Software Ranked++

Team Guinan - Project Report

ENSE 374 - Dr. Tim Maciag

December 2nd, 2022

Nathan Cameron, Shazil Siddiqui, Vida Pathan

Table of Contents

List of Figures

Project Introduction

Background and Motivation

As the modern world continues to develop, and with the rapid advance of technology, we have access to numerous tools to solve our daily queries. However, with a large abundance of software technologies, future developers are faced with a new challenge of selecting the right languages to focus on. In order to ease the software language selection process that all developers inevitably face, team Guinan has created Software Ranked++.

Project Idea and Purpose

Software Ranked++ is an online website that allows seasoned programmers from verified organizations to rank software languages, anywhere from a 1 to a maximum ranking of 5. This will allow young developers, and even start-ups, to benefit from these certified rankings, as it provides them with a quantified view of professionals on trending languages. For those interested in seeking more knowledge, we have implemented a detail page that outlines the pros and cons, general syntax for “hello world”, and a graph depicting the trend for stability with respect to each programming language.

Audience

We have divided our audience into two categories. The first category is composed of those qualified to submit rankings. These rankers will be provided with an authentication key which is required in the sign-up process. Once authenticated and signed in, they can submit a carefully decided rank per language. Any secondary rankings will replace the original rank. This feature maintains the validity of our product. The second category is composed of non-authenticated users who are not required to login and cannot submit rankings. They will simply benefit from viewing the rankings and surfing through the detail pages.

Project Planning

Documents

The first document that we considered was the Business Case document, which reflects on the very proposal of the project idea. It allowed us to explore the project background and business needs. A large portion of this document explores the cost to benefit analysis, which for our project idea, was not entirely applicable, although it allowed us to discuss future advertisement expenses and hours invested in development. The Project Charter was another initial document that sheds light on those involved and affected by the project. It allowed us to envision the proper milestones of our product. We held a meeting and collectively decided on reasonable deadlines with respect to incremental deliveries for the first Activity, and solidified our objectives.

This was a good segue to both the Project Requirements and Scope Statement documents, which allowed us to identify the tools we wanted to use to achieve our previously defined objectives and declare our envisioned Model View Controller and the Minimum Viable Product. The team decided to create the bare structure with HTML5 and CSS3, and later aimed to use Node.js and Express.js to support our backend. With the Scope Document, the team was able to highlight key features of the product, containing a login/sign-up page, home page with and without the ranking ability, and detail pages for each language.

Once the overall structure of Software Ranked++ became clear, the team traversed the Project Roles and Responsibilities document which allowed the team to assign titles for future delegation of tasks. Mr. Shahzil Siddiqui accepted the roles of a Project Manager and Full Stack Developer. He was, thus, made responsible for organizing scrum meetings, ensuring deliverables were on track, and putting together the vlogs. Mr. Nathan Cameron’s extensive experience with GitHub qualified him as the GitHub manager and Full Stack Developer. He was responsible for managing the GitHub and Wiki Pages. Miss. Vida Pathan was responsible for the extensive research for each language, and as such, she was the Project’s Research Manager and Developer.

Diagrams

While the objectives were made clear through the documentation, interactions of objects and defined structure remained to be explored. As such, the Data Flow diagram was explored as an easy visual representation of our Project Scope document. Using the Communications and Collaboration diagrams, the team further understood how defined entities, such as the User's and Rank's objects would interact. We solidified this understanding, as the team investigated the MVC of the product, using the Project Requirement document as a reference. The user interacts with the view (composed of languages and rankings), and ultimately the controller (submit ranking buttons, navigation header), which manipulates the model (composed of Users', Ranks', Averages' entities), essentially updating the view. With both the objectives and their interactions/relations clarified, team Guinan decided to explore the Low-Fidelity Prototyping option due to the ease of alterations to design and time management.

Feedback

The feedback that we received through activity four was overall very positive. Comments that stuck out were all about how useful the project could be to new developers. They enjoyed the out-of-the-box thinking and how it was not just another tier list where you can only view the average rankings from any user. Other comments added other features that we could add in the future such as adding a bar chart to compare the rankings of languages. In the time constraints of the class we have not implemented these features but if we had more time these features would have been a very beneficial tool for our website. One neutral comment was that they would have liked to see a High Fidelity design mock up, to more accurately depict our website. When we got this feedback we had already started the development of the website and it was at a place where a Hi-Fi was no longer necessary. Thus, we did not act on this feedback for this project but it would be something that we will carry into future projects.

Technical Breakdown

Development Process

Our team used an agile frame with weekly scrum meetings and a Kanban board. Each week we would assign tasks to each member which would be completed before the next scrum meeting. This allowed us to build on past tasks and stay on top of where the other teammates were with their tasks and their thoughts about how we should move forward. For the development tasks, we used minimum viable products (MVPs). For our first MVP we really wanted our frontend to communicate with our backend. The stylings were not slotted for this week but they ended up getting done rather quickly and we really like how our site is looking at the moment. For the second MVP we were more focused on improving the user experience and security. In this we added a navigation header, popularity graphs and we added Passport encryption to keep our users secure. Near the end we started to plan our MVP 3 but we did not have the time to start on these tasks.

Development languages

For the development of the website we decided to use the same tools as we did in the labs so for the frontend we use Express.js (EJS), for the backend we used MongoDb and Node.js. These tools were used for their familiarity to the team which allowed all members to develop if needed. Tools such as react were considered, but to keep everything accessible to our teammates we chose not to use it. Currently we are running our website locally as the time constraints did not allow for a hosted database and hosting the website would not be beneficial if it did not have a database to run.

User Interactions

When a user visits our site they are greeted with a login page where they can register, login or view our public view of the rankings and information pages. The public page is only for viewing the average rankings and information pages.

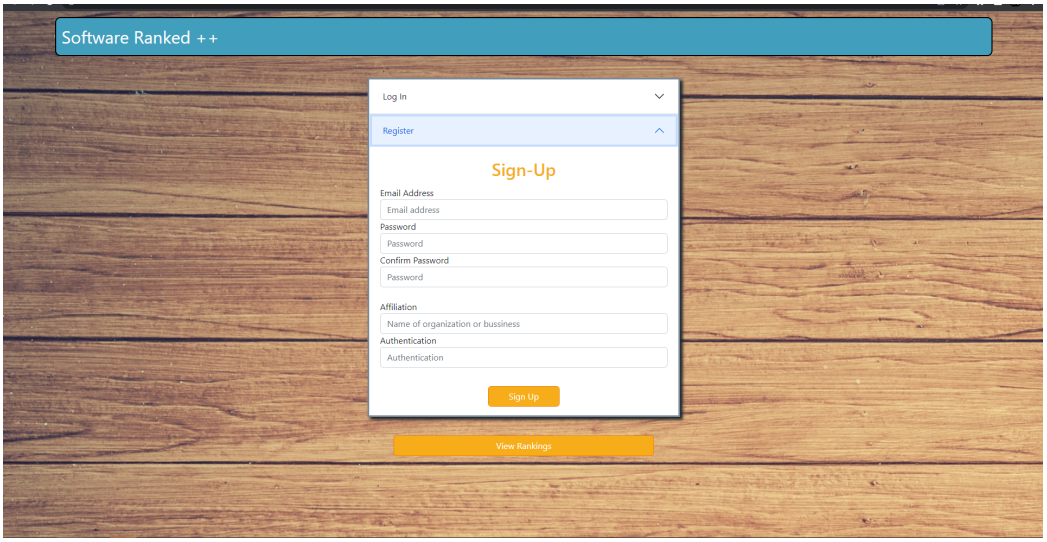

Figure 1.1: Login/Signup Page

Figure 1.2: Viewing user view

When a user signs up and logs in they are greeted with the ranking cards where they are able to rank languages or click into the cards to view the information pages. Users can log out or navigate around pages using the header to further increase the usability of the product.

Figure 1.3: Logged in user view

The information pages show some general information, benefits and syntax for each language as well as a popularity graph. This gives future developers a good idea of what the language can be used for and the syntax can help them get started with a basic hello world. Having the popularity graph also allows both new developers and companies to see how languages are doing, and whether it is worth it to learn or use, or if they should be looking into other options.

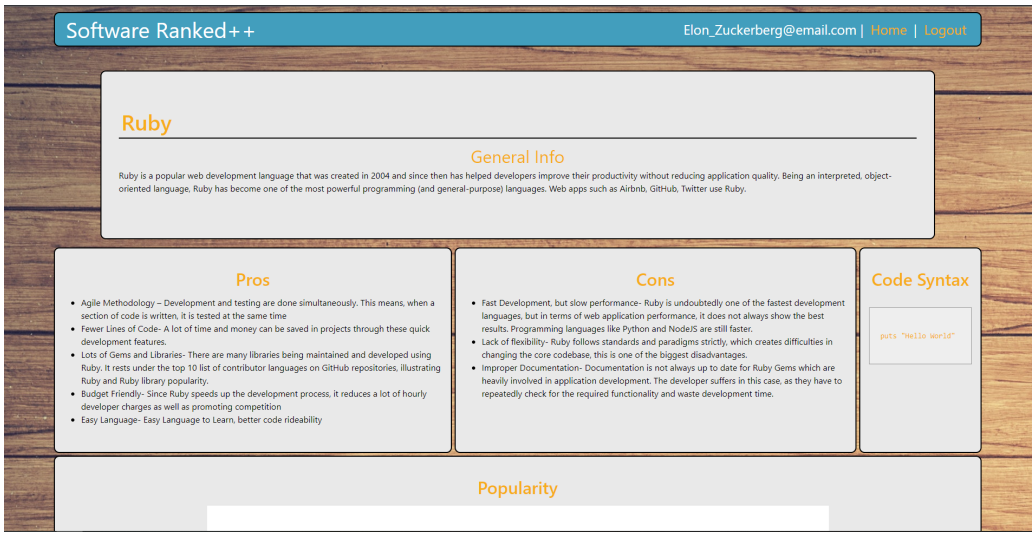

Figure 1.4: Detailed page view

Conclusion

Goals Achieved

The Software Ranked++ was designed with the intention to help new programmers to see which software languages are popular that can help them for the startup to create a new website. Over the course of the project timeline, our website was planned, developed and implemented as an actual functioning website that can solve real world problems. The development of this project was an achievement for our team as the target goals were accomplished. The feedback we received from our peers was very helpful to finalize our project.

Now exploring our MVPs, we managed to complete our first MVP which was our goal and then moved on to our second MVP. We were testing our code manually as we went along and proved to be fully functioning. We did our best to implement all user stories in our MVPs. We created a login/signup page for users to create an account and our homepage includes ranking cards which link to detailed information pages of different software language and rankings, that users can see and could rank them which directly uploads to our database. Later we also implemented the option for passport encryption for the authorized users. Overall our final website reflected the MVPs and had enough core features for a fully functioning website. Software Ranked++ has many fundamental interactions of both types of users with a different interface for each. It allows users to gain knowledge on different software languages and would allow them to rank as per their preferable choice. It would help users to gain credibility in business aspects too.

Reflection

As a team, we really enjoyed this project as it summed up everything we learned throughout the semester. At the end of this project we feel that we were successful as a team because we had a fully functional MVP 2 which was more than our initial goal of finishing the first MVP. As this was not really expected we feel that our project was a success.

The part of the project we felt most proud of was the design of the site. When we initially talked about doing MVP 1 we did not prioritize the visual aspects of the site. We thought we would have a basic form with number inputs and subpar styling. In the end we had cards for each language that not only functioned properly but looked great.

Shahzil: I have learned the important lesson of understanding when something is good enough and not requiring more detail. This reminds me of the lecture where Dr. Tim gave an example of over detailing his paintings. As Voltaire says, “Perfect is the enemy of good.”

Nathan: Throughout the project I learned the importance of communication. Near the beginning we had some issues with communication and it made it quite difficult to get the activities done to the standard that I was hoping for but later our communication increased and the project went more smoothly.

Vida: At the end of the project, what matters the most is what I have learned from our group project. During the tenure of working together, situations cropped up and tested on the scales of making decisions and handling the pressure and expectations from our peers was the most challenging thing. Communication was the key for the success of this project.

Team: The project has taught us a lot about working as a team, providing deliveries with deadlines, how to document for future projects and most importantly being professional and responsible. As a team, we planned to work together and divide our tasks and duties to ensure we meet our deadlines. During those deadlines we did encounter some difficulty in communicating with one of our team member due to covid issues that lead us to some inconvenience but despite that we did overcome those problems and were supportive enough to understand that, “sometimes life just happens.”

Some help that could have been helpful would be more guidance on the documentation file. They are quite open ended and this leaves a lot for interpretation which for developers that have not worked on a full app before can be quite daunting.

Overall, the project helped us understand the team collaborations and possible obstacles that we might face in the future. We all worked tirelessly to ensure that we got a positive outcome. As a team we learned that

we communicated well through Zoom, whether it was meetings that took a few hours, or a quick text to update each other about the deliveries or the deadlines. It was a fun learning experience while collaborating new ideas and developing software technologies that can help solve real problems. Brainstorming new ideas, and working effectively as a team would definitely help us as future software engineers. The versatility of team Guinan, each member being able to handle a variety of tasks, had led us to strive towards success, and not perfection. We consider the successful completion of this project as a significant stepping stone toward our careers, and we believe the journey through and lessons learned would be a great point of reference for future group projects.

Resources

Advantages and disadvantages of C programming language. (n.d.). Retrieved December 2, 2022, from https://www.includehelp.com/c/advantages-and-disadvantages-of-c-programming-language.aspx

Bhagat, V. (2022, November 2). *Pros and cons of python programming language*. PixelCrayons. Retrieved December 2, 2022, from https://www.pixelcrayons.com/blog/python-pros-and-cons/

Deed, I. (2022, August 10). *Pros and cons of JavaScript development*. 2022 | Hire flutter, mobile & JS software devs | Pangea.ai. Retrieved December 2, 2022, from https://www.pangea.ai/dev-javascript-resources/best-practices/

Downey, A. (2016). *Think python*. O'Reilly Media.

Editor. (2020, February 27). *The good and the bad of Swift programming language*. AltexSoft. Retrieved December 2, 2022, from https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-swift-programming-language/

Horstmann, C. S. (2023). *Core java for the impatient*. Addison-Wesley.

Programming Ruby. Programming ruby: The pragmatic programmer's guide. (n.d.). Retrieved December 2, 2022, from https://ruby-doc.com/docs/ProgrammingRuby/

Srivastava, V. (2022, June 8). *Pros and cons of ruby development*. 2022 | Hire flutter, mobile & JS software devs | Pangea.ai. Retrieved December 2, 2022, from https://www.pangea.ai/dev-ruby-development-resources/best-practices/

Team, D. F. (2022, April 18). *Pros and cons of java: Advantages and disadvantages of Java*. DataFlair. Retrieved December 2, 2022, from https://data-flair.training/blogs/pros-and-cons-of-java/

Wikimedia Foundation. (2022, December 2). *C (programming language)*. Wikipedia. Retrieved December 2, 2022, from https://en.wikipedia.org/wiki/C_(programming_language)

Wikimedia Foundation. (2022, November 24). *Swift (programming language)*. Wikipedia. Retrieved December 2, 2022, from https://en.wikipedia.org/wiki/Swift_(programming_language)

Wikimedia Foundation. (2022, November 27). *Python syntax and semantics*. Wikipedia. Retrieved December 2, 2022, from https://en.wikipedia.org/wiki/Python_syntax_and_semantics