host an instance of Gitlab which has a code repository and a continuous integration system rolled into one.

5 Conclusion

The first choice that needs to be made for Auction Hunter is what programming languages the team will use which will drive a lot of the other technology choices. Things like the hosting solution and build system depend greatly on what languages they need to support. A hybrid solution of Python and JavaScript seems to be the best solution for the front-end which will allow the Auction Hunter team to use the same language for the front and back ends. This means that self-hosting will be the best option due to its affordability and flexibility. This will also allow the Auction Hunter team the ability to host their own code repository, continuous integration system, and any other tools that might be needed as development progresses.

References

[1] The good and the bad of javascript full stack development. [Online]. Available: https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-javascript-full-stack-development/

[2] Advantages and disadvantages of php frameworks. [Online]. Available: https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-php-frameworks-c046d50754e5

[3] Structuring web services in app engine. [Online]. Available: https://cloud.google.com/appengine/docs/standard/python3/configuration-files