Tribhuvan University

Institute of Science and Technology

Project Report on

PROSTRIKE GEAR – Sports Product Recommendation using Content Based Filtering

Submitted to:

Department of Bachelor of Science in Computer Science and Information Technology (BSc.CSIT)

National College of Computer Studies

In partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Information Technology (BSc.CSIT)

Submitted by:

Swarup Dahal [28896/078]

Regd. No.: 5-2-551-100-2021

Under the supervision of

Mr. Dinesh Khadka

August, 2025

SUPERVISOR RECOMMENDATION

It is my pleasure to recommend that this project work report titled **"PROSTRIKE GEAR – Sports Product Recommendation using Content Based Filtering"** is prepared under my supervision by **Swarup Dahal** in partial fulfillments of the requirement of the degree of Bachelor of Science in Computer Science and Information Technology (BSc. CSIT) Their report is satisfactory to process for the future evaluation

………………………………

Mr. Dinesh Khadka

Supervisor

National College of Computer Studies

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the Department of Computer Science and Information Technology for acceptance, a project report entitled " PROSTRIKE GEAR – Sports Product Recommendation using Content Based Filtering " submitted by Swarup Dahal [28896/078] in partial fulfillment for the degree of Bachelor of Science in Computer Science & Information Technology.

……………………..

Dinesh Khadka

Project Supervisor

………………………

Prajwal Baniya Chhetri

Program Director

……………………….

External Examiner

ACKNOWLEDGEMENT

I would like to thank every individual and organization who contributed to the completion of this project. I would like to express our gratitude to NCCS College for providing the opportunity to showcase our learnings into this project

First of all, I would like to thank our project supervisor Mr Dinesh Khadka for his invaluable guidance, encouragement and support throughout the project. His expertise, suggestions and constructive criticism are important in determining the direction and quality of work

Finally, I would like to thank all our friends and those who helped in the successful completion of our project. This project has given us wonderful experience where I learnt about the implementation of our academic learnings into our work.

With respect,

Swarup Dahal

ABSTRACT

This project introduces PROSTRIKE GEAR a content-based sports product recommendation system designed to improve the online shopping experience by delivering personalized suggestions, with a primary focus on cricket and football gear. Using a publicly available dataset sourced from Kaggle, the system processes a wide range of sports products including bats, jerseys, shoes, gloves, and accessories.

To generate recommendations, TF-IDF (Term Frequency-Inverse Document Frequency) vectorization is used to transform product descriptions into numerical feature vectors that capture important textual characteristics. Cosine similarity is then applied to compute the similarity between products, enabling the system to recommend items that closely match a user's interests based on product content alone.

Unlike collaborative filtering approaches that rely heavily on user behavior data, this content-based method makes recommendations using only product attributes. This allows it to function effectively even when user interaction data is sparse or unavailable. The model is fine-tuned and validated to ensure both relevance and diversity in the recommendation results. By focusing on item similarity, PROSTRIKE GEAR provides accurate, meaningful, and personalized sports product recommendations—especially in the cricket and football domains—offering users a more engaging and tailored shopping experience.

KEYWORDS: *Content-Based Filtering, Cosine Similarity, TF-IDF, Cricket Gear, Football Equipment, Kaggle Dataset, Sports Product Recommendation, Personalized E-Commerce*

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
CS	Cosine Similarity
CSS	Cascading Style Sheet
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JS	JavaScript
SQL	Structured Query Language
TF-IDF	Term Frequency - Inverse Document Frequency
UI	User Interface
URL	Uniform Resource Locator

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Introduction

Online shopping has made it easy to browse and buy products from anywhere—but with thousands of items, especially in sports categories like cricket and football, finding the right product can still be overwhelming. Most recommendation systems rely on collaborative filtering, which uses past user behavior to suggest products. While helpful, this approach struggles with new users and less popular items due to a lack of data.

To solve this, we developed PROSTRIKE GEAR a content-based recommendation system that suggests sports products by analyzing their actual features, such as names, descriptions, and categories. Instead of depending on user history, it compares product content using TF-IDF for text analysis and cosine similarity to measure how closely items are related.

Built with Python, Flask, and JavaScript, the system is designed to deliver personalized product suggestions even when user interaction data is limited. This makes it ideal for recommending niche or newly added gear, especially in cricket and football.

By focusing on what makes each product unique, PROSTRIKE GEAR offers users smarter, more relevant recommendations enhancing the overall shopping experience with minimal effort.

1.2 Problem Statement

In today's crowded e-commerce space, users often struggle to find the right products due to an overwhelming number of options and a lack of personalization. Concerns around privacy, along with generic shopping experiences, can reduce user trust and engagement. On the business side, poor inventory tracking and missed recommendations may lead to lost sales.

This project aims to solve these challenges by building a user-friendly sports e-commerce platform PROSTRIKE GEAR that uses a content-based recommendation system. By analyzing product features and aligning them with user preferences, the system delivers personalized suggestions that simplify product discovery and boost both user satisfaction and business growth.

1.3 Objectives

The primary objectives of implementing this system are:

- To build a responsive and secure web platform using HTML, CSS, JS, Python, and Flask.
- To integrate a TF-IDF vectorizer and cosine similarity-based recommendation system.
- To implement full CRUD operations for users and products.

1.4 Scope and Limitation

1.4.1. Scope

1. Collect and organize product information, including names, descriptions, prices, and images.
2. Extract meaningful features from product data using techniques like TF-IDF for text analysis.
3. Compute the similarity between products using algorithms such as cosine similarity.

4. Capture user input, such as search queries or recently viewed products, to personalize recommendations.
5. Generate product recommendations by matching user input with the similarity matrix.
6. Display the recommended products dynamically on the website's frontend
7. Manage and serve static content, including product images, CSS, and JavaScript files.
8. Store product details and pre-computed similarity matrices in the database for quick retrieval.

1.4.2. Limitations

The limitations of PROSTRIKE GEAR are as follows:

1. The report is based on best practices for e-commerce websites in general and may not account for highly specialized technologies or platforms used by large enterprises
2. The report does not provide in-depth financial analysis or cost-benefit assessments for implementing recommendations
3. The report does not involve direct testing or evaluation of live website.
4. The Dataset are to limited products only i.e. mainly focused on cricket and football item.

1.5 Development Methodology

In this project waterfall method has been used as the developmental method. Each phase is completed before the next phase can begin and there is no overlapping in the phases. The waterfall model follows the following procedure.

i. **Planning:**

   In the initial phase, the requirements of the HamroPasal website are gathered as per the trends of any general e-commerce website. The requirements of the selected topic were specified, relevant documents were collected, and a literature review and background study of similar systems were conducted. Consumers and their tentative shopping patterns are kept as main focus for the analysis of project's requirements

ii. **Analysis:**

Based on the gathered requirements, the feasibility of the E-Commerce system using content-based filtering was assessed. The requirement was analyzed, and constraints that could affect the schedule were identified. The work was divided, and guidance was sought from the mentor on how to proceed with the project.

iii. **Design:**

In this phase, all necessary designs were developed to ensure the system could be built without issues during the development phase. The system architecture was defined, and the design of the user interface, database schema, and data flows was initiated

During data modeling, 'sports_product_dataset.csv' was chosen from Kaggle as the basis for content-based filtering system. The dataset contains more than 10 columns for the product that is used for item filtration.

iv. **Implementation:**

Firstly dataset were refined and cleaned. Then algorithm was implemented on the dataset. Then the design of system was started.

After completing the design phase, frontend development and model development were initiated in parallel. Once the frontend was completed, backend implementation began to integrate the components seamlessly.

v. **Testing**

In the testing phase, the system underwent unit and system testing to ensure reliability and functionality Unit test focused on individual components, such as the recommendation function for generating appropriate item suggestions based on the single product content System testing validated the integration of components, ensuring the end-to-end process, from single product selection to similar product recommendation worked seamlessly Various scenarios were tested to ensure the system performed accurately under different conditions, making it robust and ready for deployment.

vi. Documentation:

The project report for the final year project was prepared, documenting all the completed aspects of the project in detail.

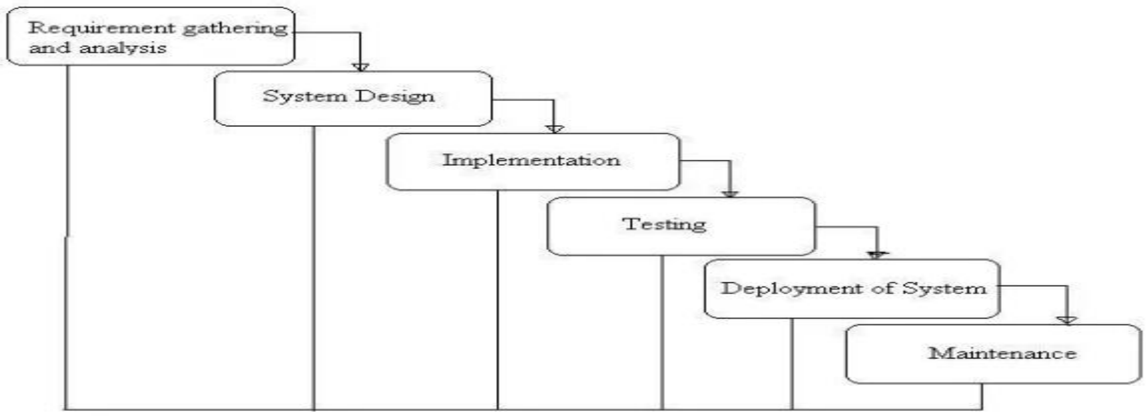


Figure 1.1.1: Waterfall Model

1.6 Report Organization

The report organization on “PROSTRIKE GEAR” comprises of six chapters

Chapter 1:

Introduction to the project, its objectives, limitations and scope, methodology and report organization.

Chapter 2:

Presents background theories, concepts, and previous works related to project.

Chapter 3:

It is an analysis part which includes system analysis, requirement analysis and feasibility analysis.

Chapter 4:

It includes overall design of system

Chapter 5:

Discusses the output of the model, implementation and testing evaluates its performance using accuracy and other metrics, and analyzes how well the system meets its objectives.

Chapter 6:

Concludes the project by summarizing key achievements and outlines potential improvements and additional features for future development.

CHAPTER 2
BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

The e-commerce industry revolutionized shopping and doing business by placing itself at the center of modern commerce. This expanding growth has some problems with regard to customer satisfaction. Of these problems is the huge amount of products provided, which floors users in finding their wanted items quickly. Personalized shopping experience seems very important in order to treat individual preferences with respect and, consequently, breed loyalty Hence, recommendation systems are claimed to be a very important area of interest for improving e-commerce platforms

The art of recommendation systems has come a long way. Early systems were based on cataloged and static data, basic search functionality with extremely limited behavior. Then, at the end of the '90s, collaborative filtering was introduced, which depended on making recommendations using the user's actions and preferences. Successively, content-based filtering came into play by associating attributes in products and users' profiles, hence making related suggestions to the user. More recent systems use very fine machine learning paradigms with advanced algorithms to churn through very accurate results in terms of recommendation systems. The content-based recommendation system relies on the understanding of the features of an item and the preference of a user for them in making recommendations.

In a content-based system, it always deals with text relevant to the product through category or keyword. Key techniques in this approach would be TF-IDF and Cosine Similarity. They also have many problems of their own, including those of scalability and the 'cold-start' problem when there are new users or items. The alternative of overspecialization will cut down diversity in recommending. These gaps have affected e-commerce in a negative manner. Properly developed content-based recommendation systems, supported by algorithms such as TF-IDF and cosine similarity, among others, will help increase user satisfaction by making accurate and unique suggestions that match personalized needs. The present project will develop a scalable, user-friendly platform to overcome these existing limitations and set a milestone for a personalized shopping experience

2.2 Literature Review

In content based filtering (CBF), behavioral data, such as ratings and purchases of products, are typically not considered. Instead, metadata of items are analyzed in order to recommend items that are similar to the preferences of a specific user. There are other approaches which recommend items based on how similar their content is to the content of other items. User profiles can also be compared to find similar users. The most common methods in content based filtering are adopted from information retrieval and the vector space model, eg. to measure similarity between vectors where each element corresponds to the frequency of a word in the content metadata. The content metadata can be item or profile descriptions. It is standard to use tf-idf (term frequency-inverse document frequency) for these word frequencies, which is a statistic intended to filter out the words that are common in every description. [1]

There have been several proposed topologies for categorizing the different kinds of recommender systems that have emerged. The most common one is probably the one introduced by Balabanovic and Shoham [2], which divides recommender systems into three different categories: collaborative filtering (CF), content based filtering (CBF) and hybrid systems combining the two aforementioned techniques. There are other classifications which further divides collaborative filtering into memory-based and model-based systems. Apart from these, there are mentions in the literature of systems using knowledge based filtering, demographic filtering and utility based systems. [3]

According to [4], content-based methods can increase the coverage of a recommender, ie. increase the number of different products that are recommended and cover more of the product catalogue. However, since the similarity is based on the items' similarity to the reference item, recommenders based on content based filters tend to recommend items that are very similar to the reference one and miss connections such as if you are buying a mouse, you might want to buy a mouse pad or a keyboard as well. The tendency to recommend very similar items is also called over specialization [5]

Filtering based on content suggests elements for users which are practically identical to those that the user had previously chosen or wished First the relationship between the object and its properties are established in the term of the matrix, and then machine similarity based on the features of the contrasted items using different mathematical functions selects the most related items to the target item. The most common feature of similarity is the Modified Coefficient of Cosine, Cosine or Pearson. A high level of prediction can result in strong similarity steps. [6]

Information filtering deals with the delivery of items selected from a large collection that the user is likely to find interesting or useful and can be seen as a classification task. Based on training data a user model is induced that enables the filtering system to classify unseen items into a positive class c (relevant to the user) or a negative class c (irrelevant to the user). The training set consists of the items that the user found interesting. These items form training instances that all have an attribute This attribute specifies the class of the item based on either the rating of the user or on implicit evidence Formally, an item is described as a vector () nX x, x,…..,x, x =1 2 of n components[7]. The components can have binary, nominal or numerical attrributes and are derived from either the content of the items or from information about the users preferences. The task of the learning method is to select a function based on a training set of m input vectors that can classify any item in the collection. The function h() X will either be able to classify an unseen item as positive or negative at once by returning a binary value or return a numerical value. In that case a threshold can be used to determine if the item is relevant or irrelevant to the user. [8]

CHAPTER 3
SYSTEM ANALYSIS

3.1 System Analysis

It collects and interprets facts on identifying the problems and decomposition of a system into its components in order to understand how they interact and how they can be improved. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work to accomplish their purpose.

3.1.1 Requirement Analysis

Requirements analysis determines user expectations for a new or modified product involving, identifying and understanding the needs, objectives, and constraints of the project Requirement analysis is one of the most important processes for determining if a system or software project will be successful or not.

3.1.1.1 Functional Requirement

The functional requirements of PROSTRIKE GEAR are as follows:

i. Search Functionality: The users should be able to search for the products by entering keywords or the partial product name to show the product information. The search bar should provide product suggestions as the user types the product's name
ii. Recommendation Generation: The system should use content-based filtering to generate relevant product recommendations by analyzing product attributes. It uses techniques like TF-IDF and cosine similarity for the similarity between the products
iii. Admin: An admin panel should be available for managing product information, performing CRUD (Create, Read, Update, Delete) operations on items, and monitoring recommendation performance.

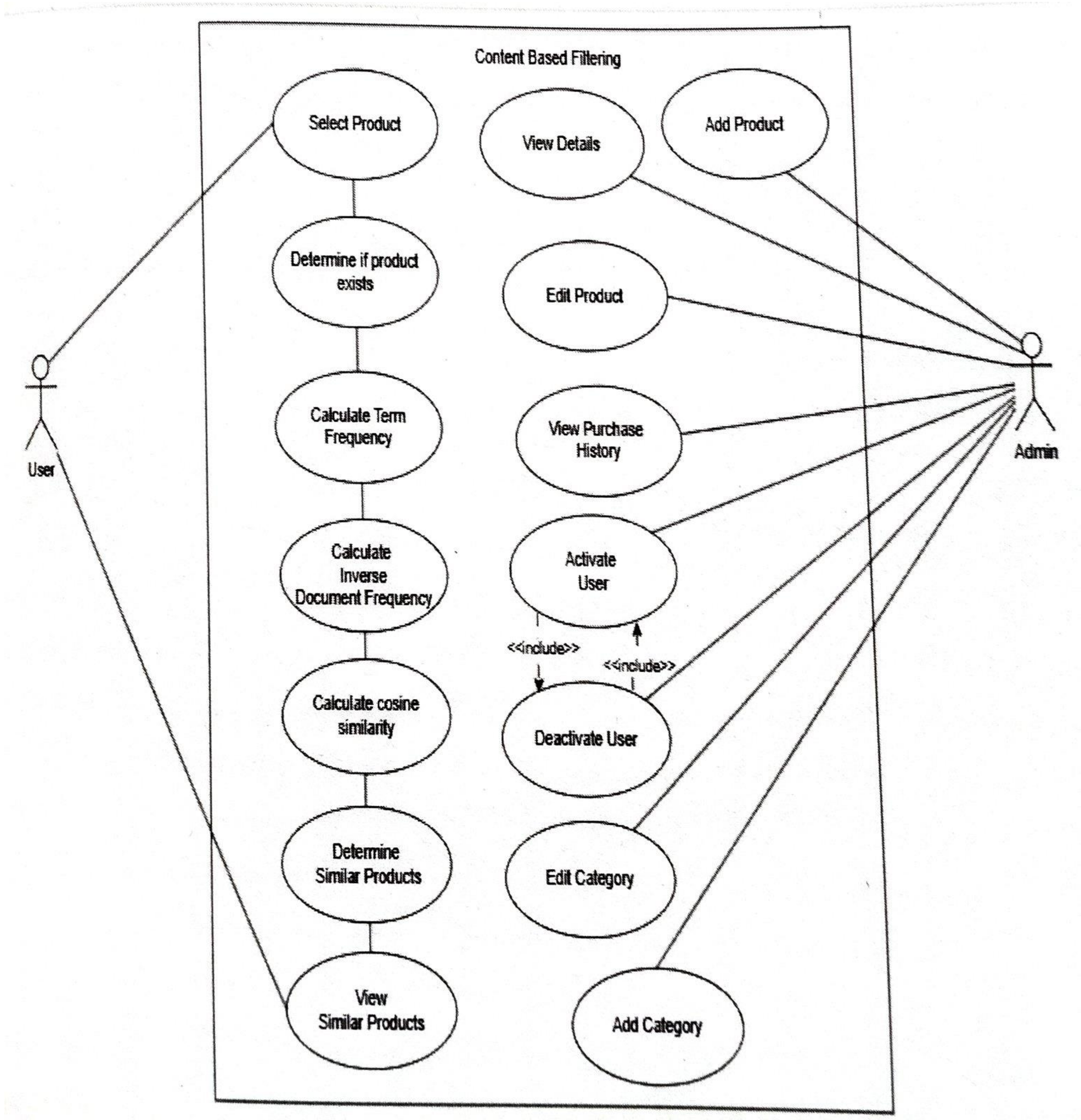


Figure 3.1: Use Case Diagram

3.1.1.2 Non Functional Requirement

The non-functional requirements of PROSTRIKE GEAR are as follows:

1. **Performance** Pages, including recommendations, should load within 2-3 seconds and recommendations only must be generated in under 1 second.
2. **Usability**: The website must have easy navigation and clear product suggestions
3. **Reliability**: The system must be robust and handle errors gracefully.
4. **Security**: The system must protect user data and ensure privacy.
5. **Maintainability**: The system must be designed for easy maintenance and updates.

3.1.2 Feasibility Analysis

1. **Technical feasibility**

The project is technically feasible with a modern, full-stack architecture. HTML5, CSS3, and JavaScript will be utilized for frontend development with responsive design and user interaction. The backend development will be built with Python and Flask as the framework, with the benefits of native support for security, authentication, and database integration. SQLite will be used in the development phase, and PostgreSQL will be the scalable option for deployment.

To serve the purpose of intelligent recommendations, the system will use TF-IDF and cosine similarity to compare user and product vectors of browsing history and personal interests. This recommendation systems algorithm guarantees precise and personalized product recommendations.

2. **Operational feasibility**

The site is designed with ease of operation for both sellers and buyers, with a quick and easy interface. Buyers will be able to search and buy products, and sellers can list and manage their products using an easy-to-use dashboard. An admin panel will provide system administrators with the ability to easily manage users, products, and orders. Role-based access and secure user management provides smooth and trouble-free operation for all user types.

3. **Economic feasibility**

From an academic perspective, for my PROSTRIKE GEAR system project, it is economically feasible because it is cost-effective because it uses all open-source software like Python, Flask, HTML, CSS, and JavaScript, without any licensing cost. It can be built on personal computers with free IDEs. It can be hosted locally or on free services like Render and GitHub Pages. It does not use any paid APIs or services, hence being very cost-effective for an academic project.

4. Schedule feasibility

The PROSTRIKE GEAR project timeline will be planned using a Gantt chart to ensure timely completion. The major phases and estimated durations are:

Table 3.1: Gantt Chart

	Week 1-2	Week 3-4	Week 5-6	Week 7-8	Week 9-10	Week 11-12	Week 13-14	Week 15-16
Requirement Analysis								
System Design								
Implementation(Frontend)								
Implementation(Backend)								
Testing and Debugging								
Deployment								
Documentation								
Final Review and Presentation								

3.1.3 Analysis

3.1.3.1 Data modelling using Class and Object Diagram

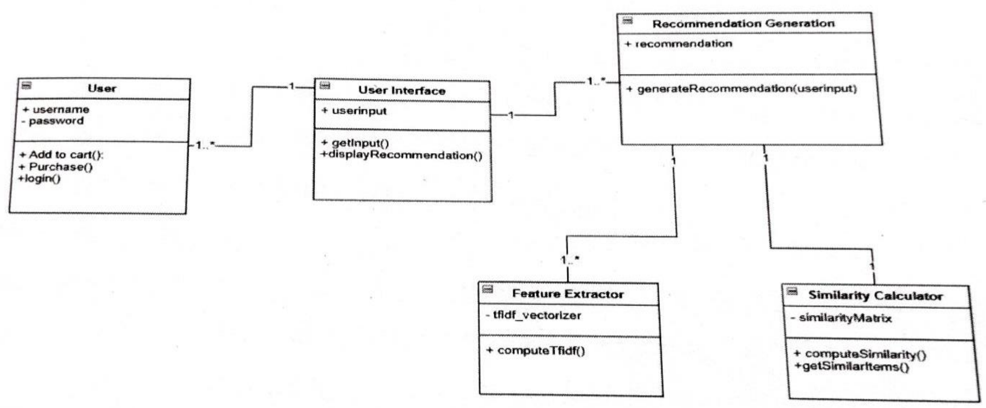


Figure 3.2: Class Diagram for Recommendation system

In the above class diagram of Recommendation Generation, we have 5 classes named as User, User Interface, Recommendation Generation which is categorized to two classes Feature Extractor and Similarity Calculator 'User' has attribute called username and password and "User Interface' consist of attributes like userinput. 'Recommendation Generation' consist of attribute recommendation and its two categorized classes 'Feature Extractor' and 'Similarity Calculator consists of data attributes

User class has following methods as

a) Add_to_cart() enables user to put product in the cart.

b) Purchase (): enables user to purchase the product

c) login(): enables user to login to the website

User Interface consists of following methods as:

a) getInput(): take the product name as an input after user clicks on a product.

b) displayRecommendation (): It shows the recommendation of the product based on the user input.

Recommendation Generation consists of following methods such as

a) generate Recommendation() take the product name as an input after user clicks on a product and generates similar products

Feature Extractor consists of following methods such as

a) compute Tfidf(), take the product name as an input and generate Tf-idf vector

Similarity Calculator consists of following methods such as

a) compute Similarity() calculates cosine similarity between the vectors

b) getSimilarltems() fetches the vectors with high cosine similarity value

3.1.3.2 Dynamic Modeling using Sequence Diagram

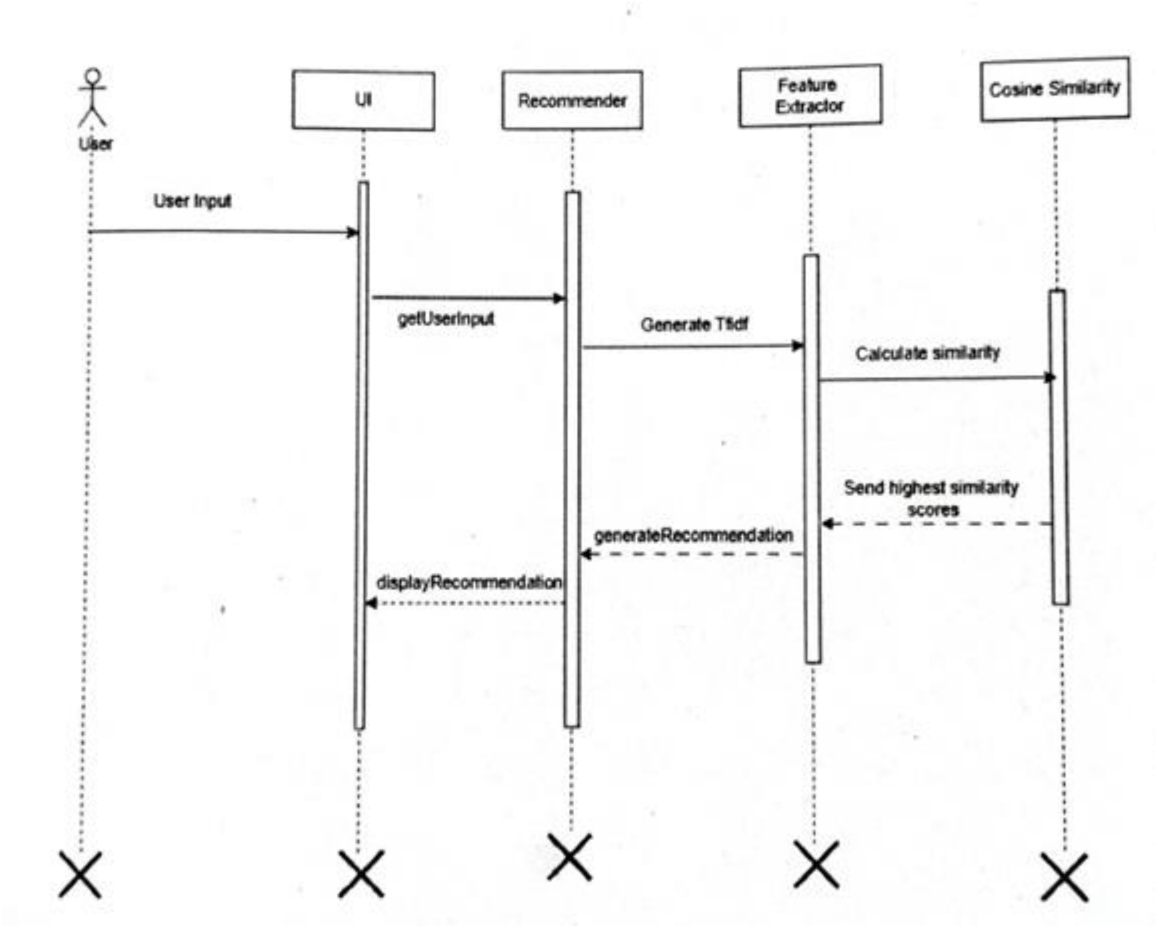


Figure 3.3: Sequence Diagram for Recommendation Generation

In the above sequence diagram, we have five components with their respective lifelines and they are User, UI, Recommender, Feature extractor and Cosine Similarity. The users provide product as an input and then UI display recommended products based on the cosine similarity of the product signal The following products before being displayed into Ul, it undergoes feature extraction method where the features are extracted for the tf-idf vectorization. Then the cosine similarity of the tf-idf is calculated and the product with the higher similarity is displayed to the user is displayed by the UI.

3.1.3.3 Process Modeling using Sequence Diagram

An activity diagram illustrates the flow of activities or actions within a system or process. It visually represents the workflow or business process by depicting the sequence of activities decisions, and transitions from one activity to another.

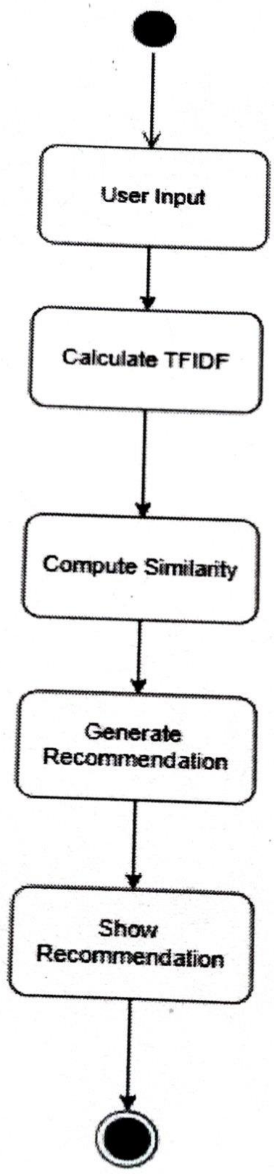


Figure 3.4: Activity Diagram for product recommendation

In the given activity diagram, the process begins with the user entering the name of a product for which they would like related products to be recommended. Then the details of the input product are retrieved along with the dataset that contains all the product names. Afterward, the system calculates the Term Frequency-Inverse Document Frequency (TF-

IDF) values of the input product with each word in a product name in the dataset. These values are then calculated to generate numerical vectors that represent their importance. This implicates that the system calculates the vector similarity by using cosine similarity for the similarity of products. It then ranks products, fetching top results, by their similarity score. The system represents the input product clearly and intuitively and its relation to similar products to enable the user to explore further meaningful recommendations with ease.

CHAPTER 4
SYSTEM DESIGN

4.1 Design

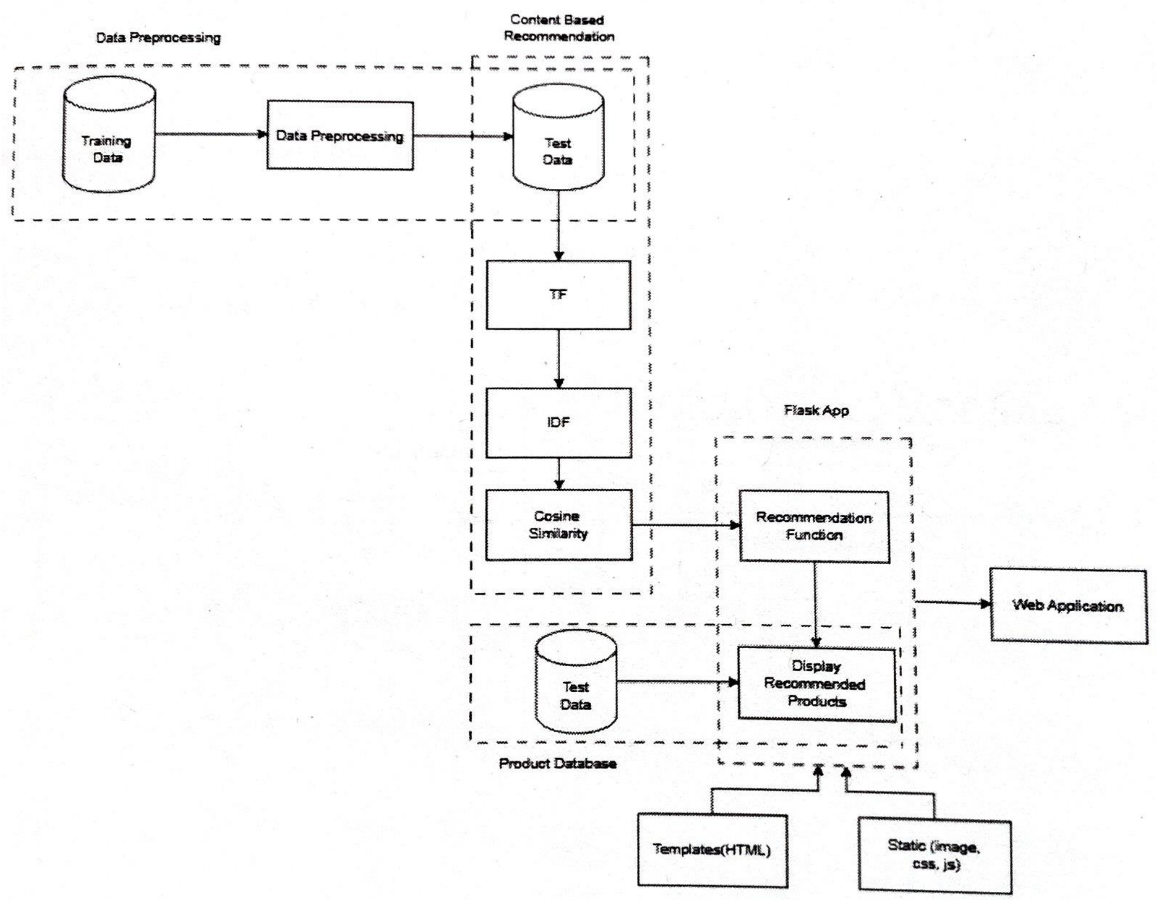


Figure 4.1: System Design of Content Based recommendation System

The diagram illustrates an E-Commerce Recommendation System that uses Content-Based Filtering to suggest products based on user input or previously viewed items. The system begins by analyzing product data, such as descriptions, categories, and images, from the Products Database. Features were extracted using techniques like TF-IDF to represent product descriptions numerically. A Similarity Matrix was then computed using algorithms like Cosine Similarity, which identifies products similar to each other based on their features. The matrix is stored in the database for efficient retrieval.

In the Web Application, user input is captured through search queries or product views. This input is processed by the Recommendation Function, which uses the pre-computed similarity matrix to find and suggest the most relevant products. The system dynamically displays recommended products on the frontend using HTML templates, styled with CSS, and enhanced with JavaScript for interactivity, Static content, such as product images, CSS files, and JavaScript files, ensures a smonth and engaging user experience. This architecture provides a personalized shopping experience by tailoring product recommendations to the user's preferences and interactions.

4.1.1 Refinement of Class Diagram

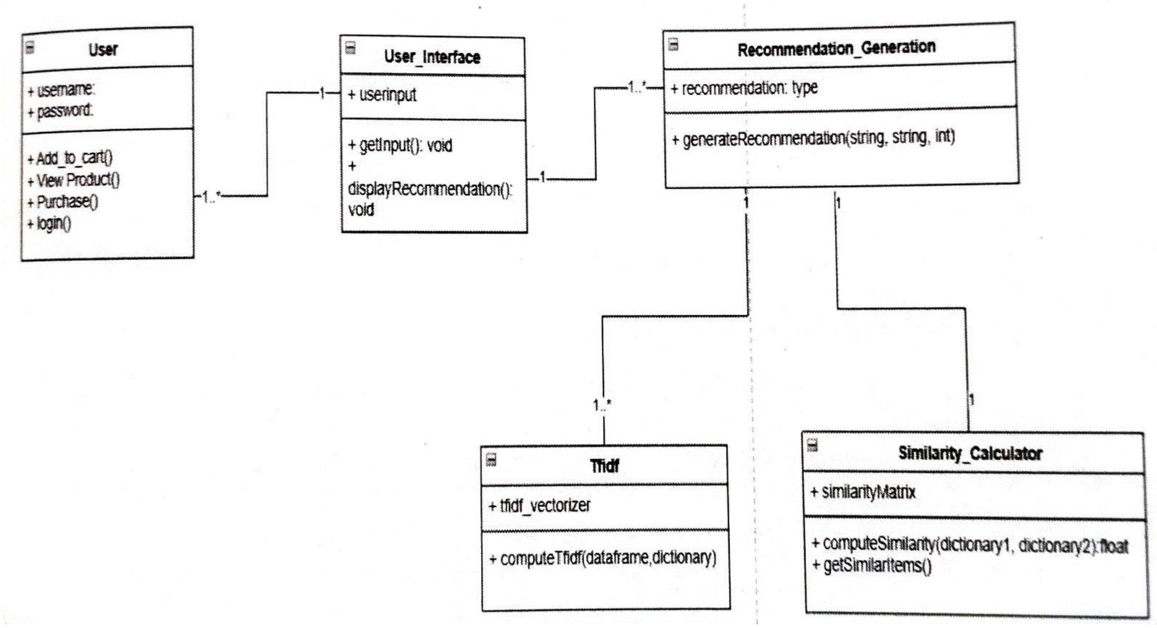


Figure 4.2: Refinement of Class Diagram

The refined class diagram is similar to above class diagram with addition of methods to give detailed view In Refined Class Diagram, we have 5 classes named as User, User_Interface, Recommendation Generation which is categorized to two classes Tf-idf and Similarity Calculator. User has attributes called username and password and User_Interface consists of attribute userinput Recommendation_Generation consist of attribute recommendation type and its two categorized classes tf-idf and Similarity_Calculator consists of data attributes.

User class has following methods as

a) Add to cart (): enables user to add products to cart in order to purchase.

b) View Product (): enables user to analyze the products added to the cart

c) Purchase(); enables user to buy the products in the cart

4) Login() enables users to add their credentials,

User Interface consists of following methods as

a) userinput(): It helps to record the products that are selected by the users in order to determine similarity among the products.

b) getinput(): This function takes the product name as an input after user clicks on a product.

c) display_recommendation() It shows the reconumendation of the product based on the user input.

Recommendation_Generation consists of method generate Recommendation(): take the product name as an input after user clicks on a product and generates similar products.

Meanwhile its two subclasses Tf-idf take the product name as an input and generate Tf-idf vector and Similarity_Calculator consists of following methods such as

a) compute Similarity() calculates cosine similarity between the vectors

b) getSimilarltems(): fetches the vectors with high cosine similarity value

4.1.2 Refinement of Sequence Diagram

A refined sequence diagram illustrates the interactions and flow of messages between objects or components in a system in a more detailed and specific manner.

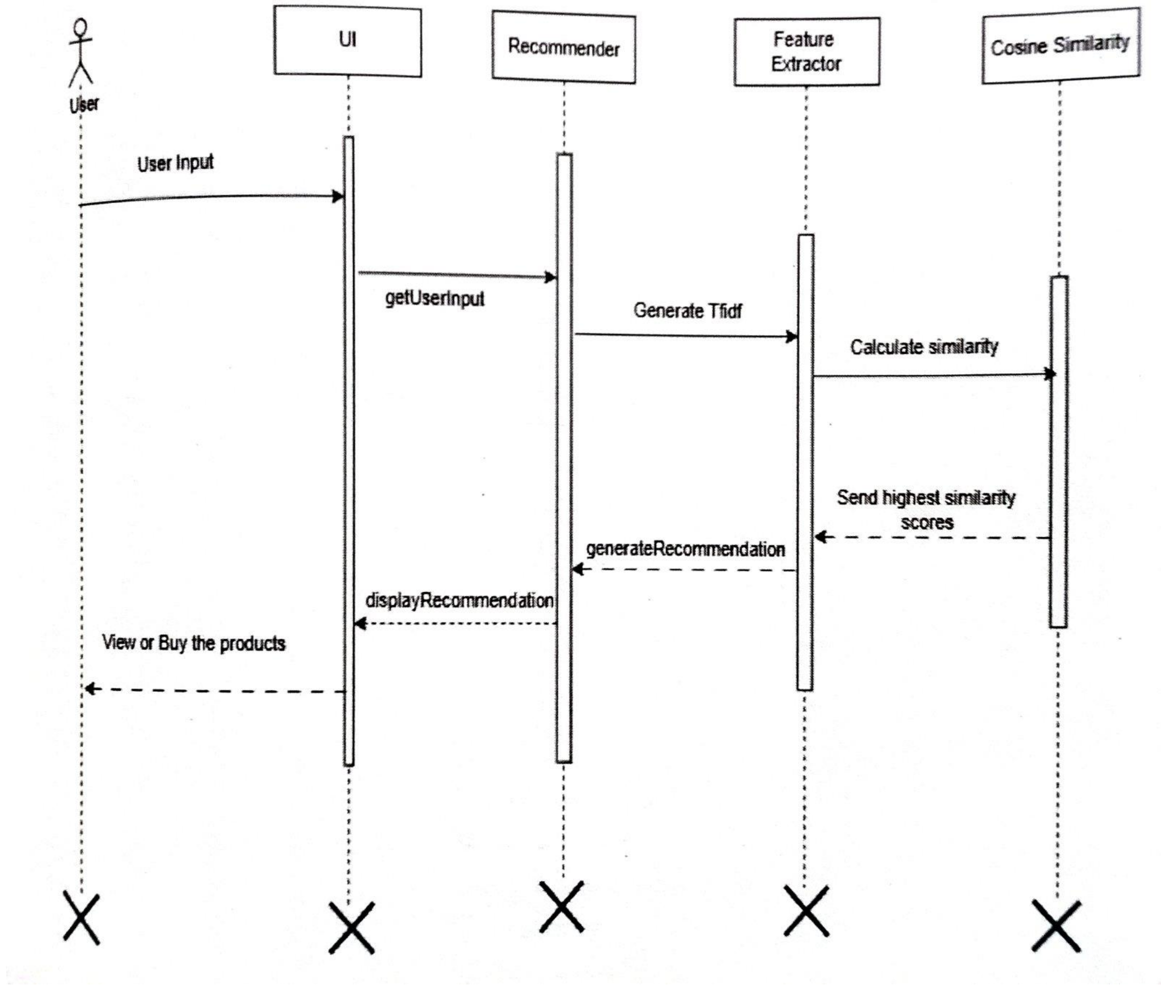


Figure 4.3: Refinement of Sequence Diagram

In above refined sequence diagram, we have five components: User, UL, Recommender, Feature Extractor and Cosine Similarity The users provides product as an input and then UI display recommended products based on the cosine similarity of the product signal. The following products before being displayed into UI, it undergoes feature extraction method where the features are extracted for the if-idf vectorization. Then the cosine similarity of the tf-idf is calculated and the product with the higher similarity is displayed to the user is displayed by the UL. The data after undergoing Cosine Similarity Calculation process, the highest similarity scores are sent to the Recommender which generates recommendations. These recommendations are displayed in the Ul which enables users to view or buy such recommended products.

4.1.3 Refinement of Activity Diagram

A refined activity diagram provides more illustrations

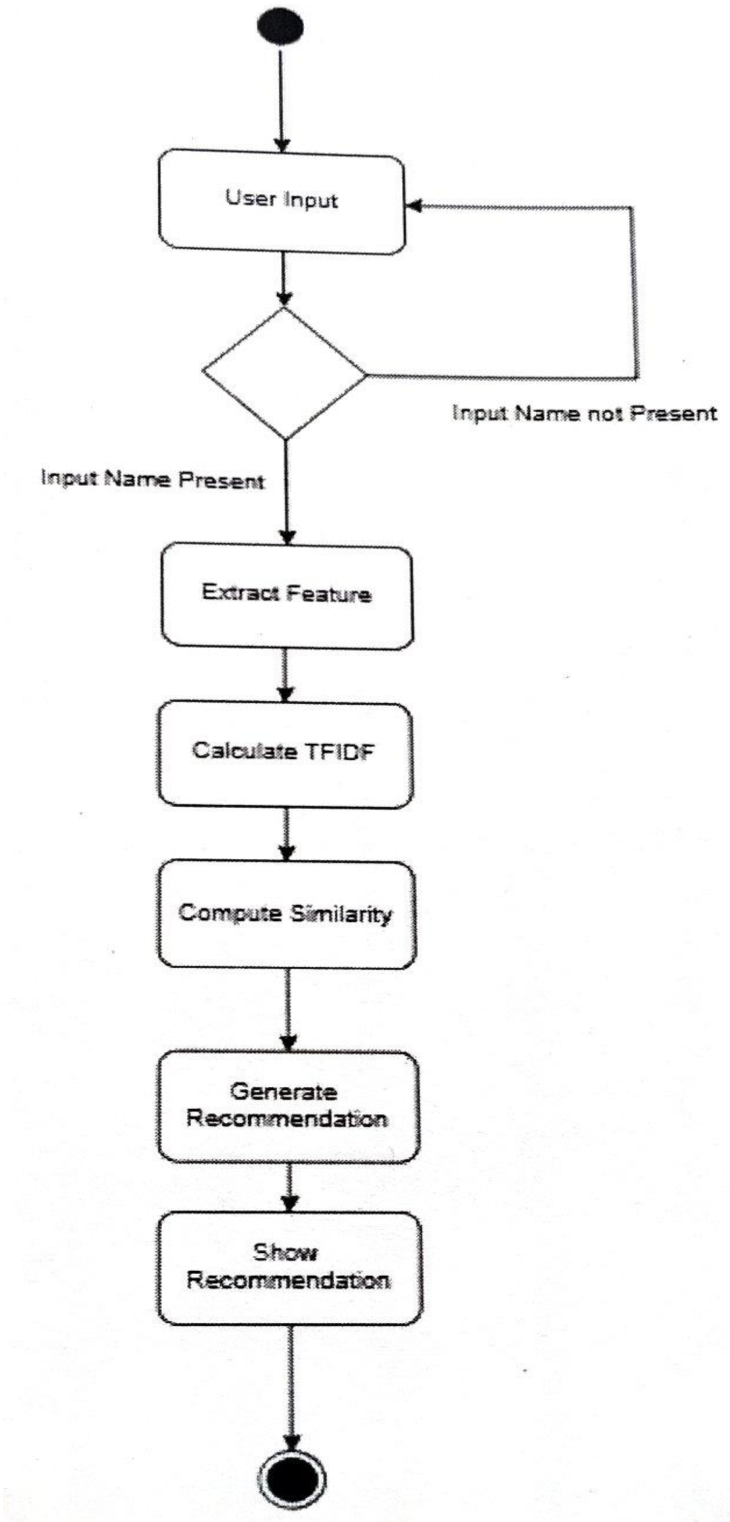


Figure 4.4: Refined Activity Diagram

4.2 Algorithm Details

The algorithm details for the PROSTRIKE GEAR project are as follows:

1. Content-Based Filtering

Content-based technique is a domain-dependent algorithm and it emphasizes more on the analysis of the attributes of items in order to generate predictions. When documents such as web pages, publications and news are to be recommended, content-based filtering technique is the most successful. In content-based filtering technique, recommendation is made based on the user profiles using features extracted from the content of the items the user has evaluated in the past. Items that are mostly related to the positively rated items are recommended to the user. CBF uses different types of models to find similarity between documents in order to generate meaningful recommendations. It could use Vector Space Model such as Term Frequency Inverse Document Frequency (TF/IDF) or Probabilistic models such as Naïve Bayes Classifier, to model the relationship between different documents within a corpus. These techniques make recommendations by learning the underlying model with either statistical analysis or machine learning techniques. Content-based filtering technique does not need the profile of other users since they do not influence recommendation. Also, if the user profile changes, CBF technique still has the potential to adjust its recommendations within a very short period of time. The major disadvantage of this technique is the need to have an in-depth knowledge and description of the features of the items in the profile.

i. Term Frequency-Inverse Document Frequency (TF-IDF)

Tags are handled by TF-IDF vectorization, as one of the more robust ways to analyze text that changes it into a more relevant numerical representation for both machine learning and recommendation systems. What TF-IDF does is that high weights are assigned to terms that show up often in a given item's content but not across the whole dataset. This will, therefore weigh a lot on the most distinct and meaningful terms. In this regard, the method would ensure capturing all the unique features in each item so as to make more accurate recommendations in terms of relevancy. TF-IDF would emphasize those terms that add more value in terms of being able to tell one item apart from another.

Term Frequency (TF): Term frequency measures how frequently a term 't' appears in a document 'd' It is calculated as:

$$Tf(t,d) = \frac{ft}{Nd}$$

Where

ft: Number of times the term 't' appears in the document d'

Nd: Total number of terms in the document 'd'

Inverse Document Frequency (IDF): Inverse document frequency measures the importance of a term across the entire dataset of documents. It is calculated as:

$$Idf(t) = \log\left(\frac{N}{1+nt}\right)$$

N: Total number of documents in the dataset.

nt: Number of documents that contain the term 't'.

1 is added to nt to prevent division by zero.

TF-IDF Score: The TF-IDF score combines these two measures to determine the importance of a term in a document relative to the entire dataset

TF-IDF(t.d)=Tf(t,d)×Idf(t)

High TF-IDF Score: Terms that occur frequently in a specific document but rarely in other documents are given higher importance.

Low TF-IDF Score: Common terms across documents (e.g. "and," "the") are assigned lower weights, reducing their influence.

ii. Cosine Similarity

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. It measures the cosine of the angle between them. If the vectors are normalized to unit length, the dot product is the cosine similarity.

Product vectors: Any product can be described as a vector (a list of numbers) that reflects important attributes such as tags, categories, and textual data (e.g., product descriptions).

How it works:

Let's treat each product as a unique point in a multi-dimensional vector space, where every dimension corresponds to an attribute (e.g., "sports," "shoes," "red," "running," and so on). Cosine similarity measures the angular distance between two product vectors. A smaller angle, tending towards zero (leading to cosine similarity values close to 1), reflects greater similarity among the product.

Why use it?

This enables the suggestion of products that are similar to the ones a user has viewed or bought, even when there are no exactly similar features. For instance, if a user is looking at a red running shoe, cosine similarity can provide an alternative running shoe in a different color but with the same or similar features.

$$similairty(A, B) = \frac{A \, . \, B}{||A|| \, . \, ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Where Ai and Bi are components of vector A and B respectively. A value of 1 means the vectors are identical, 0 means they are orthogonal (no similarity), and -1 means they are diametrically opposed. In summarization, this helps identify redundant sentences or group similar products together.

CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation

For this project, the process involves essential components such as data collection, preprocessing, feature extraction and classification arranged sequentially.

5.1.1. Tools Used

The different tools and software used in the project are as follows:

- **Platform:** Windows
- **Front end:** HTML, CSS, JAVASCRIPT
- **Back end:** Python, Flask
- **Database:** MySQL
- **IDE:** VS Code
- **Documentation:** Ms Word
- **Diagram Tools:** Draw.io

5.1.2. Implementation Details of Modules

The modules of PROSTRIKE GEAR are as follows:

Datasets: Datasets were collected from kaggle.com named sports_product_dataset containing more than 300 data with more than 10 columns.

i. TF-IDFVectorizer

The compute_tf_idf function calculates the Term Frequency-Inverse Document Frequency (TF-IDF) for a dataset of documents represented by tags. It computes the Term Frequency (TF) by counting the occurrence of terms in each document and normalizing it by the total number of terms. It then calculates the Inverse Document Frequency (IDF) by measuring how often a term appears across all documents. The function returns a TF-IDF matrix, representing the importance of each term in the dataset.

```
def compute_tf_idf(train_data):
    term_freqs = []
    doc_freqs = {}
    num_documents = len(train_data)
    for tags in train_data['Tags'].fillna(''):
        terms = tags.lower().split()
        term_count = {}
        for term in terms:
            term_count[term] = term_count.get(term, 0) + 1
        term_freqs.append(term_count)
        for term in term_count.keys():
            doc_freqs[term] = doc_freqs.get(term, 0) + 1
    tf_idf_matrix = []
    for term_count in term_freqs:
        tf_idf_vector = {}
        for term, count in term_count.items():
            tf = count / len(term_count) if len(term_count) > 0 else 0
            idf = math.log(num_documents / (1 + doc_freqs.get(term, 0))) if num_documents > 0 else 0
            tf_idf_vector[term] = tf * idf
        tf_idf_matrix.append(tf_idf_vector)
    return tf_idf_matrix
```

ii. Cosine Similarity

The cosine_similarity function computes the cosine similarity between two vwctors by calculating the dot product and dividing it by the product of their magnitudes. It returns a value between 0 and 1, representing how similar the vectors are if eithervector has a zero magnitude, it returns 0.

```
def cosine_similarity(vector1, vector2):
    dot_product = sum(vector1.get(term, 0) * vector2.get(term, 0) for term in vector1)
    norm1 = math.sqrt(sum(v ** 2 for v in vector1.values()))
    norm2 = math.sqrt(sum(v ** 2 for v in vector2.values()))
    if norm1 == 0 or norm2 == 0:
        return 0.0
    return dot_product / (norm1 * norm2)
```

iii. Content Based Recommendation

The content based recommendations function generates item recommendations based on content similarity. It first checks if the given tem name exists in the dataset. Then it computes the TF-IDF matrix for the dataset and calculates cosine similarity between the given item's TF-IDF vector and all other items vectors. The function returns the top nmost similar items (default 10), sorted by similarity, and includes product details like ID, Name, Description for the recommended items.

```
def content_based_recommendations(train_data, item_name, top_n=10):
    if item_name not in train_data['Name'].values:
        return pd.DataFrame()

    tf_idf_matrix = compute_tf_idf(train_data)
    item_index = train_data[train_data['Name'] == item_name].index[0]
    similarities = []
    for i, tf_idf_vector in enumerate(tf_idf_matrix):
        similarity = cosine_similarity(tf_idf_matrix[item_index], tf_idf_vector)
        similarities.append((i, similarity))
    similarities = sorted(similarities, key=lambda x: x[1], reverse=True)
    top_similar_indices = [idx for idx, _ in similarities[1:top_n + 1]]
    recommended_items_details = train_data.iloc[top_similar_indices][
        ['ID', 'Name', 'ReviewCount', 'Brand', 'ImageURL', 'Rating', 'Price', 'Description']]
    return recommended_items_details
```

5.2. Testing

5.2.1. Test Cases for Unit Testing

Unit testing tests each individual units or components of a software application to validate each unit of the software performs as expected according to its design and requirements.

Table 5.1: Login/Sign up Button Testing

Test Case ID	Description	Input	Expected result	Actual Result	Status
TC-001	Test Button Functionality	Click on signup/ login button	Signup/ login page must be appeared in the screen	As expected, Signup/Login page is appeared	Passed

Table 5.2: User login Testing

Test Case ID	Description	Input	Expected result	Actual Result	Status
TC-002	Enter invalid username and password	Username: prank Password: pranks	There should show an error in the screen as username is invalid	As expected, error is shown	Passed

Table 5.3: Signup using existing username

Test Case ID	Description	Input	Expected result	Actual Result	Status
TC-003	Enter existing username of database during the signup	Username: SwarupDahal Password: #nepal123#	There should be shown error in the screen as username already exists	As expected, Error message was displayed as username already exists	Passed

Table 5.4: Validate Password Length

Test Case ID	Description	Input	Expected result	Actual Result	Status
TC-004	Password length must be equal or greater than 8	Username: Mihawk Password: swords	Error message should be displayed telling password length must be equal or greater than 8	As expected, Error message is shown	Passed

Table 5.5: Email Validation

Test Case ID	Description	Input	Expected result	Actual Result	Status
TC-005	Email should contain @ and some words followed by it	Email: mihawk123	Pop message should be displayed in email box telling email must contain @ and some characters followed by it	As expected, Message was popped up	Passed

Table 5.6: Logout Button Testing

Test Case ID	Description	Input	Expected result	Actual Result	Status
TC-006	Logout button testing	Click Logout	When logout Button is clicked user should logout and redirect to index page	As expected, User is logged of the webpage	Passed

Table 5.7: Role based testing

Test Case ID	Description	Input	Expected result	Actual Result	Status
TC-007	Role based login	User login/ admin login	When user logins user should redirect to user page and when admin logins admin should redirect to admin page	As expected, User is redirected to user page and admin is redirected to admin page	Passed

Table 5.8: Item Search Testing

Test Case ID	Description	Input	Expected result	Actual Result	Status
TC-008	Item Search functionality	User enters product name in the search bar	When the user enter the item name the item should be displayed and similar products should be recommended	As expected, user is displayed with that product and recommended similar products	Passed

Table 5.9: Category Button testing

Test Case ID	Description	Input	Expected result	Actual Result	Status
TC-009	Display item based on category button	User clicks on the category button to see the products	When user clicks on the category button similar category products should be displayed on the web page	As expected, Similar categories products are displayed on the webpage	Passed

5.2.2. Test Cases for System Testing

Table 5.10: System Testing

Test Case ID	Description	Input	Expected result (Emotion)	Actual Result (Emotion)	Status
STC-001	To show recommendation of the product	Click on the product or recommended product or search the product	Recommendation for the product must be shown	As expected, Recommendation for the products are shown	Passed
STC-002	Add to cart and buy product	Click on product and first add to cart and press checkout	Product bought info should be shown	As expected, product is bought	Passed

STC-003	Admin product CRUD operation	Admin should add, delete, edit the product through the admin panel	The product changes should be reflected on the webpage	As expected, product info is changed and displayed	Passed
STC-004	Product not available message for those products which are not available on database but are displayed	User should click on the displayed product	If product is not available then product not available message should be displayed	As expected, product not available message is displayed	Passed

5.3. Result Analysis

The testing covered key components such as product recommendations, user authentication, cart functionality, and admin operations. While the system performed well under normal conditions, certain edge cases were identified that could benefit from further refinement. The results showed the project was able to meet its goals, but there is still room for improvement in terms of expanding the system's capabilities.

CHAPTER 6
CONCLUSION AND FUTURE RECOMMENDATIONS

6.1. Conclusion

In conclusion, E-commerce website optimization, therefore becomes one of the mandatory courses for companies that are interested in positioning themselves in a swiftly changing digital marketplace. In that respect, the paper has mentioned main areas that include user experience, mobile optimization, performance, security, and personalization, to concentrate on in order to enhance e-commerce sites.

This clearly shows that such optimization of an e-commerce platform is technically and operationally feasible. From an investment perspective, it can also be classified as prudent spending that would pay back in terms of customer retention by avoiding abandonment and eventually in the enhancement of business operations

The technical infrastructure to support these enhancements will be readily available, and companies can deploy modern tools, integrations, and frameworks to improve their online presence. In view of the rapidly rising e-commerce and changing customer needs, businesses need to remain agile and invest in optimizing their digital platforms.

Such recommendations of the given report could act as a roadmap for enhancing the websites of the e-commerce businesses to remain in competition with improved performances and to ensure sustainable growth in the growing digital world.

6.2. Future Recommendations

To maintain competitive advantage and adapt to the ever-evolving e-commerce landscape, businesses should focus on implementing a range of strategic recommendations that address both current trends and future opportunities. Here are several key recommendations for the continued optimization and growth of e-commerce websites.

- **Implement Collaborative and Hybrid Recommendation:**
  The system will implement a Hybrid Recommendation approach, combining Collaborative Filtering based on user preferences with content-based filtering to provide more accurate and personalized product suggestions.

- **Use Sentiment Analysis on User Reviews:**
  Analyze user reviews to determine product sentiment and feature preferences, improving the system's ability to recommend products with positive feedback and matching features

- **Strengthen Cybersecurity Measures:**
  With the increasing volume of online transactions, robust cybersecurity protocols must be implemented to protect customer data. Businesses should adopt multi-factor authentication, advanced encryption, and comply with data protection regulations.

- **Incorporate Chatbots for Customer Service:**
  Deploy Al-powered chatbots to handle customer queries in real-time, offering instant responses and improving the customer service experience. This can also help reduce operational costs by automating repetitive tasks.

REFERENCES

[1] P Lops, G. M. De and G. Semeraro, "Content-based Recommender Systems State of the Art and Trends," Recommender systems handbook, pp 73-105, 2011

[2] Y. Shoham and M. Balabanović, "Content-based, collaborative recommendation," Communications of the ACM pp. 66-72., 1997

[3] VR Talwar and K. N. "Application domain and functional classification of recommender systerns a survey," Desidoc journal of library and information technology vol. 28, p. 17-36, 2008.

[4] B. Sarwar, G Karypis, J. Konstan and J Riedi, "Itembased collaborative filtering recommendation algorithms," In Proceedings of the 10th International Conference on World Wide Web, p. 285-295, 2001

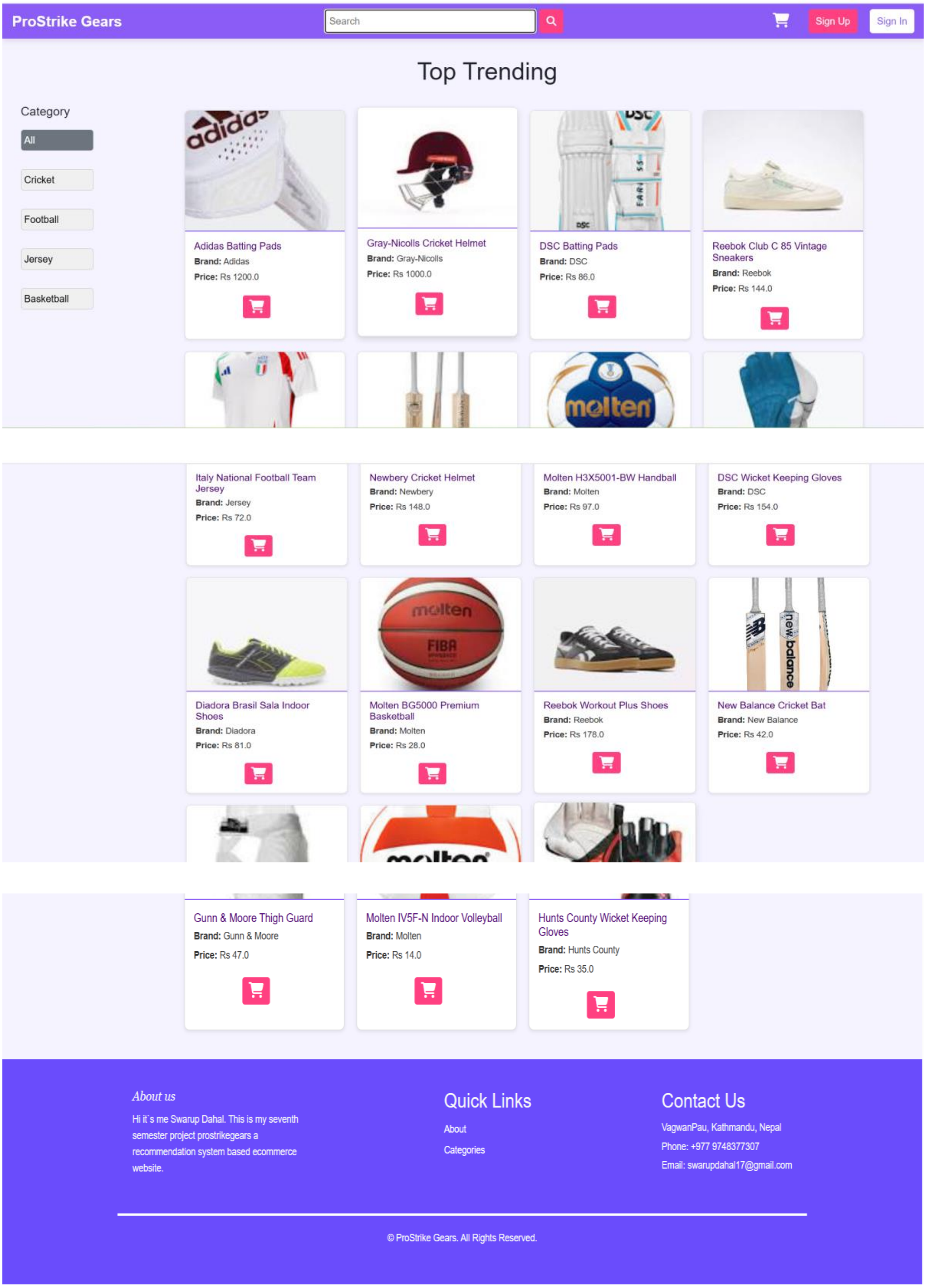
[5] F. Meyer, "Recommender systems in industrial contexts," 2012.

[6] S. H. Nallamala and a. et, "A Brief Analysis of Collaborative and Content Based Filtering Algorithms used in Recommender Systems," IOP Conference Series. Materials Science and Engineering, vol 981, no. 2, 2020.

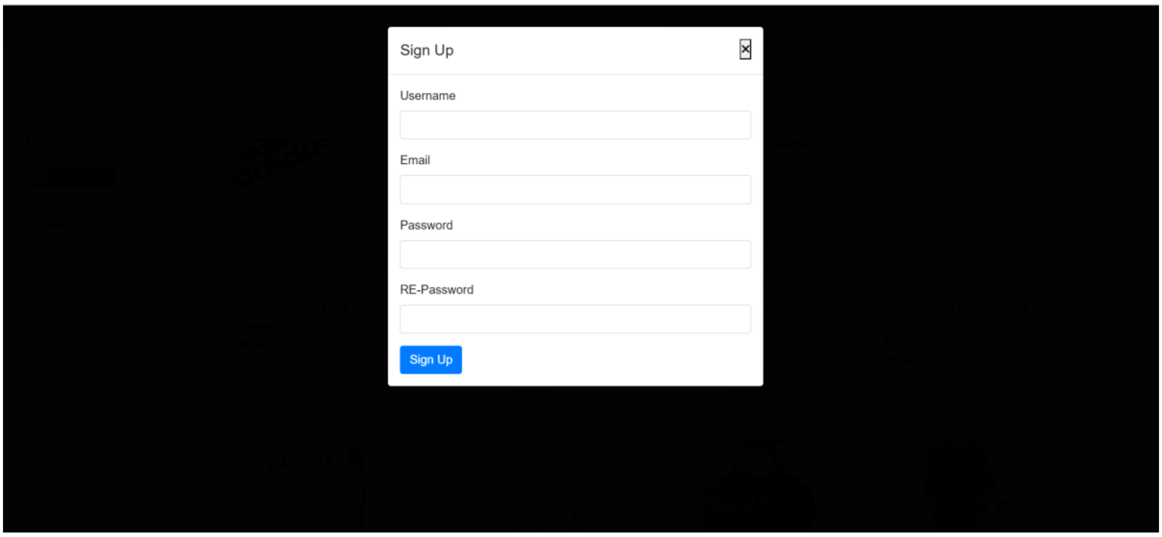
[7] M. Van and V. S. Maarten, "Using content-based filtering for recommendation.." Proceedings of the machine learning in the new information age. MLnet/ECML2000 workshop, vol. 30, 2000.

[8]"bigcommerce.com,"5May 2023. [Online]. Available:
https://www.bigcommerce.com/articles/ecommerce/recommendation-engine/.

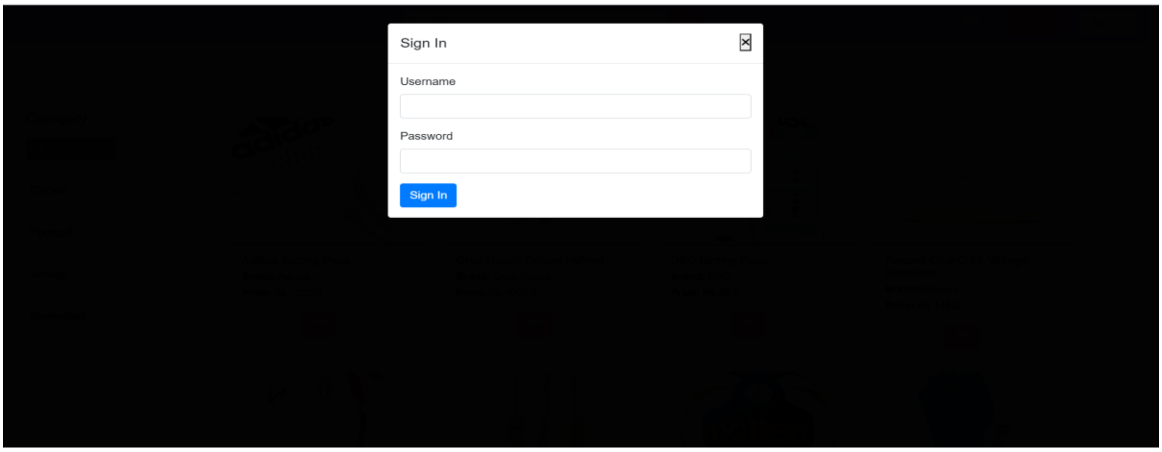
APPENDICES



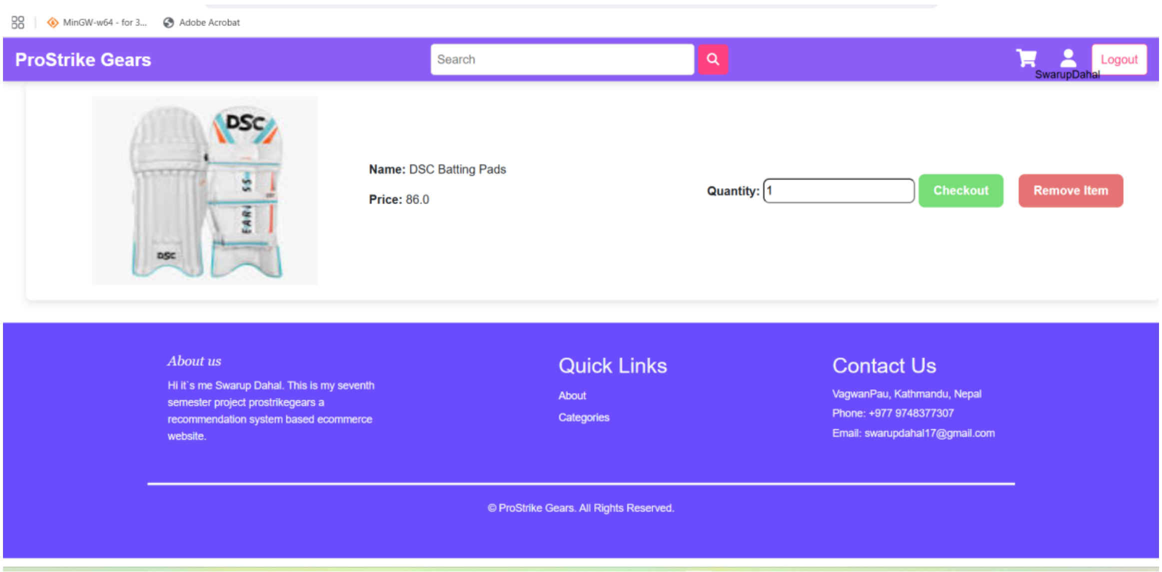
Home Page



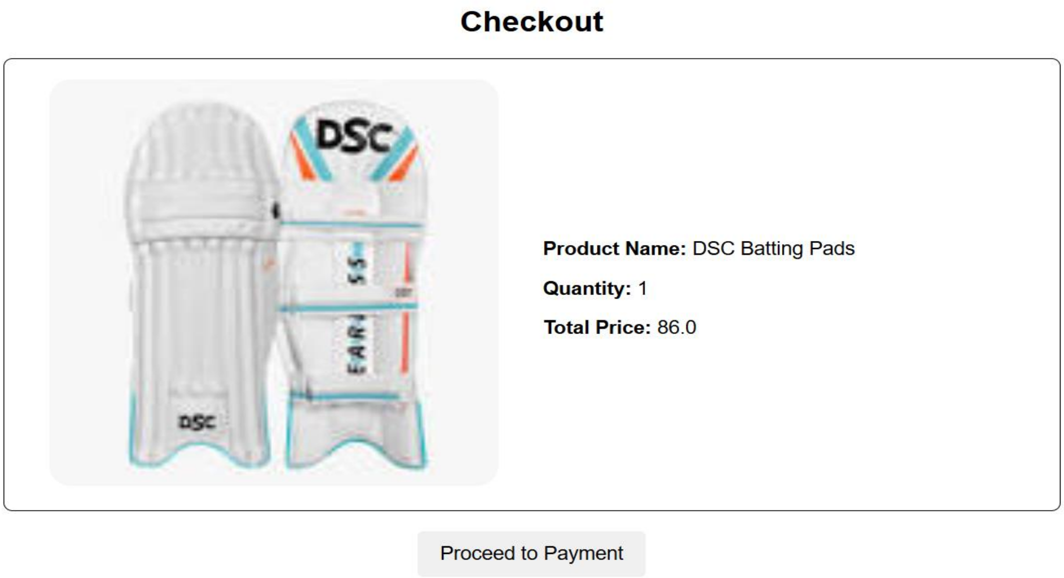
Signup Page



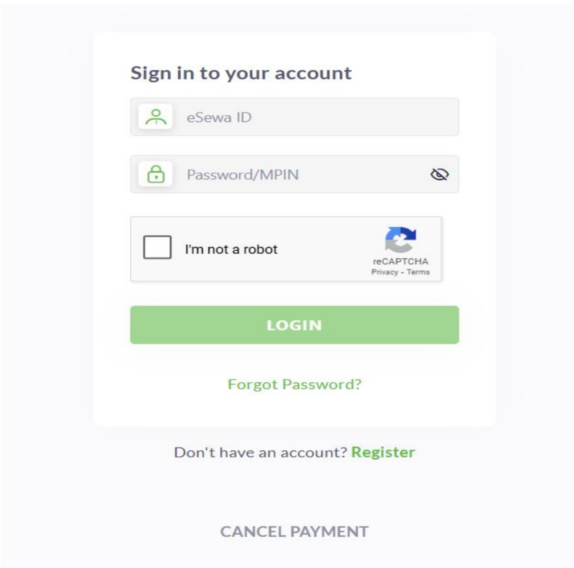
Signin Page

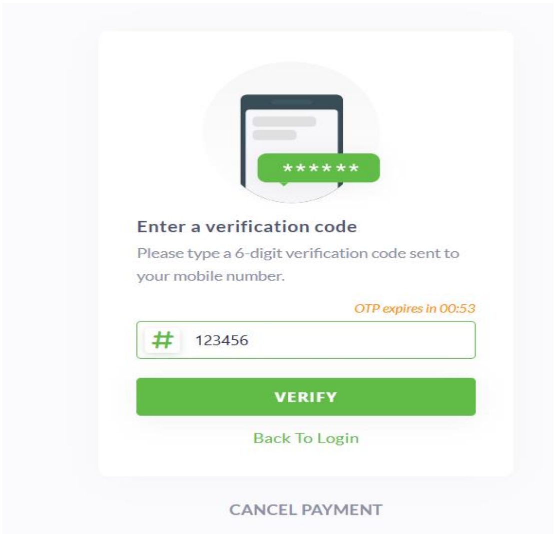


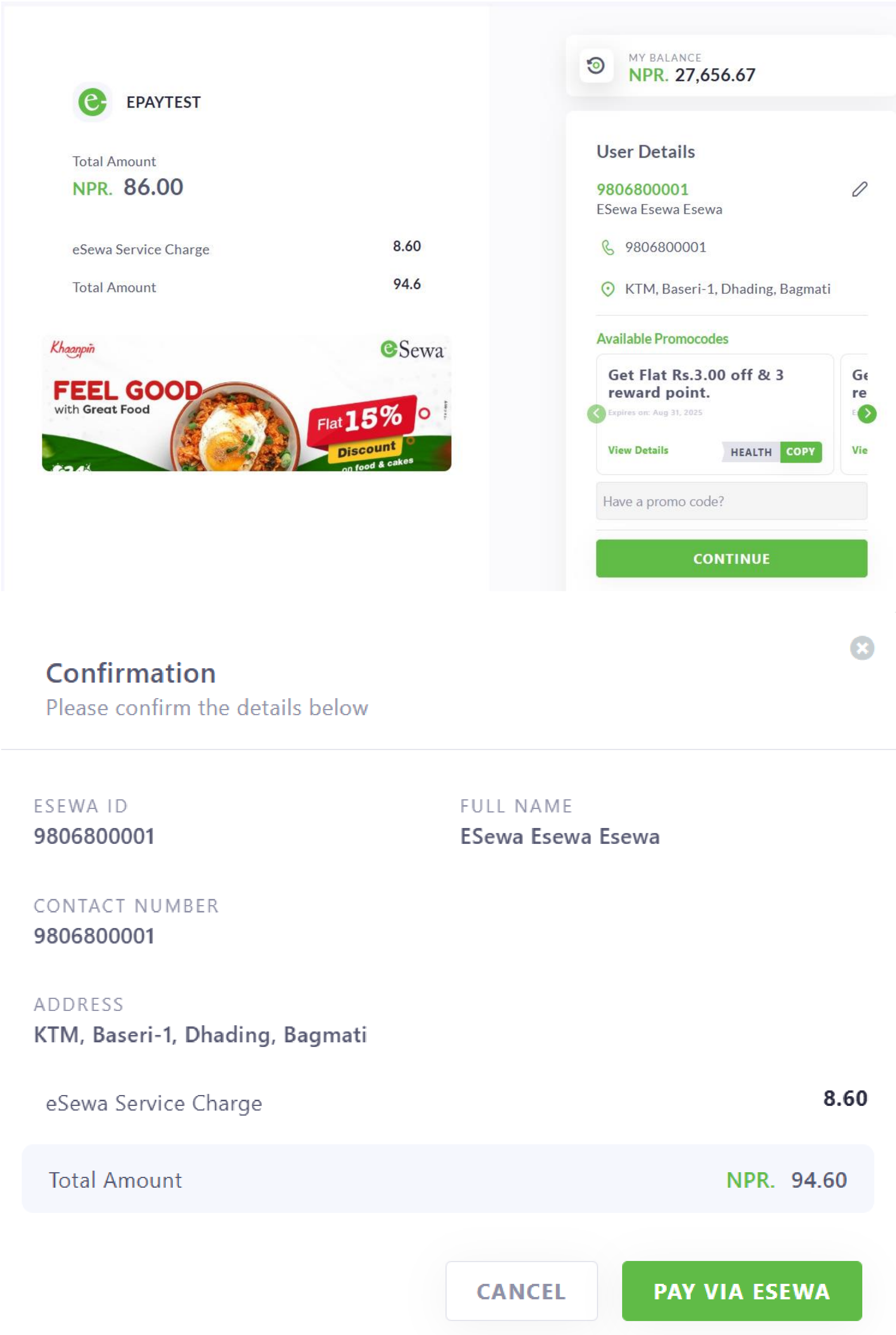
Add to cart Page



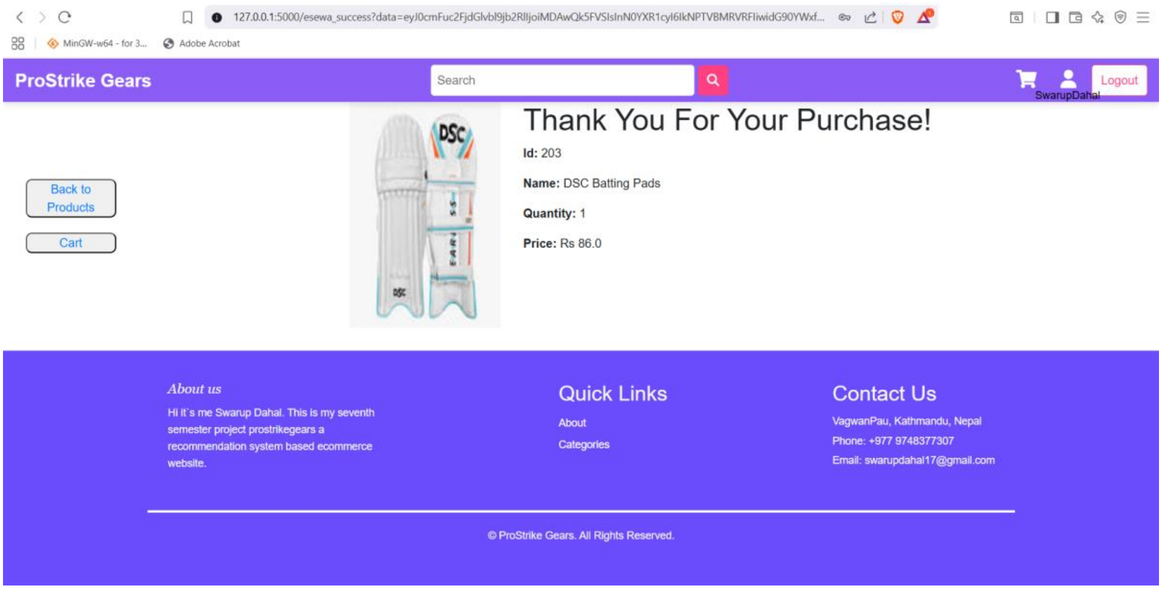
Checkout Page



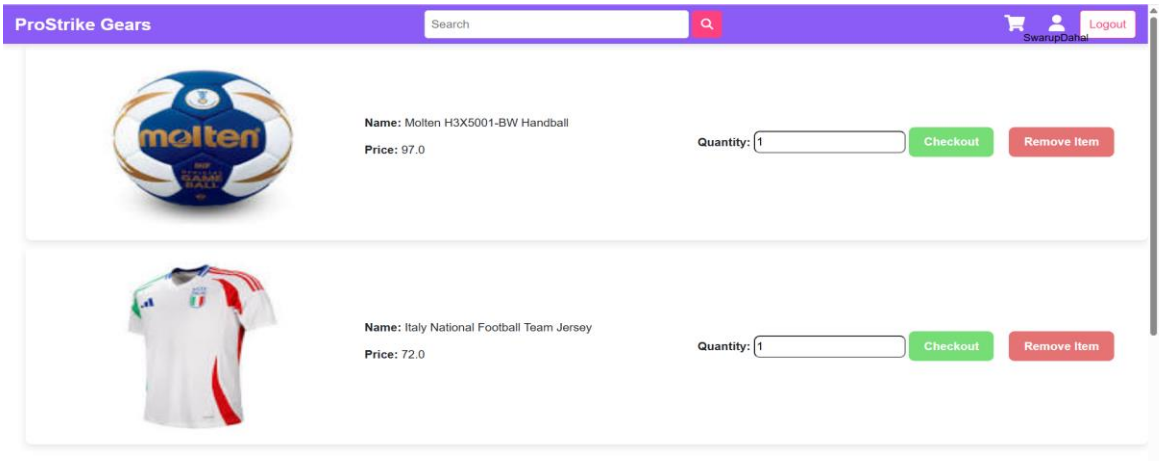




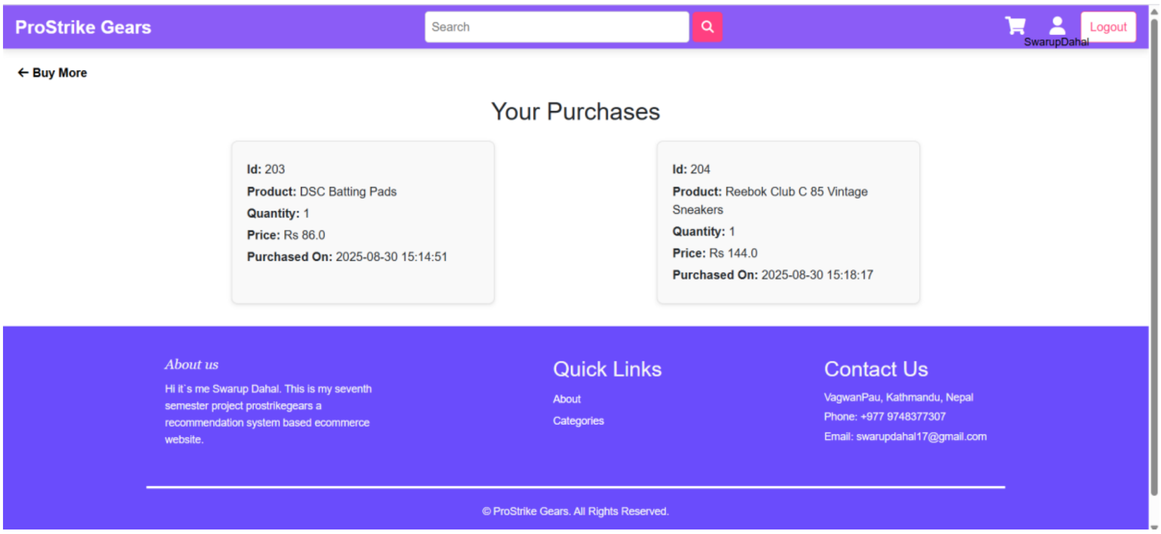
Esewa Payment API



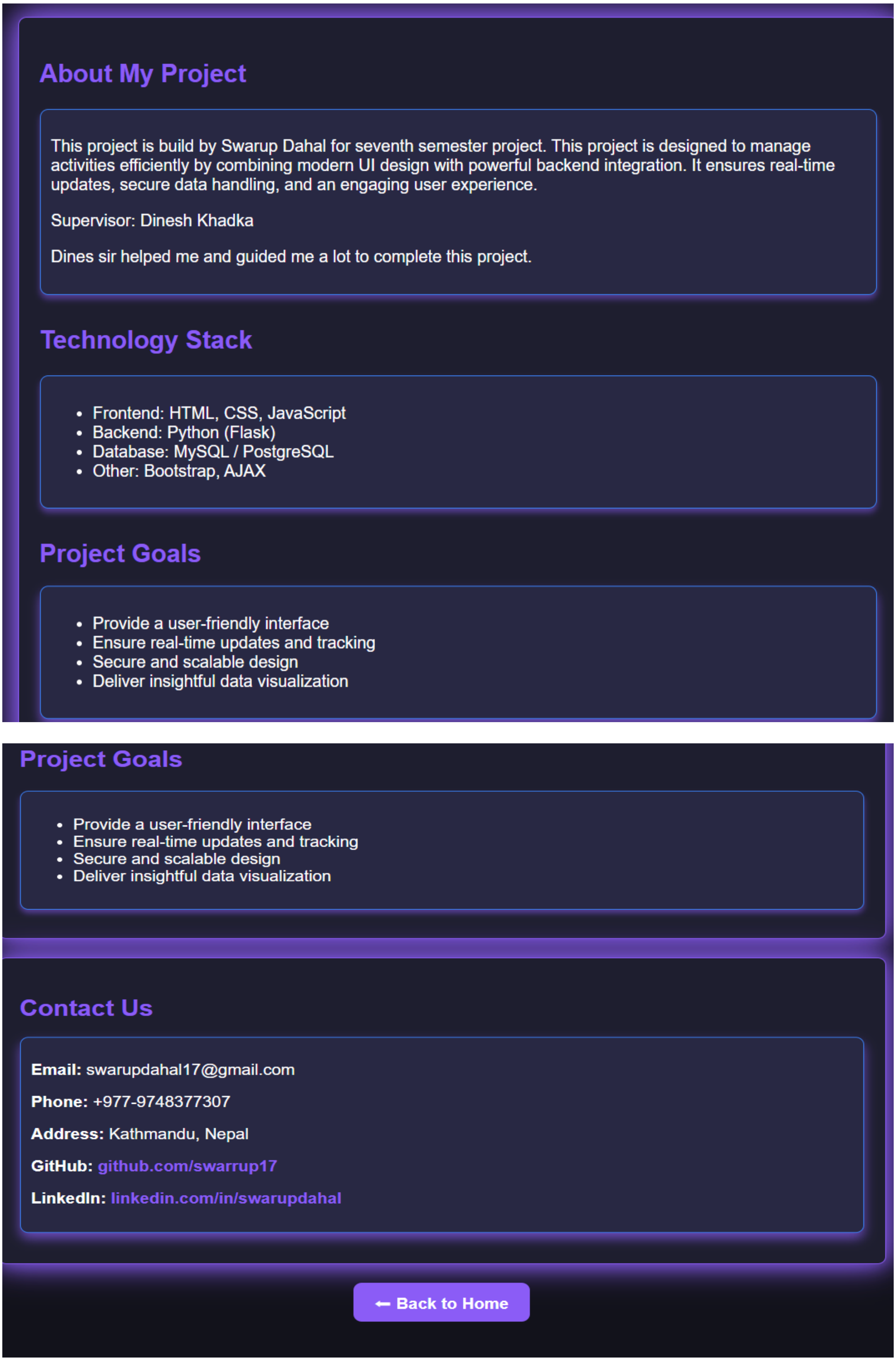
Purchase Success page



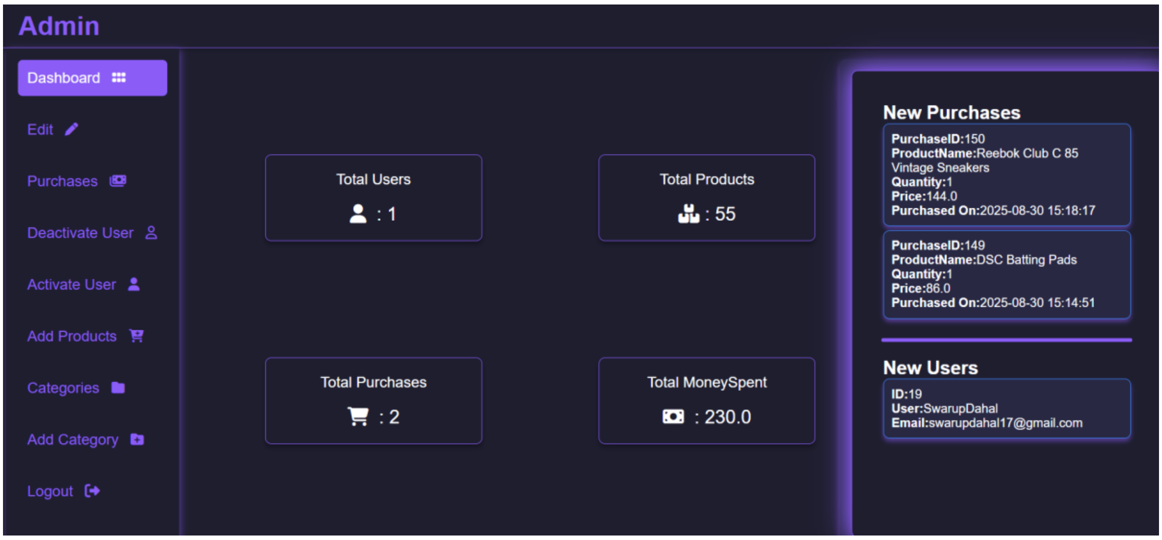
Users Cart Page



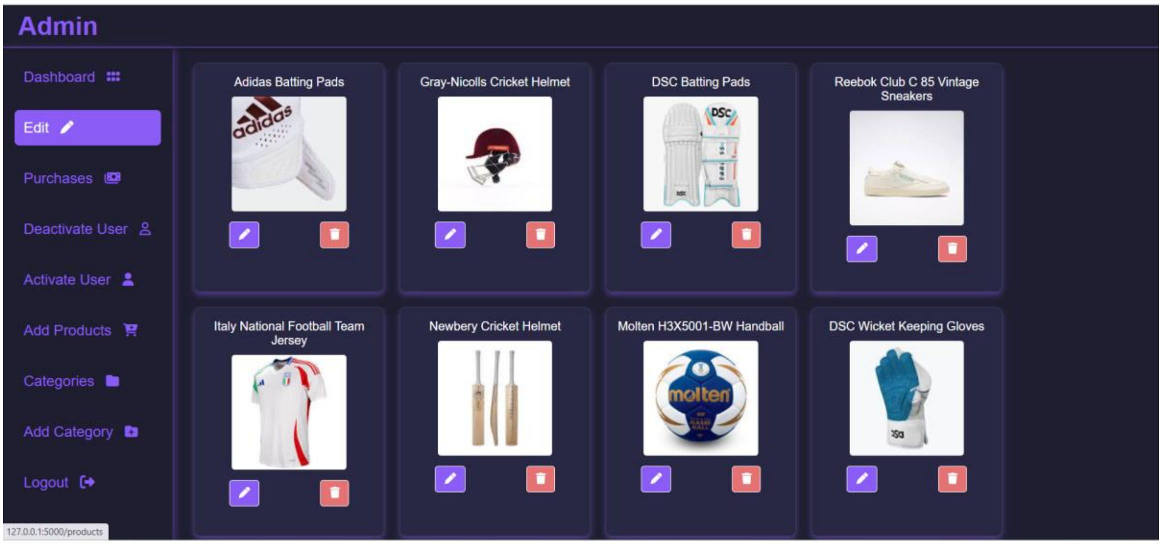
User Purchase page



About Us page

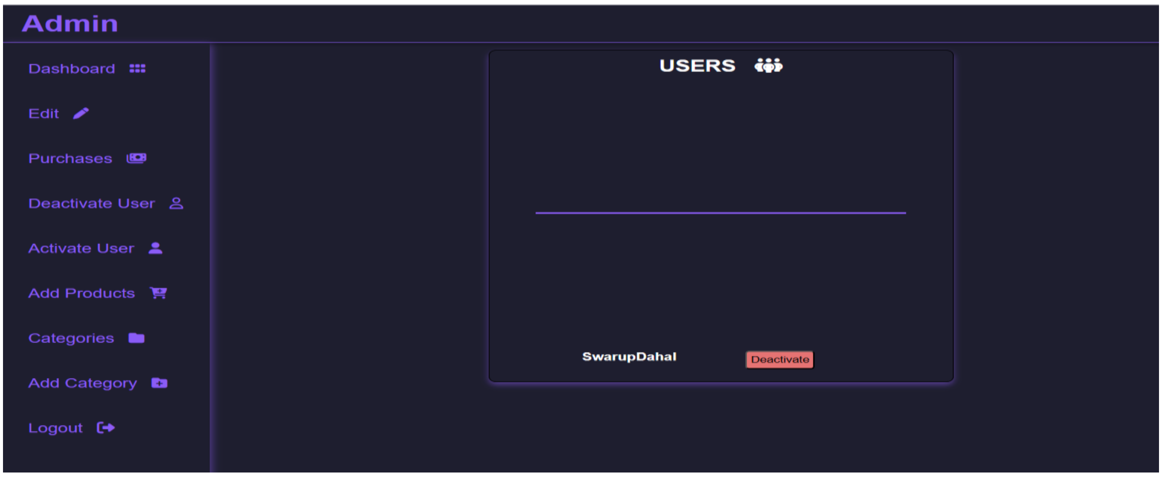


Admin Dashboard

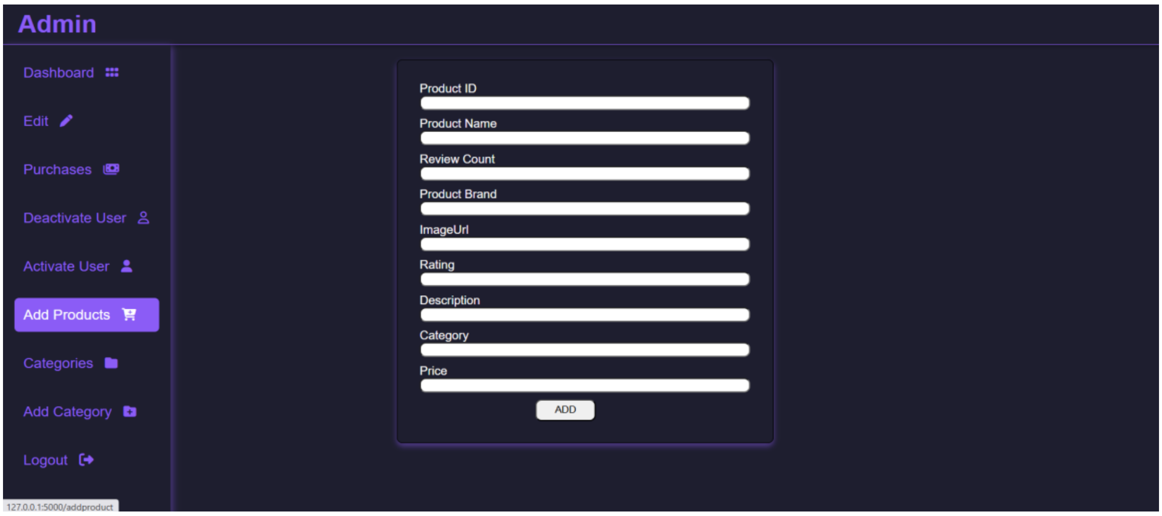


Admin edit panel

User Purchase History



User deactivation page



Add product page

Remove Category page

Table	Action	Rows	Type	Collation	Size	Overhead
admin	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
carts	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
category	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
displayproduct	Browse Structure Search Insert Empty Drop	15	InnoDB	utf8mb4_general_ci	16.0 KiB	-
products	Browse Structure Search Insert Empty Drop	40	InnoDB	utf8mb4_general_ci	16.0 KiB	-
purchase	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
signup	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
7 tables	Sum	66	InnoDB	utf8mb4_general_ci	128.0 KiB	0 B

Database Tables

←T→	id	username	email	password	repassword	status	role
Edit Copy Delete	19	SwarupDahal	swarupdahal17@gmail.com	#nepal123#	#nepal123#	1	user
Edit Copy Delete	20	pukar	pukar123@gmailcom	@pukar123	@pukar123	1	user
Edit Copy Delete	21	Abhinav	abhinav12@gmail.com	#abhi123#	#abhi123#	1	user

User Info table

←T→	cartid	userid	productid	productname	quantity	image	price
Edit Copy Delete	111	19	207	Molten H3X5001-BW Handball	1	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	97
Edit Copy Delete	112	19	205	Italy National Football Team Jersey	1	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	72

Cart Info table

←T→	id	Categories
Edit Copy Delete	1	All
Edit Copy Delete	2	Cricket
Edit Copy Delete	3	Football
Edit Copy Delete	4	Jersey
Edit Copy Delete	5	Basketball

Category table

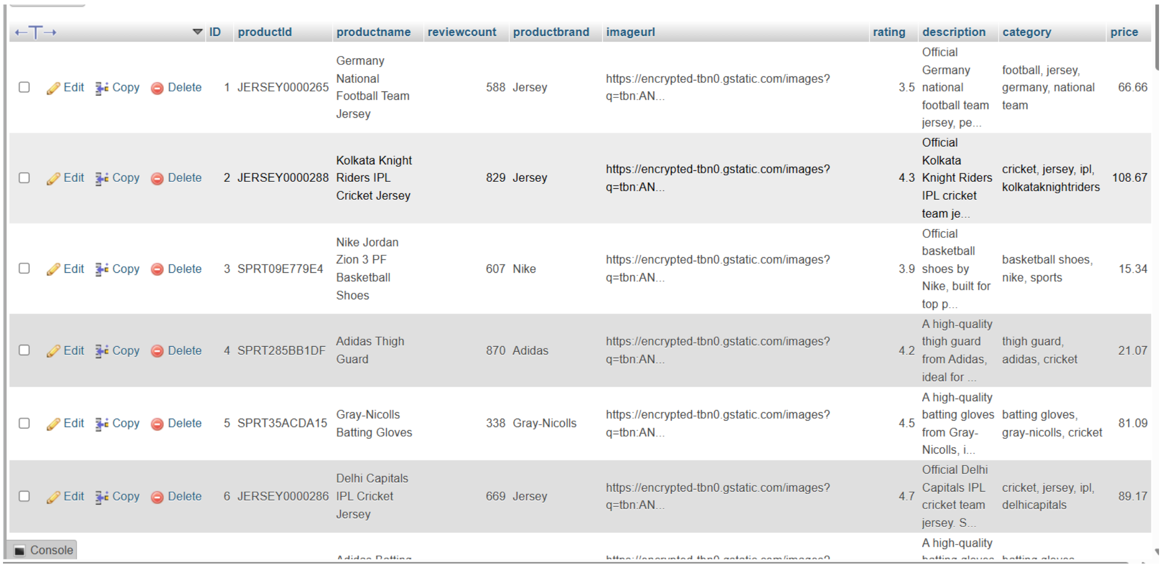
←T→	purchaseid	productid	productname	quantity	productprice	purchaseTime	userid
Edit Copy Delete	149	203	DSC Batting Pads	1	86	2025-08-30 15:14:51	19
Edit Copy Delete	150	204	Reebok Club C 85 Vintage Sneakers	1	144	2025-08-30 15:18:17	19

Purchase table

←T→	id	adminName	adminPassword	adminRepassword	role
Edit Copy Delete	1	admin	admin@123	admin@123	admin

Check all *With selected:* Edit Copy Delete Export

Admin table



	ID	productId	productname	reviewcount	productbrand	imageurl	rating	description	category	price
Edit Copy Delete	1	JERSEY0000265	Germany National Football Team Jersey	588	Jersey	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	3.5	Official Germany national football team jersey, pe...	football, jersey, germany, national team	66.66
Edit Copy Delete	2	JERSEY0000288	Kolkata Knight Riders IPL Cricket Jersey	829	Jersey	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	4.3	Official Kolkata Knight Riders IPL cricket team je...	cricket, jersey, ipl, kolkataknightriders	108.67
Edit Copy Delete	3	SPRT09E779E4	Nike Jordan Zion 3 PF Basketball Shoes	607	Nike	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	3.9	Official basketball shoes by Nike, built for top p...	basketball shoes, nike, sports	15.34
Edit Copy Delete	4	SPRT285BB1DF	Adidas Thigh Guard	870	Adidas	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	4.2	A high-quality thigh guard from Adidas, ideal for ...	thigh guard, adidas, cricket	21.07
Edit Copy Delete	5	SPRT35ACDA15	Gray-Nicolls Batting Gloves	338	Gray-Nicolls	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	4.5	A high-quality batting gloves from Gray-Nicolls, i...	batting gloves, gray-nicolls, cricket	81.09
Edit Copy Delete	6	JERSEY0000286	Delhi Capitals IPL Cricket Jersey	669	Jersey	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	4.7	Official Delhi Capitals IPL cricket team jersey. S...	cricket, jersey, ipl, delhicapitals	89.17

Console

Product Table

	pid	pname	reviewcount	brand	imageurl	rating	description	category	price
Edit Copy Delete	201	Adidas Batting Pads	420	Adidas	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	5	A high-quality batting pads from Adidas, ideal for...	Cricket	1200
Edit Copy Delete	202	Gray-Nicolls Cricket Helmet	264	Gray-Nicolls	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	5	A high-quality cricket helmet from Gray-Nicolls, i...	Cricket	1000
Edit Copy Delete	203	DSC Batting Pads	201	DSC	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	5	A high-quality batting pads from DSC, ideal for bo...	Cricket	86
Edit Copy Delete	204	Reebok Club C 85 Vintage Sneakers	248	Reebok	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	5	Official casual shoes by Reebok, built for top per...	Casual Shoes	144
Edit Copy Delete	205	Italy National Football Team Jersey	356	Jersey	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	5	Official Italy national football team jersey, perf...	Football	72
Edit Copy Delete	206	Newbery Cricket Helmet	686	Newbery	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	4	A high-quality cricket helmet from Newbery, ideal ...	Cricket	148
Edit Copy Delete	207	Molten H3X5001-BW Handball	534	Molten	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	4	Official handball by Molten, built for top perform...	Handball	97
Edit Copy Delete	208	DSC Wicket Keeping Gloves	440	DSC	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	4	A high-quality wicket keeping gloves from DSC, ide...	Cricket	154
Edit Copy Delete	209	Diadora Brasil Sala Indoor Shoes	127	Diadora	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	4	Official indoor soccer shoes by Diadora, built for...	Indoor Soccer Shoes	81
Edit Copy Delete	210	Molten BG5000 Premium Basketball	798	Molten	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	4	Official basketball by Molten, built for top perfo...	Basketball	28
Edit Copy Delete	211	Reebok Workout Plus Shoes	202	Reebok	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	4	Official training shoes by Reebok, built for top p...	Training Shoes	178
Edit Copy Delete	212	New Balance Cricket Bat	733	New Balance	https://cricketdirect.co.uk/cdn/shop/files/4tcplbw...	4	A high-quality cricket bat from New Balance, ideal...	Cricket	42
Edit Copy Delete	213	Gunn & Moore Thigh Guard	507	Gunn & Moore	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	4	A high-quality thigh guard from Gunn & Moore, idea...	Cricket	47
Edit Copy Delete	214	Molten IV5F-N Indoor Volleyball	753	Molten	https://encrypted-tbn0.gstatic.com/images?q=tbn:AN...	4	Official volleyball by Molten, built for top perfo...	Volleyball	14

Display Product table