Phase II: User Requirements and Application Specifications
Submission Deadline: 18.03.2024, 23:59

Chosen Development Model:

I have chosen the **Agile** development model due to its flexibility and adaptability to changes in project requirements. Since the application I have decided to develop has a very dynamic nature due to the continuous changes that may be required over time, the Agile development model would be the most suitable to answer these requirements, because it enables me to engage in repetitive development cycles and receive ongoing feedback regarding the continuation of my work, and the ability to respond to changes effectively.

Furthermore, since one of my main goals is to make the application user-friendly, it will need to be constantly modified while it is being developed to produce the desired results and fulfill my earlier point.

User Requirements:

a. Stakeholders:

- **End Users:** Customers seeking skincare products. Customers are the main users of this software. They interact with the software while searching for different products, selecting the categories they are interested in, selecting the products they are willing to buy, and having the ability to buy them by giving their information. Other actions they can/will be required to do include login, logout, sign in, confirm or cancel order, etc.

  Administrators within the skincare company or e-commerce platform may have access to the website for administrative purposes. They may be in charge of managing user accounts, supervising product listings, keeping an eye on sales analytics, and checking product quantity.

- **Developers:** The software developer (me) is responsible for designing developing and maintaining the software. I tend to deliver a product that meets all the customer's needs, such as a user-friendly, secure, and efficient software solution.

- **Other Parties:** The marketing team may access the website to develop marketing campaigns and update product descriptions. Furthermore, Regulatory authorities would have access to compliance with data protection laws and consumer rights.

User Stories:

- Client: As a client, I want to browse skincare products based on their purpose, brand, and price range so I can find suitable products more easily.

Requirement: Develop an interface with categorized products based on different criteria.
Benefit: Develop a user-friendly interface, that enables users to find products according to their needs leading to client satisfaction.

- Client: As a client, I want to read reviews from other customers' experiences to make informed decisions about my purchases.

Requirement: Create a review section for the products available.
Benefit: Client misunderstandings and dissatisfaction will be reduced, increasing the software's likeability.

- Administrator: As an administrator, I want to add new products to the website with their description, images, pricing information, and categorization so there will be a large variety for clients to choose from.

Requirement: Develop a product management system for administrators to add new products.
Benefit: The website will be easily adaptable to the users' needs, such as the necessity to add new products.

- Administrator: As an administrator, I want to process order cancellations by users.

Requirement: Develop an order cancellation management system accessible only to administrators.
Benefit: Increases client satisfaction by responding to their needs and requirements.

Functional Requirements:

b. Brief Description:

- Clients should be able to create an account, log in, and log out.
- Clients should be able to search for skincare products either in their categories or by keyword.
- Clients should be able to see the product details before adding them to their cart.
- Clients should be able to manage the products added to their shopping cart.
- Clients should be able to buy the products they want, check the details of their orders, and manage them.

c. Acceptance Criteria:

- Users can register to their account by using their credentials such as name, email, and password and securely log in or log out of the account.
- Users can browse the product they need by looking at the relevant category such as price range, purpose, and skin condition, or by using the available search bar.
- The product details such as name, picture, price, and description are going to be visible on the products so the users can see them.
- Users can add new products to their cart, view cart content, and update the product quantities.
- Users make an order, see the order status, and details such as shipping information, and see the order history.

Non-Functional Requirements:

d. Brief Description:

- The software must be user-friendly and easy to navigate.
- The software must have high performance and not be intersected by ads or cookies.
- The software must guarantee the privacy of customers' information and security.
- The software must be able to endure usage from a large number of clients at the same time.
- The system must be compatible with the major web browsers (Chrome, Firefox, Safari, Edge).

Acceptance Criteria:

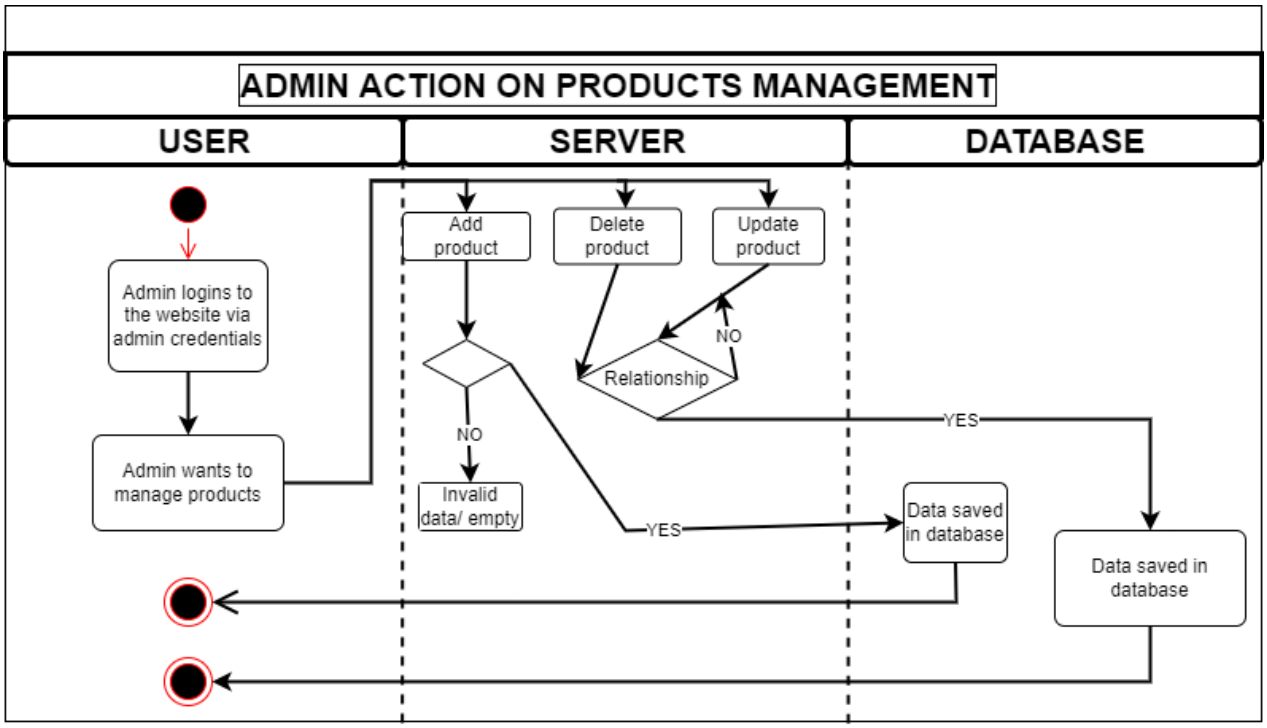
- Users can easily navigate through the system with a consistent layout and clear menus.
- All pages and features of the software must have a responding time of a maximum of 2 seconds in normal operating conditions.
- User passwords are stored by using strong encryption techniques and do not allow data breaches and users' data is visible only to the owner and not to the other users.
- The system must be able to handle a large number of users at the same time without lowering performance.
- The software must function consistently with all kinds of browsers so client usage won't be restricted by their browser.

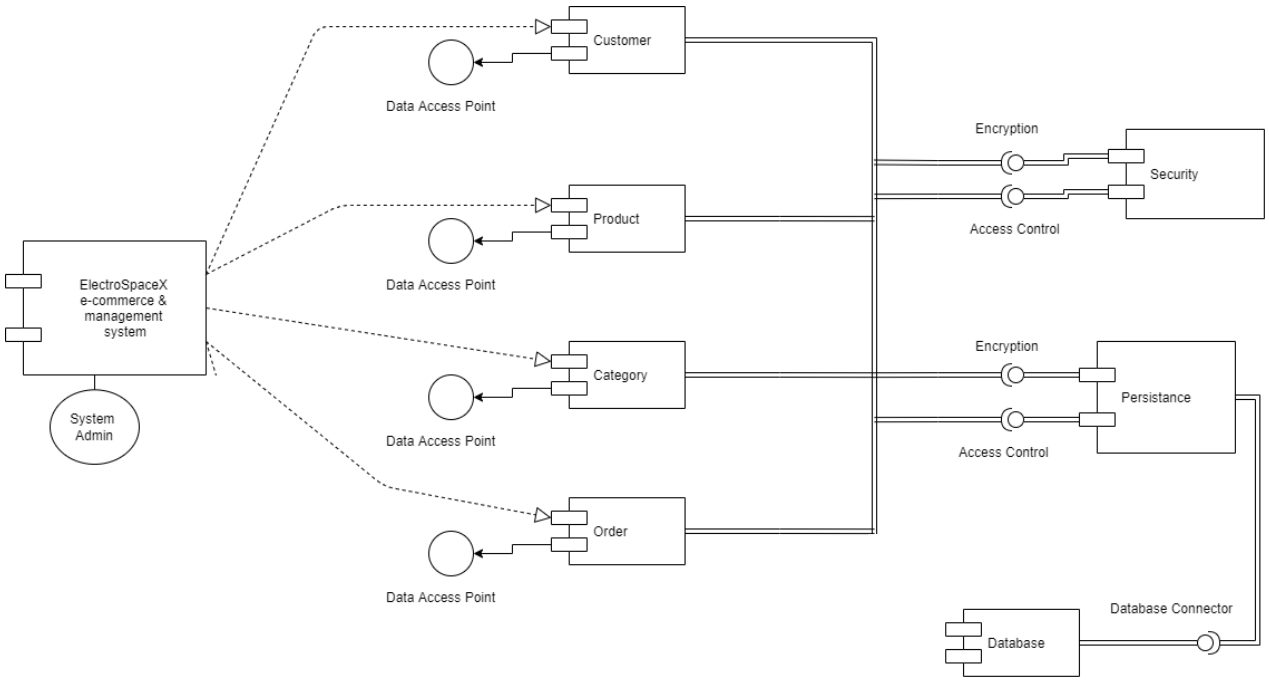
Application Specifications:

e. Architecture:

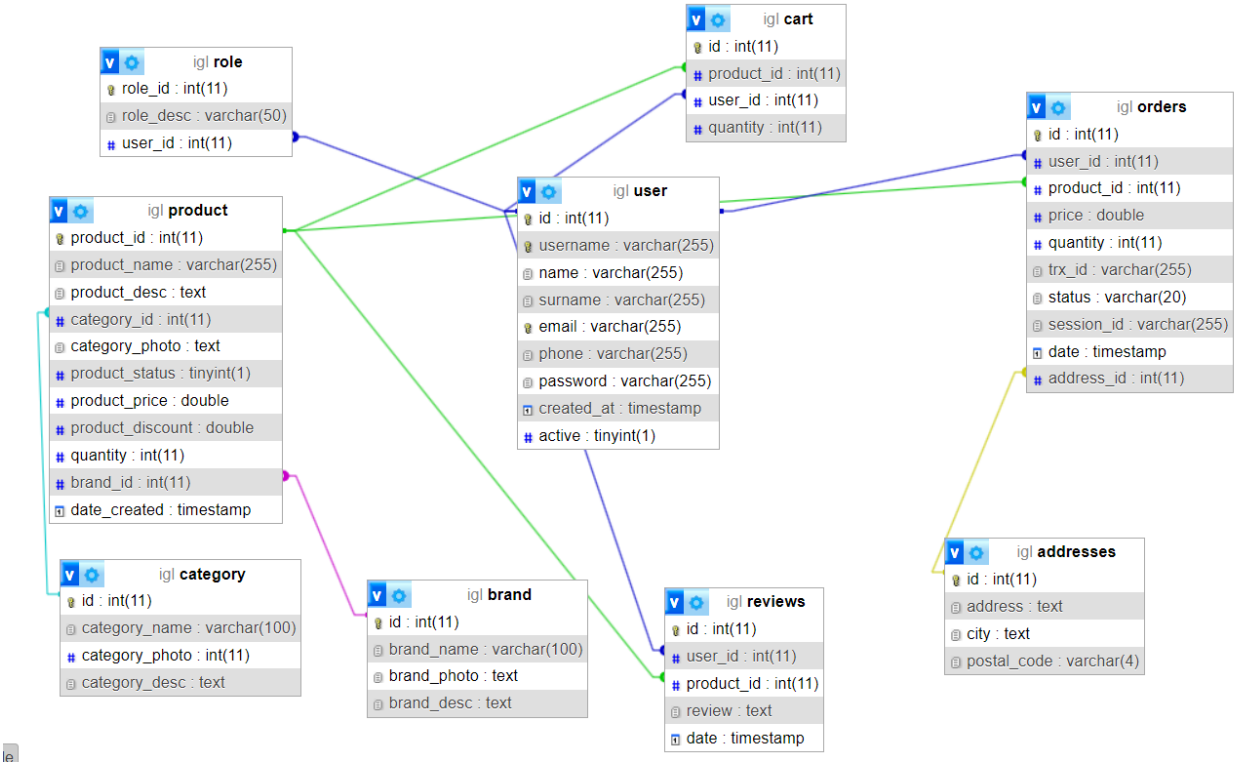
The system will follow a client-server architecture, with a web server hosting the application and serving client requests. The system follows the client-server RESTful API communication with HTTPS connections.

The system architecture utilizes a modern approach, using a combination of frontend and backend technologies to create an e-commerce skincare platform. The frontend is developed using HTML, CSS, JavaScript, and the React.js library, providing a dynamic and interactive user interface. On the backend, Node.js and Express.js handle the server-side logic and REST API endpoints, ensuring efficient data processing between the client and server. The MySQL database is utilized to store and manage product information, user data, and order details, complying ACID requirements. Security measures such as encryption using SSL/TLS and JWT for authentication are implemented to guard sensitive information and authenticate user access. Overall, this architecture enables the creation of a secure and responsive e-commerce platform tailored to meet the needs of skincare product shoppers.


ADMIN ACTION ON PRODUCTS MANAGEMENT
USER
SERVER
DATABASE
Admin logins to the website via admin credentials
Admin wants to manage products
Add product
Delete product
Update product
Relationship
NO
YES
NO
Invalid data/ empty
YES
Data saved in database
Data saved in database


ElectroSpaceX e-commerce & management system
System Admin
Customer
Product
Category
Order
Data Access Point
Data Access Point
Data Access Point
Data Access Point
Encryption
Security
Access Control
Encryption
Persistance
Access Control
Database
Database Connector

Database Model:



1. **Table: addresses**
    - Fields: **id** (Primary Key), **address**, **city**, **postal_code**
    - No relationships with other tables.
    - No constraints other than the primary key constraint.
2. **Table: brand**
    - Fields: **id** (Primary Key), **brand_name**, **brand_photo**, **brand_desc**
    - No relationships with other tables.
    - No constraints other than the primary key constraint.
3. **Table: cart**
    - Fields: **id** (Primary Key), **product_id** (Foreign Key referencing **product**), **user_id** (Foreign Key referencing **user**), **quantity**
    - Relationships:
        - **product_id** references **product(product_id)**
        - **user_id** references **user(id)**
    - Constraints:
        - Foreign key constraints on **product_id** and **user_id**.

4. **Table: category**
    - Fields: **id** (Primary Key), **category_name**, **category_photo**, **category_desc**
    - No relationships with other tables.
    - No constraints other than the primary key constraint.
5. **Table: employees**
    - Fields: **id** (Primary Key), **name**, **surname**, **salary**
    - No relationships with other tables.
    - No constraints other than the primary key constraint.
6. **Table: orders**
    - Fields: **id** (Primary Key), **user_id** (Foreign Key referencing **user**), **product_id** (Foreign Key referencing **product**), **price**, **quantity**, **trx_id**, **status**, **session_id**, **date**, **address_id** (Foreign Key referencing **addresses**)
    - Relationships:
        - **user_id** references **user(id)**
        - **product_id** references **product(product_id)**
        - **address_id** references **addresses(id)**
    - Constraints:
        - Foreign key constraints on **user_id**, **product_id**, and **address_id**.
7. **Table: product**
    - Fields: **product_id** (Primary Key), **product_name**, **product_desc**, **category_id** (Foreign Key referencing **category**), **category_photo**, **product_status**, **product_price**, **product_discount**, **quantity**, **brand_id** (Foreign Key referencing **brand**), **date_created**
    - Relationships:
        - **category_id** references **category(id)**
        - **brand_id** references **brand(id)**
    - Constraints:
        - Foreign key constraints on **category_id** and **brand_id**.
8. **Table: reviews**
    - Fields: **id** (Primary Key), **user_id** (Foreign Key referencing **user**), **product_id** (Foreign Key referencing **product**), **review**, **date**
    - Relationships:

        - **user_id** references **user(id)**
        - **product_id** references **product(product_id)**
    - Constraints:
        - Foreign key constraints on **user_id** and **product_id**.

9. **Table: role**
    - Fields: **role_id** (Primary Key), **role_desc**, **user_id** (Foreign Key referencing **user**)
    - Relationships:
        - **user_id** references **user(id)**
    - Constraints:
        - Foreign key constraint on **user_id**

10. **Table: user**
    - Fields: **id** (Primary Key), **username**, **name**, **surname**, **email**, **phone**, **password**, **created_at**, **active**
    - No relationships with other tables.
    - Constraints:
        - Unique constraints on **username** and **email**.

Technologies Used:

- Frontend: HTML, CSS, JavaScript, React.js.
- Backend: Node.js, Express.js.
- Database: MySQL.
- Security: Encryption (SSL/TLS), JWT for authentication.

1. **Frontend:**
    - **HTML, CSS, JavaScript:** These are fundamental web development languages used to create the structure, style, and interactivity of web pages. HTML provides the structure, CSS handles styling, and JavaScript adds interactivity.
    - **React.js:** React.js is a popular JavaScript library for building user interfaces. It offers component-based architecture and a rich ecosystem of libraries and tools. React.js was chosen for its performance, reusability, and ease of managing complex UI components in a modular manner.
2. **Backend:**
    - **Node.js:** Node.js is a runtime environment that allows JavaScript to be

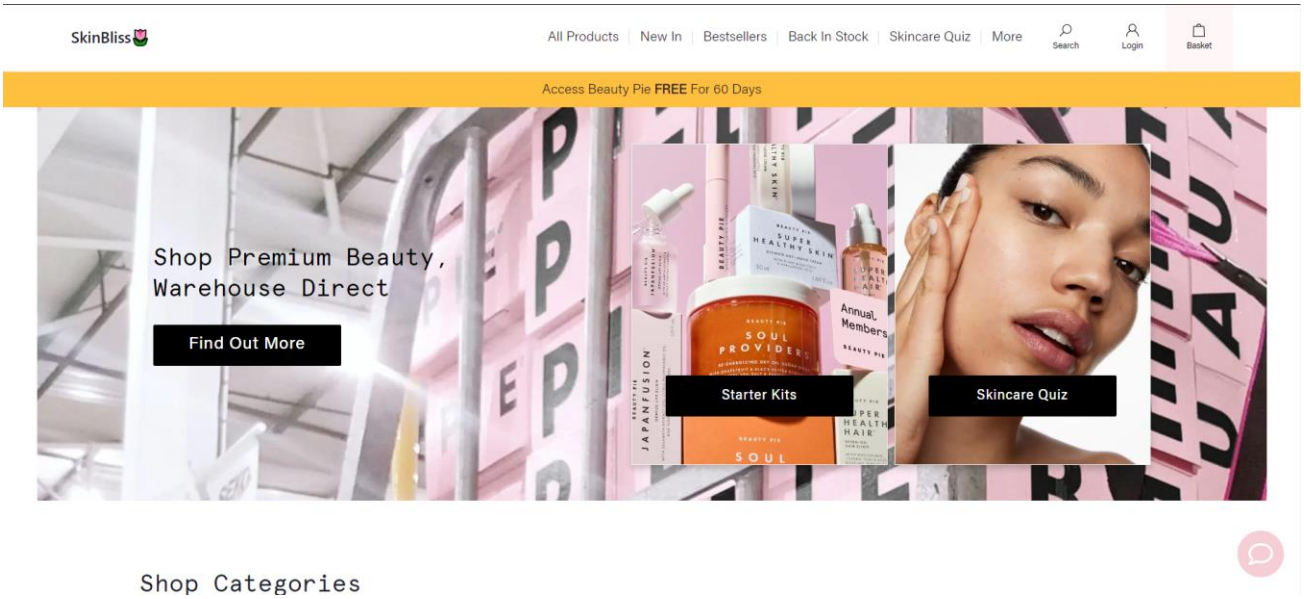
executed on the server-side. Node.js was chosen for its efficiency, scalability, and ecosystem libraries.

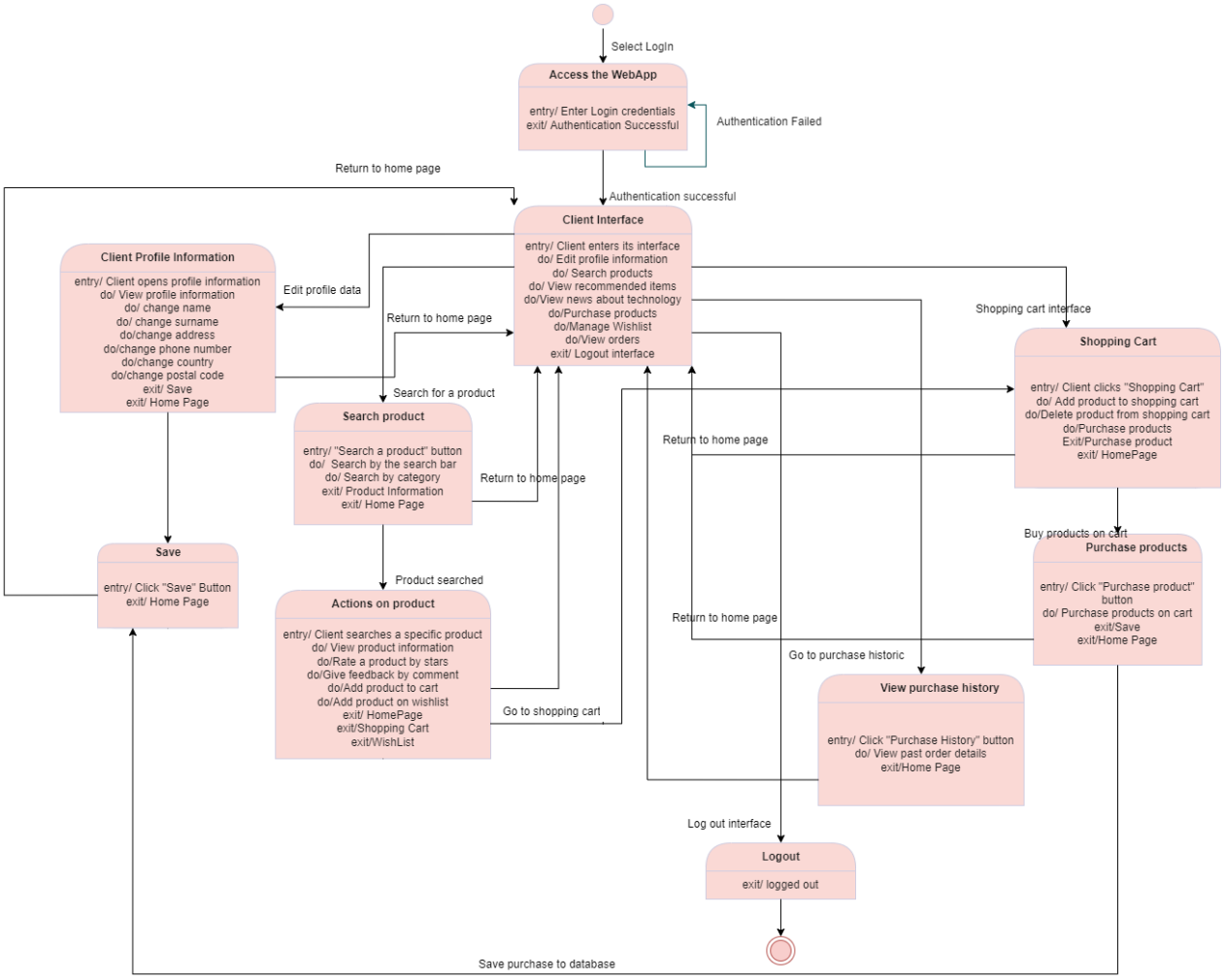
- **Express.js:** Express.js is a web application framework for Node.js. It simplifies routing and request handling, making it ideal for building RESTful APIs and handling HTTP requests/responses.

3. **Database:**

- **MySQL:** MySQL is a popular open-source relational database management system (RDBMS) known for its reliability, scalability, and performance. It provides features such as ACID compliance, transaction support, and data security.

User Interface Design:





A short overview of the software interface: The client will have the ability to create an account and then login/out of the account. After login in it will appear the frontpage with all the products displayed in different categories, from which the user can select products to check and add to the cart/basket. The user can see the carts' content, change the quality of the cart's products, and finally check in by adding all his/her information which will be strictly confidential. A user can check by their account the previous orders, orders status, or products saved. On the other hand, the admin can add or remove products from the software and also can view the customers' orders, and their status as well as cancel them.

Homepage:

The layout of the homepage will be simple and visually appealing. The first thing users will notice is the sectioning created for the various skin conditions, each of which will direct them to a separate page with the relevant information. Additionally, each product's other categories will be displayed on the side. Users can access different sections of the website, such as product categories, account management, and shopping carts, by using the navigation menus located at the top or side of the page.

Product Listings:

Skincare products will be shown in product listings in a list or grid format. There will be an

image, title, price, rating, and a brief description for every product. Users can click on a product to view the additional details. On the other hand customer reviews and ratings will be displayed to help the other users make informed purchasing decisions.

Shopping Cart:

Products from product listings or details pages can be added to users' shopping carts with just one click. The shopping cart page will display a summary of all selected items, including quantities, prices, and subtotal. Users can adjust quantities, remove items, or proceed to checkout directly from the shopping cart.

Checkout Process:

Users will be guided through the necessary steps to finish their purchase during the checkout process. The user will securely submit payment information, choose a shipping method, and enter their shipping address. Before completing the purchase, you can review the order summary and total cost. Users will receive an email with order details and a confirmation message after the process is successful.

User Account:

Registered users can view order history, manage addresses, and update personal data via their account dashboard. Users will be able to change communication preferences and email notification settings through their account settings. Users can log in or sign up for an account easily from any page of the website.

Security Measures:

Briefly discuss the security measures and protocols implemented: Touch upon encryption, authentication, and any other security features.

- Encryption (SSL/TLS): Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocols are used to encrypt data transmitted between the client and server, ensuring confidentiality and integrity. This encryption mechanism protects sensitive information, such as user credentials and payment details, from eavesdropping and tampering during transmission over the internet.
- Authentication (JWT): JSON Web Tokens (JWT) are utilized for authentication, providing a stateless mechanism for verifying the identity of users. JWT tokens are generated upon successful login and include user authentication data, such as user ID and roles. These tokens are securely stored on the client side and sent with each request to authenticate and authorize user access to protected resources.
- Authorization: This parts is mostly related with the role of the person that is going to log in on the software(user or admin) where the role based access will control the actions of the both sides and limit their functions.