A

Project - I Report
on

SHELLINABOX

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Information Technology

(Session 2018-2019)

Guided By -
Mr Amol Saxena

Guide Name(s)
Mr Amol Saxena
(Head of Department Dept. of Information Technology)

Submitted by-
VijayaNandwana-PCE15IT062
Riya Gupta –PCE15IT046
Achal Sharma-PCE15IT002
VII Semester

DEPARTMENT OF INFORMATION TECHNOLOGY
POORNIMA COLLEGE OF ENGINEERING, JAIPUR
RAJASTHAN TECHNICAL UNIVERSITY, KOTA

(November, 2018)

A

Project - I Report
On

SHELLINABOX

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Information Technology

(Session 2018-2019)

Guided By -
Mr Amol Saxena

Guide Name(s)
Mr Amol Saxena
(Head of Department Dept. of Information Technology)

Submitted by-
VijayaNandwana-PCE15IT062
Riya Gupta –PCE15IT046
Achal Sharma-PCE15IT002

VII Semester

DEPARTMENT OF INFORMATION TECHNOLOGY
POORNIMA COLLEGE OF ENGINEERING, JAIPUR
RAJASTHAN TECHNICAL UNIVERSITY, KOTA

(November, 2018)

Candidate's Declaration

I/We hereby declare that the work, which is being presented in the **Project - I Report**, **Shellinabox Project-I** in partial fulfillment for the award of degree of **Bachelor of Technology** in **Information Technology**, and submitted to the Department of **Information Technology, Poornima College of Engineering**, **Jaipur** is a record of my own work/investigations carried under the guidance of **Mr Amol Saxena** Department of **Information Technology, Poornima College of Engineering.**

We have not submitted the matter presented in this Project-I Report any where for the award of any other Degree.

[Name(s), Registration Number(s) and Signature of Student(s)]

(1) Vijaya Nandwana	PCE15IT062
(2) Riya Gupta	PCE15IT046
(3) Achal Sharma	PCE15IT002

[Counter Signed by]

Mr. Amol Saxena
Dept. of Information Technology
Head of Department

Date: _________
Place: ________

POORNIMA
COLLEGE OF ENGINEERING

DEPARTMENT OF INFORMATION TECHNOLOGY

Date:

CERTIFICATE

This is to certify that **Project - I** report titled **Shellinabox** has been submitted by **Vijaya Nandwana [PCE15IT062], Riya Gupta [PCE15IT046], Achal Sharma [PCE15IT002]** in partial fulfilment for the award of the Degree of **Bachelor of Technology** in **Information Technology** during the session 2018-19, Odd Semester.

The project work is found satisfactory and approved for submission.

(Mr. Amol Saxena) (**Mr. Shirish Nagar)**
HOD - IT Project Coordinat

Acknowledgement

A project of such a vast coverage cannot be realised without help from numerous sources and people in the organization. We are grateful to our respected Campus Director **Dr. Mahesh Bundele** & all those who helped us directly or indirectly in our endeavour.

We are also grateful to **Mr. Shirish Nagar**, assistant professor, IT Dept., PCE for their kind support who encouraged us and provided us various resources and facilities in department to complete our project.

We owe our project to Mr. Shirish Nagar, assistant professor, IT Dept., who gave us his time, precious guidance, and the tools to develop our project efficiently.

We wish to express our sincere thanks and gratitude to our guide **Mr Amol Saxena** for his affectionate and undaunted guidance as well as for his morale boosting encouragement with worthy suggestions and support.

We also thank our HOD**, Mr Amol Saxena**, for guiding and supporting us throughout our project.

Last but not least, acknowledgment will not be over without mentioning a word of thanks to all our friends & colleagues who helped us directly or indirectly in all the way throughout project preparation.

Vijaya Nandwana- PCE15IT062

Riya Gupta - PCE15IT046

Achal Sharma- PCE15IT002

B. Tech VII Semester

Information Technology

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Abstract

Services are an important part of the IT industry. Investing in services increase the operation cost of industries and organization. Organizations have to pay for personal hardware and software. A lot of time is wasted on maintenance of resources rather than focusing on the true functionality of the organization.

The project is a Cloud and hadoop service providing application. The purpose of this project is that people will no more have to invest upon personal hardware and software resources. The automation world Web-app will provide services to client and will save their time to a large extent.

Now-a-days Industries, Professionals and Students spend lot of their money on hardware and software. This increases the over-all cost of running the business. Thus the main aim of this system is to assist people in all pay as per use. By using this system the cost and time taken to manage resources will reduce and the user does not have to surf the web or pay for long periods to use resources. The user can also solve his queries by contacting other service provider on the separate built-in platform for general queries.

Chapter -1
Introduction to the Project

1. Introduction

1.1. Purpose

This document entails the detailed description of the project, “Shellinabox”. It Specifies and explain every module, their functionality, significance & relevance with the project. The main objectives of the project are:

1.1.1. Application will help the users to use different services as per the requirement.

1.1.2. Application will help the service providers to earn using their hardware and software resources with was previously of no use.

1.1.3. It helps users to only pay proportional to the amount of resources used.

1.1.4. The Tenant can focus more on their functions rather than maintaining infrastructure or other working environment.

1.2. Origin of the Problem

Services are an important part of the IT industry. Investing in service increases the operation cost of industries, organizations and individuals. Organizations have to pay for personal hardware and softwares. A lot of time is wasted on maintenance of resources rather than focusing on the true functionality of the organization

1.3. Definition of the Problem

The traditional method was not cost efficient. Every user invested in buying resources such as hardware components for infrastructure management or licensed softwares for different utilities. We would then have to pay to buy the resource eventhough it is of use only for a short duration of time. This method is not cost and time efficient. Also, it is an error prone process and needs much human resources thus complicating the task.

2. Methodology Used

2.1. Agile Methodology

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Each iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the iteration a working product is displayed to the customer and important stakeholders.

2.2. Why Agile

Agile development methodology provides opportunities to assess the direction of a project throughout the development lifecycle. This is achieved through regular cadences of work, known as sprints or iterations, at the end of which teams must present a potentially shippable product increment. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as “iterative” and “incremental.”

In waterfall, development teams only have one chance to get each aspect of a project right. In an agile paradigm, every aspect of development — requirements, design, etc. — is continually revisited throughout the lifecycle. When a team stops and re-evaluates the direction of a project every two weeks, there’s always time to steer it in another direction.

The results of this “inspect-and-adapt” approach to development greatly reduce both development costs and time to market. Because teams can develop software atthe same time they’re gathering requirements, the phenomenon known as “analysis paralysis” is less likely to impede a team from making progress. And because a team’s work cycle is limited to two weeks, it gives stakeholders recurring opportunities to calibrate releases for success in the real world. Agile development methodology helps companies build the right product. Instead of committing to market a piece of software that hasn’t even been written yet, agile empowers teams to continuously re-strategies their release to optimize its value throughout development, allowing them to be as competitive as possible in the marketplace. Development using an agile methodology preserves a product’s critical market relevance and ensures a team’s work doesn’t wind up on a shelf, never released.

Chapter -2
Synposis

Project Synopsis

1	Name of the Project	Shellinabox
2	Objective Vision	To maintain a shell for a user on which a user can have all cloud services like Saas,Paas,etc and also upload there data on cloud and analyse it using Hadoop and spark.We also provide the 3 authentication in our project like face recognition ,speech recognition and userid.
3	Users of the System	a. Everyone
4	Functional Requirements	1. A system for any type of users who need some kind of software or cloud or Bigdata services. 2. A system for any type of users who want to store there data securely and analyse it regularly. 3. Give storage to the user according to need of user. 4. Gives a personal operating system using containers in docker. 5. Gives the detail about all past analysis of data from any particular account. 6. Use NLP. 7. Own platform for coding. 8. Admin can monitor every activity which is performed by system.
5	Non-functional requirements	1. Highly secure with different authentication. 2. Access this shell 24x7. 3. Flexible service based architecture will be highly desirable for future extension

		4. Upload and analyse your secure data.
6	Optional features	1. Showing the graphs of analysed data. 2. Face recognition for unlocking the user account. 3. Speech recognition for unlocking the user account. 4. Give we a online music player.
7	User interface priorities	1. Professional look and feel 2. Use of Python-CGI,HTML,CSS at least with all registration forms 3. Browser testing and support for IE, Mozilla and Firefox. 4. Use of Graphical tool like Jupyter notebook to show data of client. 5. Reports exportable in .XLS, .PDF or any other desirable format
8	Reports	1. System will generate when user demands come.
9	Other important issues	1. No need to type ID and password all time for login. 2. It also understands human language and work according to it.
10	Team Size	3
11	Technologies to be used	BigData, spark, cloud, Linux, python, machine learning, deep learning, Ansible , Dockers.
12	Tools to be Used	1. Virtual Box 2. Jupyter notebook
13	Final Deliverable must include	1. Online or offline help to all users. 2. Project archive (.tar) with source code 3. Database backup 4.Complete Source code

Chapter -3

Software Requirement Specification

1. Introduction

1.1 Purpose

This project ‘Shellinabox’ aims to provide a shell to a user in which the user gets its personal cloud service. It is planned to satisfy everyone’s expectations. The Shellinabox is intended to provide a quick, easy and user-friendly cloud based services like (Software as a service) SaaS , (Platform as a service)PaaS , (Container as a service)CaaS , (Infrastructure as a service) IaaS , (Storage as a service)StaaS and data analytics which is done through hadoop services. The system should be designed so that management time is minimized and the product is available to clients at a very economical price. All the Services are available under a single portal. We are giving these services through a web page which has a authentication page in which we provide three way of authentication that is Face detection, speech recognition and user-id and password. After sign up its user choice that either they want email or speech or face detection for sign in there cloud.In this application user have to give some charge for using services on hour base. Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions). Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign

1.2 Feasibility

The web interface reduces the effort and time of the clients. People spend a lot of time in finding an appropriate platform which provide secure services. This web interface serves their purpose by giving them better user experience without much investment. The web interface will be made accessible to Enterprise, Professionals and Students. The data deployed by the users is secure and have backups also there is no fear of losing confidential data. The services are made accessible to the targeted clients using Web-UI Techniques which makes it easier for the clients to access these services. The user will be able to access the shell within very low cost and is available to user in any time required.

2. Functional /Nonfunctional Requirements

2.1 Functional Requirements

2.1.1 **Container:**

Containers as a service (CaaS) is a cloud service that allows software developers to upload, organize, run, scale, manage and stop containers by using a provider's API calls or a web portal interface.

2.1.2 **Platform**:

Platform as a Service (PaaS) or platform base service is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity

2.1.3 **Storage**:

Storage as a service (SaaS) is a business model in which a company leases or rents its storage infrastructure to another company or individuals to store data.

2.1.4 **Software**:

Software as a service (SaaS) is a software distribution model in which a third-party provider hosts applications and makes them available to customers over the Internet.

2.1.5 **Infrastructure** :

Infrastructure as a service (IaaS) is a form of cloud computing that provides virtualized computing resources over the internet.

2.1.6 **Highly Automated** :

The Web-App should promote "One Click Feature" which focuses on maximum automation. The portal should be designed in a way that reduces the efforts of the user. The user should not indulge in any kind of complexity while dealing with it.

2.1.7 **24x7 Availability :**

The server should not loose connectivity with its clients. It should be available 24x7 for its users.

2.1.8 **Secure client details :**

The details shared by the user with the server should be considered confidential and should not be misused under any circumstances.

2.1.9 **BackUp of Data :**

The server should keep the backup of the data in case of any loss.

2.1.10 **Hadoop Services:**

Hadoop is a big data analytics framework that stores and analyzes data in the cloud using Hadoop. Users don't have to invest in or install additional infrastructure on premises when using the technology, as HDFS is provided and managed by a third-party vendor. The open source Hadoop big data analytics framework allows large, unstructured data sets to be analysed. Hadoop's storage mechanism, the Hadoop Distributed File System, distributes these workloads across multiple nodes so they can be processed in parallel. Mapper Reducer will be used for the processing of the data as per the user requirement. Spark and Hive is also used for the analysis of data which the user upload .

2.2 Nonfunctional Requirements

Performance Requirements

a. Good User Interface:

The Web interface should be designed in a way that the user finds it easy to deal with. The interfacing should be made with maximum level of abstraction i.e. showing only the necessary details. The user should be free from any kind of uploading and downloading process to access the Server side. The user who provides us with the true information will be able to access it.

b. **Fast Deployment**:

   There should be fast deployment of data on the server. The client should not have to wait due to bad processing speed.

c. **Good User Experience**:

   The Web interface will be designed to minimize the efforts of the user. It should not ask for user input at every step of processing.

d. **Reliable**:

   The user data will be confidential and will not be misused under any condition.

Safety Requirements

1. **Privileged Users**:

   The clients of server will be provided privileged rights for accessing its personal data and containers. No user is allowed to read or edit the information of the other users. The user will not face "Permission Denied" errors.

2. **Active Database**:

   The database will contain the details of the most recent activities of the clients, the data stored by the client and the analyzed data.

3. **Data Integrity**:

   The user data will be confidential at every point of time. As for every single user a Container is launched by his name which will be accessed by the user only.

Security Requirements

1. **Secure Data:**

   The Web-App uses Password Based Authentication.

2. **User Management:**

   All the information about the user is saved and secure.

Software Quality Attributes

1. **24x7 Availability**:

   The cloud will be available 24x7 for its clients.

2. **Availability of new and updated Software**:

   It will be the responsibility of the server to entertain its clients with latest up to date software.

3. **Ease Of Data Analytics:**

   The data of the user is analyzed according to the needs and requirement.

2.3 Technical Requirments (Hardware /Software)

Operating Environment

Red Hat Enterprise Linux 7 beta is ready for whatever infrastructure choices we make, efficiently integrating with other operating environment, authentication, and management systems. Whether your primary goal is to build network-intensive applications, massively scalable data repositories, or a build-once-deploy-often solution that performs well in physical, virtual, and cloud environments, Red Hat Enterprise Linux 7 beta has functionality to support the project. We'll have better insights into what the system is doing and more controls to optimize it, with unified management tools and system-wide resource management that reduce the administrative burden. Container-based isolation and enhanced performance tools allow we to see and adjust resource allocation to each application. And, of course, there are continued improvements to scalability, reliability, and security. Developers and Dev-Ops: Red Hat Enterprise Linux 7 beta has more than just operating system functionality. It provides a rich application infrastructure with built-in mechanisms for security, identity management, resource allocation, and performance optimization. In addition to well-tuned default behaviours, we can take advantage of controls for application resources so we don't leave performance up to chance. Red Hat Enterprise Linux 7 beta includes the latest stable versions of the most in-demand programming languages, databases, and runtime environments.

Hardware Interfaces

Submit jobs with the associated deadline, cost, and execution time .Query the cluster to establish the current cost per unit time for submitting new jobs .Monitor the status of submitted jobs. Cancel jobs submitted by user. Check his usage history. The client can get connected to user whenever required according to the needs.

Software Interfaces

File Systems: Red Hat Enterprise Linux now supports XFS file systems that are up to 500TB in size. The previous support limit was 100TB.

• Ext4 supports a file system that is 50TB in size, up from 16TB.

•The Red Hat Enterprise Linux PNFS client now supports all commercially available server lowest types.

DOCKER 1.8

Docker is a computer program that performs operating-system-level virtualization, also known "containerization". It was first released in 2013 and is developed by Docker, Inc.

Docker is used to run software packages called "containers". Containers are isolated from each other and bundle their own tools, libraries and configuration files; they can communicate with each other through well-defined channels. All containers are run by a single operating system kernel and are thus more lightweight than virtual machines. Containers are created from "images" that specify their precise contents. Images are often created by combining and modifying standard images downloaded from public repositories.

ANSIBLE 1.6

Ansible is open source software that automates software provisioning, configuration management, and application deployment. Ansible connects via SSH, remote Power Shell or via other remote APIs.

The design goals of Ansible include:

• Minimal in nature. Management systems should not impose additional dependencies on the environment.

• Consistent. With Ansible one should be able to create consistent environments.

• Secure. Ansible does not deploy agents to nodes. Only Open SSH and Python are required on the managed nodes.

• Highly reliable. When carefully written, an Ansible playbook can be idempotent, in order to prevent unexpected side-effects on the managed systems.[16] It should be noted, however, that it is entirely possible to have a poorly written playbook that is not idempotent.

Security: With firewall, a firewall does not have to be stopped in order to change its rules. This increases the security of the system by eliminating vulnerability and adding the ability to respond to threats by quickly activating new rules. In addition to dynamic configuration capabilities, firewall supports a powerful rules language that simplifies configuring firewalls.

Communications Interfaces

The product is designed to provide better user experience. For Web UI Html5, CSS4, Bootstrap4, JAVA Script is used to improve the look and feel of the Web interface. The Web interface uses python-CGI to interact with the clients. Python-CGI makes it easier for the clients to interact with the server model.

The network Server Communication Protocols like HTTP, SSH, NFS, ISCSI, Docker-API are used to communicate with the clients.

3 System Features

Module 1: System features of Module1

HIGH PRIORITY

Module 1: Python Audio

Using Python Text to Speech for speech. This will be aid to the blinds. Pyttsx3 is a good text to speech conversion library in python. **Pyttsx3 library now works for both python2 and python3 and is also cross-platform.**

HOW TO APPLY VOICE BIOMETRICS TECHNOLOGY

If our application has passed the third barrier of having a viable business case, we may now move on to consider the technical feasibility of implementing Voice Biometrics within your environment.

First, Get Audio Voice Biometric systems analyze voice recordings; therefore, the first step in implementation knows how we will get voice samples of the person for whom we are creating a voiceprint. We need a way to capture audio of the person we want to authenticate and/or identify such that we have only that voice and not a mix of that voice and other voices or sounds.

The most common method to capture audio for a voiceprint is over the telephone. When we are speaking on the phone, your voice is obviously going over the telephone line, whether a mobile phone, landline phone, or VoIP phone, and it can be recorded using a range of equipment made for this purpose, possibly without we even knowing. If we have an IVR, the IVR will likely be able to record audio that is then passed to a Voice Biometric engine.

A second way is through a web browser on a computer or tablet or phone equipped with a microphone, such as every laptop computer made today. The web browser serves the purpose of being a telephone. Because nearly every modern laptop has a microphone, capturing audio through the computer is practical for many applications where the user will be near or on a computer. Likewise, voice capture is also easily done on a mobile device with a native application.

And lastly, any other way that we can get audio through a microphone and record in a format suitable for a voice biometric engine. For example, one of our customers records people speaking during a spoken language test and uses that audio to create voiceprints. Another example is using stored recordings in your call centre. Regardless of how we get the audio, note that we want to only capture the audio of the desired speaker. If we include other audio, then we will have to do extra work to separate the audio before using it with a voice biometric engine.

Then Determine User Experience Once we have a means of capturing audio, your next step is to determine what, if anything, we will be asking the person of interest to do in order to capture that person’s voice and send to the Voice Biometric engine in order to make a voiceprint. This step is called “enrollment.” Once enrolled, we can then authenticate the user by capturing a new voice sample that is compared against the voiceprint from enrollment. And we can compare that new sample against all other voiceprints in the database to determine the identity of the speaker. Several options exist for enrollment and verification. Each has pros and cons. The following section explains the options.

WHAT ABOUT LANGUAGES?

Language is fortunately not that big of an issue. Voice Biometrics technology uses the way people make sounds as a way to reveal unique characteristics. Because the voice biometric engine measures how rather than what we are speaking, it does not care about language. However, to tune the voice biometric engine to get the best results means taking into account that different languages use different sounds. Spanish, for example, has a different set of phonemes (and actually a fewer number with some overlap) than English. For this reason, we separate languages as well as use cases, resulting in use-case-language combinations.

Suppose for example, we have a population of users in the US who need to use biometrics in both Spanish and English. We would set you up with one account for Random Number English and a separate account for Random Number Spanish as the way to optimize results.

We currently support 35 languages in Production. In the unlikely event we do not have your language, we will work with we to capture a large enough sample of your language to set up your language on our voice biometric engine. We call the capture of voice samples a Data Collection Study. For each new language, we like to gather voice samples from 75 males and 75 females. That is usually sufficient to give us an excellent start. We will find that some biometric companies require substantially more data collection. This is due to the fact that their underlying algorithm requires more data for training, often on the order of a few thousand people.

WHAT ABOUT IMPLEMENTATION PROCESS?

If we made it this far, we’re ready to learn how to implement a solution. Three possibilities exist:

We directly access our Restful APIs. This option is appropriate for companies who wish to embed voice biometrics into their own product or want to include voice biometrics into a business process. For this option, skip over to our Developer Page to learn more about our Restful APIs and programming model. Check out our mobile app development page to learn how native mobile applications also can leverage our Restful APIs. Request a Free Trial to let developers work with our APIs. Most developers can get familiar with our API in minutes using our online tools and up and running in a few days using their favourite development environment.

We deploy an IVR-based system for you and integrate with your business process. We offer a cloud or premises-based IVR system to save you the hassle of implementing a system. For example, see our resource page on Multi-Factor Authentication to learn how to leverage our IVR

Quickly and easily in your business process. Request a Free Trial to test out our complete IVR and voice biometric solution in either English or Spanish. Changing our IVR application to your language of choice requires only a small fee for new prompts.

Finally, if we are looking for a turn-key solution where voice biometrics is an element within a broader security solution, we will connect you with one of our partners. Simply fill out a contact form and make note of this in your request. We will participate with our partner to design and implement a solution appropriate for you.

Module 2: Registration (Database)

2.1 Entire details of all the users and their logs will be maintained in database. It will be managed by the Database Administrator.

1. User Enters Their Credentials

When the user lands on a website, she must first request access by logging into her account. Generally, the login process will require a username and password; however, there may be additional steps the individual will take to verify her identity.

2. Credentials Are Sent to the Authentication Server

Once the user presses "Login," the credentials are sent to a local operating system or authentication server. The operating system or server is where every authorized user's information—credentials and permissions—is stored.

3. System Looks for a Match on the Server

The system will compare the information entered by the user with all the credentials on file. If no match is found, the user will be notified that her credentials couldn't be found and asked log in again. When a match is found, the system moves to the next step.

4. System Authorizes the User and Access Is Granted

When a match is found, the system will authorize the user by sending the permissions that define what the user can see and do back to the website, thus granting the user access to her account.

Module 3: Face Recognition

STEP-BY-STEP FACIAL RECOGNITION WITH KAIROS

The market has a great variety of facial recognition providers to meet all of your needs. Kairo is among the best of them, and we decided to give it a try. We had to make a choice: either use a

Kairos SDK for iOS apps or opt for the Kairos API and implement the facial recognition functionality ourselves. We chose the Kairos API for our project, because it was a much cheaper solution.

The principles of facial detection and recognition are virtually identical to those of OpenCV. So, there is no need to repeat the basics. Here's how they work together with Kairos.

During registration, the app accesses the device's camera to obtain images of the user. We require nine images to store in the database.

The image-enroll functionality in Kairos stores the images in a created gallery. The images require an ID (i.e. the name of the user). The gallery also requires a name for its identification. Parameters used: image: a publicly accessible URL, an uploaded file or a base64-encoded photo subject_id: identifier for the face gallery name: identifier for the gallery.

Recognition

The library uses neural networks to learn the features of the face and create a face template. Each time authorization is required, it compares a template of the image with the stored templates. The library may or may accept that the person is the same as the one in the submitted images.

Interpretation of results

In cases where the inputted image and the stored template match, the app displays the subject's ID and grants access to the app. The program returns the result with two parameters: name and probability (or the level of confidence). The threshold of recognition is 80% confidence by default. This means that if the confidence level is below 80%, the API will not consider it a match. Optionally, we can change this parameter to be higher or lower. However, the threshold of 80% is optimal, because if we set it higher, we run the risk of false rejects.

Response to rejection

If the inputted image doesn't match the data stored in the database, the images and text in the app are blurred.

Problems And Solutions

The main problem we faced had to do with memory allocation. Because of the constant access to the device's RAM for image processing, the device overheated. The solution was to connect an OpenGL library that handles images and multithreading and that gives access to the CPU directly.

We have yet to solve the problem with image recognition of when the user tries to take a photo when bright sunlight is behind them, which “blinds” the device’s camera.

<u>MEDIUM PRIORITY</u>

Module 4: Storage

4.1 StaaS provides space to the users. It contains two categories- Object Storage as a Service and Block Storage as a Service. StaaS uses the concepts of Partioning, Formatting, Mounting, iSCSI.

4.2 The project’s aim is to implement a standards based storage architecture and an initial range of services to address current demands, is scalable, available on-demand and online, and with the potential to be integrated with other storage platforms, whether centrally, locally or cloud provided.

Module 5: Platform

5.1 PaaS provides platform to its users. Here the user is provided Python2, Python3, JAVA and many more platforms to its clients.

Module 6: Container

6.1 Container Technology helps the individual client to manage its own personal container. As it provides every user with its own personal container having all the requirement wihich the user want as per mentioned by the shell.

Module 7: Software

7.1 SaaS provides useful and updated software to the users. The CSP ensures that the softwares are not corrupted and as per the requirement of the client.

7.2 SaaS applications run on a **SaaS** provider's servers. Instead of installing and maintaining software, we simply access it via the Internet like notepad, VLC, jupyter notebook etc.

Module 8: Infrastructure

8.1 Infrastructure as a service (IaaS) is a form of cloud computing that provides virtualized computing resources over the internet. IaaS helps we avoid the expense and complexity of buying and managing your own physical servers and other datacenter infrastructure

Module 9: Data Analytics (Hadoop)

9.1 Apache Hadoop is a 100 percent open source framework that pioneered a new way for the distributed processing of large, enterprise data sets. Instead of relying on expensive, and different systems to store and process data, Hadoop enables distributed parallel processing of huge amounts of data across inexpensive, industry-standard servers that both store and process the data. With Hadoop, no data is too big data.

9.2 HIVE: The Apache Hive data warehouse software facilitates querying and managing large datasets residing in distributed storage. Hive provides a mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL. At the same time this language also allows traditional map/reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express this logic in HiveQL.Support for exporting metrics via the Hadoop metrics subsystem to files or Ganglia; or via JMX.

9.3 Pig: Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets. At the present time, Pig's infrastructure layer consists of a compiler that produces sequences of Map-Reduce programs, for which large-scale parallel implementations already exist (e.g., the Hadoop subproject). Pig's language layer currently consists of a textual language called Pig Latin.

4 Other Requirements

We will use sql here for database .we use sql for the database which will be updated regularly by user. We can provide our products to students on the basis of free subscription of one year for the Learning purpose .Here students can implement new techonolgy in free of cost for a year.

5 Glossary

RHEL	Red Hat Enterprise Linux
CSP	Cloud Service Provider
SRS	Software Requirement Specification
HTTP	Hyper Text Transfer Protocol
SQL	Structure Query Language
API	Application Programmable Interface
CaaS	Container as a Service
PaaS	Platform as a Service
StaaS	Storage as a Service
IaaS	Infrastructure as a Service
SaaS	Software as a Service
NFS	Network File System

6 Analysis Diagrams

UseCase:

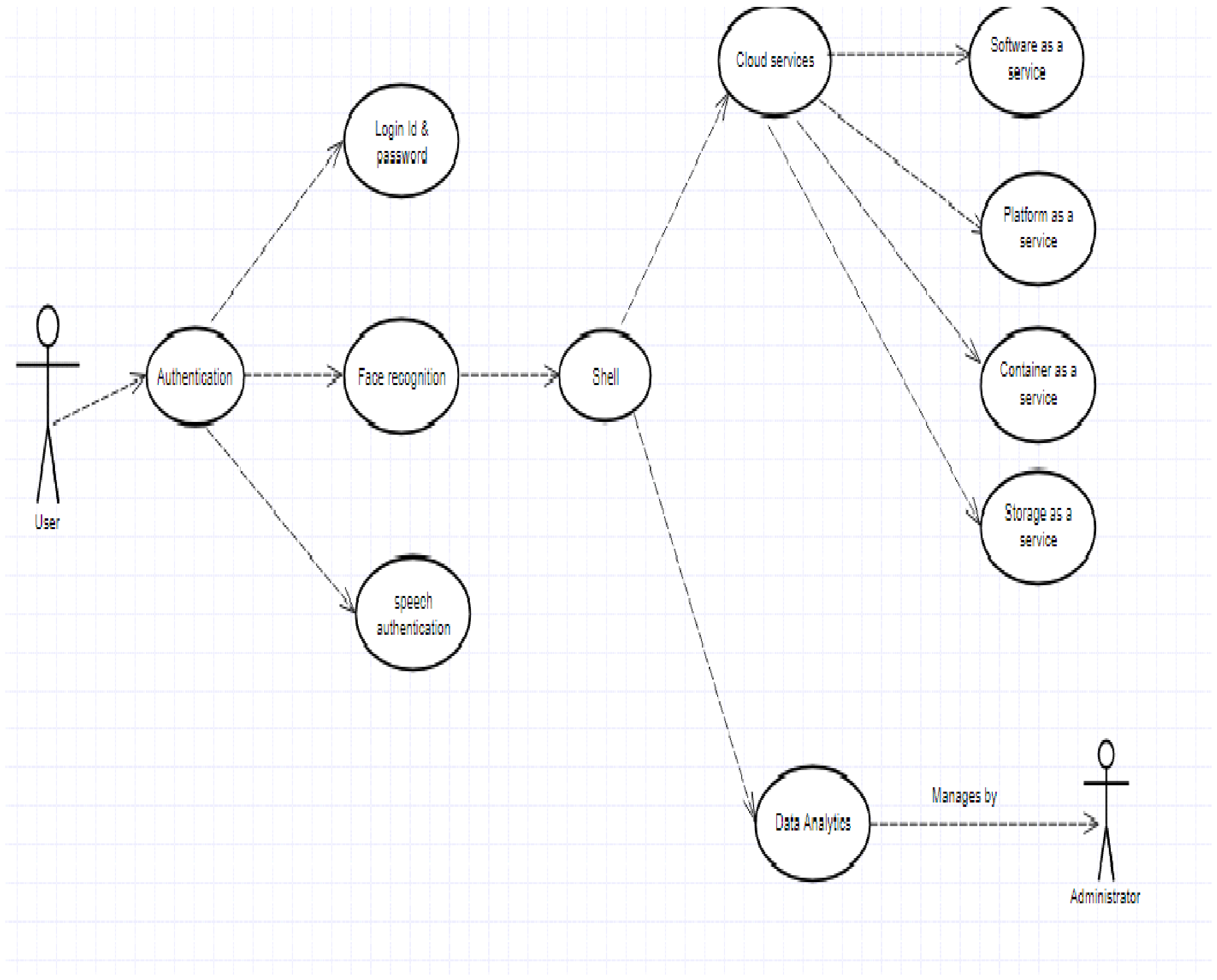


Figure no -1

System Chart / Module

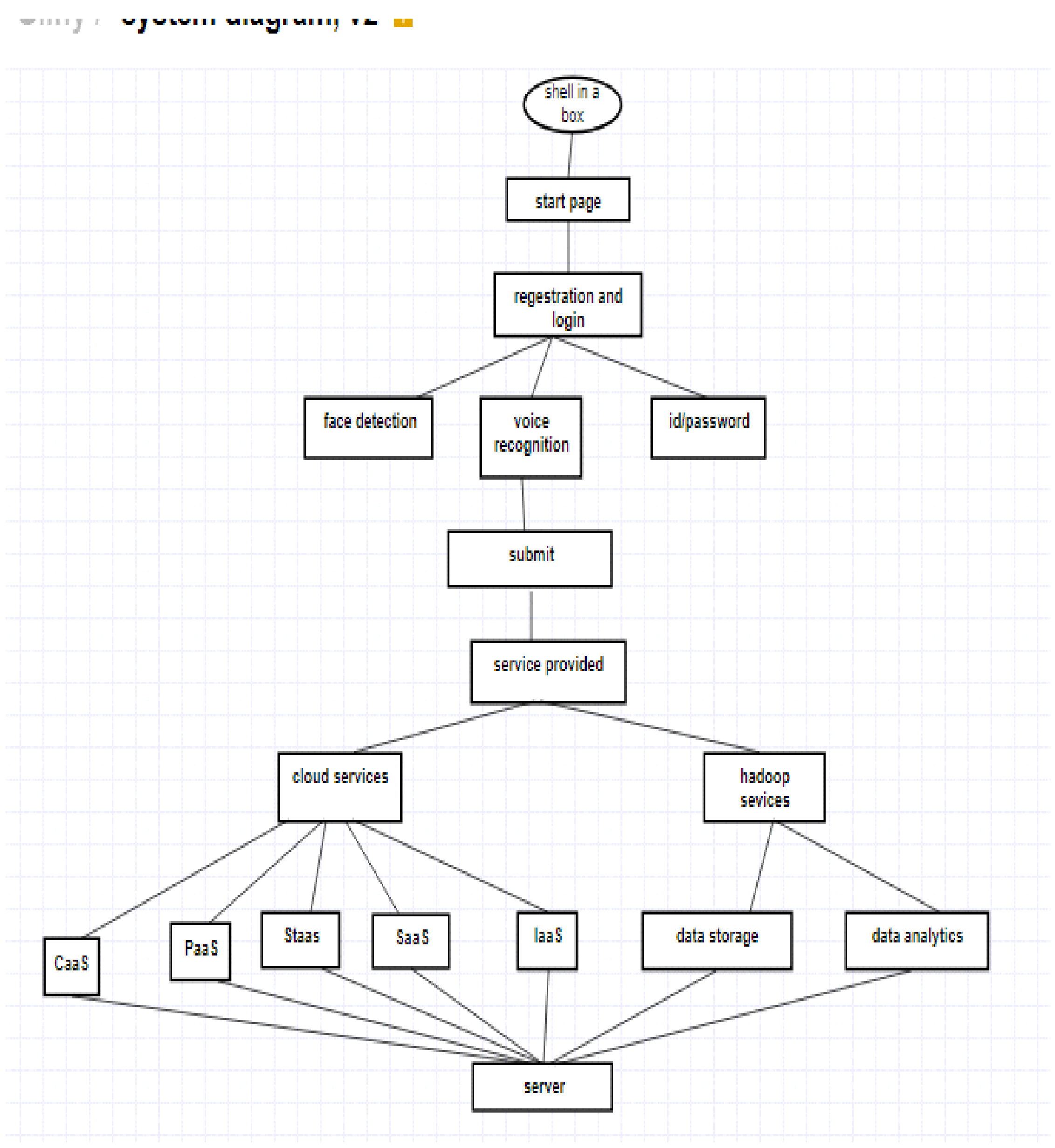


Figure no-2

Sequence Diagram:

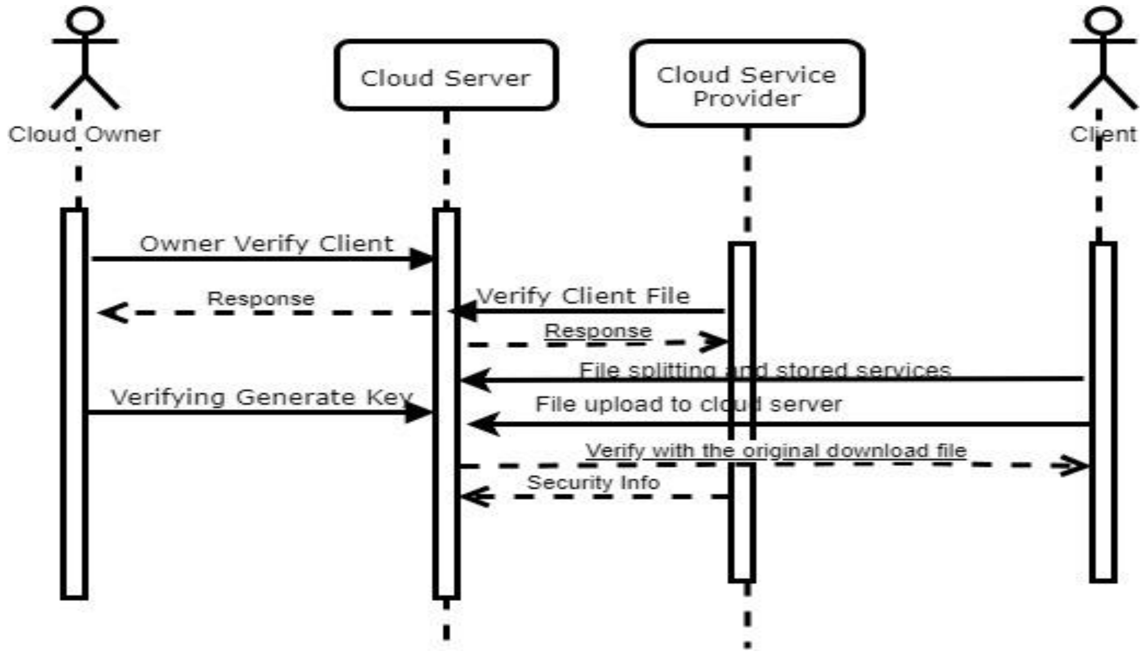


Figure no-3

Deployment Diagram

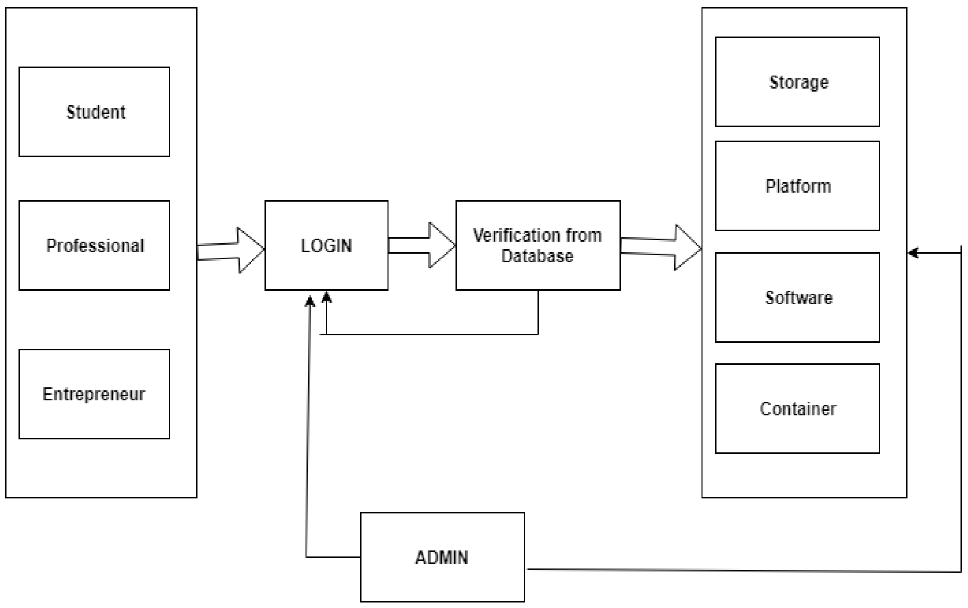


Figure no-4

Chapter -4

Software Design Document

1. Introduction

1.1 Purpose

This project is based on the Cloud based and Bigdata based Scalable Organization Management System with Semantic Web, Mobile Interface that will serve as a foundation for the final product. It is planned to satisfy everyone's expectations. The Shellinabox is intended to provide a quick, easy and user-friendly cloud-based services like SaaS, PaaS, CaaS, IaaS, StaaS and data analytics which is done through hadoop services. The system should be designed so that management time is minimized and the product is available to clients at a very economical price. All the Services are available under a single portal. Software Design Specification is a written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. An SDD usually accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design. Practically, the description is required to coordinate a large team under a single vision, needs to be a stable reference, and outline all parts of the software and how they will work.

1.2 Feasibility

1.2.1 The web-App reduces the effort and time of the clients. People spend a lot of time in finding an appropriate platform which provide secure services. This web-app server their purpose by giving them better user experience without much investment. The web-app will be made accessible to Enterprise, Professionals and Students. The data deployed by the users is secure and have backups also there is no fear of losing confidential data. The services are made accessible to the targeted clients using WebUI Techniques which makes it easier for the clients to access these service.

2. Architectural Design (System Flow Chart)

System Architecture

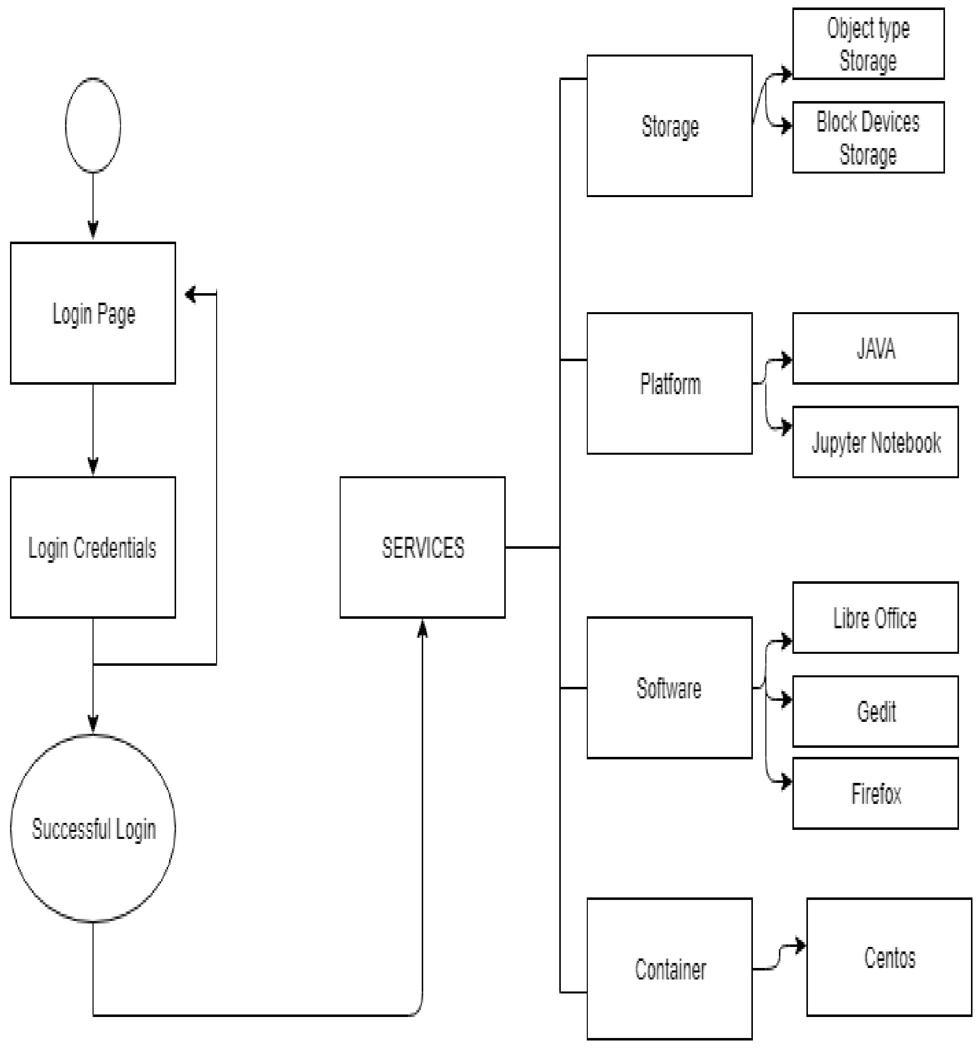


Figure no-5

3. UML (Class Diagram)

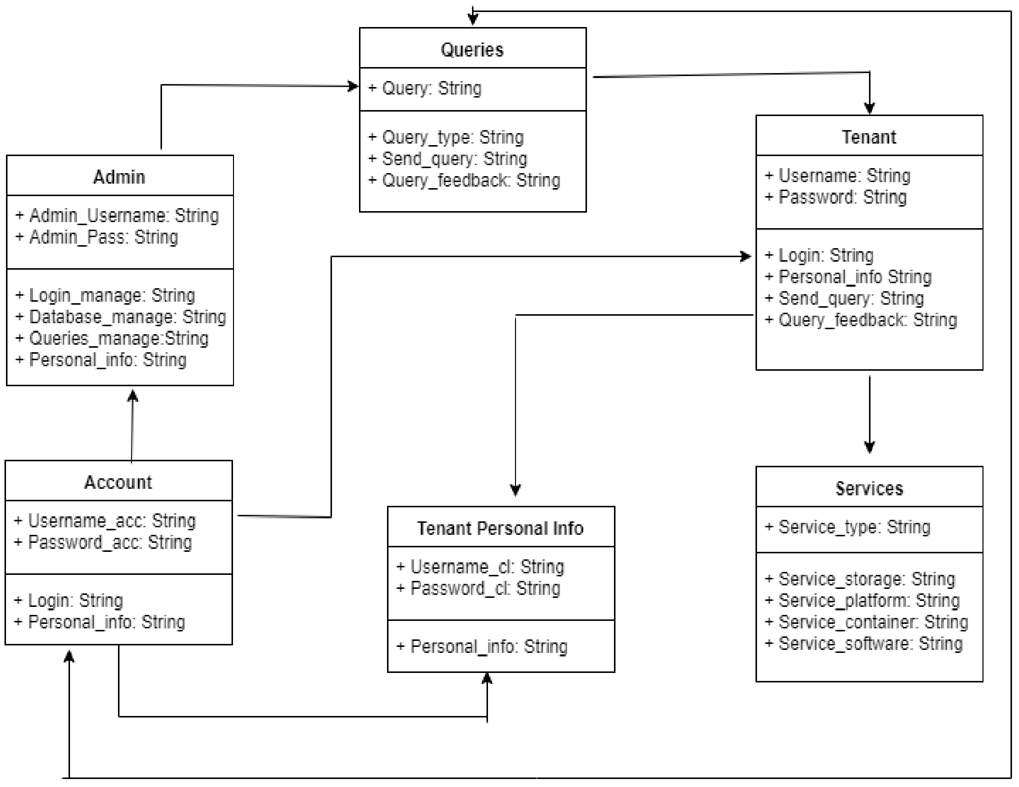


Figure no -6

4. Database Diagrams

4.1 Database Design of Registration

Field Name	**Datatype**	**Key Constraints**
Username	Varchar	PRIMARY KEY
First name	Varchar	NOT NULL
Password	Varchar	NOT NULL

Table no -1

4.2 Database Design of Login

Field Name	Datatype	Key Constraint
Username	Varchar	NOT NULL
Password	Varchar	NOT NULL

Table no-2

4.3 Database design for Containerization

Field Name	Datatype	Key Constraint
Container name	Varchar	PRIMARY KEY
Version	Varchar	NOT NULL
Creation date	Varchar	NOT NULL
Status	Varchar	NOT NULL
Operating System	Varchar	NOT NULL

Table no-3

4.4 Client Management Database

Field Name	Datatype	Key Constraint
IP Address	Int	PRIMARY KEY
Hostname	Varchar	NOT NULL
Service Type	Char	NOT NULL
Protocol	Char	NOT NULL

Table no-4

4.5 Container as a Service Database

Field Name	Datatype	Key Constraint
Status	Char	NOT NULL
Container Name	Varchar	PRIMARY KEY

Container Image	Char	NOT NULL

Table no-5

4.6 Software as a Service

Field Name	**Datatype**	**Key Constraint**
IP Address	Int	PRIMARY KEY
Software Name	Char	NOT NULL
Container Name	Char	NOT NULL

Table no-6

4.7 Storage as a Service

Field Name	**Datatype**	**Key Constraint**
IP Address	Int	PRIMARY KEY
Hostname	Varchar	NOT NULL
Storage Type	Char	NOT NULL

Table no-7

4.8 Platform as a Service

Field Name	**Datatype**	**Key Constraint**
IP Address	Int	PRIMARY KEY
Platform Name	Varchar	NOT NULL
Container Name	Char	NOT NULL

Table no-8

4.9 Hadoop Services

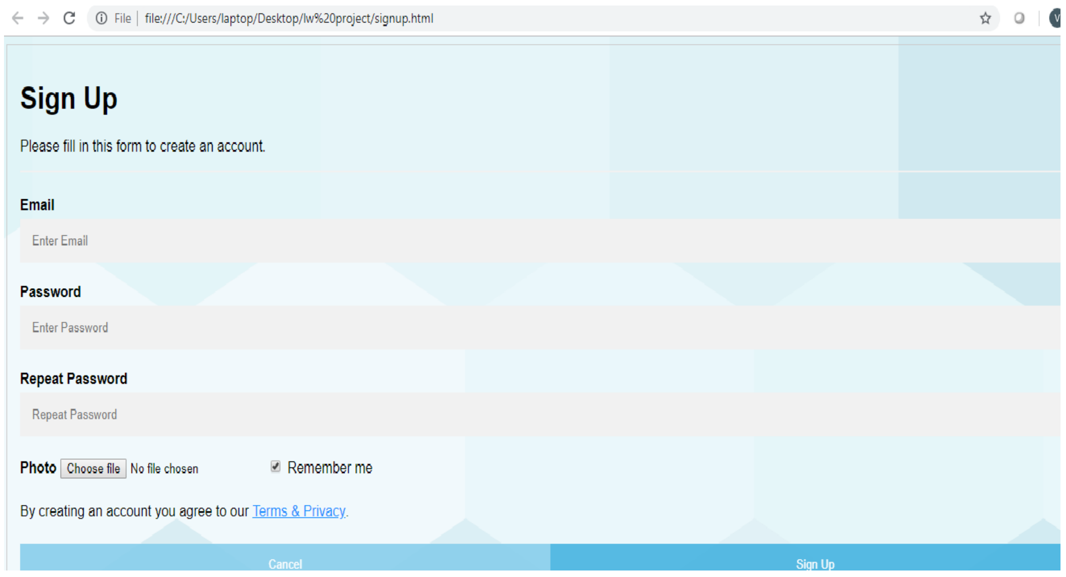
Field Name	**Datatype**	**Key Constraint**
Status	Char	NOT NULL
Container Name	Varchar	PRIMARY KEY
Number of nodes required	Int	PRIMARY KEY

Table no-9

5. GUI Design

5.1 Design for Signup Page

This is signup page. if client want to access our website then it can fill this form first to connect with our website. it is a first step for a new client. In this client can enter a unique number it is a port number of docker so it is different number for different client.



file:///C:/Users/laptop/Desktop/lw%20project/signup.html

Sign Up

Please fill in this form to create an account.

Email

Enter Email

Password

Enter Password

Repeat Password

Repeat Password

Photo Choose file No file chosen ☑ Remember me

By creating an account you agree to our Terms & Privacy.

Figure no -6

5.2 Design for login page

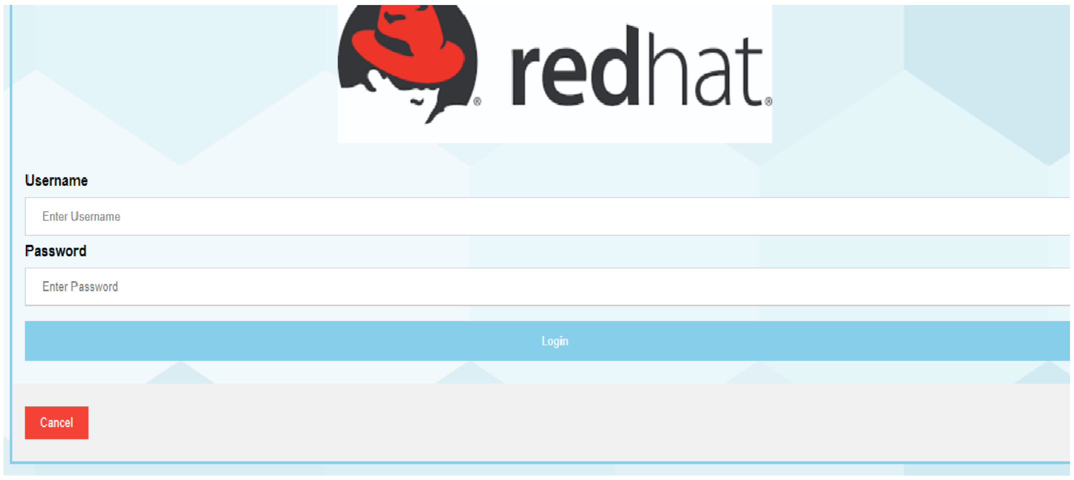
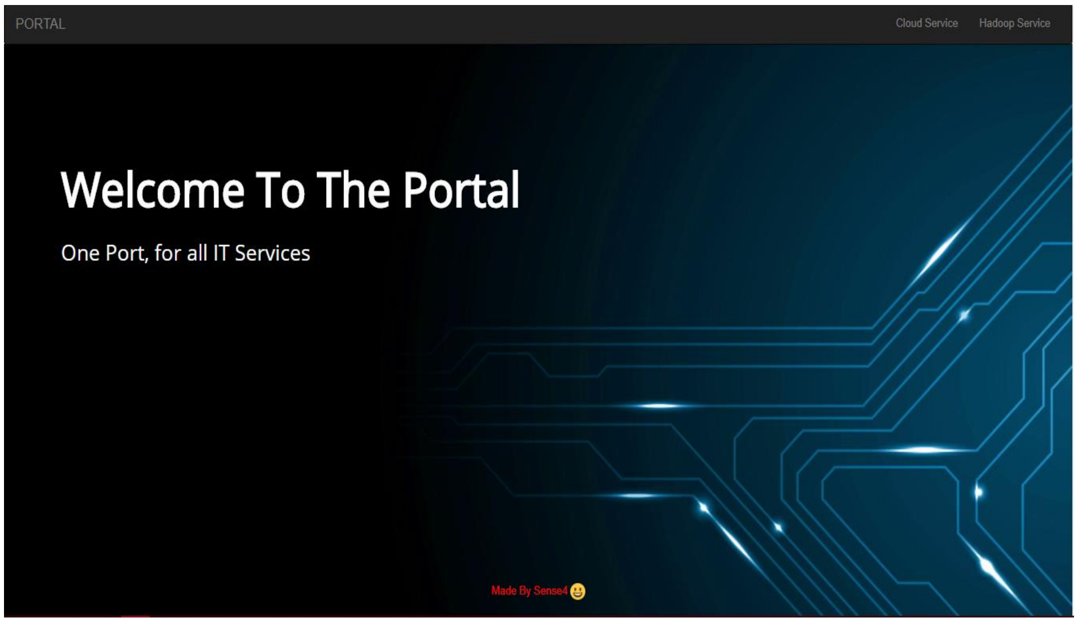


Figure no -7

5.3 GUI design for Service page



Welcome To The Portal

One Port, for all IT Services

Figure no -8

5.4 GUI design for front page

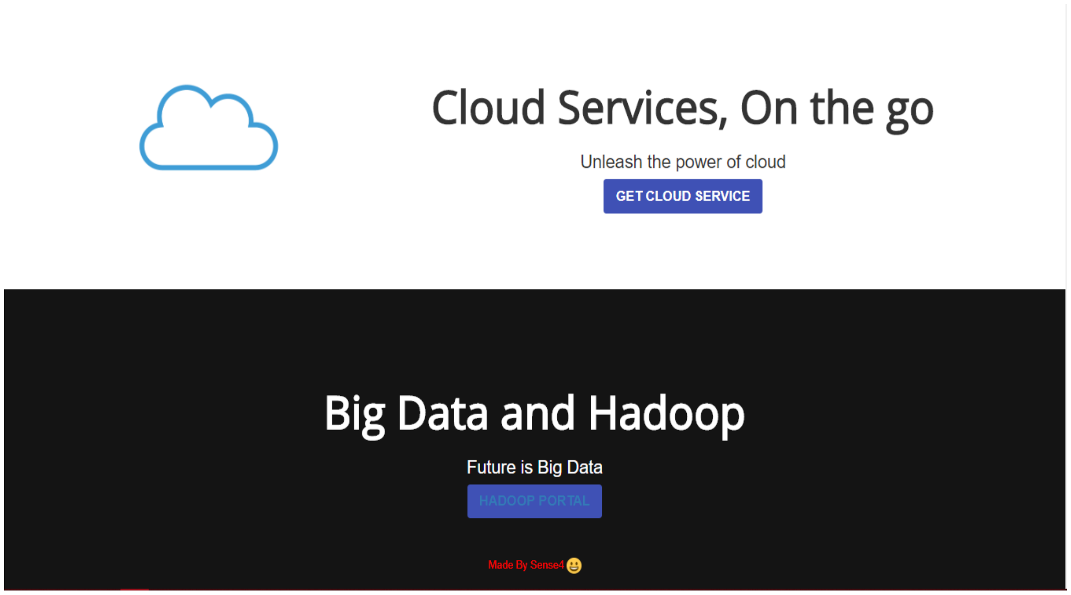


Figure no -9

5.5 GUI design for cloud portal

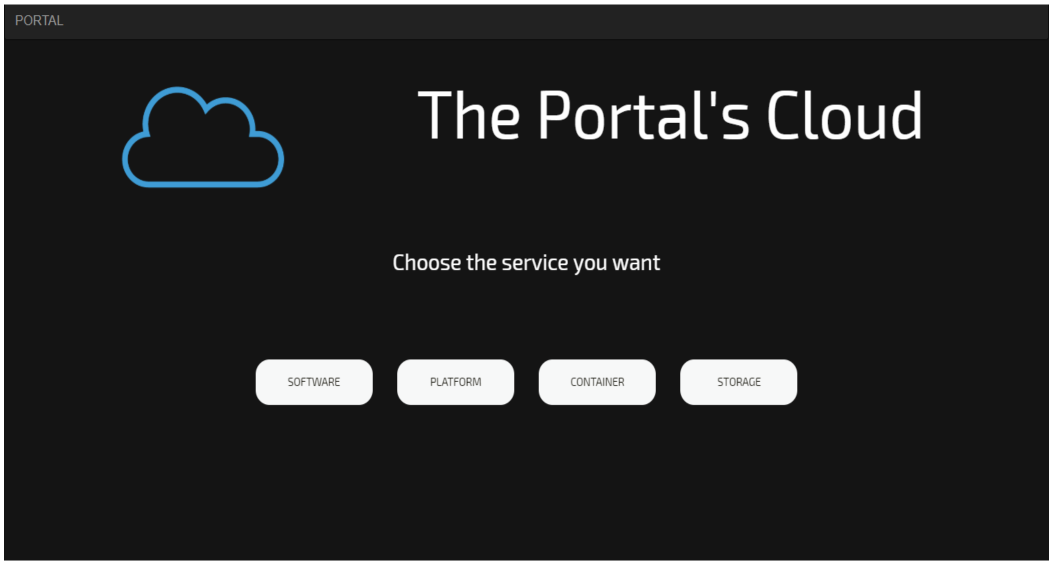


Figure no -10

4.6 GUI Design for SAAS

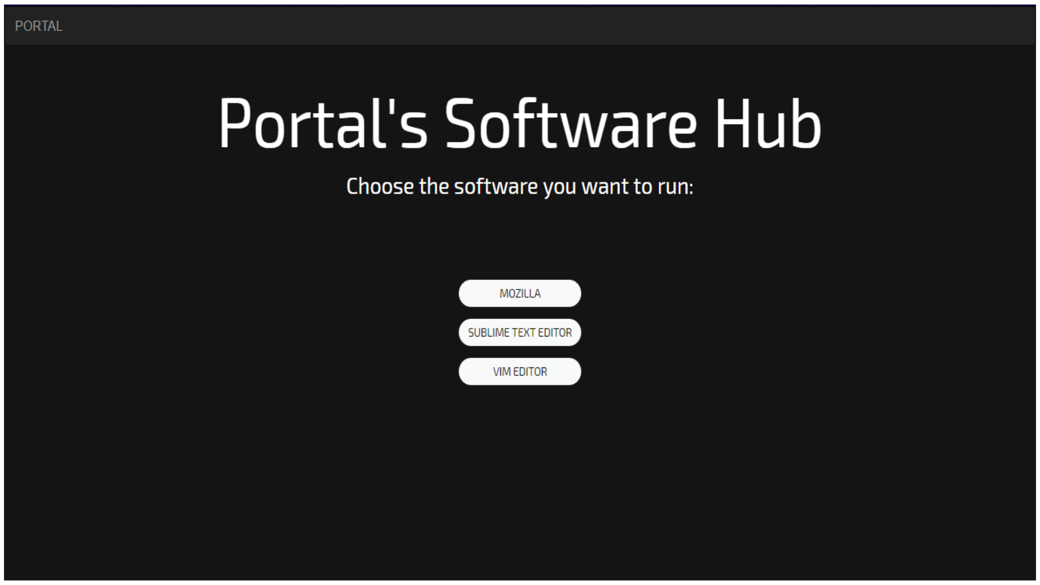


Figure no -11

5.6 GUI Design for PAAS

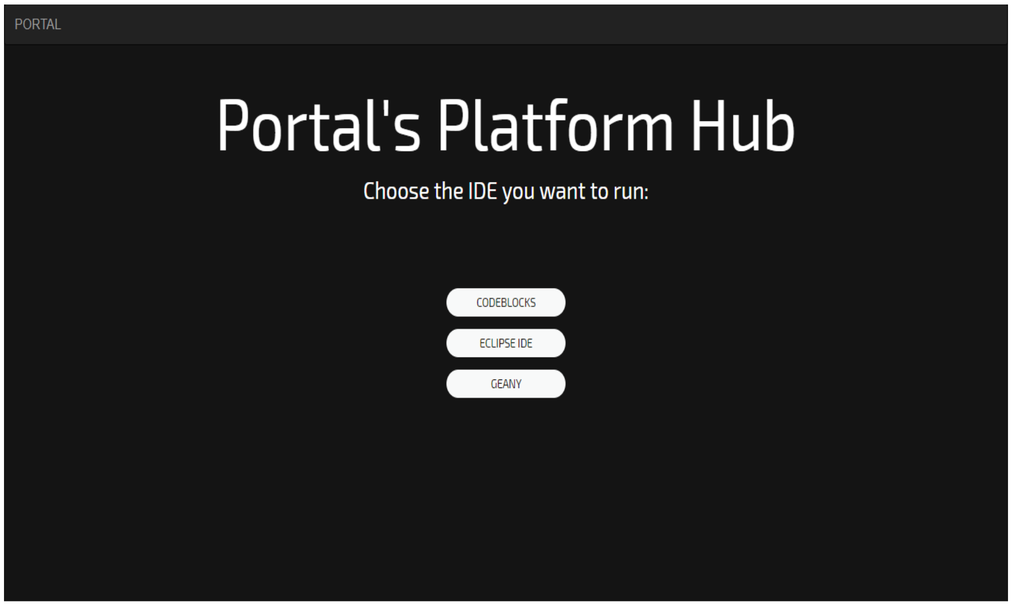
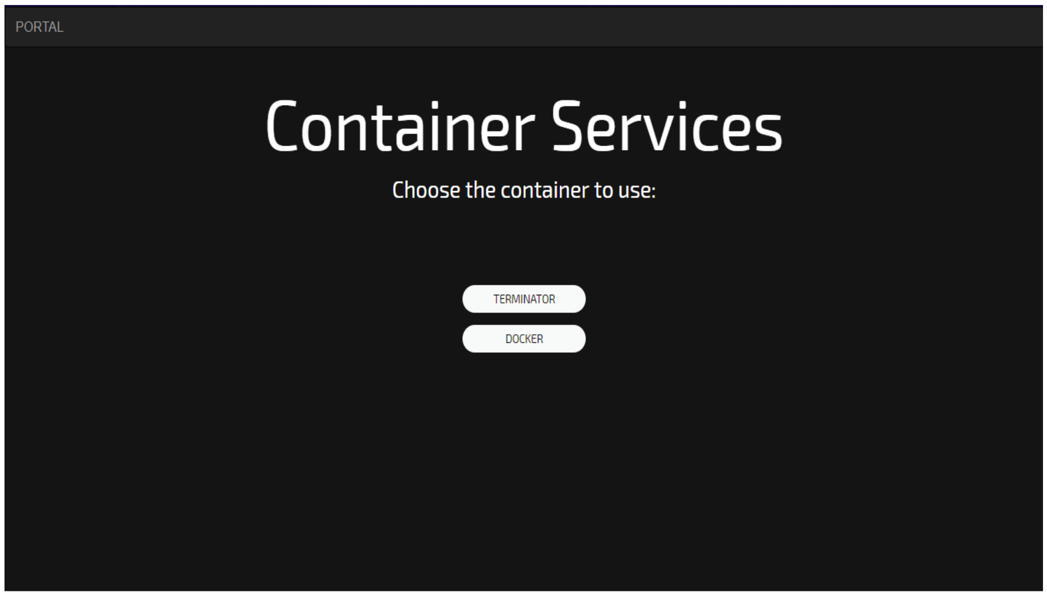


Figure no -12

5.7 GUI Design for CAAS

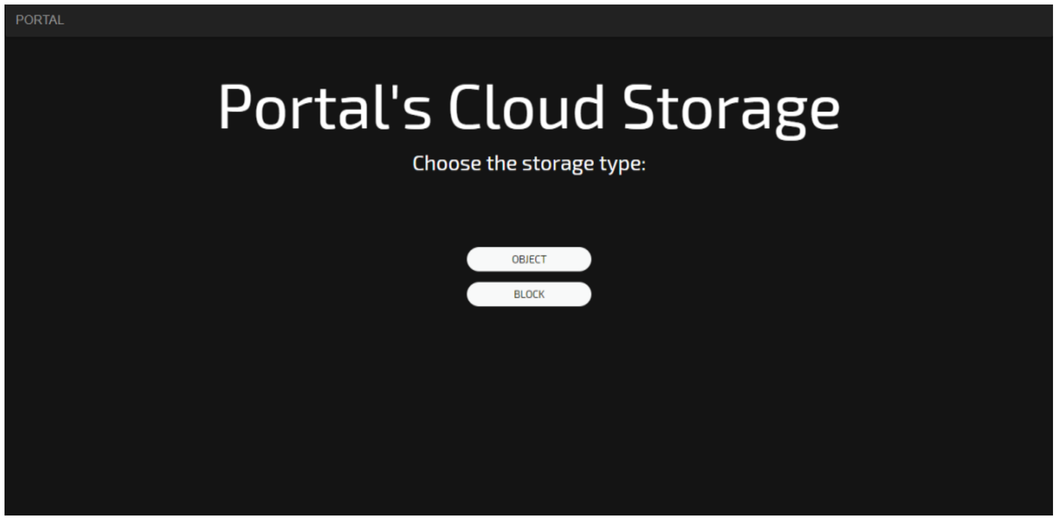


Container Services

Choose the container to use:

Figure no -13

5.8   GUI Design for STaas



Portal's Cloud Storage

Choose the storage type:

Figure no -14

5.9 GUI design for BigData Services

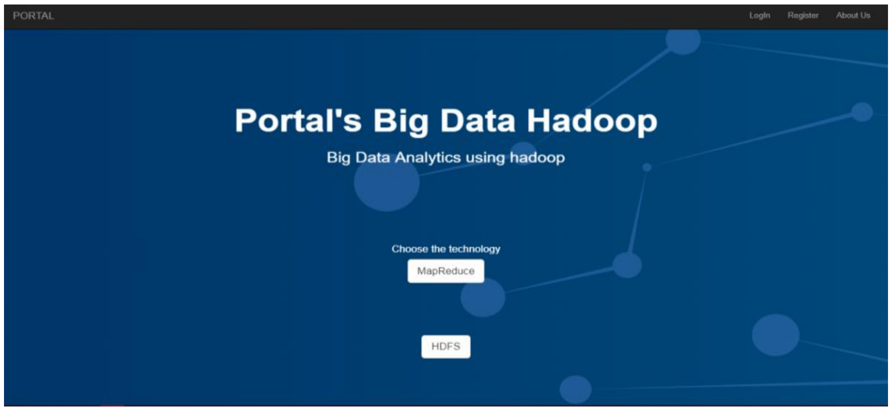
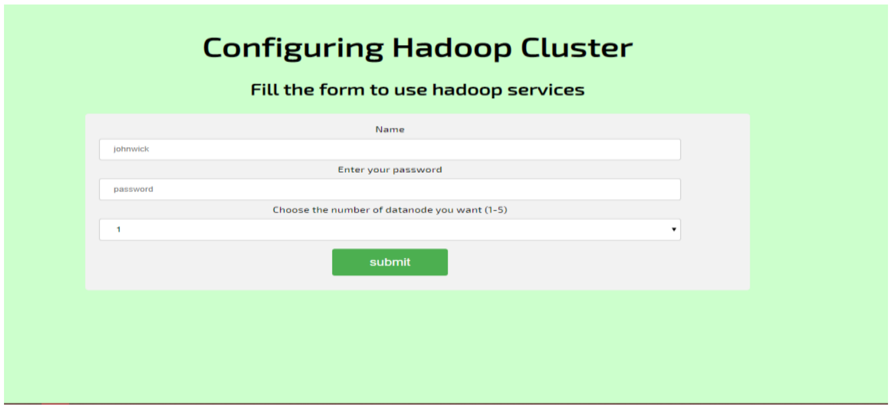


Figure no -15

5.10 GUI design for Bigdata configuration


Configuring Hadoop Cluster
Fill the form to use hadoop services
submit

Name: johnwick
Enter your password: password
Choose the number of datanode you want (1-5): 1

Figure no -16

6. Glossary

RHEL	Red Hat Enterprise Linux
API	Application Programmable Interface
CaaS	Container as a Service
PaaS	Platform as a Service
StaaS	Storage as a Service
IaaS	Infrastructure as a Service
SaaS	Software as a Service

Chapter -5

Report from Guide & Recommendation

Chapter -6

Conclusion & Future Scope

This report is aimed to exhibit our project’s requirement details.
First summary of project **SHELLINABOX** introduced. Then, the requirement details of the project are described.This report tried to focus on the aspects which thought to be important. So, there is no part that reflects irrelevant or useless information. This report was very useful for clarifying the project’s scope. Also, it’ll be beneficial for further planning.

The project is having main part Cloud & its Services and Bigdata(Hadoop) so as to compete with the performance & quality of the desired system, hard work is needed. From the beginning to the end, the whole process will be heavy and challenging to accomplish, but with the contribution of our group members and teaching assistant, we can come over all of the difficulties. We believe that this product will take its place in the market and users will see what they expected. The future scope will include the application being adapt to the changing trends in the market and thus making it smarter and smarter.

Frequently Asked Questions

Q1. What is Cloud?

Ans. Cloud computing is a type of computing that relies on shared computing resources rather than having local servers or personal devices to handle applications. In its most simple description, cloud computing is taking services ("cloud services") and moving them outside an organization's firewall.

Q2. What are Cloud Services?

Ans. Cloud-based is a term that refers to applications, services or resources made available to users on demand via the Internet from a cloud computing provider's servers.

Q3. What is Bigdata (Hadoop)?

Ans. The term Big Data refers to all the data that is being generated across the globe at an unprecedented rate. Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it’s not the amount of data that’s important. It’s what organizations do with the data that matters. Big data can be analyzed for insights that lead to better decisions and strategic business moves.

Q3. What are the key features of the project?

Ans. Some of them are - Feasibility, scalability, Flexibility, Extensibility andmost important is a user-friendly environment so that it is easy for the new users to understand and use it.

Q4. Why are we using Python?

Ans. At present, Python is supports many operatingsystems. Also its large and robust standard library makes Python score over other programming languages. The standard library allows we to choose from a wide range of modules according to wer precise needs.

Q5. What is cloud services used for?

Ans. Applications, storage and other services are accessed via the Web. The services are delivered and used over the Internet and are paid for by the cloud customeron an as-needed or pay-per-use business model.

Q6. **Why is Hadoop Services used for?**

Ans. Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.

Q6. What are examples of cloud computing?

Ans. Drop box, Gmail, Facebook, Maropost Marketing Cloud,Hub spot, Adobe Marketing Cloud, SlideRocket, Rat type, Amazon Web Services,Clear DATA, Dell's Secure Healthcare Cloud, IBM Cloud.

References

i. www.redhat.com
ii. www.searchcloudcomputing.techtarget.com
iii. www.techopedia.com
iv. access.redhat.com
v. http://w3schools.com
vi. https://docs.ansible.com/
vii. https://docs.docker.com/
viii. https://docs.python.org/

APPENDIX

Spark:

Apache Spark is a fast and general-purpose cluster computing system. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, MLlib for machine learning, GraphX for graph processing, and Spark Streaming.

For more detail go to this link: https://spark.apache.org/docs/2.3.0/

Ansible:

Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates.

Ansible's main goals are simplicity and ease-of-use. It also has a strong focus on security and reliability, featuring a minimum of moving parts, usage of OpenSSH for transport (with other transports and pull modes as alternatives), and a language that is designed around auditability by humans–even those not familiar with the program.

For more detail go to this link: https://docs.ansible.com/ansible/latest/index.html

Dockers:

Docker is a platform for developers and sysadmins to develop, deploy, and run applications with containers. The use of Linux containers to deploy applications is called containerization. Containers are not new, but their use for easily deploying applications is.

For more detail go to this link: https://docs.docker.com/get-started/