**CHAPTER 6**

**…………………………………………………………………………………………………**

**Implementation**

**Chapter 6: Implementation**

**6.1 Introduction**

In Computer Science, an implementation is a realization of technical specification or algorithms as a program, software component, or other computer system through computer programming and deployment. Many implementations may exist for a given specification or standard. For example, web browsers contain implementations of World Wide Web Consortium-recommended specifications, and software development tools contain implementations of programming languages. Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. To implement a project means to carry out activities proposed in the application form with the aim to achieve project objectives and deliver results and outputs. Its success depends on many internal and external factors. Some of the most important ones are a very well organized project team and effective monitoring of project progress and related expenditures. This system is implemented based on a JavaScript Library, React. We created a web app with the help of JSX to view the user interface. And we use the JavaScript programming language and firebase for backend to communicate with the user.

**6.2 Application Architecture**

Application architecture is the process of defining the framework of an organization’s application solutions. Application architecture refers to the high-level structures of a software system, the discipline of creating such structures, and the documentation of these structures. Scalability is one of the most important things for a mobile application it has to be written to support growth from the very beginning. “Architecting” an app at this phase is crucial and cloud prevents you from having to completely rewrite it when it needs to grow. This phase happens parallel to design so someone is ready to kick off the development phase next.

A web application has two parts:

● The Front End

● The Back End

**Front-end**

* Web App developed with React consists of various components.
* It is all about what a user is able to see.
* It may be buttons, images, input fields etc.
* To design these components in React, we need JSX and JavaScript programming languages. Therefore, these programs are called front-end programming.
* There is at least one JS file for each component.

**Back-end**

* Backend of anything is not visible to the general user or administrator. It can be a piece of code or a program running on the server machine to serve the user's needs.
* Firebase is used in the backend to communicate with the user and for code execution.

**6.3 Process of Implementation**

Mainly, our web application has several sites for the users. Some of the featured screens are given below:

1. Login Page

2**.** Sign-up page

3. Home Page

4. Profile Page

5. Product Page

6. Services

7. Management of Products

8. User Dashboard

9. Review Page

**6.3.1 Login Page**

Login page is one of the pages of our web application. Users will be automatically redirected to the Login Page if the user is not authenticated and the particular requires authentication. As shown in fig 6.1 it contains two input fields Email Address and Password and a Login button. The Login form also requires form validation. It also contains a “Register” button in case the user is not registered.

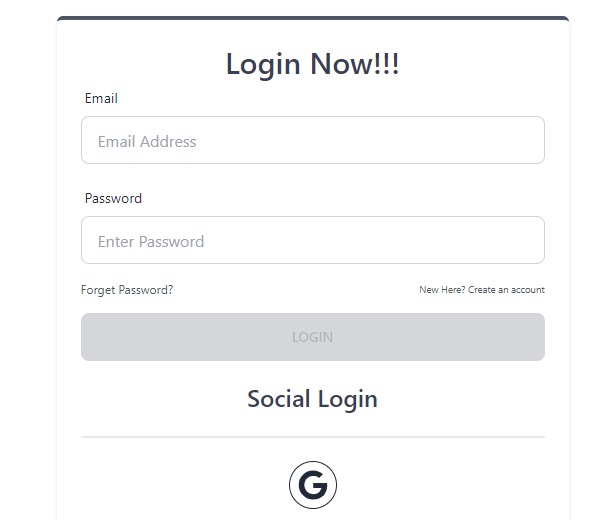
****

Fig 6.1: Login page

**6.3.2 Sign-up page**

If any user wants to register for any service he will sign-up to the page using email, password, and phone no by giving his basic information; this is given below:

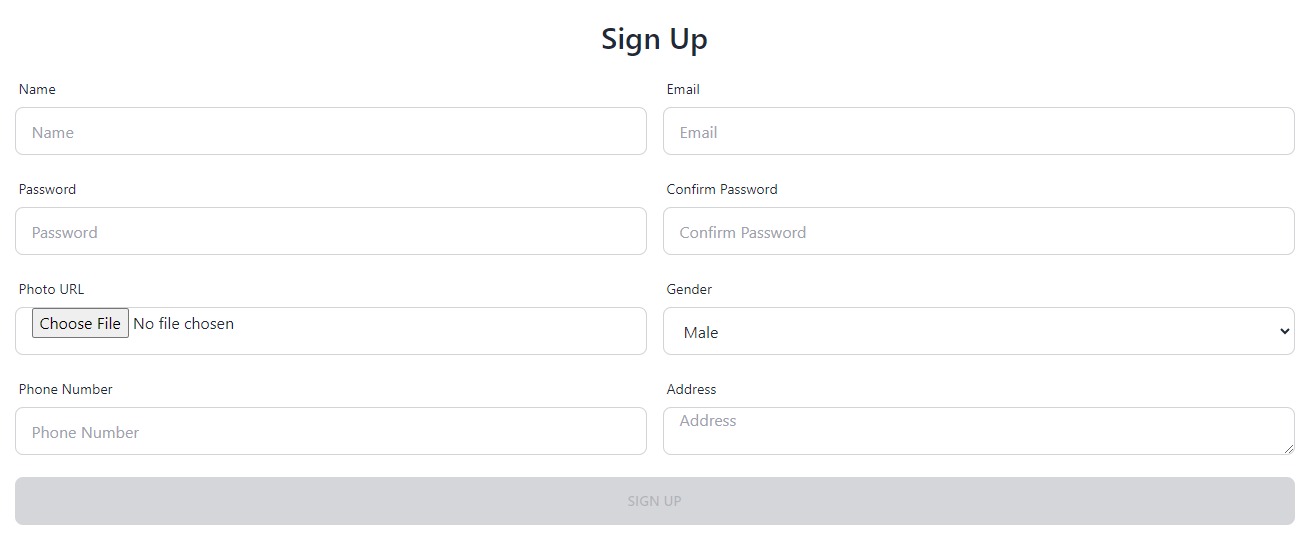
****

Fig 6.2: Sign-up Page

**6.3.3 Home Page**

Home page is the first page of the web application. The user doesn't have to be logged in to view this page. It contains a message briefly describing the overall functionality of the web application (shown in fig 6.3: Home Page). From the home page, user can navigate to any page of the web application.

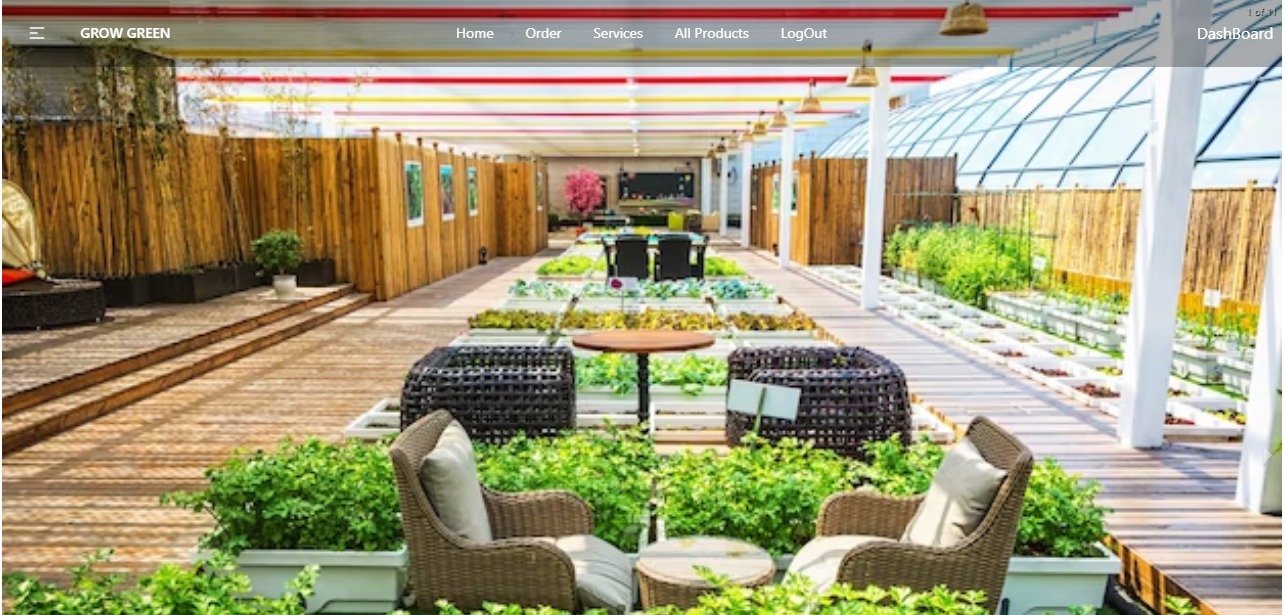
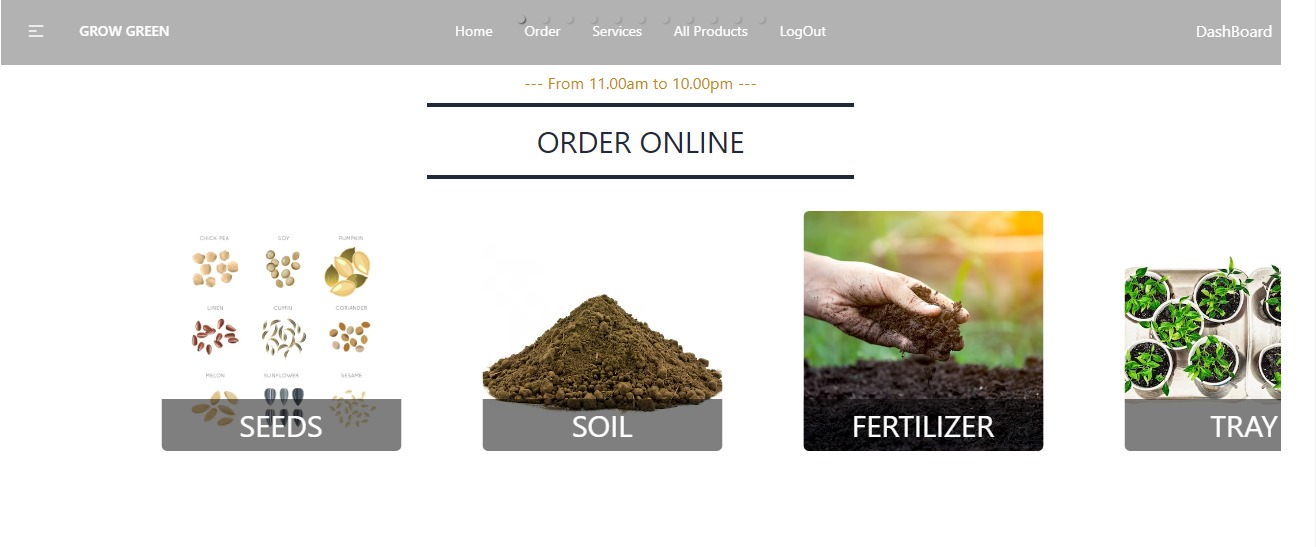


Fig 6.3: Home Page

 Fig 6.4: Profile Page

**6.3.4 Product Page**

It contains product details about plants, seeds, gardening accessories and other things. The fig 6.4 contains product details:

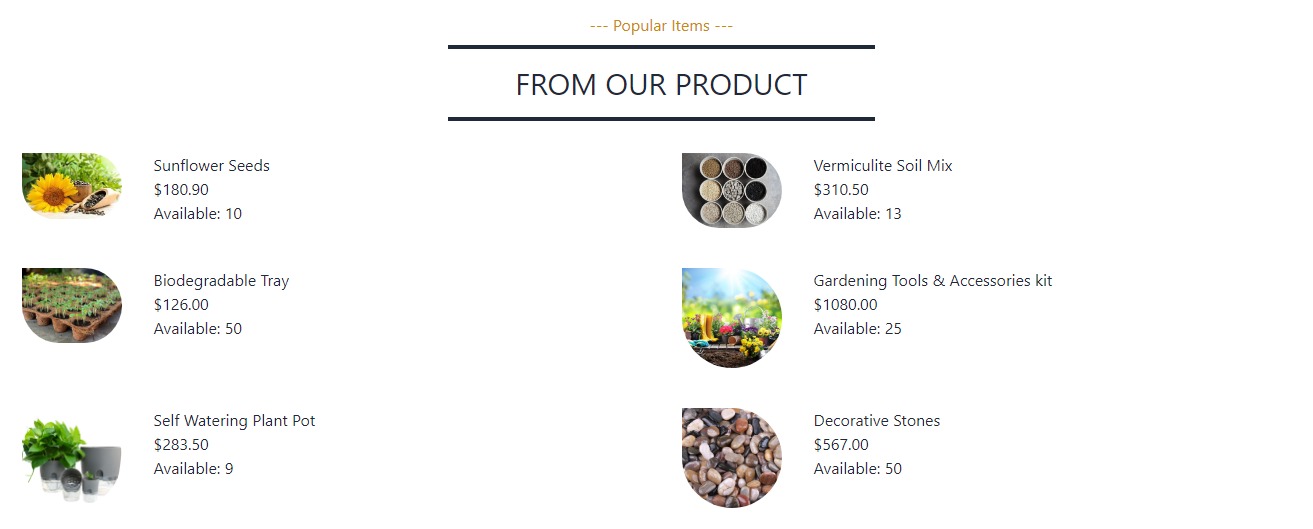


Fig 6.5: Product Page

**6.3.5 Services**

It contains service details about our web application. Fig 6.5 contains image of our services:

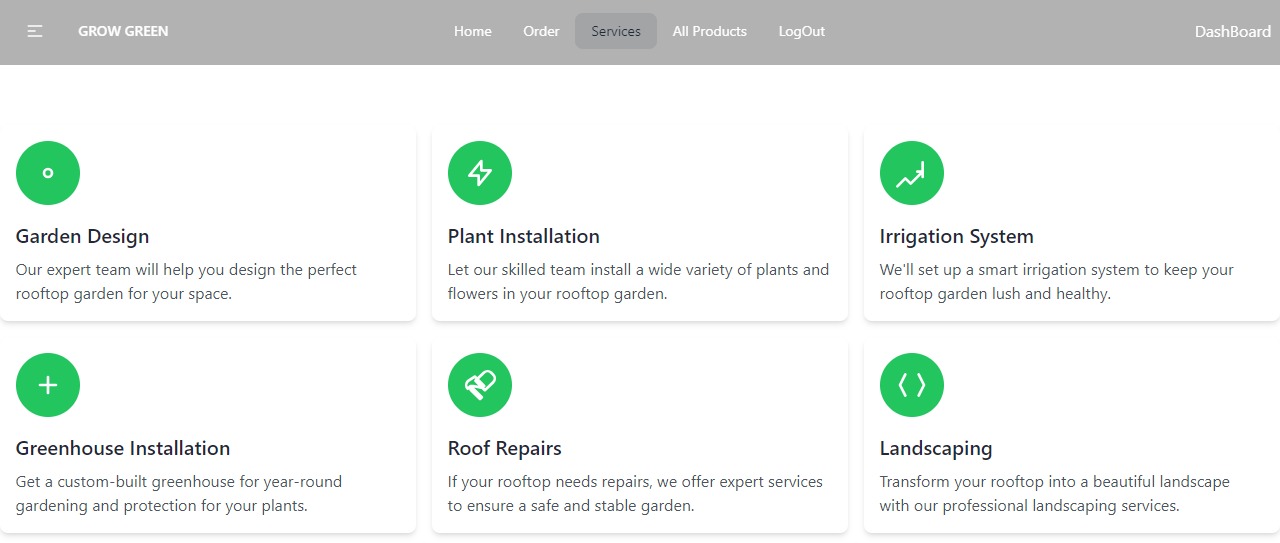
****

Fig 6.6: Services

**6.3.6 Management of Products**

The services of this web application will be managed by Admin and admin will give access to salesman to add product on craft. No of available products, price and other information’s will be managed by this section:

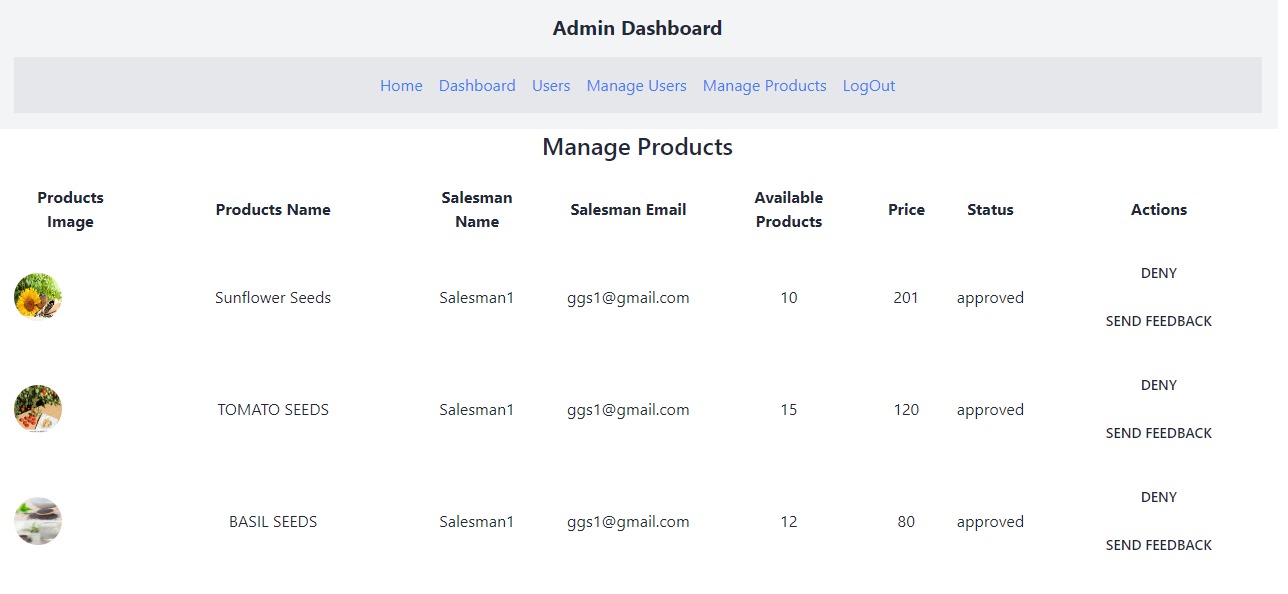


Fig 6.7: Manage Products

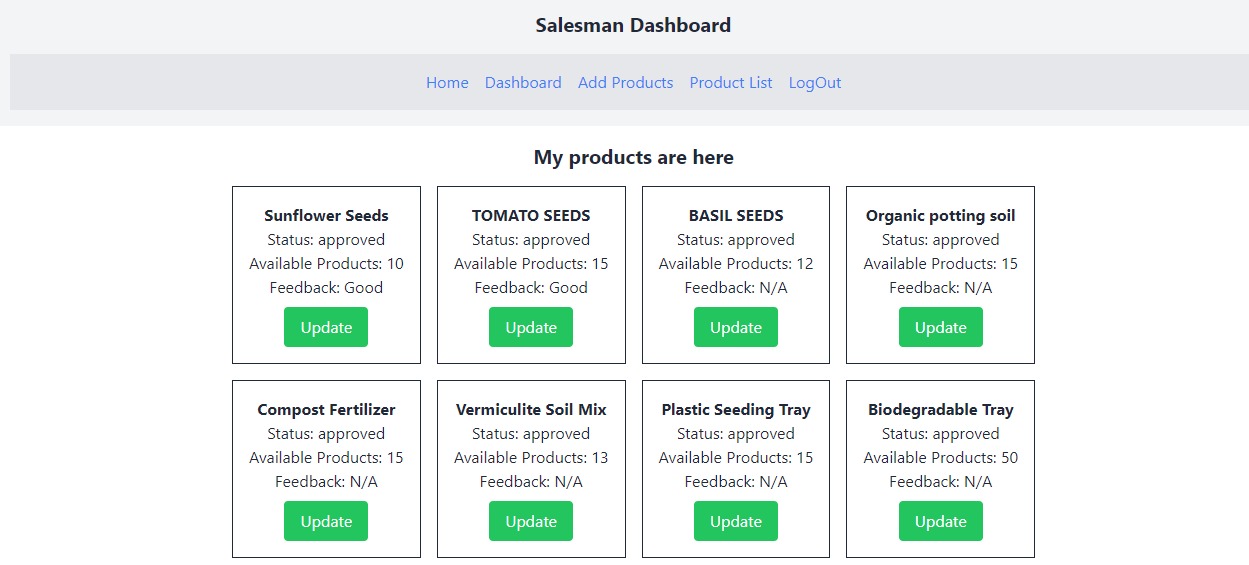


Fig 6.8: Management of Services

**6.3.7 User Dashboard**

It will contain user information when user will choose his product, will do payment and cart no and all this things. The Following figures will show the user dashboard information details in image:

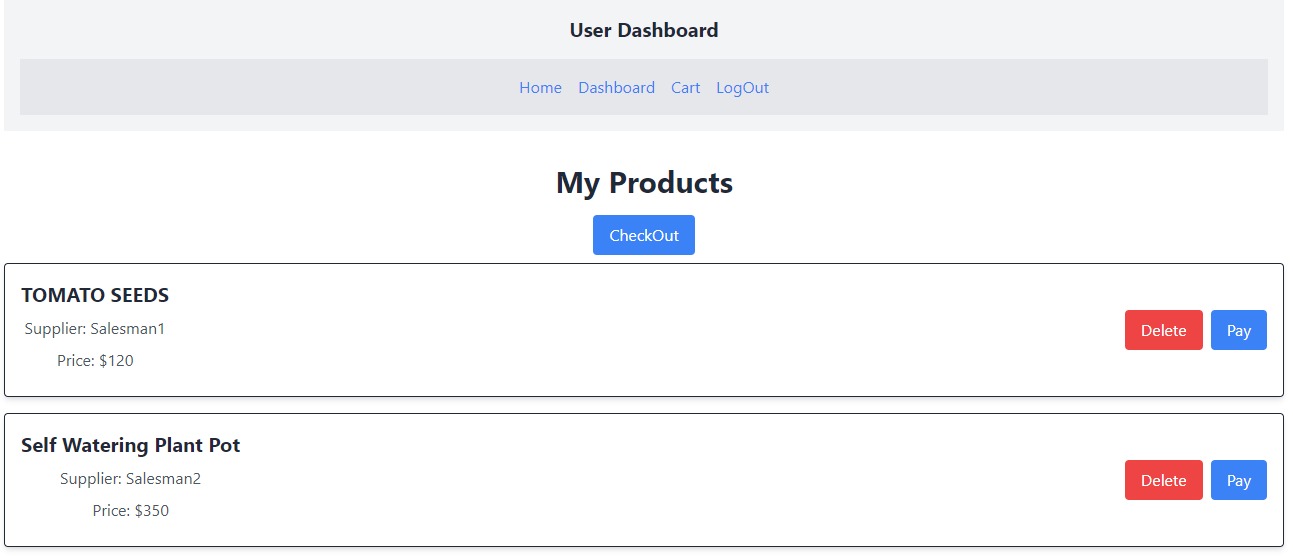


Fig 6.9: Product information

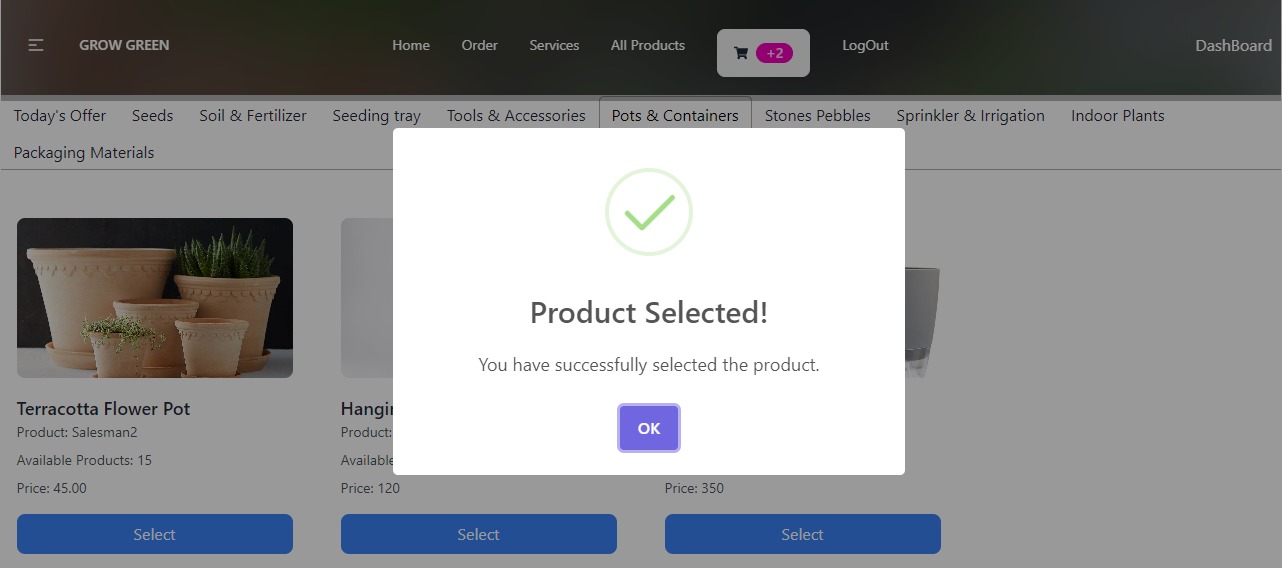


Fig 6.10: Product Selection

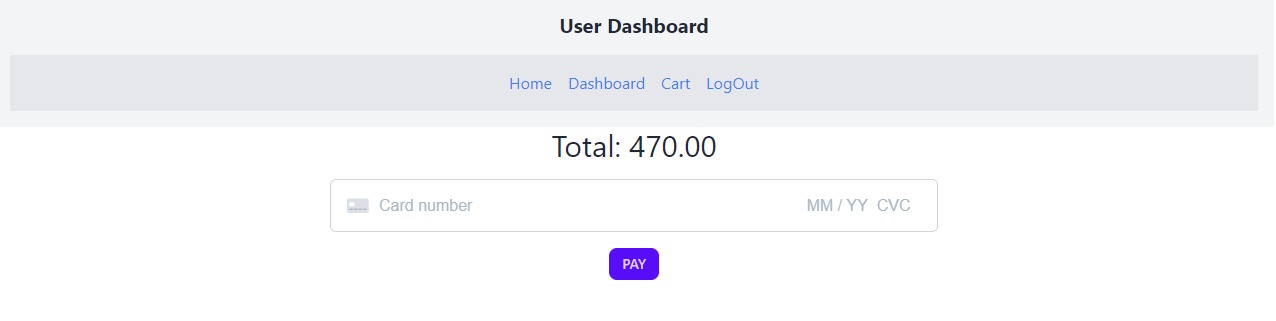


Fig 6.11: Payment System

**6.3.8 Review Page**

It contains review of customers. They can also provide rating to our web application. Testimonials will help to enrich our site. Figure 6.12 will show this:

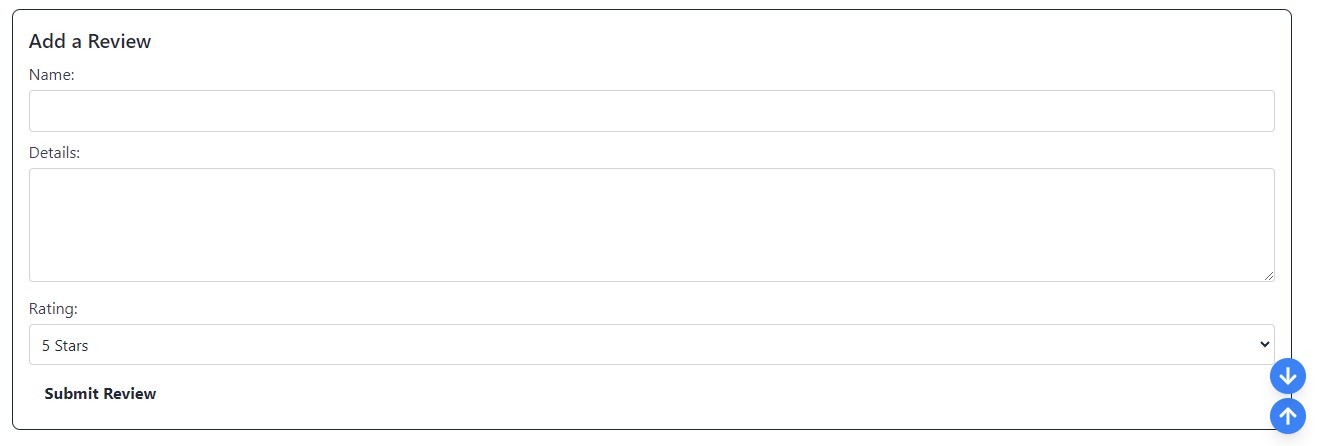


Fig 6.12 Review Page

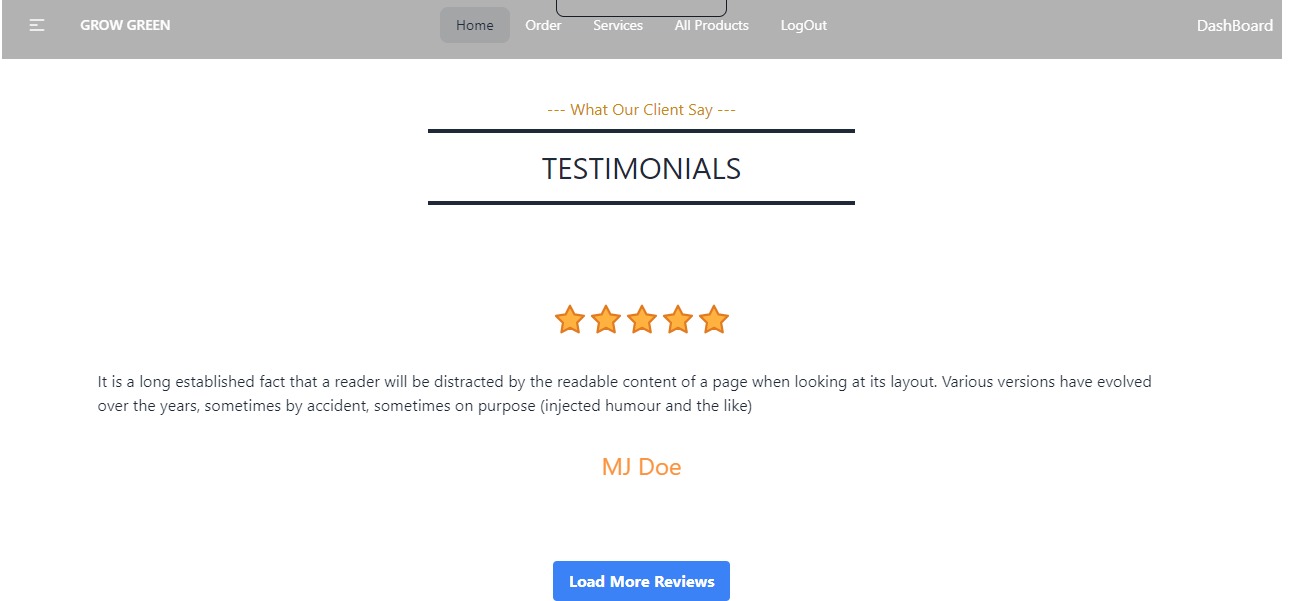


Fig 6.13 Rating Page

**6.4 Throughput Analysis**

To conduct a throughput analysis of an online platform for a rooftop gardening project, we would need to gather data on various aspects, such as the number of users, orders, and interactions on the platform. Analyzing this data will help to understand the platform's performance and identify potential bottlenecks or areas for improvement.

Some key metrics to consider include:

**Website Traffic**: Measure the number of visitors and unique users over a specific period.

**Conversion Rate:** Calculate the percentage of website visitors who make a purchase or take a specific desired action.

**Order Processing Time:** Analyze how long it takes from order placement to fulfillment.

**Server Response Time:** Measure the time it takes for the platform to respond to user requests.

**User Engagement:** Assess the level of user interaction with the platform through comments, reviews, or social media shares.

**Page Load Speed:** Check the time it takes for web pages to load, as slow loading times can affect user experience.

**Customer Support Response Time:** Evaluate how quickly customer inquiries or issues are addressed.

**Inventory Management:** Monitor how well the platform manages its inventory and avoids stock outs.

By analyzing these metrics, we can gain insights into the platform's efficiency, user satisfaction, and potential areas for optimization. Remember that data collection and analysis should be an ongoing process to ensure continuous improvement in the platform's performance.

**6.5 Complexity Analysis**

Complexity analysis of the web-based platform for rooftop gardening and management into various components:

**Front-end Complexity:**

* The use of HTML, CSS, and JavaScript will determine the complexity of the user interface and user experience.
* Responsiveness and cross-browser compatibility should be considered to ensure a smooth experience across different devices and browsers.

**Back-end Complexity**:

* Implementing a back-end server to handle user data, product inventory, and gardening project details will require careful planning.
* Database integration and management to store user information, order history, and gardening project details will add to the complexity.

**User Authentication and Authorization:**

Implementing secure user authentication and authorization mechanisms is crucial to safeguard user data and maintain privacy.

**E-commerce Functionality:**

Integrating a secure payment gateway for handling transactions will add to the complexity of the platform.

**Inventory Management:**

Implementing a system to manage plant and gardening accessory inventory, including stock updates and real-time availability information, can be challenging.

**Order Processing:**

Handling order placement, tracking, and fulfillment requires robust backend processing and communication with suppliers.

**Gardener Management:**

Creating a system for managing gardeners, their availability, and matching them with appropriate gardening projects will require careful planning.

**Communication:**

Implementing a communication system to connect customers with gardeners and handling messaging will add complexity to the platform.

**Security Considerations:**

Ensuring the platform is secure from potential attacks, such as SQL injection and cross-site scripting, is vital.

**Scalability:**

Designing the platform with scalability in mind will allow it to handle an increasing number of users and products.

**Performance:**

Optimizing the platform's performance to reduce load times and ensure smooth user interactions is crucial for a positive user experience**.**

**6.6 Conclusion**

This chapter gives a complete overview of our system. A detailed analysis of the layout of the system, how the web-application looks like and the other details.

**Chapter 7**

**…………………………………………..**

**Conclusion & Future Work**

**Chapter 7: Conclusion & Future work**

**7.1 Introduction**

The Project (Online platform for rooftop gardening and management services) presents an innovative and sustainable solution for urban dwellers to transform their rooftops into green spaces. the platform offers an intuitive and user-friendly interface, enabling customers to explore a wide variety of plants and gardening accessories for their rooftop projects.

The inclusion of a gardener contact feature enhances the platform's utility by connecting users with skilled professionals who can assist in implementing and maintaining their rooftop gardens. Through this platform, users can not only access a comprehensive selection of gardening products but also find reliable expertise to guide them through the gardening process.

**7.2 Future Work**

The nature of work is changing; implementing this project can involve several areas of improvement and expansion. Here are some potential avenues for further development:

**Enhanced User Experience**: Continuously improve the user interface and user experience to make it more intuitive and visually appealing. Implement user feedback mechanisms to gather insights for further improvements.

**Mobile App Development:** Consider developing a mobile app version of the platform to cater to users who prefer accessing services on their Smartphone’s or tablets.

**Integration of Augmented Reality (AR):** Integrate AR technology to allow users to virtually visualize how different plants and landscaping ideas would look on their rooftops before making a decision.

**Community Engagement:** Create a community platform within the application where users can share their rooftop gardening experiences, tips, and collaborate on projects.

**Gratification and Rewards:** Implement gratification elements to engage users and offer rewards for achieving gardening milestones or referring friends to the platform.

**Data Analytics and Insights:** Utilize data analytics to gather insights on user behavior, preferences, and trends, which can inform decision-making and future improvements.

Continually evolving the web-based application with these future developments will help the platform remain competitive, attract more users, and contribute to the advancement of rooftop gardening and management services.

**7.3 Conclusion**

Our project is only a humble venture to satisfy the needs to manage gardening work. Several user-friendly services have also been adopted .GrowGreen is a web-based rooftop gardening and management services platform serves as a valuable resource, encouraging users to embrace rooftop gardening as a sustainable and aesthetic way to utilize urban spaces.