

1. Introduction (2-3 paragraphs):

The opening section describes the document's purpose and a preview of what it contains. You will also provide an overview of the primary technical tasks you were assigned by your project partner.

NOTE: Please use Section 2 or 3 of this paper to address topics like technical debt and the restrictions that were placed on you by project partners. (*Technical debt* might be thought of the lower quality technical solutions you or a previous team implemented for a short-term deliverable where a better technical solution would take longer and therefore be more costly.)

In the past few decades, humans have developed rapidly in the computer science field, which has also led to a large number of research and application development of data collection, storage, and analysis technologies, which greatly reduces the cost, time, and time of data processing. An era of "big data" is gradually unfolding before us.

Once, boring data and statistics were only the concern of a few mathematicians, statisticians, and economists. The innovation of science and technology and the revolution of data have made it possible for people to process massive amounts of data, and it has become more and more convenient and effective. This has attracted a large number of people across professional disciplines to invest in the development and application of big data. In this era of background, it will help our team develop innovative data visualization technology.

From a certain perspective, data visualization is the same as machine learning. Both are tools for data analysis, but from a different perspective. Machine learning (ML) uses different models to list the characteristics of the analyzed data (data sets) from the perspective of the model; data visualization is from the perspective of images, such as bar graphs, allowing users to quickly understand the differences between different data (data sets). Its essence is to allow blunt data to talk to human vision. Data visualization uses graphical means to convey and communicate information clearly and effectively.

The ability to efficiently assess the efficacy of a data visualization is at the core of this project. When a data visualization designer creates a visualization, they are not always certain that people are able to understand the data that is being portrayed. Sometimes data visualizations are very hard to comprehend, or lead readers or customers to the wrong conclusions. To get an idea of the efficacy of a data visualization, data visualization designers like to receive feedback on their designs. One of the ways that this can be done is by using a survey taking platform that allows a data visualization designer to pay for survey takers. However, the process for receiving feedback from data visualizations is a manual intensive process, which is in need of automation. Previously, a data visualization designer would have to go through the manual process of creating a survey through Qualtrics, exporting the survey, and uploading it to a service such as Amazon MTurk that would collect feedback from a specified sample of survey takers. The data from the feedback

would then need to be analyzed to determine how effective the data visualization was in conveying information. A survey would therefore consist of questions related to identifying data points in the data visualization or asking questions that give the designer a good understanding of how many people are able to quickly understand the data being portrayed. This entire process is attainable when there is a small number of data visualizations that need to be processed, but fails to scale for a large number of data visualizations.

In order to automate the feedback collection process for data visualizations, an automated process needs to be created. The goal of this project is to create a website where designers can go to upload their data visualizations with their surveys, which are sent automatically to Amazon MTurk, the survey platform. The data visualization designer would be able to specify how many survey responses they would like, as well as other criteria such as age or demographics. Once the surveys have been completed, the data visualization designer would get a notification and could then view the results of the survey. The data visualization designer would also be able to check on the progress of the surveys, indicating a need to interface with Amazon MTurk to retrieve the current progress every time a user would like to check. Finally, the data visualization designer would be able to run a set of statistical tests on the returned data to get an understanding and interpretation of the survey results. From here, the data visualization designer could make adjustments to their creation and make modifications to make a more effective visualization. For the time being, the combination of a website and servers used for processing survey results would be the extent of the capstone project and would be provided as a service. The project is likely to start internally for OSU as a service available to students or professors, but has the ability to expand as a service available to the public as well.

Link of example dataset: <https://www.unb.ca/cic/datasets/ids-2017.html>

Dataset description:

Intrusion Detection Evaluation Dataset (CIC-IDS2017). It is provided by Canadian Institute for Cybersecurity.

This dataset contains benign and the most up-to-date information of common network attacks, which are similar to the true real-world data (PCAPs). What's more, the result of network traffic analysis, source and destination IPs, source and destination ports, protocols and CSV files are also included.

2. Tools and Technical Approaches (about one paragraph per tool or technical approach):

In this section, you will want to review the tools and technical approaches you implemented to address the technical tasks mentioned in the Introduction. What tools did you use? What is your satisfaction with the tools? What other choices did you wish you had

made or explored? How would they have helped or hindered progress? Would you choose the same tools and technical approaches again? If not, what would you do instead? Testing: What sort of testing framework are you using? (etc.)

Tools and Frameworks

A. Auth0

Auth0 was the third party service we used for user accounts. Auth0 provides a login, sign up and password reset page all out of box which was largely beneficial in speeding up the development process. Overall we have been very satisfied with the service, as it provides an easy way to also store metadata with each user that we use to store survey specific information, thus avoiding the need to store this information in the database. If we were to look at the project again, we would probably look more closely at Firebase, however we felt that it made more sense to use Auth0 as it provided a few more UI components that were useful. Additionally, we would need a google account for Firebase which makes it harder to maintain for the project partner.

B. .NET Core

.Net Core and C# is part of our application stack for the site. It is a framework maintained by Microsoft that is cross platform and allows for the deployment of both console and web applications. The project partner had a requirement at the beginning of the project that the website should be written in a stack that they were familiar with and .NET was one of them. .NET Core has advantages such as its cross platform capabilities which is a huge advantage for this project. Considering that the final application will be deployed to school managed servers, there is a possibility that the school changes the types of servers that they make available. Therefore, it is useful to have an application that can run on Linux or Windows and would only require a changing of the deployment specification. For this application, we also created a monolith, with the backend and frontend being deployed in the same application. Although there is a rather strong push in industry to move away from monoliths of this type, the lack of direct access to the production servers makes it beneficial to reduce the complexity of the deployment configuration and the number of moving parts.

C. Qualtrics API

The Qualtrics API is what is used to create the surveys behind the scenes. The Qualtrics API has documentation that is not very user friendly and can sometimes be

confusing. Unfortunately, there are not many platforms that can support the expansive features that Qualtrics supports, and therefore is very essential to the project in its current capacity. For this project in particular, Qualtrics is used to present a random visualization from the set of visualizations that the designer submitted to the survey taker. This allows the data visualization designer the ability to see which design is the most effective by asking each survey taker the same questions but alternating the design shown to them .

D. Github

Github is used to store the codebase and keep track of changes across multiple branches and team members. To add a new feature to the main branch, a group member will first need to create a separate branch and commit changes on that branch. Then, the group member can create a pull request from that branch into the main branch. We created a CODEOWNERS file that requires that a member of the group other than the pull requester review and approve the code that is being brought into the main branch. Although we are not able to practice all aspects of extreme programming, having other group members review code changes will be a way for the entire team to stay up to date on code changes. The repository also does not allow anyone to push changes directly to the main branch, which ensures that we do not have any accidental pushes into our repository. Once a pull request has been approved, the pull requester can then go ahead and merge the pull request using the git squash command. The git squash command will help keep the main branch clean of unnecessary commits and will be useful for tracking all of the commits related to a feature. Once a pull request is made, the team is notified via discord that a pull request is open. In order to update the discord server for the notifications, a new web hook integration with Github would need to be set up in the new discord server.

E. Bootstrap 4 and JQuery

Our project made use of Bootstrap for styling the frontend and JQuery for simplifying the use of javascript on the frontend. Bootstrap comes with a flexbox based grid system and CSS classes that are used for styling in our HTML. Additionally, JQuery was used to implement dynamic actions on certain pages and communicate to the backend using AJAX without needing to refresh the page. Overall both of the libraries were easy to use and helped reduced the time spent on styling simple components of the application. The project partner also had a hand in picking which bootstrap theme we would be using in our website. Specifically Andrea Marks, who is a design professor here at Oregon State, has been very helpful in choosing a theme that has a “look and feel” which matches our project as a whole.

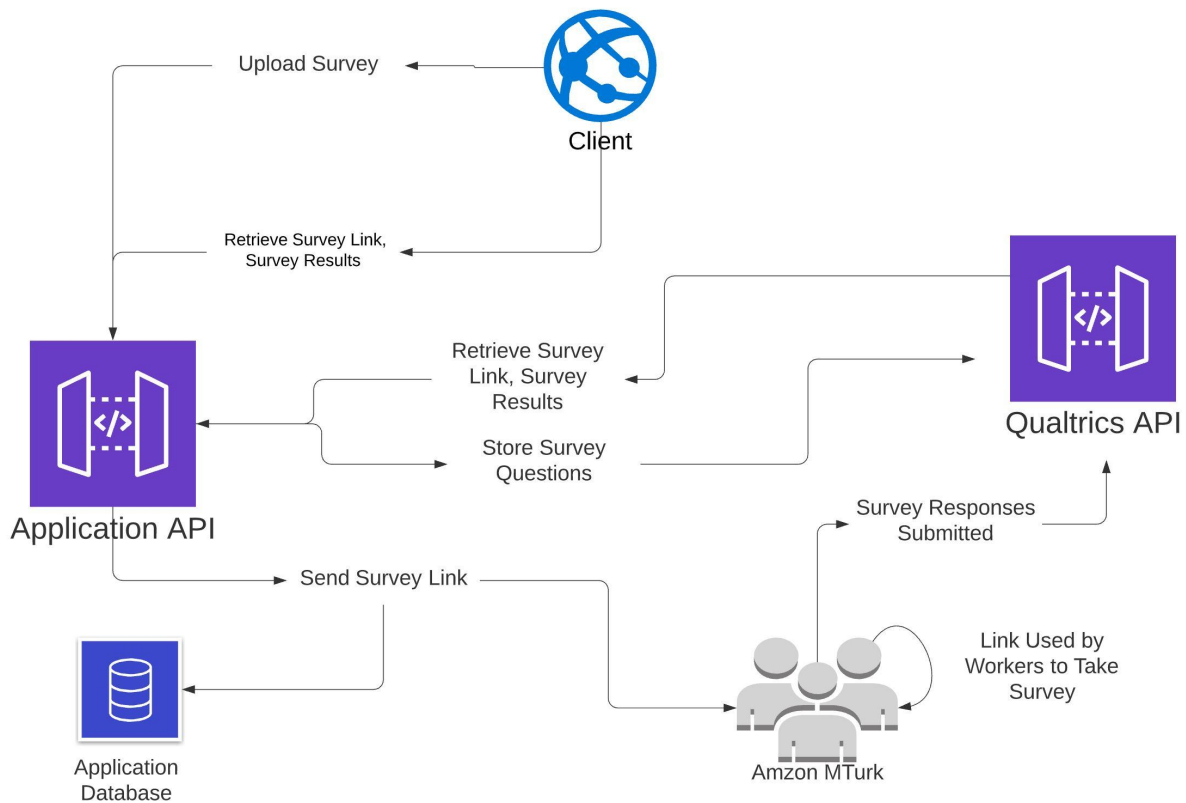


Figure 1: High Level Backend Overview

A. Collecting Survey Data

In the above diagram, the entire process for the project is described, including stretch goals. The process begins in the client, in this case the website. The user submits a survey on the website with their questions, possible answers, correct answers and data visualizations. Once the survey is submitted, the application API receives the survey and communicates with Qualtrics via the Qualtrics API. To generate a Qualtrics survey using the Qualtrics API a survey block needs to be created. Each question will reside inside of each survey block, along with the possible answers to the questions which will be done in each subsequent request to the Qualtrics API. Once the survey is created in Qualtrics, the Qualtrics API will send back a survey link which the application API will provide to the client.

B. Mapping Users to Surveys

The application API will send the survey link to the database where a user map to the survey link will be created. This will involve a database with a table of users and a many-to-many relationship with a table of surveys. Since we cannot assume that survey links will be unique, we will need to create the join table between the survey entity and the user entity which will keep track of the survey link of the survey that the user is associated with. Users will be identified by a unique string that is returned by Auth0, allowing the application to identify who is logged in and which surveys belong to them. The reason that this added flexibility is needed is for the future uses of the application, that may need multiple survey links that all point to the same survey.

C. Saving Survey Responses

At this point, the application API sends the survey link to a platform such as Amazon MTurk where workers can take the survey using the link. The survey responses will save to Qualtrics from the survey link. Once the application API detects that the survey responses have been recorded, the application API will allow the user to download the survey results as a CSV from the website. The detection process will likely involve polling for updates, unless either the survey crowdsourcing platform allows a webhook or call back or Qualtrics provides a programmatic mechanism to determine when a sufficient number of responses have been recorded.

D. Database Considerations

Since we need the ability to determine which users submitted which surveys, a database mapping between users and a unique survey key is needed. This database process will also allow the application API to query the Qualtrics API with the survey key to retrieve results for the survey. This would allow us to make survey results available for download from the website. As seen in Figure 2, there is a user table with a single column of id. This id column serves as the identification column, and is not auto incremented. When a user creates a survey, the user's GUID will be retrieved from Auth0, and compared to the existing GUID's in the system. If the GUID is not found in the database, then the GUID will be added. If the GUID is found then we move to the UserSurveys table which is the join table between the survey entity and the user entity. The UserSurveys table will have an auto generated primary key, and will have a foreign key from the User table and the Survey table. Finally, the last column will contain the survey link for the survey. The Survey table will contain a unique identifier for the survey that will be retrieved from Qualtrics when the survey is originally created.

3. Project Management (2-to-3 paragraphs):

This section is an autopsy of the group's management. How did you divide up the responsibilities for completing all the tasks of the project? Knowing what you know now how would you approach the project and tasks differently? When were members of the team idle and not working? Where was time wasted? How would you assign sprint tasks differently if you were to start over? What problems have you had with integration, and how did you overcome them? Were there any logistical problems? What kind of planning would have made things go more smoothly? Where did you go to acquire necessary knowledge or resources?

Responsibilities were split up by focus area. We divided the team into backend and frontend focused, which helped with deciding who should get which task. Each week met on Wednesdays to partake in our sprint meetings. During the start of our sprint we hosted a sprint planning meeting which involves creating user stories and adding them to the

backlog. Then, we chose a certain number of items that we could complete by the next sprint. During each subsequent week we had a check-in meeting to discuss sprint progress and alleviate any roadblocks. During the final week of the spring we had a sprint retrospective to reflect on how the sprint went.

A scrum master was selected for each sprint, during the final sprint meeting of the previous sprint. The scrum master was responsible for conducting the sprint ceremonies and keeping documentation for the sprint. We used a Trello board to track our sprint progress, and the information in our sprint backlog was stored there. The scrum master was responsible for updating the sprint documentation in the code repository with the different user stories that were completed as well as who completed them. In Trello, each member of the team was able to assign themselves to a card and update the status of the card at any time, which helps the scrum master identify which user stories need additional attention.

4. Advice and Recommendations (about one paragraph per recommendation):

What recommendations would you have offered your project partner at the beginning of the project had you been able to do this over? What advice would you give to a team who would be inheriting your project?

We need to continue to maintain an optimistic and respectful working relationship. Under ideal conditions, this can ensure that we have a good attitude of active cooperation. Everyone is good at different programming languages, which leads to the way in which everyone's thinking is processed, so we all need to study and understand patiently, so that we can integrate everyone's code together and make an excellent project. What's more, we will continue to cooperate in the spring term, continue to maintain weekly meetings, and code sharing. I hope everyone in the team will have good luck in their study and life in the next term.