National University of Sciences & Technology (NUST)

School of Electrical Engineering and Computer Science (SEECS) Department of Electrical Engineering

Final Year Project

Project Report

For

ArtiZen: An E-Commerce-Based Application for Cultural Artifacts of Pakistan

Team Members:

Hammad Mukhtar	[282678]
Hamna Saeed	[295150]
Mehyar Ali	[288484]

Advisory Committee:

Dr. Adnan Rashid
Mr. Maajid Maqbool

Date of preparation:

10 – 05 – 2023

Table of Contents:

List of Figures:

List of Tables:

Revision History:

Name	Date	Reason for Changes	Version
Hammad Hamna Mehyar	10-05-2023	N/A	1.0

1. Introduction:

1.1. Introduction:

Pakistan is a developing country where arts, culture, and other forms of expression are rapidly gaining traction. Pakistan has evolved into a place where Pakistanis are making space for themselves and their ideas. Pakistan is rich in culture but has no marketplace for the buying and selling of artifacts. Our project ArtiZen is basically an online platform that connects skilled craftsmen with buyers. We intend to make it easier for the craftsmen to sell their product easily and similarly provide buyers with a convenient platform to find quality crafts. This project will not only facilitate craftsmen with employment opportunities but also help the natives and tourists from all over the world to buy good quality artifacts at their ease.

To actualize this idea, we will be developing a mobile application. To accomplish this project, we will connect with local artisans living in far flung and remote areas of Pakistan. The craftsmen will be able to elevate their standard of living by earning at the ease of their own places.

1.2. Problem Statement:

Pakistan has a lot of artifacts that represent its culture but unfortunately there is no convenient market place to buy or sell these things. A few online stores that offer these kinds of products deliver poor quality products which results in customer dissatisfaction. Moreover, the artisans making these kinds of products face a downturn due to the unavailability of market places for these crafts.

1.3. Proposed Solution:

This online platform “ArtiZen” will help to promote and sustain Pakistani culture. It will not only provide artisans with the opportunity of employment but also facilitate the people of Pakistan and tourists from the entire world to buy best quality Pakistani artifacts at the ease of their homes.

1.4. Literature Review:

We have come across some online platforms such as Vceela[1], which was originally meant to sell these artifacts by employing credible artisans but now they are also supporting small and home based businesses, discarding their original purpose. Some other platforms that we came across that were selling these artifacts including Daraz[2], did not ensure high quality of the product making them an ill-suited choice to buy Pakistani crafts. One of the international platforms offering this type of services for artifacts from around the world is Ten Thousand Villages [3].

1.5. Development methodology:

There are multiple methodologies for developing software and applications. In our project, we will implement the **Agile methodology**. It helps to divide the project into sub-units, which is very helpful in testing the overall product. At the start of project development, it's difficult to completely define the design and requirements of the software. With time, requirements change and it's difficult to do development again. But using Agile Methodology, we are flexible with that. We will complete the project by its functionalities and requirements. Once a function is completely designed and verified, then we go on to the next one.

1.6. Tech Stack:

Thousands of tools are available for making different projects in different scenarios. Since, we will be developing both, an android application and a web application. Therefore,

❖ **For web admin panel**, we will be using MERN Stack. It helps to build a more attractive front-end along with the backend. And data storing and retrieving are very quick and reliable. Making different REST APIs and integrating them is also done without third-party libraries. With the help of MERN Stack, we can build a high-quality web application and enhance product performance and scalability.

❖ **For the android application**, we will imply React Native. It is very helpful for designing a smooth and responsive User-Interface. The working of the app is very fast with react native. It provides the functionality of quick live data updating which increases customer satisfaction.

1.7.Intended Audience and Reading Suggestions:

This document is intended for a specific audience with specific interests as explained below. Some audiences required prerequisites to read and understand this document.

Audience	Report Content	Sequence for Reading	Usage
Customers	• Description • System Interfaces • System Features • Functional Requirements	• Introduction • Purpose • Intended Audience • Product Functions • System Features	When the user wants to buy or check any product
Developers		• Introduction • Purpose • Scope • Intended Audience • User Classes • Operating Env.	When they want to do changings to the code

	- Non-Functional Requirements	- Product Functions - External Interfaces - System Features	
Testers		- Introduction - Intended Audience - Product Functions - System Features - Requirements	When testing of specific features or the whole product is required
Admin		- Product Scope - System Features - Requirements - Business Rules	Monthly or when get notified about any internal issue.
Sellers		- Introduction - Product Scope - Intended Audience - External Interfaces - System Features - Business Rules	Daily to see orders
Managers		- Introduction - Product Scope - Intended Audience - System Features - Requirements	For weekly or monthly report generation.

Table 1: Intended Audience and Reading Suggestion

1.8. Product Scope:

This product will help to promote and sustain Pakistani culture. People will be able to buy the best quality cultural artifacts from authentic artisans with the ease of one click. It will not only provide artisans with the opportunity of employment but also facilitate the people of Pakistan and tourists from the entire world to buy the best quality Pakistani artifacts at the ease of their homes.

2. Overall Description:

2.1. Product Perspective:

The main concept behind ArtiZen is to provide a such platform that will promote and enhance Pakistani culture. Along with providing employment opportunities to artisans, this product will also facilitate Pakistani people and foreign tourists. Artisans from different areas will create their stores and start selling precious artifacts and promoting Pakistani culture. This product contains three major users. **Customers** will buy products from **Seller** and Seller which are artisans who will create their stores and sell products. While **Admin** is the one who will manage the whole system and will keep an eye on the smoothness of the system and communication between Artisans and Customers.

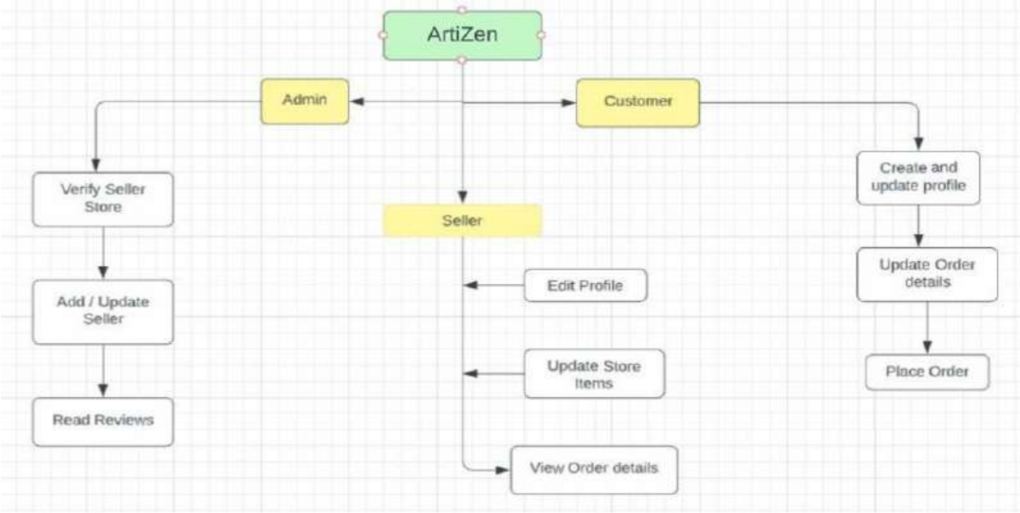


Figure 1: Overall System Diagram

2.2. Product Functions:

The following are major functions that will be performed by our product after deploying:

For Customers:

- Customers don't need to view products by visiting places on-site. They can see all products by just visiting our application and browsing products virtually.
- Secured user's account creation and profile modification feature.
- See reviews of other customers before buying any product.
- Easy to add and remove items from the Cart for customers.
- Secure Online Payment method and Cash on Delivery option.
- Tracking their Order activities.
- Trusted Refund Policies.
- A "Contact Us" feature where customers can ask their queries.

For Artisans:

- Secure Account creation and availability of modification function.

- Providing Urdu language Integration that will assist artisans to use this platform to earn.
- Employment Opportunities by living in your place.
- Chances of expanding their business by creating a store on our application.

2.3. User Classes and Characteristics:

Typically, anyone who is familiar with the basic use of the Internet and mobile applications can access and use our application.
Following are the main users and their characteristics of our product:

Administrators:

- These users must have experience in managing online shopping platforms.
- Must be able to work in highly integrated teams.
- Must know e-commerce jargon such as products, cart, database, dashboard, etc.

Seller:

- This product is a dedicated platform for Pakistani cultural crafts. So, only those products will be listed on the website and application that is a Pakistani cultural artifact made by local artisans.
- The seller must be able to communicate with the customers and address their queries if any.
- The seller must be familiar with basic online shopping terminologies.

This platform is intended to create earning opportunities for local artisans, so it will be built in such a way that will create maximum ease for the artisans such as tutorials to get started with an online store and Urdu language integration.

Customer:

- Must have information about shopping Mall.
- Must know about products the user wants to see or buy.
- Must be familiar with different payment methods (online bank transfer and cash on delivery).
- Must be able to read, understand and interpret English or Urdu language.

2.4. Operating Environment:

ArtiZen products will be operated in the following environment:

- Windows Operating System

- Android Phones
- **Version**: All versions.
- **Database**: Mongo DB

ArtiZen is a web-based admin panel application and android mobile application that will be available on the world wide web and google play store as well. Therefore, operating environments in the case of **Web-based applications** are:

- Google Chrome Browser
- Safari Browser
- MS Internet Explorer
- Mozilla Firefox
- Opera with v 6.0 or higher.

While operating environments for **mobile applications** are:

- Google Play Store for android phones.

The hardware configuration includes Monitor, Basic keyboard, Hard Disk, and Mouse.

Note: Most important requirement to use this product is the presence of **internet connectivity**.

3. External Interface Requirements:

3.1. User Interfaces:

Different users have different interfaces to see the content of the applications. Admin users can create, read, update, and delete different products. While users can view only information from the application. The interface of different pages will look like this:

4. System Features:

Use Case Diagram:

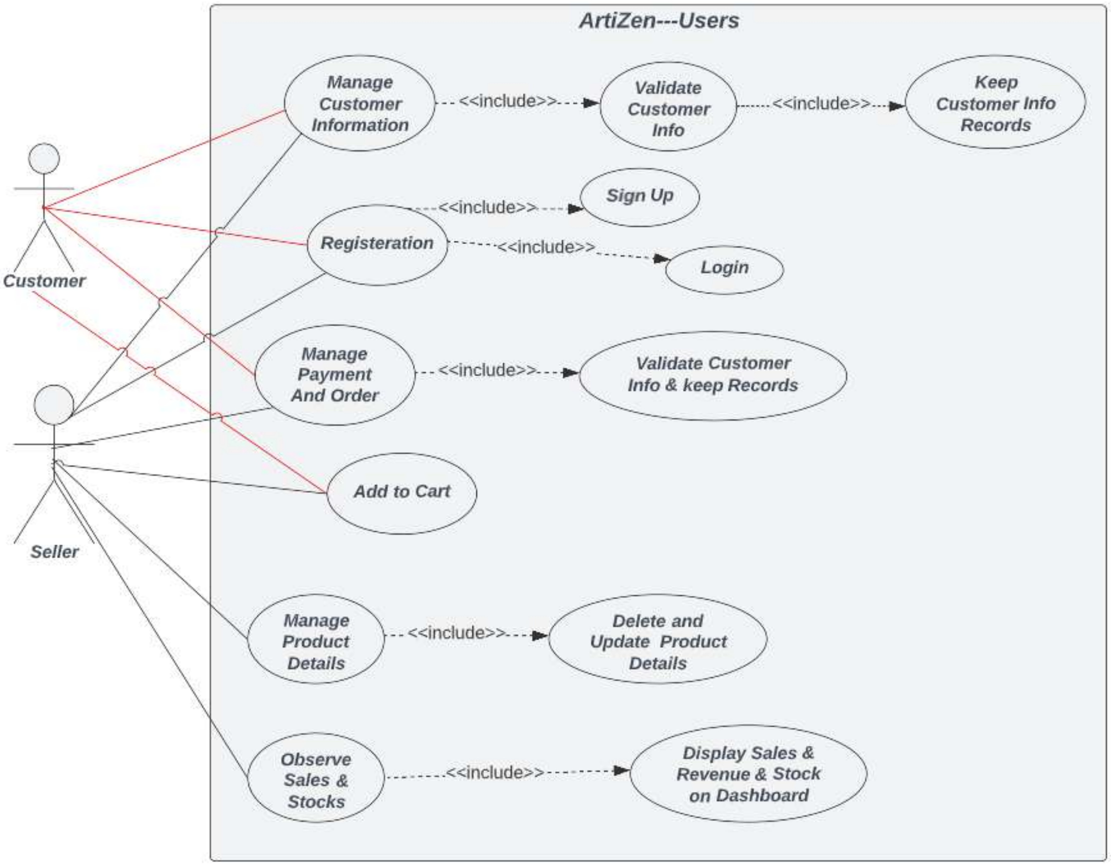


Figure 3: Use Case Diagram of ArtiZen Users

4.1. User Accounts:

4.1.1. Description: Most important feature of the product. Creation of an account based on personal details which will be used to access website data in the future. To do online transactions between sellers and customers over products, both users must have accounts in the system. Without accounts existence, users can only browse products and pages but don't have permission to do any purchasing. Customer accounts will hold information specifically about the name, email, password, phone number, and address. Users of this system can view their account details, edit those details and delete them as well. On the other side, Seller accounts hold information about their store including name, product details, contact number, address, and revenue.

Priority Level: High

4.1.2. Stimulus/Responses Sequences:

User Accounts

Stimulus	Response
Click on the signup button and write credentials to proceed further.	A user account will be created after entering the required credentials.
Click on the Login button and write credentials to access application features.	After matching credentials in the database, upon success, the user will be redirected to the home page with access to all features.
Click on the edit button and do some changes accordingly and click on the save button.	Changes will be saved in the database if done according to specified rules and conditions.

Table 4: Event-Response Table of User Accounts

Sequence Diagram:

a. Sign Up Module

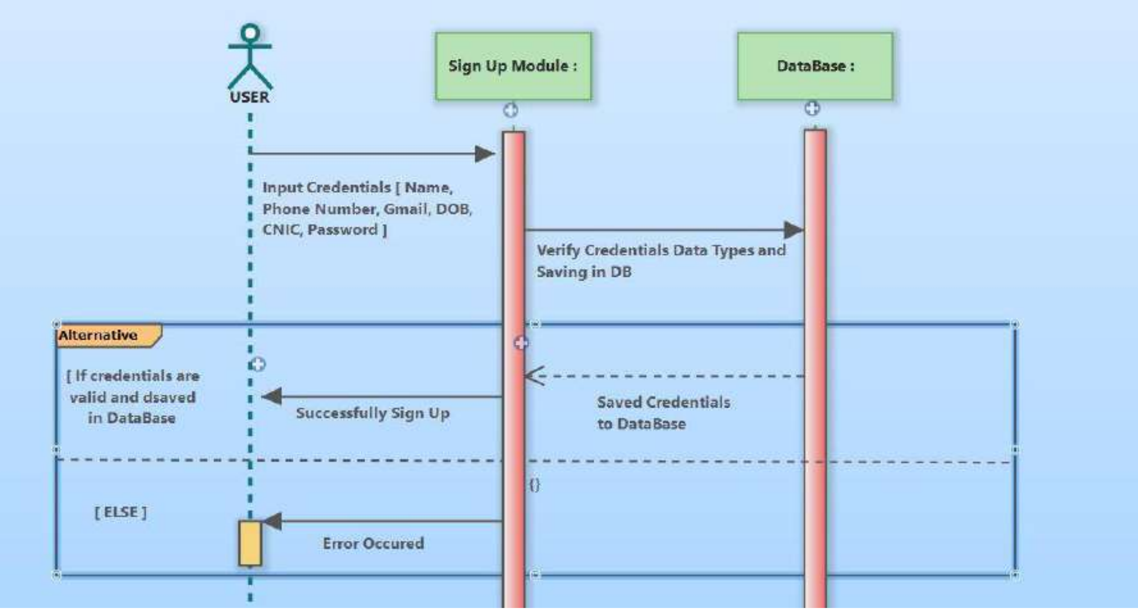

USER
Sign Up Module :
DataBase :
Input Credentials [Name, Phone Number, Gmail, DOB, CNIC, Password]
Verify Credentials Data Types and Saving in DB
Alternative
[If credentials are valid and dsaved in DataBase
Successfully Sign Up
Saved Credentials to DataBase
[ELSE]
Error Occured

Figure 4: Sequence Diagram of Sign Up Module

b. Login Module

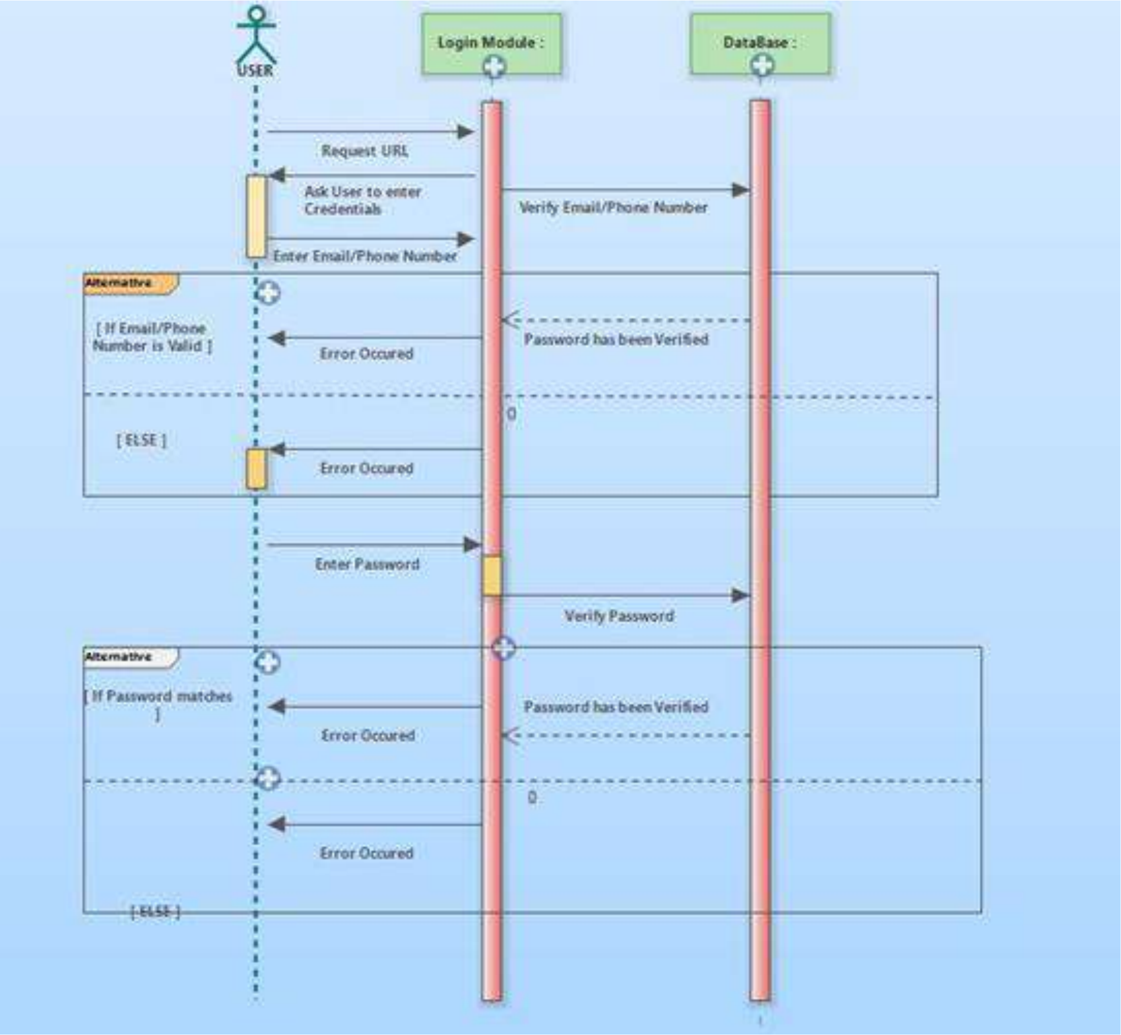


Figure 5: Sequence Diagram of Login Module

4.2.Dashboard:

4.2.1. Description:

4.2.1. Description: An important feature of the product. Different users have a dashboard with different features. In the case of the Customer, his access only to view account details and orders he has placed. While for Seller, he can view account details, store details, and product details. The revenue generated from his business and its details will also be shown in the dashboard. In the case of the Admin, he has full access to all features of the system. He can see details of sellers, customers, products, and stores. Moreover, information regarding revenue generated by different stores is also accessible to the admin.

Priority Level: High

4.2.2. Stimulus/Responses Sequences:

Dashboard	
Stimulus	**Response**
Users (Seller, Customer, Admin) will be on the dashboard button on the home page.	The system will redirect the user to the dashboard with all features.
Click on any feature and then click the edit button and do some changes accordingly and click on the save button.	Changes will be saved and updated in the database if done according to specified rules and conditions.

Table 5: Event-Response Table of Dashboard

Sequence Diagram:

a. Dashboard Module

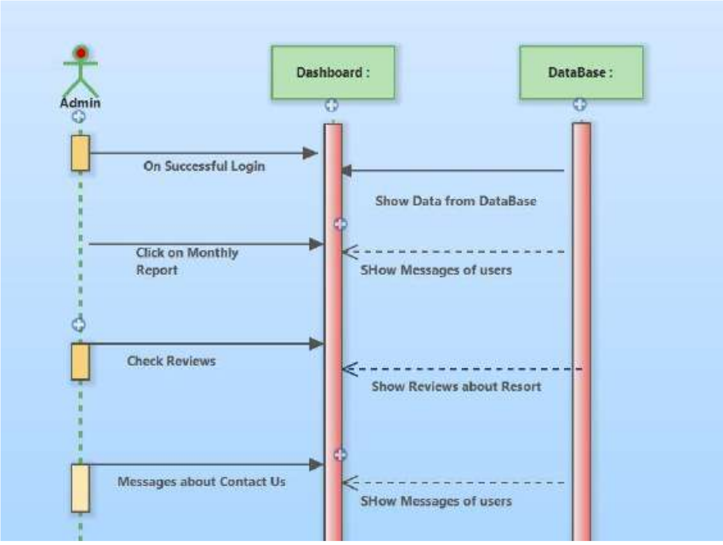


Figure 6: Sequence Diagram of Dashboard Module

4.3. Search Engine:

4.3.1. Description: There are multiple orders on a single page with multiple categories. Customer wants to search for a specific item from a large catalog of products in a shopping application. Users can search for a specific product by entering a keyword or the complete name of the product. A drop-down menu will be shown with suggested words to search for products.

Priority Level: High

4.3.2. Stimulus/Responses Sequences:

Search Engine

Stimulus	Response
Type keywords or names of products to search for the specific item in the search box.	Immediately, a drop-down menu will be displayed with suggestion keywords.
Click on the keyword user wants to view.	The system will show the required product and relevant products to the user.

Table 6: Event-Response Table of Search Engine

Sequence Diagram:

a. Search Engine Module

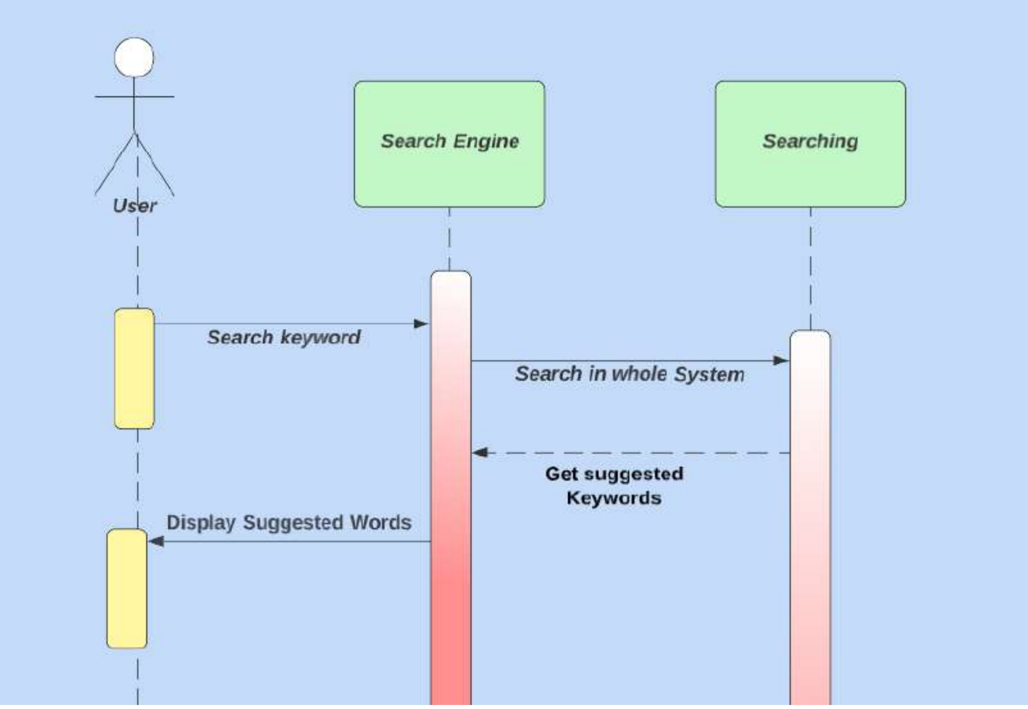


Figure 7: Sequence Diagram of Search Engine Module

4.4.Wish list Items:

Sequence Diagram:

a. Wish List Module

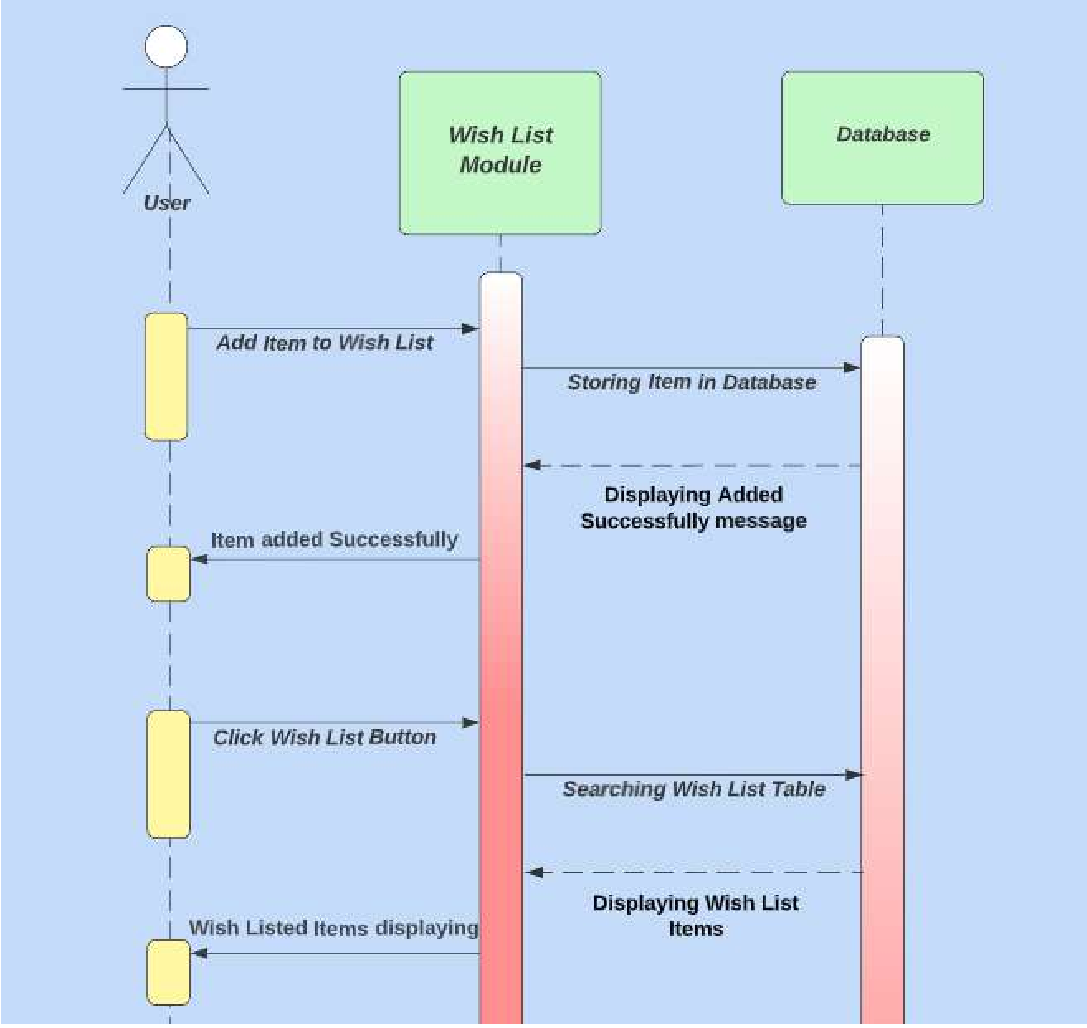


Figure 8: Sequence Diagram of Wish List Module

4.4.1. Description: Users can see all orders and if there are few orders, users want to buy but not at that time. The user aims to buy after a few days. He wants to save that product to save time for searching again, he can simply add that product to wish lists for future use and save time.

Priority Level: Medium

4.4.2. Stimulus/Responses Sequences:

Wish list Items	
Stimulus	**Response**

Click on the wish list button, shown beside the item's details.	Add that item to the wish list.
Click on the wish list button on the home page to view all wish-listed items.	The platform will show the page where all wish-listed items are stored.

Table 7: Event-Response Table of Wish List Items

4.5.Shopping Cart:

Sequence Diagram:

a. Shopping Cart

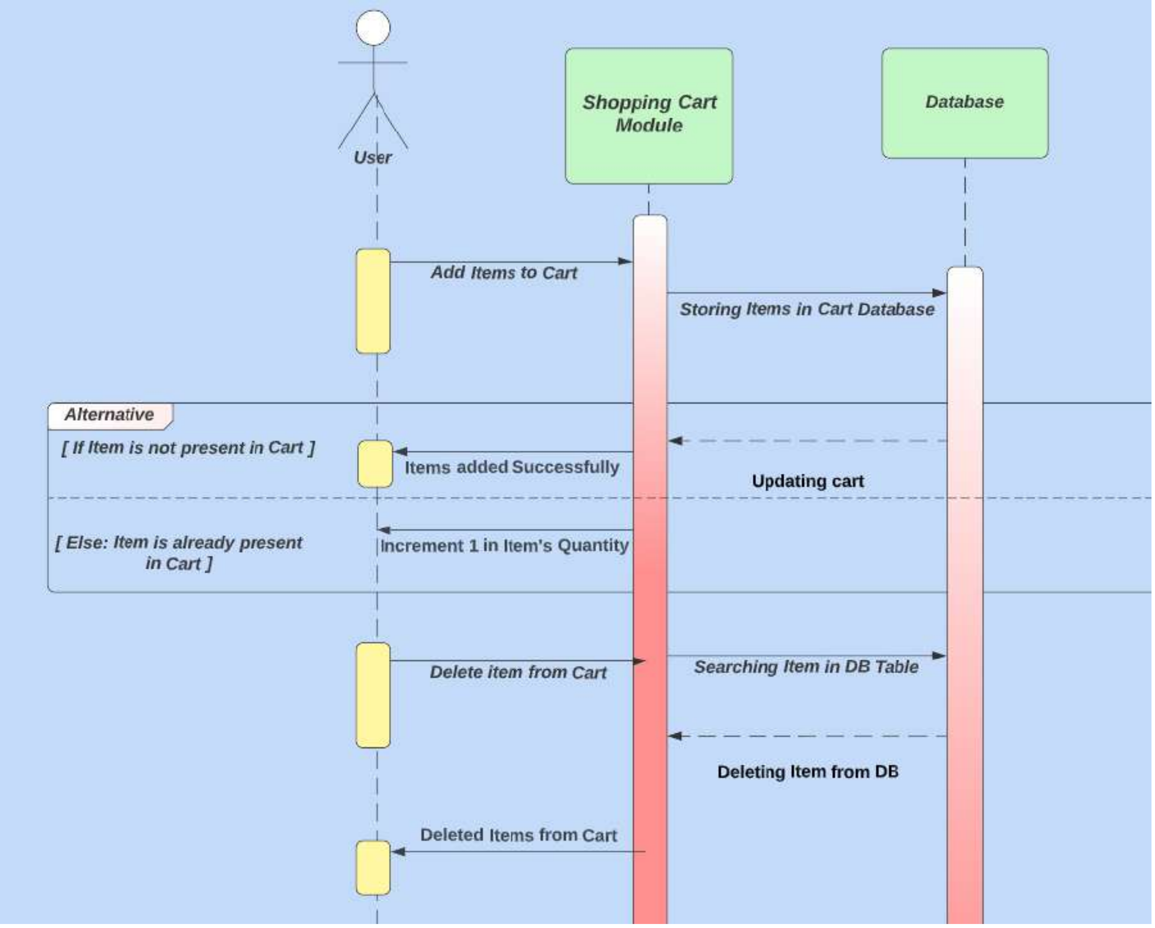


Figure 9: Sequence Diagram of Shopping Cart Module

4.5.1. Description: A shopping cart is considered to be an important feature of shopping applications. After viewing all products, now the customer will select specific items and will want to purchase them. All the items the customer intended to buy will be added to the shopping cart. One shopping cart is associated with one user. After adding items to the shopping cart, the customer can again browse other items and can add more items to the shopping cart and purchase them with one click.

Priority Level: High

4.5.2. Stimulus/Responses Sequences:

Shopping Cart	
Stimulus	**Response**
Click on add to cart button on the product page and continue browsing other items.	Items will be added to the shopping cart list.
Click on the View Shopping cart button to see a list of all items added to the cart.	A list of items will be displayed with individual item names, quantities, and prices.
Click on the plus or minus button to change the quantity of the items.	The number of items will be increased upon clicking plus button and decreased upon clicking the minus button.
Click on the delete button to remove items from the shopping cart.	Item will be removed from the shopping card list.
Click on check out to proceed next step to place an order.	A page with detail of items' names, prices, quantity, and the total bill will be displayed.

Table 8: Event-Response Table of Shopping Cart

4.6.Payment Feature:

4.6.1. Description: After the checkout order, the next step is to choose a payment method to purchase items. After clicking on proceed to payment button, the user will be redirected to the payment page where all details of buying items, prices, quantity, home address, and delivery charges will be included. To proceed payment step, the system will ask the user to enter details of the selected payment method. Next is to click confirm button to place an order.

Priority Level: High

4.6.2. Stimulus/Responses Sequences:

Payment Feature	
Stimulus	**Response**
Click on proceed to buy items on the Shopping Cart page	A payment page with complete order details will be displayed.
Enter complete address details to further proceed with the order.	The address will be recorded for the current order.
Choose a payment method to do payment of the order. Enter complete details of payment method.	After verifying payment details, OTP will be sent to the user via email or phone number.
Enter the correct OTP for completing the payment process.	On successful payment, A page with “Order Placed successfully. Continue Shopping” will be displayed.
Click on continue shopping.	The user will be redirected to the home page.

Table 9: Event-Response Table of Payment Feature

Sequence Diagram:

a. Payment Module

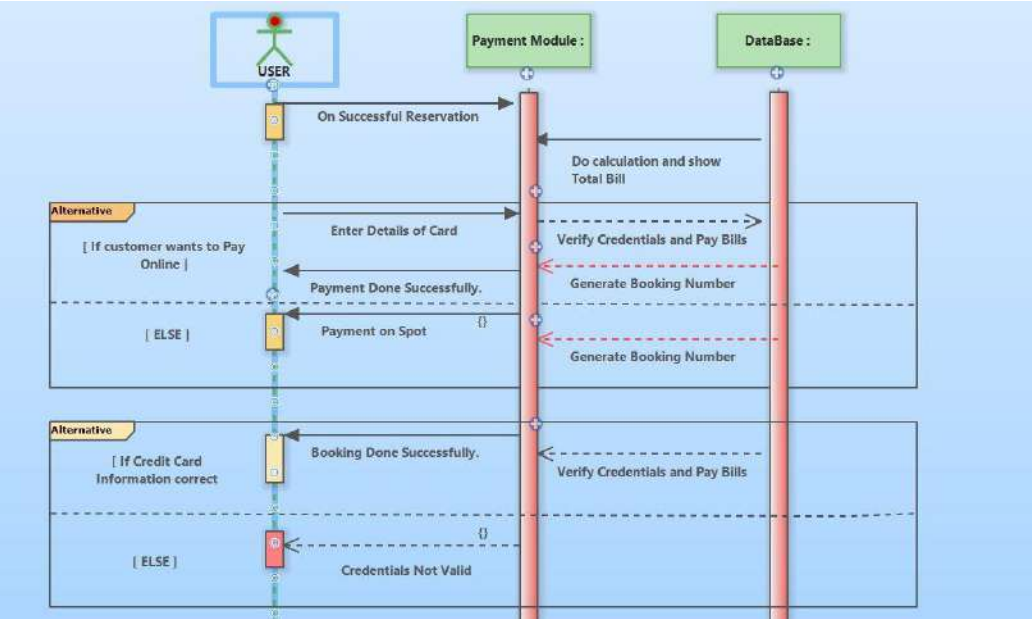

USER
Payment Module :
DataBase :
On Successful Reservation
Do calculation and show Total Bill
Alternative
[If customer wants to Pay Online]
Enter Details of Card
Verify Credentials and Pay Bills
Payment Done Successfully.
Generate Booking Number
[ELSE]
Payment on Spot
{}
Generate Booking Number
Alternative
[If Credit Card Information correct
Booking Done Successfully.
Verify Credentials and Pay Bills
[ELSE]
{}
Credentials Not Valid

Figure 10: Sequence Diagram of Payment Module

4.7. Seller Features:

4.7.1. Description: For the customer to view and buy items, there must be stores with different types and categories of items. The stores will be created by the seller. Seller will sell their products using our application and the process to upload product images will be very streamlined. Just enter the list of items and let the system redirect them on the home page. A notification will be received when the customer placed the order. Revenue generated by the seller will also be visible to them.

Priority Level: High

4.7.2. Stimulus/Responses Sequences:

Seller Features	
Stimulus	**Response**
Click on the upload items button and enter the detail of the items (including name, quantity, size, color, pictures, videos, brand), and press the save button.	The list of items the seller desires to upload will be stored in the database and will be shown on the home page.

Click on view orders to see the detail of all orders placed by the customer.	A list of placed orders will be shown with customer details (including the order number, items, name, address, and payment status).
To choose a delivery method, click on the self-delivery option to provide the order to the customer by hand and click on delivery through the external source to deliver orders by hiring another person.	The delivery method will be finalized and the process of delivering the order will be initiated.

Table 11: Event-Response Table of Seller Features

Sequence Diagram:

a. Seller Features

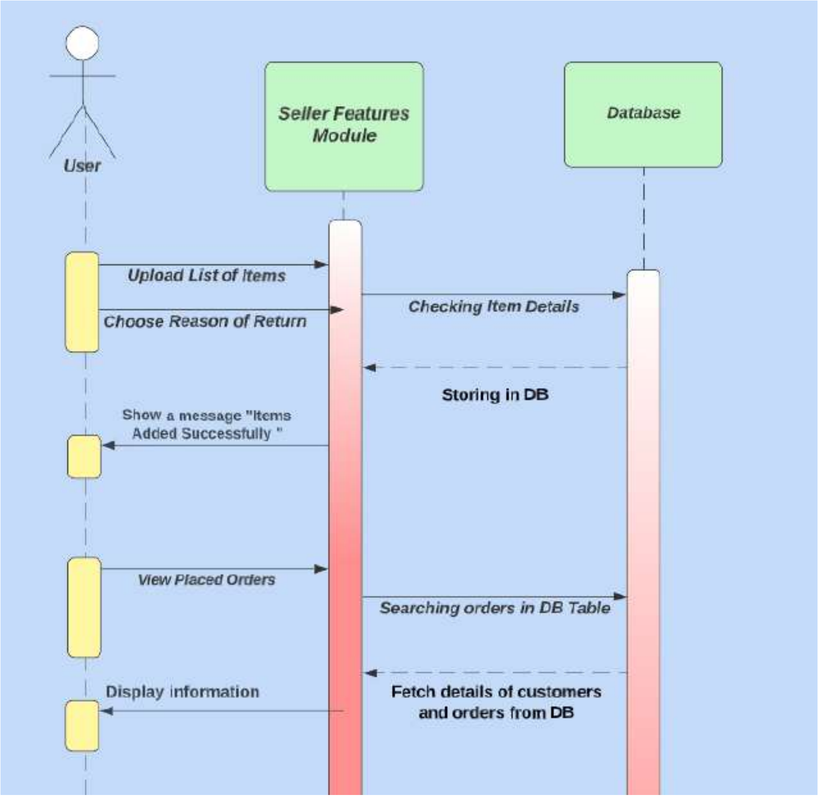


Figure 12: Sequence Diagram of Seller Module

Review Classification Model:

We have successfully implemented a review classification model using sentence transformer. The model leverages the power of transformer architectures, such as BERT, to encode and embed textual data, allowing for effective classification of reviews based on sentiment or predefined categories.

To accomplish this, we followed several key steps:

Dataset Preparation: We gathered a labeled dataset of reviews, ensuring it covers a wide range of sentiments or categories relevant to the classification task. The dataset was diverse and representative of the target application.

Text Preprocessing: We performed thorough data cleaning and preprocessing on the review text. This involved removing irrelevant information, punctuation, and stop words. We also normalized the text by converting it to lowercase and addressed any spelling or grammatical errors.

Sentence Embedding: We utilized the sentence transformer model to encode the preprocessed review text into high-dimensional vector representations, known as sentence embeddings. These embeddings capture the semantic and contextual information of the reviews, providing a foundation for accurate classification.

Model Training: We designed the architecture for the review classification model. This typically involved adding additional layers on top of the sentence transformer encoder to perform the desired classification task. For sentiment analysis, we incorporated a dense layer followed by a softmax activation function to predict sentiment labels. For category classification, we designed an architecture that accounted for the complexity of the task, potentially incorporating attention mechanisms or recurrent layers.

Loss Function and Optimization: we defined an appropriate loss function, such as categorical cross-entropy, to measure the discrepancy between the predicted labels and the true labels. To optimize the model, we utilized optimization algorithms like stochastic gradient descent (SGD) or Adam, allowing the model's parameters to be updated and the loss function to be minimized during training.

Training Process: We trained the model using the labeled dataset. This involved feeding the preprocessed review text into the sentence transformer encoder, passing the encoded representations through the added layers, and computing the loss. Through backpropagation and gradient descent, the model's parameters were iteratively updated, continuously improving its performance.

Model Evaluation: We assessed the performance of the trained model on a separate evaluation dataset. Evaluation metrics such as accuracy, precision, recall, F1-score, or area under the ROC curve were used to measure how well the model classified the reviews. This evaluation provided insights into the model's generalization and effectiveness in predicting sentiment or categorizing reviews accurately.

Inappropriate Content Detection:

We have successfully implemented a machine learning model to detect inappropriate content. The model is designed to automatically analyze and classify textual or visual content to identify and flag materials that violate certain guidelines or standards.

Here are the key steps we took to accomplish this:

Data Collection: We gathered a diverse dataset consisting of both appropriate and inappropriate content. This dataset included examples of different types of inappropriate content, such as hate speech, offensive language, explicit imagery, or any other form of content that violates the specified guidelines.

Data Preprocessing: We performed necessary preprocessing steps to clean and transform the data. This involved removing irrelevant information, handling any spelling or grammatical errors, and normalizing the text or images to ensure consistent and reliable input for the model.

Feature Extraction: For textual content, We extracted relevant features using techniques such as bag-of-words, TF-IDF (Term Frequency-Inverse Document Frequency), or word embeddings (e.g., Word2Vec, GloVe). For visual content, we employed image processing techniques like resizing, cropping, or extracting visual features using convolutional neural networks (CNNs) or pre-trained models such as ResNet or VGGNet.

Model Selection and Training: We selected an appropriate machine learning algorithm, such as a deep learning model (e.g., CNN, LSTM) or traditional machine learning algorithms (e.g., Support Vector Machines, Random Forests). We trained the model using the preprocessed data, ensuring a balanced representation of both appropriate and inappropriate content to avoid bias.

Labeling and Annotation: We labeled the dataset to indicate whether each example contains inappropriate content or not. This labeling process was crucial for supervised learning, enabling the model to learn from labeled data and make predictions on unseen content.

Model Evaluation: We assessed the performance of the trained model using evaluation metrics such as accuracy, precision, recall, F1-score, or area under the ROC curve. Additionally, we conducted extensive testing on a separate validation or test set to measure how effectively the model detects inappropriate content and minimizes false positives or false negatives.

Iterative Refinement: Based on the evaluation results, We fine-tuned the model and adjusted hyper parameters to improve its performance. This iterative process involved experimenting with different architectures, regularization techniques, learning rates, or batch sizes, aiming to achieve the best possible results.

Deployment and Integration: Once the model demonstrated satisfactory performance, we deployed it for real-world use. This involved integrating the model into an application or system where it can process and analyze incoming content, automatically detecting and flagging inappropriate materials.

Ongoing Monitoring and Maintenance: We established a system for continuous monitoring and maintenance of the model. This includes periodically retraining the model with new data, ensuring it remains effective and adaptable to evolving trends or new types of inappropriate content. Regular feedback and monitoring from users can help identify any false positives or false negatives, allowing for model refinements and further improvements.

Product Based Recommendation Engine:

We have also successfully implemented a product recommendation model that leverages the power of embeddings computed through sentence transformer and utilizes similarity metrics to rank products. This model aims to provide personalized recommendations based on the similarity of product descriptions or attributes.

Here's an overview of the steps we took to develop the recommendation model:

Dataset Cleaning: We carefully cleaned the dataset of product information, ensuring consistency and removing any irrelevant or duplicate entries. This involved handling missing values, standardizing attributes, and addressing any inconsistencies in the data.

Text Preprocessing: We performed text preprocessing techniques to clean and normalize the product descriptions or attributes. This included removing stop words, punctuation, and special characters, converting text to lowercase, and handling any spelling or grammatical errors. Preprocessing ensures a consistent representation of the text data for further analysis.

Sentence Embedding: We utilized the sentence transformer model to compute high-dimensional vector representations, known as sentence embeddings, for the product descriptions or attributes. These embeddings capture the semantic and contextual information of the text, allowing for a more comprehensive understanding of product similarities.

Similarity Computation: Using the computed sentence embeddings, we employed similarity metrics, such as cosine similarity or Euclidean distance, to quantify the similarity between product descriptions or attributes. These metrics enable the model to identify products that share similar characteristics or attributes.

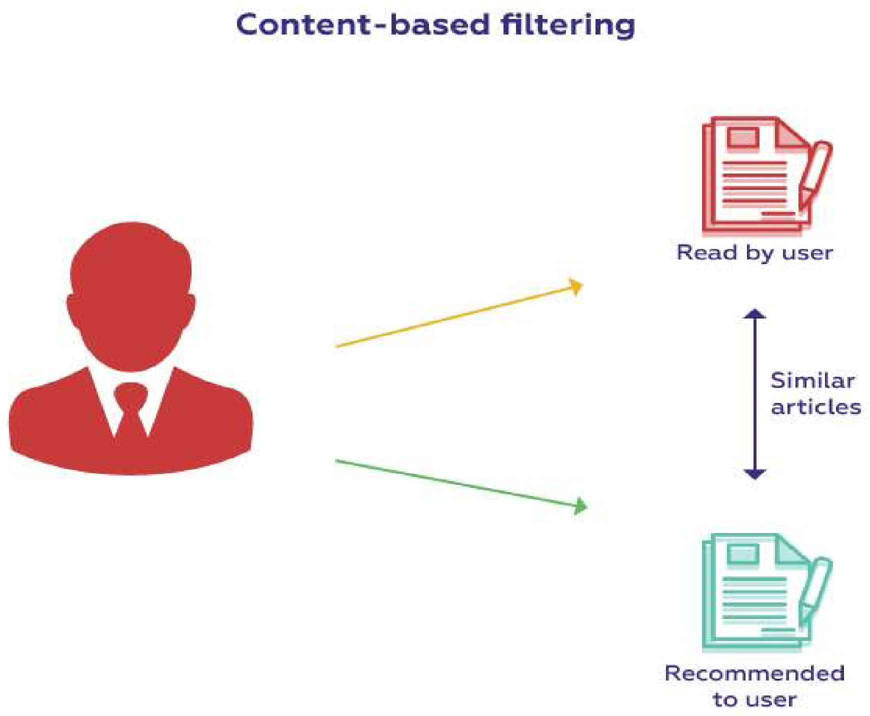
Ranking and Recommendation: Based on the computed similarities, we ranked the products to generate personalized recommendations. The ranking can be determined by selecting the top N most similar products to a given input or by applying thresholds to filter out less relevant recommendations. This approach ensures that the most relevant products are suggested to users.

Evaluation and Validation: To assess the performance of the recommendation model, we used evaluation metrics such as precision, recall, or mean average precision (MAP). We compared the recommended products against a validation set or user feedback to measure the model's effectiveness in providing accurate and relevant recommendations.

Fine-tuning and Optimization: We fine-tuned the model's hyperparameters, including the selection of the sentence transformer model, the similarity metric, and any thresholds or ranking strategies. Through experimentation and validation, we optimized the model to improve its recommendation quality and ensure it aligns with the desired objectives.

Deployment and Integration: Once satisfied with the performance, we deployed the product recommendation model, making it accessible to users. This could involve integrating the model into a web or mobile application, where users can input their preferences or search queries to receive personalized product recommendations.

Ongoing Monitoring and Maintenance: To ensure the model remains effective over time, we established a monitoring system to track user feedback, evaluate recommendation performance, and identify any potential drift or changes in user preferences. Regular updates and retraining can help adapt the model to evolving user needs and maintain its recommendation quality.



User Based Recommendation Engine:

We have successfully implemented a user-based recommendation system using TensorFlow Recommenders (TFRS). TFRS provides a powerful framework for building and training recommendation models based on user and product features. Here's an overview of the steps we have taken, keeping in mind the TensorFlow Recommenders structure:

Dataset Cleaning: We carefully cleaned the dataset of user and product information, ensuring consistency and removing any irrelevant or duplicate entries. This involved handling missing values, standardizing attributes, and addressing any inconsistencies in the data.

Data Preparation: We prepared the dataset by structuring it into appropriate TensorFlow data structures, such as tf.data.Dataset. This enabled efficient processing and training of the recommendation model.

Feature Engineering: We identified the relevant features for the recommendation model, such as user ID, product ID, ratings, user region, and other attributes. These features provide valuable information for modeling user preferences and item characteristics.

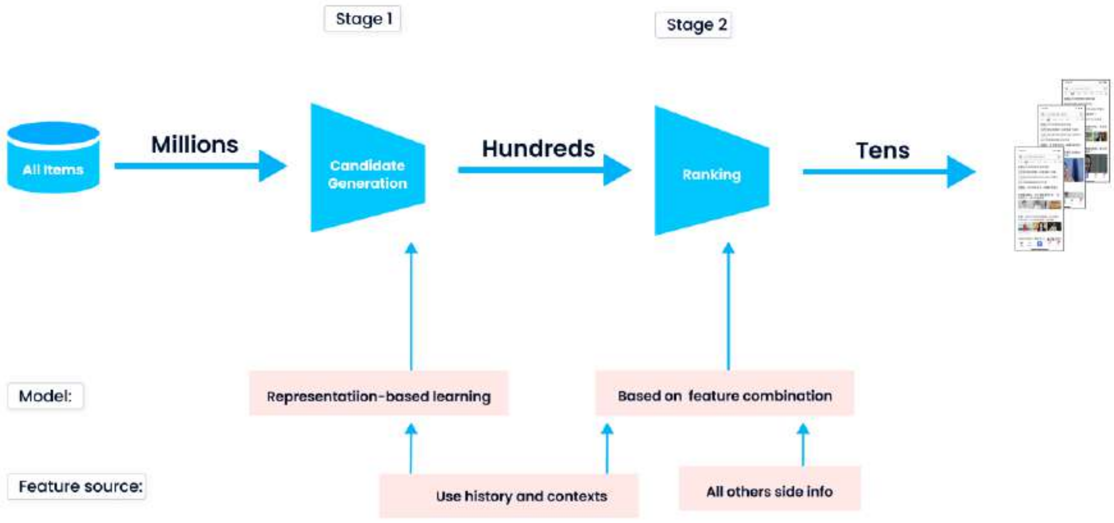
Data Encoding: Using TensorFlow's feature columns, we encoded the categorical features, such as user ID and product ID, into numerical representations. This encoding step allows the model to understand and process the categorical features effectively.

Model Architecture: We designed the architecture of the recommendation model using TensorFlow Recommenders. This typically involves defining the user and product embedding layers, which capture the latent representations of users and items. We may have also incorporated additional layers, such as deep neural networks, to capture complex interactions between user and item features.

Training: We trained the recommendation model using the prepared dataset and the defined architecture. TensorFlow Recommenders provides training utilities and loss functions specifically tailored for recommendation tasks, such as the pointwise or pairwise loss functions. During training, the model learns to predict user ratings or item preferences based on the given features.

Evaluation: We evaluated the performance of the trained recommendation model using appropriate evaluation metrics, such as mean squared error (MSE) or mean absolute error (MAE), to measure the accuracy of the predicted ratings. This evaluation helps assess how well the model generalizes to unseen user-item interactions and provides insights into its performance.

Retrieval Task: In addition to rating prediction, we incorporated a retrieval task into the recommendation system. This task aims to retrieve the most relevant items for a given user, based on their preferences and past interactions. TensorFlow Recommenders provides retrieval utilities, such as candidate generation and ranking models, to facilitate this task and improve the recommendation quality.



5. Non-Functional Requirements:

5.1. Performance Requirements:

An online shopping business has multiple organizational levels, and its overall success is influenced by a variety of factors.

System of information:

The infrastructure and organization of an information system can have a significant impact on its performance in the following ways.

- Average Response time of Web pages
- Site Maintenance Cost
- Failure Rate

To retain an acceptable performance while handling the maximum number of requests from a single client, any number of users should be able to use the system, at any time. All e-commerce programs must have a nice UI/UX. While a visually pleasing design is crucial, image optimization and other approaches may be used to keep the site from becoming too heavy. A systematic answer to numerous continuing difficulties is provided through customizable smart goals, third-party data integration, and real-time optimization.

Logistics:

- Availability of Products
- Liability Failure Rate
- Delivery Quality

- Average Delivery Time

To solve logistical challenges, the service must guarantee that the proper product is available at the appropriate moment and in acceptable quality.

The activity of Sales:

- Purchasing cost over the first customer
- Awareness of Brands

By ensuring these mentioned aspects, sales activity can be enhanced.

Following are some important points that illustrate major performance criteria for structural analysis:

- **Sales Processing:**
    - Personal Cost
    - Return on Sales
    - Marketing Cost
    - Attainment Cost
    - Total Turnover

- **Market and Customer Analysis:**
    - Number of new individual customers
    - Frequency of Average Visiting
    - Average of Order Price
    - Frequency of First Buyers

6. Other Requirements:

6.1. Recommendation Model:

The server is responsible for keeping the recommendation model up to date. The recommendation engine is used to personalize the goods that appear on the customer's home page. These are created by utilizing collaborative filtering models. This model will remove unwater content from the whole application and recommend the most highly rated items.

6.2. Rating and Reviews:

The client subsystem has access to the rating and review functionalities, which are used to develop a consumer community. Each consumer has the option of reviewing purchased items and giving ratings or levels of satisfaction. Products with low satisfaction ratings may be terminated by the system.

7. Architecture and Activity Diagrams

7.1.UML class Diagram:

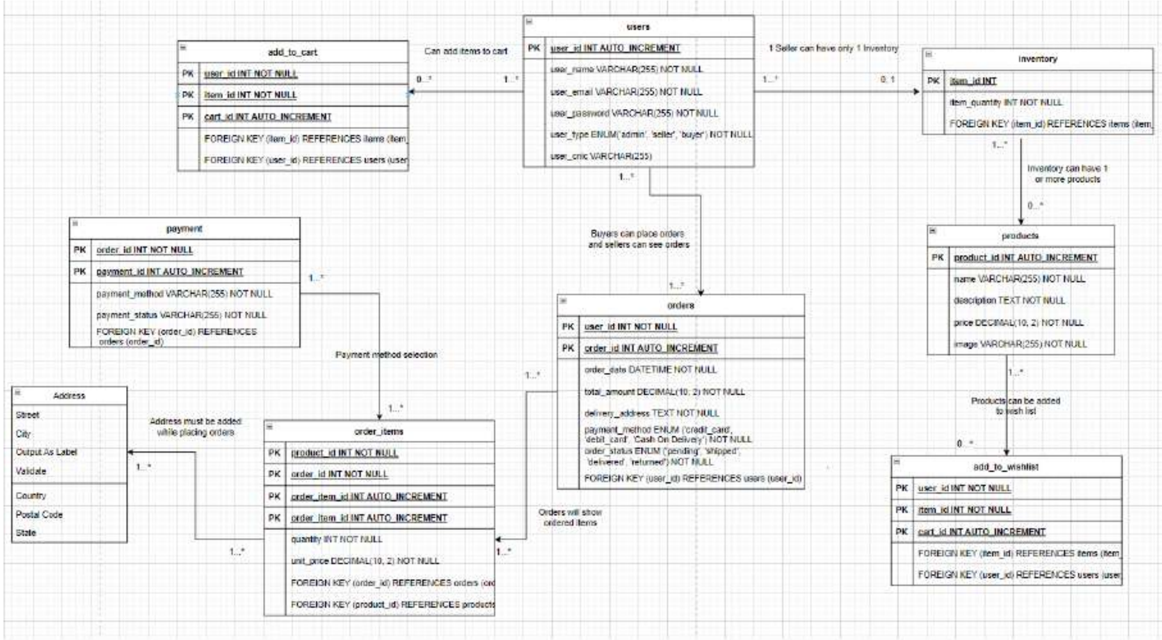

add_to_cart
PK user_id INT NOT NULL
PK item_id INT NOT NULL
PK cart_id INT AUTO_INCREMENT
FOREIGN KEY (item_id) REFERENCES items (item_
FOREIGN KEY (user_id) REFERENCES users (user
Can add items to cart
0...*
1...*
users
PK user_id INT AUTO_INCREMENT
user_name VARCHAR(255) NOT NULL
user_email VARCHAR(255) NOT NULL
user_password VARCHAR(255) NOT NULL
user_type ENUM('admin', 'seller', 'buyer') NOT NULL
user_cnic VARCHAR(255)
1 Seller can have only 1 Inventory
1...*
0..1
inventory
PK item_id INT
item_quantity INT NOT NULL
FOREIGN KEY (item_id) REFERENCES items (item_
1...*
Inventory can have 1 or more products
0...*
products
PK product_id INT AUTO_INCREMENT
name VARCHAR(255) NOT NULL
description TEXT NOT NULL
price DECIMAL(10, 2) NOT NULL
image VARCHAR(255) NOT NULL
1...*
Products can be added to wish list
0...*
add_to_wishlist
PK user_id INT NOT NULL
PK item_id INT NOT NULL
PK cart_id INT AUTO_INCREMENT
FOREIGN KEY (item_id) REFERENCES items (item_
FOREIGN KEY (user_id) REFERENCES users (user
1...*
Buyers can place orders and sellers can see orders
1...*
orders
PK user_id INT NOT NULL
PK order_id INT AUTO_INCREMENT
order_date DATETIME NOT NULL
total_amount DECIMAL(10, 2) NOT NULL
delivery_address TEXT NOT NULL
payment_method ENUM ('credit_card', 'debit_card', 'Cash On Delivery') NOT NULL
order_status ENUM ('pending', 'shipped', 'delivered', 'returned') NOT NULL
FOREIGN KEY (user_id) REFERENCES users (user_id)
1...*
Orders will show ordered items
1...*
payment
PK order_id INT NOT NULL
PK payment_id INT AUTO_INCREMENT
payment_method VARCHAR(255) NOT NULL
payment_status VARCHAR(255) NOT NULL
FOREIGN KEY (order_id) REFERENCES orders (order_id)
1...*
Payment method selection
1...*
order_items
PK product_id INT NOT NULL
PK order_id INT NOT NULL
PK order_item_id INT AUTO_INCREMENT
PK order_item_id INT AUTO_INCREMENT
quantity INT NOT NULL
unit_price DECIMAL(10, 2) NOT NULL
FOREIGN KEY (order_id) REFERENCES orders (ord
FOREIGN KEY (product_id) REFERENCES products
Address
Street
City
Output As Label
Validate
Country
Postal Code
State
Address must be added while placing orders
1...*
1...*

Fig 13: UML Diagram

7.2 Activity Diagram:

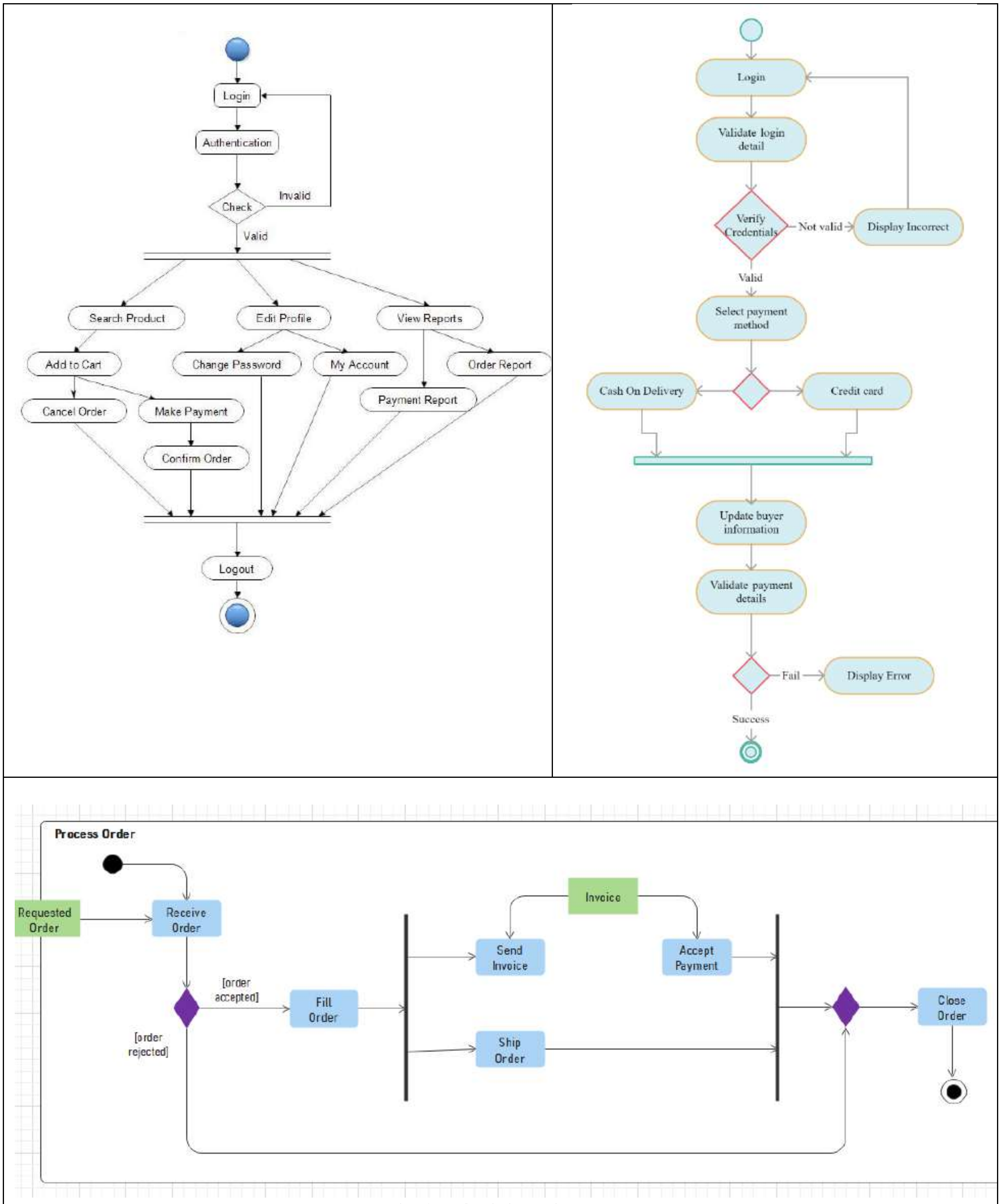


Fig 14: Activity Diagram

7.3 Data flow diagrams:

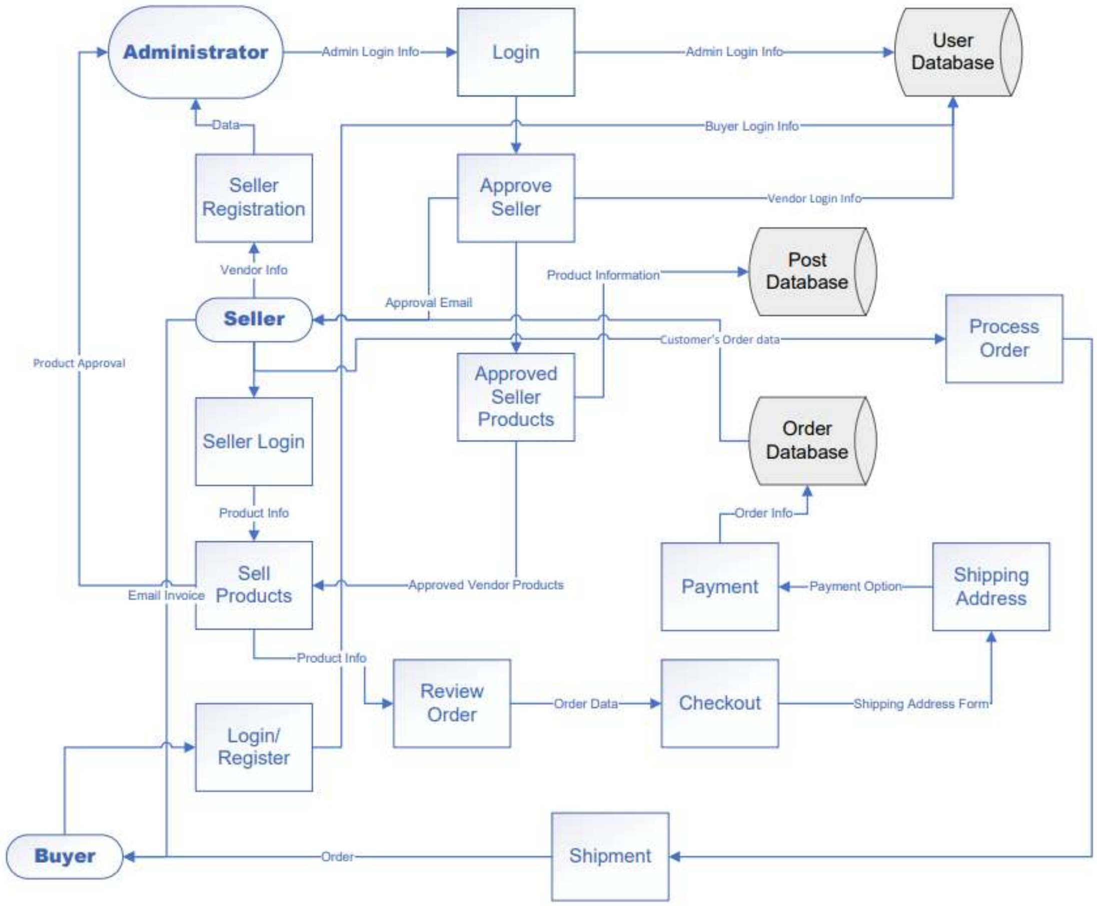


Fig 15: Dataflow Diagram

7.4. ERD Diagram:

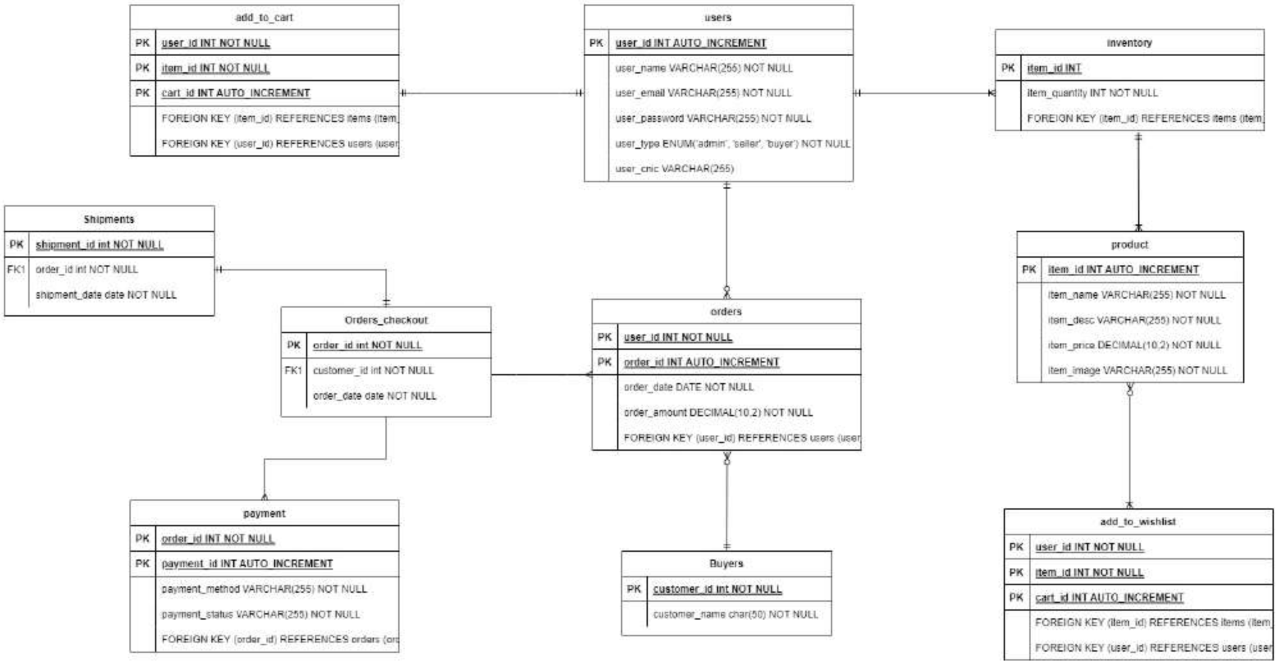
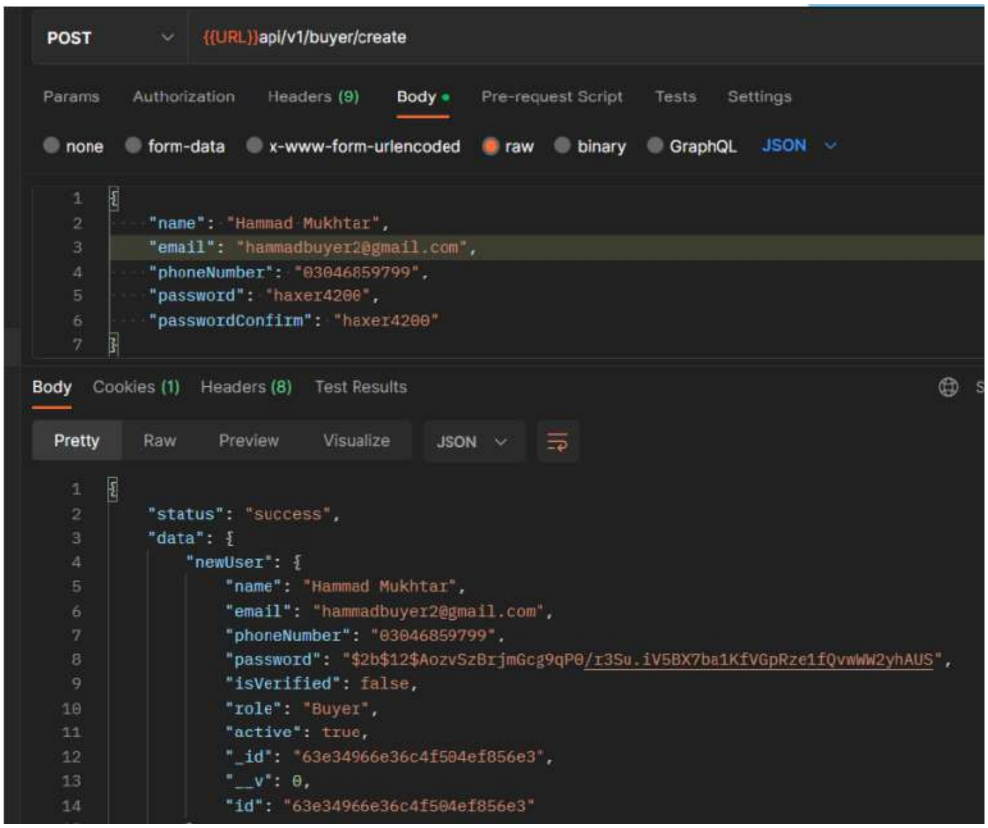

add_to_cart
PK user_id INT NOT NULL
PK item_id INT NOT NULL
PK cart_id INT AUTO_INCREMENT
FOREIGN KEY (item_id) REFERENCES items (item_
FOREIGN KEY (user_id) REFERENCES users (user
users
PK user_id INT AUTO_INCREMENT
user_name VARCHAR(255) NOT NULL
user_email VARCHAR(255) NOT NULL
user_password VARCHAR(255) NOT NULL
user_type ENUM('admin', 'seller', 'buyer') NOT NULL
user_cnic VARCHAR(255)
inventory
PK item_id INT
item_quantity INT NOT NULL
FOREIGN KEY (item_id) REFERENCES items (item_
Shipments
PK shipment_id int NOT NULL
FK1 order_id int NOT NULL
shipment_date date NOT NULL
Orders_checkout
PK order_id int NOT NULL
FK1 customer_id int NOT NULL
order_date date NOT NULL
orders
PK user_id INT NOT NULL
PK order_id INT AUTO_INCREMENT
order_date DATE NOT NULL
order_amount DECIMAL(10,2) NOT NULL
FOREIGN KEY (user_id) REFERENCES users (user
product
PK item_id INT AUTO_INCREMENT
item_name VARCHAR(255) NOT NULL
item_desc VARCHAR(255) NOT NULL
item_price DECIMAL(10,2) NOT NULL
item_image VARCHAR(255) NOT NULL
payment
PK order_id INT NOT NULL
PK payment_id INT AUTO_INCREMENT
payment_method VARCHAR(255) NOT NULL
payment_status VARCHAR(255) NOT NULL
FOREIGN KEY (order_id) REFERENCES orders (ord
Buyers
PK customer_id int NOT NULL
customer_name char(50) NOT NULL
add_to_wishlist
PK user_id INT NOT NULL
PK item_id INT NOT NULL
PK cart_id INT AUTO_INCREMENT
FOREIGN KEY (item_id) REFERENCES items (item_
FOREIGN KEY (user_id) REFERENCES users (user

Fig 16: ERD Diagram

8. Testing:

Here is how we done unit testing using Postman. Which check each unit value and provide error of that value, if occurs.



POST {{URL}}api/v1/buyer/create

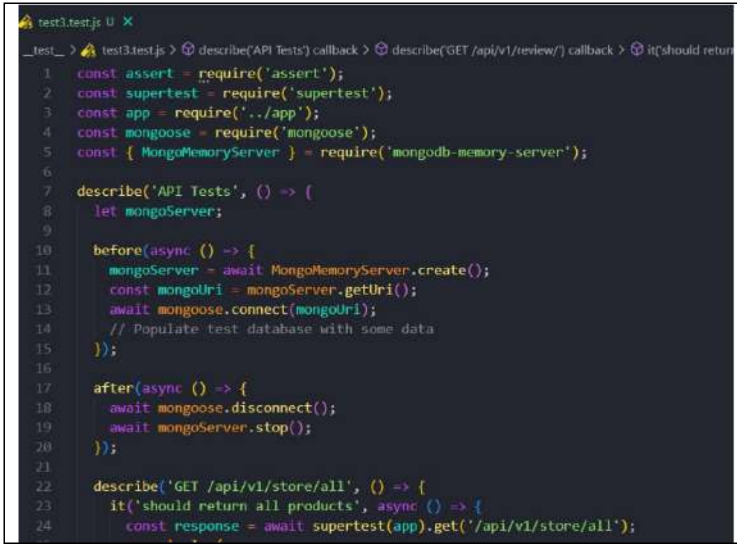
● raw JSON

```
{
····"name": "Hammad Mukhtar",
····"email": "hammadbuyer2@gmail.com",
····"phoneNumber": "03046859799",
····"password": "haxer4200",
····"passwordConfirm": "haxer4200"
}
```

Body Pretty JSON

```
{
    "status": "success",
    "data": {
        "newUser": {
            "name": "Hammad Mukhtar",
            "email": "hammadbuyer2@gmail.com",
            "phoneNumber": "03046859799",
            "password": "$2b$12$AozvSzBrjmGcg9qP0/r3Su.iV5BX7ba1KfVGpRze1fQvwWW2yhAUS",
            "isVerified": false,
            "role": "Buyer",
            "active": true,
            "_id": "63e34966e36c4f504ef856e3",
            "__v": 0,
            "id": "63e34966e36c4f504ef856e3"
```

We used "Mocha JS" library for api's end point testing. Here are some snaps:


test3.test.js U ×
test > test3.test.js > describe('API Tests') callback > describe('GET /api/v1/review/') callback > it('should retur

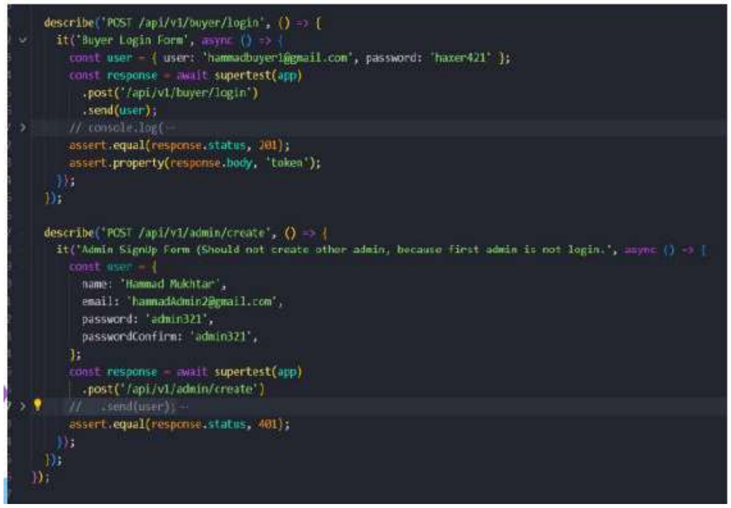
```
const assert = require('assert');
const supertest = require('supertest');
const app = require('../app');
const mongoose = require('mongoose');
const { MongoMemoryServer } = require('mongodb-memory-server');

describe('API Tests', () => {
  let mongoServer;

  before(async () => {
    mongoServer = await MongoMemoryServer.create();
    const mongoUri = mongoServer.getUri();
    await mongoose.connect(mongoUri);
    // Populate test database with some data
  });

  after(async () => {
    await mongoose.disconnect();
    await mongoServer.stop();
  });

  describe('GET /api/v1/store/all', () => {
    it('should return all products', async () => {
      const response = await supertest(app).get('/api/v1/store/all');
```

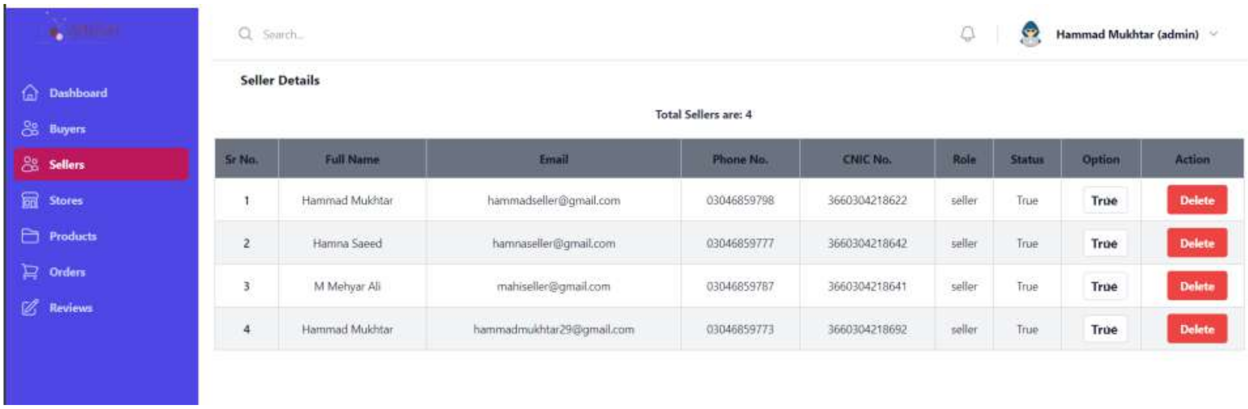


```
  describe('POST /api/v1/buyer/login', () => {
    it('Buyer Login Form', async () => {
      const user = { user: 'hammadbuyer1@gmail.com', password: 'haxer421' };
      const response = await supertest(app)
        .post('/api/v1/buyer/login')
        .send(user);
      // console.log(…
      assert.equal(response.status, 201);
      assert.property(response.body, 'token');
    });
  });

  describe('POST /api/v1/admin/create', () => {
    it('Admin SignUp Form (Should not create other admin, because first admin is not login.', async () => {
      const user = {
        name: 'Hammad Mukhtar',
        email: 'hammadAdmin2@gmail.com',
        password: 'admin321',
        passwordConfirm: 'admin321',
      };
      const response = await supertest(app)
        .post('/api/v1/admin/create')
      //   .send(user); …
      assert.equal(response.status, 401);
    });
  });
});
```

9. Front-End:

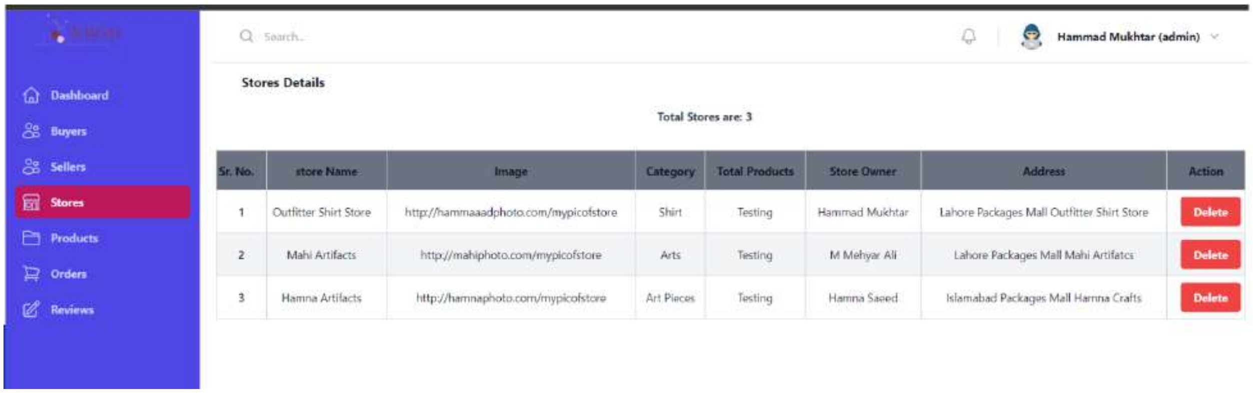
Here are some frontend screenshots of Dashboard of "Admin" and "Seller":



Seller Details

Total Sellers are: 4

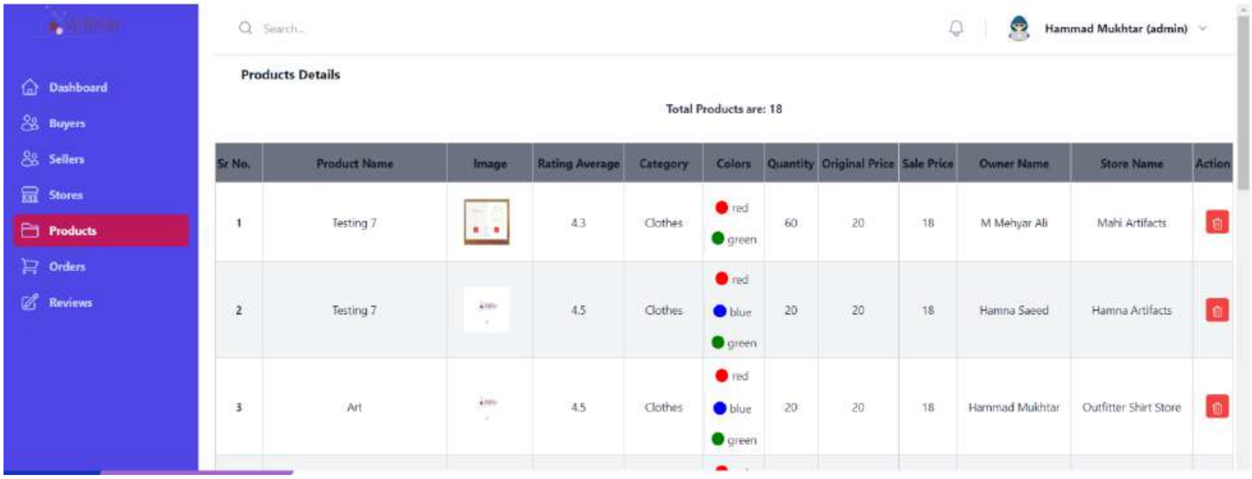
Sr No.	Full Name	Email	Phone No.	CNIC No.	Role	Status	Option	Action
1	Hammad Mukhtar	hammadseller@gmail.com	03046859798	3660304218622	seller	True	True	Delete
2	Hamna Saeed	hamnaseller@gmail.com	03046859777	3660304218642	seller	True	True	Delete
3	M Mehyar Ali	mahiseller@gmail.com	03046859787	3660304218641	seller	True	True	Delete
4	Hammad Mukhtar	hammadmukhtar29@gmail.com	03046859773	3660304218692	seller	True	True	Delete



Stores Details

Total Stores are: 3

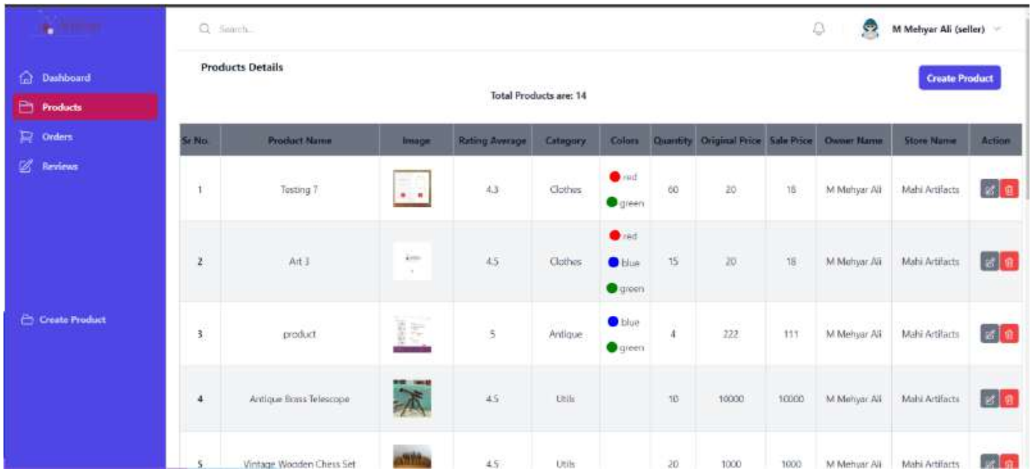
Sr. No.	store Name	Image	Category	Total Products	Store Owner	Address	Action
1	Outfitter Shirt Store	http://hammaaadphoto.com/mypicofstore	Shirt	Testing	Hammad Mukhtar	Lahore Packages Mall Outfitter Shirt Store	Delete
2	Mahi Artifacts	http://mahiphoto.com/mypicofstore	Arts	Testing	M Mehyar Ali	Lahore Packages Mall Mahi Artifatcs	Delete
3	Hamna Artifacts	http://hamnaphoto.com/mypicofstore	Art Pieces	Testing	Hamna Saeed	Islamabad Packages Mall Hamna Crafts	Delete



Products Details

Total Products are: 18

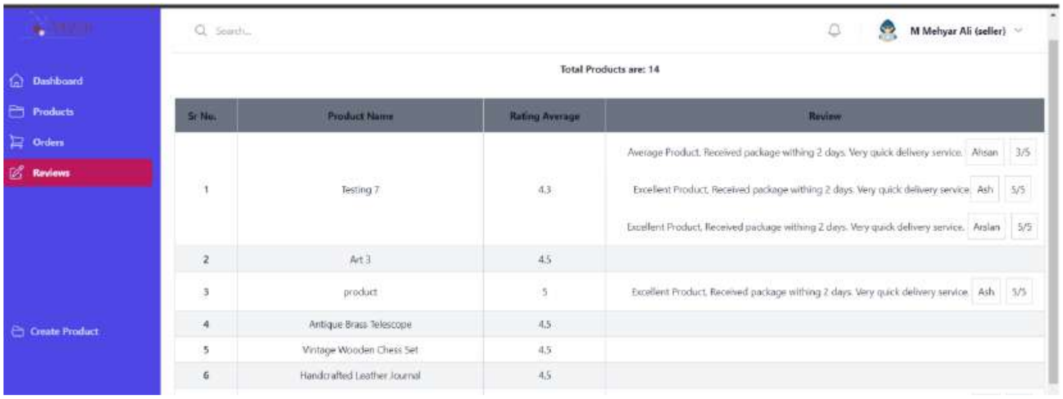
Sr No.	Product Name	Image	Rating Average	Category	Colors	Quantity	Original Price	Sale Price	Owner Name	Store Name	Action
1	Testing 7		4.3	Clothes	● red ● green	60	20	18	M Mehyar Ali	Mahi Artifacts	🗑
2	Testing 7		4.5	Clothes	● red ● blue ● green	20	20	18	Hamna Saeed	Hamna Artifacts	🗑
3	Art		4.5	Clothes	● red ● blue ● green	20	20	18	Hammad Mukhtar	Outfitter Shirt Store	🗑



Products Details

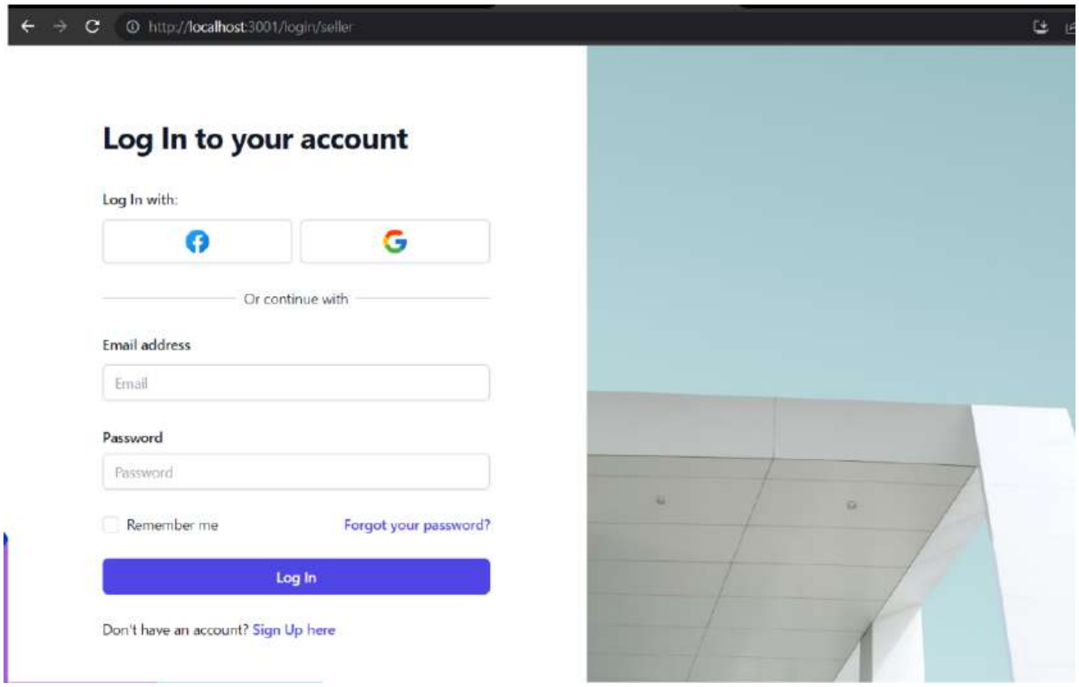
Total Products are: 14

Sr No.	Product Name	Image	Rating Average	Category	Colors	Quantity	Original Price	Sale Price	Owner Name	Store Name	Action
1	Testing 7		4.3	Clothes	● red ● green	60	20	18	M Mehyar Ali	Mahi Artifacts	
2	Art 3		4.5	Clothes	● red ● blue ● green	15	20	18	M Mehyar Ali	Mahi Artifacts	
3	product		5	Antique	● blue ● green	4	222	111	M Mehyar Ali	Mahi Artifacts	
4	Antique Brass Telescope		4.5	Utils		10	10000	10000	M Mehyar Ali	Mahi Artifacts	
5	Vintage Wooden Chess Set		4.5	Utils		20	1000	1000	M Mehyar Ali	Mahi Artifacts	



Total Products are: 14

Sr No.	Product Name	Rating Average	Review
1	Testing 7	4.3	Average Product. Received package withing 2 days. Very quick delivery service. Ahsan 3/5 Excellent Product. Received package withing 2 days. Very quick delivery service. Ash 5/5 Excellent Product. Received package withing 2 days. Very quick delivery service. Arslan 5/5
2	Art 3	4.5	
3	product	5	Excellent Product. Received package withing 2 days. Very quick delivery service. Ash 5/5
4	Antique Brass Telescope	4.5	
5	Vintage Wooden Chess Set	4.5	
6	Handcrafted Leather Journal	4.5	



Log In to your account

Log In with:

Or continue with

Email address

Email

Password

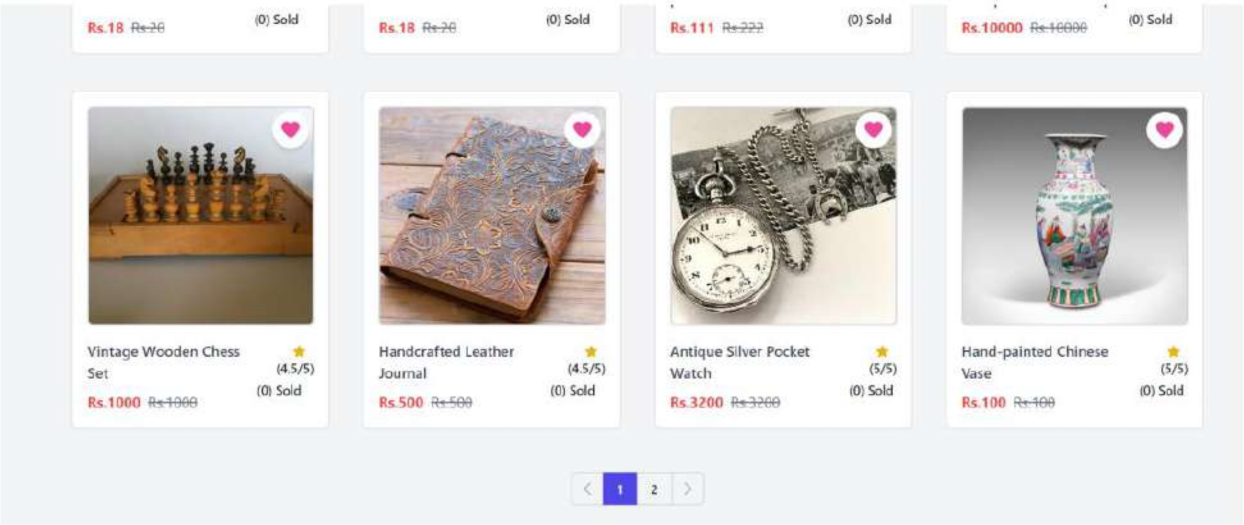
Password

□ Remember me Forgot your password?

Log In

Don't have an account? Sign Up here

Here are some screenshots of our website frontend:



Rs.18 ~~Rs.20~~ (0) Sold

Rs.18 ~~Rs.20~~ (0) Sold

Rs.111 ~~Rs.222~~ (0) Sold

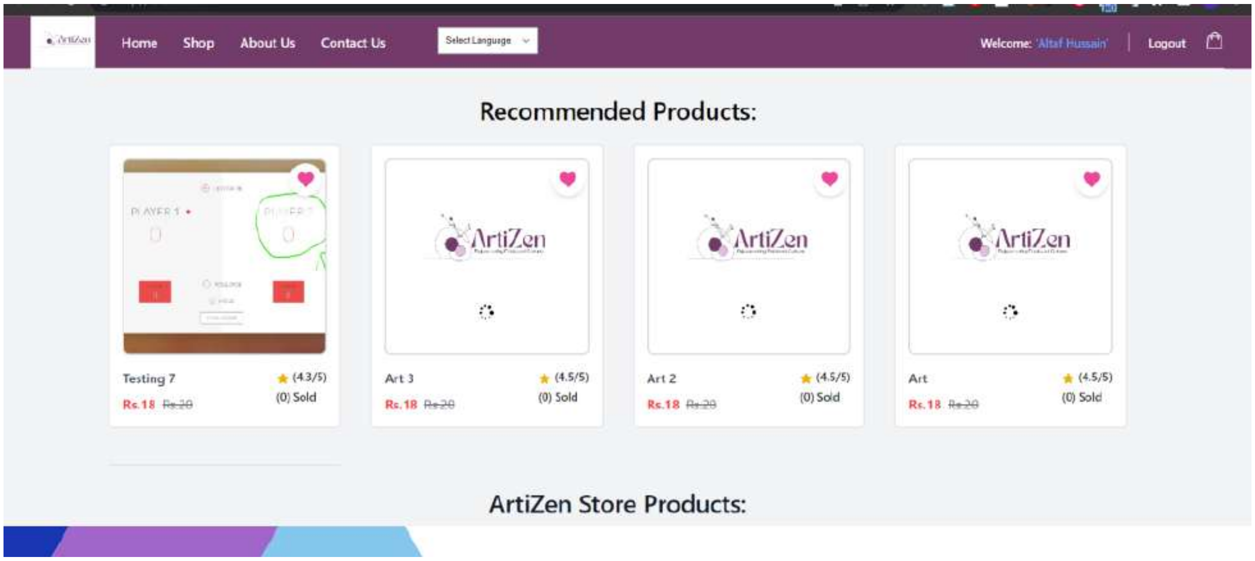
Rs.10000 ~~Rs.10000~~ (0) Sold

Vintage Wooden Chess Set ★ (4.5/5) (0) Sold
Rs.1000 ~~Rs.1000~~

Handcrafted Leather Journal ★ (4.5/5) (0) Sold
Rs.500 ~~Rs.500~~

Antique Silver Pocket Watch ★ (5/5) (0) Sold
Rs.3200 ~~Rs.3200~~

Hand-painted Chinese Vase ★ (5/5) (0) Sold
Rs.100 ~~Rs.100~~



Recommended Products:

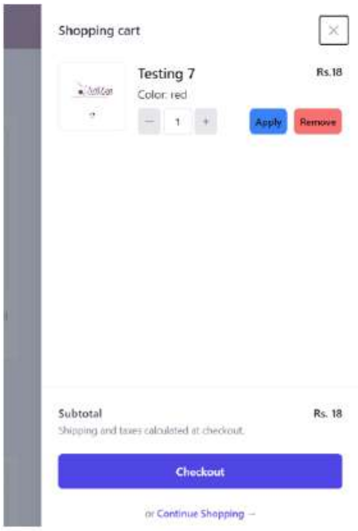
Testing 7 ★ (4.3/5) (0) Sold
Rs.18 ~~Rs.20~~

Art 3 ★ (4.5/5) (0) Sold
Rs.18 ~~Rs.20~~

Art 2 ★ (4.5/5) (0) Sold
Rs.18 ~~Rs.20~~

Art ★ (4.5/5) (0) Sold
Rs.18 ~~Rs.20~~

ArtiZen Store Products:



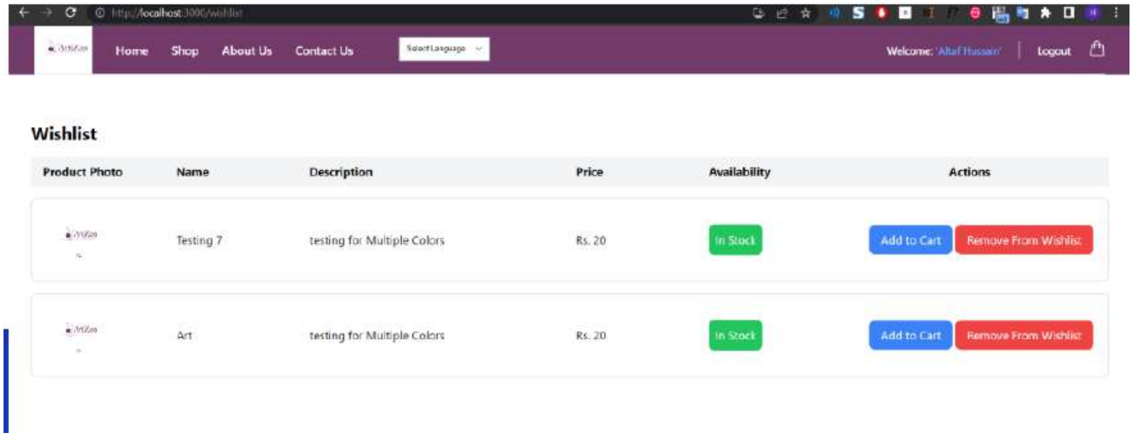
Shopping cart

Testing 7 Rs.18

Color: red

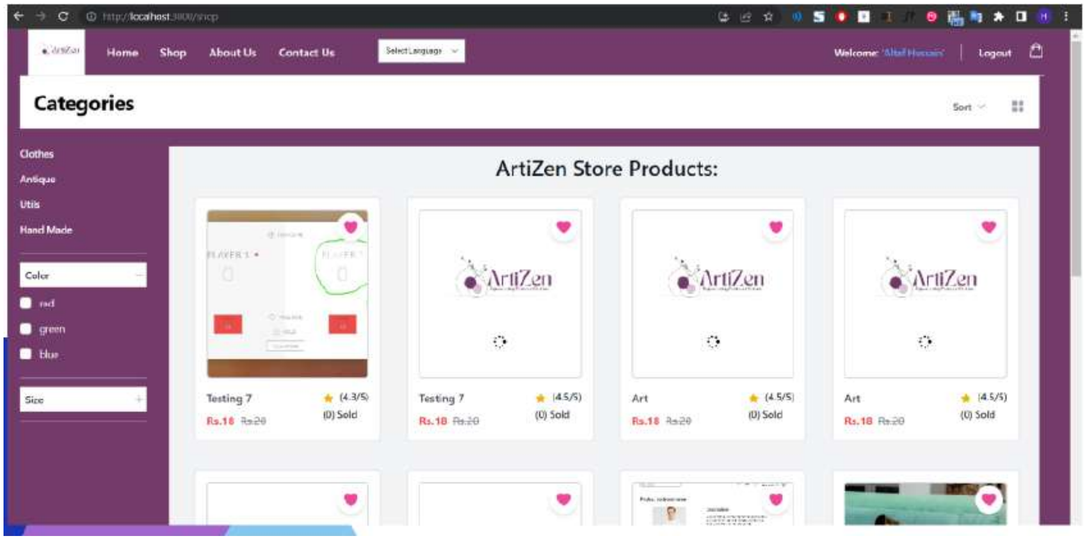
Subtotal Rs. 18

Shipping and taxes calculated at checkout.



Wishlist

Product Photo	Name	Description	Price	Availability	Actions
	Testing 7	testing for Multiple Colors	Rs. 20	In Stock	Add to Cart Remove From Wishlist
	Art	testing for Multiple Colors	Rs. 20	In Stock	Add to Cart Remove From Wishlist



Categories

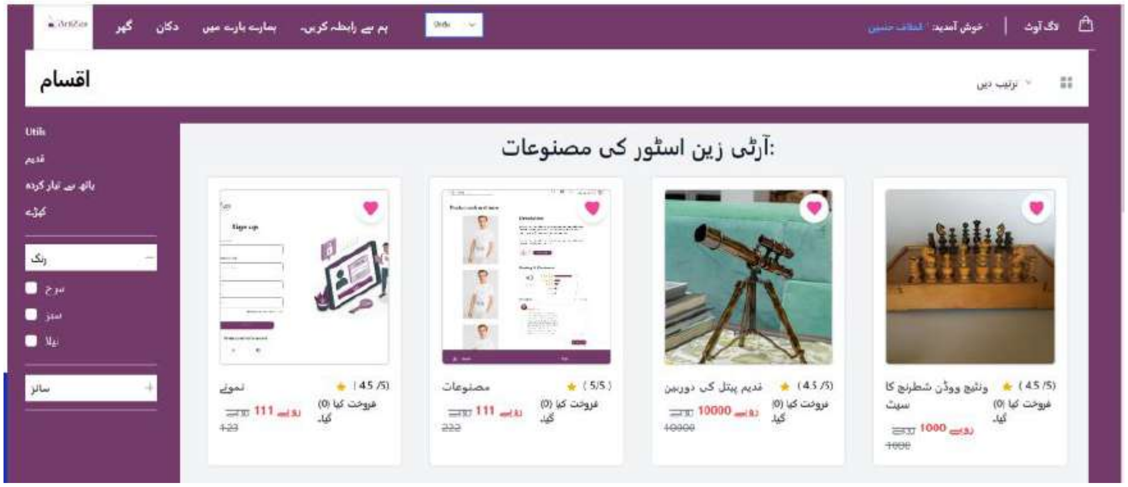
ArtiZen Store Products:

Testing 7 (4.3/5) Rs.18 Rs.20 (0) Sold

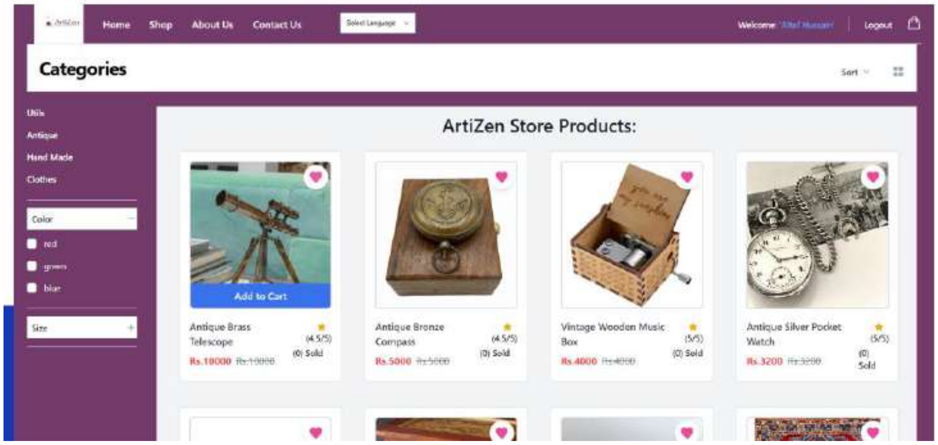
Testing 7 (4.5/5) Rs.18 Rs.20 (0) Sold

Art (4.5/5) Rs.18 Rs.20 (0) Sold

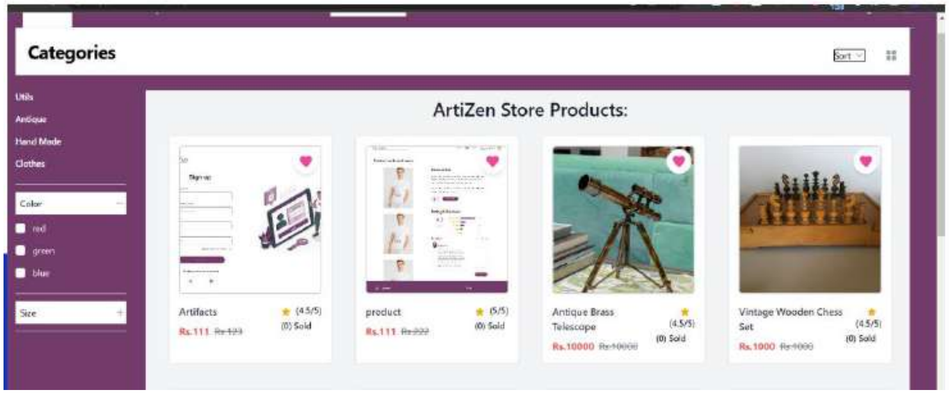
Art (4.5/5) Rs.18 Rs.20 (0) Sold


ArtiZen
گھر
دکان
ہمارے بارے میں
ہم سے رابطہ کریں۔
Urdu
خوش آمدید: الطاف حسین
لاگ آوٹ
اقسام
ترتیب دیں
Utils
قدیم
ہاتھ سے تیار کردہ
کپڑے
رنگ
سرخ
سبز
نیلا
سائز
آرٹی زین اسٹور کی مصنوعات:
نمونے (4.5 /5) روپے 111 ~~روپے 123~~ فروخت کیا (0) گیا۔
مصنوعات (5/5) روپے 111 ~~روپے 222~~ فروخت کیا (0) گیا۔
قدیم پیتل کی دوربین (4.5 /5) روپے 10000 ~~روپے 10000~~ فروخت کیا (0) گیا۔
ونٹیج ووڈن شطرنج کا سیٹ (4.5 /5) روپے 1000 ~~روپے 1000~~ فروخت کیا (0) گیا۔

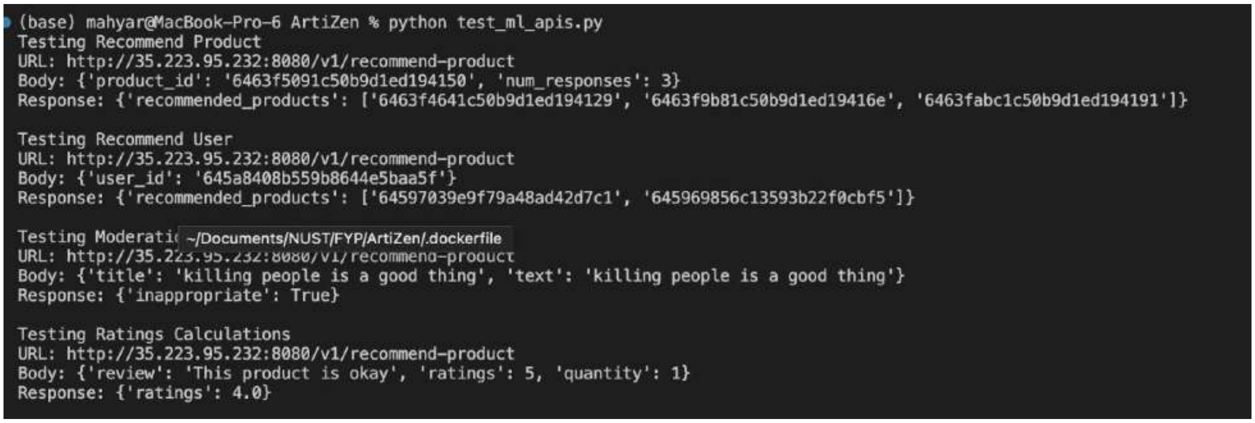
Website: Products Sort by High to Low


ArtiZen
Home
Shop
About Us
Contact Us
Select Language
Welcome Altaf Hussain
Logout
Categories
Sort
Utils
Antique
Hand Made
Clothes
Color
red
green
blue
Size
ArtiZen Store Products:
Add to Cart
Antique Brass Telescope (4.5/5) (0) Sold Rs.10000 ~~Rs.10000~~
Antique Bronze Compass (4.5/5) (0) Sold Rs.5000 ~~Rs.5000~~
Vintage Wooden Music Box (5/5) (0) Sold Rs.4000 ~~Rs.4000~~
Antique Silver Pocket Watch (5/5) (0) Sold Rs.3200 ~~Rs.3200~~

Website: Products Sort by Newest


Categories
Sort
Utils
Antique
Hand Made
Clothes
Color
red
green
blue
Size
ArtiZen Store Products:
Artifacts (4.5/5) (0) Sold Rs.111 ~~Rs.123~~
product (5/5) (0) Sold Rs.111 ~~Rs.222~~
Antique Brass Telescope (4.5/5) (0) Sold Rs.10000 ~~Rs.10000~~
Vintage Wooden Chess Set (4.5/5) (0) Sold Rs.1000 ~~Rs.1000~~

10. Machine Learning APIs Testing:

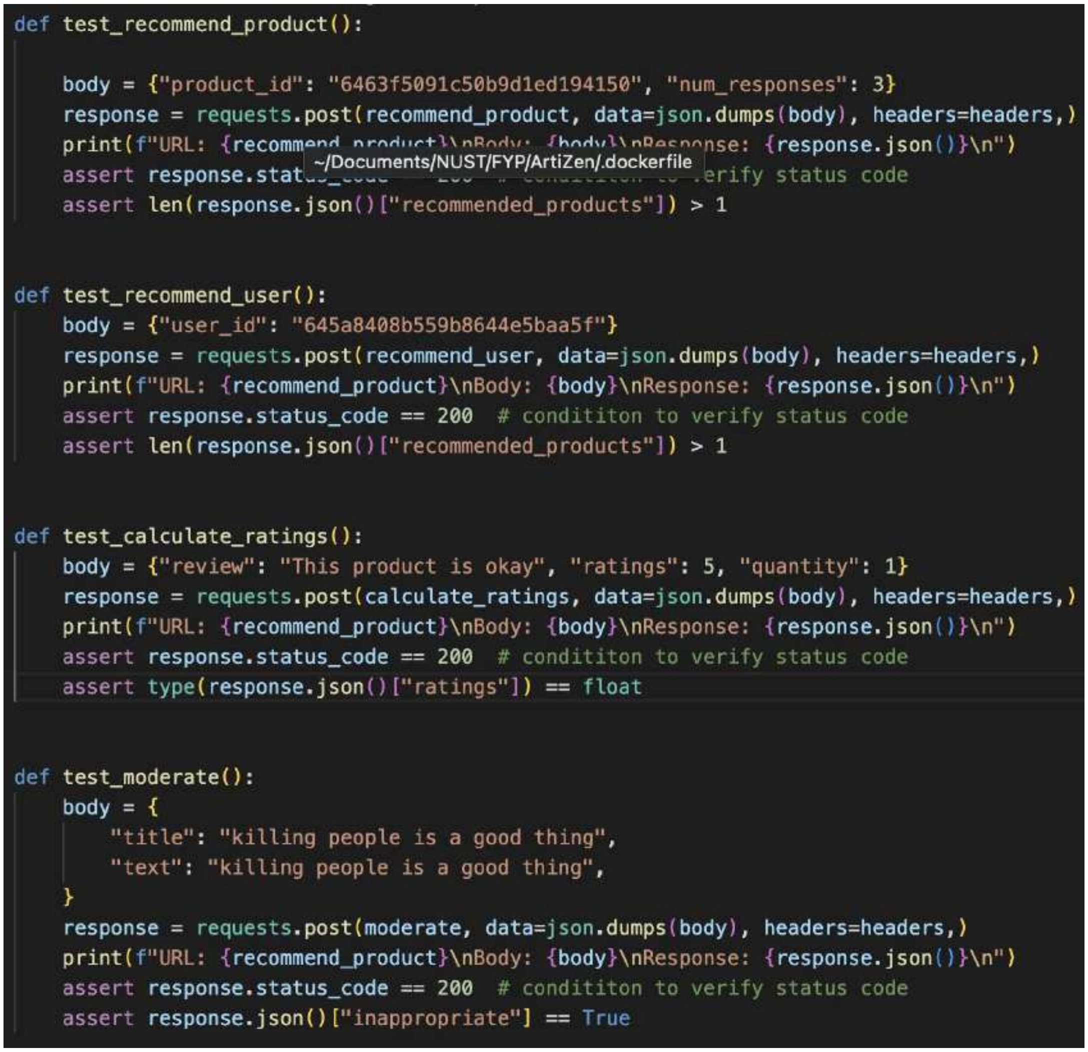


```
(base) mahyar@MacBook-Pro-6 ArtiZen % python test_ml_apis.py
Testing Recommend Product
URL: http://35.223.95.232:8080/v1/recommend-product
Body: {'product_id': '6463f5091c50b9d1ed194150', 'num_responses': 3}
Response: {'recommended_products': ['6463f4641c50b9d1ed194129', '6463f9b81c50b9d1ed19416e', '6463fabc1c50b9d1ed194191']}

Testing Recommend User
URL: http://35.223.95.232:8080/v1/recommend-product
Body: {'user_id': '645a8408b559b8644e5baa5f'}
Response: {'recommended_products': ['64597039e9f79a48ad42d7c1', '645969856c13593b22f0cbf5']}

Testing Moderati ~/Documents/NUST/FYP/ArtiZen/.dockerfile
URL: http://35.223.95.232:8080/v1/recommend-product
Body: {'title': 'killing people is a good thing', 'text': 'killing people is a good thing'}
Response: {'inappropriate': True}

Testing Ratings Calculations
URL: http://35.223.95.232:8080/v1/recommend-product
Body: {'review': 'This product is okay', 'ratings': 5, 'quantity': 1}
Response: {'ratings': 4.0}
```



```python
def test_recommend_product():

    body = {"product_id": "6463f5091c50b9d1ed194150", "num_responses": 3}
    response = requests.post(recommend_product, data=json.dumps(body), headers=headers,)
    print(f"URL: {recommend_product}\nBody: {body}\nResponse: {response.json()}\n")
    assert response.stat ~/Documents/NUST/FYP/ArtiZen/.dockerfile erify status code
    assert len(response.json()["recommended_products"]) > 1


def test_recommend_user():
    body = {"user_id": "645a8408b559b8644e5baa5f"}
    response = requests.post(recommend_user, data=json.dumps(body), headers=headers,)
    print(f"URL: {recommend_product}\nBody: {body}\nResponse: {response.json()}\n")
    assert response.status_code == 200  # condititon to verify status code
    assert len(response.json()["recommended_products"]) > 1


def test_calculate_ratings():
    body = {"review": "This product is okay", "ratings": 5, "quantity": 1}
    response = requests.post(calculate_ratings, data=json.dumps(body), headers=headers,)
    print(f"URL: {recommend_product}\nBody: {body}\nResponse: {response.json()}\n")
    assert response.status_code == 200  # condititon to verify status code
    assert type(response.json()["ratings"]) == float


def test_moderate():
    body = {
        "title": "killing people is a good thing",
        "text": "killing people is a good thing",
    }
    response = requests.post(moderate, data=json.dumps(body), headers=headers,)
    print(f"URL: {recommend_product}\nBody: {body}\nResponse: {response.json()}\n")
    assert response.status_code == 200  # condititon to verify status code
    assert response.json()["inappropriate"] == True
```

11.References:

- Bandakkanavar, R. (2022, July 11). *Software Requirements Specification document with example*. Krazytech. https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

- Nurnobi, S. M. (n.d.). *Online eCommerce website srs*. https://www.slideshare.net/OnlyTechForYou/online-ecommerce-website-srs

- studocu (2015). *System Requirement Specification for E-commerce Web Store – Marvel Electronics and Home Entertainment*. [online] StuDocu. Available at:
- https://www.studocu.com/in/document/rashtrasant-tukadoji-maharaj-nagpur-university/book-keeping-and-accountancy/system-requirement-specification-for-ecommerce-web-store/5353426

- https://vceela.com/

- https://www.daraz.pk/

Appendix A: Glossary

Word	Explanation
ArtiZen	Product Name
IEEE	Institute of Electrical and Electronics Engineers
DB	Database
OS	Operating System
GB	Gigabytes
DB	Database
API	Application Programming Interface
OTP	One Time Password
RAM	Random Access Memory
CPU	Central Processing Unit
ID	Identification
HTTP	Hyper Text Transfer Protocol
Interface	Used to communicate across different medium
Class Diagram	Static structure Diagram describes system structure using system cases, entities, attributes, and relationships between them.
Use Case	A moderate-level diagram that describes the basic idea of the product.

Table 13: Table of Glossary