# Chapter 01 – Implementation

## Chapter Overview

In this chapter the technological and the functional aspect of the system is defined. And the functionality of our system “Market Value Prediction System” is explained in detail and it’s explained in detail for the better understanding. A clear description of the technologies used and how these technologies are used in the web application is included.

Although we planned to develop a mobile application with the web application finally we decided to develop only the web application considering the users using our system. As this system is mostly used by the farmers and considering their financial status and educational level and the complexity in using a mobile application also we came into a conclusion of developing the web application with our fullest effort of the group members and providing the best output for the users of the system and providing them a very simple and a user-friendly system which can be used with a minimum amount of data and with a minimum expense when using our application. So we were able to implement a web based application having a higher accuracy level in the prediction and also with a simple and an easy interface which allows the farmers to use it without any difficulty and which allows the non-educated to use our system with ease without any difficulty. So we implemented it in the best and simplest way for it to be hundred percent user-friendly for the users.

## Overview of the Prototype

* The front end consists of mainly three sections. One for the data science component and one for the farmers section and the other for the sections of the buyer. Farmer will get access to the data table updated by the buyer side which is consisting of attributes Seller, Vegetable, Maximum Accepted stock and Available Stock, Price per (1KG). Where the farmer will be able to see the maximum stock and by consideration of the available stock the farmer will be able to request to supply goods.
* Buyer can accept the requests sent by the farmers who are the sellers and after accepting a notification will be sent to the farmers to notify them to supply the vegetables as requested.
* Front end is developed using React JS with the integration of AdminITE and material UI to get a better output for a simple and a user-friendly output for the farmer.

## Technology Selections

Mainly we focused on creating our project as a web application and developed the Frontend and back end for our web application mainly based on React Js. The table below shows the core technologies used in our application.

|  |  |
| --- | --- |
| Component | Technology Used |
| Web application (Front end) | React js(with intergration of AdminITE), Material UI, |
| Web application (Back end database) | Firebase Authentication, Firebase Realtime database. |
| Data Science Component | Phython 3.8, Flask API, Rain forest, Jupiter notebook, Google Collab |

Table 1 - Technologies Used

Table 2.3 - Technologies Used

With the development of technology the field of prediction systems are relatively developed and improved. An entire dissertation can be written about the development and software process we’re using within this project. This section provides a concise summary of the development process, highlighting the most salient points and parts in our project.

Our system is developed with a Feature Driven Methodology which will be similar to the waterfall methodology which is used mostly in the industry. We used this method to get the main benefits of the structured approach which is offered by the Feature Driven Development Methodology which could be combines with the use of the prototype used in the evolutionary development model. We used this method in getting the main benefits of the structured approach which is offered by the Feature Driven Methodology which will be combined with an approach of prototype used in the evolutionary development model.

We are using ReactJs with the integration of AdminITE for the simplicity of the application and for it to be more user friendly as this is mostly used by farmers and as they’re not much familiar with technology . Our backend of the application is used with Firebase real-time authentication as it should be updated real time as the stocks and prices of the products are updated with the buyers and sellers in our real-time database. Firebase authentication is used for the login of the application and it’s mostly used as two sections for the farmer and the seller/buyer. And by logging into the system the farmer will be able see the table of content where the farmer will be able to see the sellers/buyers with the vegetable prices.

The data science component is developed with Python

## Implementation of the Data Science Component

Data Science component is implemented where the farmer can get a predicted real-time price of a selected vegetable and the farmer or the user will need to enter the Vegetable type and the required time to get a prediction of a vegetable at that time according to the dataset we implemented in our data science component. Rain forest is used in implementing the machine learning model and the reason why we can’t build back propagation neural network is due to the insufficient amount of data available. In order to build the model we’ve used Jupiter notebook for the data science component.

## Implementation of the backend component

Our backend has two parts in this prototype. In first part there can be see authentication, crud operation that store data and display data, and the other part is rain forest algorithm is used to price prediction. As in the first part, we mentioned about authentication. We had used firebase authentication to identify the login users to this prototype and only the authorized users can log in to this system an proceed. And we use firebase real-time database for the crup operation that we use in the prototype to enter sale details and others. To connect the firebase database to our prototype, first we created a firebase web database and added the auto generated firebase API key to the prototype files.

As we are needing a real-time database as in we’re needing a database where the seller can update the tables within the time frame and also the tables are updated when the farmers supply items and the stocks are updated we thought of using Firebase real-time database which is having a strong user based security and also which provides a Firebase Authentication and as it can be used to allow access based on the user identity or with pattern which matches the data. And also we implanted a system where the farmers will be able to send a request if they like to supply vegetables to a required buyer and where the buyer will be able to accept the request within the backend if the buyer is is interested in the request of the farmer. And if the request is accepted by the buyer a notification will be sent to the farmer for the confirmation and also which confirms that the supply request is accepted. We have implemented a section to check the predicted price of a vegetable at a specific time that user gives. That part we implemented by connecting rain forest algorithm with our firebase real-time database.

## Implementation of the frontend component

This prototype technology contained react js(with the integration of AdminITE Bootstrap) and Material UI. In this prototype frontend, there are four parts,

* Log In

This login implemented using react js and material UI. This login use to log in for two users. Farmer and Seller can log in separately to each home page.

* Sign Up

Sign Up page also implemented using react js and material UI. Farmer and Seller can be created an account separately in this Sign-Up.

* Seller Home with a table

This part was implemented using the integration of the AdminITE Bootstrap template and react js. The table should contain Buyer Name, Vegetable, max stock, available Stock, and pricePerKg. Sellers should be able to add orders via a model. Sellers are able to delete or update their orders. You can embed two buttons for every table row. One for delete and one for the update.

* Farmer Home with a table

This farmer home was also implemented using the integration of the AdminITE Bootstrap template and react js and The table should contain Buyer Name, Vegetable, max stock, availableStock, and pricePer(1Kg). If the user is a farmer above two buttons should hide and display a button called Accept Order.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Seller | Vegetable | Maximum Stock Accepted | Available Stock | Price per (1Kg) |
|  |  |  |  |  |

Table 3 - Table used in Farmer side Interface

## 1.7 Deployments CI-CD Pipeline

## 1.8 Chapter Summery

The above chapter explains the implementation process in advance and a detailed description of the technologies, the interface and about the system is described. Although at first we planned in implementing two sections as a mobile application and web application we limited it only to a web application by considering the educational and financial status of the users considering the expenses and the data used for the usage of the applications. And designed the web application with a simple and an easy user interface for the users to get a better use of the application and for the users to use the application without any prior understanding or an idea about the application. Data science component is also described separately with the implementation and the user interface of the application is explained. Simply the project is explained and the functionalities are described in detail. Testing criteria’s will be explained in the next chapter.

# Chapter 02 – Testing

## 2.1 Chapter Overview

From the previous chapter the implementation phase of the Market Value prediction system is discussed. This chapter will be focused on the testing phase of our Market Value Prediction System. After the goals and objectives of testing are discussed, the testing criteria will be discussed in detail. Testing of Functional and Non Functional requirements will be discussed after that to make sure that it will meet all the requirements for the implementation report and to ensure it meets all the required standards. Description about the limitations faced during the testing process with an evaluation of the testing results are concluded in this chapter.

## 2.2 Goals and objectives of testing

We carried out the testing phase to verify the prototype functions which are implemented according to the requirements which are identified during the requirement elicitation phase.

The objectives of the testing phase for the implemented prototype is shown below.

* To identify the errors and bugs which are possible in the implemented application and to ensure that they are clearly fixed before the final product is deployed.
* To improve and enhance the application we designed based on the test results.
* To test and verify the functional requirements of the web application as proposed.
* To test and verify the non-functional requirements of the web application as proposed.

## 

## 2.3 Testing Criteria

## 2.4 Testing functional requirements

## 2.5 Testing non-functional requirements

## 2.6 Unit testing

## 2.7 Performance testing

## 2.8 Usability testing

## 2.9 Compatibility testing

## 2.10 Chapter Summary

This chapter initially contained the test results for the functions of the system. Next it showed  
the performance testing results for the client side and server side of the mobile application. It  
described the performance results for the raspberry pi system as well.

# Chapter 3 – Evaluation

## 4.1 Chapter Overview

## 4.2 Achievements of aims and objectives

## 4.3 Legal, social, ethical and professional issues

## 4.4 Limitations of the research

## 4.5 Future enhancements

## 4.6 Extra work (Competitions, research papers, etc)

## 4.7 Concluding remarks