COSC6376 Final Report
ClickStream Insights Accelerator : A Recommendation System for E-commerce Website on AWS
Team ClickStream Crusuaders

Anirudh Kalva - 2288613 - akalva@cougarnet.uh.edu
Venkata Kausik Renduchintala - 2260419 - vrenduch@cougarnet.uh.edu
Sujan Chithaluri - 2304283 - schithal@cougarnet.uh.edu

Contents

List of Figures

1 Abstract

In the highly competitive e-commerce landscape, the ability to provide real-time product recommendations based on a user's behavior is a powerful tool. E-commerce platforms generate vast amounts of data daily which is very useful. A cloud-based recommendation engine can handle this data influx seamlessly, ensuring that the system can grow with the business without concerns about hardware limitations. Hence we developed a cloud based recommendation system using AWS that delivers personalized product suggestions to customers while they browse our e-commerce platform by analyzing the real-time clickstream data using, purchase history, and various customer profile attributes using advanced machine learning models which enhance the user experience and drive sales. .

2 Introduction

Today's e-commerce applications are supposed to fulfil the demands of thousands of customers which can cause huge loss of revenues. Hence, the success of any online company highly depends on the potential to captivate visitors. It is feasible for the company to track the data about customer interaction through the so-called click stream data. It is the principal source of information for the companies to adapt their service according to their customers. Clickstream analysis may help these organizations determine customer loyalty, improve marketing strategies, the effectiveness of promotional campaigns, provide more customized data to visitors, effective website structure, etc. Hence, understanding user behavior in Web applications has become necessary for e-commerce.

While the expectation for customer-level data analysis is high, there are still problems such as customers receiving a significant amount of uninterested mail advertisements and online recommendations are still far from absolute. To build more accurate consumer behavior models for customers, firms need to recognize their customers better. This includes understanding customers' preferences and customers' behavior through web history data.

So we are bringing up a cloud based recommendation system using AWS for a e-commerece website which gives user specific product recommendation with the help of the user click stream data and other attributes.

3 Literature Review

Drawing on a more holistic perspective of online shopping behavior, there was a shift from behavioral approaches to data-driven methods predicting online purchase behavior in general. Typically, such predictions are based on clickstream data (see, for example, Moe & Fader, 2004a; Sismeiro & Bucklin, 2004; Van den Poel & Buckinx, 2005). Clickstream data model the navigation path a customer takes through the online shop (Montgomery, 2001; Montgomery et al., 2004) and can be extracted from log files which register all requests and information transferred between the customer's computer and the company's commercial web server (Bucklin & Sismeiro, 2003).

924 International Journal of Market Research 00(0) International Journal of Market Research 64(1) Examples for using clickstream data to predict online shopping behavior are—inter alia—Moe and Fader (2004a) who proposed a conversion model predicting each customer's probability of making a purchase based on purchase and visit history. The same authors (Moe & Fader, 2004b) also developed a model for evolving visiting behavior and further, they examined the relationship between visiting frequency and purchasing propensity. They found consumers visiting an e-commerce site more frequently to have a greater propensity to buy (Moe & Fader, 2004b). Van den Poel and Buckinx (2005) predicted purchase behavior and investigated the contribution of different variables: they proved (1) general clickstream variables (i.e., number of days since last visit, and speed of clickstream behavior during last visit), (2) more detailed clickstream variables (i.e., number of accessories [and personal pages and products, respectively] viewed during last visit), (3) demographic variables (i.e., gender and the fact of supplying personal information), and (4) historical purchase behavior (i.e., number of days since last purchase and number of past purchases) to be meaningful predictors. Montgomery et al. (2004) set up different models to predict purchase conversion probability by modeling path information.

Moreover, clickstream data were frequently utilized by research to predict not only purchase behavior but further similar outcome variables. For instance, Bucklin and Sismeiro (2003) investigated drivers affecting the length of time spent viewing a website and the visitor's decision to continue browsing or to exit the website. Sismeiro and Bucklin (2004) decomposed the purchase process into sequences that must be completed for a purchase to take place (i.e., completion of product configuration, input of personal information, and order confirmation with provision of credit card data) and predicted the probability of completion for each task with covariates of browsing behavior, repeat visitation, use of decision aids, input effort, and information gathering.

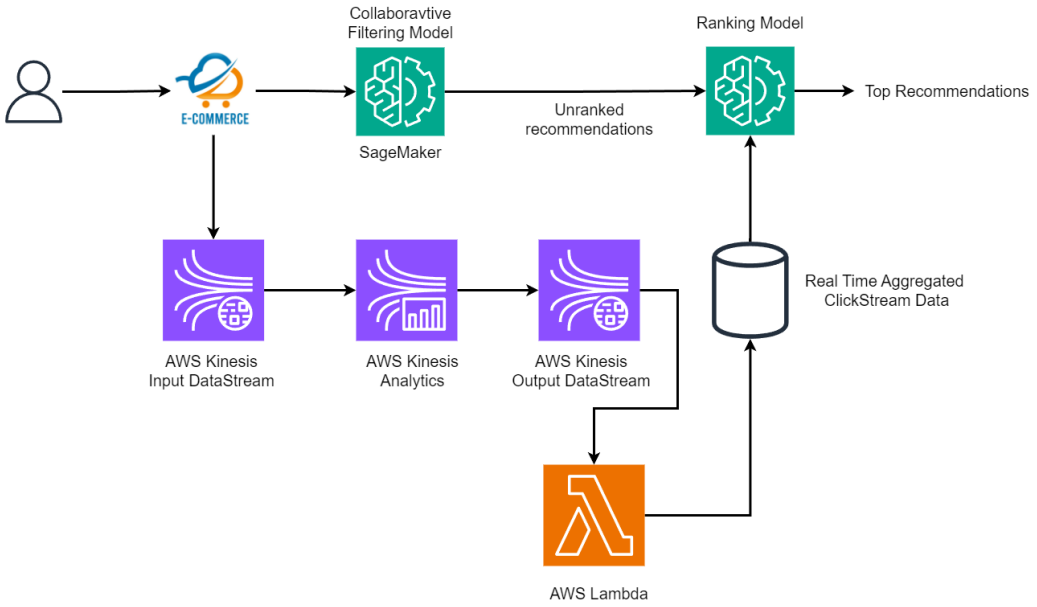

Figure 1: Architecture Diagram

4 Methodology and Implementation

4.1 System Architecture

AWS SageMaker: The entire machine learning lifecycle can be streamlined with AWS SageMaker, a fully managed machine learning service. For model development, it works with well-known frameworks like PyTorch and TensorFlow. Simple experimentation, automated model optimization, and smooth cooperation between data scientists and developers are all made possible with SageMaker. It makes training models, deploying them, and integrating them with other AWS services easier with scalable infrastructure. The service is a complete solution for many cloud-based machine learning applications because it has tools for model monitoring.

AWS Kinesis DataStream: AWS Kinesis Data Streams is a managed service designed for real-time streaming of large datasets with elastic scalability. It seamlessly integrates with various AWS services for streamlined data processing and analysis. The service ensures data durability by replicating it across multiple availability zones. With applications in analytics and

monitoring, Kinesis Data Streams offers a cost-effective and scalable solution within the AWS environment. Its versatility caters to diverse streaming data scenarios, making it a valuable component for real-time data-driven applications.

AWS Kinesis Data Analytics: For real-time analytics on streaming data, AWS Kinesis Data Analytics is a fully managed solution. For the benefit of data engineers and analysts alike, it facilitates processing and analysis via SQL or Apache Flink. With serverless architecture and built-in scalability, the service interfaces with AWS with ease. It makes scalability automatic, streamlines real-time analytics, and makes integrating with many data sources and destinations simple. Kinesis Data Analytics essentially makes it possible to efficiently extract insights from streaming data while maintaining a low level of operational complexity.

AWS Lambda: With serverless computing, code is executed without the need for server management thanks to AWS Lambda. It charges for actual compute time, scales automatically, and is multi-programming language compatible. It enables event-driven designs and smoothly connects with other AWS services. It is triggered by events. Lambda makes application development easier by providing a productive way to create event-driven apps and microservices that are scalable and affordable in the cloud.

Amazon EC2: Elastic Compute Cloud, or Amazon EC2, is a scalable cloud computing service that lets customers hire virtual servers whenever they need them. It offers a range of instance types designed for various applications and workloads. A wide range of operating systems and applications are supported by EC2 instances, which are simple to configure and customize. In order to maximize cost effectiveness, users pay for compute capability on an hourly or reserved basis. Because of its scalability and flexibility, the service is an essential component for hosting workloads and applications in the AWS cloud.

Amazon RDS: AWS's fully managed database service, Amazon RDS (Relational Database Service), supports a number of engines. Database operations like scalability and backups are automated, making administration easier. High availability, dependability, and security are guaranteed by RDS

thanks to features like automated backups and encryption. Database instances may be quickly launched and scaled by users as needed, offering a flexible solution. It is an affordable choice for relational database hosting on the AWS cloud.

4.2 Implementation

The proposed methodology begins by leveraging the AWS SageMaker Feature Store to store essential features for both model building and real-time recommendations. Specifically, the SageMaker offline feature store is utilized for collaborative filtering model training, while the online feature store is employed for the ranking model to personalize recommendations in real-time. The ingestion of clickstream data into the online feature store is facilitated through Kinesis Data Streams, Kinesis Analytics Applications, and Lambda functions

The initial steps involve creating feature groups in SageMaker for customers, products, orders, customer clickstream history, and current clickstream. Following the creation of these feature groups, training data is ingested into the respective groups, and the system awaits the availability of ingested data in the SageMaker offline feature store.

Subsequently, the offline feature store is queried to construct a comprehensive dataframe that combines data from the customer feature group and their clickstream history feature group. This dataframe is prepared for collaborative filtering model training by encoding categorical columns through one-hot encoding. Additionally, product names are transformed using the TfIdfVectorizer module from the Python sklearn library. Figure 2 shows the feature groups created in sagemaker feature store. Figure 3 represents the AWS glue tables created for each feature group created in AWS SageMaker.

The constructed dataframe is then partitioned into training and testing datasets, with 80% allocated for training and 20% for testing. To comply with the factorization-machines algorithm's expectations, the training and testing datasets are converted into RecordIO format and uploaded to Amazon S3.

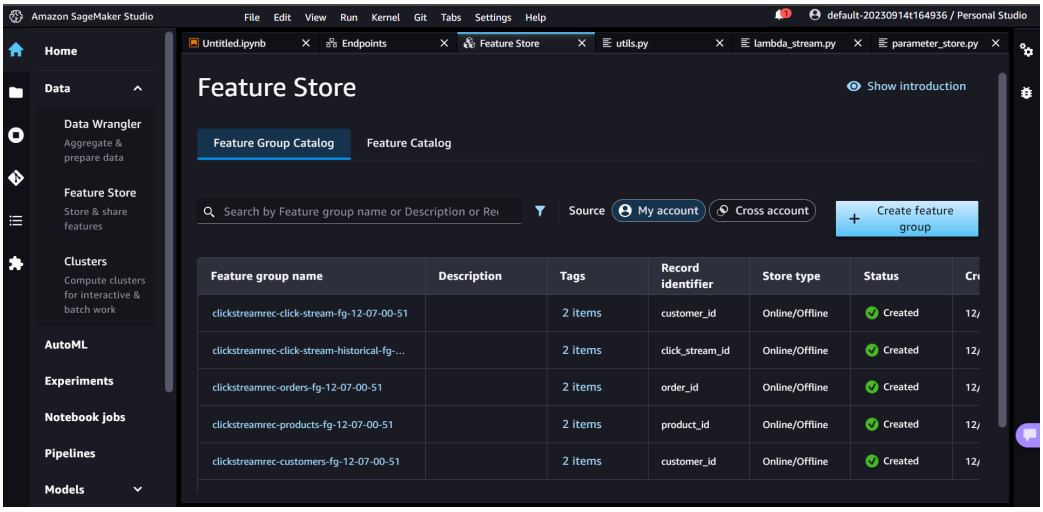

Figure 2: SageMaker FeatureStore

The collaborative filtering model is built using the **factorization-machines algorithm**, chosen for its versatility in handling both classification and regression tasks. The algorithm image is retrieved from SageMaker, and an estimator is created for training using the transformed training data. The collaborative filtering model is deployed on an **ml.m4.xlarge** instance for future use.

Following the deployment of the collaborative filtering model, attention turns to training the ranking model. This model is trained on datasets obtained by combining multiple feature groups queried from the offline SageMaker feature store. Similar to the collaborative filtering model, the training dataset undergoes one-hot encoding transformation for product categories, and the dataset is split into training and testing sets. The ranking model is trained using the **XGBoost algorithm**, and upon completion of the training process, it is deployed into SageMaker for operational use. This marks the completion of collaborative filtering and ranking model training and deployment. Figure 4 depicts the endpoints of the deployed collaborative filtering and ranking models.

A customer's past two minutes of activity on the e-commerce website will be used by the Ranking model to suggest ranked products to them.A Kinesis

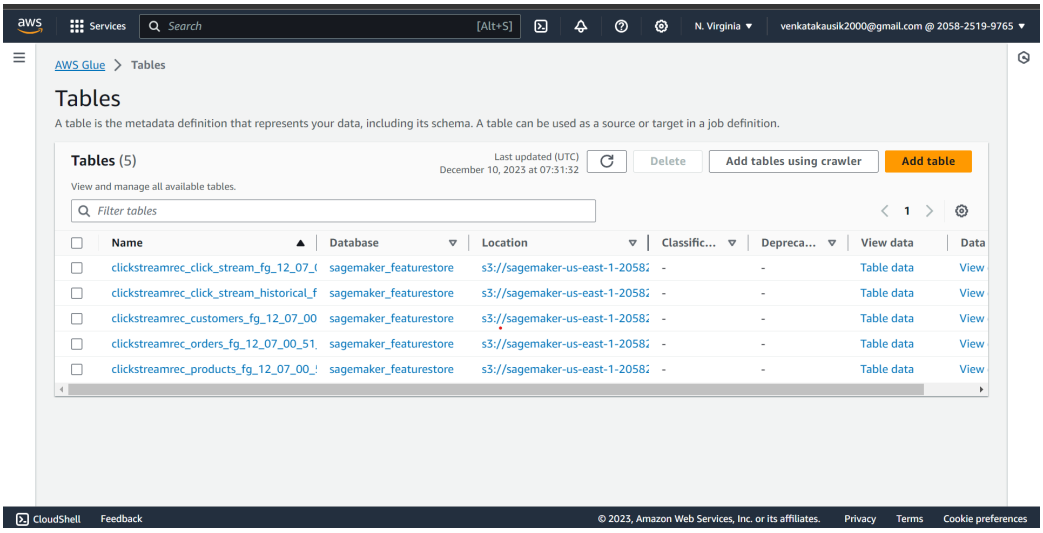

Figure 3: AWS Glue tables created for SageMaker feature groups

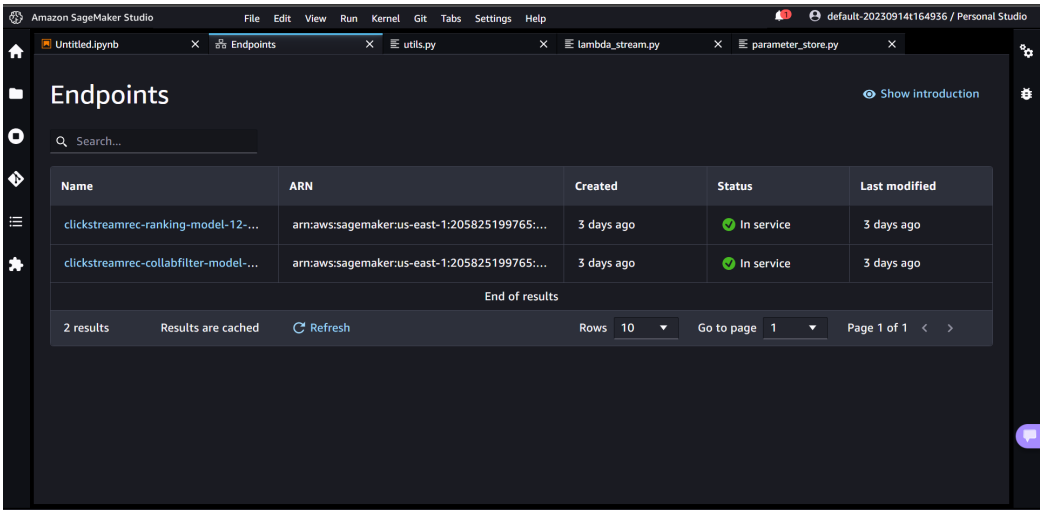

Figure 4: SageMaker Model Endpoints

Data Analytics (KDA) as shown in Figure 5 will be created and used, to aggregate the streaming information from the last two minutes. By utilizing SQL transformations, KDA can analyze data from Amazon Kinesis Data Streams with sub-second delay.

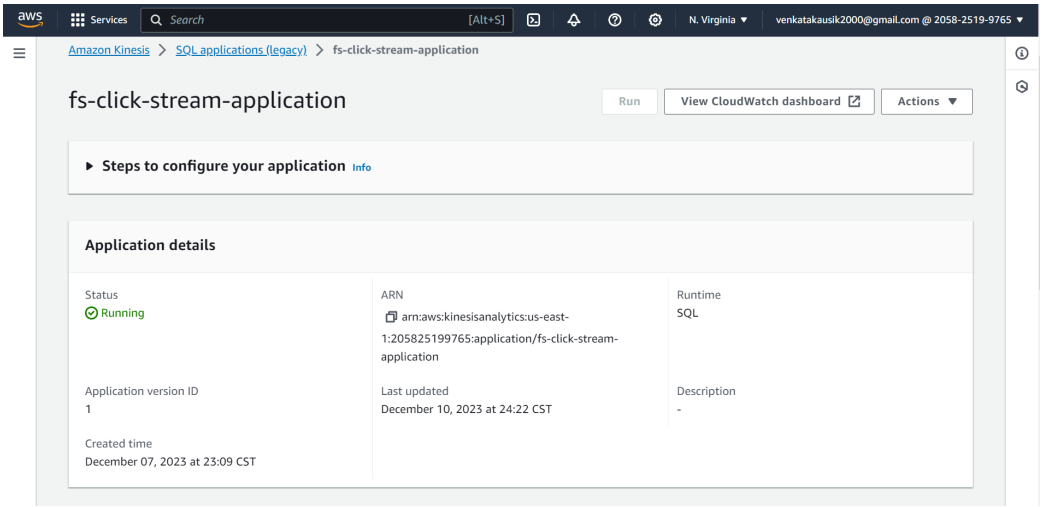

Figure 5: Kinesis Data Analytics Application

Two Kinesis DataStreams are established, serving as the input and output streams for the Kinesis Data Analytics application as shown in Figure 6. Within each two-minute window, KDA reads incoming data from the input stream, computes both the average product health index and the total user activity, and subsequently dispatches the processed output data to the designated output stream. Additionally, a Lambda function, as depicted in Figure6, is created to trigger whenever new data is inserted into the output stream from the Kinesis Data Analytics application. This Lambda function is responsible for ingesting the incoming data from the output stream into the SageMaker online feature store, a pivotal component used for ranking user recommendations based on their activity within the last two minutes.

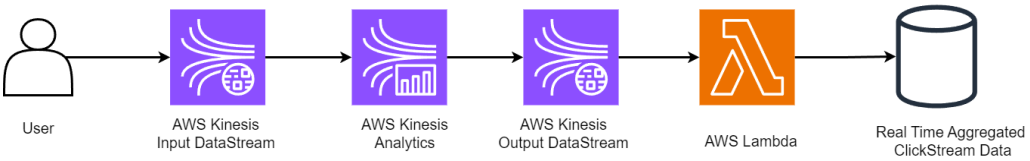

Figure 6: Real time stream processing flow

Subsequently, a Python Flask application is crafted to serve as the backend, featuring REST APIs for user authentication, event display, user action

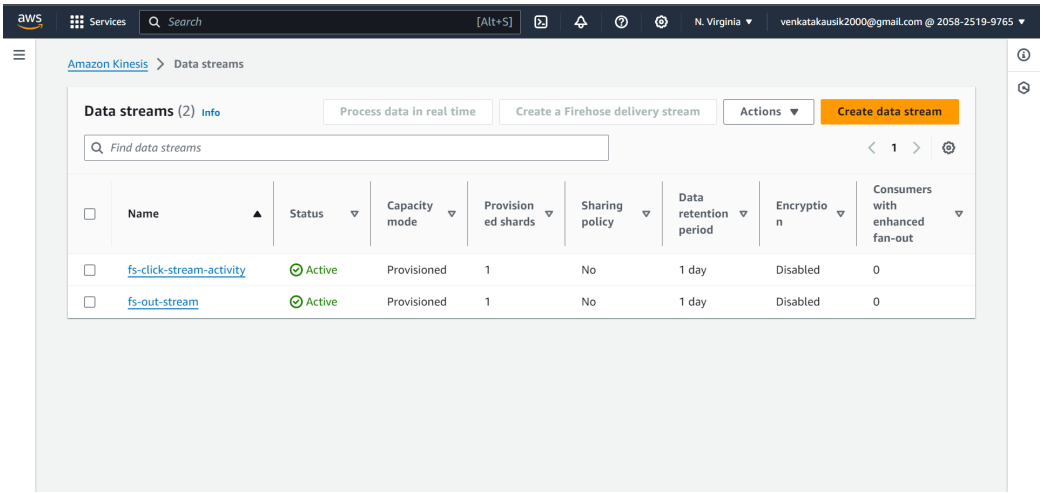

Figure 7: Input & Output Kinesis datastreams

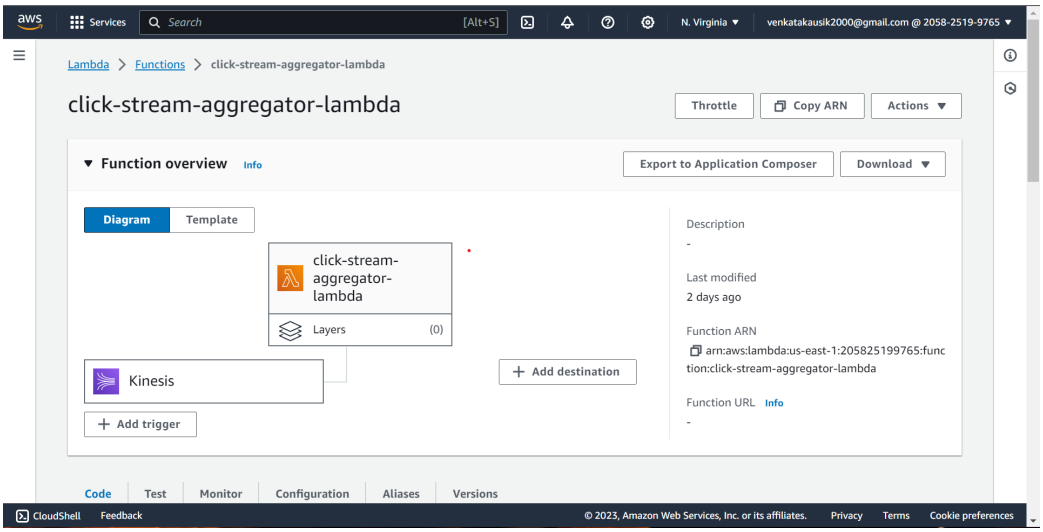

Figure 8: Lambda function triggered on Kinesis DataStream ingestion

submission to the Kinesis input stream, and retrieval of personalized product recommendations. This Flask app is deployed on Nginx on an EC2 instance within the AWS environment. In parallel, a React frontend application is developed to facilitate user interaction with the e-commerce website. Real-time streaming of user actions such as adding products to the cart, liking a

product, or saving a product for later is seamlessly integrated into the Kinesis input stream for further processing.

As previously mentioned, Kinesis Data Analytics (KDA) periodically reads incoming data from the input data stream every two minutes, aggregating this information and subsequently ingesting the processed data into the output stream. Within the Python Flask application, an API is meticulously crafted to invoke the collaborative filtering model. This API retrieves unranked product recommendations for a user and utilizes the XGBoost ranking model to assign ranks to the products. The application then furnishes user-specific recommendations based on the user's clickstream activity.

5 Results and Evaluations

The images in Figure 9-27 illustrate the seamless integration of an e-commerce frontend React application with a backend Flask app, both running on EC2 instances.

In Figure 27, we observe the dynamic recommendations presented to users based on their clickstream data. These personalized suggestions are delivered to the user interface through the Flask backend, which, in turn, generates recommendations using machine learning models deployed in SageMaker. The clickstream data is continuously streamed through a Kinesis Analytics application, ensuring real-time insights and enhancing the overall user experience.

Figure 18-23 depicts the UI functionalities to like, wishlist, save a product, and add the product to the cart. Each of these click events is captured and sent to the backend flask application to send the data to the Kinesis datastream and processed further to get personalized recommendations in real-time.

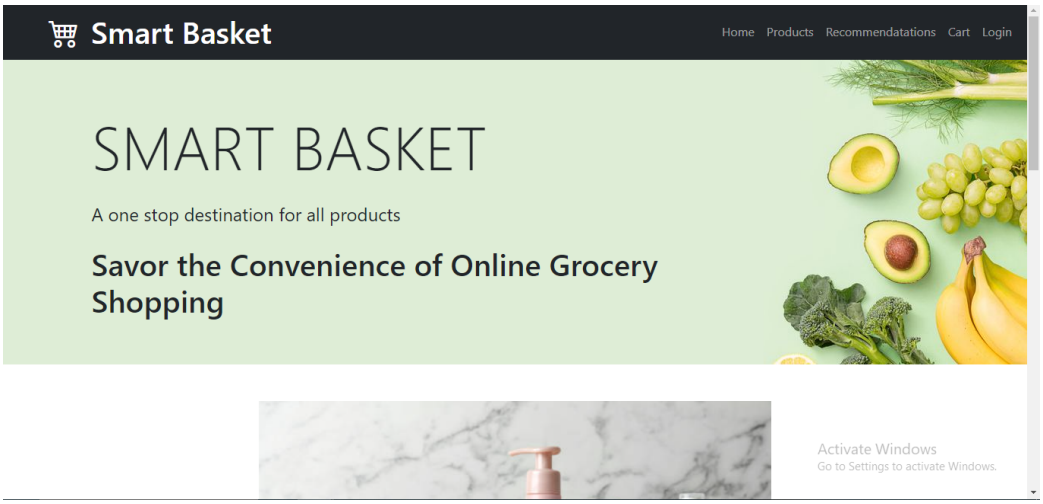

Figure 9: Home Page

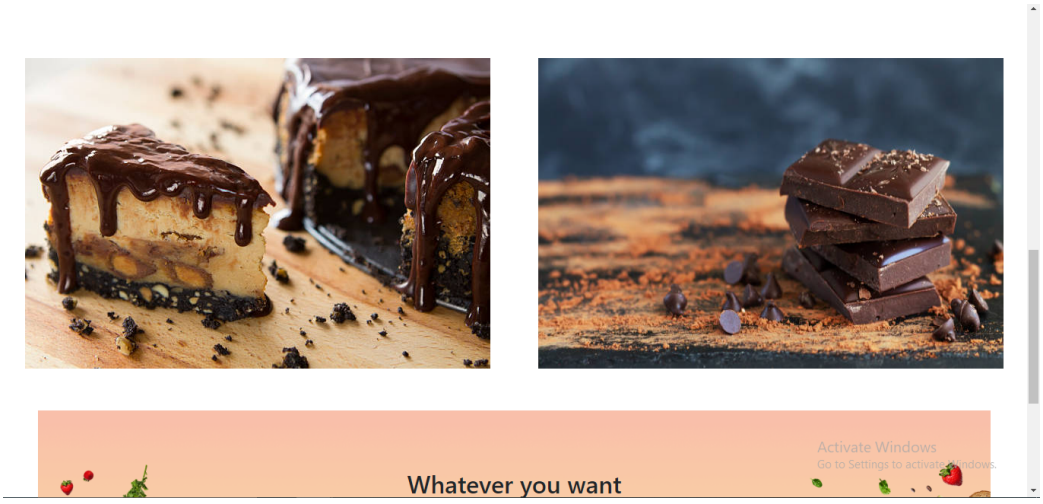

Figure 10: Home Page - 2

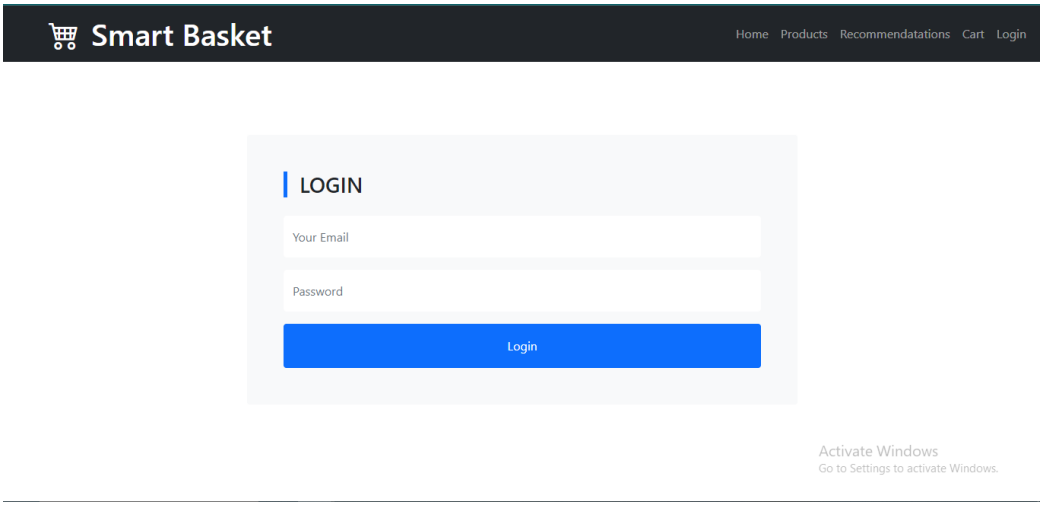

Figure 11: Login Page

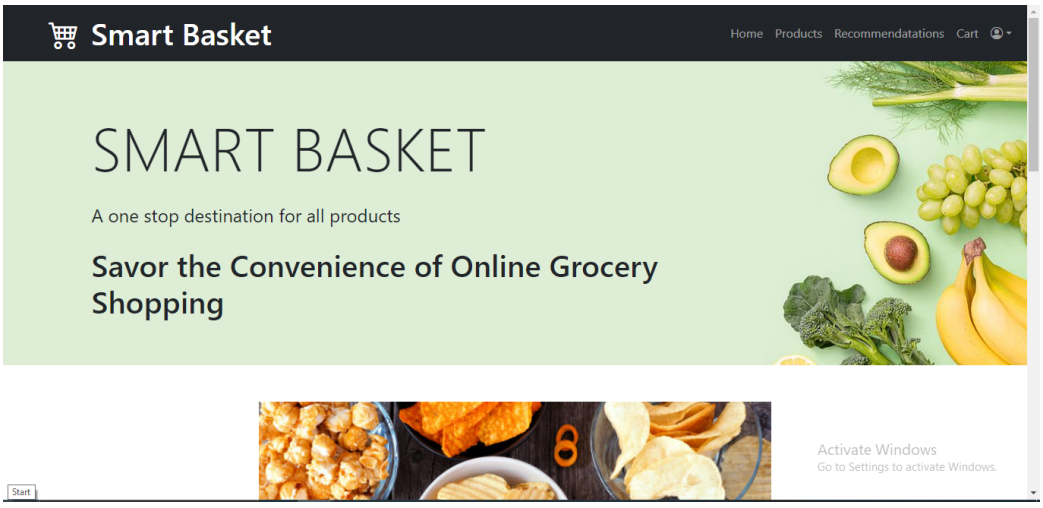

Figure 12: Home Page Post Login In

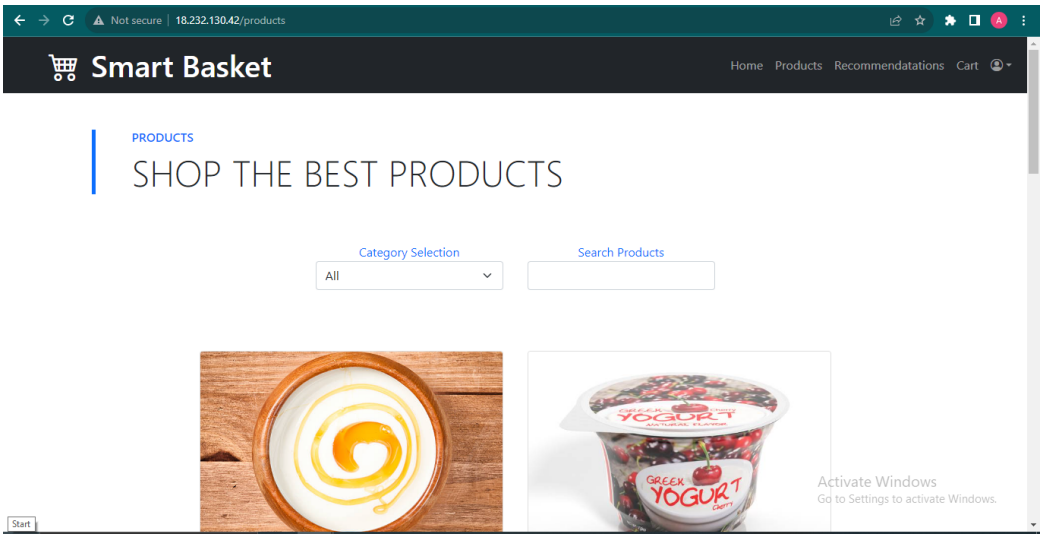

Figure 13: Products Page Showing Products of All Categories-1

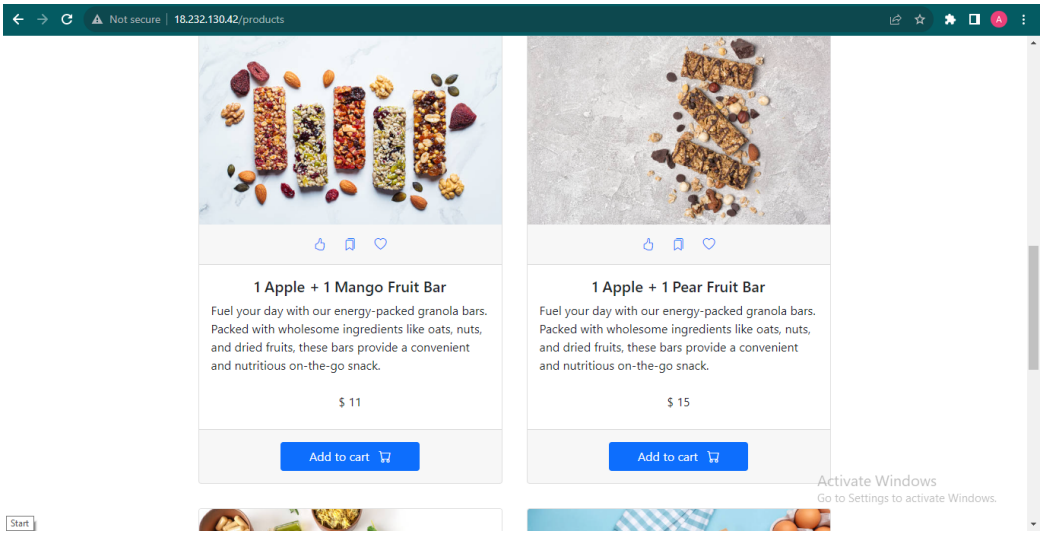

Figure 14: Products Page Showing Products of all Categories-2

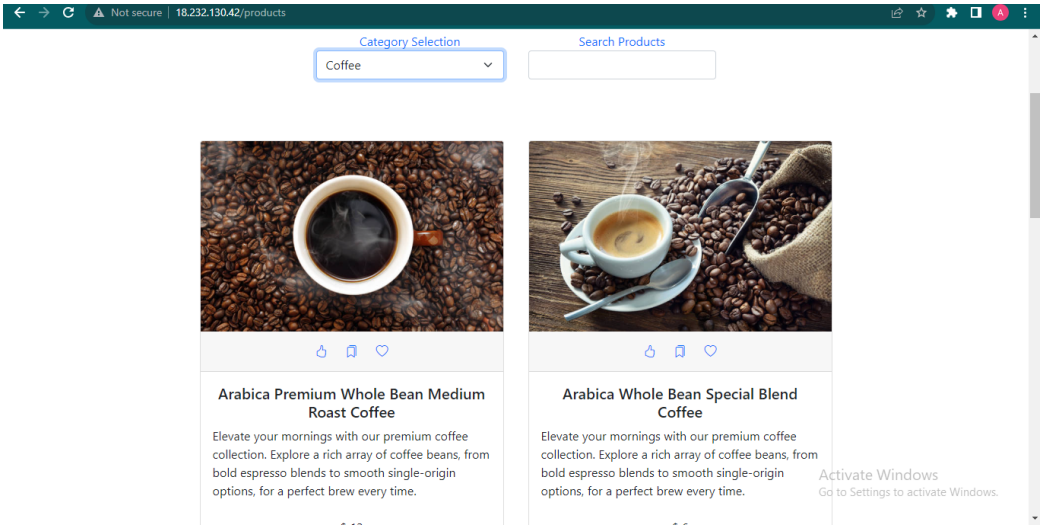

Figure 15: Products Page Showing Products of a Particular Category

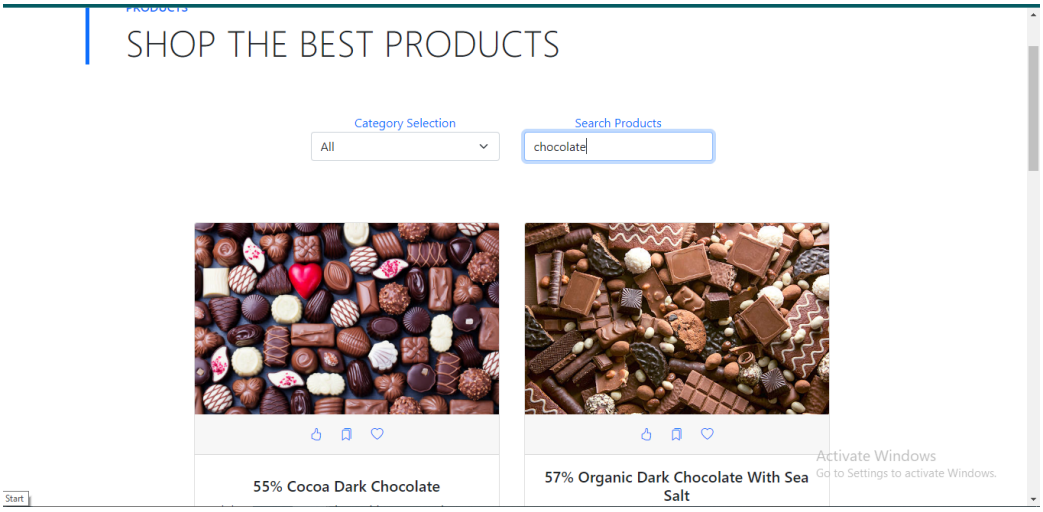

Figure 16: Searching For Products

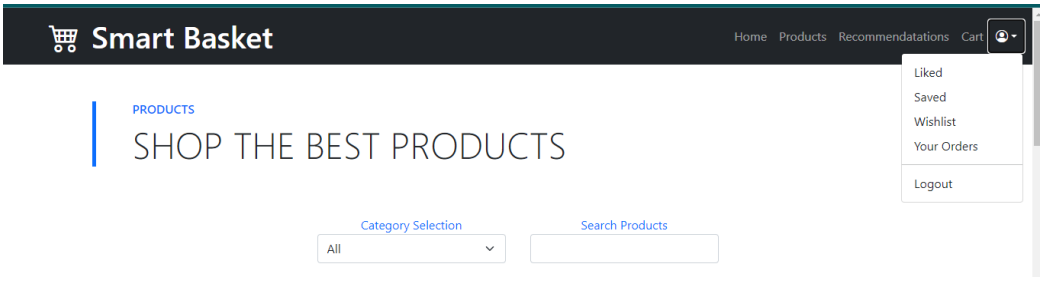

Figure 17: User Options For Navigating To Different Pages

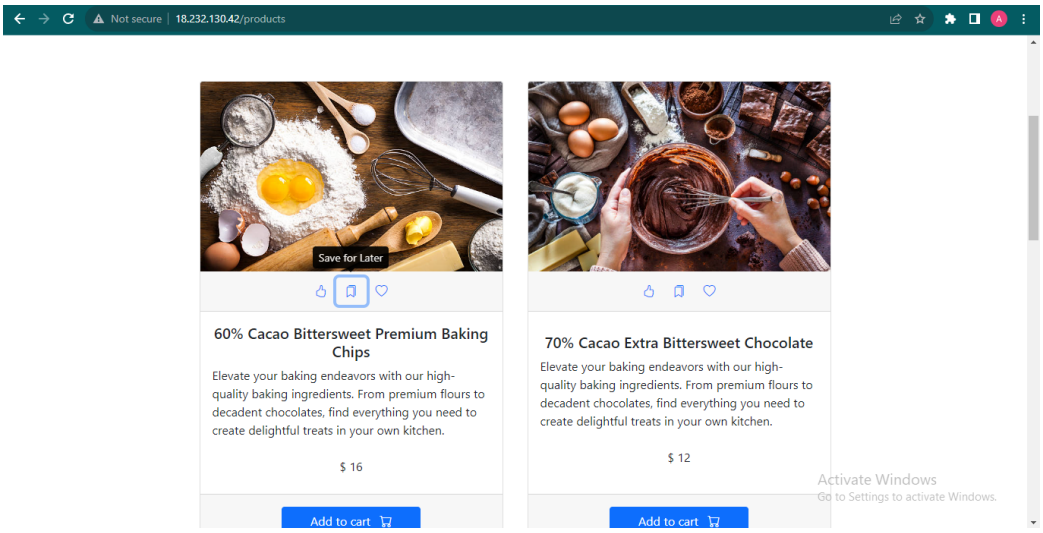

Figure 18: Saving a Product for Later

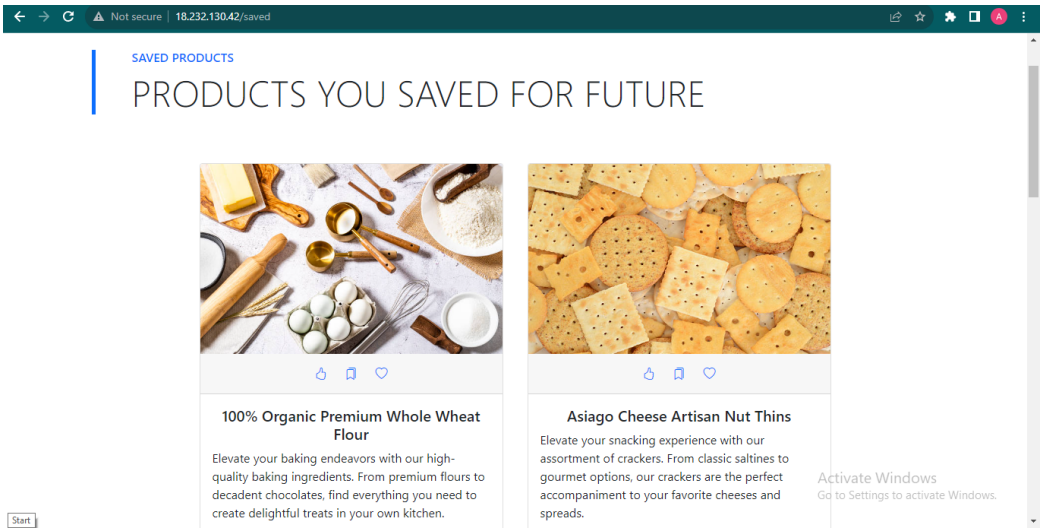

Figure 19: Page Showing Products Saved

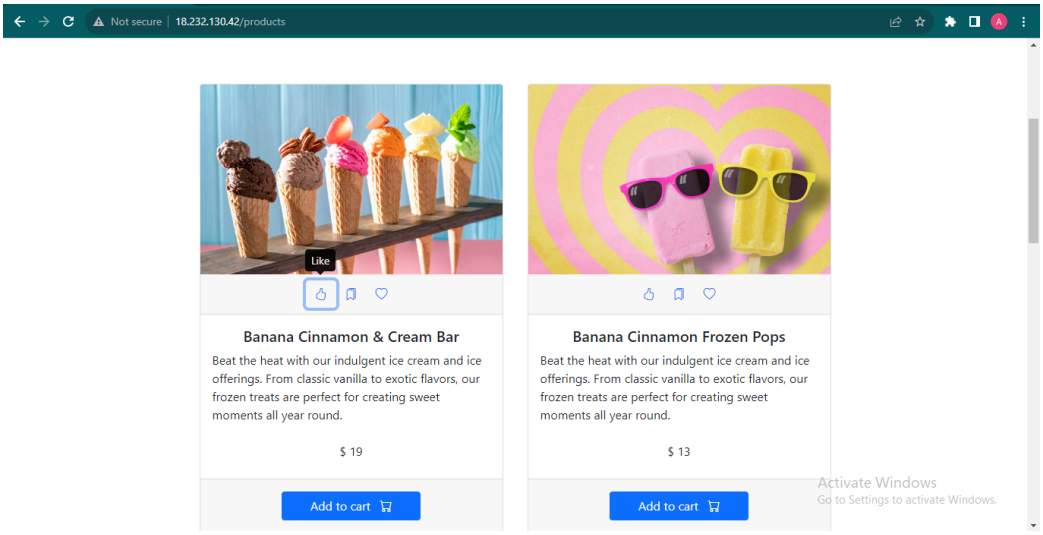

Figure 20: Liking a Product

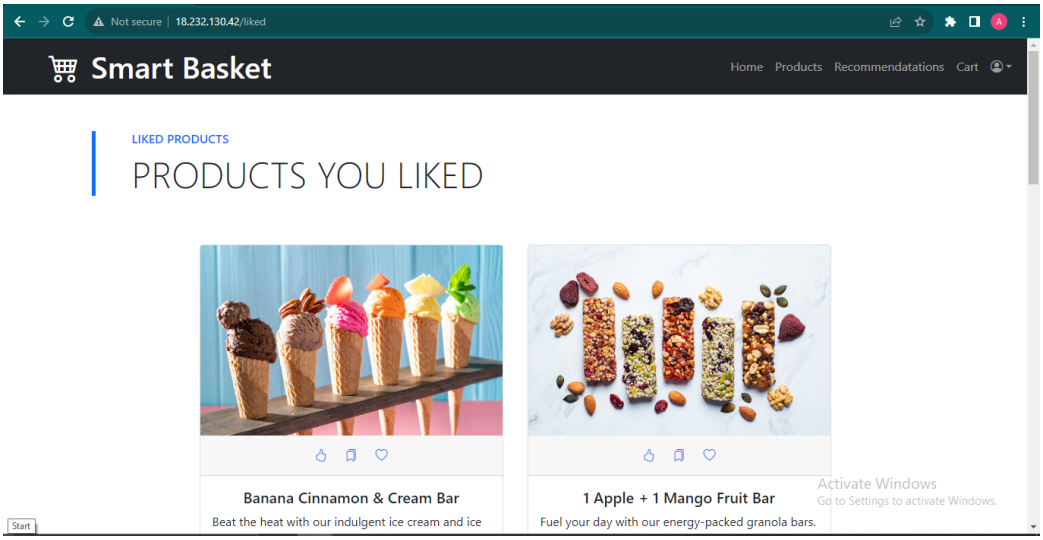

Figure 21: Page showing products liked

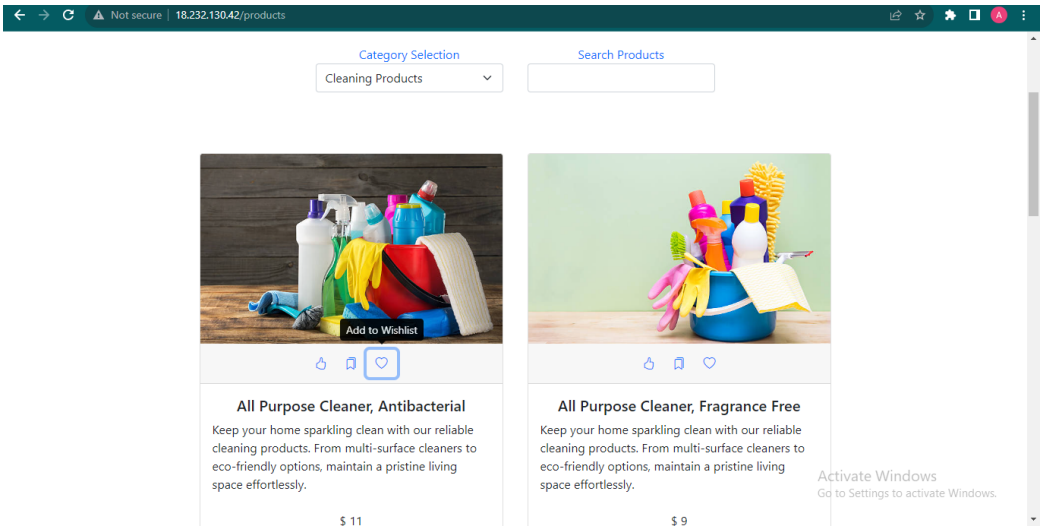

Figure 22: Product Adding To Wish List

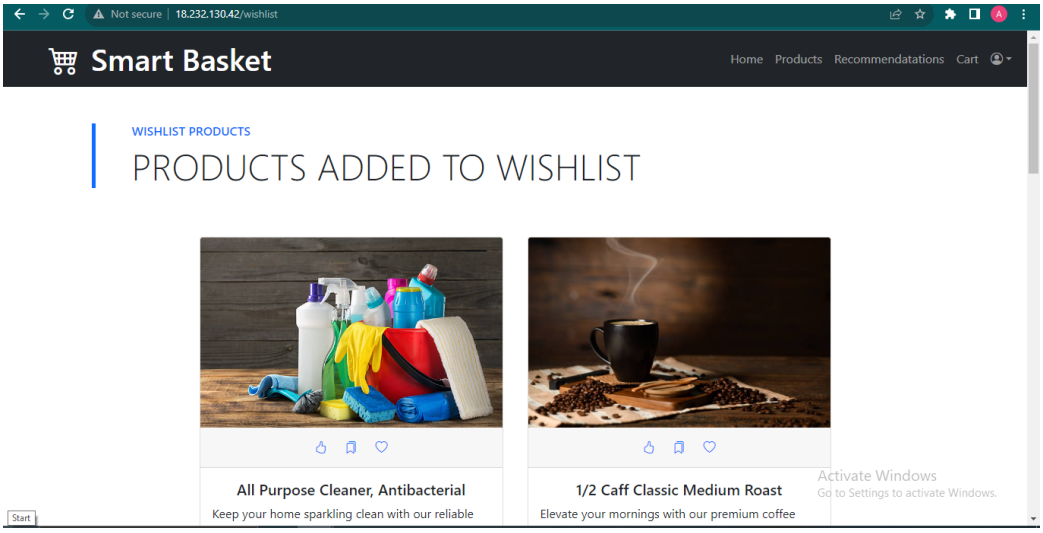

Figure 23: Wish List Page

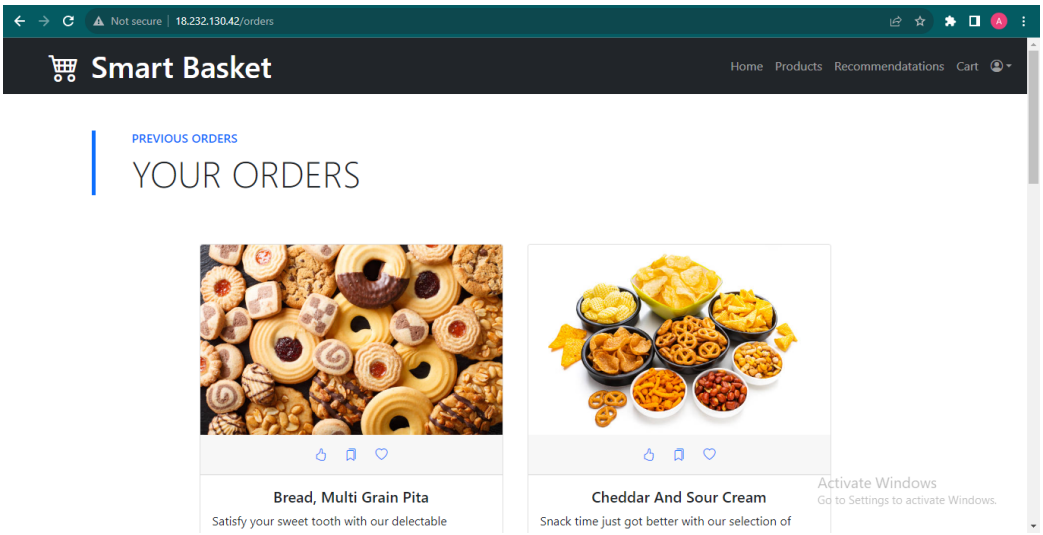

Figure 24: Orders Page

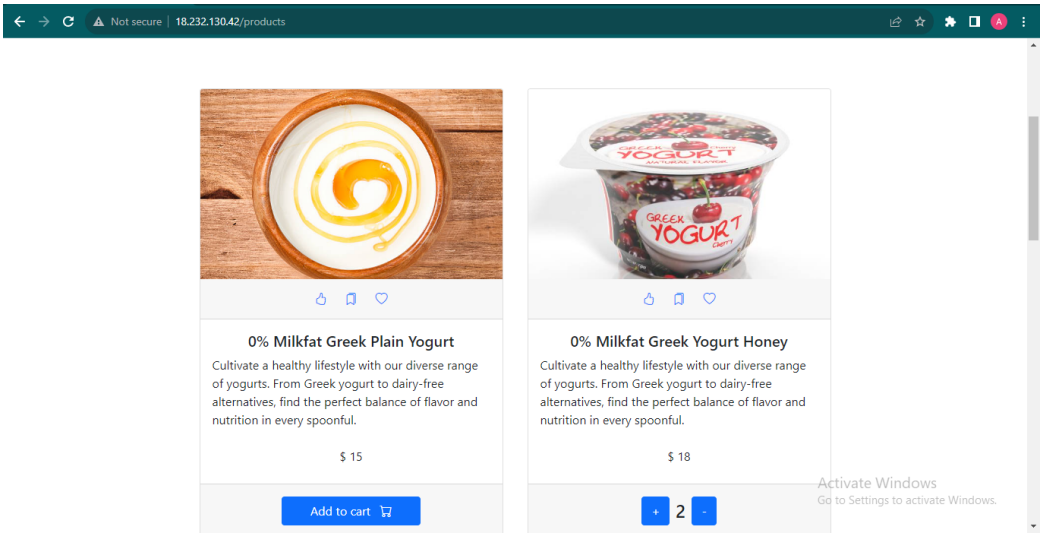

Figure 25: Adding Product To Cart

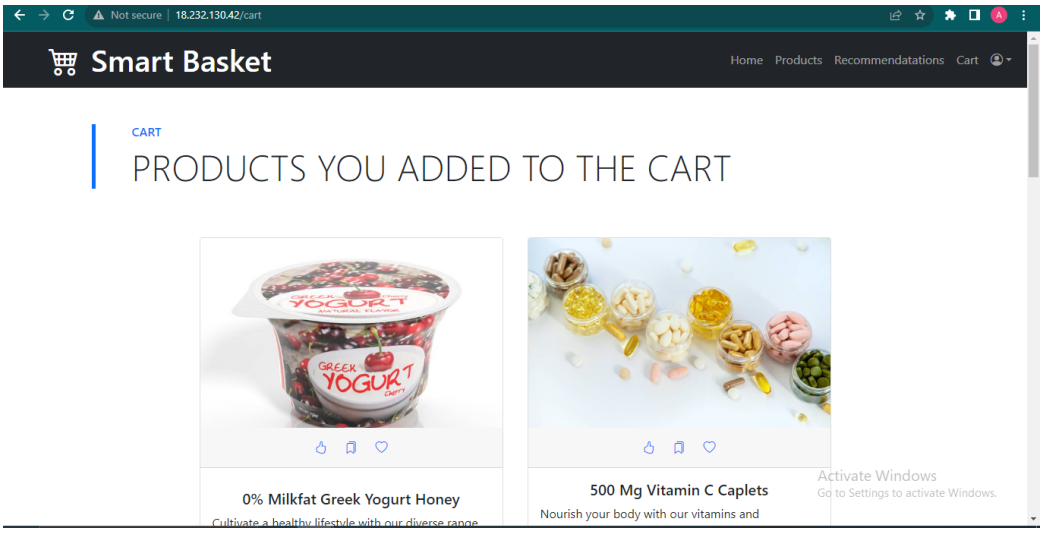

Figure 26: Cart Page Showing Added Products

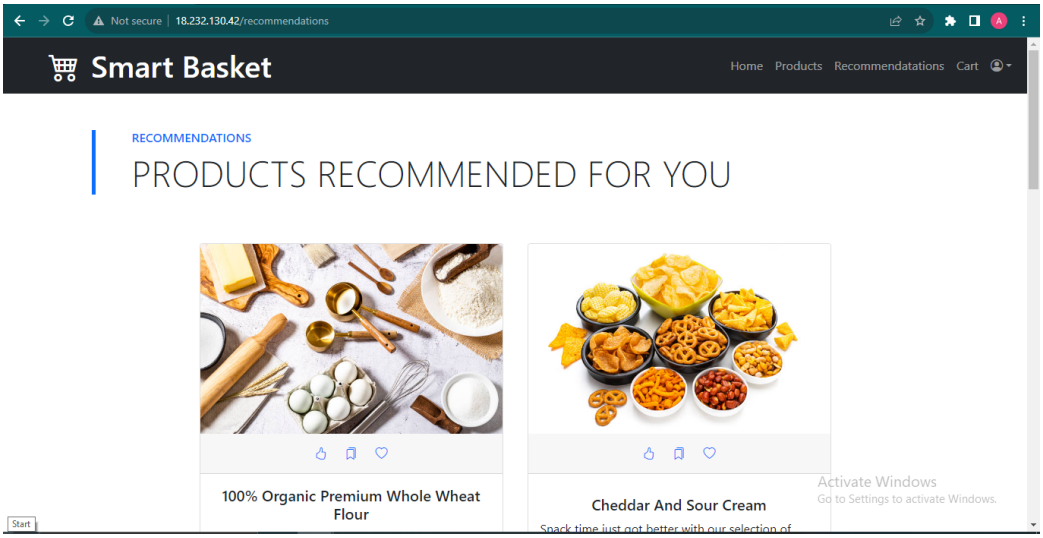

Figure 27: Recommendations Page showing recommendations based on user click stream and other attributes

6 Conclusion

In conclusion, our cloud recommendation engine powered by AWS represents an innovative stride in the e-commerce landscape. Leveraging real-time click-stream data, diverse client profiles, and purchase history, our advanced machine learning models deployed in Sagemaker can seamlessly deliver personalized product recommendations. The scalability inherent in our cloud architecture ensures adaptability and establishing a future-proof strategy without concerns about hardware limitations. This creative endeavor has significantly elevated the customer experience, providing relevant and timely advice, ultimately driving increased engagement and sales. Committed to staying ahead in technology, our cloud-based recommendation system stands as a testament to the successful integration of cutting-edge machine learning and data-driven insights, optimizing the online shopping journey for our users.

7 Future Scope

Incorporate more sophisticated user context analysis, such as location, time of day, and user demographics, to provide context-aware and accurate recommendations.

Utilizing the gathered data and user behavior patterns for generating valuable business intelligence insights that can inform marketing, product development, and content strategies.

8 References

- https://journals.sagepub.com/doi/full/10.1177/1470785320972526
- https://library.ucsd.edu/dc/object/bb8503744c/_2_1.pdf
- https://www.researchgate.net/publication/357839212_Click_Stream_Analysis_in_E-Commerce_Websites-a_Framework
- https://link.springer.com/article/10.1007/s10660-021-09518-4