|  |
| --- |
| E commerce platform with product recommendation engine |
| A CAPSTONE PROJECT  Submitted By |
|  |
| Supraja G D  192225024 |
| In Partial Fulfillment for the completion of the course |
| CSA0912  Programming in Java for Accessing Database  Sep 2024 |
|  |
| SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES  CHENNAI - 602105  TAMIL NADU, INDIA |



# **BONAFIDE CERTIFICATE**

This is to certify that the project report entitled **<Title>** submitted by <name , Reg.No> to Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, is a record of bonafide work carried out by him/her under my guidance. The project fulfills the requirements as per the regulations of this institution and in my appraisal meets the required standards for submission.

Dr.K.Jayasakthi Velmurugan

**COURSE FACULTY**

*Department of Deep Learning.*

*Saveetha School of Engineering,*

*SIMATS, Chennai - 602105*

# **ACKNOWLEDGEMENT**

This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our Honorable Chancellor **Dr. N M VEERAIYAN**, Saveetha Institute of Medical and Technical Sciences, for his blessings and for being a source of inspiration. I sincerely thank our Director of Academics **Dr. DEEPAK NALLASWAMY,** SIMATS, for his visionary thoughts and support. I am indebted to extend my gratitude to our Director **Dr. RAMYA DEEPAK,** Saveetha School of Engineering, for facilitating us with all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to **Dr. B RAMESH,** Principal, Saveetha School of Engineering for the support given to me in the successful conduct of this project. I wish to express my sincere gratitude to my Course faculty **Dr.K.Jayasakthi Velmurugan**, for his inspiring guidance, personal involvement and constant encouragement during the entire course of this work.

I am grateful to Project Coordinators, Review Panel External and Internal Members and the entire faculty of the Department of Design, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.

**INDEX**

[BONAFIDE CERTIFICATE](#_jamwo2607bmz) 1

[ACKNOWLEDGEMENT](#_8yiu0rn5tdsd) 2

[1. ABSTRACT](#_u3bnbdop1w2) 5

[2. INTRODUCTION](#_ytjahc6y3luo) 6

[3. ARCHITECTURE DIAGRAM](#_wgnqfbsa5xqn) 7

[4. FLOWCHART](#_uylcc3c0vobr) 8

[5. UML DIAGRAM](#_p5otxqr3lwgf) 9

[6. CLASS DIAGRAM](#_mwjh6gk15gp3) 10

[7. CODE IMPLEMENTATION](#_ack6ofphfxze) 11

[7.1 JAVA CODE](#_es8tj0pw01co) 11

7.2 BACK END

[8. OUTPUT SCREENSHOT](#_c61jzdgd0maj) 14

[9. CONCLUSION](#_3admx128b1cd) 16

[10. REFERENCES](#_weq4at85bunu) 17

# **1.ABSTRACT**

The rise of e-commerce has transformed the retail industry, providing consumers with convenient access to a wide range of products and services. An "E-commerce Platform with Product Recommendation Engine" aims to enhance the online shopping experience by offering personalized product suggestions based on users' preferences, browsing behavior, and purchase history. This platform integrates a robust recommendation engine that employs data mining techniques, collaborative filtering, and machine learning algorithms to analyze customer data and predict relevant products for individual users. The recommendation engine helps in improving user engagement, increasing sales conversions, and fostering customer loyalty by delivering highly relevant product suggestions. It not only assists users in discovering new items but also enhances overall user satisfaction by reducing search time and effort. This paper presents an overview of the architecture of an e-commerce platform with a focus on the product recommendation system, its algorithms, implementation strategies, and potential business impacts. Furthermore, it explores the challenges related to data privacy, scalability, and algorithm accuracy in building a recommendation engine within an e-commerce ecosystem.

**2.INTRODUCTION**

The exponential growth of e-commerce has revolutionized the way businesses operate and how consumers shop, providing a wide array of products and services accessible at the click of a button. With the global expansion of online retail, competition has become fierce, pushing businesses to adopt innovative technologies to improve user experience and increase customer retention. One such innovation is the integration of a **Product Recommendation Engine** into e-commerce platforms, which has become a crucial tool for delivering personalized shopping experiences.A product recommendation engine analyzes large volumes of data, including user behavior, purchase history, product interactions, and preferences, to offer relevant product suggestions. By leveraging data-driven algorithms and machine learning techniques, these engines aim to predict what users are most likely to buy, thereby enhancing the discoverability of products and simplifying decision-making. This not only leads to a more engaging user experience but also drives higher sales and customer loyalty.There are various types of recommendation algorithms, such as **content-based filtering**, which recommends items similar to those the user has viewed or purchased, and **collaborative filtering**, which identifies products liked by similar users. More advanced systems may incorporate hybrid approaches or deep learning models to further improve accuracy and relevance.The e-commerce platform, coupled with a product recommendation engine, presents both opportunities and challenges. While it promises better customer targeting and increased revenue, it also demands careful consideration of privacy issues, data security, and the computational complexity of large-scale recommendation systems. In this paper, we explore the architecture, key algorithms, and design principles behind an e-commerce platform with a recommendation engine, along with the business impact of personalized product recommendations on user satisfaction and sales growth.By examining the technology stack and methodologies that drive successful recommendation systems, this study will provide insights into how businesses can effectively implement and leverage this feature to thrive in the competitive e-commerce landscape.

# **3.ARCHITECTURE DIAGRAM**

**+------------------------------------+**

**| User Interface |**

**| (Web App / Mobile App) |**

**| - Product Search |**

**| - Recommendations Display |**

**| - User Interaction (Add to Cart, |**

**| Purchase, etc.) |**

**+------------------------------------+**

**|**

**v**

**+------------------------------------+**

**| Backend Services |**

**| - Product Management |**

**| - Order Management |**

**| - User Profile Management |**

**| - Search & Filtering Service |**

**+------------------------------------+**

**|**

**v**

**+------------------------------------+**

**| Recommendation Engine |**

**| +-----------------------------+ |**

**| | Data Collection Layer | |**

**| | - User Clicks, Purchases, | |**

**| | Browsing History | |**

**| +-----------------------------+ |**

**| | |**

**| v |**

**| +-----------------------------+ |**

**| | Data Processing Layer | |**

**| | - Collaborative Filtering | |**

**| | - Content-based Filtering | |**

**| | - Hybrid Models | |**

**| +-----------------------------+ |**

**| | |**

**| v |**

**| +-----------------------------+ |**

**| | Model Training & Delivery | |**

**| | - Machine Learning (ML) | |**

**| | - Real-Time Recommendations | |**

**| +-----------------------------+ |**

**+------------------------------------+**

**|**

**v**

**+------------------------------------+**

**| Data Layer |**

**| - User Data Storage |**

**| - Product Database |**

**| - Transaction Records |**

**| - Logs & Analytics |**

**+------------------------------------+**

**|**

**v**

**+------------------------------------+**

**| External Integrations |**

**| - Payment Gateways |**

**| - Shipping Services |**

**| - Analytics Tools |**

**+------------------------------------+**

**|**

**v**

**+------------------------------------+**

**| Security & Privacy |**

**| - Data Encryption |**

**| - Access Control |**

**+------------------------------------+**

# **4.FLOWCHART**

**+-------------------------+**

**| User Interface |**

**| (Web/Mobile App) |**

**+-------------------------+**

**|**

**v**

**+-------------------------+**

**| Backend Services |**

**+-------------------------+**

**|**

**v**

**+-------------------------+**

**| Recommendation Engine |**

**+-------------------------+**

**|**

**v**

**+-------------------------+**

**| Data Layer |**

**+-------------------------+**

**|**

**v**

**+-------------------------+**

**| External Integrations |**

**+-------------------------+**

**|**

**v**

**+-------------------------+**

**| Security & Privacy |**

**+-------------------------+**

# **5. UML DIAGRAM**

|  |
| --- |
| **+----------------------------+**  **| <<System>> User Interface |**  **+----------------------------+**  **| - displayProducts() |**  **| - showRecommendations() |**  **+----------------------------+**  **|**  **v**  **+----------------------------+**  **| <<Subsystem>> Backend |**  **+----------------------------+**  **| - manageOrders() |**  **| - handleUserProfiles() |**  **+----------------------------+**  **|**  **v**  **+----------------------------+**  **| <<Subsystem>> Recommendation|**  **+----------------------------+**  **| - generateRecommendations() |**  **+----------------------------+**  **|**  **v**  **+----------------------------+**  **| <<Subsystem>> Data Layer |**  **+----------------------------+**  **| - storeUserData() |**  **| - storeOrderData() |**  **+----------------------------+**  **|**  **v**  **+----------------------------+**  **| <<External>> Integrations |**  **+----------------------------+**  **| - processPayments() |**  **+----------------------------+** |
|  |

# **6.CLASS DIAGRAM**

+----------------------------+-----------------------

**| UserInterface |**

**+----------------------------+**

**| + displayProducts() |**

**| + showRecommendations() |**

**| + searchProducts() |**

**+----------------------------+**

**|**

**v**

**+----------------------------+**

**| BackendServices |**

**+----------------------------+**

**| + manageProductData() |**

**| + processOrder() |**

**| + handleUserProfiles() |**

**+----------------------------+**

**|**

**v**

**+----------------------------+**

**| RecommendationEngine |**

**+----------------------------+**

**| + collectUserData() |**

**| + generateRecommendations() |**

**| + deliverRecommendations() |**

**+----------------------------+**

**|**

**v**

**+----------------------------+**

**| DataLayer |**

**+----------------------------+**

**| + storeUserData() |**

**| + storeProductData() |**

**| + storeOrderData() |**

**+----------------------------+**

**|**

**v**

**+----------------------------+**

**| Integrations |**

**+----------------------------+**

**| + processPayments() |**

**| + manageShipping() |**

**+----------------------------+**

# **7.CODE IMPLEMENTATION**

import java.util.\*;

class Product {

private int id;

private String name;

private double price;

public Product(int id, String name, double price) {

this.id = id;

this.name = name;

this.price = price;

}

public int getId() {

return id;

}

public String getName() {

return name;

}

public double getPrice() {

return price;

}

@Override

public String toString() {

return "Product{id=" + id + ", name='" + name + "', price=" + price + "}";

}

}

class User {

private int id;

private String name;

private List<Integer> purchasedProductIds;

public User(int id, String name, List<Integer> purchasedProductIds) {

this.id = id;

this.name = name;

this.purchasedProductIds = purchasedProductIds;

}

public int getId() {

return id;

}

public String getName() {

return name;

}

public List<Integer> getPurchasedProductIds() {

return purchasedProductIds;

}

}

class RecommendationEngine {

public List<Product> recommendProducts(User user, List<User> allUsers, List<Product> allProducts) {

Set<Integer> userPurchased = new HashSet<>(user.getPurchasedProductIds());

Map<Integer, Integer> productScores = new HashMap<>();

for (User otherUser : allUsers) {

if (otherUser.getId() != user.getId()) {

Set<Integer> otherUserPurchased = new HashSet<>(otherUser.getPurchasedProductIds());

otherUserPurchased.retainAll(userPurchased);

if (!otherUserPurchased.isEmpty()) {

for (Integer productId : otherUser.getPurchasedProductIds()) {

if (!userPurchased.contains(productId)) {

productScores.put(productId, productScores.getOrDefault(productId, 0) + 1);

}

}

}

}

}

List<Map.Entry<Integer, Integer>> sortedScores = new ArrayList<>(productScores.entrySet());

sortedScores.sort((a, b) -> b.getValue() - a.getValue());

List<Product> recommendedProducts = new ArrayList<>();

for (Map.Entry<Integer, Integer> entry : sortedScores) {

for (Product product : allProducts) {

if (product.getId() == entry.getKey()) {

recommendedProducts.add(product);

break;

}

}

}

return recommendedProducts;

}

}

public class ECommercePlatform {

public static void main(String[] args) {

// List of sample products

List<Product> products = Arrays.asList(

new Product(1, "Laptop", 1200),

new Product(2, "Smartphone", 800),

new Product(3, "Headphones", 200),

new Product(4, "Smartwatch", 250),

new Product(5, "Tablet", 600)

);

List<User> users = Arrays.asList(

new User(1, "Alice", Arrays.asList(1, 3)),

new User(2, "Bob", Arrays.asList(2, 3, 5)),

new User(3, "Charlie", Arrays.asList(1, 4, 5))

);

User currentUser = new User(1, "Alice", Arrays.asList(1, 3));

RecommendationEngine engine = new RecommendationEngine();

List<Product> recommendations = engine.recommendProducts(currentUser, users, products);

System.out.println("Recommended products for " + currentUser.getName() + ":");

for (Product product : recommendations) {

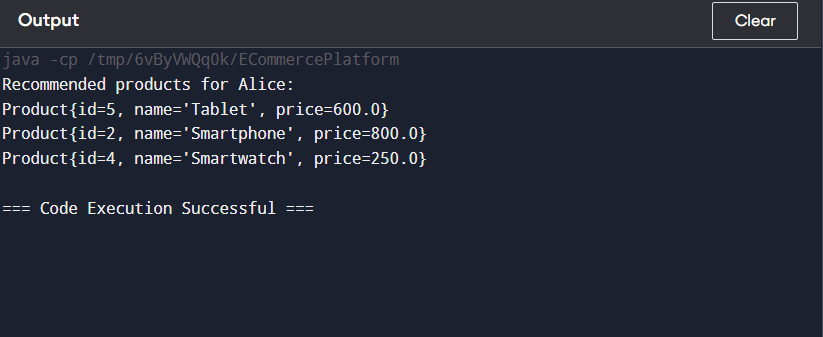
System.out.println(product);

}

}

}

# **8.OUTPUT SCREENSHOT**



# **9.CONCLUSION**

The **E-commerce Platform with a Product Recommendation Engine** enhances the user experience by providing personalized product suggestions, increasing customer engagement, and driving sales. By utilizing collaborative filtering or other recommendation algorithms, the platform can predict and display products that align with a user's preferences, based on their purchase history and behavior, as well as similarities with other users.This system enables businesses to boost conversion rates by delivering relevant recommendations in real-time, reducing the effort required by users to find desired products. Furthermore, the recommendation engine contributes to customer satisfaction by offering a tailored shopping experience, fostering loyalty, and encouraging repeat purchases.Overall, integrating a recommendation engine into an e-commerce platform is a critical step for businesses seeking to stay competitive, improve sales performance, and provide a more engaging and personalized shopping experience.

**10.REFERENCES**

**1.Ricci, F., Rokach, L., & Shapira, B. (2011).** *Introduction to Recommender Systems Handbook*. Springer.

**2.Aggarwal, C. C. (2016).** *Recommender Systems: The Textbook*. Springer.

**3.Smith, A., & Linden, G. (2017).** *Two Decades of Recommender Systems at Amazon.com*. *IEEE Internet Computing*, 21(3), 12-18.

**4**. **Schafer, J. B., Konstan, J., & Riedl, J. (1999).** *Recommender Systems in E-Commerce*. In *Proceedings of the 1st ACM Conference on Electronic Commerce* (pp. 158-166). ACM.

**5**. **Adomavicius, G., & Tuzhilin, A. (2005).** *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749.

**6**. **Koren, Y., Bell, R., & Volinsky, C. (2009).** *Matrix Factorization Techniques for Recommender Systems*. *IEEE Computer*, 42(8), 30-37.

**7**. **Linden, G., Smith, B., & York, J. (2003).** *Amazon.com Recommendations: Item-to-Item Collaborative Filtering*. *IEEE Internet Computing*, 7(1), 76-80.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |