Project Proposal, Deployment & Delivery Model Critical Analysis and Response

KRYPTON

BY STORMTROOPERS

CSCI5409: Advanced Cloud Computing

Group 40 (Team 3): Stormtroopers

Tanvi Pruthi	–	B00875949
Sidharth Mahant	–	B00899439
Mayank Sareen	–	B00899565

Table of Contents

Overview

As the trend of Fintech and digitalization is hitting the global market at a much pace, many companies are utilizing innovation and offering numerous services related to finance and technology. Cryptocurrencies have evolved with digital payments systems and digital wallets expansion over many years. There are thousands of cryptocurrencies available in the market, where bitcoin was the first one founded and is still the best known today. Cryptocurrencies transactions and exchanges are traded daily with a large, estimated number of active users worldwide. Furthermore, all the companies prefer to host and deliver their websites on a cloud platform for better performance, scalability, reliability, and flexibility of the services. The combination of Fintech website and cloud computing platform is a game-changing digital transformation.

Stormtroopers, as a team, will implement a cryptocurrency exchange website 'Krypton' on Amazon Web Services (AWS) cloud platform to deliver an application with the latest technologies. Krypton will let users log in or register to buy or sell any cryptocurrency with a more user-friendly website. Along with the transactions, the user will get the option to check the market trends, latest market news, a personalized prediction model of the crypto currencies available in the user holding's wallet. It will provide the user with a decision-making platform and the flexibility to use many features in one place. Also, the visualization graphs will have a unique effect on the statistical analysis of personal holdings with the market trends. The development of each feature will be complete considering the security, privacy, and performance of the website on the AWS platform. The AWS services such as Virtual Private Network, EC2 instance, DynamoDB, Amazon Cognito, Elastic Load Balancing will deploy and deliver the website on the cloud as a Software as a Service (SaaS) platform because the web application will be available as software for use. The developers will use the AWS services to build all the website features such as Homepage, dashboard, holdings, wallet, trade, and personalized prediction. But the users will directly use the services and not access the infrastructure, storage or network used to develop the website. The goal is to provide a flexible and high-performance user experience with Krypton. Various API calls will help reach the crypto market and fetch the data for the users.

Although numerous websites have recently started to provide cryptocurrency-related information, there is still a significant market demand for alternatives easier to use and comprehend. Krypton is committed to providing users with thorough, high-quality, practical market research on the cryptocurrency market. Krypton will provide users with unrivalled insights into the market through a seamless and tailored approach by offering a prediction feature that estimates the value of a cryptocurrency based on real-time data and past data. Also, it will allow users to keep an eye on their preferred cryptocurrencies through a personalized watchlist feature. Krypton will be a decentralized application built using Python, JavaScript and various on-demand AWS cloud services for the back end, front-end development, and integration of backend and front-end. Team stormtroopers believe that Krypton will outshine on the speed front, where users can expect fast and seamless access to the information daily. Thanks to the features and functionalities of cloud computing platforms, such as on-demand availability of the computer system resources and the ability to distribute the website across multiple locations.

Program Functionalities

The website divides into five pages, where each page will have unique functionality for the user experience.

1. Homepage

The first page of the website is the home page, which would contain a navigation bar for the user to navigate to different sections of the website. The homepage would show different views like the market trends, Latest news, top 10 trading volume, top 10 cryptocurrencies prices. The navigation bar would have the options “login/ register”, “about us”, “buy crypto”, “credits”, “Markets”, “blogs”.

a. User login and Account Creation

The user signup and login page will handle the user accounts. It will have different user credential validations and store the user data in the cloud database. Additionally, the user will have the option to log in or register an account using social media for easy access and faster login.

b. Top 5 Exchanges and Top 10 Cryptocurrencies

The homepage will display the top 5 exchanges and top 10 cryptocurrencies to everyone, i.e., the user does not need to log in to see the information. The rates will refresh as per the live data retrieved using the API calls from cryptocurrency exchange APIs. Also, the graphs showing the market trend of cryptocurrencies will add visualization to the homepage.

c. News Feed

The news feed section on the homepage will retrieve the latest news for the top cryptocurrencies using the newsfeeds using API calls. This section will also be visible to everyone who visits the website. The purpose of this section is to demonstrate the top cryptocurrencies current news.

d. Footer

The footer section of the homepage will display the brief details of the Krypton website. Also, it will have various links to the social media pages as Contact Us icons. The links will redirect to the respective social media accounts of krypton.

2. Dashboard

Once logging in is successful, the website will redirect the user to the dashboard page. It will consist of the following three sections:

a. Watch list

This section allows users to select the top fifteen preferred cryptocurrencies and display them on the watchlist. By default, the user will see the top five cryptocurrencies displayed on the watchlist. Once the user adds the cryptocurrencies to the watchlist, the database will update and refresh. The customized watchlist will display the cryptocurrencies by retrieving the values from the database on subsequent logins.

b. Graph

The graph will show the cryptocurrency trends of the cryptocurrencies selected in the watchlist. That graph will show the top five cryptocurrencies auto-populated for first-time users.

c. Holdings

The holdings section on this page will show the latest five holdings. The user will click a hyperlink which will redirect to another page to check out the rest of those holdings, which will show the remaining holdings of the logged-in user.

3. My Holdings and Wallet

"My wallets" hyperlink on the dashboard will redirect the user to this page. This module will let the user select an action whether that user wants to Deposit, withdraw, transfer, or see the transaction history.

a. Holdings

This section will display all the user holdings with a buy price, date, profit, or loss percentage as of today's date, the invested amount for individual holdings, etc.

b. Current Balance

The amount of money the user has invested buy cryptocurrencies will display in this module. Also, after completing the buy or sell orders, the amount calculated based on those orders will reflect in this section. It will give a clear picture to the user of the investment amount as of today's date for all the holdings.

c. Profit and Loss (P&L)

The amount of overall profit and loss daily, monthly, quarterly, and yearly basis will display in the P&L module as of the current date. Also, this module will provide the P&L output for a particular cryptocurrency the user specifically wants to know.

d. Trade book

This section will cover the transactions of all the buy/sell orders placed by the user to date, i.e., buy/sell date, name of the cryptocurrency, the buy price, the selling price, percentage profit or loss, etc.

4. Trade

a. Buy/Sell (limit)

This module will allow the user to decide any limit at which that user wants to buy or sell any cryptocurrency. If the user chooses the buy option and the limit price or lower price hits, the order will complete successfully. Similarly, if the user selects the sell option and the limit price or higher price hits, the sell order will complete successfully.

b. Buy/Sell (market price)

The user will get the option in this module to buy or sell any cryptocurrency at the market price, i.e., the current price at a particular time. Also, the user will not get the option to retract the buy/sell order placed once completed.

c. **Open Orders**

This section will display all the open buy/sell orders in the queue and are waiting for the decided limit placed by the user to hit.

d. **Completed Orders**

This section will display all the completed buy/sell orders in the queue when the decided limit placed by the user hits.

5. Personalized Prediction

a. **Watchlist**

The watchlist module in the dashboard will repeat in the personalized prediction page for better analysis and visualization of the cryptocurrencies the user holds in the account.

b. **Recommendation system according to previous holdings**

This module will display the machine learning model of the predictive analysis of the cryptocurrency holdings in the user account. For example, based on the prediction results and the trend that the model represents, if a user wants to sell any holding, the machine learning model will help the user to decide whether that user should sell the holding or not.

c. **Graphs for predicted and current holding**

Based on the results of the recommendation model, this module will display graphs of the predictive results along with the holdings of the user.

Program Components Interaction Flow Chart

Figure 1 shows the project flow diagram and the interaction of the components of the program.

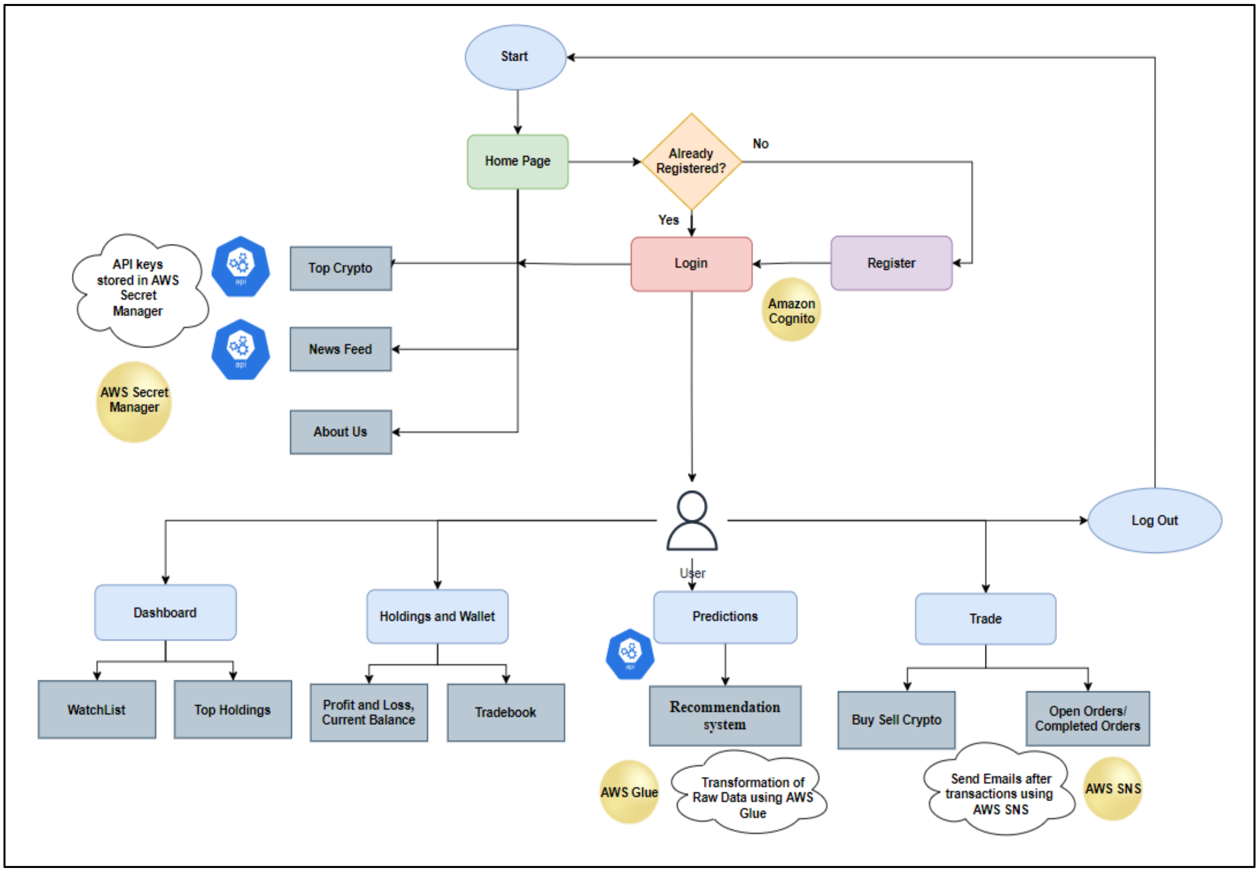


Figure 1: Project Flow Chart

AWS Services

Table 1 lists the AWS services planned for the delivery and deployment of the Krypton website on the AWS cloud.

Table 1: List of services to deploy the web application on AWS cloud

CATEGORY	SERVICES
COMPUTE	AWS EC2 Docker & AWS Elastic Beanstalk
STORAGE	AWS S3 AWS DynamoDB
NETWORK	AWS Virtual Private Cloud

	AWS API Gateway
GENERAL	Amazon Cognito AWS SNS AWS Secret Manager AWS Elastic Load Balancing AWS Glue

1. COMPUTE

a. AWS EC2

Amazon Elastic Compute Cloud provides Virtual Servers or Compute Instances to run the applications on the cloud and configuration of those virtual servers through Amazon Machine Image (AMI), e.g., Windows, Linux etc. Once the instance configures and runs, an IP address generates, and billing starts based on the time and usage of that instance. Our web application Krypton will be deployed on EC2 instance based on the volume of the application built, i.e., CPU, memory and storage required to run that application on the cloud server. This virtual server will help deliver a safe, high-performance application with flexible and scalable computational power on the cloud.

b. Docker & AWS Elastic Beanstalk

The docker platform creates an image of the source code using the docker file and pushes that image to the docker hub platform. Once an EC2 instance is up and running, build and run the docker image using AWS Elastic Beanstalk on that EC2 instance with the help of the docker-compose file and pulling the respective image from Docker Hub. The krypton application deployment will run on Elastic Beanstalk in EC2 instance using Docker for the quick deployment and management of the application, where Docker will take care of the respective infrastructure through docker images.

2. STORAGE

a. AWS S3

Amazon Simple Storage Service stores the data in the buckets as objects, and objects store the data as key-value pairs and is a NoSQL database. Moreover, it allows the creation of multiple versions of the data, and restoration is easy if accidentally deleted or overwritten. The krypton application will utilize the S3 buckets to store data such as logs, images, etc., where creation, deletion and updating of the buckets and objects will take place using REST API.

b. AWS DynamoDB

DynamoDB is also a NoSQL database service available as an Amazon web service where it stores data as key-value pairs in documents. DynamoDB will hold all the user-related data of the Krypton application, e.g., when the user logs in or registers to the website, when trading of the cryptocurrencies happens, the user's account balance details, etc. As DynamoDB is fast and handles traffic efficiently, the application will do the transactions and store the details of those transactions with high performance.

3. NETWORK

a. AWS Virtual Private cloud

VPC is a private network created in a particular region to separate the network from the public availability. Launch the EC2 instances in a public or private subnet of this private network, and the network connects to the internet through the internet gateway. A particular region can have multiple VPCs created as per the requirement of an application. The deployment of the Krypton web application will take place in an EC2 instance, which connects to a virtual private network to ensure resource placement, connectivity, and security of the application.

b. AWS API Gateway

API Gateway is an endpoint that connects various third-party applications to amazon web services such as EC2. The Krypton application will use the private endpoint in the same Virtual Private Cloud network to connect to the EC2 instances hosted in the same VPC network. It will help reduce latency and securely expose the REST APIs to the same VPC network.

4. GENERAL

a. Amazon Cognito

Amazon Cognito is a user identity and data synchronization service which helps reduce back-end development to store the data when any user logs in or registers to a website or an application through Cognito API. It helps to integrate the user identity with the various social networks of that user and stores the data as key-value pairs. Amazon Cognito will store the user data of the Krypton website with the feature of logging in or registering through email, Facebook, and Gmail. DynamoDB will have further links to the user credentials and data to access the data for the cryptocurrency transactions.

b. AWS SNS

Amazon Simple Notification Service is a messaging service that helps send messages such as email, SMS, mobile push, etc. So, once the user logs in or registers to the Krypton website, the SNS service will send an email to the user email id, which that user used to log in or register. Also, when the user buys or sells any cryptocurrency, AWS will trigger the SNS service to send an email to that user with the transaction details.

c. AWS Secret Manager

The Secrets Manager stores the API keys, database credentials, or any hardcoded credentials securely, i.e., if anyone has access to the code in the future, the essential secrets are not revealed, thereby maintaining privacy and security. The Secrets Manager will store any credentials or keys used in the Krypton website development, and those credentials are retrieved whenever there is a requirement over the HTTPS channel.

d. AWS Elastic Load Balancing

The Elastic Load Balancing redirects and redistributes the traffic coming from various sources and multiple hits from those sources simultaneously. The response time to those destinations becomes faster, as the response will come from many targets where the traffic redirects. The Krypton website will incorporate the Elastic Load Balancing, where multiple servers or resources will be added or removed as per the requirement to improve the performance and scalability of the application.

e. AWS Glue

The Krypton website will receive live raw data using API calls from internet sources and store the data in an S3 bucket. The AWS Glue service will extract the raw data from the S3 bucket, transform the data as per the portfolio requirement, and load the transformed data in the S3 bucket. Furthermore, the website will display the transformed data for a better user experience and better results of the personalized prediction of the cryptocurrencies for the users.

High Level Architecture

Figure 2 shows the high-level architecture of the AWS services planned for the project.

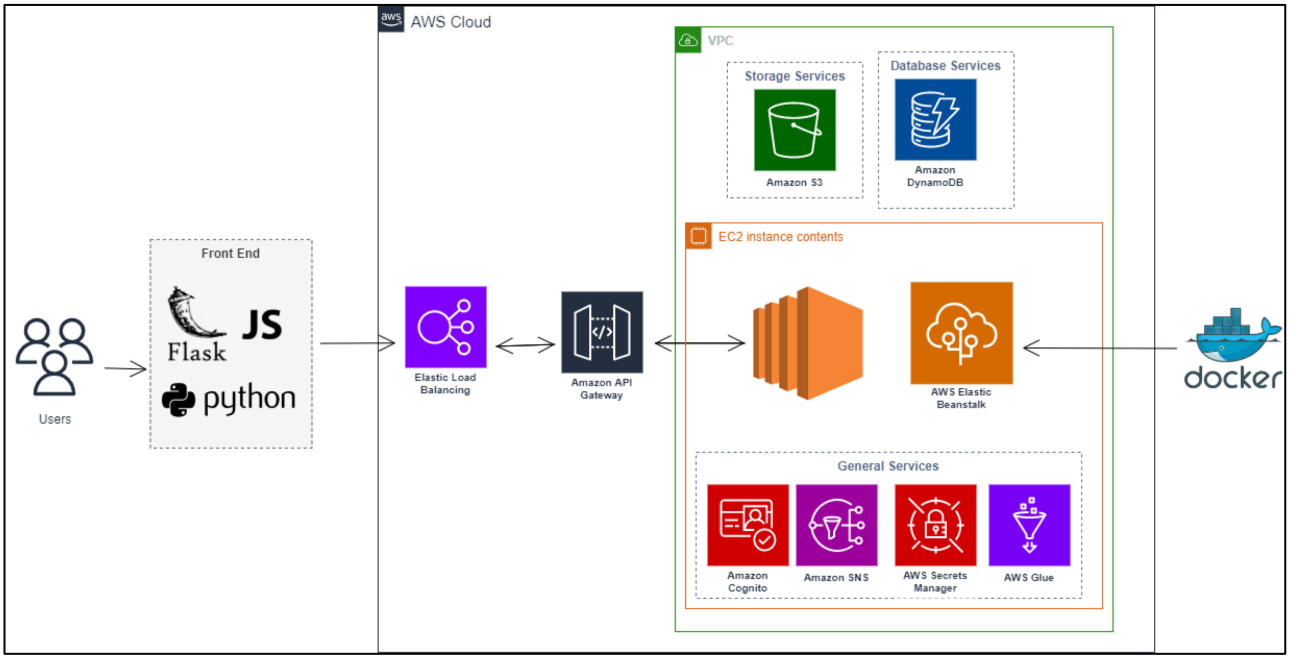


Figure 2: AWS Services high level architecture

Project Plan

The developers will develop the Krypton website in five phases. Each will cover around 20-25% of the application, including requirement gathering, solution architecture, design, development, testing, and deployment using agile methodology.

Sprint 1: (5 Feb – 12 Feb) - 10%

1. Requirement gathering
2. Project idea discussion
3. Critical Analysis
4. Project Proposal

Sprint 2: (13 Feb – 19 Feb) - 30%

1. Homepage
2. Dashboard
3. Data gathering
4. Architecture design and structure

Sprint 3: (20 Feb – 5 March) - 50%

1. User authentication implementation
2. Holdings and Wallet implementation
3. Initial deployment and testing

Sprint 4: (5 March – 19 March) - 75%

1. Trading
2. Personalized Prediction

Sprint 5: (19 March – 26 March) - 100%

1. Final project implementation
2. Final Report
3. Final video presentation

Also, we will apply the Agile development approach for the project in the following ways to deliver it incrementally.

1. As per the sprint timelines mentioned above, we have recognized the product backlogs and the features we planned to cover in a sprint. Each Sprint would last around 1-2 weeks and its own set of tasks to cover.
2. We will share every task of equal complexity among the stormtroopers teammates, ensuring that everyone gets an equal amount of work and contributes equally to the project.

3. We will have a daily scrum meeting to discuss the day's progress and utilize that meeting as a Sprint Retrospective meeting to discuss peer evaluations or TA's feedback to improve the process.
4. Once a week, we will have a Sprint Planning Meeting to plan the next Sprint and create a sprint backlog. Also, we will utilize the same meeting to check for pending items in the Product Backlog and reprioritize the work.
5. We will have the main branch in Gitlab for the production-ready code and a Developer branch for development and testing.
6. After complete feature implementation, a developer will integrate the functionality with the main branch and perform the integration testing. Once tested and reviewed by the other team members, the developer will merge the code to the GitLab main branch.

Critical Analysis and Response

1. **Describe the deployment model for the cloud-based software system you intend to build. Why did your group pick this deployment model?**

   To deploy all the application parts of the website on the cloud, we are using the public cloud as the deployment model for the cloud-based software system. The whole application will run on the cloud without virtualization or on-premises resource management. It will utilize the hardware, database and network on the cloud and a thin client, i.e., a web browser, to access the AWS cloud services. The Software-as-a-Service (SaaS) platform provided by the cloud service provider will deliver and manage the underlying infrastructure for the website on its own.

   We picked the public cloud deployment model to use the on-demand resources as per the requirement and traffic on the website. Also, we wish to deploy the application by using all the cloud-based resources, i.e., AWS services, without any use of the existing on-premises resources.

2. **What are the benefits of your deployment model?**

   The following are the benefits of using the public cloud deployment model with the Software-as-a-Service (SaaS) platform:

   a. **Scalability:** The public cloud lets the users use the resources and services as per the requirement, and the billing of those services starts if they are in use. It gives on-demand flexibility to the customers to decide the infrastructure, network, storage, etc. align with the application and system demands.
   b. **No maintenance:** As cloud provider manages and maintains the infrastructure, customers need not worry about any maintenance or software update. The latest infrastructure is available on the cloud to the customers to build applications.

c. **On-demand service and lower costs:** The public cloud providers let users pay for the services if used for building the applications. It reduces the overall cost as no operational or maintenance expenses billing is done to the businesses.

d. **Disaster recovery**: There is always a risk of losing data while maintaining any application. But on the public cloud, the customers can recover the lost data and infrastructure if any disaster occurs because there is always a backup available on the cloud.

3. **What are the drawbacks or risks of your deployment model?**

The following are the drawbacks of using the public cloud deployment model with the Software-as-a-Service (SaaS) platform:

a. **No control:** The cloud providers manage the entire services on the public cloud. The users cannot customize the infrastructure or the system according to their needs. Hence it gives them less control over the services.

b. **Limited customization:** Only the services available on the cloud can be used and cannot be customized. Customers must build the applications from the existing configurations and limited options only.

c. **Performance:** There are many customers around the globe using public cloud services, which can sometimes affect the application performance in some locations.

d. **Cloud expertise:** The Public cloud requires a specialized skill set and experience to understand the cloud models thoroughly. The complexity of the infrastructure creates a gap between the requirements and the IT professional's roles.

4. **Describe the delivery model for the cloud-based software system you intend to build. Explain the tradeoffs of this model in comparison to other delivery models (IaaS, FaaS, PaaS, SaaS).**

The four delivery models available in cloud computing are Software-as-a-Service (SaaS), Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Function-as-a-Service (FaaS). All four platforms help easy access of programs, software, database, etc. and provide more flexibility to businesses in the IT world.

The SaaS model helps run the application on a web browser without worrying about the infrastructure. The cloud service provider takes care of all the on-demand resources as per the application requirement, and billing starts if resources are in use. Moreover, limited configuration is required, and updated software is available for program execution. A general example of the SaaS model is Netflix, which uses this platform extensively.

Another model used by many organizations is IaaS to customize infrastructure as per business requirements and provides the feature to only pay for the space used. Although the cost required to use the IaaS platform is high, it offers flexibility to the customers to reduce maintenance of on-premises data centers. For example, customers use Microsoft Azure as IaaS to configure, deploy and manage infrastructure.

The third model, PaaS, is a platform that helps developers build applications to build, test and run applications, where the cloud provider manages the configuration of the resources. One of the examples of PaaS is SAP open business platform. There are many existing libraries and modules for the developers on the SAP platform. Therefore, it minimizes the complexity of the build process.

The FaaS model provides a platform for the developers to build and run the code as functions with no control over the infrastructure and reduces the complexity of maintaining and managing the underlying infrastructure. AWS Lambda is one of the popular FaaS platforms where developers write business logic as functions and run them on any operating system provided as a platform by the cloud provider.

Comparison of the four models according to the metrics below:

a. Flexibility: IaaS > PaaS > SaaS > FaaS
b. Cost: FaaS > SaaS > PaaS > IaaS
c. Complexity: IaaS > PaaS > SaaS > FaaS
d. Usage-friendly: FaaS > SaaS > PaaS > IaaS

Tentative Work Distribution

Table 2 distributes the implementation of the project among three developers for the equal contribution of the work.

Table 2: Project Work Distribution

NAME	**WORK ITEMS**
MAYANK SAREEN	1. Homepage 2. Holdings and Wallet implementation 3. AWS SNS 4. Initial EC2 deployment
SIDHARTH MAHANT	1. Dashboard 2. Personalized prediction 3. AWS Glue 4. Data gathering
TANVI PRUTHI	1. User authentication implementation – AWS Cognito 2. Trading 3. AWS Secrets Manager 4. Initial Testing

Wireframes & Feels

Figure 3 shows the overview of the Krypton website homepage, where all the homepage components are aligned to demonstrate the perspective of the website look.

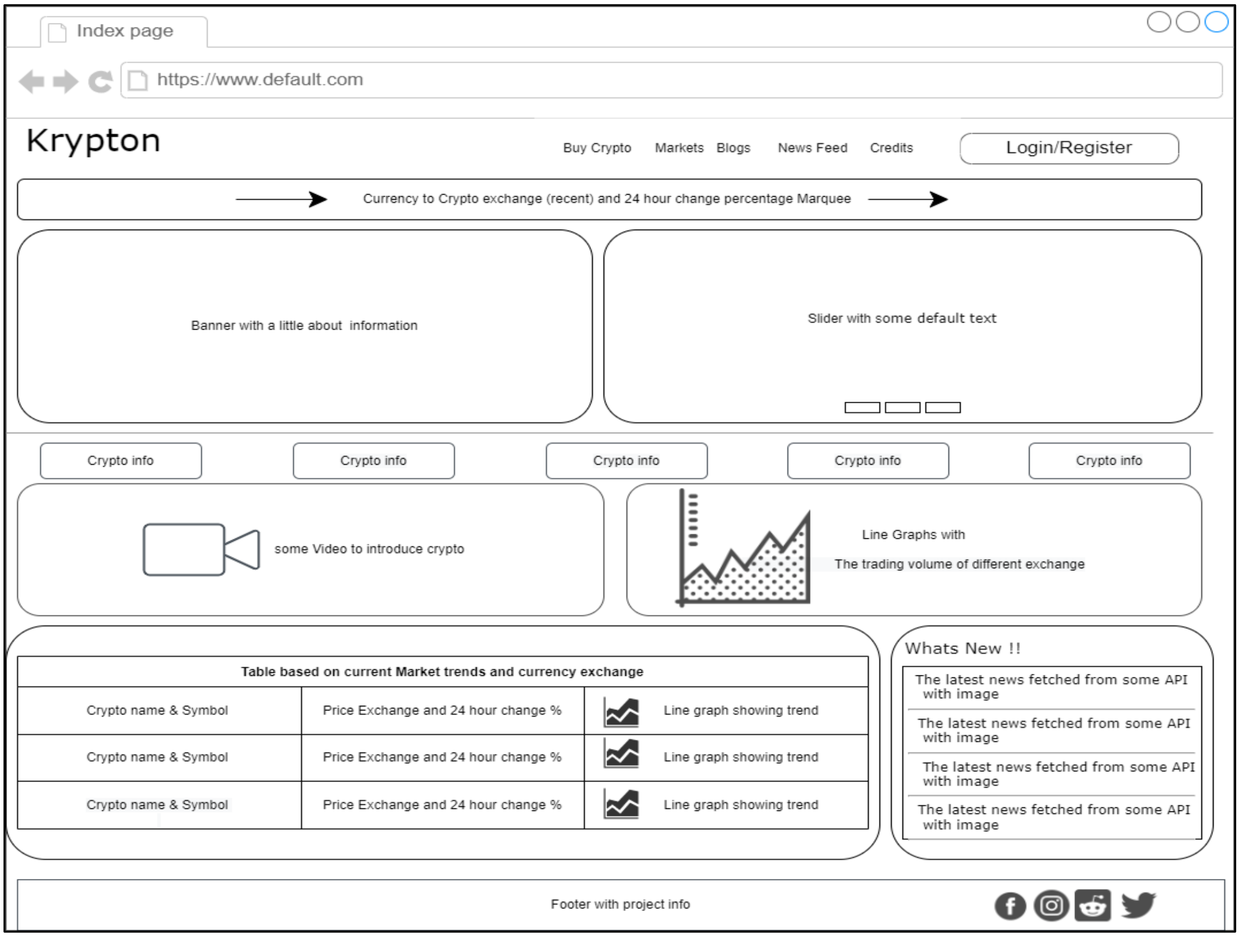
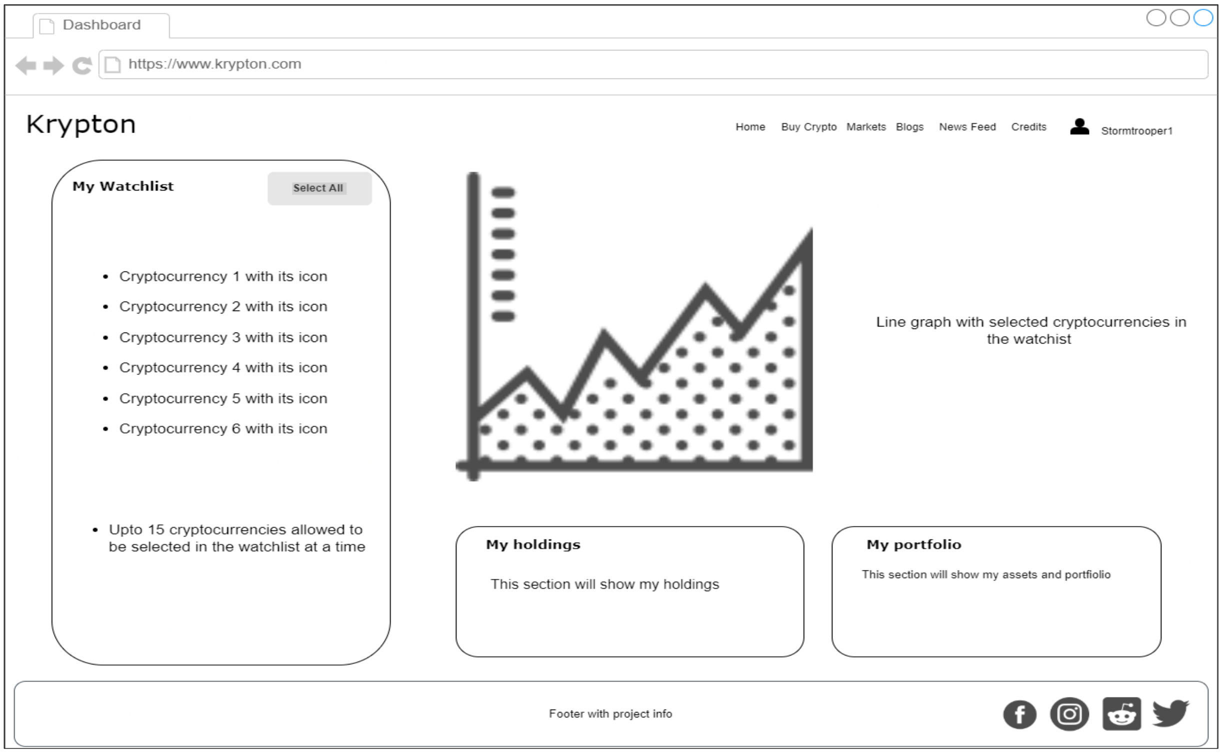

Index page
https://www.default.com
Krypton
Buy Crypto
Markets
Blogs
News Feed
Credits
Login/Register
Currency to Crypto exchange (recent) and 24 hour change percentage Marquee
Banner with a little about information
Slider with some default text
Crypto info
Crypto info
Crypto info
Crypto info
Crypto info
some Video to introduce crypto
Line Graphs with
The trading volume of different exchange
Whats New !!
The latest news fetched from some API with image
The latest news fetched from some API with image
The latest news fetched from some API with image
The latest news fetched from some API with image
Footer with project info

Table based on current Market trends and currency exchange		
Crypto name & Symbol	Price Exchange and 24 hour change %	Line graph showing trend
Crypto name & Symbol	Price Exchange and 24 hour change %	Line graph showing trend
Crypto name & Symbol	Price Exchange and 24 hour change %	Line graph showing trend

Figure 3: Homepage

Below Figure 4 shows the dashboard page of the website where it visualizes the watchlist and graph components of the page.



- Cryptocurrency 1 with its icon
- Cryptocurrency 2 with its icon
- Cryptocurrency 3 with its icon
- Cryptocurrency 4 with its icon
- Cryptocurrency 5 with its icon
- Cryptocurrency 6 with its icon

- Upto 15 cryptocurrencies allowed to be selected in the watchlist at a time

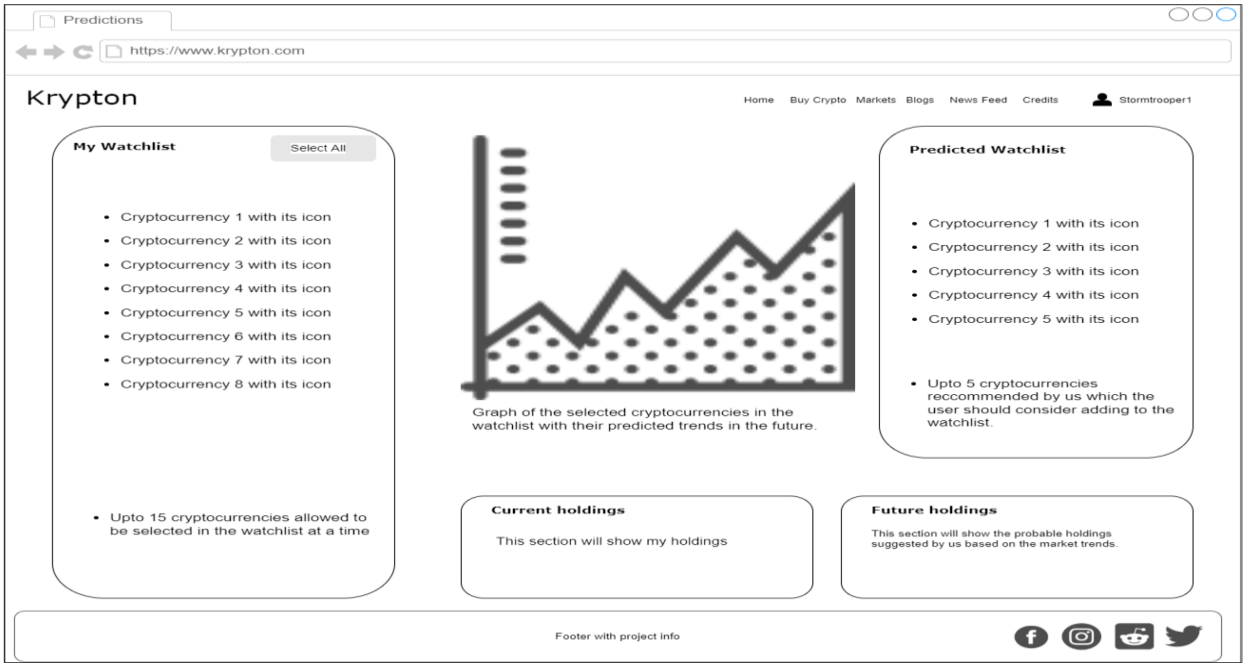
Line graph with selected cryptocurrencies in the watchist

This section will show my holdings

This section will show my assets and portfiolio

Figure 4: Krypton website Dashboard

Below Figure 5 describes the view of the Predictions page of the Krypton website.



- Cryptocurrency 1 with its icon
- Cryptocurrency 2 with its icon
- Cryptocurrency 3 with its icon
- Cryptocurrency 4 with its icon
- Cryptocurrency 5 with its icon
- Cryptocurrency 6 with its icon
- Cryptocurrency 7 with its icon
- Cryptocurrency 8 with its icon

- Upto 15 cryptocurrencies allowed to be selected in the watchlist at a time

Graph of the selected cryptocurrencies in the watchlist with their predicted trends in the future.

- Cryptocurrency 1 with its icon
- Cryptocurrency 2 with its icon
- Cryptocurrency 3 with its icon
- Cryptocurrency 4 with its icon
- Cryptocurrency 5 with its icon

- Upto 5 cryptocurrencies reccommended by us which the user should consider adding to the watchlist.

This section will show my holdings

This section will show the probable holdings suggested by us based on the market trends.

Figure 5: Prediction page of Krypton website

Figure 6 demonstrates the Holdings and wallet website page, where it visualizes the transaction history and my holdings components.

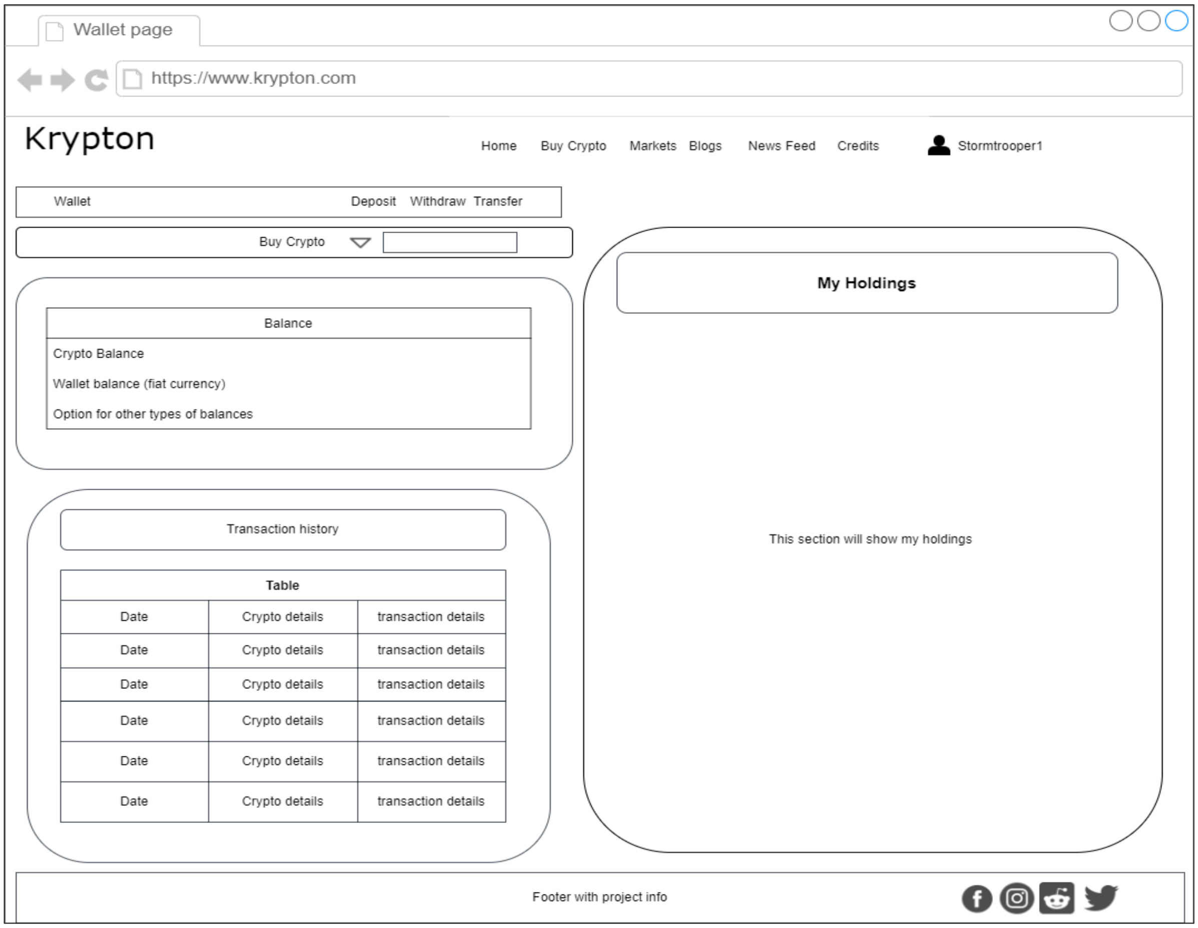


Table		
Date	Crypto details	transaction details
Date	Crypto details	transaction details
Date	Crypto details	transaction details
Date	Crypto details	transaction details
Date	Crypto details	transaction details
Date	Crypto details	transaction details

Figure 6: Wallet Krypton Website Page

Below Figure 7 shows the trading page of the Krypton website, where it demonstrates the buy and sell components of cryptocurrencies.

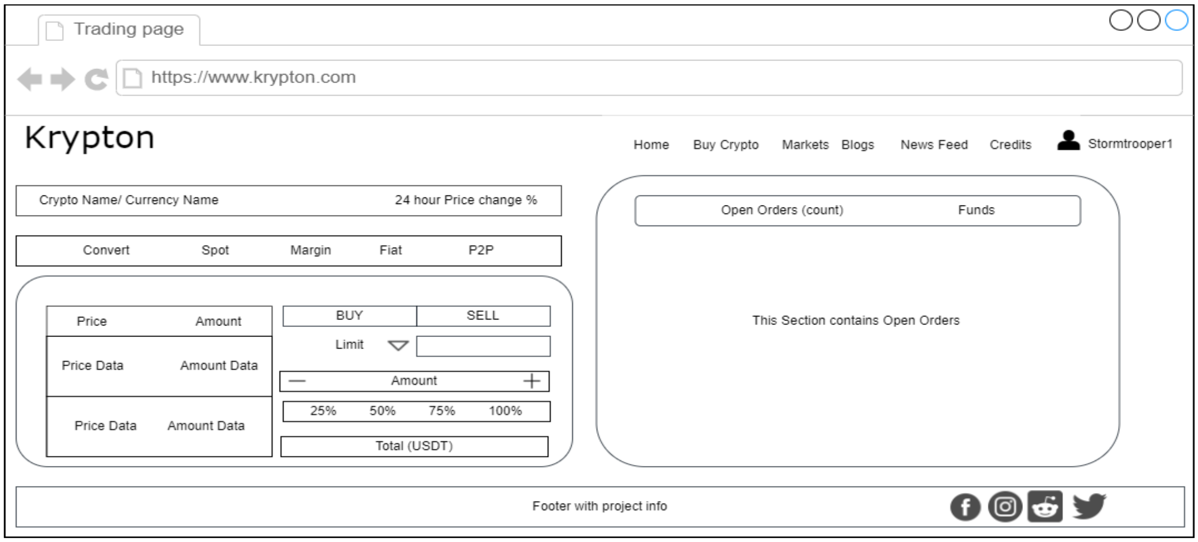


Figure 7: Trading Website Page

References

[1] “Cloud Computing,” Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Cloud_computing. [Accessed: February 10, 2022].

[2] “The 3 Cloud Computing Service Delivery Models Explained,” ExitCertified [Online]. Available: https://www.exitcertified.com/blog/cloud-computing-service-delivery-models. [Accessed: February 10, 2022].

[3] “What is SaaS, PaaS and IaaS? With examples,” StackOverflow [Online]. Available : https://stackoverflow.com/questions/16820336/what-is-saas-paas-and-iaas-with-examples. [Accessed: February 11, 2022].

[4] “What is IaaS?” Microsoft Azure [Online]. Available: https://azure.microsoft.com/en-ca/overview/what-is-iaas/. [Accessed: February 12, 2022].

[5] “17 Infrastructure Software as a Service (IaaS) Examples,” Web Hosting Secret Revealed [Online]. Available : https://www.webhostingsecretrevealed.net/blog/web-business-ideas/iaas-examples. [Accessed: February 12, 2022].

[6] “15 Popular Platform as a Service (PaaS) Examples,” Web Hosting Secret Revealed [Online]. Available https://www.webhostingsecretrevealed.net/blog/web-business-ideas/paas-examples. [Accessed: February 12, 2022].

[7] “Platform as a service,” Wikipedia [Online]. https://en.wikipedia.org/wiki/Platform_as_a_service. [Accessed: February 12, 2022].

[8] “Draw.io,” Draw.io [Online]. https://app.diagrams.net. [Accessed: February 12, 2022].

[9] “What is public cloud? Everything you need to know,” [Online]. https://www.techtarget.com/searchcloudcomputing/definition/public-cloud [Accessed: February 14, 2022].

[10] “Amazon web services,” [Online]. https://aws.amazon.com/ [Accessed: February 10, 2022].