End-to-End Web Development in a Startup Environment

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Zachary Wang

Spring, 2024

Technical Project Team Members
Zachary Wang

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Brianna Morrison, Department of Computer Science

End-to-End Web Development in a Startup Environment

CS4991 Capstone Report, 2024

Zach Wang
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
zjw5qs@virginia.edu

ABSTRACT

Over the summer of 2022, I served as a software engineering intern for the company QuoteWell, formerly Capacity Post, an insurance technology startup whose goal was to enable insurance agents to bind insurance more efficiently for their small-to-medium business clients. To this end, I designed and implemented an end-to-end feature internal tracking of customer submissions on the product website. I used code, no-code services, and third-party APIs to create the feature. This project lasted over a month and was a core aspect of the company's product and workflow. As an essential component to customer service, this project was used by the company until their goals and product vision changed. As the company grows, the need for and configuration of this project will be reevaluated as the customer needs evolve.

1. INTRODUCTION

Working in a startup means building from the ground up. Products are designed and created as needed and improved as necessary. To start, a startup's product will be barebones, only providing the essential features to users. However, this model is not sustainable; as a company grows, so must its product. In the case of QuoteWell, that meant improving the product internally so that the customer could receive value faster.

In the insurance world, businesses hire insurance agents to find suitable insurance for the business's needs. An agent typically has to collect information from the business, then send that information to multiple insurance carriers to find the right match. This is done through portals on insurance carrier websites, or sometimes through filling out PDFs to send to the insurance carriers. This is very tedious work.

QuoteWell provided a solution to the redundant filling of forms by aggregating carrier portals and PDFs into a single form, enabling the insurance agent to fill out the business information only once. An insurance agent fills out the form, then QuoteWell sends it out to the carriers on behalf of the agent.

Previously, agent submissions had to be processed and handled manually. Therefore, a system for tracking the status of the submission for both internal use and for the external(?) user was needed. This project was essential because it would allow QuoteWell's main product offering to be more organized and efficient, reducing workloads for the company and saving more time for the customer.

2. RELATED WORKS

The main inspiration for the design and high-level functionality of the project came from personal correspondence with my direct supervisor. Aleshin had a vision for the system in mind, which outlined what the user experience would be, and how the internal use of the project would function (T. Alesin, personal communication, June 2022). Because Aleshin was not a developer, he did not have specific implementation details laid out.

NestJS documentation, which goes into detail on how the NestJS framework can be used to create REST APIs, was essential for determining how to implement this project. I used the relevant features of the framework to develop the project, which amounted to only a small portion of NestJS's functionality (*Documentation: Nestjs - a progressive node.js framework*).

3. PROJECT DESIGN

My project consisted of 3 stages. First, is the requirements stage, where the goal of the project is defined. This involves determining what the product needs, and what value a project will bring to the customer. The second phase is design. This phase details the specific implementation strategies, including the technologies to be used and how they will come together to form the project. Lastly, the implementation itself where the code is written, applications are connected, and the project is deployed.

3.1 Requirements

The first phase of a new software development project is understanding and analyzing the requirements that the software needs to fulfill to bring value to the customer. This process helps determine if a project is worth completing. Additionally, it helps to outline and steer the design of the project further down the line and ensure that it aligns well with overall business goals.

In the case of my project at QuoteWell, users needed a way to view the status of their submission in real time, because these submissions were essential to their business operations. Keeping the user informed about the status of their submission was very important to the company as well, because transparency of the internal process was a key tenet of company goals. We determined that we would develop software for our existing website to provide this functionality to customers.

In addition to providing status tracking for the user, we needed a system to keep track of the submission internally. This system would need to help the team manage multiple submissions, as well as keep the team informed about who is working on what submissions and the stage of the process the submissions were currently in.

3.2 Design

The second phase of software development is designing the implementation of a system. This involves determining what technologies will be used, what coding strategies, how interfaces will look, and so on. Planning such details is essential beforehand so that when the time comes for actual implementation, it can be done smoothly.

We determined that the internal tracking process would start as soon as the user clicked the "submit" button on their application. After some deliberation, we also decided to use Airtable as the internal tracking tool. Airtable is similar to Excel with rows and columns, but with many robust customization features that make it suitable for a database accessible to non-technical team members. Doing so would allow the team to update the status of a submission without having to work with a backend database.

The system flow would start with the user clicking "submit" on their completed application. The website would show a status of "Received" to the user. This would update an Airtable base with information about the new submission. The team would internally handle the submission, updating the status in the Airtable as progress was made. Once the status of the submission in Airtable reached "Complete," the status of the submission would show as "Complete" on the website for the user, as well. Slack notifications would be automatically sent to company channels as certain milestones were reached.

Beyond the high-level functionality defined above, specific implementation details need to be planned, as well. Airtable provides an API that can be used to create new data in a base with code. A service called Zapier was to be used to connect Airtable to Slack. Zapier could also be used to connect the Airtable to a custom API for the website, which would be used to update the status of the submission for the user.

3.3 Implementation

The next stage of the project was the actual building and implementation. The first step was to lay the groundwork that was necessary to create the rest of the project.

The functionality of the submission button already existed. The Airtable base needed to be created and configured to our specifications and needs. Usage of the Airtable API would be based on the configuration of the base. Once the base satisfied our needs, I was ready to begin coding.

The first step that I implemented through code was the retrieval of relevant submission data from the company Postgres database at the time of user submission so that a new Airtable record could be created. This involved using NestJS to query Postgres. I then used that queried data to populate a new Airtable record, which also included a "status" field that could be easily managed by the team.

The next step was to implement the Slack and Airtable integrations using Zapier. When a new record is created in Airtable, due to a new user submission a Slack notification was sent to a designated channel. Zapier can "watch" an Airtable base and send a Slack message when a new record is added. Similarly, Zapier can watch for updates to existing records, which was used to send a Slack notification whenever a submissions status changed in the Airtable base.

Finally, the status of the submission needed to be updated for the user in real time. By default, when "submit" was pressed, the status would read as "Received." Another Zapier was set to watch submission records in Airtable for status updates to "Complete." Zapier can also be configured to hit custom API webhooks. Doing so allowed the status on the website to be updated in real time to "Complete."

4. RESULTS

Implementation of this project was not a simple task. I did not have experience with any of the required technologies previously, so mentorship and research were essential. Despite initial challenges, a successful project that met requirements was carried out.

Because of the nature of the project and the company's goals, the features implemented were essential to business operations. My work provided a basis for the system that would continue to be built upon after my time as an intern was over. Although specific numbers are not available, my system was used to provide service to at least two official company customers. Because the company was a very early-stage startup, they only had two or three customers at the time. However, as the company developed and business goals eventually shifted and evolved about a year later, my system's functionality was no longer needed.

5. CONCLUSION

My Project at QuoteWell was a fantastic learning experience. I was exposed to many new technologies that are used throughout the software engineering world that will be helpful in my future career. These include Airtable, NestJS, Postgres, Zapier, and React. Connecting these technologies in a novel way was an interesting challenge. I also learned about the development process in the real world.

Beyond learning, this project was fulfilling because of its impact on the company. It was an essential business function and was used frequently. It helped streamline the process of submission tracking internally, which led to faster turnaround times for the customer. Furthermore, working with a very knowledgeable team and creating a project that fulfilled their requirements and impressed them helped grow my confidence as a software engineer.

6. FUTURE WORK

My project could be improved in many ways. Additional features would be helpful, such as more robust and useful notifications sent to other services, not limited to Slack. Furthermore, refactoring the Airtable to be more user-friendly would help increase productivity even more. The incorporation of usage reporting and analytics of new submission would provide more useful data for the team that could be used to further improve customer experience.

Additional features are useful, but not the only aspect of improving technology. Scalability and performance optimization are very important factors to the longevity of a project. If a feature is to be used as a company grows, we must ensure that the feature can handle larger and more frequent usage. Another strategy is to solicit feedback from both customers and users of my project so that I know what areas have room for improvement.

REFERENCES

Documentation: Nestjs - a progressive node.js framework. NestJS. (n.d.). https://docs.nestjs.com/

T. Aleshin (personal communication, June 2022)