CADT
IDT
វិទ្យាស្ថានបច្ចេកវិទ្យាឌីជីថល
Institute of Digital Technology

Report

Capstone Project I

Shellify

Project Supervisor
Mr. Saly KEO

Authors of the Project
Keopitou Doung - Chandara Yi - Channudam Ray
Whatanak Kean - Danet Reab

Date of Final Submission
May 30 Tuesday 2023

Report Summary

This Capstone Project, titled "Shellify", supervised by Mr. Saly KEO and authored by Keopitou Doung, Chandara Yi, Channudam Ray, Whatanak Kean, and Danet Reab. The date of final submission is May 30, 2023.

In this report, we will include sections such as Introduction, Background, Design Requirements, Implementation, Discussion, Conclusions, and Recommendations. This project aims to simplify the process of Internet exposure by developing an easy-to-use web-based interface for creating and managing Ubuntu Linux shell sessions. It also aims to reduce the overall costs of cloud services and create affordable cloud solutions that are specifically designed for beginners or students.

We will be discussing various aspects of the project including its objectives, scope, significance, economic implications, environmental considerations, ethical issues, health and safety considerations, social implications, previous work, design requirements, technology and services used in implementation, problems encountered during design and implementation and their solutions and workarounds. It also concludes with recommendations for future work on Shellify.

Table of Content

Table of Tables

Table of Figures

Note

To clarify, **"QuickShell"** is the code name used during the development of the product, while **"Shellify"** is the final product name that will be used for marketing and commercial purposes. Throughout this report, we may use these terms interchangeably to refer to the same product.

Introduction

1. Problem Statement

Exposing an application to the Internet can be complex and expensive, creating significant challenges and barriers for developers and small businesses. In particular, cloud computing services are often required, which can be costly and require technical expertise. This creates a significant cost barrier for students and small businesses with limited resources. As a result, there is a need for a solution that can simplify the process of Internet exposure and reduce the cost barriers associated with cloud computing services.

In this report, we present a digital product that aims to address this problem by providing a user-friendly and affordable solution for Internet exposure. Our digital product is designed to simplify the process of configuring applications for Internet exposure and reduce the reliance on costly cloud computing services. It incorporates intuitive tools and guides to help developers navigate the technical complexities of cloud computing and network configuration. Additionally, it provides affordable pricing plans that cater to the needs of small businesses and students. By addressing these issues, our digital product can help reduce the cost barriers associated with Internet exposure and make it more accessible to a broader range of users.

2. Objectives and Scope

The objective of this project is to simplify the process of Internet exposure by developing an easy-to-use web-based interface for creating and managing Ubuntu Linux shell sessions. The project aims to reduce the barriers to entry for developers, particularly those who are new to the field, by providing more accessible resources and support to help developers navigate the technical requirements.

Another objective of the project is to reduce the overall costs of cloud services by negotiating lower rates with cloud providers or developing alternative solutions that are more cost-effective for small businesses and students. The project also aims to create affordable cloud solutions that are specifically designed for beginners or students, making it easier for developers to gain experience and develop their skills, regardless of their financial situation.

The scope of the project includes the development of a web-based interface for creating and managing Ubuntu Linux shell sessions, one-click functionality to create a web shell session on Ubuntu Linux, a full-featured Ubuntu Linux environment accessible through the web shell session, user authentication and authorization, integration with cloud hosting providers, support for different types of devices and browsers, customization of the web shell session environment, data backup and recovery features, and user support and documentation.

The project is limited to Ubuntu Linux distributions and does not support other Linux distributions or operating systems besides Ubuntu Linux. The project does not integrate with third-party services or applications not related to hosting the Ubuntu Linux environment or managing the web shell session. The project also does not integrate with hardware or physical devices beyond the user's local device for remote access and does not deploy the QuickShell application on-premises. Additionally, the project does not support non-standard programming languages or frameworks that are not commonly used in the Ubuntu Linux environment.

3. Significance of Addressing These Opportunities

Addressing the opportunities identified in this report can have a significant impact on the accessibility and affordability of cloud computing and Internet exposure. By simplifying the process of Internet exposure and reducing the costs associated with cloud computing, Shellify can help to remove barriers for developers, particularly those who are new to the field or have limited resources. This can result in a more diverse and inclusive developer community, with greater participation from individuals who might otherwise be deterred by the complexity and costs associated with cloud computing and Internet exposure.

OPPORTUNITY	POTENTIAL BENEFITS
Simplify Internet Exposure	Reduced barriers to entry for developers, particularly novices
Reduce Cloud Costs	Increased affordability of cloud services for small businesses and students
Create Affordable Cloud Solutions	Increased accessibility to cloud resources and technical knowledge for students and individuals with limited resources

Table 1 - Opportunities and Potential Benefits

Additionally, by creating affordable cloud solutions and providing more accessible resources and support, Shellify can help to bridge the technical knowledge gap for developers, enabling them to gain experience and develop their skills regardless of their financial situation. This can lead to a more skilled and knowledgeable workforce in cloud computing and related fields, with a broader range of individuals able to contribute to the development of new digital products and technologies.

Overall, addressing these opportunities can lead to a more equitable and accessible technology landscape, with greater participation and diversity in the development of digital products and services. It can also contribute to the growth and development of the technology industry, with more individuals able to access the resources and support they need to succeed in this field.

Background

1. Context and History

   The field of web development has undergone significant changes over the past few decades, with the rise of cloud computing and the increasing importance of web applications. As more businesses and individuals move their operations online, there is a growing need for developers to have access to tools and platforms that simplify the process of building and deploying web applications.

   One key aspect of web development is the use of command-line interfaces (CLI) to interact with operating systems and perform tasks such as installing packages, managing files, and configuring servers. However, CLI tools can be complex and intimidating for new developers, and they require a certain level of technical expertise to use effectively.

   To address these challenges, a number of web-based shell environments have been developed over the years, such as Google Cloud Shell, AWS Cloud9, and Microsoft Azure Cloud Shell. These platforms provide developers with a web-based interface for accessing a Linux command-line interface, eliminating the need for local installations and enabling remote access to the shell environment.

   However, these existing platforms have limitations in terms of their features, cost, and ease of use. Many developers still find the process of setting up and configuring a shell environment to be time-consuming and difficult, particularly if they are new to the field. Additionally, existing platforms can be expensive to use, particularly for small businesses and individual developers.

   The Shellify project aims to address these limitations by providing a simple and affordable web-based shell environment that can be accessed from any device with an internet connection. The platform will provide a range of features and functionality designed to simplify the process of developing and deploying web applications, including one-click access to a full-featured Ubuntu Linux environment, customizable user preferences, and data backup and recovery features.

2. Economic Implications

   The development of Shellify has the potential to impact the economy in various ways. Firstly, there are costs associated with the development of the product, including research and development, materials, manufacturing, marketing, and distribution costs. These costs need to be taken into consideration to ensure that the final product is financially viable and can be competitively priced in the market.

   Secondly, Shellify can have a significant impact on the cloud computing industry, which has seen significant growth in recent years. As more and more businesses move their operations to the cloud, there is a growing demand for cost-effective cloud solutions that can meet the needs of small and medium-sized enterprises (SMEs). Shellify has the potential to fill this gap by providing an affordable and accessible cloud solution that can be used by businesses of all sizes.

   In addition, the development of Shellify can create new job opportunities in the technology and manufacturing sectors. As the product gains traction in the market, there will be a need for skilled professionals in areas such as software development, cloud computing, and hardware manufacturing.

This can have a positive impact on the local and national economy by creating new jobs and increasing economic activity.

Overall, the development of Shellify has the potential to be financially beneficial for both the developers and the wider economy. However, careful consideration must be given to the costs of development and the target market to ensure that the final product is financially viable and can compete in the market.

3. Environmental Considerations

The use of cloud computing and virtualization technologies, such as those used in the Shellify project, can have a positive impact on the environment. By reducing the need for physical hardware and infrastructure, cloud computing can decrease the amount of energy and resources required to maintain and operate traditional data centers. This can lead to a reduction in carbon emissions and energy consumption.

However, it is important to note that the environmental impact of cloud computing and virtualization depends on how these technologies are implemented and used. For example, the use of inefficient data centers or poorly optimized virtual machines can actually result in a higher carbon footprint than traditional data centers. Therefore, it is important for the Shellify project to prioritize energy efficiency and sustainability in its design and implementation.

In addition, the use of cloud computing and virtualization technologies raises concerns about e-waste. As hardware becomes outdated or replaced, it can contribute to the growing problem of electronic waste. To address this issue, the Shellify project can prioritize the use of sustainable and recyclable materials in its hardware and encourage users to responsibly dispose of outdated hardware.

4. Manufacturability

Manufacturability for a cloud-based product like Shellify is less focused on physical components and more on the scalability and reliability of the cloud infrastructure. The use of off-the-shelf cloud services from major providers such as Amazon Web Services, Google Cloud, and Microsoft Azure allows for rapid development and deployment of the product without the need for custom components. This approach also enables the product to be easily scaled to meet the demands of a growing user base.

One potential concern for manufacturability is the reliance on a third-party cloud provider for the infrastructure. This could result in limited control over the underlying hardware and software, as well as potential service interruptions or outages caused by the provider. To mitigate this risk, Shellify could explore the use of multiple cloud providers and redundancy in the infrastructure design.

As Shellify is a cloud-based product, the concept of traditional manufacturing is not applicable. However, we can provide a table outlining the technical considerations for the development and deployment of Shellify:

CONSIDERATION	DESCRIPTION
Material Availability	The availability and reliability of the cloud infrastructure, including servers, storage, and networking hardware, as well as software tools and platforms.
Custom Components	The need for custom software or tools to build and deploy Shellify, such as specialized APIs or integrations with other cloud services.
Hostile Environments	The ability of Shellify to function in different types of cloud environments, such as public, private, or hybrid clouds, and the potential challenges that may arise in each.
Scalability	The ability of Shellify to scale up or down to meet demand, and the impact of increased usage on performance and cost.
Security	The measures in place to ensure the security and privacy of user data and the overall integrity of the system, including encryption, authentication, and access control.

Table 2 - Summary of Manufacturability Considerations for Shellify

Overall, the cloud-based nature of Shellify allows for a high degree of manufacturability and scalability with minimal custom components required.

5. Ethical Issues

As with any technology, there are potential ethical concerns associated with the development and implementation of Shellify. One issue is data privacy and security, as the platform would be storing and processing sensitive user information on the cloud. Ensuring that user data is protected and properly secured is a key responsibility of the developers.

Another ethical concern is the potential for misuse of the platform. While Shellify is intended to be a tool for simplifying internet exposure and reducing cloud costs, it could potentially be used for malicious purposes, such as launching cyber attacks or conducting illegal activities. It is important for developers to take steps to prevent such misuse, such as implementing measures to monitor user activity and identify suspicious behavior.

Finally, there may be ethical questions about the impact of Shellify on the job market. By making it easier and more affordable for developers to access cloud resources and develop their skills, the platform could potentially disrupt traditional roles and job requirements in the tech industry. However, this could also lead to greater democratization and diversification in the field, as more people are able to participate and contribute. It will be important for the developers to consider and address these potential impacts as they design and promote Shellify.

Overall, it is important for the developers of Shellify to be mindful of the potential ethical implications of the platform and take proactive steps to address them. This will help ensure that the product is developed and used in a responsible and beneficial way.

6. Health and Safety Considerations

The Health and Safety Considerations subsection in the background section discusses the potential positive or negative impacts of Shellify on the health and safety of individuals or society. This could include issues such as data privacy and security, as well as any physical risks associated with the use of the product. It is important to consider these factors in the development of Shellify to ensure that the product is safe and secure for its users.

Here are some considerations for the safety impacts of Shellify

CONSIDERATION	DESCRIPTION
Cybersecurity	Measures must be taken to protect against cyber-attacks that could compromise user data and put their health and safety at risk.
Data Privacy	Users will be providing personal and sensitive information, so it is important to ensure that this data is protected and their privacy is maintained.

Table 3 - Summary of Potential Health and Safety Considerations for Shellify

7. Social Implications

The development and adoption of cloud-based technologies, such as Shellify, have significant social implications for various aspects of society, including education, culture, communication, and entertainment.

Education: Shellify provides a platform for learning and practicing coding and programming skills, especially for students who may not have access to expensive hardware or software. This can contribute to greater digital literacy and democratizing access to technical skills.

Culture: The use of cloud computing and related technologies has the potential to create a global culture of innovation and collaboration, enabling people from different parts of the world to connect and work together on projects that were once out of reach due to geographical or financial constraints.

Communication: Shellify can facilitate remote collaboration and communication, allowing people to work together from different locations and time zones. This can improve efficiency, reduce costs associated with travel and commuting, and create opportunities for work-life balance.

Entertainment: The development of cloud-based gaming platforms, enabled by technologies such as Shellify, is becoming increasingly popular. This provides an opportunity for gamers to access high-quality games without the need for expensive gaming hardware, opening up gaming to a wider audience and promoting diversity in the gaming industry.

Overall, Shellify has the potential to transform various aspects of society and contribute to a more connected, collaborative, and diverse world. However, it is important to consider the potential negative impacts and address them proactively to ensure that the benefits of the technology are maximized while minimizing any harm.

8. Previous Work

Research and development of cloud-based applications have been ongoing for several years. The advent of cloud computing has led to an increase in the development and use of cloud-based applications. There have been numerous research works on cloud-based application development, deployment, and management. Researchers have explored different aspects of cloud computing, including security, scalability, reliability, performance, and cost.

Some of the previous works in cloud-based application development include the use of different cloud platforms such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. Researchers have also explored the use of different programming languages and frameworks in developing cloud-based applications. For example, the use of JavaScript and Node.js has become increasingly popular in recent years.

In the specific domain of shell interfaces, there have been several previous works that have explored different aspects of shell interface design and development. Researchers have explored different types of shell interfaces, including graphical shell interfaces, web-based shell interfaces, and command-line shell interfaces.

Some of the previous works in shell interface design include the development of graphical shell interfaces using tools such as Qt and GTK+. Researchers have also explored the use of web-based technologies such as HTML, CSS, and JavaScript in developing shell interfaces. In addition, there have been several research works on the development of command-line shell interfaces, including the use of different programming languages and libraries such as Python and Readline.

Overall, there is a significant body of work in cloud-based application development and shell interface design that provides a foundation for the development of a cloud-based shell interface such as Shellify. The previous works provide valuable insights into the design, development, and deployment of cloud-based applications and shell interfaces, which can be leveraged to build a robust and effective cloud-based shell interface.

Design Requirements

1. Technology and Services

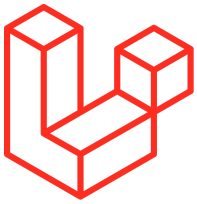


Figure 1 - Laravel Logo

Laravel is a free and open-source PHP web framework created by Taylor Otwell for the development of web applications following the model–view–controller (MVC) architectural pattern and based on Symfony. It has an expressive, elegant syntax and provides powerful features such as thorough dependency injection, an expressive database abstraction layer, queues, scheduled jobs, unit and integration testing, and more.

Figure 2 - Vue.js Logo

Vue.js is a progressive JavaScript framework for building user interfaces. It builds on top of standard HTML, CSS, and JavaScript and provides a declarative and component-based programming model that helps you efficiently develop user interfaces, be they simple or complex. Vue.js is approachable for developers who already know HTML, CSS, and JavaScript and is versatile with an incrementally adoptable ecosystem that scales between a library and a full-featured framework.

Figure 3 - Ansible Logo

Ansible is an IT automation tool that can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero-downtime rolling updates. Its main goals are simplicity and ease of use. Ansible handles configuration management, application deployment, cloud provisioning, ad-hoc task execution, network automation, and multi-node orchestration.

Figure 4 - ExpressJS Logo

ExpressJS, or simply Express, is a back-end web application framework for building RESTful APIs with Node.js. It is designed for building web applications and APIs and has been called the de facto standard server framework for Node.js. Express is a fast, unopinionated, minimalist web framework that provides a robust set of features for web and mobile applications.

Figure 5 - MySQL Logo

MySQL is an open-source relational database management system (RDBMS). It is based on the Structured Query Language (SQL), which is a popular language for accessing and managing records in databases. MySQL is supported by Oracle Company and is available under the GNU license.

Figure 6 - Jenkins Logo

Jenkins is an open-source automation server that provides hundreds of plugins to support building, deploying, and automating any project. It is used for continuous integration and continuous delivery (CI/CD). Jenkins can be easily set up and configured via its web interface and can be extended via its plugin architecture.

Figure 7 - Docker Logo

Docker is an open platform for developing, shipping, and running applications. It enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. Docker Desktop is an application for MacOS, Linux, and Windows machines for building and sharing containerized applications and microservices.

Figure 8 - Digital Ocean Logo

DigitalOcean is a cloud infrastructure provider that offers a variety of products and services for developers and businesses. They offer virtual machines (Droplets), managed databases, object storage (Spaces), block storage (Volumes), load balancers, container registry, and more. Their pricing is simple and predictable with monthly caps and flat pricing.

Figure 9 - PlanetScale Logo

PlanetScale is a MySQL-compatible serverless database that brings you scale, performance, and reliability without sacrificing developer experience. It is powered by Vitess, an open-source database technology invented at YouTube in 2010 to solve their scaling issues with their massive MySQL database. With PlanetScale, you get the power of horizontal sharding, non-blocking schema changes, and many more powerful database features without the pain of implementing them.

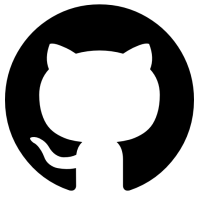


Figure 10 - GitHub Logo

GitHub, Inc. is an Internet hosting service for software development and version control using Git. It provides the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project.

Implementation

1. Project Description and Functions

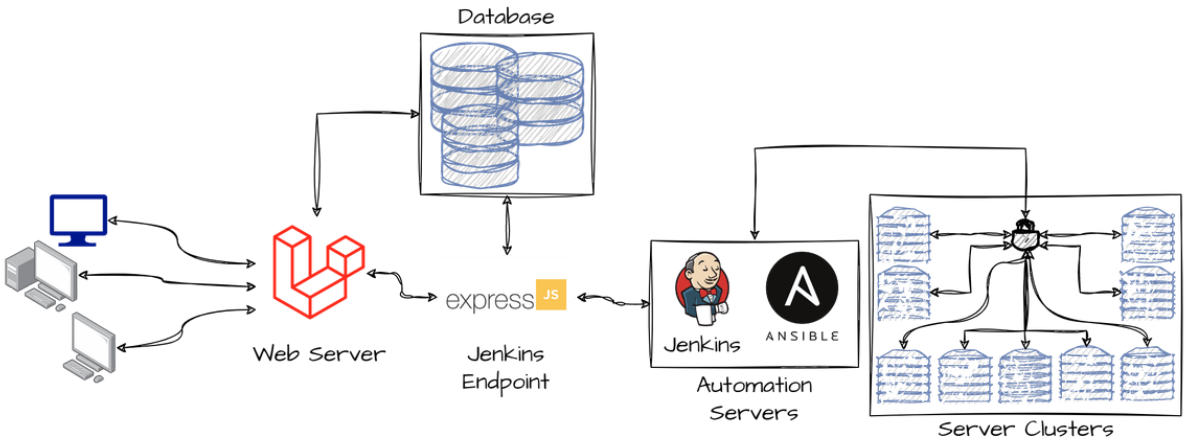
   Shellify aims to provide a simple and efficient way for users to access a full-featured Ubuntu Linux environment through a web shell session. By eliminating the need for local installations or configurations, Shellify can save users time and reduce technical barriers to accessing a Linux environment. Shellify can also improve the efficiency and productivity of developers and system administrators who need to access a Linux environment quickly and easily. In addition to its practical benefits, Shellify can also provide a valuable learning tool for students who want to practice programming in a Linux environment without investing in local hardware or software.

   Shellify is a powerful tool that provides users with easy access to a full-featured Ubuntu Linux environment through a web shell session. Its main technical functionality is to provide a seamless and efficient way for developers and system administrators to access a Linux environment without the need for local installations or configurations. The main technical features and capabilities of Shellify can be listed in the table below

Main Functionality	Description
Login	The login functionality allows users to securely access their Shellify account using their email and password. This ensures that only authorized users can access their shells and subdomains.
Register	The register functionality allows new users to create a Shellify account by providing their email, password, and other required information. This is the first step in accessing the powerful features of Shellify.
Create and Delete Shells	This functionality allows users to quickly and easily create or delete a new web shell session within their Shellify account. This provides them with access to a full-featured Ubuntu Linux environment without the need for local installations or configurations.
Load Shell	This functionality allows users to load their existing web shell(s).
Create or Delete Subdomain (For the Shell)	This functionality allows users to create or delete a custom subdomain for their web shell session. This makes it easy for them to access their shell or the application(s) running in the shell from any device with an internet connection.
Point Domain	This functionality allows users to point their existing domains to a created subdomain within our application. This will allow them to access their application with their own domain name.

Table 4 - Main Functionalities of Shellify

2. Diagrams and Figures
    a. System Architecture



.Figure 11 - System Architecture Diagram

The Shellify system architecture is based on a microservices approach. It consists of the following components:

Web Servers	**Frontend Web Application**	The frontend web application is responsible for providing users with a web-based shell interface. It communicates with the backend API server to execute shell commands and manage user sessions.
	Backend API Server	The backend API server is responsible for handling user authentication, managing user sessions, and executing shell commands on behalf of users. It communicates with the database server to store user and session data.
	Websocket Server	The Socket Server is a key component of Shellify that enables real-time communication between the user's web browser and the Linux environment running on the server, providing a fast and efficient way for users to interact with their shell session, including sending commands and receiving the outputs.
Database Server		The database server is responsible for storing all mandatory data for the backend server and Jenkins Endpoints to work with.
Jenkins Server		The Jenkins server is responsible for building and deploying container images to the Server cluster and automated deployment of the applications for the developers.
Automation Servers		The automation server is responsible for managing the Server cluster infrastructure. It handles container images deployment and manages the infrastructure.

Table 5 - Summary of Shellify's Microservices

b. User Interface

Here are some screenshots of the user interface of Shellify

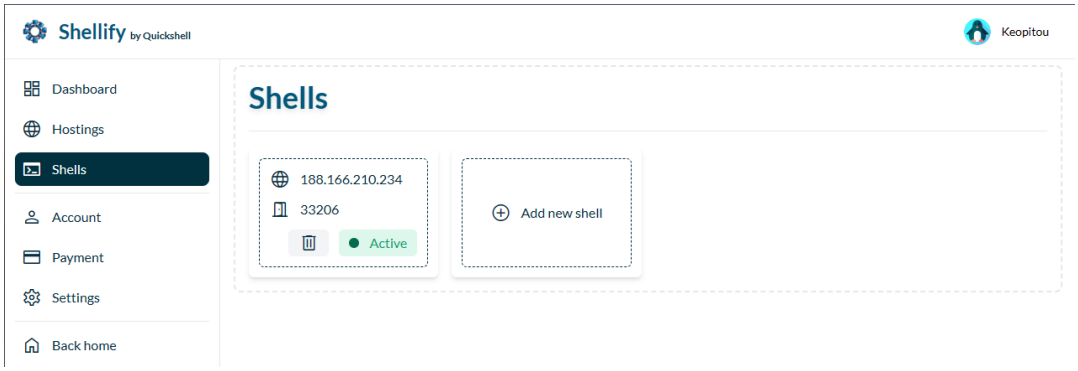


Figure 12 - Tab "Shells" of Shellify's User Dashboard

Figure 13 - Terminal Interface of Shellify's Shell

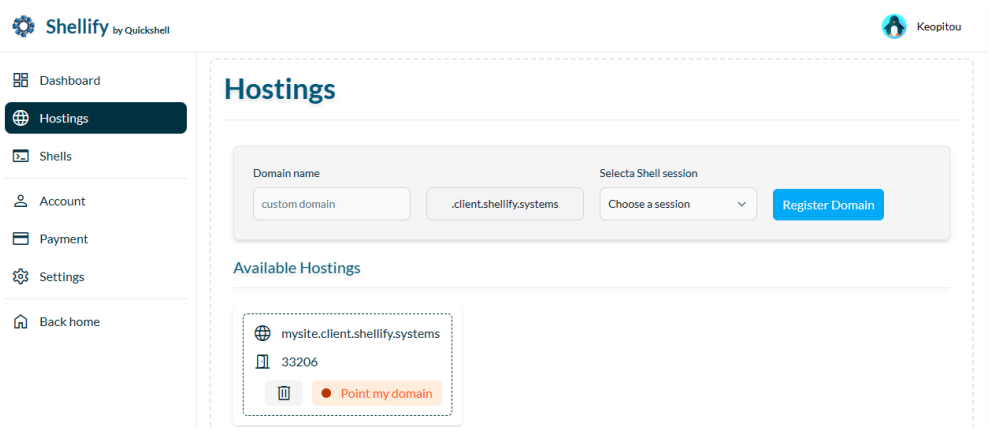


Figure 14 - Tab "Hostings" of Shellify's User Dashboard

c. Workflow

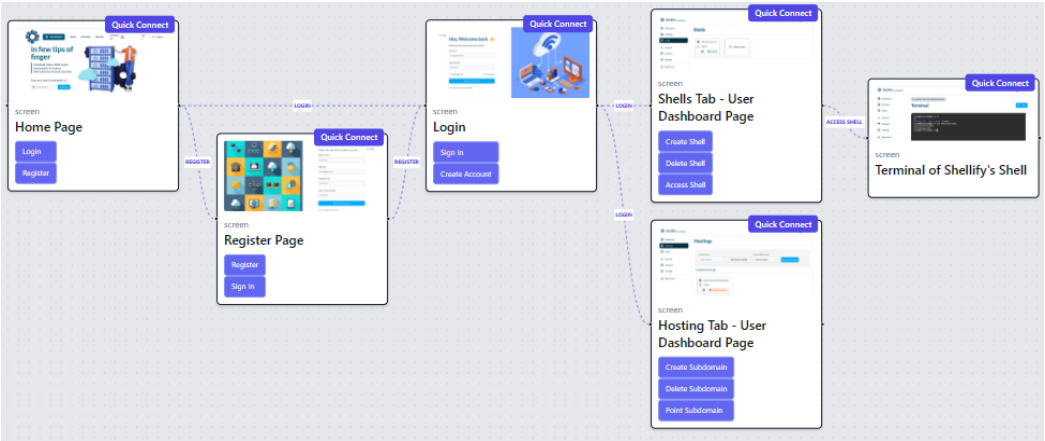
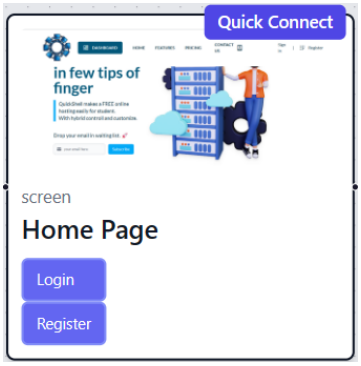
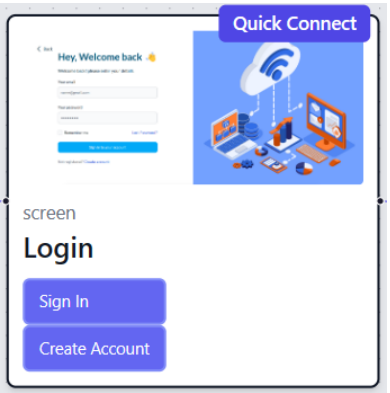
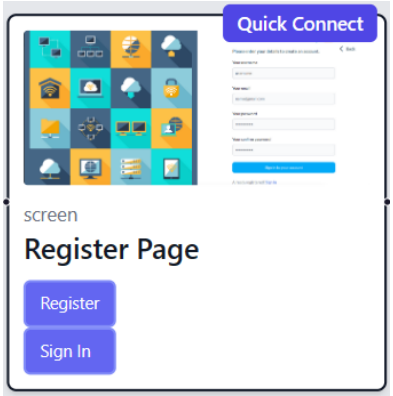
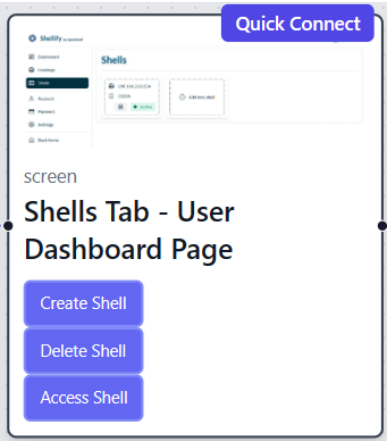


Figure 15 - Overview of Shellify Frontend Application Workflow

Due to the size of the image, please visit the **Indices** section to view the diagrams for each page.

Page	Figure	Description	Activity Diagram
Home Page	 *Figure 16 - Shellify Home Page*	The homepage of Shellify contains general information regarding the services provided by Shellify, Pricing…etc.	Appendix [1]

Login Page	 *Figure 17 - Shellify Login Page*	The login page allows users to enter their credentials and sign in to their Shellify account.	Appendix [2]
Register Page	 *Figure 18 - Shellify Register Page*	The register page allows new users to create a Shellify account.	Appendix [3]
Shells Tab	 *Figure 19 - Shellify Shells Tab*	The shell page provides users with access to a web-based shell session.	Appendix [4]

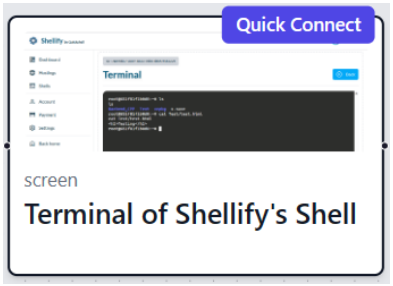
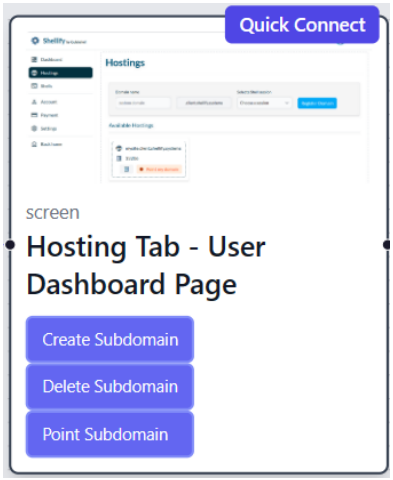
Terminal	 *Figure 20 - Shellify Shell's Terminal*	The terminal page provides users with a more advanced interface for interacting with the Ubuntu Linux environment provided by Shellify.	Appendix [5]
Hosting Tab	 *Figure 21 - Shellify Hosting Tab*	The Hosting Tab page allows users to manage the domain names associated with their Shellify account.	Appendix [6]

Table 6 - Shellify Frontend Web Application Pages

3. Problems Encountered During Design and Implementation

a. Technical Difficulties

During the development of Shellify, our team encountered several technical difficulties and unexpected issues. These challenges were faced by different teams at different stages of the project.

- In the first week, the Backend team faced challenges with using an open-source API for image randomizing, which might slow down performance. They decided to use static profiles for testing instead. The Backend team also needed to deploy a test version for the Cloud and had not yet communicated with the Frontend team.
- The Frontend team needed an API to sign in or sign up and found that xterm.js only accepts websocket as a communication method. They also discovered that xterm.js does not include any authentication mechanism and decided to use Socket.IO to establish a link between the client and our Linux server.
- The Cloud team faced challenges with container isolation and was waiting for the Web team to finish their endpoint. They also had to decide how to limit the container's

storage limit and were considering either changing the filesystem from ext4 to xfs or creating a docker volume and limiting the volume while modifying the endpoint code.

- In the second week, the Backend team encountered an issue with embedding the RSA Private Key directly in the code, which produced an invalid private key exception. The Frontend team faced challenges with routers requesting the HTTP address instead of HTTPS and with CORS blocking requests from Shellify's website to its websocket.
- The Cloud team was unable to send information to the database through the endpoints of the web due to a change in the API key for accessing the web's endpoints, which they were unaware of. They also faced challenges with Jenkins' jobs only running on specific nodes.

- In the third week, both the Backend and Frontend teams did not encounter any problems. However, the Cloud team was waiting for updates from other teams and faced an issue with unfiltered domain names being sent back to the web, causing an unknown name issue.

- In the fourth week, no problems were reported across all teams.

Our team worked collaboratively to address these challenges and find solutions or workarounds. By communicating effectively and adapting to unexpected issues, we were able to overcome these obstacles and successfully develop Shellify.

b. Design Trade-offs

During the development of Shellify, our team faced a design trade-off in the third week when we had to make a change to our product roadmap. Originally, we had planned to develop an admin dashboard, but we decided to change our focus to developing the Hosting Tab page instead in order to increase user functionalities.

This decision required us to make a trade-off between two different priorities: providing advanced features for administrators versus providing more functionality for users. By choosing to prioritize the development of the Hosting Tab page, we were able to provide users with more options for managing their domain names and improving their overall experience with Shellify.

However, this decision also meant that we had to deprioritize the development of the admin dashboard. This may have delayed the availability of advanced features for administrators and required us to make changes to our project plan and timeline.

This design trade-off illustrates the challenges that can arise when balancing different priorities and goals during the development process. By carefully considering the needs of our users and making informed decisions, we were able to make a trade-off that ultimately benefited the project and its users.

c. Unexpected Issues

During the development of Shellify, our team encountered several unexpected issues that arose during the project. These issues were not directly related to technical difficulties but rather were caused by external factors or changes in project requirements.

One unexpected issue that arose was in the second week when the Cloud team was unable to send information to the database through the endpoints of the web. This issue was caused by a change in the API key for accessing the web's endpoints, which the Cloud developers were unaware of. This highlights the importance of effective communication and coordination between different teams.

Another unexpected issue occurred in the third week when the Cloud team was waiting for updates from other teams. This issue highlights the interdependence of different teams and the need for effective coordination and collaboration.

These unexpected issues illustrate some of the challenges that can arise during a complex project such as Shellify. By being adaptable and responsive to changing circumstances, our team was able to address these issues and continue to make progress on the project.

d. Solutions and Workarounds

In response to these difficulties, we implemented various solutions and workarounds to overcome them and continue making progress on the project.

In the first week, when the Backend team faced challenges with using an open-source API for image randomizing, they decided to use static profiles for testing instead. This workaround allowed them to continue testing the performance of Shellify without being slowed down by the open-source API.

When the Frontend team discovered that xterm.js only accepts websocket as a communication method, they decided to use Socket.IO to establish a link between the client and our Linux server. This solution allowed them to overcome the limitations of xterm.js and provide users with a seamless experience.

In the second week, when the Cloud team was unable to send information to the database through the endpoints of the web due to a change in the API key for accessing the web's endpoints, they were able to resolve this issue by updating their API key. This solution allowed them to continue sending information to the database and avoid further delays.

Overall, these solutions and workarounds illustrate our team's ability to adapt and respond to challenges during the development of Shellify. By being resourceful and creative, we were able to overcome obstacles and continue making progress on the project.

4. Functional Group Organization

During the development of Shellify, our team was organized into four main functional groups: the Frontend Team, the Backend Team, the Cloud Engineer Team, and the Product Manager. Each of these groups played a critical role in the success of the project.

Role	Description	Who
Frontend Team	The Frontend team was responsible for developing the user interface and user experience of Shellify. This included designing and implementing the web pages and features that users interact with, such as the login, register, shell, terminal, and domain pages.	Channudam Ray
Backend Team	The Backend team was responsible for developing the server-side functionality of Shellify. This included designing and implementing the APIs and services that support the frontend features, as well as managing data storage and processing.	Chandara Yi
Cloud Engineer Team	The Cloud Engineer team was responsible for managing the Cloud infrastructure that powers Shellify. This included setting up and maintaining the servers, containers, and other cloud resources that are required to run the application.	Danet Reab Whatanak Kean
Product Manager	The Product Manager played a key role in guiding the overall direction of the project. This included defining the product roadmap, setting priorities, and making strategic decisions about the development of Shellify.	Keopitou Doung

Table 7 - Shellify's Functional Group Organization

Our team was organized in a way that allowed us to effectively collaborate and coordinate our efforts. By dividing responsibilities among different functional groups and working together towards a common goal, we were able to successfully develop Shellify.

Discussion, Conclusions, and Recommendations 2-4

1. Discussion

   During the development of Shellify, our team faced several challenges and obstacles. These included technical difficulties such as compatibility issues and unexpected bugs, as well as design trade-offs and unexpected issues. However, through effective communication and collaboration, we were able to overcome these challenges and implement solutions and workarounds.

   One of the key successes of Shellify was its ability to provide users with a simple and efficient way to access a Linux environment. By eliminating the need for local installations or configurations, Shellify was able to save users time and reduce technical barriers. This made it an attractive option for developers and system administrators who needed quick and easy access to a Linux environment.

   In addition to its practical benefits, Shellify also provided a valuable learning tool for students who wanted to practice programming in a Linux environment without investing in local hardware or software. This made it an attractive option for educational institutions and other organizations that wanted to provide their students or employees with access to a Linux environment.

   Shellify has successful achieved its goals and providing value to its users. Through effective collaboration and problem-solving, we were able to overcome challenges and deliver a powerful tool that met the needs of our target audience.

2. Conclusions

   In conclusion, our project to develop Shellify was successful in delivering a powerful and user-friendly tool for accessing a Linux environment through a web shell session. Our team was able to overcome technical and design challenges through effective collaboration and problem-solving.

   One of the key successes of Shellify was its ability to reduce technical barriers and save users time. By eliminating the need for local installations or configurations, Shellify provided a convenient and efficient way for developers and system administrators to access a Linux environment.

   Shellify also proved to be a valuable learning tool for students who wanted to practice programming in a Linux environment. Its user-friendly interface and easy accessibility made it an attractive option for educational institutions and other organizations.

   Overall, our project demonstrated the potential of web-based tools to provide users with easy access to powerful computing resources. We are proud of what we have accomplished with Shellify and are confident that it will continue to provide value to its users in the future.

3. Recommendations

   Expand User Functionality: One recommendation for future work on Shellify could be to continue expanding the functionality available to users. This could include adding new features or improving existing ones to provide users with even more options for accessing and interacting with a Linux environment.

Improve Performance and Scalability: Another recommendation could be to focus on improving the performance and scalability of Shellify. This could include optimizing the underlying infrastructure and architecture to support larger numbers of users and more demanding workloads.

Enhance Security and Privacy: A third recommendation could be to enhance the security and privacy features of Shellify. This could include implementing stronger encryption and authentication mechanisms, as well as providing users with more control over their data and how it is used.

Collaborate with Educational Institutions: A final recommendation could be to collaborate more closely with educational institutions to provide students with access to Shellify as a learning tool. This could include developing custom curricula or lesson plans that incorporate Shellify into programming or computer science courses.

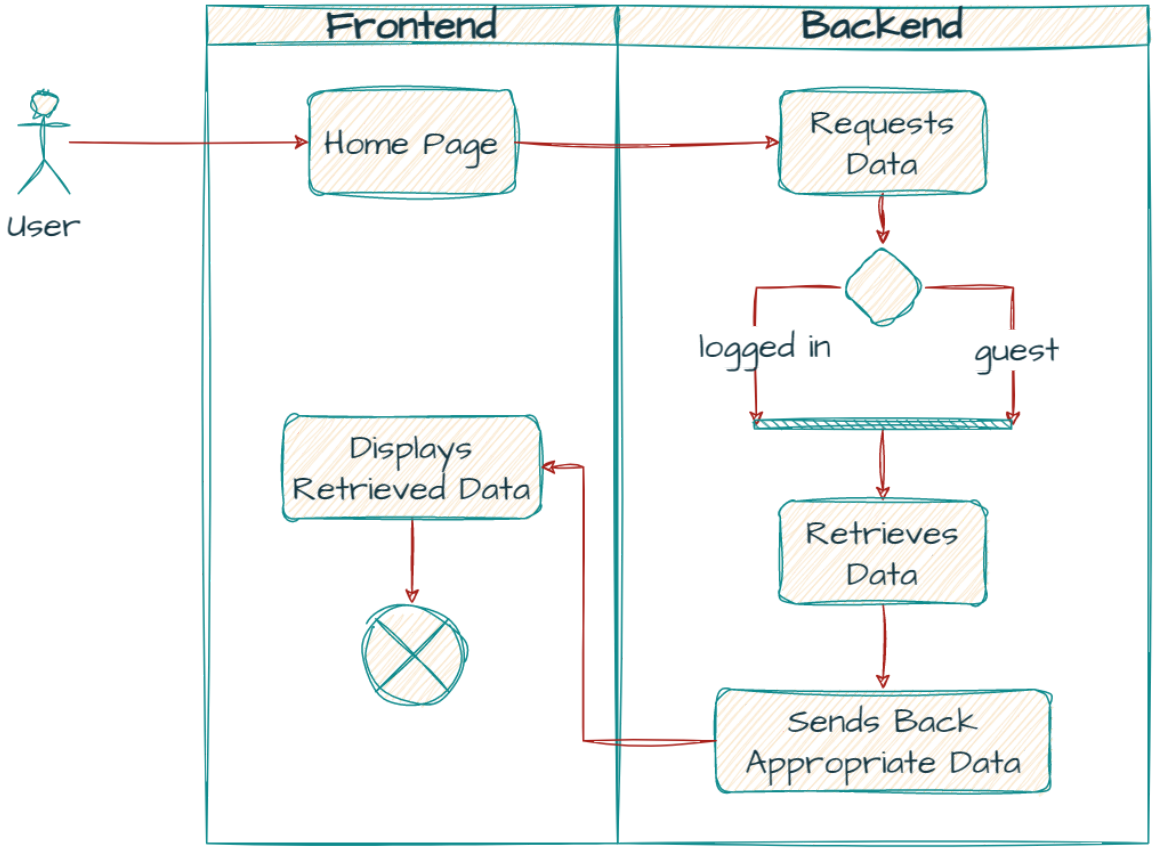
These recommendations are based on our experience developing Shellify and our understanding of its potential for future growth and improvement. By continuing to expand user functionality, improve performance and scalability, enhance security and privacy, and collaborate with educational institutions, we believe that Shellify can continue to provide value to its users in the future.

References

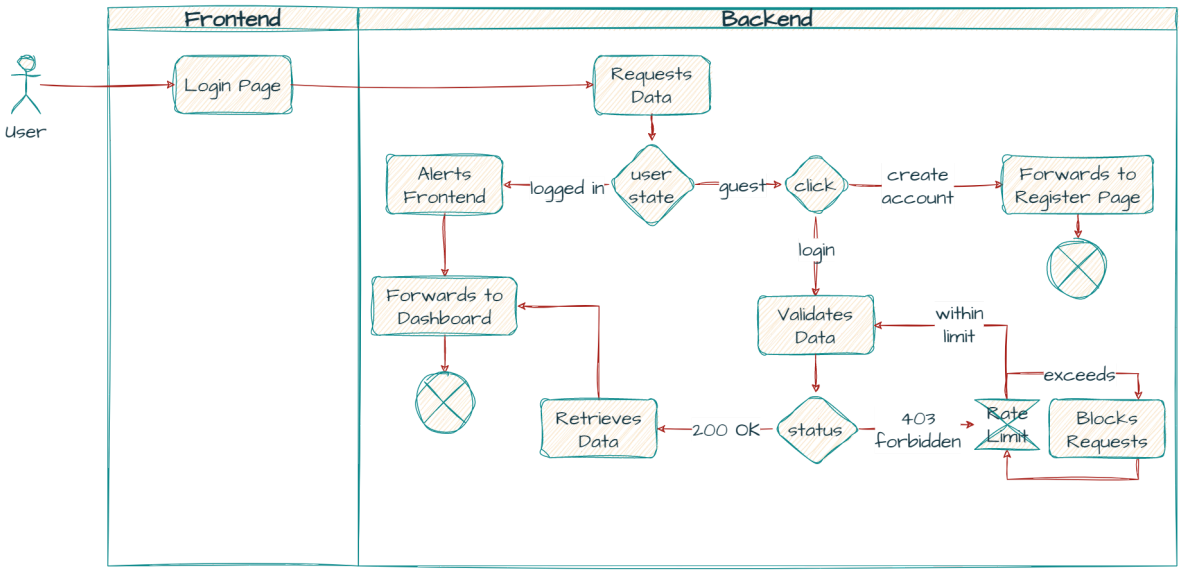
1. Evans, D. (2014). The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. Cisco Internet Business Solutions Group.
2. Mattern, F., & Floerkemeier, C. (2010). From the Internet of Computers to the Internet of Things. In From active data management to event-based systems and more (pp. 242-259). Springer.
3. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7), 1645-1660.
4. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. Computer networks, 54(15), 2787-2805.
5. Mazhelis, O., Tyrväinen, P., & Koskinen, K. T. (2017). IoT Platform Architectures—A Review. Sensors, 17(8), 1764.
6. Wang, L., Tafreshi, V. H. F., & Sarkar, N. I. (2018). Internet of Things (IoT) technologies, applications, challenges and solutions: A review. In 2018 IEEE 4th World Forum on Internet of Things (WF-IoT) (pp. 375-380). IEEE.
7. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376.
8. Shrouf, F., Ordieres-Meré, J., & García-Sánchez, A. (2014). Smart factories in Industry 4.0: A review of the concept and of energy management approaches. In Proceedings of the IEEE international conference on industrial engineering and engineering management (pp. 697-701). IEEE.
9. Mell, P. & Grance, T. (2011). The NIST definition of cloud computing. National Institute of Standards and Technology.
10. Gupta, S., Singh, A., & Chana, I. (2018). Cloud computing: A green computing approach. Sustainable Computing: Informatics and Systems, 19, 174-180.
11. Yigitbasi, N., Ozonat, K., & Isik, C. (2014). A survey and classification of energy-aware management techniques for cloud computing systems. Journal of Supercomputing, 68(2), 683-707.
12. Boothroyd, G., & Dewhurst, P. (2010). Product design for manufacture and assembly. CRC Press.
13. Ulrich, K., & Eppinger, S. (2017). Product design and development. McGraw Hill Education.
14. Chauhan, A. K., & Singh, K. (2018). Manufacturing of Shell for Pressure Vessels: A Review. International Journal of Engineering Science and Computing, 8(6), 14969-14974.

Appendices

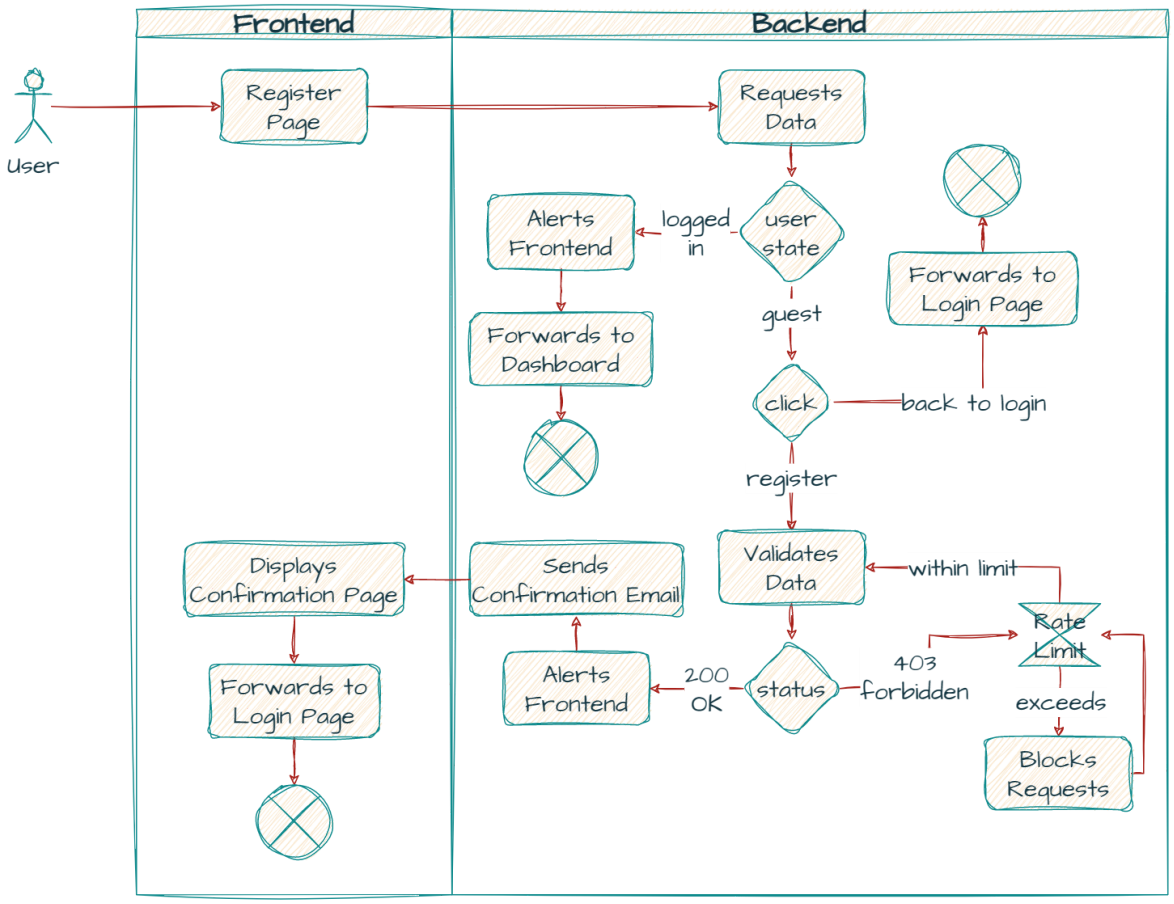
1. Home Page



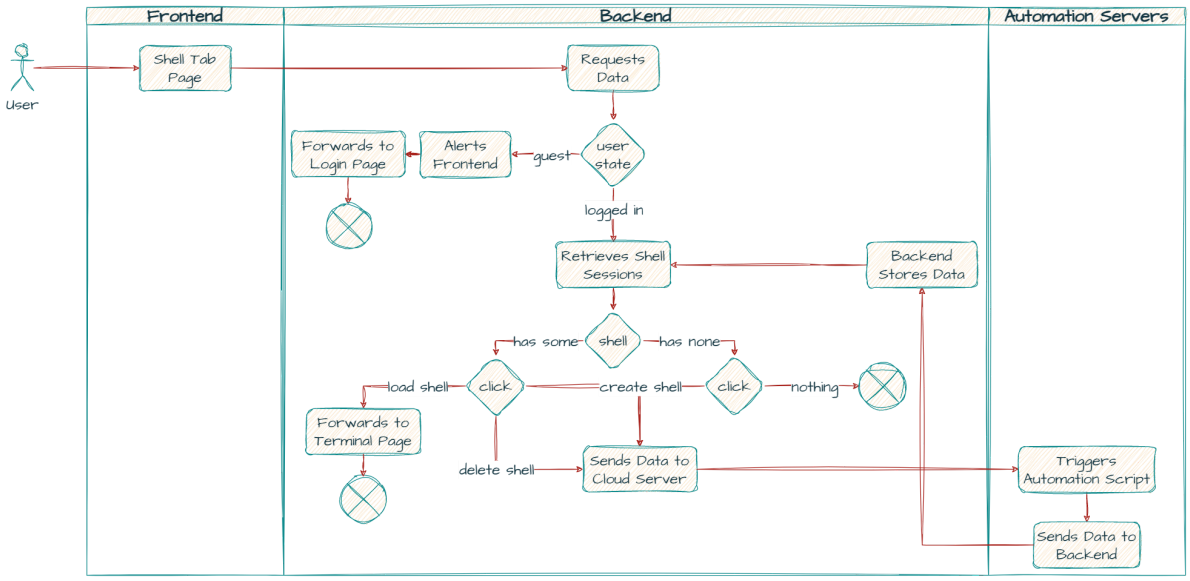
2. Login Page


Frontend
Backend
User
Login Page
Requests Data
Alerts Frontend
logged in
user state
guest
click
create account
Forwards to Register Page
login
Forwards to Dashboard
Validates Data
within limit
exceeds
Retrieves Data
200 OK
status
403 forbidden
Rate Limit
Blocks Requests

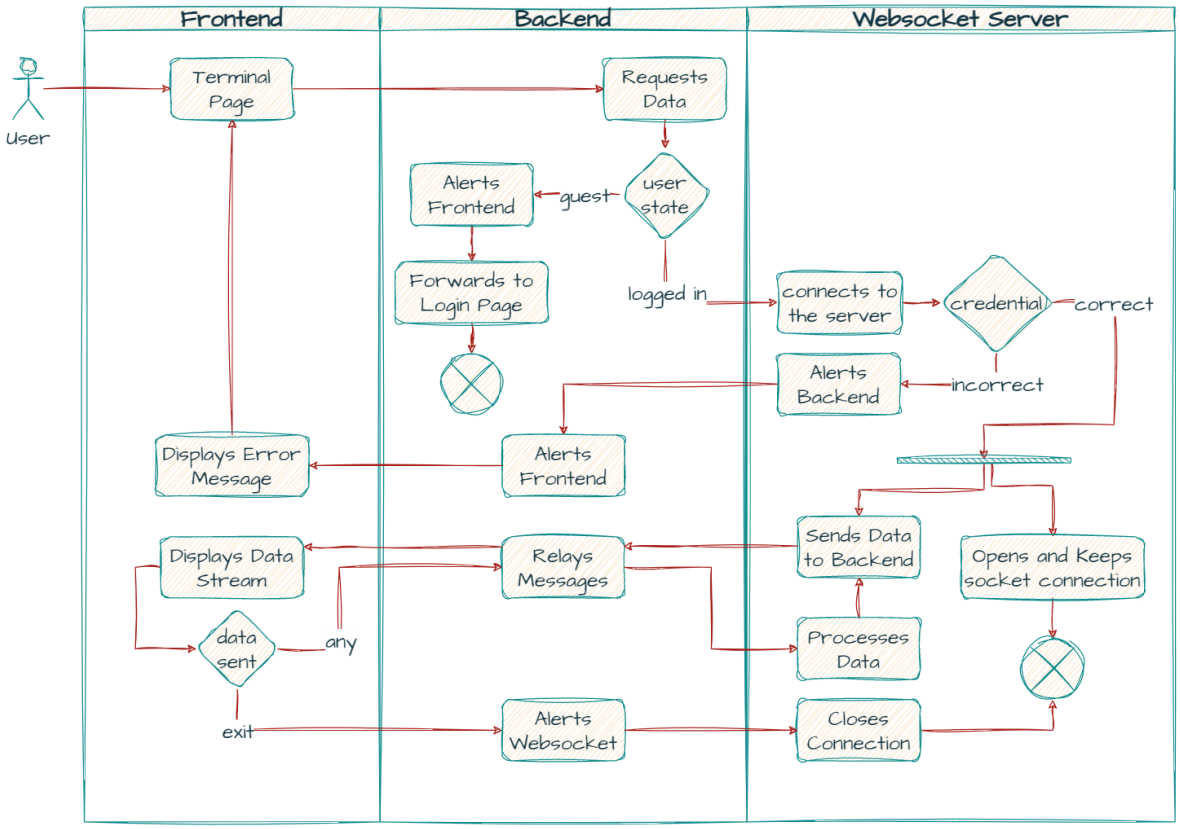
3. Register Page



4. Shells Tab


Frontend
Backend
Automation Servers
User
Shell Tab Page
Requests Data
user state
guest
Alerts Frontend
Forwards to Login Page
logged in
Retrieves Shell Sessions
Backend Stores Data
shell
has some
has none
click
load shell
create shell
click
nothing
Forwards to Terminal Page
delete shell
Sends Data to Cloud Server
Triggers Automation Script
Sends Data to Backend

5. Terminal


Frontend
Backend
Websocket Server
User
Terminal Page
Requests Data
Alerts Frontend
guest
user state
Forwards to Login Page
logged in
connects to the server
credential
correct
Alerts Backend
incorrect
Displays Error Message
Alerts Frontend
Sends Data to Backend
Opens and Keeps socket connection
Displays Data Stream
Relays Messages
data sent
any
Processes Data
exit
Alerts Websocket
Closes Connection

6. Hosting Tab