FINAL YEAR PROJECT

GROUP No: 21

E-COMMERCE STORE RECOMMENDATION SYSTEM

**DESCRIPTION**:

E-commerce stores use recommendation systems to provide personalized and relevant product and store recommendations to their users, based on their interests and behavior. These systems use machine learning algorithms to analyze user data, such as browsing history, search queries, and past purchases, to provide recommendations for products and stores that are likely to be of interest.

The first step in building an e-commerce recommendation system is to collect user data. This can include information about the user's location, demographic information, and past interactions with the website. Machine learning algorithms are then trained on this data to identify patterns and make predictions about which products and stores a particular user is likely to be interested in.

One popular approach for building e-commerce recommendation systems is collaborative filtering. This algorithm uses the similarity between users' preferences and behaviors to recommend products and stores to customers. In other words, if two users have similar purchase histories and behavior on the website, the algorithm will recommend products and stores that one user has purchased from or visited to the other user.

Another approach is content-based filtering, which uses the characteristics of the products and stores, such as product categories, prices, and promotions, to recommend products and stores to customers. For example, if a user has searched for and purchased products related to a particular category, the algorithm will recommend products and stores that specialize in that category.

The front-end of the application is developed using modern web development technologies such as HTML, CSS, and JavaScript, while the back-end is developed using Python and Django. The database is built using MySQL.

The application has a user-friendly interface that allows users to browse products, add them to their cart, and make purchases. The recommendation system is seamlessly integrated into the application, and users can view personalized product recommendations on the home page or a dedicated recommendations page.

The application is scalable, and the recommendation system can handle large amounts of data, ensuring that the system can cater to a growing user base. The system can also be customized to suit the needs of different e-commerce stores, and additional features can be added to enhance the shopping experience.

**SUPERVISOR**: SIR FAHAD NAJEEB

**GROUP MEMBERS:**

1. NAME: SAMEER KHAN

EMAIL ADDRESS: Sameer.53051@iqra.edu.pk

REGISTRATION ID: 53051

1. NAME: ALIRAZA KHAN

EMAIL ADDRESS:aliraza.53081@iqra.edu.pk

REGISTRATION ID: 53051

1. NAME: ANEESH RAJA NAGDEV

EMAIL ADDRESS:aneesh.53104@iqra.edu.pk

REGISTRATION ID: 53104

**FEATURES:**

* **User** **profiles**: Users can create profiles that capture their preferences, purchase history, and browsing behavior.
* **Product catalog**: The system has a comprehensive catalog of products that are available for purchase on the e-commerce store.
* **Personalized product recommendations**: The system generates personalized product recommendations for users based on their profiles, browsing behavior, and purchase history.
* **Search functionality**: The system provides a search bar that allows users to search for products by name, category, or brand.
* **Product ratings and reviews**: Users can rate and review products, helping other users make informed purchasing decisions.
* **Cart and checkout:** The system has a shopping cart that allows users to add products to their cart and make purchases securely.
* **Order history:** Users can view their order history, which includes details of past purchases and delivery status.
* **Notifications**: The system sends notifications to users about order updates, new products, and personalized recommendations.
* **Admin panel**: The system has an admin panel that allows store administrators to manage products, orders, users, and other aspects of the e-commerce store.
* **Security:** The system ensures that all transactions are secure and encrypted, protecting user data and financial information.

**Functional Requirements:**

* User Registration and Authentication: Users should be able to create an account and log in to access the recommendation system.
* Product Catalogue: The system should have a catalog of products that are available for purchase on the E-commerce store.
* User Profile Management: Users should be able to manage their profiles, update their personal information, and preferences that will help the recommendation system to suggest relevant products.
* Recommendation Engine: The core functionality of the system should be the recommendation engine that utilizes machine learning algorithms to provide personalized product recommendations to users.
* User Feedback System: The system should allow users to provide feedback on recommended products to improve the accuracy of future recommendations.
* Order Tracking and Management: The system should enable users to track their orders and manage their purchase history.

**Non-Functional Requirements:**

* Performance: The system should be fast and responsive, with minimal downtime.
* Scalability: The system should be designed to handle a large number of users and products and be able to scale up or down based on user demand.
* Security: The system should be secure, with appropriate measures in place to protect user data and prevent unauthorized access.
* Accuracy: The recommendation engine should provide accurate recommendations that are relevant to the user's preferences and history.
* User Experience: The system should have a user-friendly interface that is easy to navigate and understand.
* Maintenance: The system should be easy to maintain and update, with minimal effort required from the development team.

**DETAILED PLAN**

**STEP1-COLLECTION OF DATA:**

When collecting data for an e-commerce store recommendation system, you may consider collecting the following types of data**:**

**Product data:** This includes information about the products available on the e-commerce store, such as product descriptions, images, prices, categories, and attributes (e.g., size, color, material).

**User data:** This includes information about the users of the e-commerce store, such as their purchase history, browsing behavior, search queries, and demographic information (e.g., age, gender, location).

**Feedback data:** This includes information about user feedback on products and recommendations, such as product ratings, reviews, and comments.

**External data:** This includes information from external sources that can enrich the recommendation system, such as social media data, weather data, or news data.

**STEP 2-DATA PREPROCESSING:**

**Data cleaning:** The first step is to clean the collected data and remove any irrelevant or erroneous information. This can include removing duplicate records, fixing inconsistencies, and handling missing or incomplete data.

**Data normalization:** The next step is to normalize the data by scaling and transforming it into a standardized format. This can involve techniques such as feature scaling, mean normalization, or logarithmic transformation.

**Data reduction:** In some cases, the collected data may be too large or complex to be processed efficiently. In this case, data reduction techniques can be used to reduce the data size or dimensionality while preserving its essential features. Examples of data reduction **techniques** include principal component analysis (PCA), feature selection, and feature extraction**.**

**Data integration:** If multiple data sources are used, they need to be integrated into a single dataset that can be used for training and evaluation. This can involve merging, joining, or concatenating datasets, or using data fusion techniques to combine the data.

**Data splitting:** The final step is to split the preprocessed data into training, validation, and test sets. The training set is used to train the recommendation algorithm, the validation set is used to fine-tune the algorithm parameters, and the test set is used to evaluate the algorithm's performance.

**STEP 3- FEATURE EXTRACTION:**

**Identify relevant features:** The first step in feature extraction is to identify the features that are relevant for making recommendations. This can include user features, such as demographic information and purchase history, and product features, such as product category, price, and popularity.

**Encoding categorical features:** If some of the features are categorical, they need to be encoded into a numerical format that can be processed by the recommendation algorithm. This can involve techniques such as one-hot encoding, label encoding, or binary encoding.

**Handling missing data:** If some of the features have missing values, they need to be imputed or filled in with appropriate values. This can involve techniques such as mean imputation, median imputation, or regression imputation.

**Feature scaling:** Some of the features may have different scales or units, which can make it challenging to compare and combine them. To address this, feature scaling techniques can be used to rescale the features into a common range or unit. Examples of feature scaling techniques include min-max scaling, standardization, and normalization.

**Feature engineering:** In some cases, it may be necessary to create new features or transform existing features to capture additional information or improve the performance of the recommendation algorithm. This can involve techniques such as feature aggregation, feature interaction, or feature selection.

**STEP-4 ARCHITECTURE:**

Collect data from user profiles, purchase history, and product catalog.

Preprocess the data to ensure consistency and remove noise.

Extract relevant features and encode them into a numerical format.

Build a recommendation algorithm using techniques such as collaborative filtering or content-based filtering.

Evaluate the algorithm using appropriate metrics and fine-tune its parameters.

Present recommendations to users in a user-friendly interface.

Deploy and maintain the system, monitoring performance and retraining the algorithm as needed.

**STEP-5 MODEL TRAINING:**

**Algorithm Development:**algorithm development involves selecting an appropriate algorithm, tuning its parameters, building a robust algorithm, evaluating its performance, and selecting it for deployment. This ensures that the recommendation system has high accuracy and robustness.

**Algorithm testing and validation:** Algorithm testing and validation involve evaluating the performance of the developed algorithm on a separate testing dataset. This ensures that the algorithm generalizes well to new data and is not overfitting to the training dataset. Techniques such as k-fold cross-validation and holdout validation can be used to evaluate the algorithm's performance. The evaluation results can be used to fine-tune the algorithm parameters and improve its performance.

**Evalution Metrics:** Evaluation metrics are used to measure the performance of the developed algorithm. Common evaluation metrics for a recommendation system include precision, recall, F1 score, and mean average precision. These metrics provide insight into how well the algorithm is able to recommend relevant items to users. The evaluation results can be used to fine-tune the algorithm parameters and improve its performance**.**

**Algorithm Deployment:** algorithm deployment involves integrating the developed algorithm into the production environment. This includes deploying the model on a cloud platform or on-premise infrastructure, creating APIs for the model, and integrating it with the user interface. It is important to ensure that the deployed algorithm is scalable, reliable, and able to handle real-world traffic. Ongoing monitoring and maintenance of the deployed algorithm are also necessary to ensure its continued effectiveness.

**STEP-6 DEVELOPMENT OF USER INTERFACE:**

The development of the user interface involves creating an interactive and user-friendly interface for the recommendation system. This includes designing the layout and visual elements of the interface, implementing the frontend and backend components of the interface, and integrating it with the deployed recommendation algorithm. The user interface should provide users with an intuitive and engaging experience while also presenting relevant and personalized recommendations. It is important to continuously gather feedback from users and iterate on the user interface to improve its usability and effectiveness.

**STEP -7 TESTING AND EVALUTION:** the developed recommendation system is tested on various datasets and evaluated using relevant evaluation metrics. This involves using a testing dataset to simulate how the recommendation system would perform in a real-world scenario and measuring its accuracy, robustness, and efficiency. The evaluation metrics used for testing and evaluation may include precision, recall, F1 score, mean average precision, and others, depending on the specific requirements of the recommendation system. The results of the testing and evaluation phase can be used to fine-tune the recommendation system, improve its performance, and provide insights for future development.

**STEP-8 DEPLOYMENT AND MAINTENANCE:**In the deployment and maintenance phase, the recommendation system is deployed in a production environment and maintained to ensure its continued effectiveness. This involves integrating the system with the necessary infrastructure and services, monitoring its performance and usage, and making any necessary updates or changes to the system. The deployment and maintenance phase may also include implementing security measures, such as access control and data encryption, to protect the recommendation system and its users from potential threats. Ongoing monitoring and maintenance are necessary to address any issues that may arise and ensure the recommendation system remains up-to-date and relevant. Additionally, user feedback and usage analytics can be used to guide future development and updates to the recommendation system.

**LANGUAGE USED TO DEVELOP THE SYSTEM:**

**FRONTEND:**

HTML

CSS

JavaScript

React

Angular

Vue.js

**BACKEND:**

Python

Flask

Django rest framework

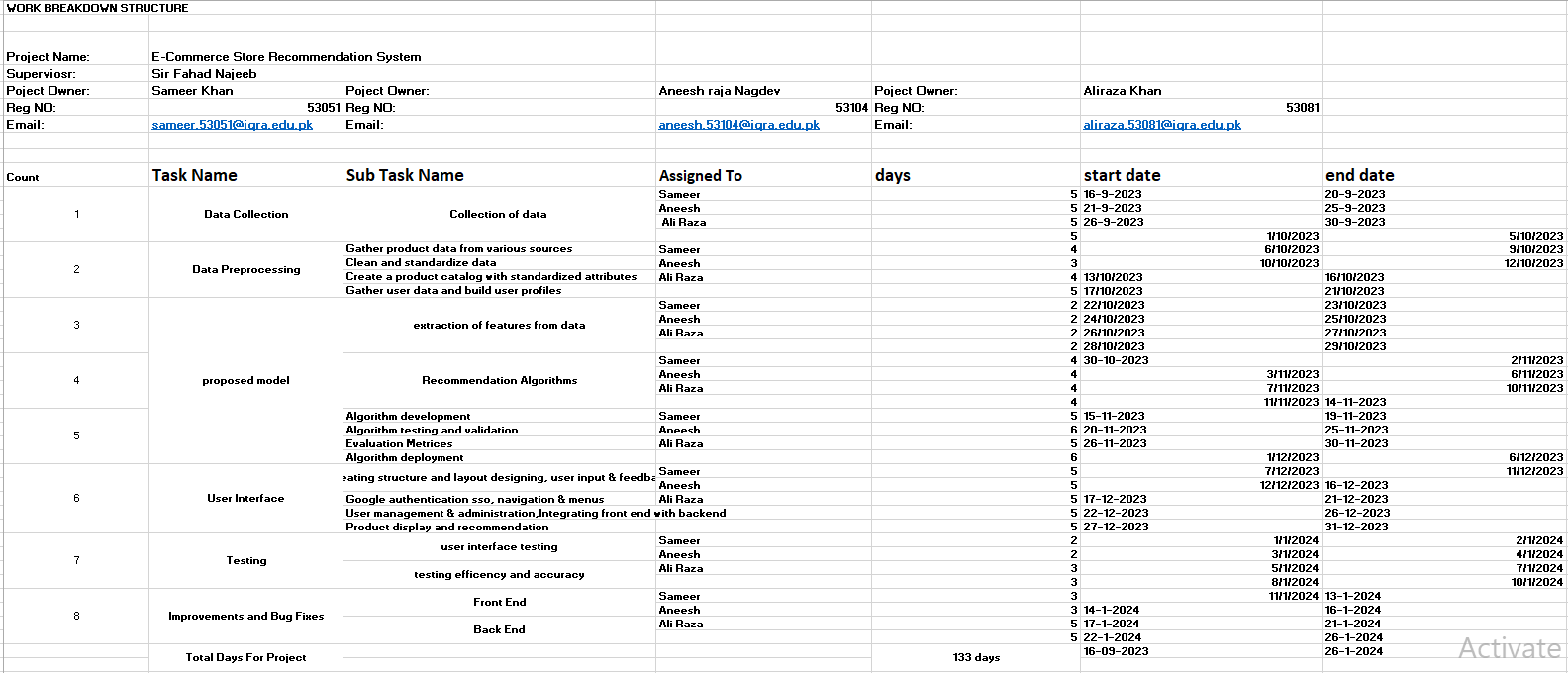
Django

**DATASET:**

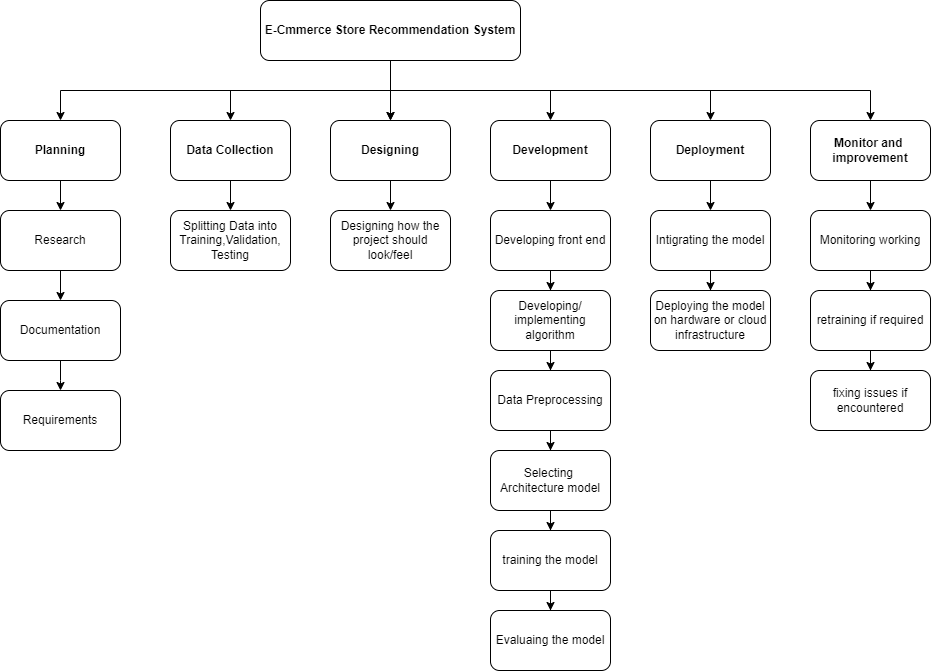
DARAZ (ALI BABA) DATASET

GANTT CHART:

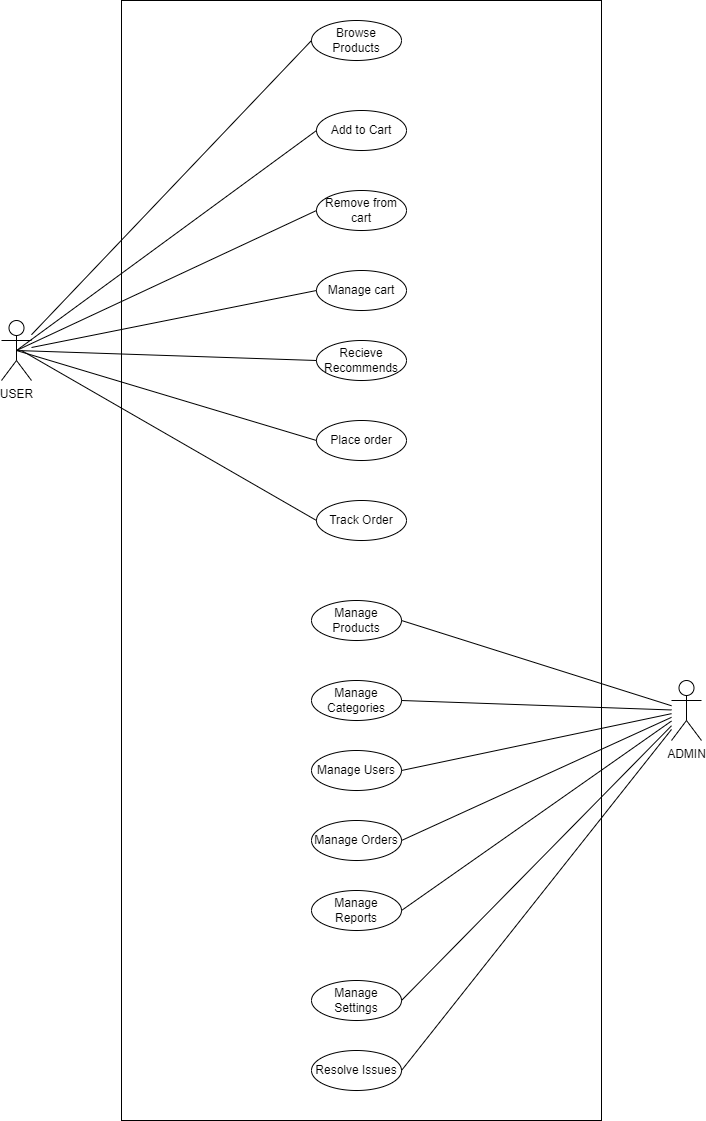




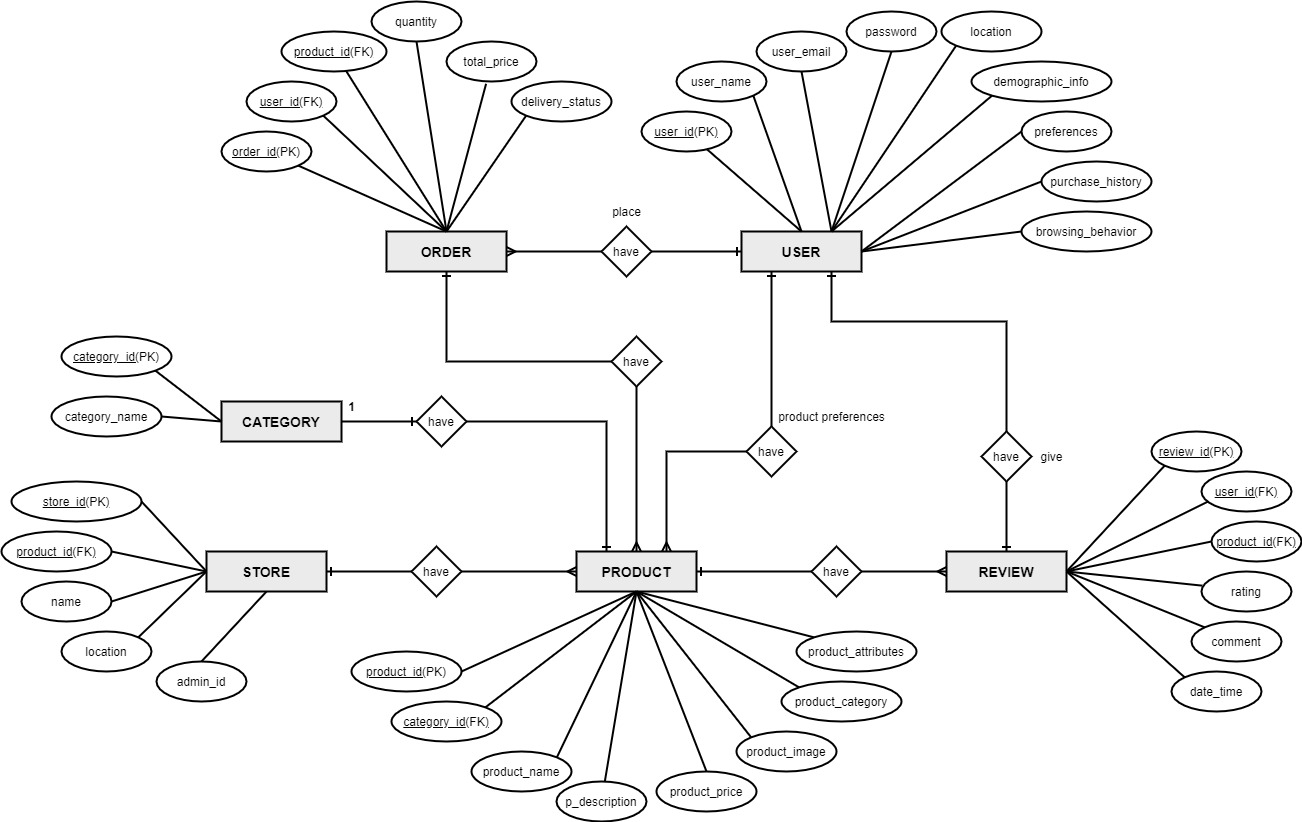
WBS of E-Commerce Store Recommendation System



**USECASE DIAGRAM:**



**ERD**:



**Explanation all entities:**

**User**: The User entity represents the individuals who use the e-commerce recommendation system. Users can create profiles that capture their preferences, purchase history, and browsing behaviour. The attributes of the User entity may include User ID, Name, Email Address, Password, Location, Demographic Information, Preferences, Purchase History, and Browsing Behaviour.

**Product**: The Product entity represents the items available for purchase on the e-commerce store. It includes information about the products such as Product ID, Name, Description, Price, Image, Category, and Attributes (e.g., size, colour, material). The Product entity is essential for providing personalized recommendations and managing the product catalogue.

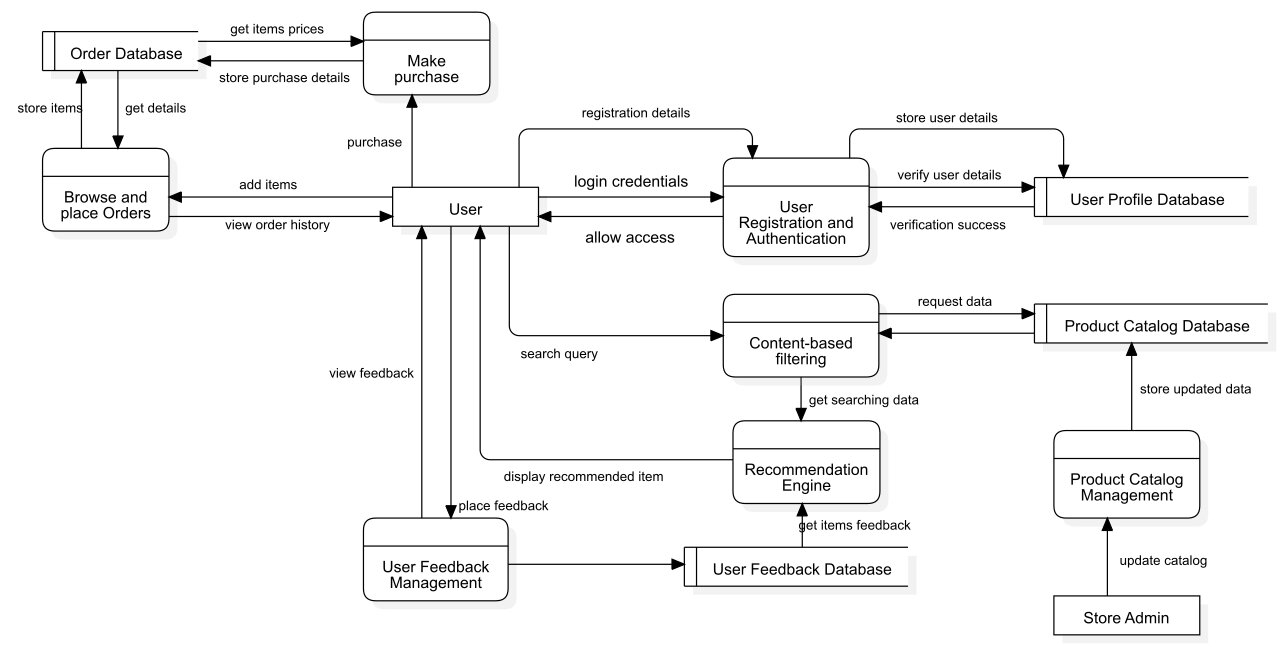
**Order**: The Order entity represents the orders placed by users. It includes attributes such as Order ID, User ID (foreign key), Product ID (foreign key), Quantity, Total Price, and Delivery Status. The Order entity allows users to track their orders and manage their purchase history.

**Review**: The Review entity represents the user reviews and ratings for products. It includes attributes such as Review ID, User ID (foreign key), Product ID (foreign key), Rating, and Comment. Users can rate and review products, which helps other users make informed purchasing decisions.

**Category**: The Category entity represents the different categories to which products belong. It includes attributes like Category ID and Name. Categorizing products allows for better organization and filtering of recommendations based on user preferences and interests.

**Store**: The Store entity represents the e-commerce store itself. It includes attributes such as Store ID, Name, Location, and Admin ID. The Store entity helps in managing the store's products, orders, and other aspects through the admin panel.

**DFD**:



Prototype Design

*This is a prototype Design that explains how our E-Commerce store recommendation system will work!*

**Analysis on Design of UI**  
**Screen 1 & 2. Create Account/Login Screen:**

- This screen allows users to log into the system using their credentials.

- It typically includes fields for entering the username and password.

- Users can also have an option for password recovery or creating a new account if applicable.

**Screen 3.**

- The categories have been made easy for the user. If a user wants to purchase a product, they can go directly to the categories and easily order the product from there.

**Screen 4 & 5.**

- This screen shows that the user is viewing shirts from the men's fashion category after navigating to **the categories section.**

**Screen 6.**

- This screen shows how the algorithm worked based on the user's past behavior and preferences to Recommend red shirts, and that the user is placing their order.

**Screen 7.**

- This screen shows that another user has visited the women's fashion category.

**Screen 8.**

- This screen shows that the user is interested in female sports running shoes.

**Screen 9.**

- This screen shows that the algorithm recommended shoes to the user based on the information provided, and the user has placed an order.

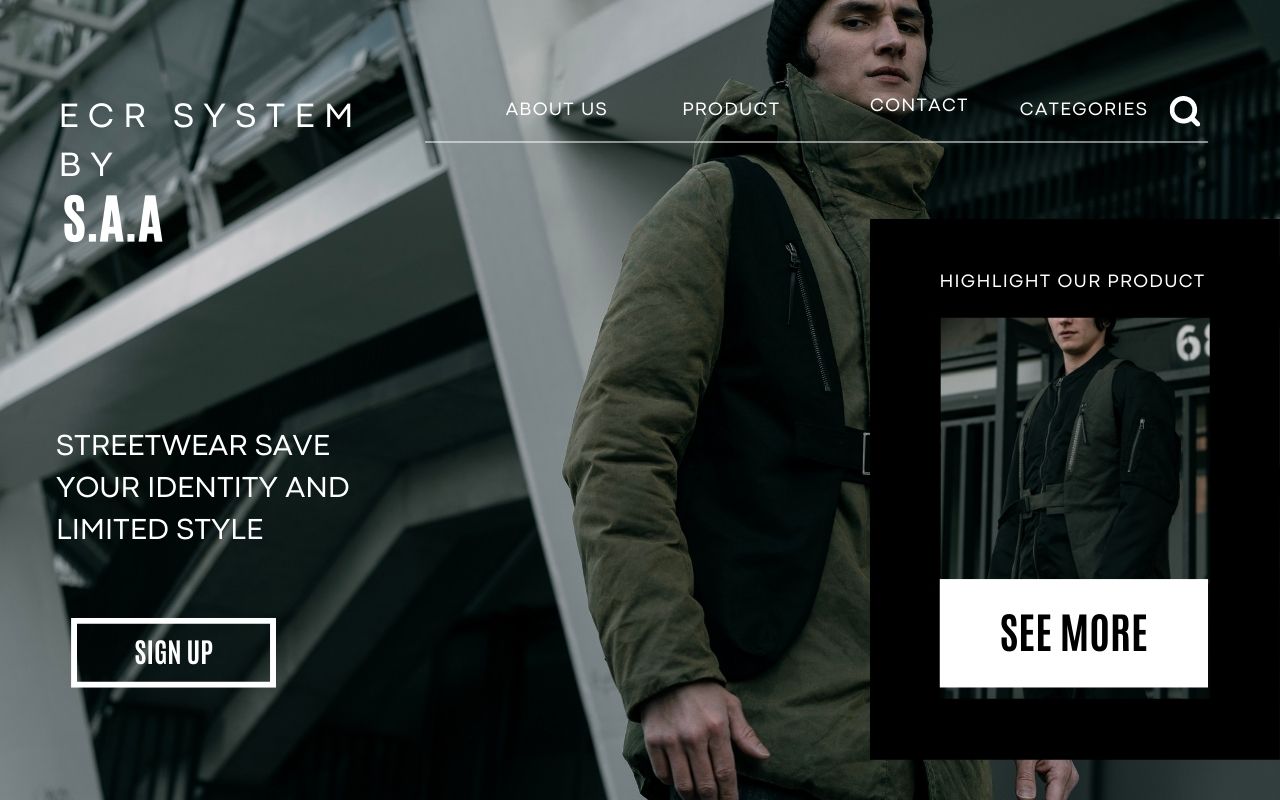
**Screen 10.**

- This screen shows the total price and requires the user to fill in their details, including the location where they want to receive their order, and the payment method. The user is given the option to either pay cash on delivery or make an online payment.

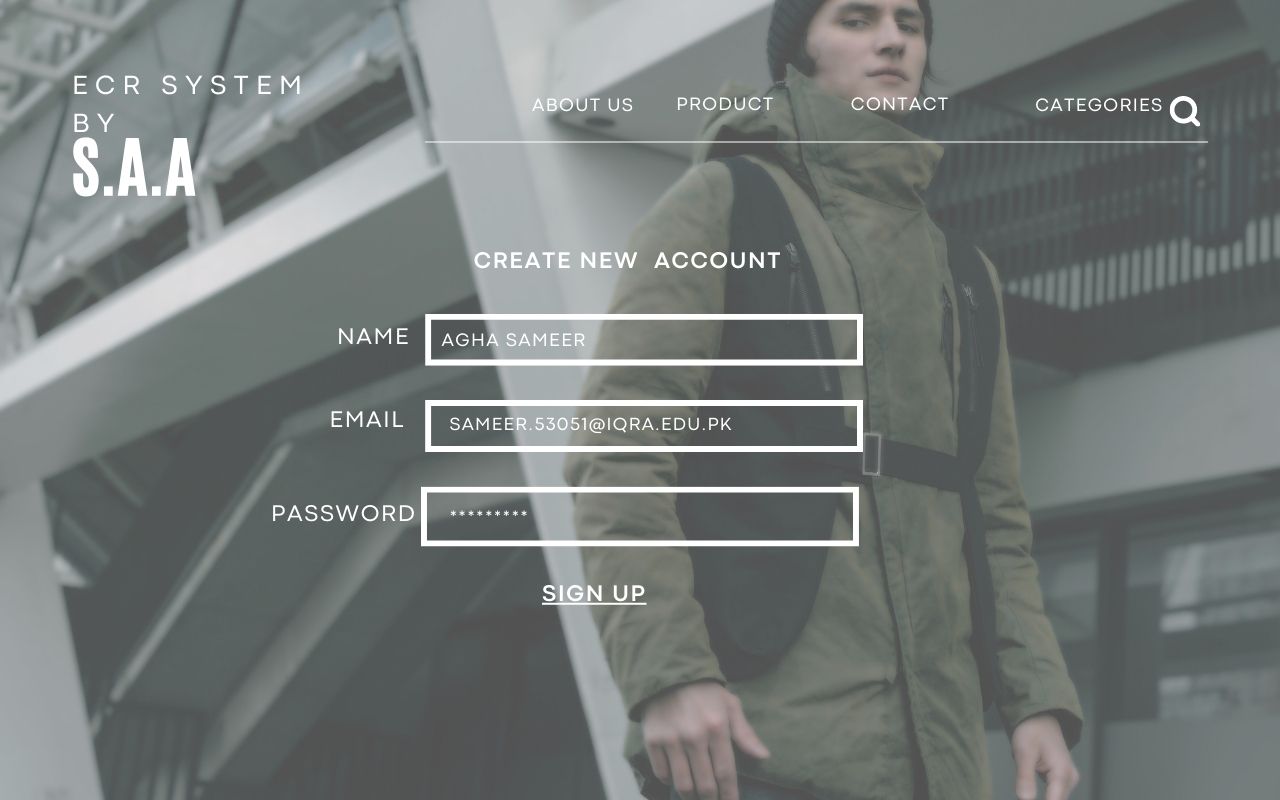
**Screen 11.**

- This screen shows that the user is being asked to provide ratings and reviews, which will help the admin identify any problems and understand the user's preferences.

Screen 1:



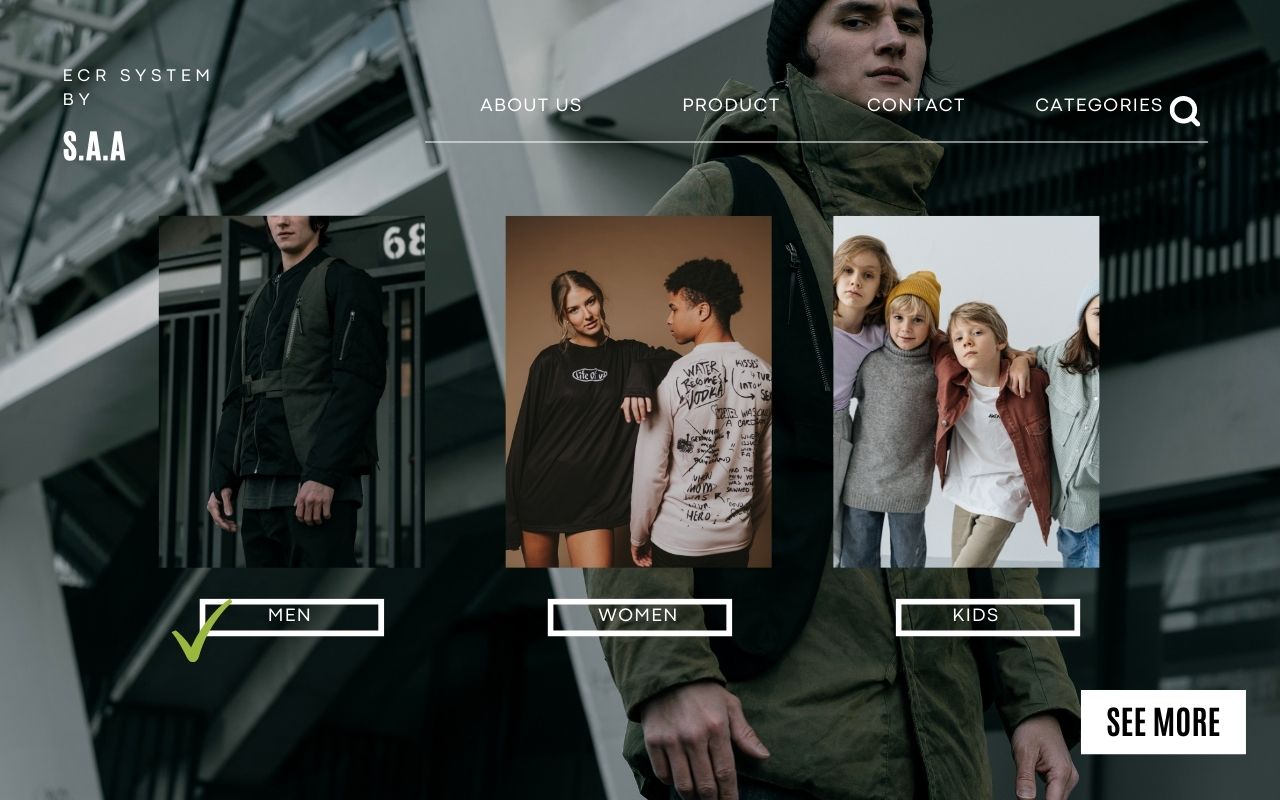
Screen 2:



Screen 3:



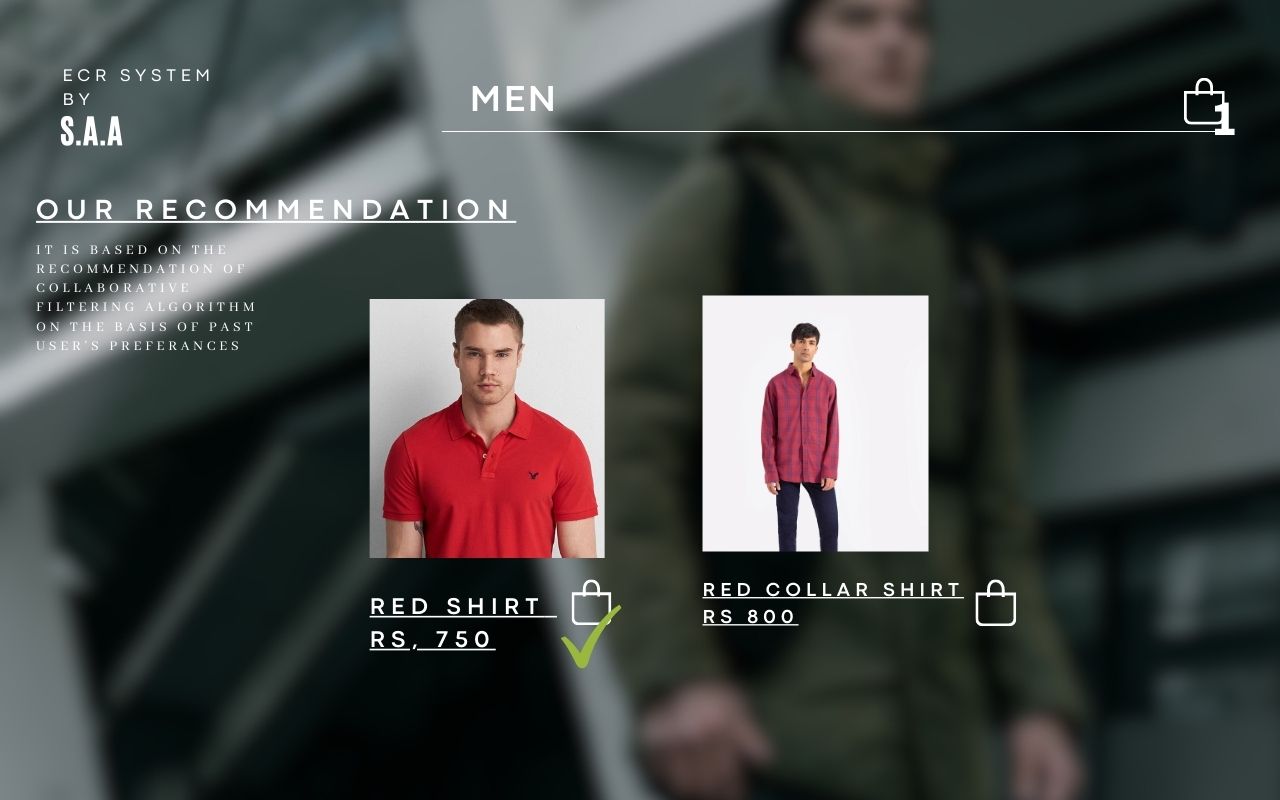
Screen 4:



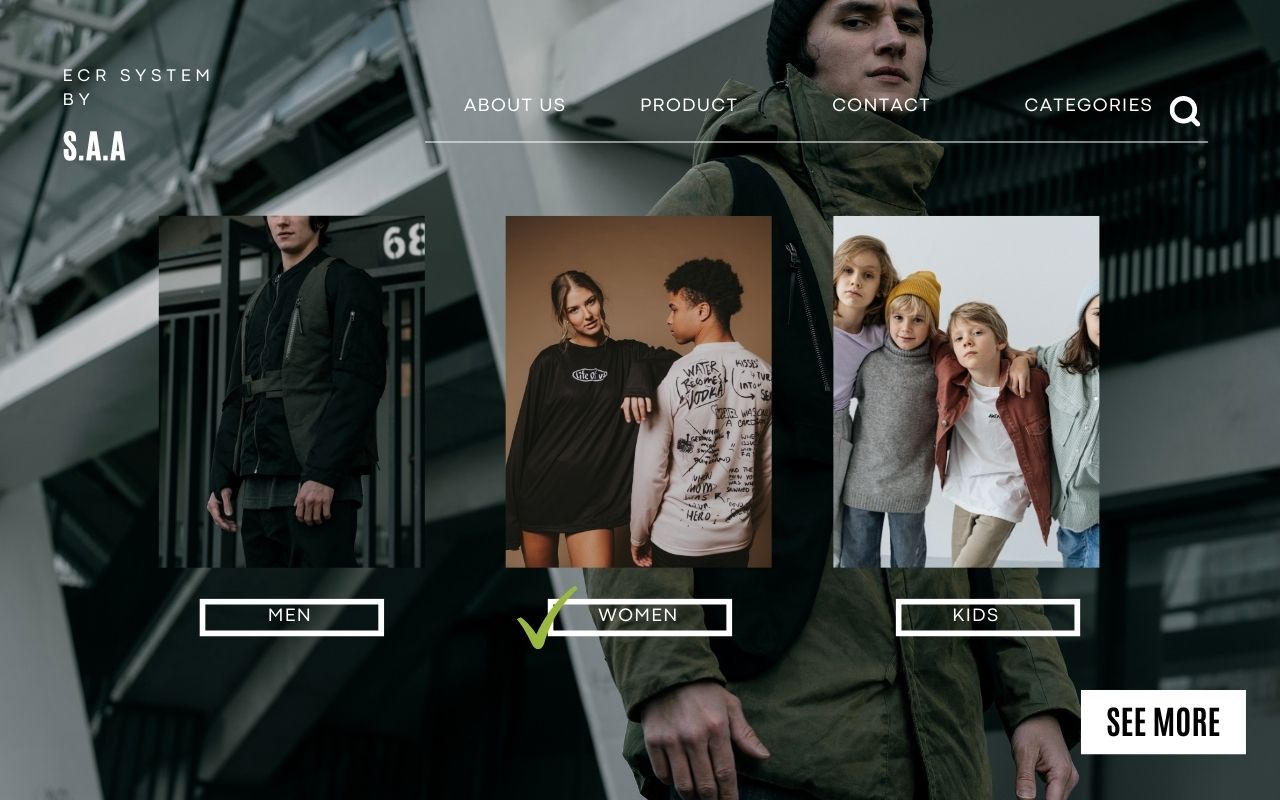
Screen 5:



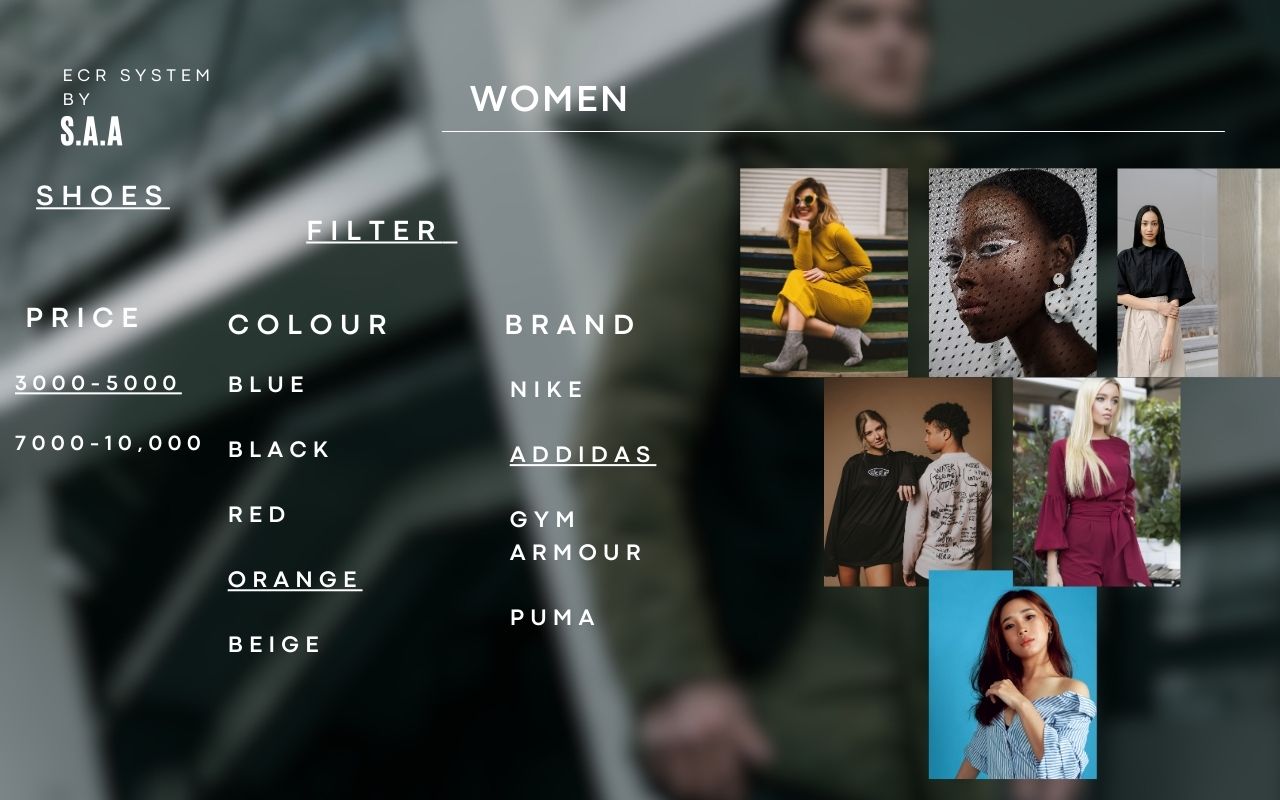
Screen 6:



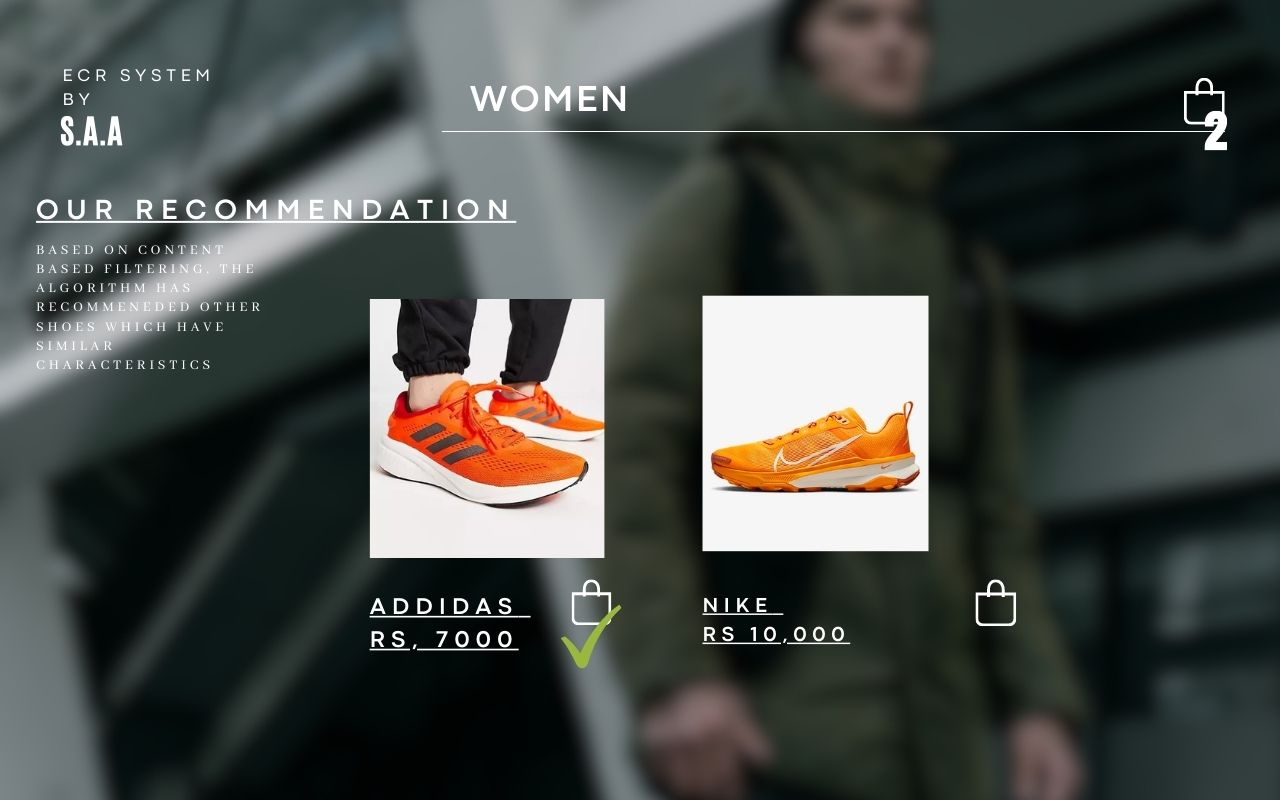
Screen 7:



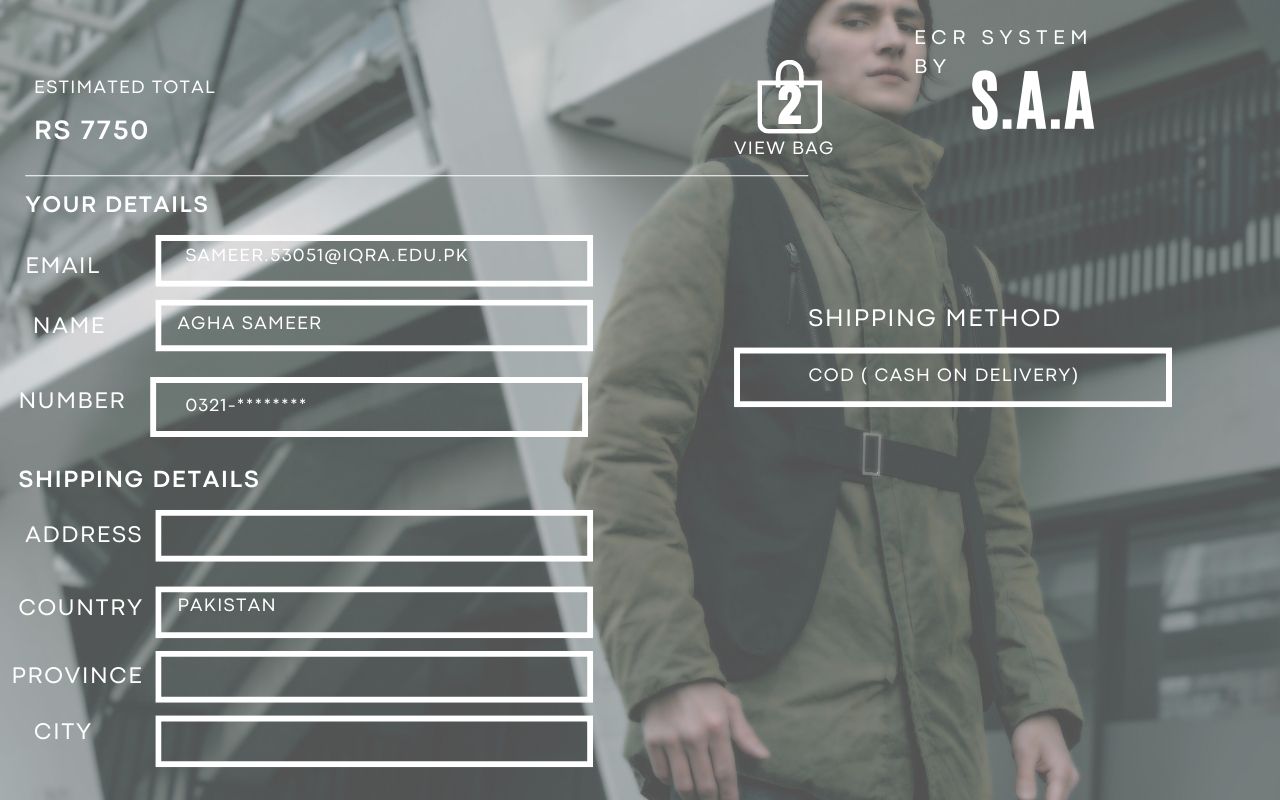
Screen 8:



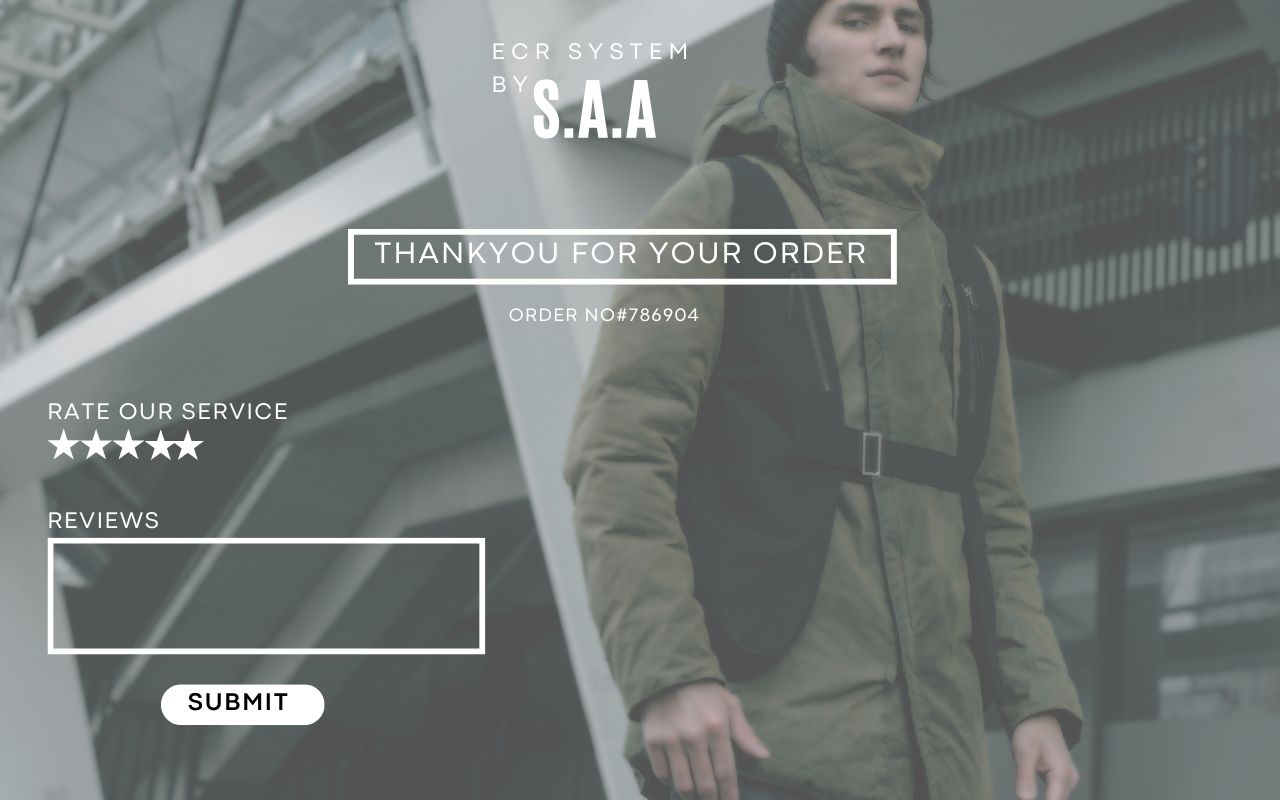
Screen 9:



Screen 10:



Screen 11:



THANKYOU