An Undergraduate Internship on "SwapCamp" web application

By

Khadija Rejjaoui

Student ID: **1830131**

Summer, 2021

Supervisor:

Sheikh Abujar

Lecturer

Department of Computer Science & Engineering

Independent University, Bangladesh

September 6, 2021

Dissertation submitted in partial fulfillment for the degree of Bachelor of Science in Computer Science

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

This is to certify that the report is completed, by me Khadija Rejjaoui (ID:1830131), submitted in partial fulfillment of the requirement for the Degree of Computer Science and Engineering from Independent University, Bangladesh (IUB). It has been completed under the guidance of Sheikh Abujar. I also certify that all my work is original, and it is The fruit of what I learned during my university life and training. All sources of information used in this project and report have been duly acknowledged.

Signature	Date
Khadija Rejjaoui	
Name	

Acknowledgement

I would first like to express my deepest gratitude to Almighty God for giving me the strength and composure to finish the job on time. It has been a great honor to work for Techdojo as an intern. I have received a lot of support and encouragement from the people of Techdojo who have years of experience in software development.I would like to thank the members of Techdojo for taking their valuable time and knowledge, which was necessary to complete this project. I would also like to express my sincere gratitude to my instructor Mr. Sheikh Abujar Lecturer, Department of Computer Science and Engineering, Independent University, Bangladesh (IUB), for his continuous support, patience, motivation, and tremendous knowledge. His guidance helped me plan and write this report. I couldn't have imagined a better advisor and mentor like him. I also would like to express my deepest gratitude to my outside supervisor and mentor, Ms. Shama Hoque, for allowing me to be a part of this organization. Her support and pioneering ability was the driving force for this project. My gratitude also extends to all other employees of Techdojo who helped me learn a lot during the process of developing my skills and make me fit for the environment. Many thanks to the developers involved in this project for their time, effort, and expert skills. Finally, I proudly acknowledge the great sacrifices, good wishes, moral support, fruitful advice, inspiration, and encouragement from my family, relatives, and friends.

Letter of Transmittal

September 6, 2021
Mr. Sheikh Abujar
Lecturer
Department of Computer Science and Engineering Independent University, Bangladesh.

Subject: Internship Report submission Summer, 2021.

Dear Sir,

It is with great pleasure that I present my internship report on the SwapCamp web application. This report contains all the information regarding the work that I have done in the past three months. While preparing it, I have tried to make it communicative and comprehensive.

I did my internship in "Tech dojo limited " which is a Software Company. The main goal of my internship was to apply what I learned during these years at university. I didn't want to continue learning without knowing if what I learned was paying off. Moreover, I wanted to know exactly what I should learn to keep up with the job market.

During my internship at Techdojo Limited, I had to learn and adapt to advanced technologies used in different situations, and to be able to apply them in real-life projects.

I would like to thank you so much for all your guidance and support. I hope and pray that this report will meet all the requirements and live up to your expectations.

Sincerely,
Khadija Rejjaoui, 1830131

Evaluation Committee

.....

Signature

.....

Name

.....

Supervisor

.....

Signature

.....

Name

.....

Internal Examiner

.....

Signature

.....

Name

.....

External Examiner

.....

Signature

.....

Name

.....

Convener

Abstract

Excessive productivity of products leads to environmental destruction and pollution. The high cost of living and prices deprive many people of getting products they desire and deprive others of saving their money. Although there are many sites that enable people to exchange or sell their products, they rely only on chatting without any supervision, which causes a lot of problems such as scams. To deal with such a situation, a group of developers at Techdojo; Including me, decided to work on a web application where people can exchange products with each other indirectly and without any problem.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview/Background of the Work

The environment and everything inside it are important to every human being, so they must be preserved, but excessive production causes us to lose natural resources, such as water, trees, etc. Moreover, every human being in this world; dreams of getting the products he desires, whether he is rich or poor. So, to let people get anything they want and preserve the environment, the idea came to create a secure swap site called SwapCamp that allows people to get whatever they want in exchange for something they have. That will enable people to save their money and use them for something else.

1.2 Objectives

The main objective of this project:

1- Save the environment.

2- Help people to save their money.

3- Help people to get their desired items cheaply and easily.

4- Create a secure site that prevents users from falling for scams and fraud.

1.3 Scopes

Features that will available to the users after the development of this project:

For customer:

- Sign In/ Registration.
- Add swap request.
- Check categories and products.
- Check single product info.
- Search for a product.
- Filter by category/price.
- Sort by newest/ low to high/ high to low.
- Add to Cart.
- Place Order.
- Check Orders.
- Update Profile.

For Admin:

- Sign In.
- Check swap request.
- Add categories and products.
- Delete/ Edit products.
- Update user info including credits.
- Delete users.
- Check Orders Info.
- Delete Orders.
- Make order as delivered.
- Update his profile.

Chapter 2

Literature Review

2.1 Problem Statement

There are a lot of problems that can result from the current swap websites. Among these problems, we have fraud because many of them allow users to post used stuff; the exchanging process is done through a conversation between users without any supervision where users decide where to meet (offline procedure). Another problem is that some users may post something out of context, and sometimes it's hard for the administrator to control and check all the posts. In addition, not all users get a chance to swap their stuff. Given these issues, we have to design and implement a web application that will solve these issues.

2.2 Relationship with Undergraduate Studies

This project has a deep relationship with my undergraduate courses because each software-based course taught me different skills that helped me develop this project. Some of the courses are:

• CSE-214 Object-Oriented Programming: This was one of the Cornerstone courses that gave me an idea and taught me how to handle and manipulate complex arrays, objects, classes, and write code in the modularized form, etc. This knowledge was a big help to write the code in a better and clean way, find bugs, and track them easily.

• CSE-303 Database Management: This course was a great help to develop this project as it does not only teach about the database and how to use it, it also introduces how to plan and design a project, like drawing a Rich picture, six-element analysis, schemas, etc. The database is an important part of any software and in this project as well. As It helps to store, organize and manage data in a practical and systematic way.

• CSE-307 System Analysis: This course was full of information about the methodologies, techniques, tools, and perspectives essential for systems analysts. The techniques and tools that I have gained from this course helped me to implement a user-friendly UI that can be understandable and easier for non-technical people (Normal users). Moreover, I could draw some UI diagrams that helped me in designing this system.

• CSE-309 Web Application: In this course, I learned how to create web applications using HTML, CSS, MongoDB, JavaScript, Express, and Nodejs. It would have been more difficult to complete this project without this course as the project was implemented using the MERN stack (MongoDB, Express, React, Nodejs).

2.3 Related works

Examples of some similar applications existing in the market:

• Swap:

Figure 2.1: Swap application interface.

SWAP is a combination of e-commerce and re-commerce marketplace in Bangladesh where customers can purchase anything by personalized orders as well as sell their unneeded/surplus products such as smartphones, laptops, appliances, vehicles, etc. and exchange also.

• Gumtree Swap Shop:

Figure 2.2: Gumtree Swap Shop logo.

Gumtree is a place for buying and selling second-hand stuff, but they also have a Swap Shop where you can advertise things you want to exchange for other items.

Chapter 3

Project Management & Financing

3.1 Work Breakdown Structure

WBS is a common productivity technique used to make the work more manageable and approachable by breaking down the project into smaller tasks [1]. In the SwapCamp project, we have produced a WBS that will help us make things faster and more efficient as it provides the basis for planning and managing project schedules, costs, and changes.

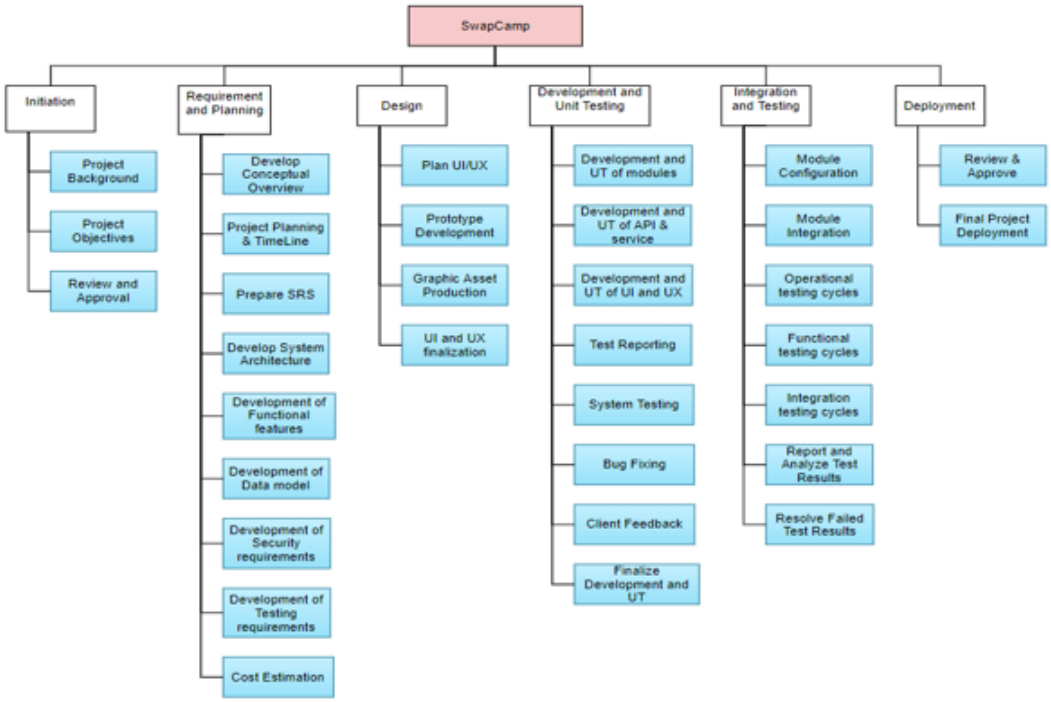

Figure 3.1: WBS of SwapCamp

3.2 Process/Activity wise Time Distribution

For Each task we have described in the WBS Diagram for SwapCamp, we made a time allocation. The Table below shows them in detail:

Task	Days	Work Percentage
Requirement Analysis	15	20
Design Layout	10	10
Development	25	40
User Acceptance Testing (UAT)	15	15
Deployment	15	15
Total	80	100

Table 3.1: task wise time allocation

3.3 Gantt Chart

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart, we have the list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflect the start date, duration, and end date of that activity [2].

The Gantt chart was useful for monitoring the progress of this project once it was started. it helps to get a clearer view of what should have been achieved in a given time frame.

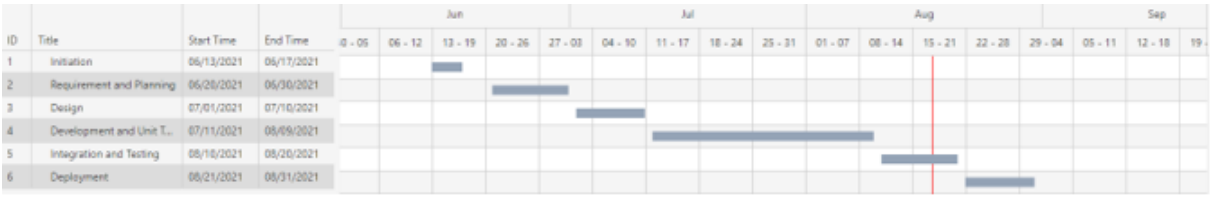

Figure 3.2: SwapCamp Gantt Chart Project Planner

3.4 Estimated Costing

The costs given below was estimated by the company.

Work Distribution	Costing
Development	$14,000
Domain Name	$9 / year
Hosting	$1,000 / year
Website maintenance	$5,000

Table 3.2: Estimated Costing

Chapter 4

Methodology

4.1 The software development methodology

The software development methodology [3] is a process or series of processes used in software development. It is also called a software development life cycle (SDLC). It is a process that produces software with the highest quality and lowest cost in the shortest time possible. This methodology concentrates on the following stages of software development:

- Requirement analysis
- Planning
- Software design such as architectural design
- Software development
- Testing
- Deployment

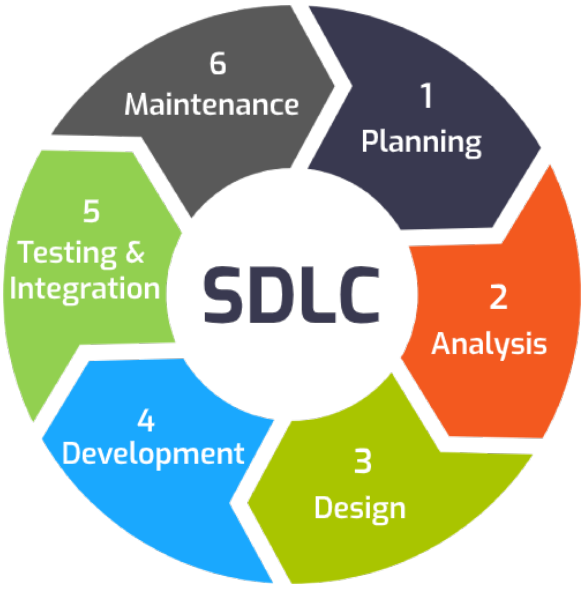

Figure 4.1: Software Development Life Cycle(SDLC)

There are several system development methodologies or models that are used in developments, some of them are:

- Waterfall Model.
- Prototyping.
- Agile.

Extreme Programming (XP) Methodology

As for the methodology, Techdojo Limited developers follow the latest in system development methodologies, which is the Agile method. To be more precise, we follow the Extreme Programming (XP) method [4], which is an Agile software development methodology.It aims to produce higher quality software.

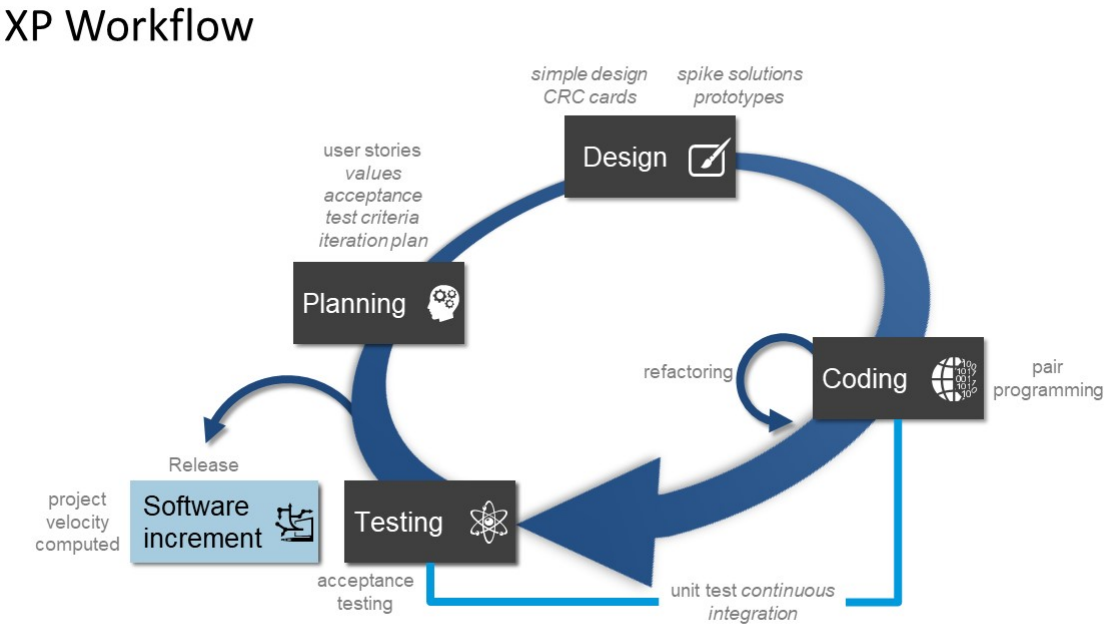

Figure 4.2: XP workflow

XP has simple rules that are based on 5 values. Those values are Simplicity, Communication, Feedback, Courage, and Respect [5].

• Simplicity: Find the simplest things that can work. By this, we mean removing all the complexity and waste in development by doing only the necessary work.

• Communication: transparent, face-to-face, frequent, online meetings are best for a project team. Co-located when possible, you do not for an email for three days, for example.

• Feedback: Feedback on the project and the team progress as well. The team can identify areas for improvement and revise their practices.

• Courage: The course to speak up, to put ego aside, be vulnerable. Code is visible to everyone all the time in the XP project. Courage to put your work out there for others to review, inspect, and edit.

Respect: The team respects each others' ideas, cultures, values, and how they work to get the results. Quality and the success or failure of the project are everyone's responsibility.

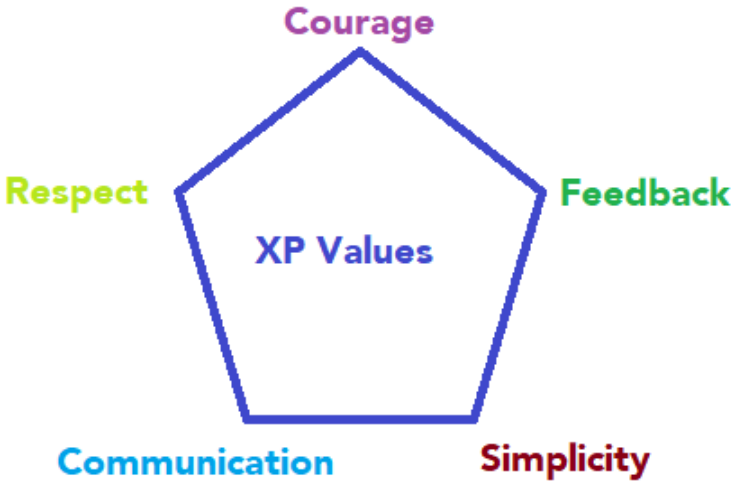

Figure 4.3: Values of Extreme Programming (XP) Methodology.

Chapter 5

Body of the Project

5.1 Work Description

SwapCamp is an online platform that allows users to swap their used or unwanted stuff with new ones. I was assigned several tasks in this project, the first of which was to find a way for the swap to be online and not expose any user to fraud or any problem, because most of the similar sites make this process take place through a chat between the exchangers, in order to determine a suitable place for them to do the swap, but this could put their lives at risk. The solution I found to this problem is to give users a number of credits. These credits are swap credits, which members receive in exchange for items they send to the company. With these credits, users can order items from the site. The amount awarded for each item depends on its condition and value. My second task was to implement this idea using the MERN stack (MongoDB, Express, React, NodeJS) and the necessary libraries. Additionally, I had to allow users to add items to their cart, place orders, and check their order history. Furthermore, give the admin the ability to add, delete and edit products, check and delete orders, edit and delete users. Last but not least, I had to design the home page using react and styled-components.

5.2 System Analysis

5.2.1 Six Element Analysis

Process	System roles					
	Human	Non-Computer Hardware	Computing Hardware	Software	Database	**Communicationand Network**
Swap	**A-Customer** 1- The customer will create an account. 2- Enter information about the item that he has. 3- He will get a call from the delivery man to collect the item. 4- Customer will give the item to the delivery man. 5- The customer will receive credits on his account depending on the item value and conditions. **B- Delivery man** 1- Will call the customer to get the item from him. 2- He will collect the item from the customer. **C- Admin** 1- He will check the swap request. 2- He will inform the delivery man. 3- He will add the product to the store. 4- He will update customer credits.	**A- bag/ box:** 1- where delivery man put the item.	**A- computer/ tablet:** 1- used by the admin and customer to get connect to the web application. **B- Phone:** 1- used by the customer to get connected to the website. 2- used by the delivery man to call the customer.	**A- SwapCamp web application:** 1- has the interface where the user enter data related to his item, and where the admin saw the swap request.	**A-Mongo DB** 1- is the database used to store data in SwapCamp	**A- Internet:** 1- SwapCamp is a web-based application so internet is required to access.

Place Order	**A- Customer** 1- Login to the website. 2- View products. 3- Add products to the cart. 4- Enter shipment address information.5- If the customer has sufficient credits in his account his order will be placed, if not he will need to submit some swap items. **B- Admin** 1- will see the customer order.	**N/A**	**A- computer/ tablet:** 1- used to get connect to the web application. **B- Phone:** 1- used by the customer to get connected to the website. 2- used by the delivery man to call the customer.	**A- SwapCamp web application:** 1- has the interface where admin view the orders and customer can place orders.	**A-Mongo DB** 1- is the database used to store data in SwapCamp	**A- Internet:** 1- SwapCamp is web-based application so internet is required to access.
Delivery	**A- Admin:** 1- will inform the specific department to ready the order. **B- Delivery man.** 1- will take the order to the doorstep of the customer. **C- Customer** 1- will receive the order.	**A- bag/ box:** 1- where delivery man put the order.	**A- computer/ tablet:** 1- used to get connect to the web application. **B- Phone:** 1- used by the customer to get connected to the website. 2- used by the delivery man to call the customer.	**A-SwapCamp web application:** 1- has the interface where admin view the orders and customer can check order history.	**A-Mongo DB** 1- is the database used to store data in SwapCamp.	**A- Internet:** 1- SwapCamp is web-based application so internet is required to access.

Table 5.1: Six element analysis

5.2.2 Feasibility Analysis

A feasibility study [6] is a study to evaluate the feasibility of a proposed project or system. It is one of stage among important four stages of the Software Project Management Process. As the name suggests feasibility study is the feasibility analysis, or it is a measure of the software product in terms of how beneficial product development will be for the organization from a practical point of view. The feasibility study is carried out based on many purposes to analyze whether software product will be right in terms of development, implantation, the contribution of the project to the organization, etc. Main elements of Feasibility Study are:

• Technical feasibility: evaluates the available infrastructure (such as hardware and software) and technologies needed to meet the consumer needs of software under time and budget constraints. It reports whether there exists correct required resources and tech-

nologies which will be used for project development. Along with this, the feasibility study also analyzes the technical skills and capabilities of the technical team, existing technology can be used or not, maintenance and up-gradation is easy or not for chosen technology, etc.
"SwapCamp" was built using React.js, Node.js, Express.js, and MongoDB. These are the technologies that are very popular and famous in the current industry, and all the people involved in this project have the skills to work with at least one of the mentioned technologies. Thus, it can be concluded that the project is technically feasible.

• Operational feasibility: In operational feasibility, the degree of service provision of the requirements is analyzed, along with how easy it is to operate and maintain the product after deployment. Along with these other operational scopes, determine the usability of the product and determine whether or not the solution proposed by the software development team is acceptable, etc.
SwapCamp is an online platform built with current market technologies, but for any end user, it is a completely straightforward and easy to use website. Even if the user is confused about how to use it, they can simply refer to the how to use section to get a clearer idea of how the web application works. Thus, this project is operationally feasible.

• Economic feasibility: In the economic feasibility study, the cost and benefit of the project are analyzed. Detailed analysis of what will be the project cost for development which includes all costs required for final development such as hardware and software resources required, design and development cost, operational cost, etc. Next, it is analyzed whether the project will be beneficial in terms of funding the organization or not.
During the development of SwapCamp, there were not that many services that needed to be paid for, it can be easily covered from the estimated revenue gained from the shipment cost, ads, monthly/yearly subscription, and when user like to buy credits using their credit card or PayPal. Thus, in conclusion, it can be said that the project is Economically feasible.

5.2.3 Problem Solution Analysis

There are plenty of websites that help people to exchange the things that they have, using their used or unwanted stuff or money, but the problems of these websites are:

• Not safe: these websites only provide a platform where people can post their items, and they have to chat with other people who are interested to swap their products and meet them personally. Many of the kidnappings, scams, and fraud were done through these sites because everyone can open an account and chat with any post owner.

• Hard to check: Since these websites allow users to post their stuff, some of them might post something out of context or in the wrong category, making it hard for an admin to check multiple posts in a few seconds.

• Not guaranteed: not all the users of these websites get a chance to swap their items.

There are thousands of people who always like to buy new clothes and things and this makes them waste a lot of money, and on the other hand, there are people who are not able to buy the things they want because they are expensive.
SwapCamp will solve all the problems mentioned above as:

• All the processes are done online, and there are no transactions between users. The user only needs to submit information about the item that he has. A delivery man will contact him to get the product from him. Depending on the item, the user will get a number of credits that he can use to get any product from the site he wants.

• Only the admin can post products on the site.

• All the swap requests will be collected, and if any user did not like the number of credits, he/she got for his/her item, they can get it back in the first 7 days.

• Depending on the credits that each user has in his account and, the user can get any item he wants, and as much he swaps as much, he can get more credits. Through this process, users can save their money and get any desired item they want.

5.2.4 Effect and Constraints Analysis

This website will help users to save their money and at the same time get what they want in a simple process. It will also help to save water because the production of a pair of cotton jeans requires about 7,000 liters of water – the equivalent of filling over 50 bathtubs. As well as it is going to reduce using pesticides because the production of cotton uses about 11 percent of global pesticides and 25 percent of the global insecticides. Moreover, it will save the environment as there are about 1.3 hectares of farmland is needed to produce a single tone of cotton. For those reasons we are still trying to add more changes and features to our website that will help and encourage people to start the habit of swapping.

5.3 System Design

5.3.1 Rich Picture

A rich picture is a cartoon-like representation that identifies all the stakeholders, their concerns, and some of the structure underlying the work context. A rich picture is a tool for recording and reasoning about these aspects of the work context, in particular, how they should affect the design.

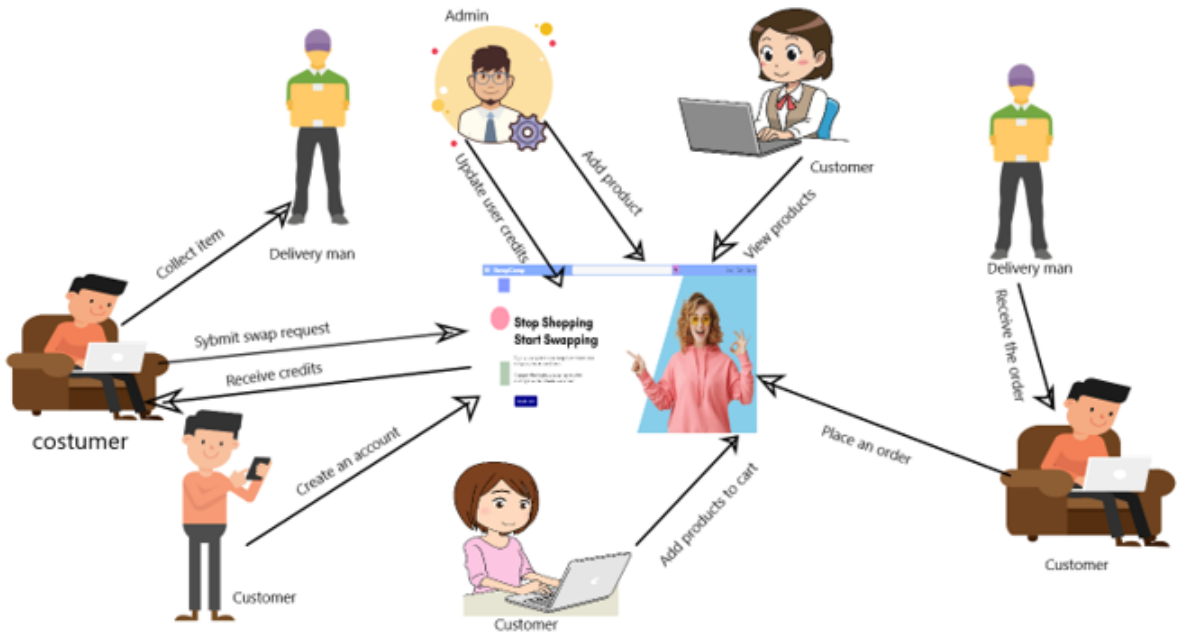

Figure 5.1: Rich Picture of SwapCamp

5.3.2 UML Diagrams

UML provides various types of diagrams to represent the working of the system or software in pictorial format that can be categorized based on two factors, one is a structural diagram and another is behavioral diagram. Structural diagram represents the static aspect of the system which includes UML class diagram, UML object diagram, UML component diagram, and UML deployment diagram. The behavioral diagram represents both static and dynamic aspect of the system which includes UML sequence diagram, UML use case diagram, UML activity diagram, UML Collaboration diagram and UML state chart diagram [7].

We have selected to represent this project with Class diagram from Structural diagrams and use case diagram from the behavioral diagrams.

Class diagram

Class UML diagram can also be considered as the most common diagram type needed for software documentation [6]. As most of the software being created today are still based upon the OOP paradigm, so if we use class diagrams in order to document these software turns out to be a common-sense solution. This also occurs since OOP depends upon classes and the relations.

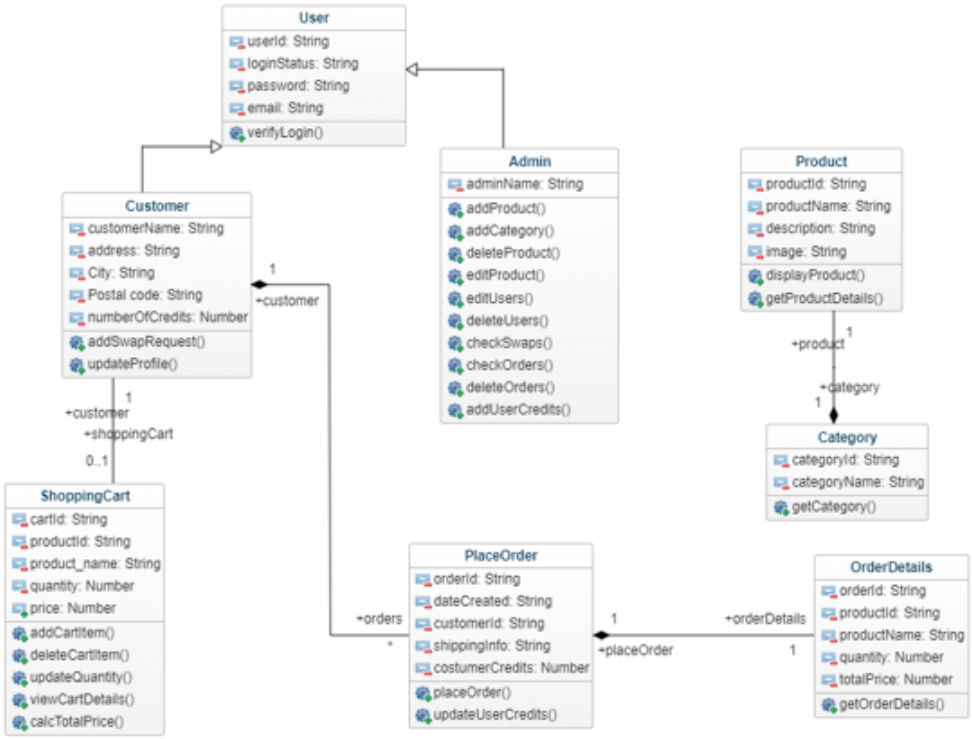

Figure 5.2: Class diagram

Use Case diagram

Use Case diagrams are essentially needed to analyze high-level requirements of the system. Now, these requirements can be expressed with the help of different use cases [7].

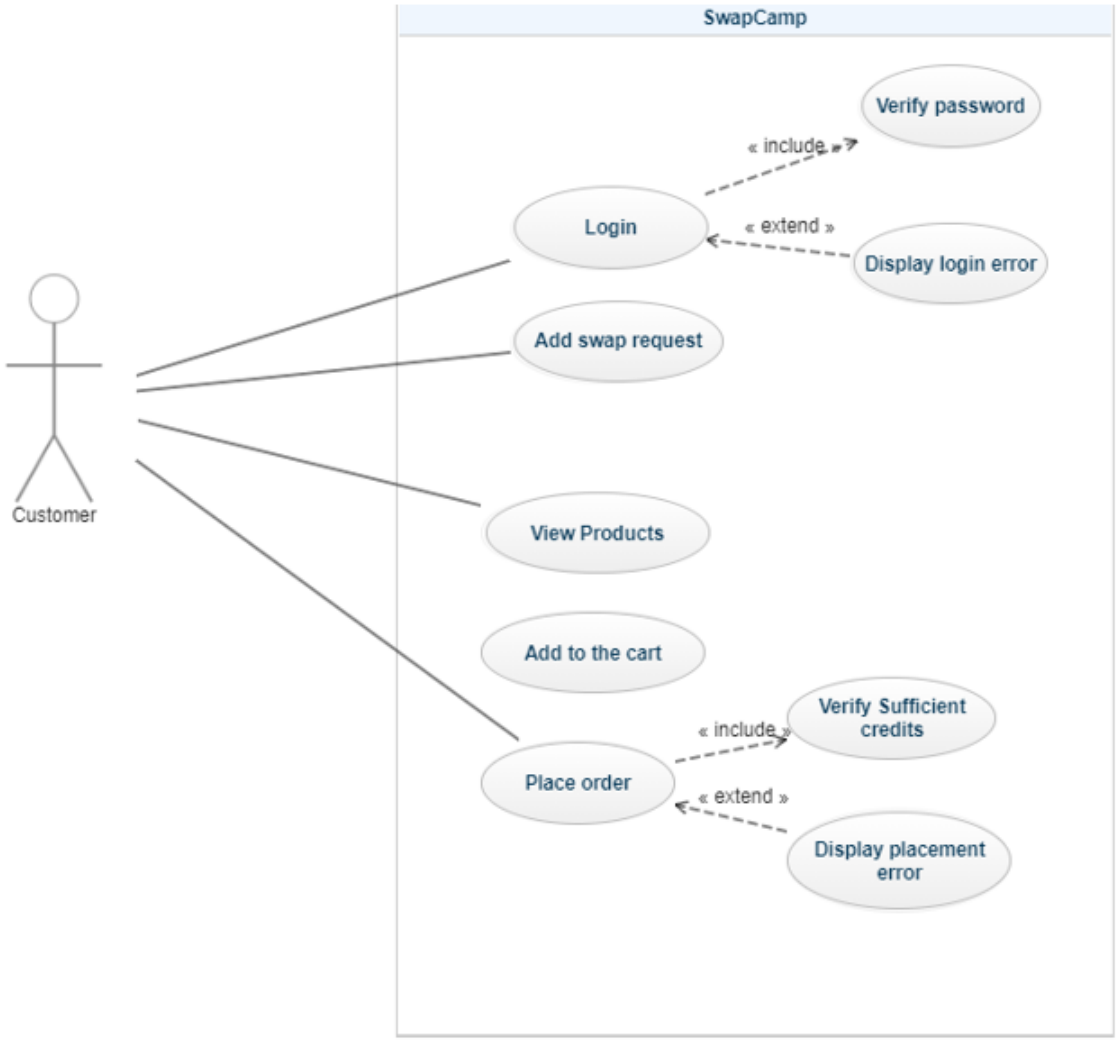

Figure 5.3: Use Case diagram for customer

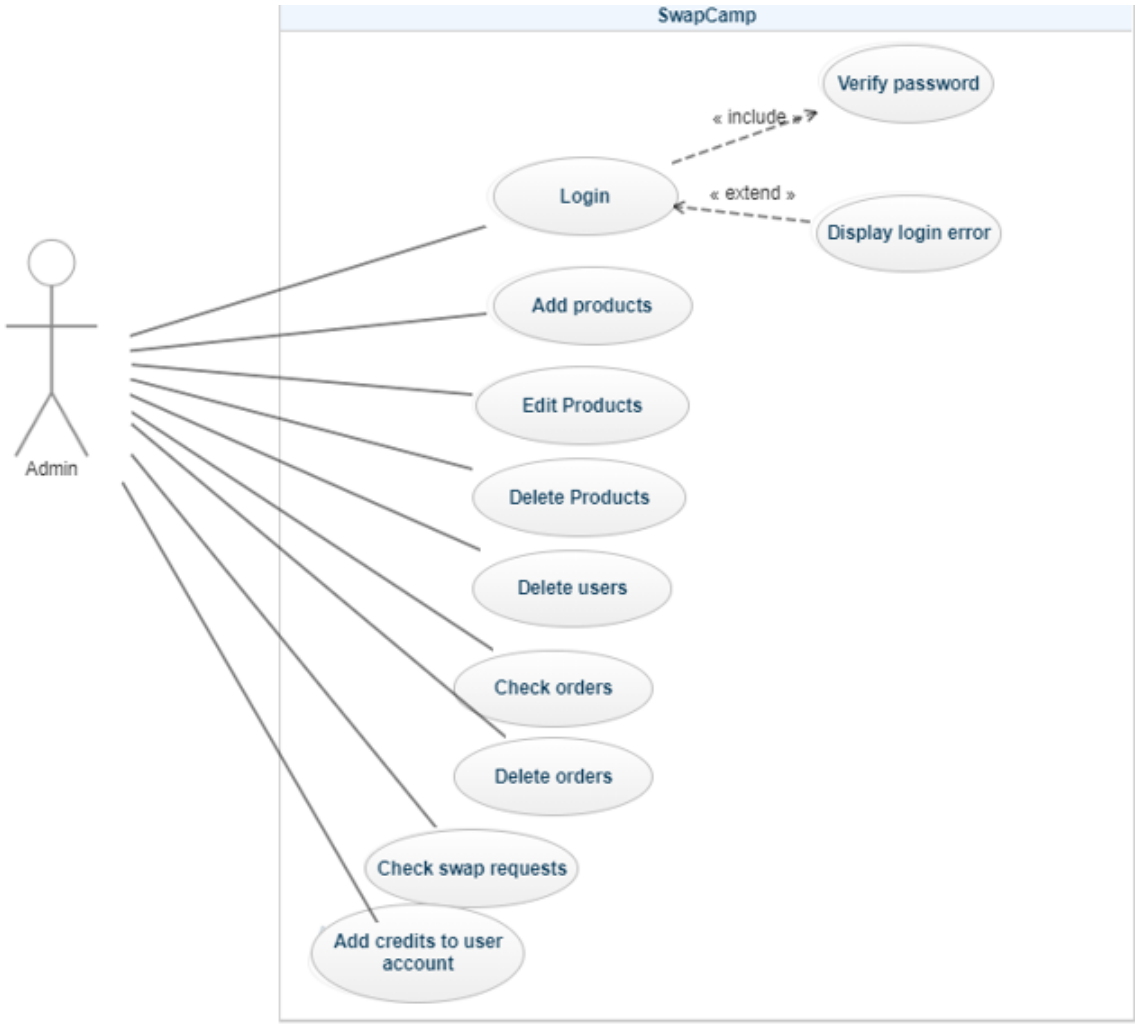

Figure 5.4: Use Case diagram for Admin

5.4 Data Flow Diagram

A data flow diagram shows the way information flows through a process or system.

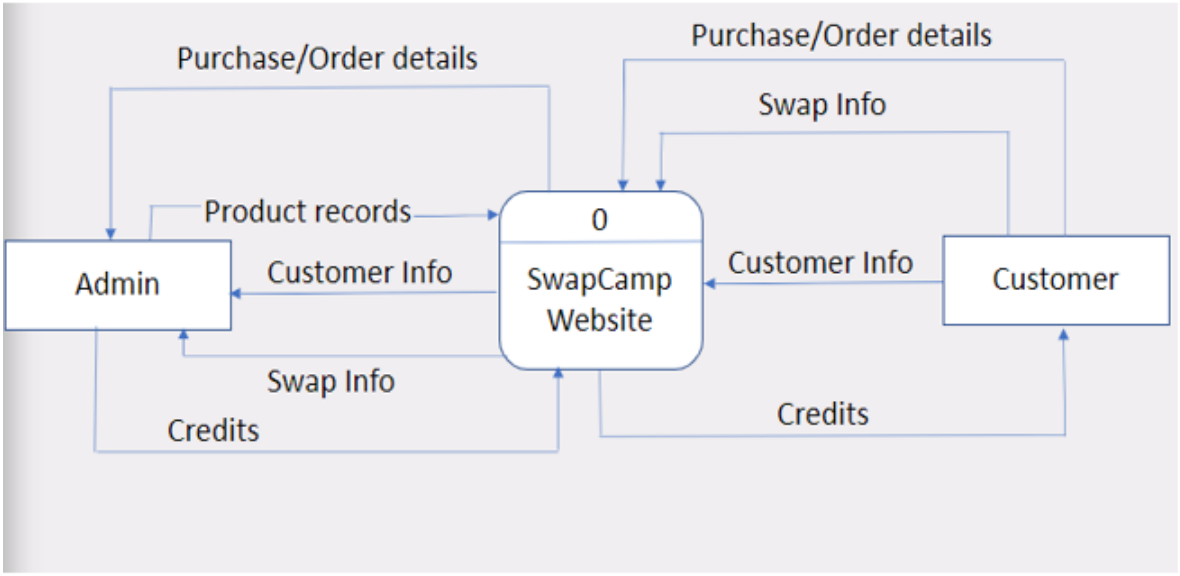

Figure 5.5: Zero level Data Flow Diagram

5.4.1 Functional and Non-Functional Requirements

The software requirements are descriptions of the features and functionalities of the target system. Requirements convey the expectations of users from the software product. These requirements can be obvious or hidden, known or unknown, expected or unexpected from the client's point of view. There are two types of requirements functional and non-functional[8].

Functional Requirements

A functional requirement is a function or feature that must be included in an information system in order to satisfy the business need and be acceptable to the users. It defines what an application and its components are and what these components are supposed to accomplish [9]. The following functional requirements were gathered with our decided requirements gathering methods. The inputs, processes and output are discussed below:

Function: Must be compatible with all types of browsers.		
Input: N/A	Process: Web application must be developed in a common development environment	Output: Web application can be accessible from all sorts of devices
Precondition: User must have a working device with internet Connection.		
Postcondition: Everyone can use this website.		

Table 5.2: Functional Requirement 01- Compatibility

Function: Navigate from One Screen to Another		
Input: Select View to Navigate to	Process: Set up Screen Navigation	Output: Navigate to desired Screen
Precondition: The web application is turned on		
Postcondition: Screen navigated to, will be displayed and further Navigations are possible.		

Table 5.3: Functional Requirement 2- Navigation

Function: Create an account.		
Input: User name, email, password.	Process: Call API to server to create a new user from provided data.	Output: user is created and will be Navigated to Home Screen.
Precondition: User in Registration Screen and device must be connected to the internet.		
Postcondition: User will get a success message and will be navigated to Home Screen.		

Table 5.4: Functional Requirement 3- Create an account.

Function: Create a Swap request.		
Input: Have to provide all item related details.	Process: Go to the "Add swap" page and add your item details.	Output: The swap request has been created.
Precondition: User in Add Swap Screen and device must be connected to the internet.		
Postcondition: User will get a success message.		

Table 5.5: Functional Requirement 4- Create a swap request.

Function: View Products.		
Input: Navigate to any product category.	Process: Call API to server to fetch all products of that category.	Output: View all available products of that category.
Precondition: Web application is on and device is connected to the internet.		
Postcondition: User will be able to view all the available products of that category.		

Table 5.6: Functional Requirement 5: View Products.

Function: View product details.		
Input: Navigate to any product.	Process: Call API to server to fetch all information about that product.	Output: View all information about that specific product.
Precondition: User in products screen and device is connected to the internet.		
Postcondition: User will be able to view all information related to that product.		

Table 5.7: Functional Requirement 6- View Product details.

Function: Add product to the shopping cart.		
Input: Navigate to any product.	Process: Call API to server to fetch all information about that product. Click on add to cart.	Output: The product has been added to the cart and will be Navigated to Cart Screen.
Precondition: User in selected product screen and device is connected to the internet.		
Postcondition: Product added to the cart screen and user will be Navigated to cart screen.		

Table 5.8: Functional Requirement 7- Add to Cart Screen.

Function: Place Order.		
Input: Navigate to Cart Screen.	Process: 1- Call API to server to fetch all information stored in the cart. 2- User needs to click on proceed to checkout. 3- User will be navigated to shipment information Screen. 4- User will have to enter shipment information and click on Continue. 5- If user have enough credits in his account, he will be able to place his order successfully and his credits will decrease. If not, he will be navigated to Add Swap Screen in order to gain credits.	Output: The order has been placed successfully. Or user have to gain more credits and get navigated to Add Swap Screen.
Precondition: User in Cart screen and device must be connected to the internet		
Postcondition: Order has been placed/Failed to place the order.		

Table 5.9: Functional Requirement 8: Place Order.

Function: View orders history.		
Input: Navigate to Order History Screen.	Process: Call API to server to fetch all orders placed.	Output: View all orders placed.
Precondition: User in order history Screen and device must be connected to the internet		
Postcondition: User will be able to view the all his placed orders.		

Table 5.10: Functional Requirement 9: View orders history.

Function: View specific order history.		
Input: Navigate to Order History Screen.	Process: 1-Call API to server to fetch all orders placed. 2- click on any order details. 3- Call API to server to fetch details of that order.	Output: View specific order details.
Precondition: User in order history Screen and device must be connected to the internet		
Postcondition: User will be able to view order details.		

Table 5.11: Functional Requirement 10- View specific order history.

Function: Edit user Profile.		
Input: Navigate to User Profile.	Process: 1-Call API to server to fetch user Info. 2- User can apply changes on his name or email or password and then click on Update.	Output: The user will get a success message and his profile will be updated.
Precondition: User in User Profile Screen and device must be connected to the internet		
Postcondition: User profile will be updated.		

Table 5.12: Functional Requirement 11: Function: Edit user Profile.

Non-functional Requirements

A nonfunctional requirement is a description of the features, characteristics, and attributes of the system as well as any constraints that may limit the boundaries of the proposed solution. Non-functional requirements are briefly described below:

1. Performance: Represents the performance of the system required to display and satisfy users' needs. This web application should provide a smooth user experience, and there should be no input lag such as submitting swap requests, getting and using credits successfully. These should give a positive experience for users.

2. Information: represents information that is relevant to users in terms of content, timeliness, accuracy, and format. It has to do with the necessary inputs and outputs and how

to manage them, what types of data will be required to be stored, how the information will currently be saved in the system, how the interfaces of external systems will work, etc.

3. Backup: The system should back up and save the user's data.

4. Availability: The system should be available online without having any trouble.

5. Response Time: The system should respond very fast.

6. Excellent UI Experience: All the interfaces of this app are easy to understand, clean and user friendly.

7. Maintenance: This system/application will be maintained twice a year, to make it run smoothly and not slow down or lag.

8. Security: Security is always a concern. The system should have higher security because it will contain users' information. In this project, Node.js express.js and JWT are used for backend technology, which has different layers of security. The passwords are also encrypted and then stored in the database.

9. Efficiency: represents the ability of the system to produce output with the least amount of waste. We have tried to get rid of redundant steps in processes and use resources in an efficient manner. Keeping our code non-redundant with reusable code and components is how we have achieved efficiency.

5.5 Product Features

Features of this project are:

For customer:

- Sign In/ Registration.
- Add swap request.
- Check categories and products.
- Check single product info.
- Search for a product.
- Filter by category/price.
- Sort by newest/ low to high/ high to low.
- Add to Cart.

- Place Order.
- Check Orders.
- Update Profile.

For Admin:

- Sign In.
- Check swap request.
- Add categories and products.
- Delete/ Edit products.
- Update user info including credits.
- Delete users.
- Check Orders Info.
- Delete Orders.
- Make order as delivered.
- Update his profile.

5.5.1 Input

Some input screen shoots:

Registration Page

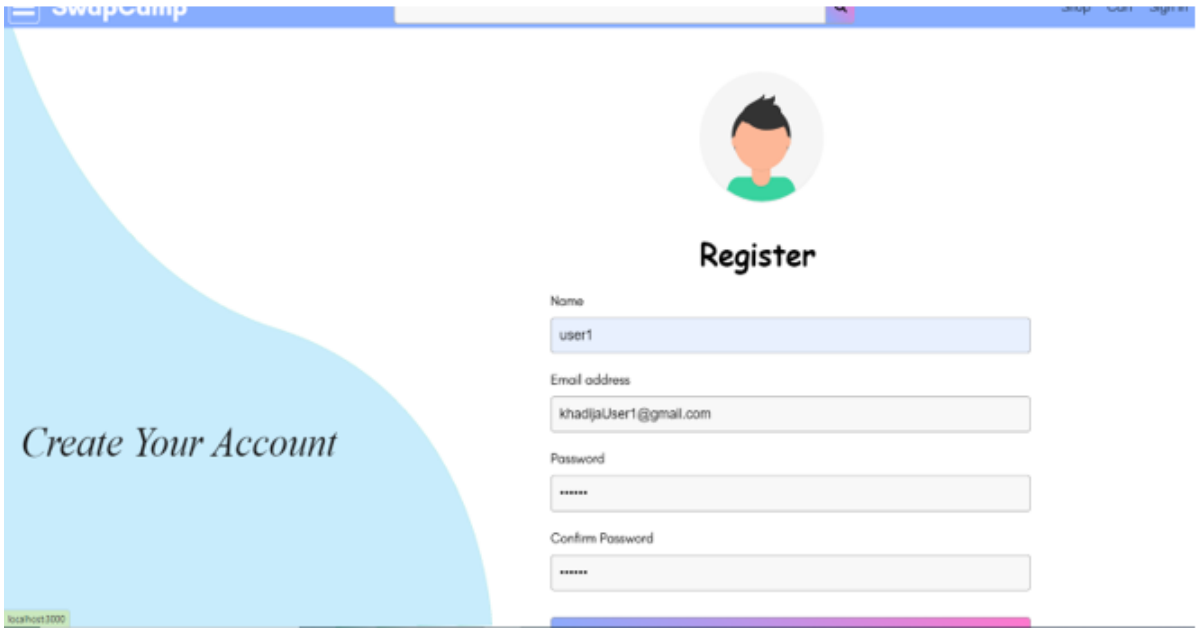

Figure 5.6: Registration Page

Login Page

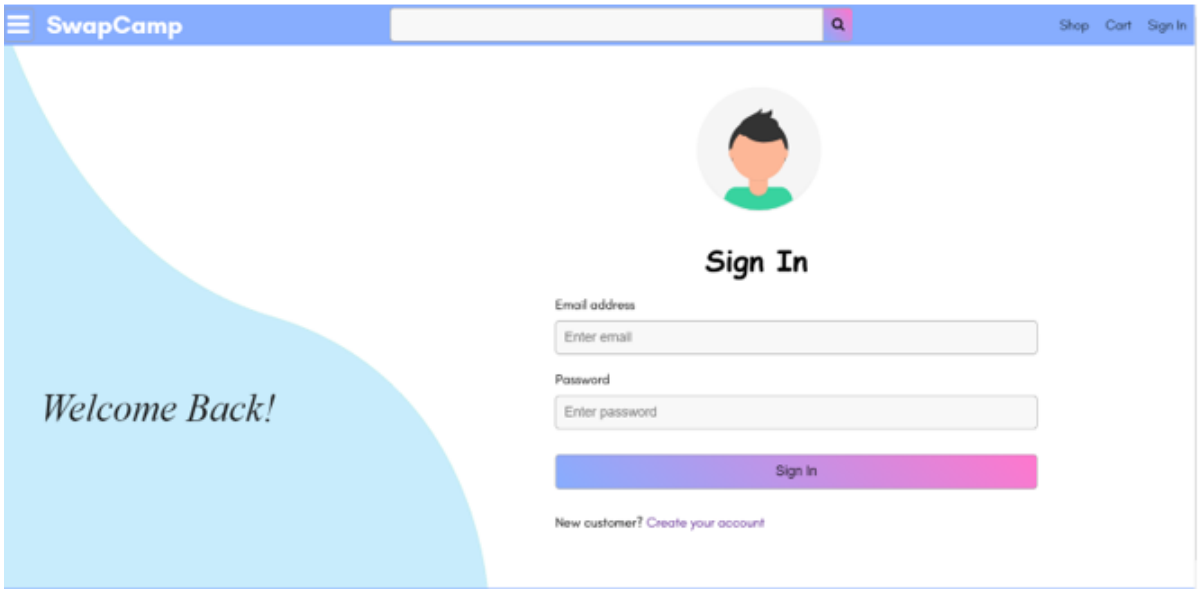

Figure 5.7: Login Page

Shipment Address

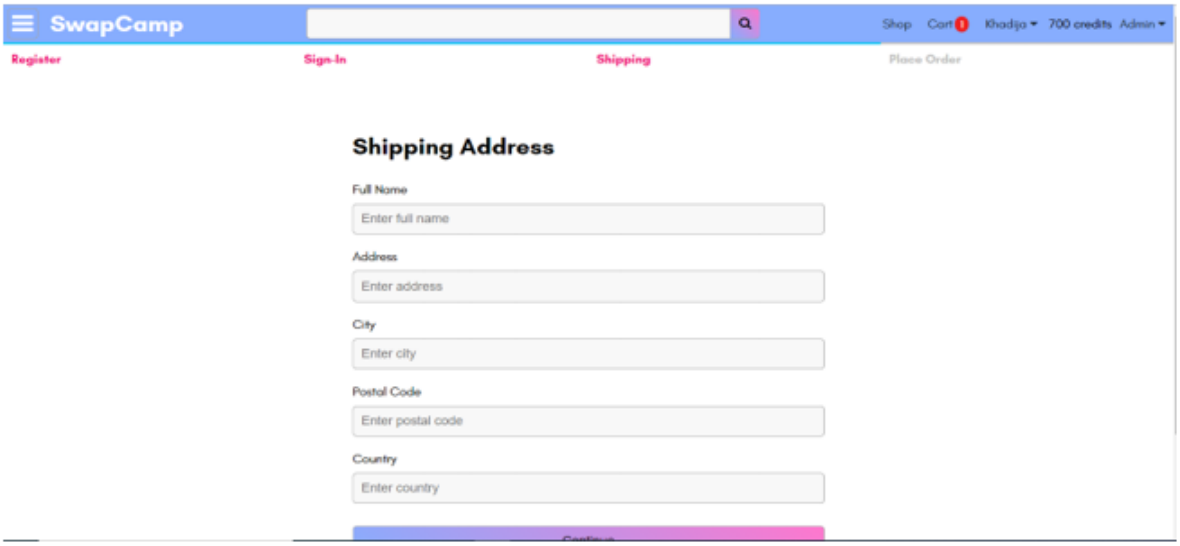

Figure 5.8: Shipment Address input page

Add a swap request input page

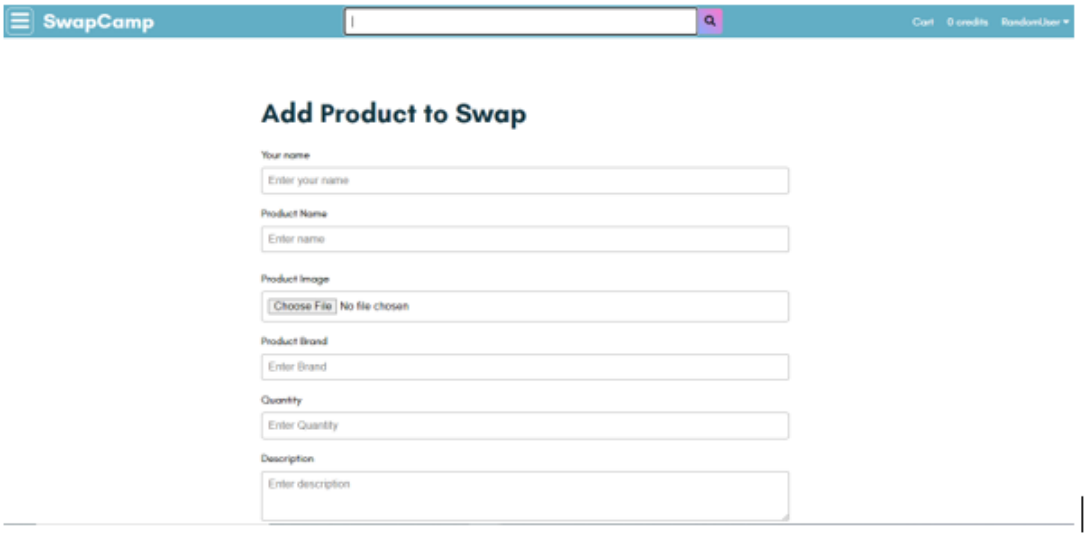

Figure 5.9: Add a swap request input page

5.5.2 Output

Registration/Sign in output:

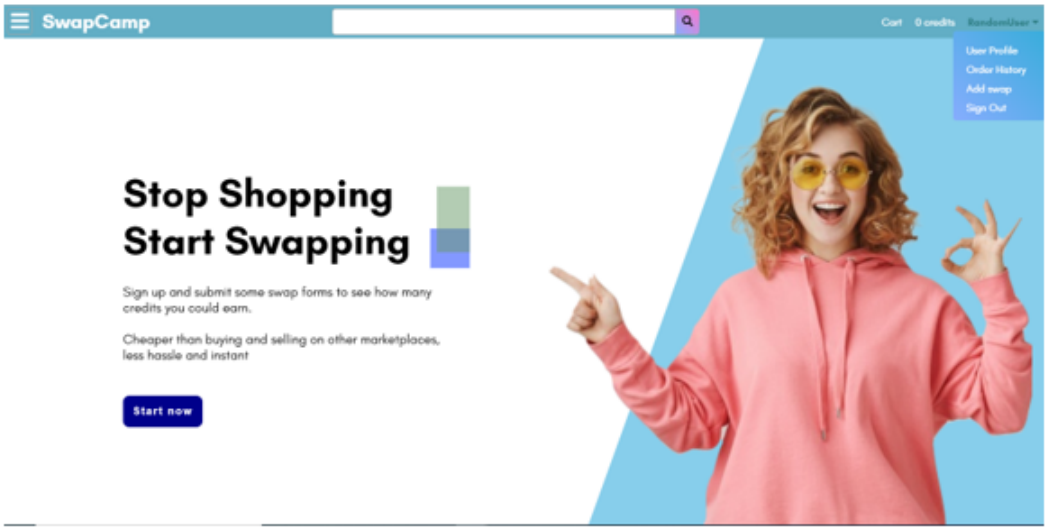

Figure 5.10: Registration/Sign in output

Order Successfully placed:

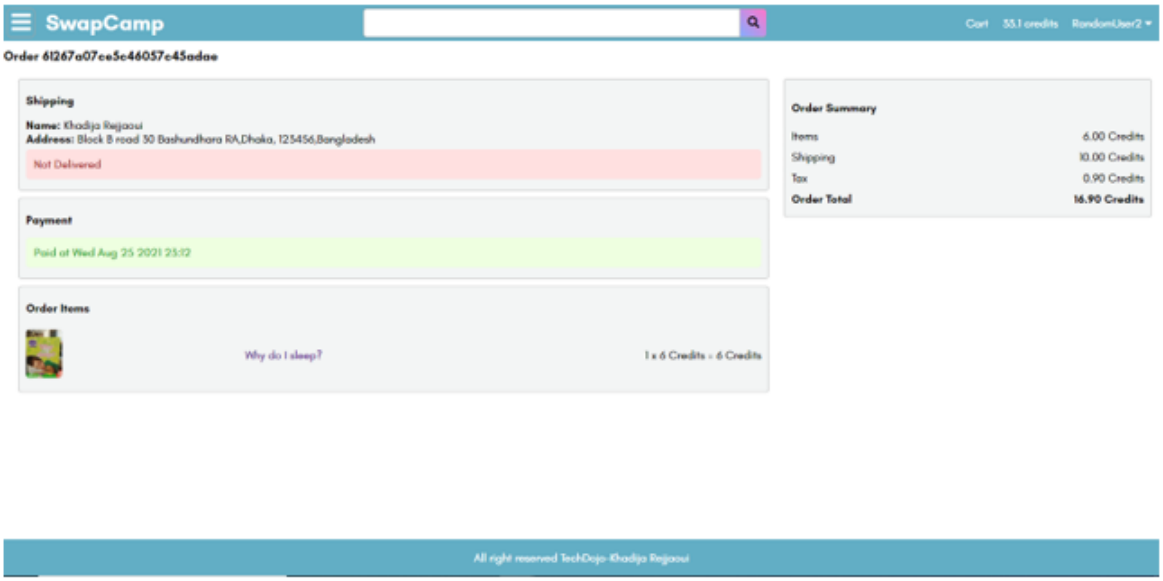

Figure 5.11: Order Successfully placed screen.

Order Failed to placed:

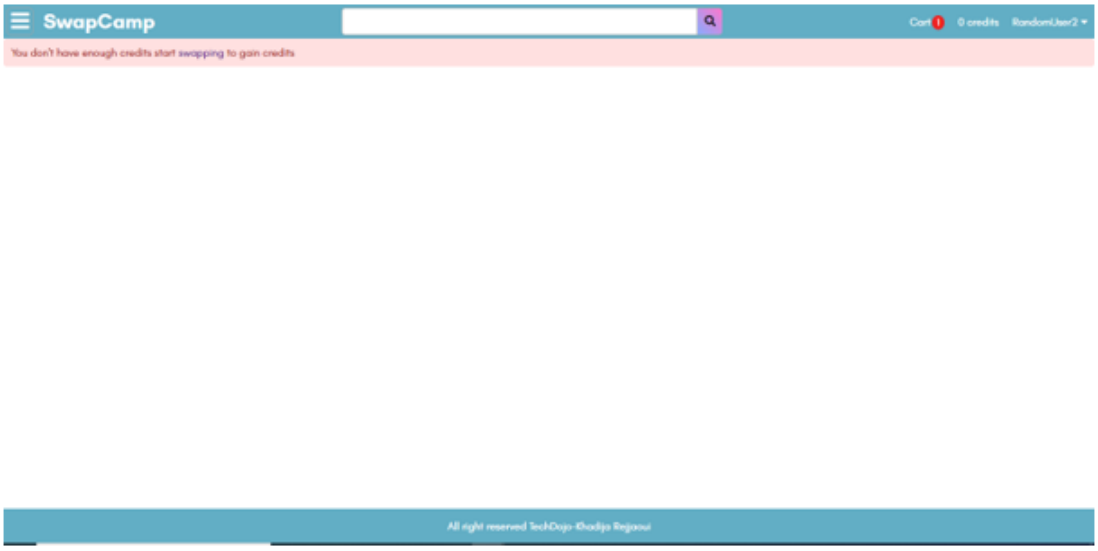

Figure 5.12: Order Failed to placed screen.

Add swap Output:

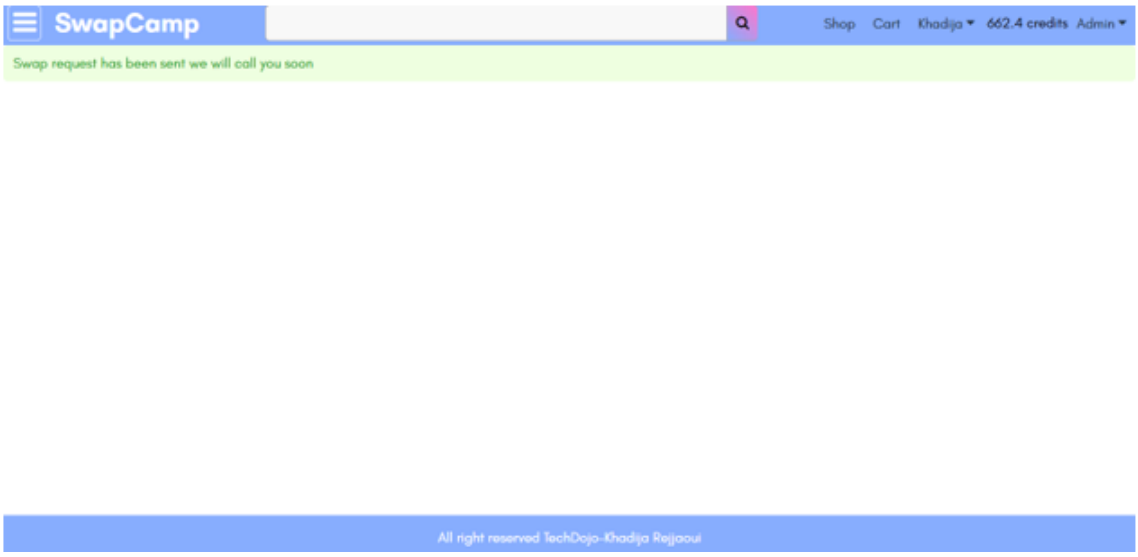

Figure 5.13: Add swap output screen.

5.5.3 Architecture

Software architecture refers to the underlying structures of a software system and the discipline to create such structures and systems. Each architecture includes software elements, their relationships, and the properties of each of the elements and relationships. It also describes the organization and interaction of program components. There are

various kinds of architectures that are used among them. In SwapCamp, we are using a client-server architecture.

The client-server architecture divides the application into two parts, 'client' and 'server'. Such an application is implemented on a computer network, which connects the client to the server. The server part of this architecture provides the central function: i.e., any number of clients can connect to the server and request the execution of a task. The server accepts these requests, performs the requested task, and returns any results to the client, as appropriate.

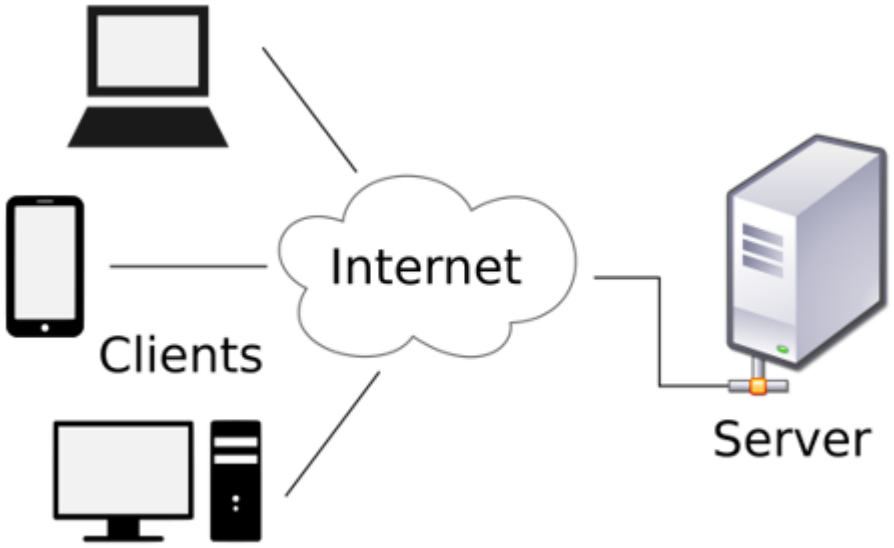

Figure 5.14: The client-server architecture

Chapter 6

Results & Analysis

As mentioned, in the development of this project I have used React which is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. It is fast, simple, and scalable. In addition to that, I have used redux, which allows me to manage this app's state in a single place and keep changes in this app more predictable and traceable. It makes it easier to reason about changes occurring in this app. In the backend, I have used MongoDB, express, and Nodejs. The main reason I have used Node.js and express was they are lightweight, support routing, sessions, and caching, and most importantly integrate the MongoDB database.

Screenshots of this project are given below with short descriptions:

• Home page: The user will get an idea about the website, how he can use it, why he should use it, and he gets to know the vision of the company.

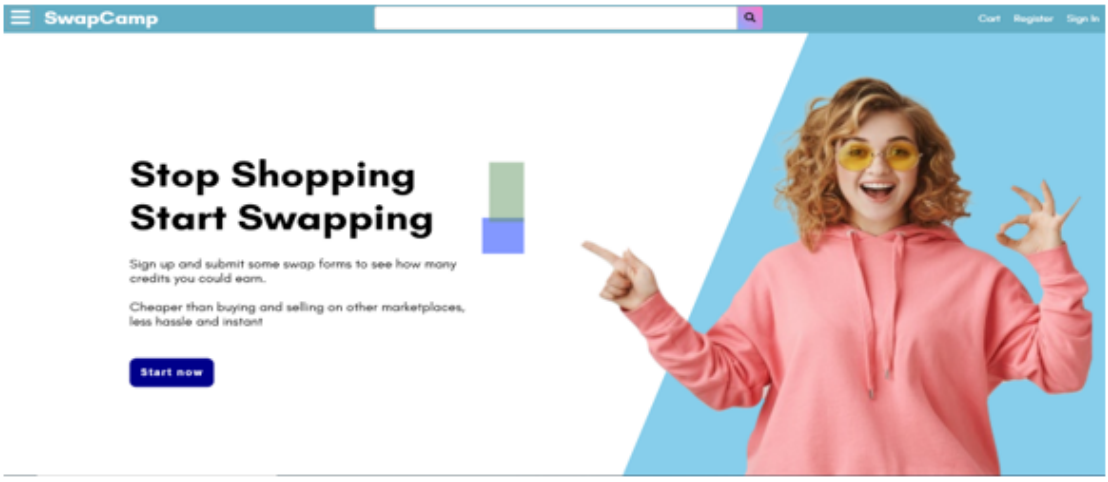

Figure 6.1: The first part of the home page.

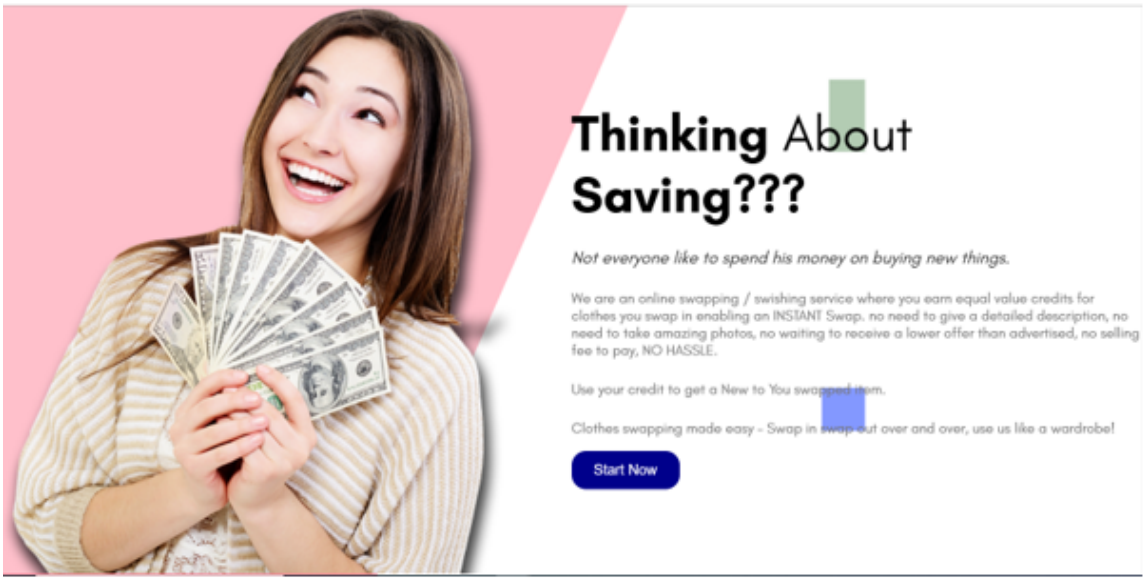

Figure 6.2: The second part of the home page.

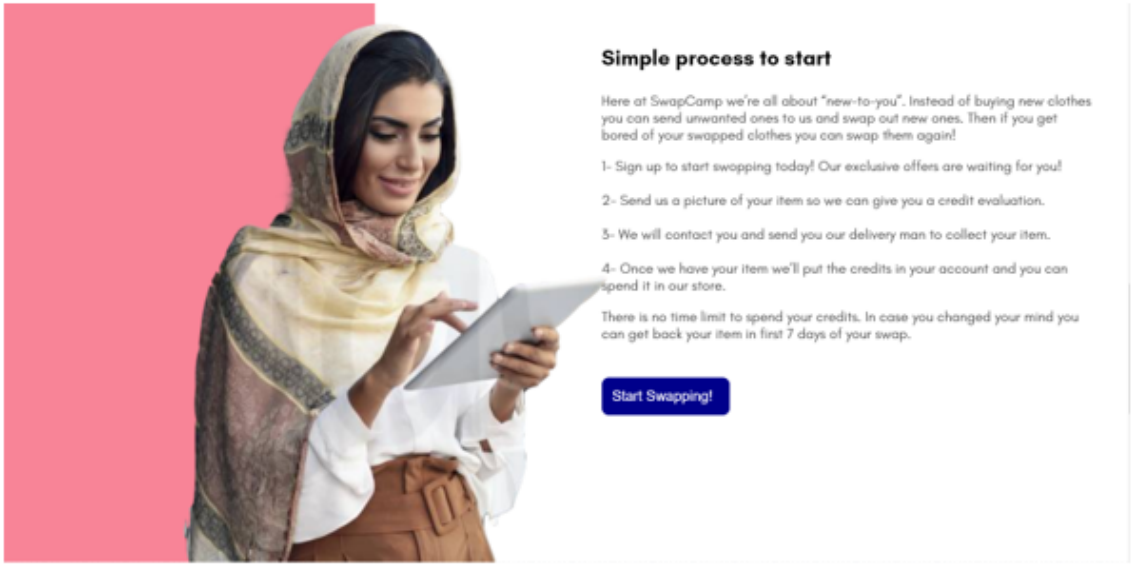

Figure 6.3: The third part of the home page.

Figure 6.4: The forth part of the home page.

• Sign in or registration: The user can either create a new account by clicking on Register, or he can sign in if he already has an account.

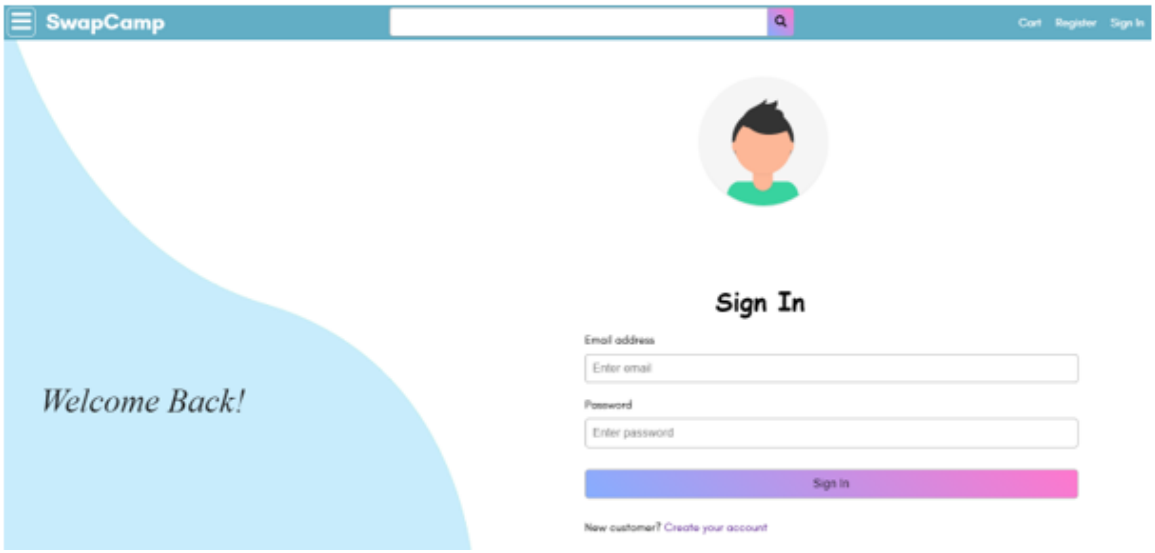

Figure 6.5: Sign in page.

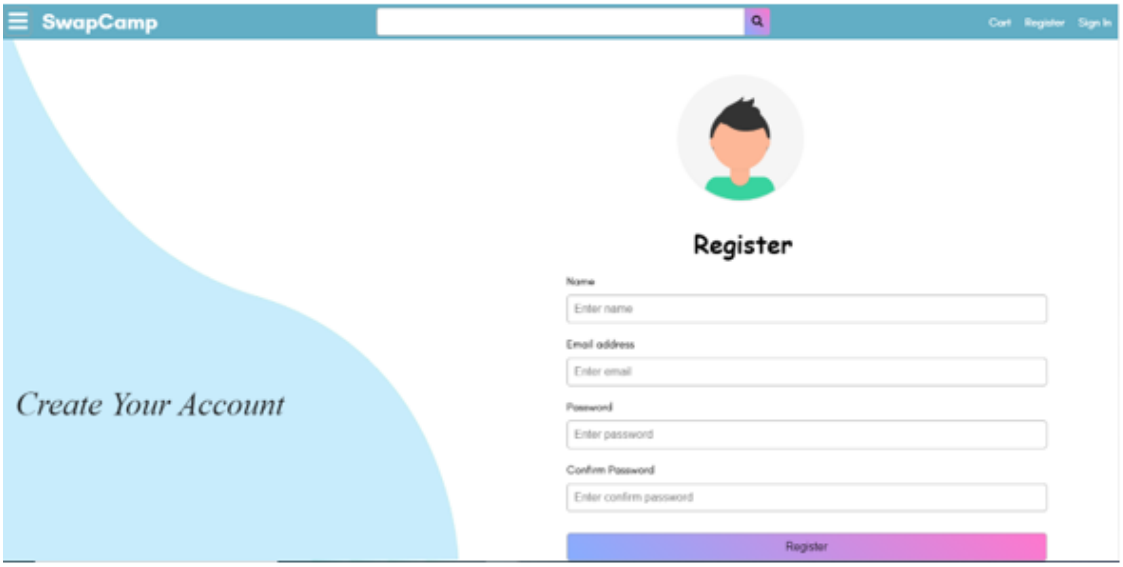

Figure 6.6: Registration page.

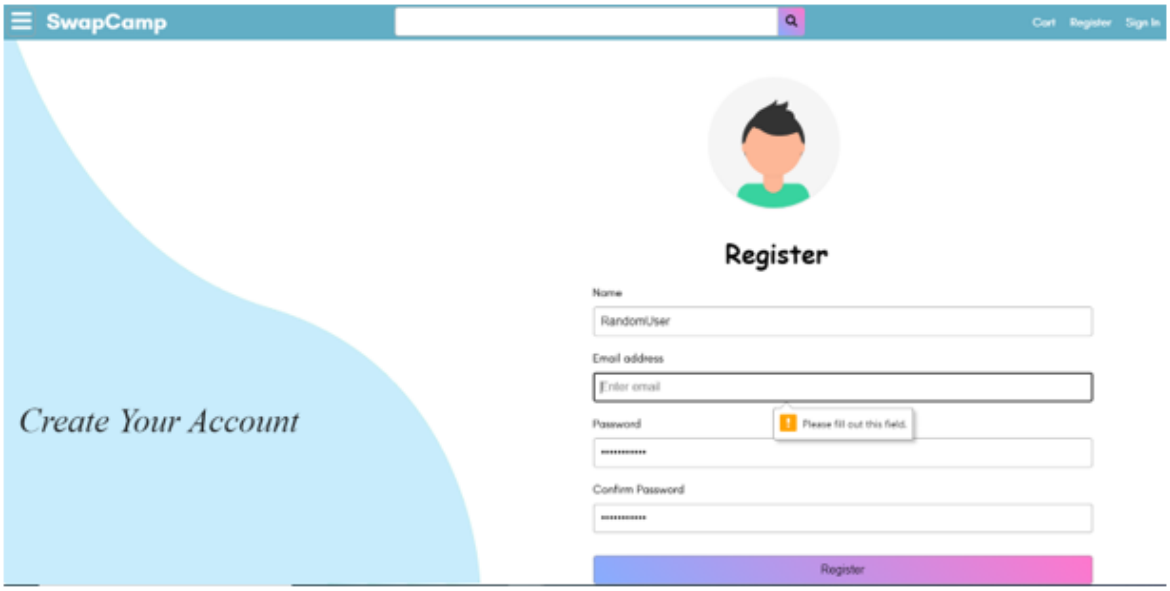

Figure 6.7: Registration page with constraints.

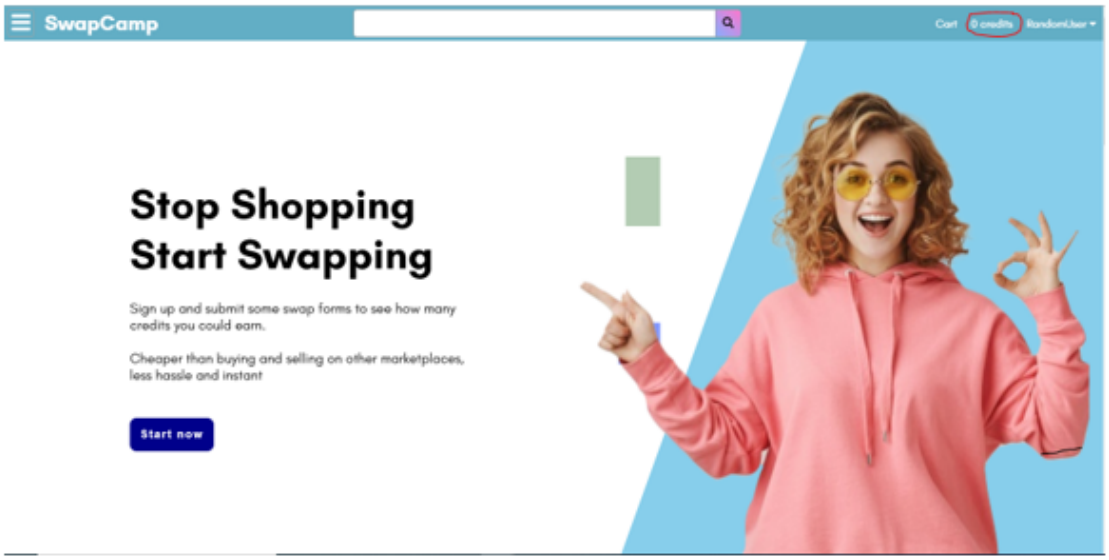

Figure 6.8: Registration success with free 10 credits

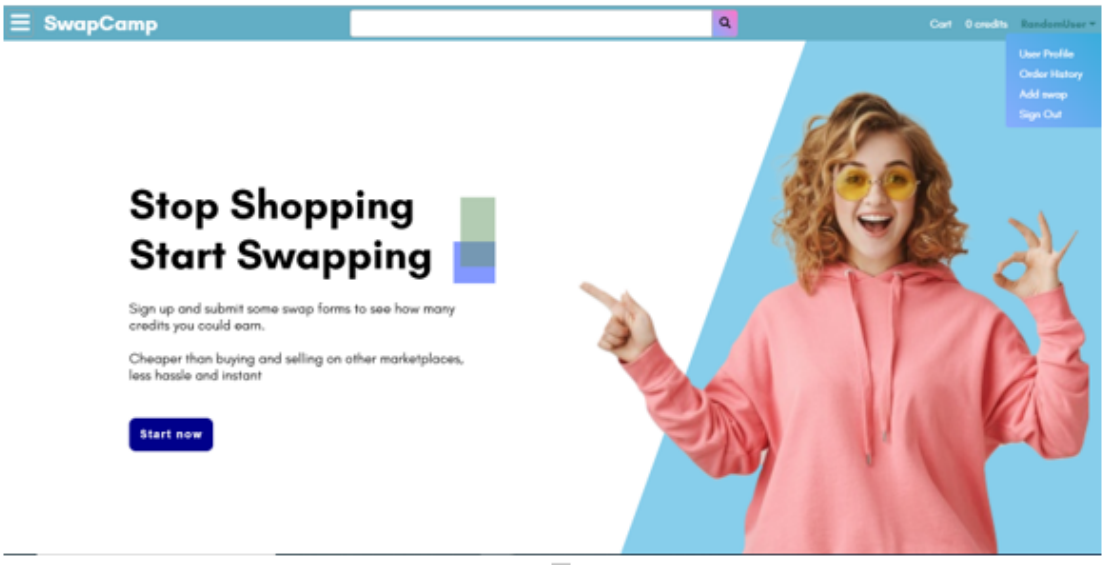

Figure 6.9: User home page.

In fig 6.8 and 6.9 we can see that the user has created his account successfully and he got 10 credits free. Also, he has options to edit his profile, to add swap requests, and to check his orders history.

• Edit profile: The user can edit his username, his email, and password easily.

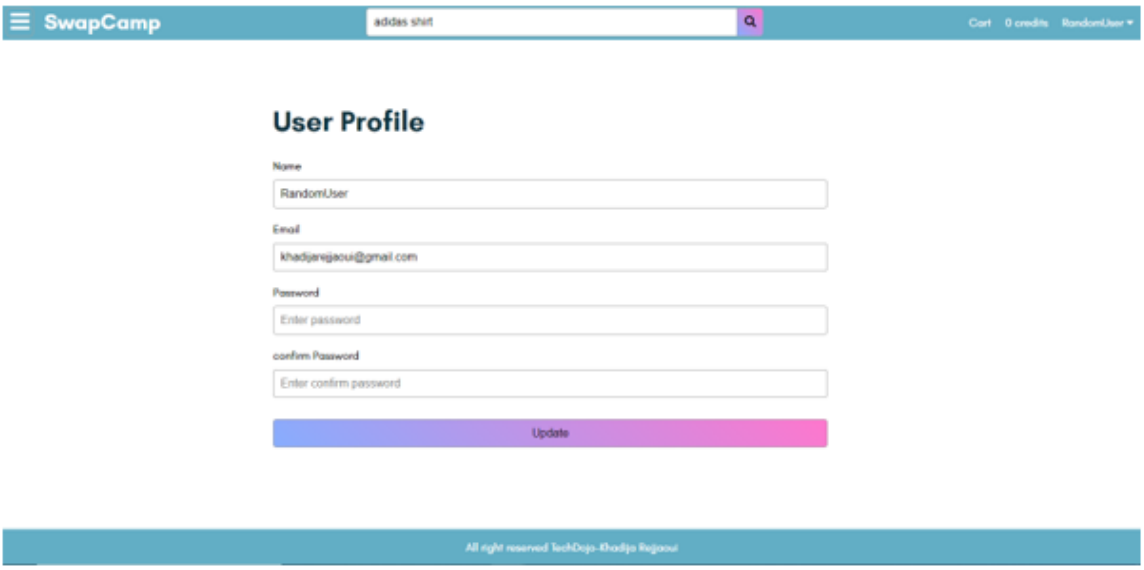

Figure 6.10: User Profile page.

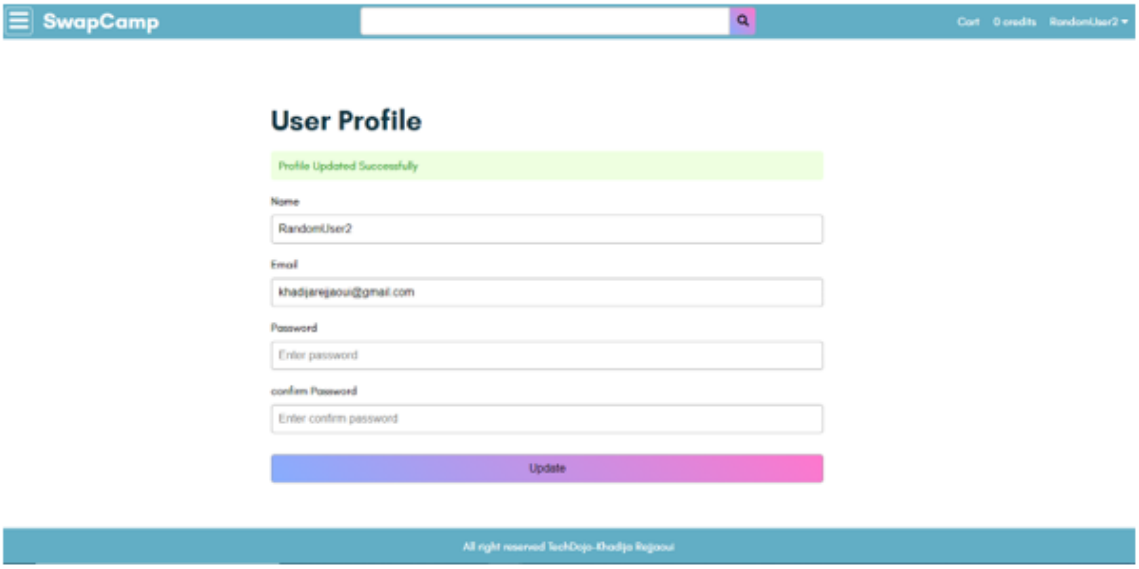

Figure 6.11: Profile updated successfully.

• Add swap: The user adds information about the item that he has. When the admin will check his request, he will inform the delivery man to collect that item. After the item will be received, the admin will assign swapping credits to the user, to use them in the swap camp store. The user will not be able to get any item from the swap store if he does not have enough credits.

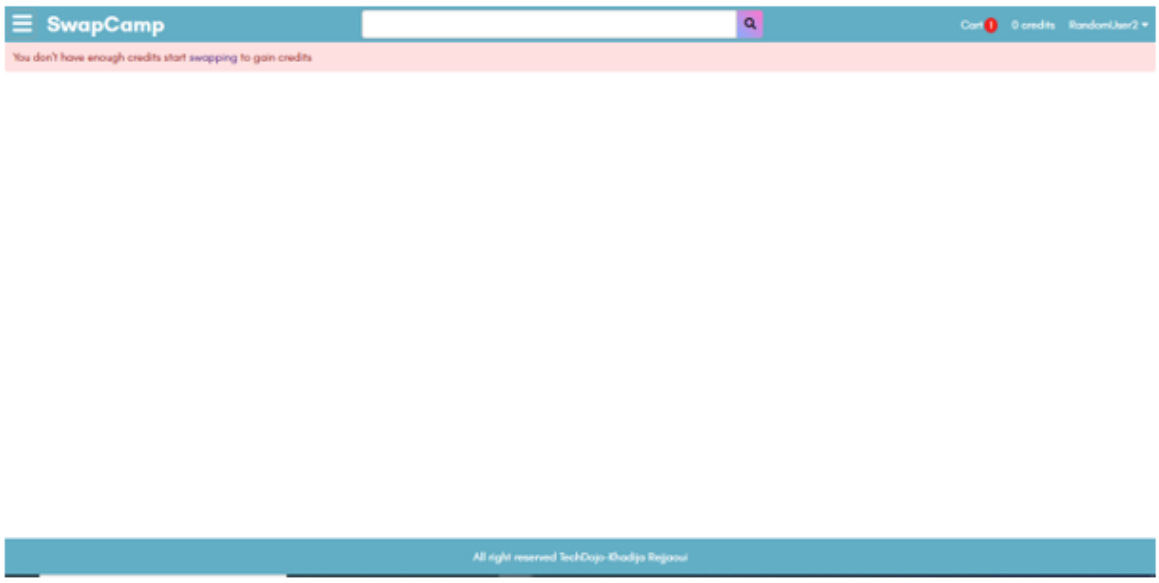

Figure 6.12: Fail to place an order

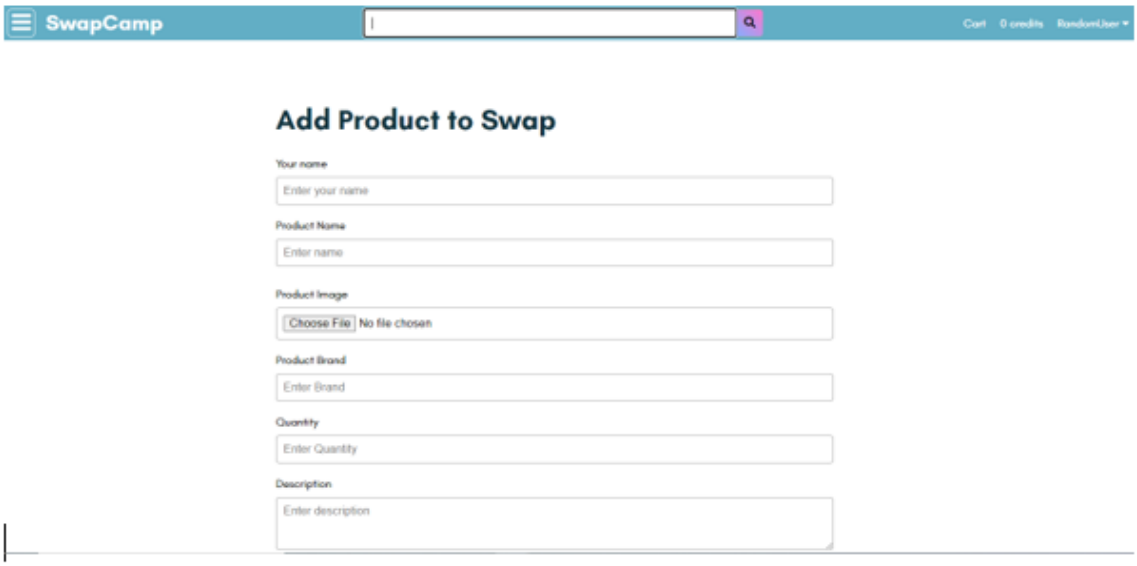

Figure 6.13: add product to swap screen

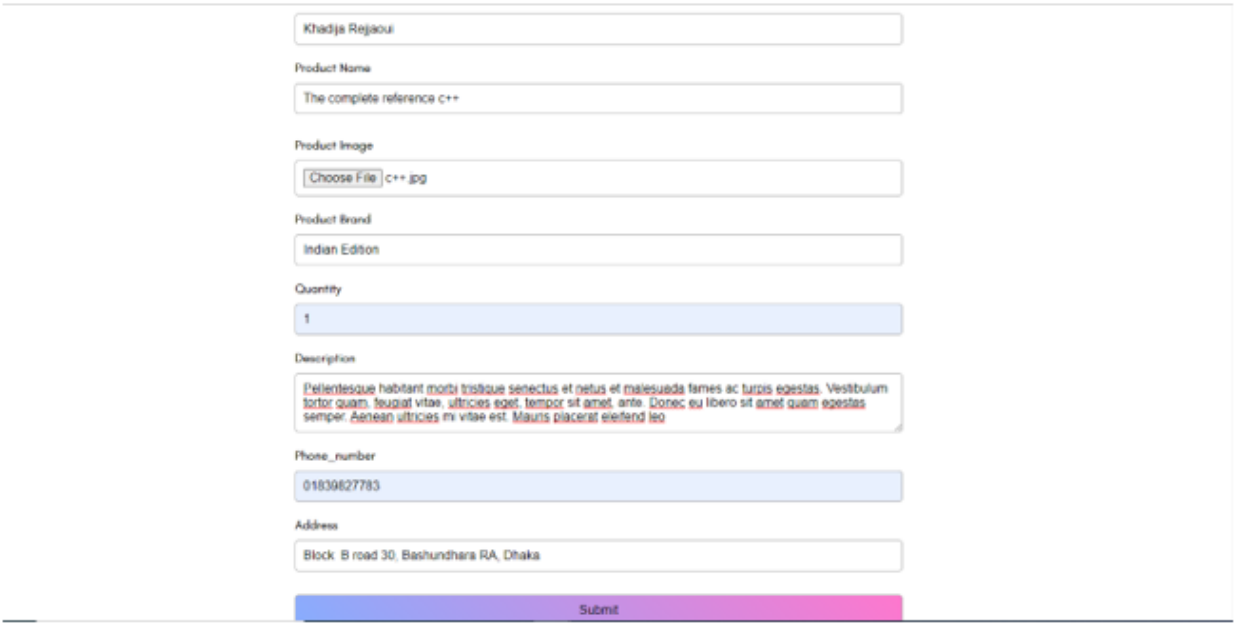

Figure 6.14: swap Form filled with information

• Check Swap: The admin checks the swaps and get information about the users and their item; he can also delete that request.

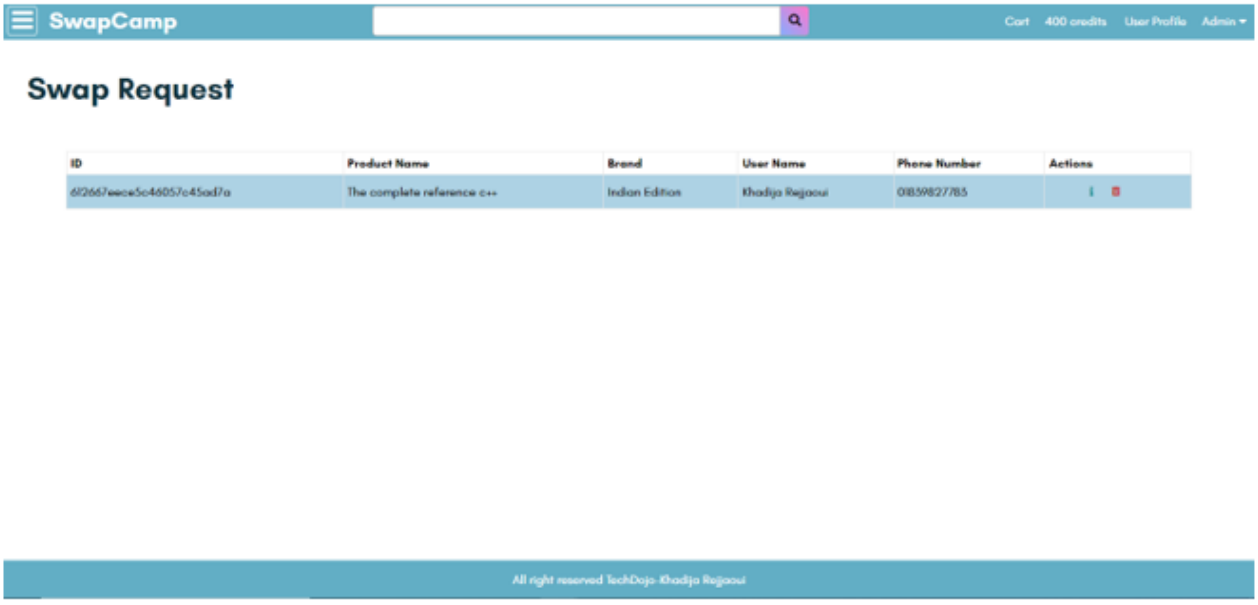

Figure 6.15: Admin's table of checking swaps.

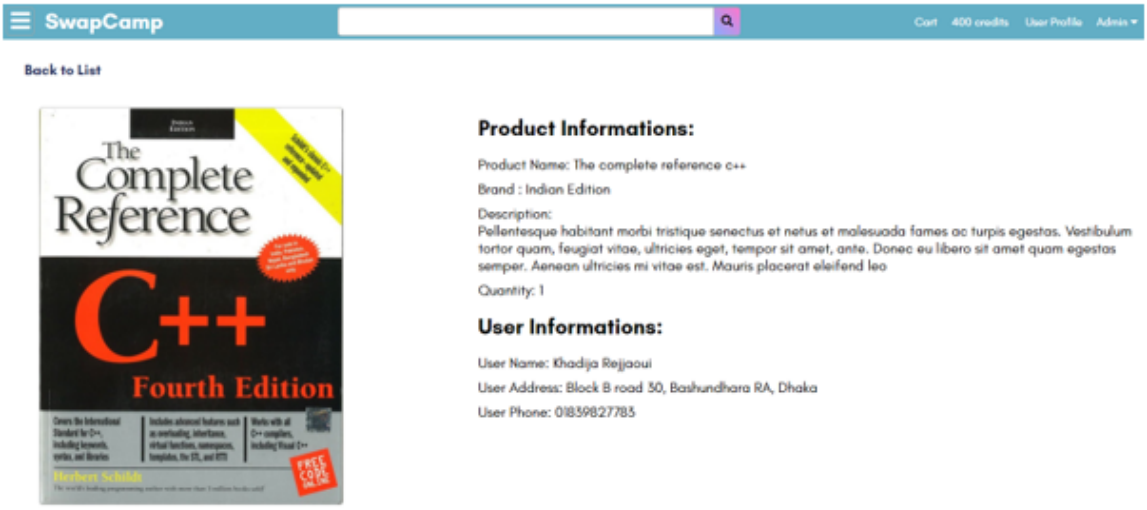

Figure 6.16: Swap item info page

• Assign credits: After receiving the item from the user, the admin can assign swap credits to that user, depending on the quality and value of that item.

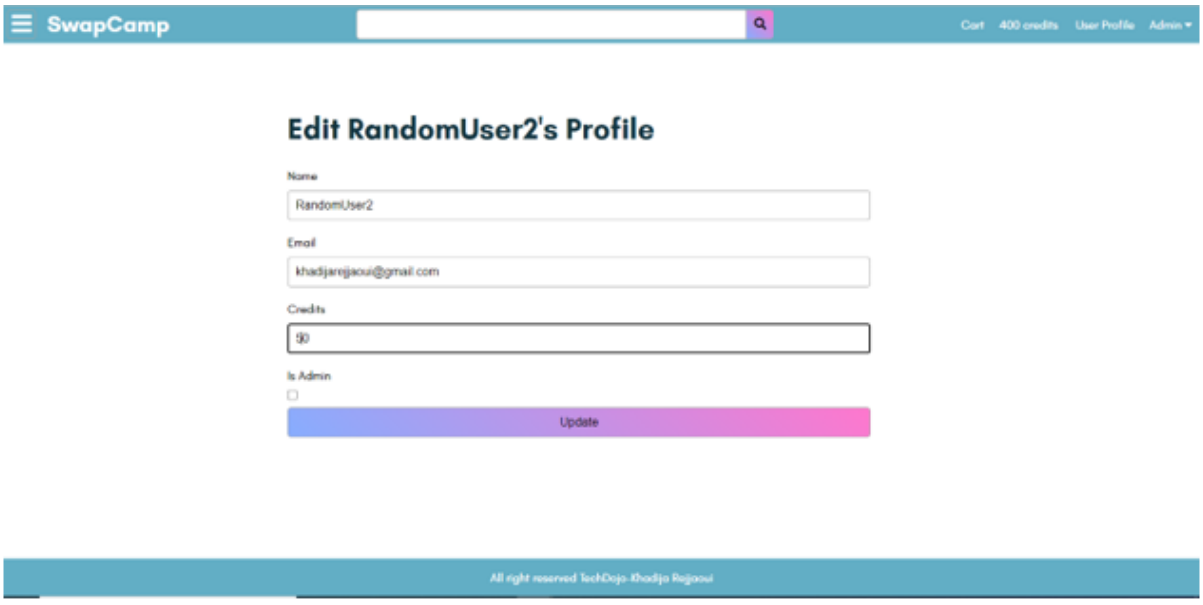

Figure 6.17: Admin assign new credit value to the user.

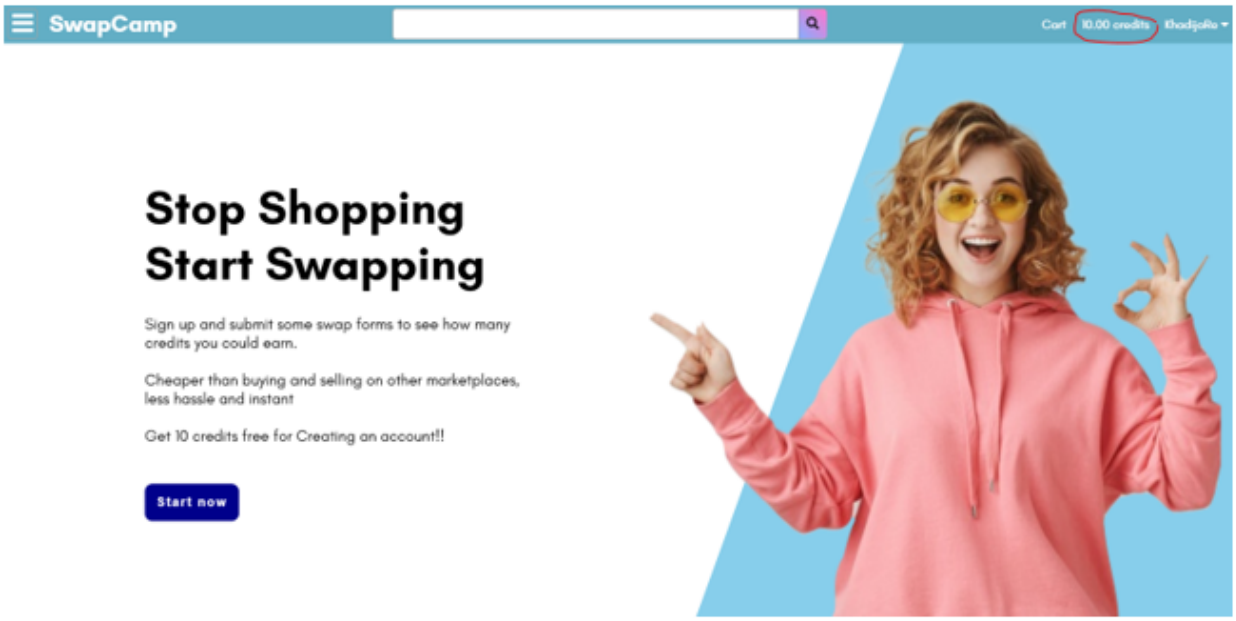

Figure 6.18: User received his swap credits.

• Complete swap process: When the user receives the swap credits, in his account he can check the available categories in the website, to get any product he wants depending on the credits that he has. The user can also sort the product by new arrivals, from low to high, and from high to low credits. Moreover, he can filter by category or credits. After selecting the product that he wants, he can add it to cart and place his order.

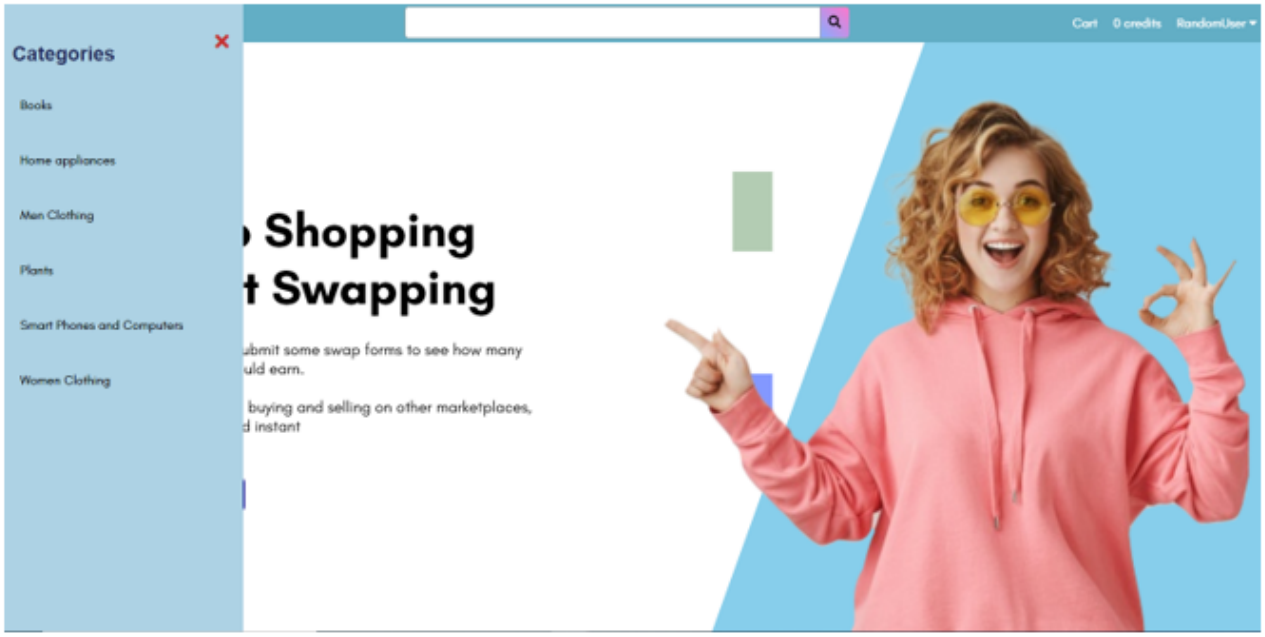

Figure 6.19: Available categories.

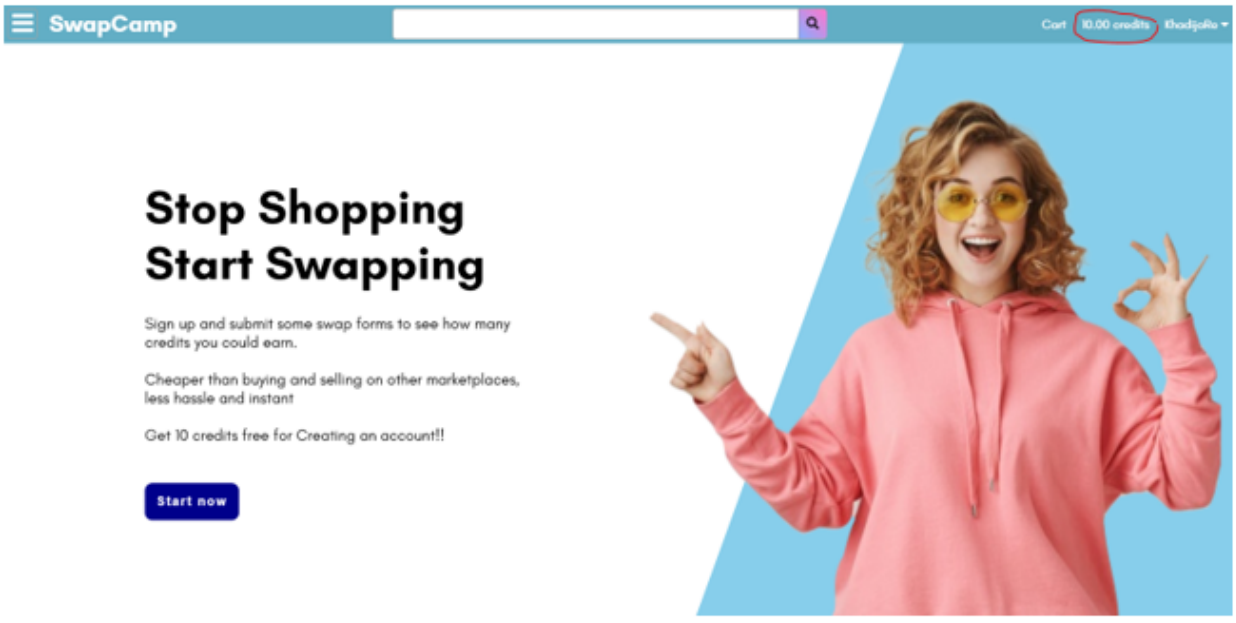

Figure 6.20: Book Category.

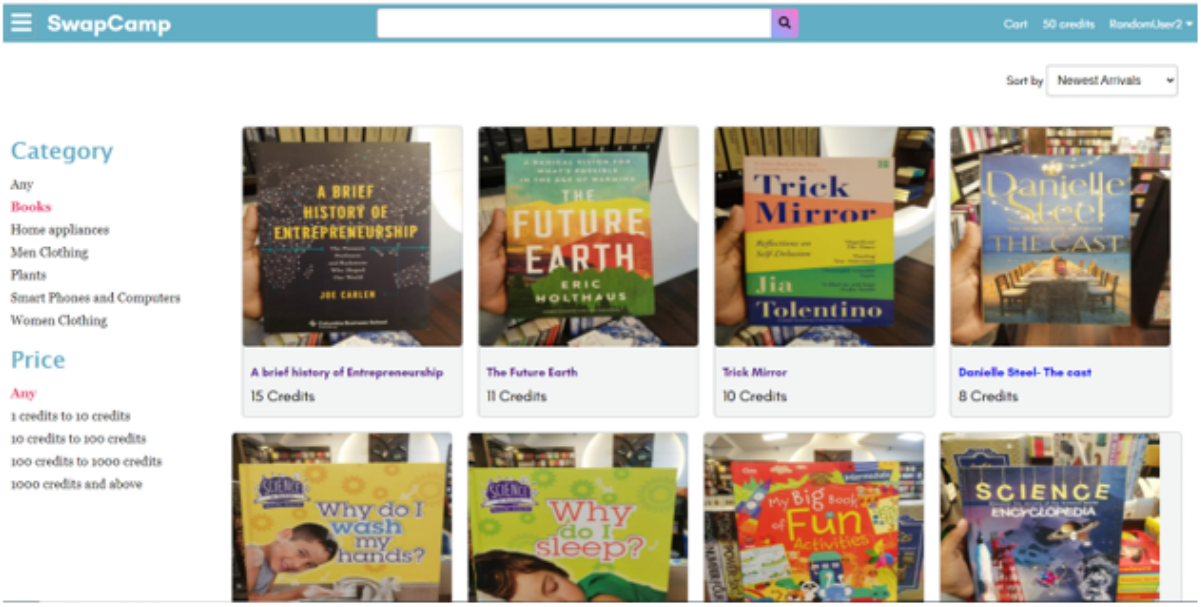

Figure 6.21: User received his swap credits.

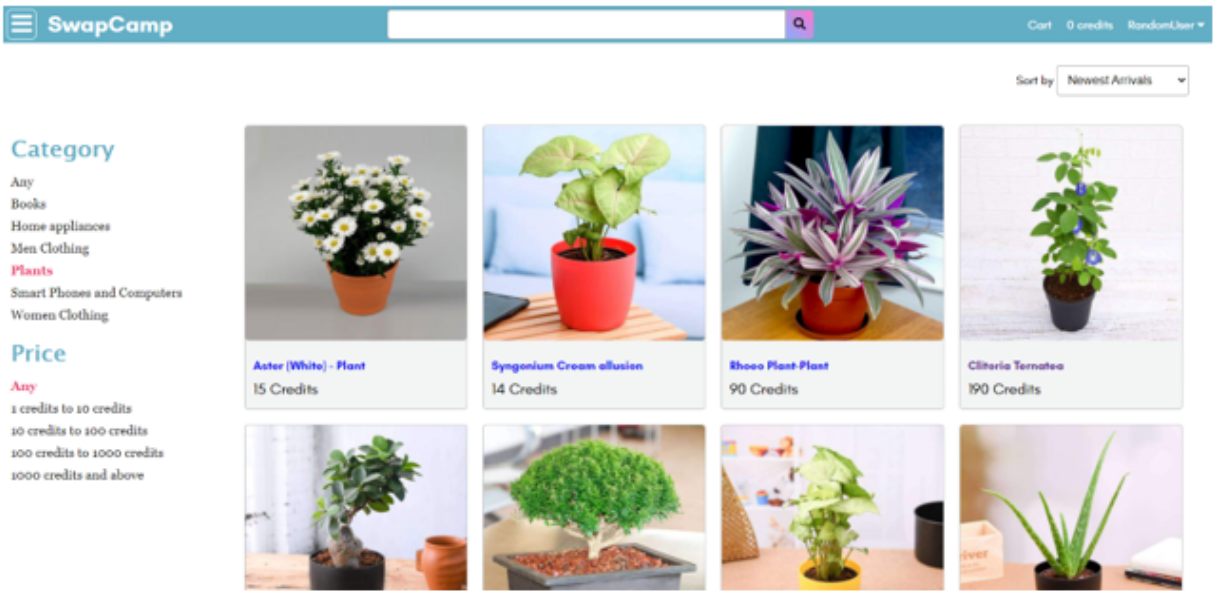

Figure 6.22: Plant category sorted by new arrivals.

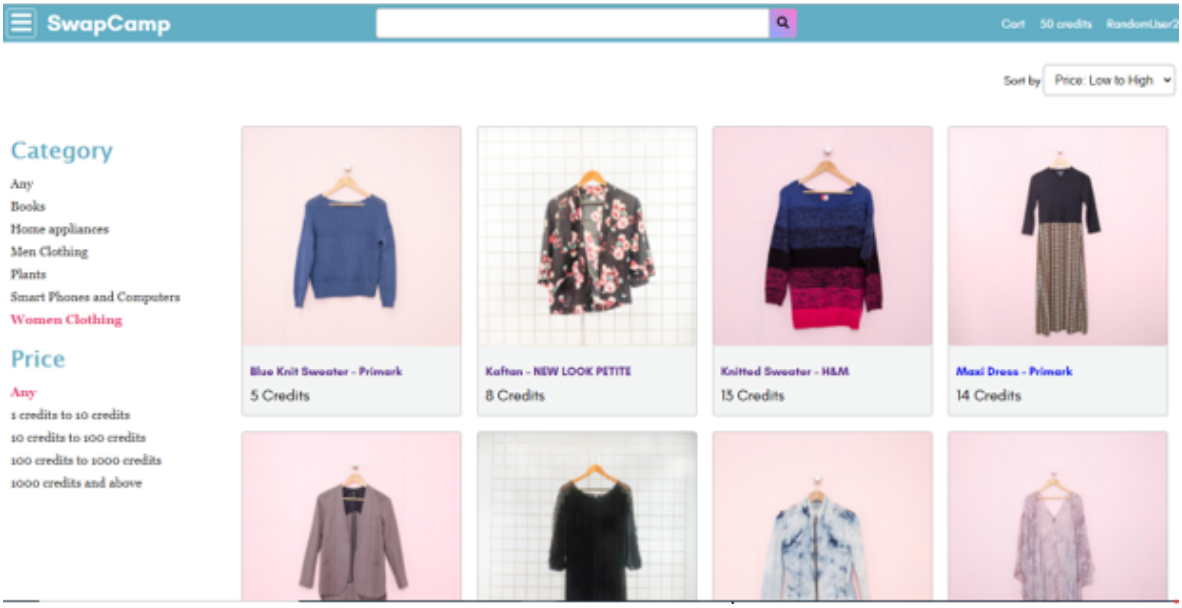

Figure 6.23: Women clothing category sorted by low to high.

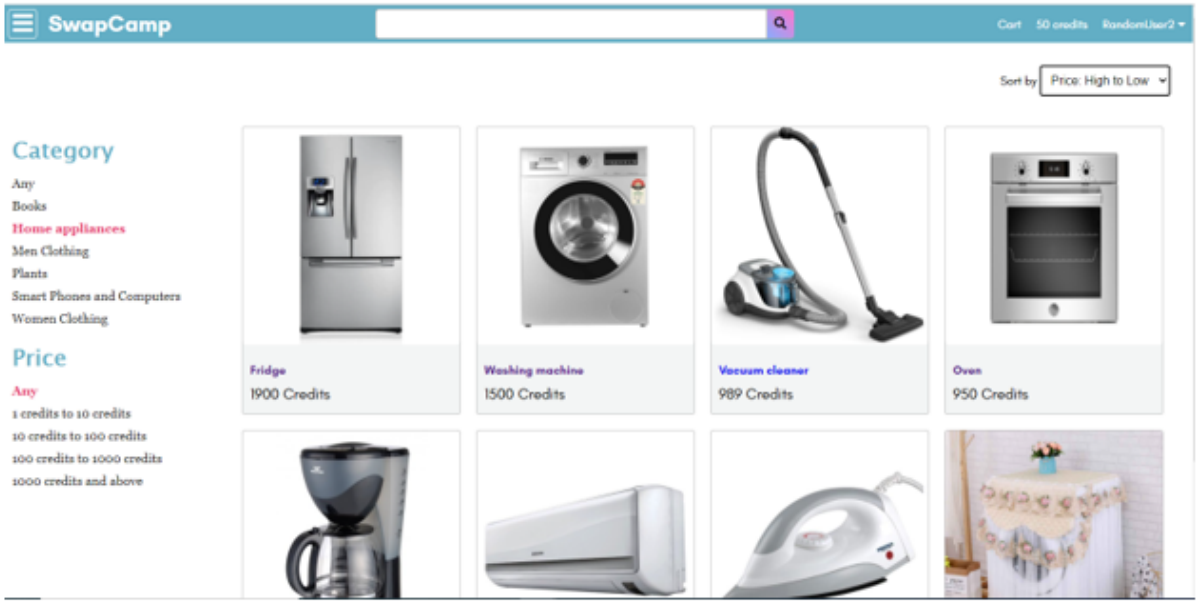

Figure 6.24: Home appliances category sorted by high to low.

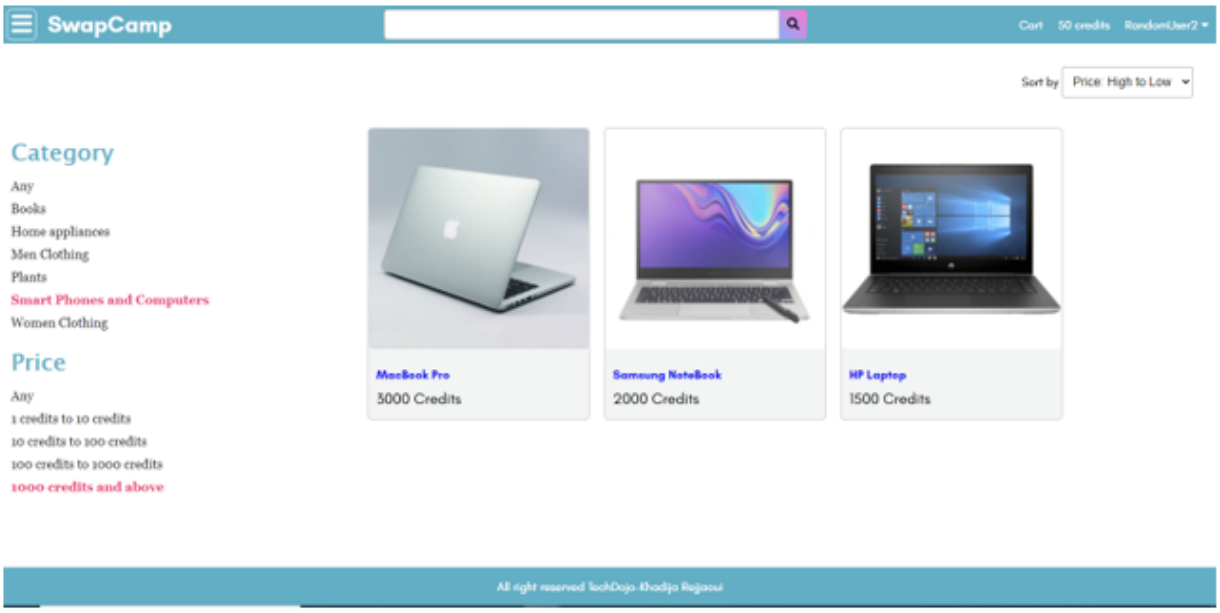

Figure 6.25: Smart phoned and computers category sorted by high to low and filtered by price.

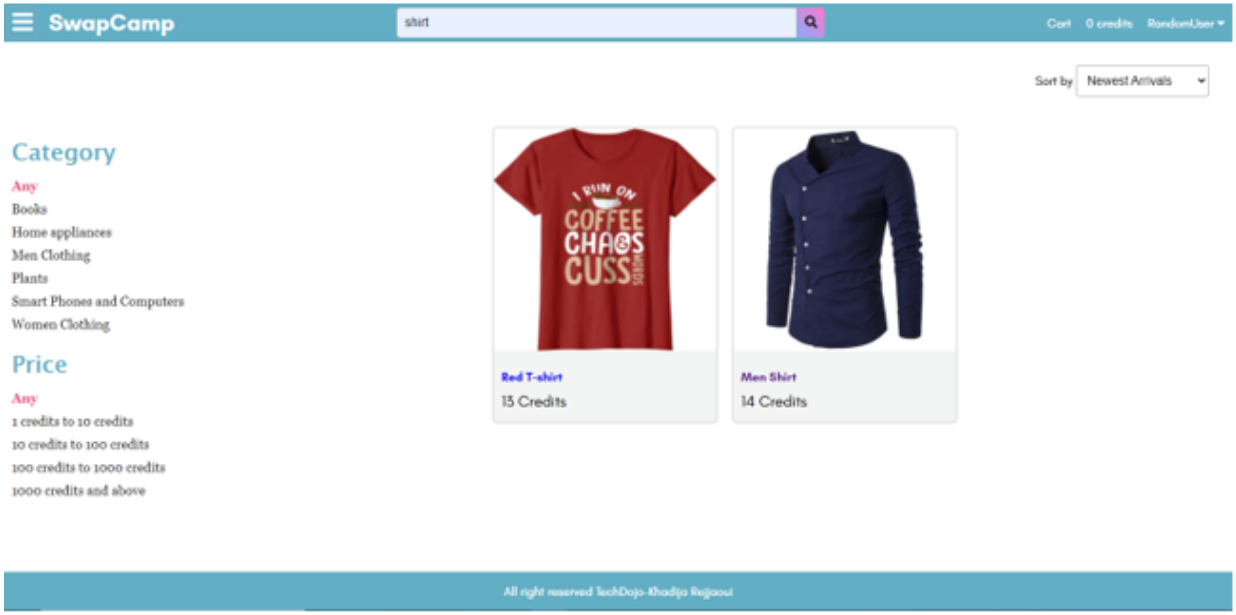

Figure 6.26: Search bar is used to find available shirts.

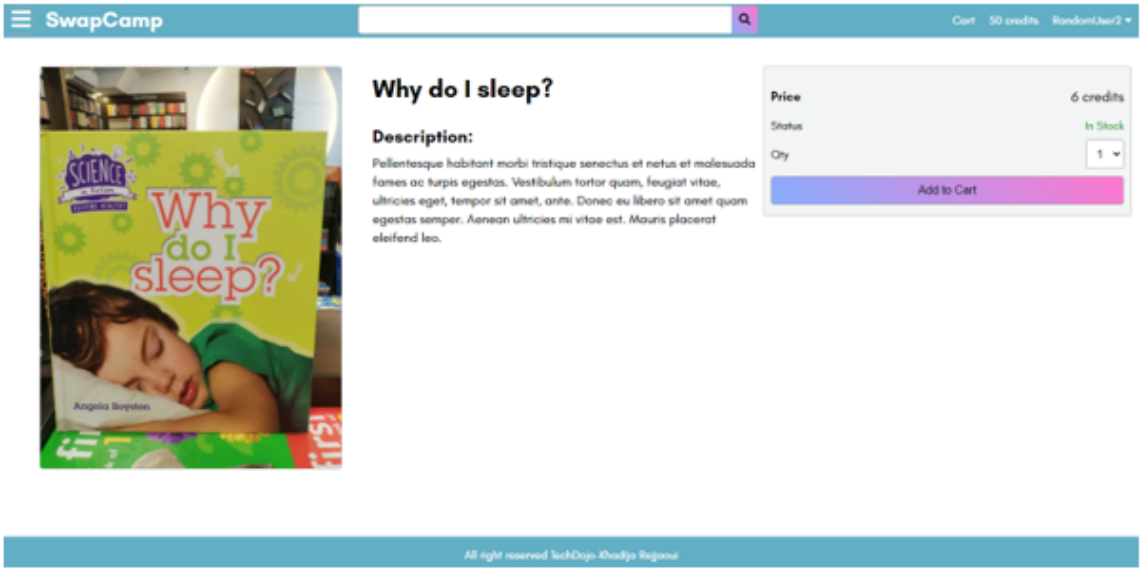

Figure 6.27: Product description.

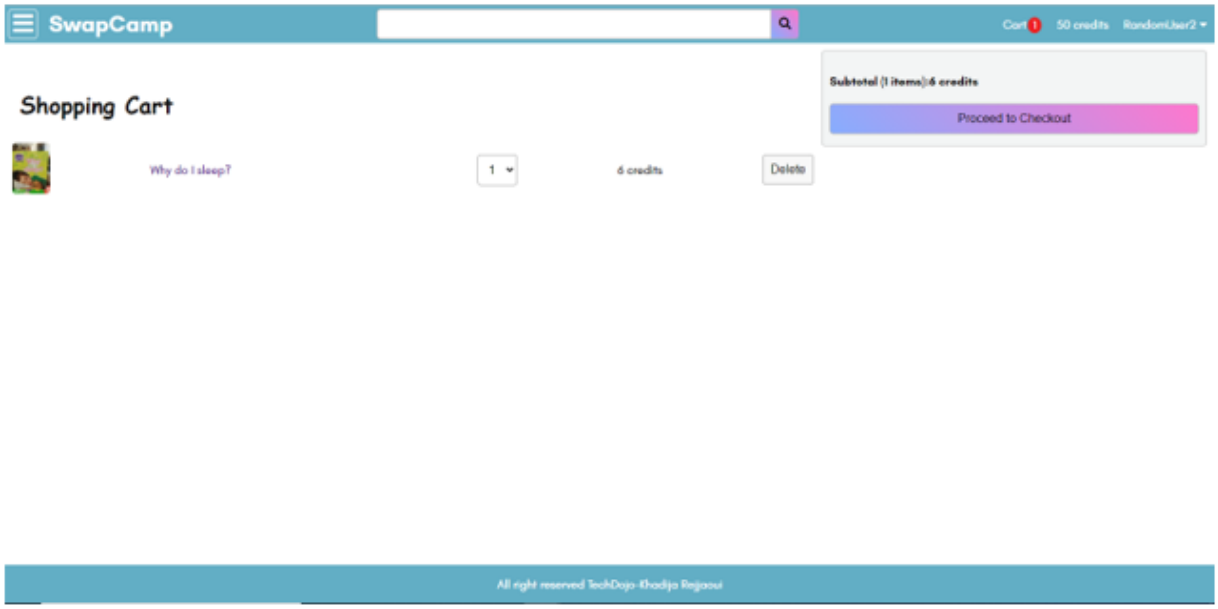

Figure 6.28: Shopping cart page.

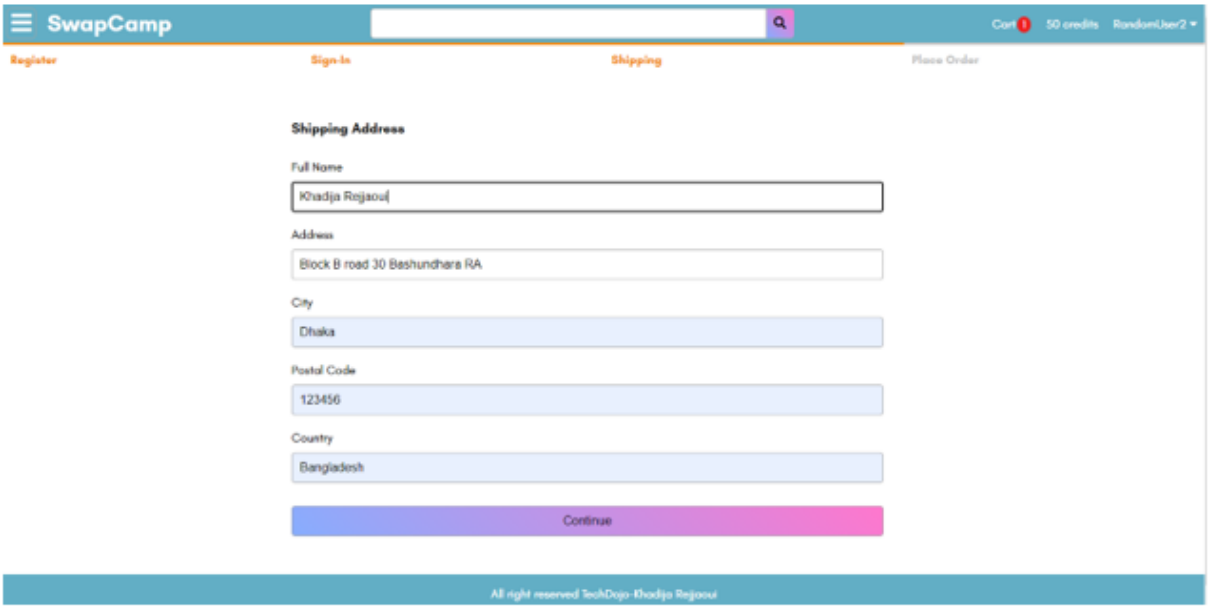

Figure 6.29: Shipping address form page.

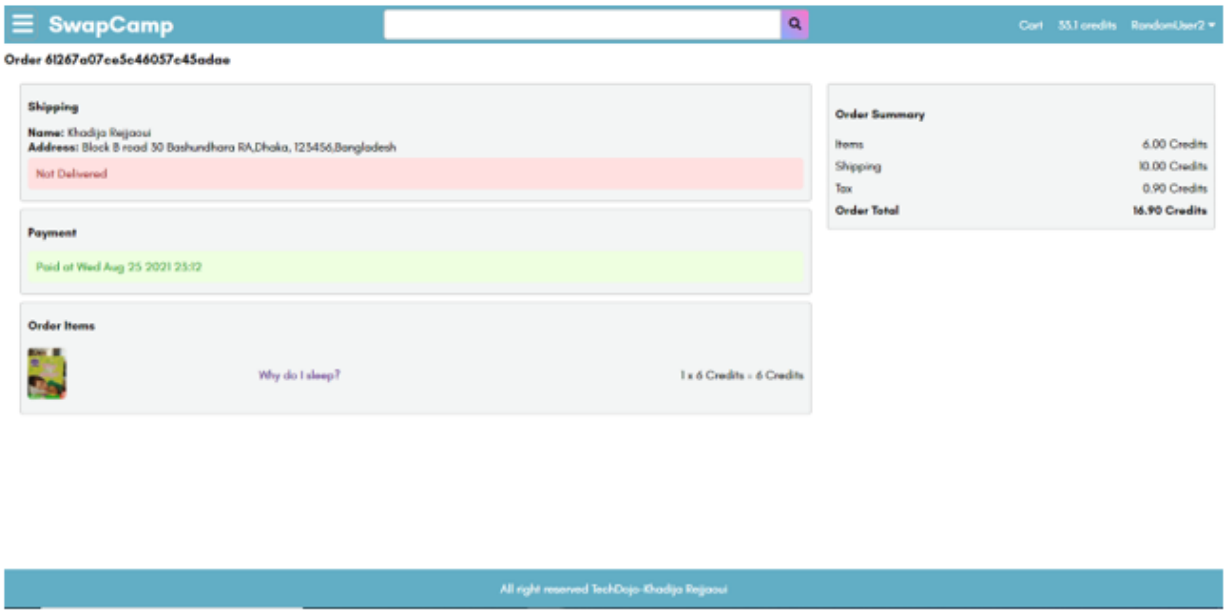

Figure 6.30: Order placed and credits got updated.

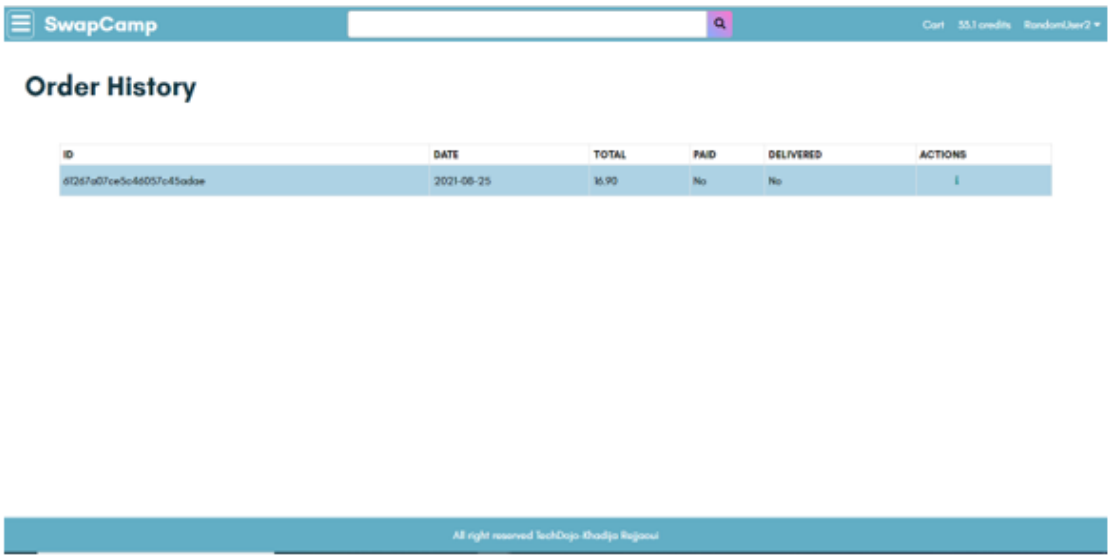

Figure 6.31: User order history.

- The Admin can:

- Add, delete, edit products.

- Check and delete orders.

- Make order as delivered.

- Delete, edit users, and assign credits.

- Check swaps and delete them.

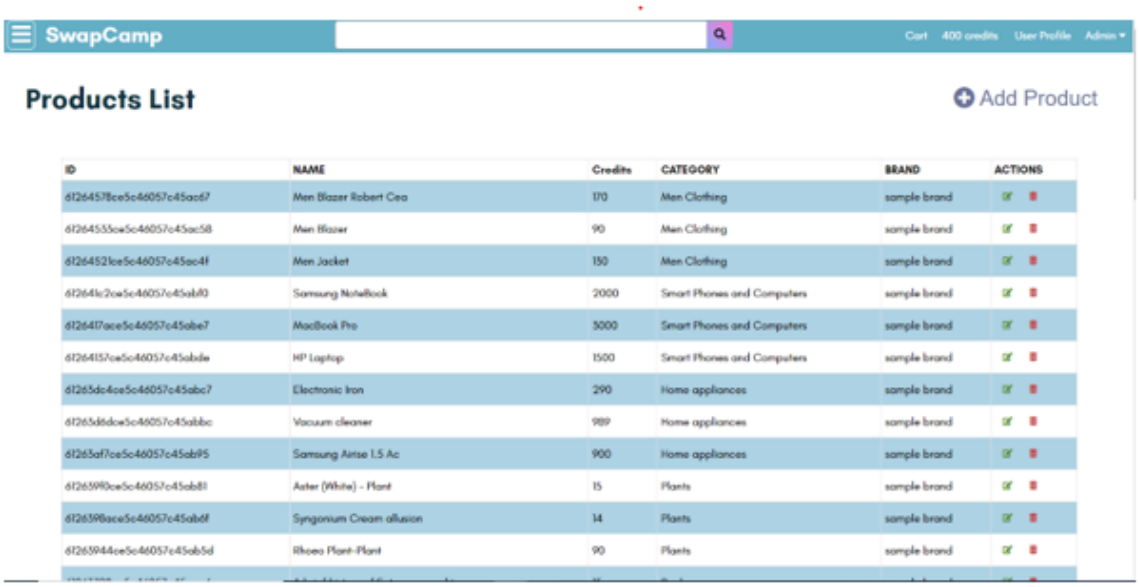

Figure 6.32: Admin products CRUD.

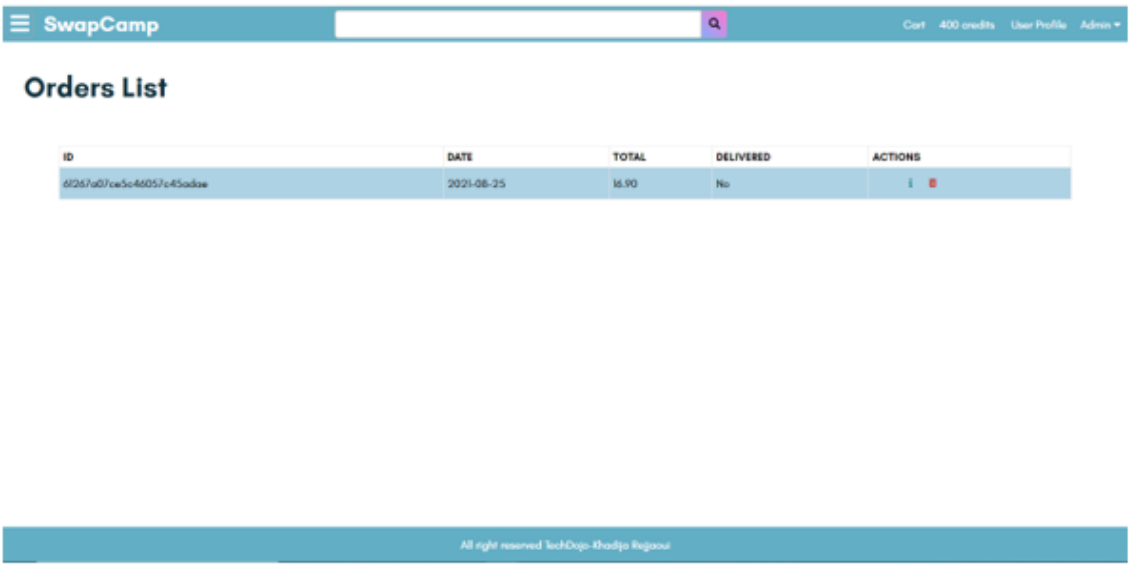

Figure 6.33: Admin orders CRUD.

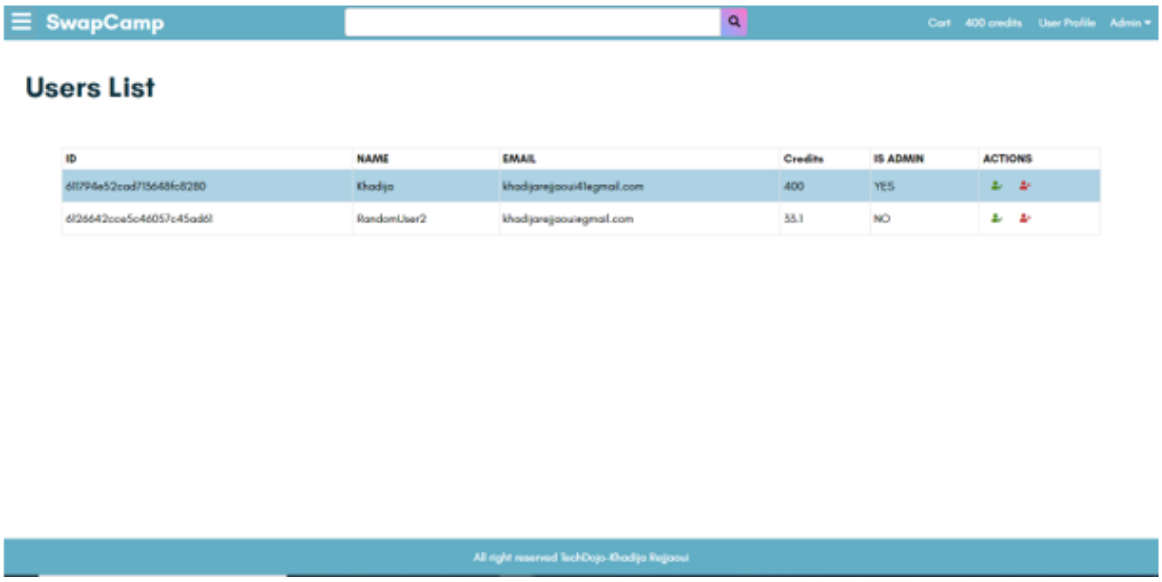

Figure 6.34: Admin users CRUD.

Chapter 7

Project as Engineering Problem Analysis

7.1 Sustainability of the Project/Work

Product sustainability refers to the ability to continually maintain and update the system. A product can be sustainable in three categories: Community, financial, and organization sustainability.

• Community Sustainability: This means how much the users will interact and support this project. There are many support forms such as using the application, subscribing to paid services, positive word of mouth, etc.
SwapCamp is a unique and new idea that will meet the requirements of users. For this reason, we believe that it will have a strong user base after its release. As the user base grows, the community will grow. Therefore, it is sustainable in terms of community.

• Financial Sustainability: this indicates how to maintain the cost of the application after its release and whether it will generate sufficient revenue as acceptable profits. The cost includes server cost, database storage cost, third-party API cost, etc.
When SwapCamp will be launches, it will have advertising and shipping costs to generate revenue. Also, in the future, there will be a monthly subscription to allow users to send as many items as they want at the same time. In addition, they will be able to buy credits using their PayPal or credit card. Moreover, they be able to get products from the store and pay them through cash on delivery. So, SwapCamp is Financially Sustainable.

• Organizational sustainability: relates to how the organization continues to operate after the application is released. After an app is released, an organization typically maintains the app via its existing team, extended team, or new team. Also, organizations update

their project by adding new features.
SwapCamp will be maintained and updated after its released. In addition to that, there are many planned features to be added to this project. In conclusion, it can be said that the project is Organizationally Sustainable.

7.2 Social and Environmental Effects and Analysis

Social Effects

The main reason for this web app is to help people save their money and stop spending them madly to get the items they want. On the other hand, to help others get any product they want in a cheaper way. As the saying goes, "One man's trash is another man's treasure." Through this site, we aspire to have a better society that contains the values of cooperation, caring, and sharing away from Extravagance and waste.

Environmental Effects

This project has many environmental dimensions that it seeks to achieve some of them are:

• Helping save water: We waste a lot of water making new products; for example, producing a pair of cotton jeans requires about 7,000 liters of water - the equivalent of filling more than 50 bathtubs. So, if we reduce the production of this type of clothing and start the habit of exchanging, we can save water.

• Stop Using Pesticides: During the manufacture of the products, a lot of pesticides that pollute the environment are used. Only the production of cotton uses about 11percent of global pesticides and 25 percent of the global insecticides.

• Saving Trees: When people will start the habit of exchanging books with each other, paper production will decrease, therefore tree cutting will decrease as well.

Chapter 8

Lesson Learned

8.1 Problems Faced During this Period

During my internship program, I have faced lots of challenges while working on this Project. The main ones are:

• Understand the tasks and requirements: Since I was new to the company, it was really hard to understand exactly what I had to do. .

• New Technologies: This was the first time, I have ever worked on a web application using react so, it was really difficult for me. .

•Time management: I had to complete a specific task on every week, and as I react was a new language for me, I had to do all my best to complete the task on a specific time. .

• Troubleshooting: Many times, some bugs were hard to find, and even after they were discovered, it became a huge problem to solve..

8.2 Solution of those Problems

As time went on, I started to have a better understanding of the requirements, it was all about being patient and listen to mentors' advice. Moreover, by practicing more and reading multiple documents, I was able to adapt to technologies. In addition, Planning and managing the time was the solution to keeping the speed up. And finally, searching on the internet and asking for help from my mentors was the solution to find bugs and solve them.

Chapter 9

Future Work & Conclusion

9.1 Future Works

Since SwapCamp is still in the development stage and many more features are planned to be added in the future. some of them are:

- Log in via other services such as Google or Facebook.
- Admin dashboard.
- Monthly and annual subscription.
- Develop an app for iOS and Android.
- Support chat.
- Show user statistics.
- Buy credits using PayPal and credit card.
- Pay through cash on delivery.
- Add delivery man as a user type and create his interfaces.

9.2 Conclusion

During my internship, we worked on a web application called SwapCamp. In this application, people can exchange products with each other indirectly and without any problem.

My experience with Techdojo Limited was very interesting, enjoyable, and full of learning.

I learned a lot about developing different types of applications. In addition, I worked with the latest technologies such as React- which was among the main lessons I gained from this internship. Furthermore, I got to know more about myself, what I can do and what I have to learn. I also collaborated with my mentors and seniors to solve problems. Moreover, this project indirectly helped me to be independent, patient, initiative, and able to solve problems. Besides, my communication skills are also enhanced when communicating with others. In addition, I developed my programming skills and sharpened the skills I learned throughout my university life.

Bibliography

[1] "Work breakdown structure - wikipedia."

[2] "What is a gantt chart? gantt chart software, information, and history."

[3] "What is sdlc? understand the software development life cycle – stackify."

[4] "What is extreme programming (xp)? — agile alliance."

[5] "What is extreme programming (xp)? — agile alliance."

[6] "7 types of feasibility analysis - simplicable."

[7] "Uml - standard diagrams."

[8] "What is a functional requirement in software engineering? specification, types, examples."

[9] "What is a functional requirement in software engineering? specification, types, examples."