

# Technical Feasibility

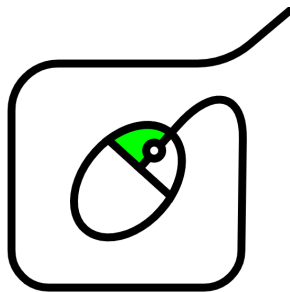
## Version 1.0

3 November 2021

SuperGeeks

CS 476  
Requirements Engineering

---



**Team Members:** Tim Giroux, Kyle Austria, Gustavo Valencia, Pengfei Liu

**Clients:** Dr. Marco Gerosa, Dr. Igor Steinmacher

**Mentor:** Felicity Escarzaga

# TABLE OF CONTENTS:

<b>1. Introduction</b>	<b>2</b>
<b>2. Technical Challenges</b>	<b>4</b>
<b>2.1 Technical Analysis: Web Hosting</b>	<b>5</b>
<b>2.2 Technical Analysis: Survey Platform</b>	<b>10</b>
<b>2.3 Technical Analysis: Web Framework</b>	<b>15</b>
<b>2.4 Technical Analysis: Database</b>	<b>20</b>
<b>3. Technical Integration</b>	<b>24</b>
<b>4. Conclusion</b>	<b>26</b>
<b>5. References</b>	<b>27</b>

# 1. Introduction

Software engineering research is critical to the success of many industries. It provides insights that are highly valuable to users and developers alike. Empirical data about what does and does not work in software engineering is very important in the world today, especially for critical software systems in the medical and structural engineering industries. Research in this field often depends on human participants. Researchers need to coordinate with software professionals in order to track the efficacy of various engineering strategies. Without this coordination, much of this critical research would be too challenging to conduct.

While researcher-participant coordination does take place in software engineering research today, it is difficult to facilitate and limited in scope. Researchers face a number of core problematic business functions:

- *Limited scale:*  
Large-scale research projects could result in important discoveries but are often discarded because there is so much difficulty in obtaining participants.
- *Inefficient recruitment:*  
Recruitment efforts are ineffective and create a large amount of overhead for the researchers.
- *Sampling bias:*  
Typically, only a subset of desired participants for a study actually respond to recruitment efforts. This introduces bias because the other desired participants will not be included in the final results of the study.

The current solutions to these problems take much more time and effort than should be necessary. Researchers might send out mass emails to software professionals based on their Github profiles, but this kind of email is often considered as spam by the developers. Developers can sometimes be irritated by an email like this, or they may simply ignore it. Of those who respond, many do so only to politely decline.

This is why the project sponsors, Dr. Gerosa and Dr. Steinmacher of the SICCS Software Engineering Research Laboratory, came up with the idea for GeekSurvey: a simple web platform to bridge the gap between researchers and participants. The clients produce a large volume of software engineering research each year, and their primary means of obtaining participants is mass email efforts. They report that this is a prominent bottleneck in their research process. They have also reported that the same problem impacts many of their colleagues. After trying existing products like Prolific and Amazon Mechanical Turk, they found that these products lack the ability to easily filter participants based on their professional

qualifications. The clients need a product that will appeal to the needs of both software professionals and researchers.

GeekSurvey aims to be a product that could solve these problems. The plan for GeekSurvey is a public web application where users can create an account and either create or participate in research studies. To facilitate cooperation between researchers and willing participants, GeekSurvey will need the following core features:

- Two account types: researchers and participants.
- Participant accounts will include information about the user such as their occupation, level of education, experience with certain tools, and a link to their Github account.
- Researchers have the ability to create surveys and specify necessary criteria for participation.
- Participants are given access to surveys which they are eligible to participate in.

## 2. Technical Challenges

Before creating this kind of web application, there are some technological challenges that may hinder the development of the project. The four main challenges are as follows:

- *Web hosting:*

Users must be able to reliably access GeekSurvey through an internet connection. This can be accomplished by placing the GeekSurvey project on a web server. There are numerous ways to acquire access to a web server. Hardware capabilities, monetary cost, and developer experience can vary. The challenge is to find an optimal web hosting solution for this project.

- *Survey platform:*

GeekSurvey will need to integrate with an external survey platform. Survey platforms allow researchers and participants to cooperate on high quality research. GeekSurvey will provide support for researchers to add links to surveys that they create on an external platform. The challenge is to choose a platform that will meet the needs of GeekSurvey users and developers.

- *Web framework:*

Web frameworks provide developers an efficient way to develop quality web applications using recommended tools or development strategies. In addition, frameworks can be conducive to easy debugging and software maintenance. This is a challenge for GeekSurvey as choosing a quality web framework to guide development will be critical to the quality of the project. A good web framework should be applied to GeekSurvey to support a high quality product.

- *Database:*

Databases allow developers to manage the storage of important data. GeekSurvey will need to integrate a database to store user accounts and survey tracking information. This is a challenge for GeekSurvey as the web application will need to integrate with a secure and quality database that can work effectively with other project dependencies.

Three candidate solutions have been selected for further analysis for each of the challenges listed above. For example, Northern Arizona University Information Technology Services (NAUITS), Amazon Web Services (AWS), and DigitalOcean are the candidates for web hosting. Since there are different candidates for each challenge, metrics are needed to compare and evaluate these solutions. Each challenge will have its own metrics to define the optimal solution, but some challenges will have similar metrics. For example, both survey platforms and web frameworks have flexibility and documentation as metrics.

## 2.1 Technical Analysis: Web Hosting

GeekSurvey is a web application, so it needs to run on a server. Developers will need to be able to upload application changes to some computer which will be responsible for running the GeekSurvey website. Users will need to be able to access GeekSurvey through an internet connection. The term **web host** refers to any method through which these necessities can be accomplished. An adequate web hosting solution will need a Linux server operating system, a static IP address, internet connection, around 10 Gigabytes of disk space for database and web file storage, and around 1 Gigabyte of RAM for handling connections. This server will need to allow for direct development with root privileges. This server will be responsible for handling HTTP requests and returning HTTP responses. It will also be running the database and the web framework for the application. The web server will need to be able to serve web pages quickly to multiple clients at any time.

The candidate solutions for web hosting are Northern Arizona University Information Technology Services (NAUITS), Amazon Web Services (AWS), and DigitalOcean. Each option provides a web hosting service that can be used to host a dynamic web application. They differ in hardware capabilities and developer experience. The metrics that will be used to compare and evaluate web hosting candidates are as follows:

- *Scalability:*

A scalable web hosting solution is one which can be easily upgraded in the future to handle a larger number of users. 5/5 in scalability would be a solution that is the simplest to upgrade, and a 0/5 would require the most developer hours.

- *Transferability:*

A goal for web hosting is to hand off a working product to the GeekSurvey clients after initial development has been completed in May of 2022. A web hosting solution with 5/5 transferability is one that can be easily and fully transferred between clients and developers. A 0/5 indicates that transfer of ownership would involve unnecessary difficulties or complications. It would be preferable if developers and clients do not have to share access to web hosting accounts with payment information, as this can complicate the process of transferring the project.

- *Cost:*

This project is limited in budget and development hours. Web hosting solutions with lower monetary and development costs will be prioritized. Ideally, a web hosting solution is both free and simple to use. A 5/5 reflects a favorable or low cost, and a 0/5 reflects an unfavorable or high cost that would hinder development.

***NAU Information Technology Services:***

The first web hosting candidate is NAU Information Technology Services or NAUITS. NAUITS is a department at the educational institution known as NAU or Northern Arizona University. GeekSurvey is a project sponsored by NAU associate professors and developed by NAU students. NAU has a sophisticated information technology department because it needs to maintain a large computer network for thousands of students, faculty, and staff [1]. This could be a great solution because GeekSurvey source code would be easily accessible and archivable for the project sponsors. If the project is hosted through NAUITS, it would have to make use of containerization. Containerization can be described as a way to package a software product with all of its dependencies. NAUITS provides hosting using a technology called Podman [2]. Podman is a tool for creating and managing containers.

- **Scalability - 2/5**

NAUITS has limited resources, so it would be troublesome to scale GeekSurvey within their systems. A benefit for scalability is that the use of containerization would make it simple to deploy the product to other, more scalable platforms when necessary.

- **Transferability - 5/5**

Transferability is the main reason for considering NAUITS. Web hosting through NAU would be managed by authentication using NAU accounts. As of November 2021, all GeekSurvey developers and sponsors have NAU accounts. This allows transfer of ownership to be accomplished easily by an IT administrator.

- **Cost - 4/5**

NAUITS is able to provide web hosting services for free, but using this hosting service would cost some additional developer hours for learning to configure and deploy containers using Podman.

***Amazon Web Services:***

Another possibility for web hosting is Amazon Web Services or AWS. Amazon's commerce website is a technological marvel that relies on complex networking infrastructure. They also offer a product known as Amazon Web Services or AWS, a large suite of cloud computing services. A developer can make an account with AWS and quickly begin using multiple interoperable cloud services. Amazon provides extensive documentation for these services, including specific tutorials for common use cases [3].

## Virtual servers

Linux/Unix Windows

\$3.50 USD/mo	\$5 USD/mo	\$10 USD/mo	\$20 USD/mo	\$40 USD/mo	\$80 USD/mo	\$160 USD/mo
512 MB Memory 1 Core Processor 20 GB SSD Disk 1 TB Transfer*	1 GB Memory 1 Core Processor 40 GB SSD Disk 2 TB Transfer*	2 GB Memory 1 Core Processor 60 GB SSD Disk 3 TB Transfer*	4 GB Memory 2 Core Processor 80 GB SSD Disk 4 TB Transfer*	8 GB Memory 2 Core Processor 160 GB SSD Disk 5 TB Transfer*	16 GB Memory 4 Core Processor 320 GB SSD Disk 6 TB Transfer*	32 GB Memory 8 Core Processor 640 GB SSD Disk 7 TB Transfer*

Figure 2.1.1 - AWS Lightsail Virtual Servers [4]

AWS provides a service called Lightsail that would particularly be of use for GeekSurvey. As shown in Figure 2.1.1, Lightsail provides multiple tiers of unmanaged Virtual Private Server (VPS) instances. A VPS is a computer that runs on cloud hardware. A developer can remotely access a VPS to configure and program on the computer. A Lightsail instance could be used to host GeekSurvey, but it would require monthly payments.

- **Scalability - 4/5**

Scaling GeekSurvey to work for a larger user base would involve increasing the memory, bandwidth, and database capabilities of the web host. This would be possible on AWS by upgrading to a more expensive Lightsail instance.

- **Transferability - 2/5**

It's possible for developers to work on a Lightsail instance through an AWS account using an official project budget that could be negotiated with the clients. This solution is not preferable because it involves complications regarding budgeting and payment processes. It would limit developer privileges within the AWS account, which could slow development. It would also leave the clients with a complex end-product and monthly payments when initial development ceases in May of 2022.

- **Cost - 2/5**

AWS requires an account with payment information because monthly fees can vary depending on services used. The minimum requirement for a Lightsail instance is \$3.50 per month, which is expensive compared to the free NAUITS solution. AWS also requires additional developer hours to learn how to work effectively in the AWS ecosystem.



## DigitalOcean:

The final candidate for GeekSurvey's web hosting is DigitalOcean. DigitalOcean is a cloud infrastructure company that prioritizes individual end-users. DigitalOcean lacks many of the features that are provided by the full AWS suite of products. They are still a viable option because they offer competitive web hosting solutions.

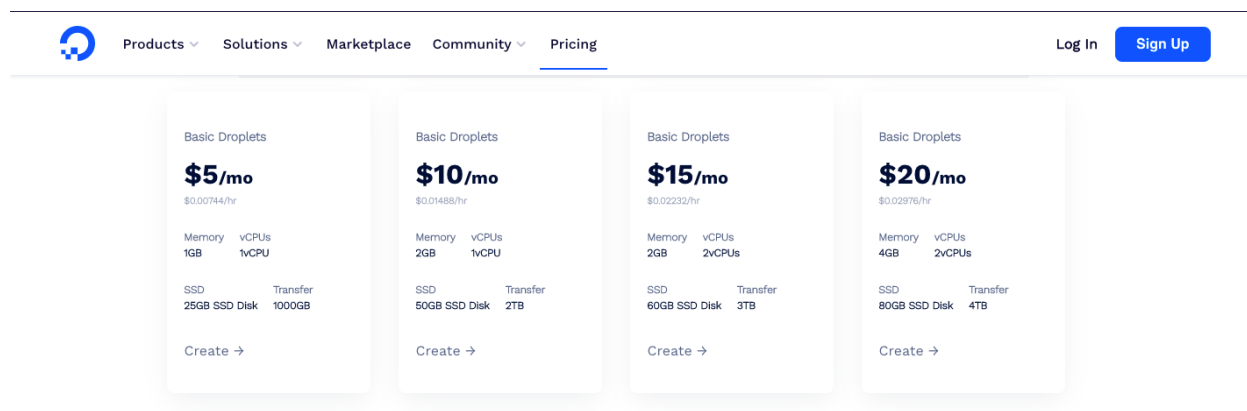


Figure 2.1.2 - DigitalOcean Droplet Pricing [5]

DigitalOcean provides a VPS service that is similar to AWS Lightsail, but differs in pricing, features, and overall developer experience. Virtual Private Servers on DigitalOcean are referred to as Droplets. DigitalOcean offers multiple tiers of Droplets depending on the budget and needs of a project. The \$5 per month Droplet, which can be seen in Figure 2.1.2, offers enough hardware capabilities for GeekSurvey.

- **Scalability - 3/5**

A DigitalOcean Droplet offers the same scalability options as AWS Lightsail, but if GeekSurvey ever requires more cloud services, it would be easier to integrate with AWS because of the many additional features it offers.

- **Transferability - 2/5**

DigitalOcean poses the same issue for transferability as AWS. Transfer of ownership for a Droplet is possible, but it is made complicated because DigitalOcean requires an account with payment information.

- **Cost - 3/5**

The lowest Droplet tier has worse hardware specifications than what AWS can provide for the same price [4], [5]. At \$5 per month, AWS offers an extra 15GB disk space and an extra 1TB transfer compared to DigitalOcean. However, DigitalOcean provides benefits for students. As of November 2021, there is currently a deal available for \$100 credit for DigitalOcean accounts which are linked to a Github student account [6]. This means student developers could operate a Droplet at zero cost during the development

process. In order to keep the product online, the clients would have to pay the full \$5 per month after transfer of ownership in May of 2022.

**Web Hosting Results:**

**TABLE 2.1.1**

	<b>Scalability:</b>	<b>Transferability:</b>	<b>Cost:</b>	<b>TOTAL:</b>
<b>NAUITS</b>	2	5	4	<b>11/15</b>
<b>AWS</b>	4	2	2	<b>8/15</b>
<b>DigitalOcean</b>	3	2	3	<b>8/15</b>

The results of the ranking in Table 2.1.1 put NAUITS as the most promising solution with a score of 11/15. The main benefits of hosting with NAUITS are in transferability and cost. Containerization of GeekSurvey will allow developers to package the entire web application for the clients and any future developers. Once packaged, GeekSurvey can be easily transferred between owners and hosting platforms. NAUITS provides support to NAU students and employees who use their Podman services. The entire process is free and contained within the NAU network.

As of November 2021, an NAUITS web hosting solution for GeekSurvey is already online and serving a simple web page. The current plan for demonstrating the feasibility of this solution is to add functionality to the existing Podman container. This demonstration should show that the web server can be configured to work with a web framework to host a responsive web application.

## 2.2 Technical Analysis: Survey Platform

GeekSurvey will need a way to link with a survey platform based on tokens sent using Universal Resource Locators (web links or URLs). Tokens provide the application the ability to specify user roles and permissions, and what resources they have access to. All required information for authentication can be contained in a token [7]. This method of coordination with survey platforms is an essential function of GeekSurvey. For the purposes of early development, it is necessary to choose a single **survey platform** for GeekSurvey to support integration with.

Survey platforms provide a way for researchers to engage with participants and obtain useful feedback. A feasible survey platform for GeekSurvey allows researchers to develop high quality and secure research without limitations that can hinder survey creation. It gives researchers the ability to export survey data seamlessly for collection and accurate analysis. Such a platform will need to have useful information about its features, functionalities, and integration. It must be possible for GeekSurvey to arrange for seamless permissioned access to surveys. Thus, with a feasible survey platform, researchers will be able to create and distribute surveys through GeekSurvey with ease.

The candidate solutions that can effectively work with GeekSurvey are Google Forms, SurveyMonkey, and Typeform. They are web services that provide users with a way to create surveys. They vary in feature limitations, user experience, and quality of available documentation.

The metrics that will be used to compare/evaluate the candidate solutions are flexibility, reliability, and documentation.

- *Flexibility:*

Flexibility concerns the different ways users can utilize the survey platform to create and manage surveys, including any limitations that users may face. This metric also considers how difficult it will be to integrate surveys with the web application. A score of 0/5 for flexibility means that the platform does not provide many features for survey creation, has various limitations, and is difficult to test. Such a score is given to a platform that does not provide effective sharing options. A score of 5/5 defines a platform with a wide range of features and customization options that can be tested with little to no difficulty. Its interface is user friendly and easy to use. There are little to no limitations on utilizing the platform effectively. This platform provides an option to efficiently embed surveys on a website.

- *Reliability:*

It's important to consider the storage, security, and management of data produced by the surveys created. The relative popularity of a platform is a testament to its reliability. A score of 0/5 for reliability describes a platform that does not provide a

secure way of storing data. With such a score, the platform offers limited management options and cannot analyze/display results in an effective manner to the researcher. Such a platform might have a relatively small user base. A score of 5/5 is given if the platform provides a secure way to store data and has various options to efficiently analyze such data. This platform is well established and used by a variety of professionals.

- **Documentation:**

This metric regards how accurate and detailed the information about integration and utilization of the platform is documented. A good candidate should provide complete, accurate, and easy to analyze documentation. The provided information should be necessary and helpful to users. A score of 0/5 for documentation means that the survey platform does not provide a clear and concise overview of what it offers. It means there is little to no information (internal or external) about efficient utilization of the platform. Candidates with this score are lacking in description of the features and their functionality, or provide dense and unnecessary information. A score of 5/5 is given to a platform that has detailed, useful information about its resources and functionalities. Such a platform provides the user with the necessary information for ease-of-use and navigation.

### **Google Forms:**

Google Forms, one of the various services Google offers, is a free web-based application for creating surveys. It includes a library of validation rules for responses, organizational features, and sharing options. Users are able to create/modify surveys with an unlimited number of responses and respondents. The platform can be used with Google Spreadsheets to effectively analyze data collected. As part of Google's web-app suite, the information collected using Google Forms is secured by Google's security features [8].

Researchers utilizing GeekSurvey need participants and data for their studies. Having no limit on how much feedback can be received is essential to successful recruitment. Google Forms also has a simple user interface. Any researcher with basic knowledge of the Internet can quickly create Google Forms surveys [9]. The scores for the platform's flexibility, reliability, and documentation are:

- **Flexibility: 4/5**

The flexibility score for Google Forms is 4/5 because of the platform's customizability, sharing options, and ease-of-use. A perfect score of 5/5 was not given since it has resource limitations. According to Sandra Melo, "It accepts texts up to 500 KB; images up to 2 MB" [9]. However, the resource limitations do not affect researchers because the effective surveys they create should not include such an abundance of resources. Having too many resources on a survey can complicate and disrupt the user experience.

In regards to GeekSurvey, a survey created with Google Forms can easily be embedded into the web application with a link.

- **Reliability: 5/5**

Google Forms is well established, used by many people for personal or business purposes. It allows users to manage data in various ways and apply various restrictions/rules for surveys. Researchers can utilize such functionalities to increase security and produce valid data. Google's privacy and terms stated, "All Google products are built with strong security features that continuously protect your information" [10]. Being a widely used platform that has such features, the reliability score for Google Forms is 5/5.

- **Documentation: 5/5**

The information about Google Forms is helpful and easy to understand. There is no unnecessary information. Everything is stated and explained clearly about its integration, limitations, and features. Thus, the documentation score for Google Forms is 5/5.

### ***Survey Monkey:***

Survey Monkey is a cloud-based tool for creating, sending, and analyzing surveys. This platform helps users with online survey projects by providing a highly customizable tool. Users are able to customize the structure of surveys to match the target audience. The tool has advanced functions designed to distribute surveys on the web. It also has automated features allowing users to view results in real time. SurveyMonkey's reporting features can process data from millions of respondents. The data generated is validated by the use of HIPAA, Norton, TRUSTe features [11].

If GeekSurvey is successful and utilized by a wide range of researchers/participants worldwide, researchers will be able to take advantage of SurveyMonkey's reporting features. Pricing is monthly for this platform, but it does have a free version with limitations. The metric scores for SurveyMonkey are as follows:

- **Flexibility: 3/5**

SurveyMonkey's flexibility is scored as a 3/5 because it is highly customizable with countless features for survey creation and analyzing results. At the same time, users are limited by a subscription. Without a subscription, users are unable to test and utilize the platform's features to its full potential, decreasing flexibility.

- **Reliability: 4/5**

The platform helps users increase data validity by using question, page, block and order randomization [12]. It provides multiple question types which can help users obtain specific data. To analyze the results of surveys, users can go to the Analyze Results

section of a survey and view a summary of the data gathered. With SurveyMonkey, users are able to export data in different formats for collaboration purposes. Security of data is ensured by the use of encryption methods [11]. Thus, the reliability score is 4/5 since the platform is secured, but advanced features for handling data are only available with the paid version.

- **Documentation: 5/5**

The documentation score for SurveyMonkey is 5/5. The platform includes descriptive and useful information about its features and security. It also has a search bar at the very top that can be used to look for keywords. This is very useful and makes it easier to navigate through the documentation. There is a lot of information that is organized in a way that improves readability.

### ***Typeform:***

Typeform is a web-based platform used to design, create, and share engaging surveys. Typeform, according to the platform documentation, is “a flexible and customizable online form service with beautiful and easy to use design” [13]. An email address is the only thing needed to create an account and use the platform for free. There are paid plans (basic, plus, or business) for users that want to use more features and increase the limits of the free version. Typeform’s versatility with features allows users to try the paid features on a trial survey. This will help in deciding if paid features are of use to users and GeekSurvey. Typeform’s flexibility, reliability, and documentation scores are as follows:

- **Flexibility: 3/5**

Typeform receives a score of 3/5 because of the wide range of features it offers for survey creation and sharing. It also has a number of limitations, hence the score. Users are able to create an unlimited number of surveys but have a limit of 10 questions per survey using the free version. Responses are limited to 10 responses per month and only basic survey features are offered. Though users can try out all the paid features for free, the surveys created using trial mode can not be deployed for actual use [13]. Therefore, the flexibility of Typeform decreases because it is difficult to test without a subscription.

- **Reliability: 4/5**

Typeform receives a score of 4/5 as the many options for extracting/analyzing results on Typeform provide for great reliability. Typeform guarantees a secure user experience by having its security and compliance framework certified to international standards. The security documentation stated that, “All of them have been audited by independent companies” [14].

- **Documentation: 3/5**

Typeform provides great information about the platform and functionalities, but the information is not user friendly. The available documentation about using the platform is very complex and could make survey creation more cumbersome than necessary for users. Thus, the documentation score is a 3/5.

**Survey Platform Results:**

**TABLE 2.2.1**

	<b>Flexibility:</b>	<b>Reliability:</b>	<b>Documentation:</b>	<b>TOTAL</b>
<b>Google Forms</b>	4	5	5	<b>14/15</b>
<b>SurveyMonkey</b>	3	4	5	<b>12/15</b>
<b>Typeform</b>	3	4	3	<b>10/15</b>

As shown in Table 2.2.1, the most viable option between the three candidate solutions is Google Forms with a score of 14/15. Google Forms is a well-established platform that only requires a Google account and common internet knowledge to use. An article by Neeraj Agarwal stated, “Assuredly, Google Forms is the most trusted form builder that has a very simple interface” [15]. The platform’s simplicity makes the process of creating and testing surveys less cumbersome than the others. Thus, it ranks above the other two candidate solutions in flexibility and reliability. Researchers will only need a Google Account to quickly and easily create effective surveys, and upload them to GeekSurvey using a web link. There is no need to pay for anything or upgrade subscriptions like Typeform and SurveyMonkey. Researchers will be able to utilize Google Forms to its full potential completely free of cost. Overall, Google Forms is the simplest, most reliable and effective tool for researchers to utilize.

The plan to prove Google Forms’ feasibility is to create a fully functional survey and test its sharing option using a link, along with its features. The first step is to create a survey with a variety of questions. Then, mock the embedding of a survey link on GeekSurvey by storing the survey access link and testing its capability for authorized participation. Finally, test a redirect link on the survey completion page. The goal is for both the access and redirect link for the test survey to work without complications. This will display Google Forms’ simplicity, ease of use, and integration with GeekSurvey via token.

## 2.3 Technical Analysis: Web Framework

As described by GeekforGeeks, a web application framework (more commonly referred to as a web framework) is “designed to support the development of web applications” [16]. Web frameworks are typically categorized as opinionated or unopinionated, where an opinionated framework tends to believe and guide developers to do something in a particular way while an unopinionated framework does not [17]. Since GeekSurvey is a web application, a quality web framework will be critical in supporting the development of a quality product. The chosen web framework would need to be well documented and easy to use, as well as provide the capability to build and implement features only as needed.

The candidate solutions for web frameworks are Express.js (also referred to as ExpressJS or Express), Django, and Ruby on Rails (also referred to as Rails). They are all open-source and free web application frameworks that provide exceptional features. They differ in their implementation, as they are written in different programming languages and have different levels of integration support for things like databases. The metrics that will be used to compare and evaluate the web framework candidates are as follows:

- *Security:*

For security, the web framework's focus on security and their security features will be taken into consideration. A score of 5/5 indicates that the web framework prioritizes and offers quality security features. A score of 3/5 indicates that the web framework may still prioritize security to an extent and offers some security features. A score of 1/5 indicates that the web framework does not prioritize security and/or does not offer security features.

- *Documentation:*

Regarding documentation, a web framework that has a rich and well outlined documentation to assist in the development of GeekSurvey is essential. A score of 5/5 indicates that the documentation is of high quality and will be helpful during development. A score of 3/5 indicates that the documentation may be of decent quality and relatively helpful. A score of 1/5 indicates that the documentation is of poor quality and generally unhelpful.

- *Customizability:*

Regarding customizability, a web framework that has its major features decoupled is ideal. This would facilitate the ability to build in only the features needed for GeekSurvey, which would ultimately reduce the development complexity/overhead and increase performance. A score of 5/5 indicates that the web framework is highly customizable. A score of 3/5 indicates that the web framework is relatively customizable. A score of 1/5 indicates that the web framework offers little to no customizability.



**Express.js:**

The first web framework candidate is Express.js. As per the official website [18], Express.js is a “Fast, unopinionated, minimalist web framework for Node.js.” Express.js was originally authored by TJ Holowaychuk in 2010 [19]. In 2014, Holowaychuk effectively “sold” the Express.js project to a Node.js based startup named StrongLoop. Holowaychuk did this by selling the ownership of the Express.js GitHub repository, the npm ownership for publishing future releases, and the “expressjs.com” domain and website [20]. This was met with an incredible amount of controversy. IBM would later acquire StrongLoop in late 2015 and hand off the Express.js project to the Node Foundation in early 2016. Today, Express.js is used by organizations like Fox Sports, PayPal, Uber, and IBM [19].

- **Security - 4/5**

The Express.js official website provides security best practices for applications in production [21]. They outline how to migrate an installation of a deprecated version of Express.js and provide advice for ensuring dependencies are secure. They also recommend the use of TLS (transport layer security), the Helmet Node.js security package, and cookies. Security for Express.js would be rated as a 4/5 as the official website clearly outlines a great amount of advice regarding security best practices, but many of these security features require manual implementation.

- **Documentation - 4/5**

The Express.js official website also contains a plethora of quality documentation [22]. The documentation is well written and is separated into sections about the creation of an Express application and the methods for application, request, response, and router objects. The documentation for Express.js would be rated as a 4/5 as the official website provides great code examples and clear explanations, but it can feel awkward to navigate.

- **Customizability - 5/5**

As stated on the Express.js official website [18], Express.js is unopinionated. Like other unopinionated frameworks, Express.js does not provide opinions regarding the correct way to handle a particular task. Express.js does not have many restrictions regarding what components must be used or what is the best way to integrate the components to handle a task [23]. The only downside to this is that it is up to the developers to find the most suitable tools/components to handle a particular task. Customizability for Express.js would be rated as a 5/5 as the unopinionated nature provides a lot of flexibility regarding customization.

**Django:**

The second candidate is Django. As per the official website [24], “Django makes it easier to build better web apps more quickly with less code.” Django was created in 2003 by Adrian Holovaty and Simon Willison [25]. Both Holovaty and Willison were working for the Lawrence Journal-World newspaper at the time. They were both tired of maintaining large websites written in PHP, and wanted to switch to doing web development in Python. They looked at Python web development tools available at the time and were dissatisfied with them. Before Willison’s internship at the paper was set to end in 2004, he assisted Holovaty in hiring his replacement, Jacob Kaplan-Moss. Holovaty and Kaplan-Moss would continue to develop Django, and in 2005 were successful in convincing the owner of the newspaper to open source the project [26]. Today, Django is used by organizations like Instagram, Mozilla, Disqus, and BitBucket [25].

- **Security - 5/5**

The Django official website proudly states that they take security seriously and help developers avoid common security mistakes [24]. They have a security overview in the documentation section of their website where they talk about Django’s built-in security features, such as protections against Cross site scripting (XSS), SQL injection, and clickjacking [27]. They also include advice on how to secure a Django-powered website. Security for Django would be rated as a 5/5 as the official website clearly outlines notable and relevant security features, as well as advice on security practices when using Django.

- **Documentation - 5/5**

The Django official website also contains a large amount of quality and well organized documentation. The documentation landing page provides you with resources for things such as first steps, getting help, a high-level overview about how the documentation is organized, and separate sections with references to documentation for specific subjects [28]. The API reference guide is also very helpful and provides coding examples for each API listed [29]. Documentation for Django would be rated as a 5/5 as the documentation is very well organized and provides quality examples.

- **Customizability - 4/5**

As far as web frameworks go, Django can be classified as “somewhat opinionated.” This means that Django takes a hybrid approach compared to more traditional opinionated and unopinionated frameworks. Django provides a set of components to handle most web development tasks and also suggests some ways to use them. With that being said, Django’s architecture strives for decoupling, meaning that the ability to pick and choose from a variety of options is available. If none of those options meet the needs of this project, support for new ones can be added [17]. Customizability for Django would be rated as 4/5, as the hybrid nature provides a great place to start while also having the flexibility to choose alternative routes if necessary.

### ***Ruby on Rails:***

The third and final candidate is Ruby on Rails. As per the official website, “Rails is a web-application framework that includes everything needed to create database-backed web applications according to the Model-View-Controller (MVC) pattern” [30]. Ruby on Rails was created in 2003 by David Heinemeier Hansson. Hanson developed Ruby on Rails when he was working on the code base for Basecamp, a tool by 37signals (now Basecamp). Hanson extracted Ruby on Rails from the Basecamp code base and released it as an open-source project in July of 2004 [31]. Today, Ruby on Rails is used by organizations like Airbnb, GitHub, Shopify, and Twitch [32].

- **Security - 3/5**

The Ruby on Rails official website provides guides on a variety of topics [33]. One of those guides regards securing Ruby on Rails applications [34]. Some of the common security problems discussed in the guide include session hijacking, cross-site request forgery (CSRF), and SQL injection. For each of these problems, they provide countermeasures that can be taken and implemented in order to reduce the occurrence of these security problems. Security for Ruby on Rails would be rated as a 3/5 as the official website outlines notable security problems and their countermeasures, but the countermeasures are lacking in detail.

- **Documentation - 3/5**

The Ruby on Rails official website provides both guides for relevant topics and documentation for the Ruby on Rails API [33], [30]. The API documentation landing page provides valuable information, such as information to get started, an overview of each layer of the MVC design pattern, and an overview of some relevant libraries. There is also a search tool for finding classes and methods in a much more efficient way. Documentation for Ruby on Rails would be rated as a 3/5 because despite the documentation being useful, the navigation of the documentation felt awkward. In addition, some methods only had the bare minimum regarding documentation.

- **Customizability - 3/5**

As far as web frameworks go, Ruby on Rails can be classified as opinionated. Ruby on Rails greatly encourages the use and following of their development philosophy, which can be seen in The Rails Doctrine [35]. The doctrine pushes for code conventions such as the MVC design pattern. Luckily, this design pattern promotes modularity and extensibility, meaning that it does allow flexibility when developing an application. Customizability for Ruby on Rails would be rated as a 3/5 since it does allow for an amount of flexibility and customizability, so long as you respect the pillars of The Rails Doctrine.

**Web Framework Results:****TABLE 2.3.1**

	<b>Security:</b>	<b>Documentation:</b>	<b>Customizability:</b>	<b>TOTAL:</b>
<b>Express.js</b>	4	4	5	<b>13/15</b>
<b>Django</b>	5	5	4	<b>14/15</b>
<b>Ruby on Rails</b>	3	3	3	<b>9/15</b>

As can be seen in Table 2.3.1, the results of the ranking put Django as the most promising solution with a score of 14/15. While no web framework is ever truly secure, Django seems to put more of an emphasis on securing applications and providing security features out of the box compared to the other frameworks. Django's documentation provides the clearest navigation and examples out of the three frameworks discussed. Despite Django not scoring the highest for customizability due to its hybrid opinionated nature, Django's customizability options should be more than sufficient for the needs of the project.

The current plan for demonstrating the feasibility of this solution is to set up an example Django project with a mock home page of what GeekSurvey could potentially look like. This demonstration should show that the web framework can be used to efficiently and effectively develop a web application.

## 2.4 Technical Analysis: Database

A quality database is crucial in website development, especially for large web applications like GeekSurvey. Without databases, developers could only create static web pages and it would be difficult to update or store historical data in real time. In addition, a suitable database system is required to establish quality interaction between the front and back ends of a web application. The front end is the web page displayed to users in the browser. The back end refers to the data interaction with the front end and the saving and reading of website data. When consumers send a request for data, the backend should receive the request and respond to it. The backend usually queries the database through Structured Query Language (SQL) statements which return a piece of data. The piece of data is used to generate a responsive web page that is usually returned to the front end in the form of HTML. Thus, a database is a key component to any complex web application. For the final product, the database should be reliable, well documented, and synergize well with the chosen web framework.

The candidate solutions for the database are PostgreSQL, MySQL and SQLite. All three candidates have wide usage in web application development. The metrics used to compare and evaluate these candidates are as follows:

- *Reliability:*

Reliability concerns the ability to protect the database from data leakage and change or damage. It can be measured by accuracy and consistency. A score of 0/5 for reliability is related to a database that does not perform well with regards to accuracy and consistency. An example of this would be a candidate that can not query data accurately or store data without leakage. A score of 5/5 for reliability is related to a database that has a good performance in both accuracy and consistency. A candidate with this score will not leak any data or query data inaccurately.

- *Documentation:*

Documentation focuses on how well developers will be able to integrate and use the databases. It helps developers understand more about the set of data descriptions and facilitates the management, maintenance, and retrieval of resources. It can be measured by readability, richness in content, and maintenance. A score of 0/5 for documentation indicates that the database does not provide a readable, clear, and rich explanation of what it offers. Developers will not receive much help from the documentation. A score of 5/5 means the documentation performs well in terms of readability, rich-content, and maintenance.

- *Compatibility:*

Compatibility is about how well the database works with other project dependencies. The chosen database will need to work very closely with Django, and will need to run on the

chosen web host. A score of 0/5 for compatibility means that the database has major difficulties working with one or more project dependencies. A score of 5/5 means the database is easily compatible with most/all of the project dependencies.

### **PostgreSQL:**

The first of the database candidate solutions is PostgreSQL. PostgreSQL is a very powerful and open source client/server Relational Database Management System (RDBMS). PostgreSQL was originally conceived in 1986 and was called the Berkley Postgres Project at that time. It was initially released in 1989 after a long process of development and evolution [36]. Since then, it has become a mature database management system. The scorings for PostgreSQL as a database solution are as follows:

- **Reliability - 5/5**

The reliability of postgresQL allows data to be saved safely, even in extreme cases where a computer is fatally damaged. Reliability is one of the main reasons to choose PostgreSQL, as PostgreSQL performs well with regards to data accuracy and consistency [37].

- **Documentation - 5/5**

The documentation of PostgreSQL is detailed and user-friendly. PostgreSQL provides a clear and concise description of what it offers and provides users with information on how to use it effectively.

- **Compatibility - 4/5**

The chosen candidate for web frameworks is Django. There is a great library for PostgreSQL integration in Python called psycopg [38]. This makes PostgreSQL highly compatible with a Django project, but has some minor difficulties working within a container because it requires more configuration than simpler databases.

### **MySQL:**

The second database candidate solution is MySQL. MySQL was first released in May of 1995 [39]. It has been used by many small and large businesses because of its speed and easy-to-use features. In terms of web applications, MySQL is one of the best RDBMS applications. The SQL language used by MySQL is the most commonly used standardized language for accessing databases. It stores data in different tables instead of putting all data in a singular, larger table. This provides an increase in both speed and flexibility. MySQL is a simple database that is relatively easy to set up and manage. The scorings for MySQL as a database solution are as follows:

- **Reliability - 4/5**

As a lightweight database system, MySQL focuses on speed and reliability [30]. The only drawback is that in order to maintain stability and reliability, the database needs to be regularly maintained.

- **Documentation - 4/5**

The documentation of MySQL provides users with basic information on how to use it effectively. The drawback of the documentation is that it is not user friendly. There are many complex concepts that can confuse new users.

- **Compatibility - 4/5**

As a widely used database system, MySQL performs well with regards to compatibility. It is convenient for Django to operate a MySQL database because each model in Django corresponds to a table in MySQL. However, it should be noted that earlier versions of MySQL are not well supported by Django as some features are not supported.

### **SQLite:**

The third database candidate solution is SQLite. It was initially released in August of 2000. It is a RDBMS contained in a relatively small C library. A single database connection is allowed by SQLite to access multiple database files. This allows for helpful features like copying data between databases with a single command [40]. SQLite can also create in-memory databases that are fast to work with. The scorings for SQLite as a database solution are as follows:

- **Reliability - 3/5**

SQLite aims to be a high-reliability storage solution that just works, but it does have some downsides. One such downside is that it does not provide built-in authentication [41]. This can lead to unsatisfactory accuracy and consistency.

- **Documentation - 4/5**

The SQLite documentation is user-friendly as there are not many professional terms that may confuse new users. This means that the documentation is both clear and readable. It also provides users with basic information about SQLite. The documentation is lacking in terms of containing rich content that goes deeper into the product itself [42].

- **Compatibility - 4/5**

Django has access to quality libraries for working with SQLite, so integrating SQLite into a Django project should be relatively simple. SQLite is less than ideal because it is

typically used for smaller projects [43], while GeekSurvey must be able to scale to the size of a relatively large project.

**Database Results:**

**TABLE 2.4.1**

	<b>Reliability:</b>	<b>Documentation:</b>	<b>Compatibility:</b>	<b>TOTAL</b>
<b>PostgreSQL</b>	5	5	4	<b>14/15</b>
<b>MySQL</b>	4	4	4	<b>12/15</b>
<b>SQLite</b>	3	4	4	<b>11/15</b>

Table 2.4.1 shows the comparisons between PostgreSQL, MySQL and SQLite in terms of the metrics. As can be seen in the table above, PostgreSQL is the most promising solution with a score of 14/15. Its documentation is user friendly and provides developers with needed information. It can also work well with other project dependencies, such as Django.

The current plan for demonstrating the feasibility of this database solution is to create a PostgreSQL database and perform some proof-of-concept tests, such as adding and removing mock data from a Django project. This demonstration should demonstrate its feasibility in terms of reliability, documentation, and compatibility.



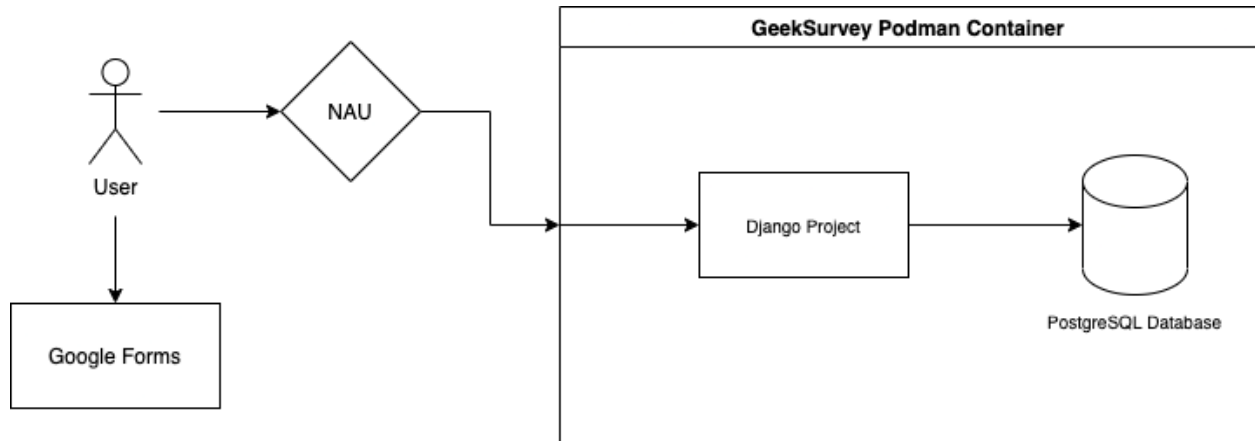
### 3. Technical Integration

It is important to investigate how these chosen solutions will work together to form a coherent architecture for GeekSurvey. GeekSurvey is a web application intended to serve the needs of a large number of users. The final version of this project will be more complex than a simple website. It will need to provide curated web pages to any given user based on their user type and specifics of their profiles, such as what surveys they're involved in or eligible for.

GeekSurvey will need to frequently access a database and use a framework to generate responsive web pages for each user. The web application will need to support the use of external survey platforms and manage information about each survey, such as which researcher initiated it and which participants have completed it. All of these requirements will be accomplished by software that will have to run on a computer. Each of these aspects of the final system will need to integrate with some or all of the other aspects.

The chosen technical solutions fit practically into one architecture in the following way. GeekSurvey will run on a containerized virtual computer using Podman with NAUTS. Developers will use Podman commands to edit and deploy the web application to an official NAU URL. A PostgreSQL database will be installed in the container and initialized with a small amount of testing data. The container will be installed with a custom Django project for handling user input and generating GeekSurvey web pages. The virtual computer will listen for HTTP requests and pass the requests to the Django project.

Django will read and write to the database using the psycopg Python library. It will do these reads and writes as a function of user input through HTTP requests. For example, when a user creates an account, the Django system will run some code to recognize the request type and write the new account information to the database. The user interface of the web application will allow researchers to add Google Forms surveys. When a participant completes a survey, Google Forms will redirect their web browser to the GeekSurvey website with a unique token. The Django application will recognize this token and record the user's completion in the PostgreSQL database.



**Figure 3.1** - GeekSurvey System Architecture Diagram

Figure 3.1 shows a visual representation of the envisioned system architecture. The arrows in this diagram represent user interaction with the final product. When a user accesses the GeekSurvey website, their requests will first be handled by an NAU web server and then passed to the GeekSurvey container. The container has two core components, the Django project and the PostgreSQL database. Incoming web requests are passed from the container to the Django project. Then the Django project may query the database to write new data or retrieve data for the user. Users have to externally interact with Google Forms for survey creation and completion. GeekSurvey will manage the surveys by storing links to Google Forms in the database.

## 4. Conclusion

Overall, the project will need a server to host its web application, a survey platform to integrate surveys, a web framework to support the development of the application, and a database system to handle large and important data.

The server must be able to handle requests and responses with ease while also running the database and web framework. It will need to be relatively easy to transfer in the event that the project needs to be hosted elsewhere. Ideally, the server should be as inexpensive as possible while still delivering everything that is needed. NAUITS is the best solution for the hosting challenge. Despite its drawbacks with regard to scalability, its excellent performance in both transferability and cost more than meets the needs of GeekSurvey.

The web application will need to provide researchers a way to obtain relevant information about software engineering research participants and set criteria that participants must meet. To do this, a survey platform that can be utilized to generate effective surveys is needed. Google Forms is the best solution for the survey platform challenge as it will allow developers to integrate surveys with ease and track survey creation and use.

The framework for the web application will support developers in the development process. The framework will allow developers to program the application with few difficulties and implement additional features that can enhance the functionality of the product. Django has outstanding security measures, customizability, and library support with detailed documentation. Therefore, Django is the chosen solution for the web framework.

Regarding the database, it should be able to handle the significant amount of data that GeekSurvey may eventually take in and produce. A database that is reliable, well documented, and is compatible with the chosen framework is desirable. PostgreSQL's reliability received a perfect score as it does everything necessary to ensure security. It also provides detailed and user-friendly documentation that serves as a guide for implementing the database to its full potential. Above all, its compatibility with Django establishes synergy for the complete GeekSurvey system architecture.

In conclusion, the candidate solutions that were investigated and chosen for GeekSurvey will work together to provide a quality user experience. These candidates will support the needs of this project, advancing the development of GeekSurvey. They are the roots of success that will result in the creation of a platform that will satisfy the client in their quest to acquire a valid pool of subjects and recruit them for research.

## 5. References

- [1] "Northern Arizona University," *Information Technology Services*. [Online]. Available: <https://in.nau.edu/its/>. [Accessed: 01-Nov-2021].
- [2] W. Wilson, "Working with Podman (docker) containers," *Podman Guide*. [Online]. Available: [https://nau.service-now.com/kb\\_view.do?sysparm\\_article=KB0016911&sysparm\\_rank=1&sysparm\\_tsqueryId=cb72ce611b1f781080628771604bcb38#](https://nau.service-now.com/kb_view.do?sysparm_article=KB0016911&sysparm_rank=1&sysparm_tsqueryId=cb72ce611b1f781080628771604bcb38#). [Accessed: 01-Nov-2021].
- [3] "AWS documentation," *Guides and API References*. [Online]. Available: <https://docs.aws.amazon.com/index.html>. [Accessed: 01-Nov-2021].
- [4] "Amazon Lightsail pricing," *Amazon*, 2010. [Online]. Available: <https://aws.amazon.com/lightsail/pricing/>. [Accessed: 19-Oct-2021].
- [5] *Pricing*. [Online]. Available: <https://www.digitalocean.com/pricing>. [Accessed: 19-Oct-2021].
- [6] "GitHub Student Developer Pack," *DigitalOcean*. [Online]. Available: <https://www.digitalocean.com/github-students/>. [Accessed: 01-Nov-2021].
- [7] J. Papiewski, "What is a URL token?," *Techwalla*. [Online]. Available: <https://www.techwalla.com/articles/what-is-a-url-token>. [Accessed: 02-Nov-2021].
- [8] A. A. A. Demarest, "What are google forms? everything you need to know about Google Workspace's online form builder," *Business Insider*, 22-Jan-2021. [Online]. Available: <https://www.businessinsider.com/what-is-google-forms>. [Accessed: 01-Nov-2021].
- [9] S. Melo, "Advantages and disadvantages of google forms," *DataScope*, 20-Aug-2021. [Online]. Available: <https://datascope.io/en/blog/advantages-and-disadvantages-of-google-forms/>. [Accessed: 02-Nov-2021].
- [10] "Privacy policy – privacy & terms," *Google*. [Online]. Available: <https://policies.google.com/privacy?hl=en#infosecurity>. [Accessed: 01-Nov-2021].
- [11] T. Black and M. Sharp, Eds., "SurveyMonkey review: Pricing, pros, cons & features," *CompareCamp.com*, 18-Dec-2018. [Online]. Available: <https://comparecamp.com/surveymonkey-review-pricing-pros-cons-features/>. [Accessed: 02-Nov-2021].
- [12] "Question & Page Randomization," *SurveyMonkey Help Center*. [Online]. Available: [https://help.surveymonkey.com/articles/en\\_US/kb/Question-Page-Randomization](https://help.surveymonkey.com/articles/en_US/kb/Question-Page-Randomization). [Accessed: 02-Nov-2021].
- [13] <https://help.typeform.com/hc/en-us/articles/360038717092-What-is-Typeform->. [Accessed: 01-Nov-2021].
- [14] "Security at typeform – help center," *Typeform*. [Online]. Available: <https://help.typeform.com/hc/en-us/articles/360029259552-Security-at-Typeform>. [Accessed: 01-Nov-2021].
- [15] N. Agarwal, "How to redirect custom URL after google forms submission?N," *FormGet*, 08-Jul-2020. [Online]. Available: <https://www.formget.com/how-to-redirect-custom-url-after-google-forms-submission/>. [Accessed: 02-Nov-2021].

- [16] "Top 10 Frameworks for Web Applications," *GeeksforGeeks*, 05-Jun-2020. [Online]. Available: <https://www.geeksforgeeks.org/top-10-frameworks-for-web-applications/>. [Accessed: 19-Oct-2021].
- [17] "Django Introduction - Learn Web Development: MDN," *MDN Web Docs*. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>. [Accessed: 19-Oct-2021].
- [18] "Node.js web application framework," *Express*. [Online]. Available: <https://expressjs.com/>. [Accessed: 19-Oct-2021].
- [19] "Express.js," *Wikipedia*, 08-Oct-2021. [Online]. Available: <https://en.wikipedia.org/wiki/Express.js>. [Accessed: 19-Oct-2021].
- [20] D. Tzur, "The Unbelievable History of the Express JavaScript Framework," *The Full Stack*, 27-Jan-2017. [Online]. Available:
- [21] "Production Best Practices: Security," *Express*. [Online]. Available: <https://expressjs.com/en/advanced/best-practice-security.html>. [Accessed: 19-Oct-2021].
- [22] "5.X API," *Express*. [Online]. Available: <https://expressjs.com/en/5x/api.html#express>. [Accessed: 19-Oct-2021].
- [23] Arvindpdmn, "Express.js," *Devopedia*, 01-Mar-2019. [Online]. Available: <https://devopedia.org/express-js#qst-ans-0>. [Accessed: 19-Oct-2021].
- [24] *Django*. [Online]. Available: <https://www.djangoproject.com/>. [Accessed: 19-Oct-2021].
- [25] "Django (web framework)," *Wikipedia*, 17-Oct-2021. [Online]. Available: [https://en.wikipedia.org/wiki/Django\\_\(web\\_framework\)](https://en.wikipedia.org/wiki/Django_(web_framework)). [Accessed: 19-Oct-2021].
- [26] S. Willison, "What is the history of the Django web framework? why has it been described as 'developed in a newsroom'?", *What is the history of the Django web framework? Why has it been described as "developed in a newsroom"?* [Online]. Available: <https://simonwillison.net/2010/Aug/24/what-is-the-history/>. [Accessed: 19-Oct-2021].
- [27] "Security in Django," *Django*. [Online]. Available: <https://docs.djangoproject.com/en/3.2/topics/security/>. [Accessed: 19-Oct-2021].
- [28] "Django documentation," *Django*. [Online]. Available: <https://docs.djangoproject.com/en/3.2/>. [Accessed: 19-Oct-2021].
- [29] "API Reference," *Django*. [Online]. Available: <https://docs.djangoproject.com/en/3.2/ref/>. [Accessed: 19-Oct-2021].
- [30] "Welcome to rails," *Ruby on Rails API*. [Online]. Available: <https://api.rubyonrails.org/>. [Accessed: 19-Oct-2021].
- [31] RailsCarma, "The History of Ruby on Rails," *Medium*, 21-Sep-2015. [Online]. Available: <https://railscarma.medium.com/the-history-of-ruby-on-rails-986ead4e0e0a>. [Accessed: 19-Oct-2021].
- [32] "Ruby on rails," *Ruby on Rails*. [Online]. Available: <https://rubyonrails.org/>. [Accessed: 19-Oct-2021].
- [33] "Ruby on Rail Guides (v.6.1.4)," *Ruby on Rails Guides*. [Online]. Available: <https://guides.rubyonrails.org/index.html>. [Accessed: 19-Oct-2021].

- [34] "Securing Rails Applications," *Ruby on Rails Guides*. [Online]. Available: <https://guides.rubyonrails.org/security.html>. [Accessed: 19-Oct-2021].
- [35] D. H. Hansson, "The Rails Doctrine," *Ruby on Rails*. [Online]. Available: <https://rubyonrails.org/doctrine/>. [Accessed: 19-Oct-2021].
- [36] "PostgreSQL", Wikipedia, 30-Sep\_2021. [Online]. Available: <https://en.wikipedia.org/wiki/PostgreSQL>. [Accessed: 19-Oct-2021].
- [37] "Reliability," *PostgreSQL Documentation*, 08-Nov-2018. [Online]. Available: <https://www.postgresql.org/docs/9.3/wal-reliability.html>. [Accessed: 19-Oct-2021].
- [38] *Psycopg*. [Online]. Available: <https://www.psycopg.org/>. [Accessed: 01-Nov-2021].
- [39] *MySQL*. [Online]. Available: <https://www.mysql.com/>. [Accessed: 01-Nov-2021].
- [40] "What Is SQLite," *SQLite Tutorial*, 08-Jul-2020. [Online]. Available: <https://www.sqlitetutorial.net/what-is-sqlite/#:~:text=SQLite>. [Accessed: 01-Nov-2021].
- [41] "SQLite vs MySQL - What's the Difference," *Hostinger Tutorials*, 20-Jul-2021. [Online]. Available: <https://www.hostinger.com/tutorials/sqlite-vs-mysql-whats-the-difference/>. [Accessed: 19-Oct-2021].
- [42] "Documentation," *SQLite Documentation*. [Online]. Available: <https://www.sqlite.org/docs.html>. [Accessed: 19-Oct-2021].
- [43] S. Wang, "Should I use sqlite, PostgreSQL, or mysql?," *DEV Community*, 22-Sep-2021. [Online]. Available: <https://dev.to/codesphere/should-i-use-sqlite-postgresql-or-mysql-1o4b>. [Accessed: 01-Nov-2021].