Project Report

on

Recommendation-Based Affiliate Marketing Mobile Application

Submitted to the
Savitribai Phule Pune University, Pune
In fulfillment of the award of the Degree of
Bachelor of Engineering in Computer Engineering

Submitted By

Aditya Kishor Shinde (B190294266)
Ankita Ajit Kulkarni (B190294238)
Tanaya Bhushan Gavande (B190294228)
Krutika Ravindra Rawal (B190294258)

Under the guidance of

Prof. R. R. Shevale

Department of Computer Engineering
(ACCREDITED BY NBA)
MARATHA VIDYA PRASARAK SAMAJ'S
KARMAVEER Adv. BABURAO GANPATRAO THAKARE COLLEGE OF ENGINEERING
Udoji Maratha Boarding Campus, Gangapur Road, Nashik, Maharashtra 422013

(2022-23)

SAVITRIBAI PHULE PUNE UNIVERSITY
2022-23

CERTIFICATE

This is to certify that Project entitled

"Recommendation-Based Affiliate Marketing Mobile Application "

Submitted by

Aditya Kishor Shinde (B190294266)
Ankita Ajit Kulkarni (B190294238)
Tanaya Bhushan Gavande (B190294228)
Krutika Ravindra Rawal (B190294258)

are bonafide students of this institute and the work has been carried out by them under the supervision of **Prof. R. R. Shevale** and it is approved for the fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

Prof. R. R. Shevale
[Guide]

Dr. D. V. Medhane
[HOD]

Dr. S. R. Devane
[Principal]

[External Examiner]

MVP's K.B.T. College of Engineering Nashik-13

Place:-
Date:-

Sponsorship Certificate

TechnoFirst
SOLUTIONS
A-55, Business Index, Makahamalabad Road, Panchavati, Nashik -422003
Contact: +919820954815 Email: Pankaj.shewale@technofirst.in

TO WHOM SO EVER IT MAY CONCERN

Dear Students,

We are pleased to offer you the sponsorship for your "**TechnoFirst Solutions**"

The project is as you requested. The company determines your project status from time to time.

As per our Company policy, you will be tied up with us to not share the same project with other companies.

We hope to have a long successful professional relationship with you and wish you all the very sponsorship for the "**TechnoFirst Solutions**" project to MVPS's KBTCOE, Nashik.

Name of students:

Ankita Ajit Kulkarni

Tanaya Bhushan Gavande

Aditya Kishor Shinde

Krutika Ravindra Rawal

TECHNOFIRST SOLUTIONS NASHIK

Authorized signatory

Acknowledgement

With a deep sense of gratitude, we would like to thank all the people who have lit our path with their kind guidance. We are very grateful to these intellectuals who did their best to help during our project work.

It is our proud privilege to express a deep sense of gratitude to, **Dr. D. V. Medhane**, Head of the Computer Engineering Department for his timely suggestion and valuable guidance. The special gratitude goes to our Internal Guide **Prof. R. R. Shevale**, our External Guide **Mr. Pankaj. Shewale** and all the staff members, technical staff members, of the Computer Engineering Department for their expensive, excellent, and precious guidance in the completion of this work.

With various industry owners or lab technicians to help, it has been our endeavor to throughout our work to cover the entire project work. We are also thankful to our parents who provided their wishful support. Lastly, we thank our all friends and the people who are directly or indirectly related to our project.

Aditya Kishor Shinde -B190294266
Ankita Ajit Kulkarni -B190294238
Tanaya Bhushan Gavande -B190294228
Krutika Ravindra Rawal -B190294258

Abstract

An affiliate marketing model involves paying third-party publishers to generate traffic to a business's products or services. Affiliates are third-party publishers who receive commissions for promoting the company. Affiliate marketing has gained exponential traction in recent years. A micro-influential marketing strategy is used here by advertisers with the help of affiliate marketers. Affiliate marketers generate leads with their custom links, and they receive a product stake when the lead converts. We have developed "Recommdy", an affiliate marketing mobile application that starts with affiliate marketers' profiles and preferences and then connects advertisers with affiliates. The affiliate marketer is given a list of ads to share based on an algorithm. Thus, affiliate marketers can find relevant products and share them more easily, resulting in more sales. Both advertisers and affiliates benefit from this process. Companies can market a product effectively using affiliate marketing at a well-contained risk level with a low budget, minimal effort, and minimal time commitment, while generating a high return on investment, increased brand awareness, and business expansion. Using Android Studio and JAVA the application is built.

Keywords— Advertising Model, Affiliate marketing, Micro-influential, Mobile Application, Recommdy, Recommendation.

Contents

List of Figures

List of Tables

Chapter 1

Recommendation Based Affiliate Marketing Mobile Application

1.1 Introduction

An affiliate marketing model involves paying third-party publishers to generate traffic to a business's products or services. Affiliates are third-party publishers who receive commissions for promoting the company. Today, e-commerce is an inseparable part of the business for a variety of reasons, including the ease of use, universal accessibility, wide variety, and manageable compassion of products from different vendors, trusted payment ways, and the convenience of home shopping with the least waste of time [3]. It is a growing marketplace that mostly focuses on a specific category of audience, based on the affiliate marketer. Using affiliate marketing, companies can market their products efficiently at low-risk levels with little effort, saving money and generating a high return on investment.

The affiliate partner has rewarded a stake for providing a specific result to the retailer or advertiser as a result of marketing the retailer's or advertiser's products using an affiliate link. Thus an affiliate earns.

Recommdy bridges the gap between affiliates and advertisers and helps in building partnerships with the right people. This makes the advertiser more visible to a bigger number of potential customers, and benefits from Facebook or Instagram bloggers with many subscribers who recommend the products.

The exact conversion rate of people is not known when advertisers place their ads in the newspaper or television. So, there is not a lot of customer data available which can be analyzed to generate customer behavior reports. Hence, to track customers and improve marketing efficiency, Recommdy uses technology.

1.2 Motivation

- The motivation behind the development of our app was to provide a solution to a common problem faced by both advertisers and affiliates in the world of affiliate marketing.
- Our motivation for creating this app was to provide a solution that could benefit both advertisers and affiliates in the field of affiliate marketing. With the collaborative filtering algorithm at the heart of our app, we hope to make the process of finding and promoting products more efficient, personalized, and profitable.
- In general, advertisements are placed on television, in newspapers, etc., but the exact conversion rate of people is unknown.
- In the absence of customer data, it is impossible to calculate exact profits or losses.
- In order to bridge the gap between publisher's and affiliate marketers' audiences, technology is used.

1.3 Problem Statement

To develop a Recommendation-Based Affiliate System that will serve as a bridge between the various entities like advertisers, affiliates, and the consumer.

1.4 Objective

1. The primary objective of our project is to develop a comprehensive application called "Recommdy" that will facilitate communication and connection among publishers, affiliate marketers, viewers, and users.
2. Our aim is to offer a powerful platform that enables marketers to engage with their target audience in a more personalized and effective way, while also providing viewers and users with a seamless experience that allows them to discover and purchase products that align with their interests and needs.
3. The goal is to create a win-win situation for all parties involved, by fostering mutually beneficial relationships between publishers, marketers, and users, and providing a valuable service that meets the evolving needs of today's digital economy.
4. Affiliate marketers gain an audience here by using their personal contacts and usual followers.

1.5 Project Scope and Limitations

The scope of our project is to develop an application called "Recommdy" that will bring together publishers, affiliate marketers, viewers, and users in one platform. The application

will allow affiliate marketers to gain an audience based on their personal contacts and usual followers, and receive customized links to products related to their audience. This will enable them to target marketing and lead generation in a more effective way.

The platform will have a set of advertisers for the advertisement placement, a set of affiliates, and consumers as end-users. The application will initially be capable of handling a user base of 100k consumers, with the potential for expansion as the platform grows.

One of the key features of the application will be the recommendation system for affiliates to optimize their usage. The recommendation system will be based on a collaborative filtering algorithm, which will analyze user behavior and make personalized recommendations to each affiliate.

Despite our efforts to create a comprehensive and useful application, there are some limitations to consider. The first limitation is related to the scalability of the application. Although we aim to handle a user base of 100k consumers initially, it may be challenging to maintain and manage the growing user base in the future. Another limitation is the potential for bias in the recommendation system. Our recommendation algorithm is based on the collaborative filtering approach, which relies on user ratings and preferences. This approach may not be suitable for all users, and the recommendations may not be accurate or personalized enough for some users. Additionally, the application may not be accessible to all users, especially those who are not comfortable with technology or do not have access to a stable internet connection. Finally, the payment gateway may be limited to certain payment methods, which may not be available or preferred by all users. Despite these limitations, we will continue to monitor and improve the application to provide the best possible user experience for our users.

1.6 Methodologies of Problem Solving

As a recommendation-based app, Recommdy will require a robust problem-solving methodology to ensure that any issues that arise can be addressed quickly and efficiently. Some methodologies applied to the app are:

1. **Root cause analysis:** This methodology involves identifying the root cause of a problem and then developing a solution to address it. For example, if users are experiencing slow load times, a root cause analysis could reveal that the app's server capacity is insufficient. A solution could then be developed, such as upgrading the server capacity, to resolve the issue.

2. **Design thinking:** This methodology emphasizes a human-centered approach to problem-solving. It involves empathizing with users to understand their needs and pain points, defining the problem, ideating potential solutions, prototyping, and testing. For example, if users are having trouble navigating the app, design thinking could help identify the specific pain points and develop a user-friendly solution.

3. **Agile methodology:** This methodology is a flexible and iterative approach to problem-

solving that involves breaking down complex problems into smaller, more manageable tasks. It emphasizes collaboration, communication, and continuous improvement. For example, if users are reporting a bug in the app, the agile methodology could involve quickly identifying the issue, assigning a team member to resolve it, and then testing the solution before releasing it to users.

4. **Lean methodology:** This methodology focuses on minimizing waste and maximizing efficiency. It involves continuous testing and iteration to improve the app's features and user experience. For example, if a feature is not being used by users, the lean methodology could involve removing it to streamline the app and focus on the features that are providing the most value.

A combination of these methodologies could be applied to address any issues that arise in Recommdy and continuously improve the app's functionality and user experience.

Chapter 2

Literature Survey

How did we carry out the literature survey:

1. Defined the research scope
2. Identified the literature
3. Critically analyzed the literature
4. Categorized the resources

The literature covered a broad range of topics, including Mobile App Development and more intricate and complex topics such as Collaborative Filtering Algorithms and Neural Networks-based Clustering.
The Affiliate Marketing literature provided a brief overview of the Vietnam market, which allowed us to gain an overall view of the new and emerging need for affiliate marketing in the country.

Table 2.1: Literature Survey

Ref.No	Paper Title	Authors	Description
[1]	A Critical Look at Affiliate Marketing in Vietnam	Nhi Luu	A critical approach about why, and how Vietnamese Gen Z choose affiliate marketing is conducted throughout the semi-structured interviews as well as the data analysis.

Ref.No	Paper Title	Authors	Description
[2]	A New QoS-Aware Web Service Recommendation System Based on Contextual Feature Recognition at Server-Side	Shun Li, Junhao Wen, Fengji Luo, Member, IEEE, Min Gao, Jun Zeng, and Zhao Yang Dong, Fellow, IEEE	Extract the contextual properties from WSDL files to cluster Web services based on their feature similarities, and then utilize an improved matrix factorization method to recommend services to users.
[3]	A systematic study on the recommender systems in the e-commerce	Pegah Malekpour Alamdari 1 , Nima Jafari Navimipour 2, Mehdi Hosseinzadeh 3, Ali Asghar Safaei 4, and Aso Darwesh 5	Illustrates a comprehensive and Systematic Literature Review (SLR) regarding the papers published in the field of e-commerce recommender systems.
[4]	Developing of Android Mobile Application using Java and Eclipse: An Application	Senay Kocakoyun	A brief about how android application works and are being developed using Java and Eclipse.
[5]	Case Study on Efficient Android Programming Education using Multi Android Development Tool	Hwansoo Kang* and Jinhyung Cho	To propose education model using Eclipse ADT with android SDK and App Inventor together for efficient teaching for developing Android applications.
[6]	A Neural Networks-based Clustering Collaborative Filtering Algorithm in E-commerce Recommendation System	Jianying Mai Yongjian Fan, Yanguang Shen	Propose a variety of recommendation algorithms, many methods of which come from the latest achievement in data mining research.

Chapter 3

Hardware & Software Requirements

3.1 Introduction

The successful implementation of the Recommdy app requires a well-defined set of hardware and software requirements. This includes the use of modern hardware and software technologies that are both scalable and reliable. The software requirements for the app include an operating system that supports the latest programming languages and frameworks. The hardware requirements include a powerful server that can handle a high volume of requests, as well as reliable network infrastructure for data transmission. Additionally, the app may require integration with external services like analytics tools. Overall, meeting these hardware and software requirements is essential for ensuring that the app is able to provide a seamless and reliable experience to users, while also supporting the business needs of the company.

3.2 Assumptions and Dependencies

Assumptions and dependencies are key factors that can impact the development and success of a project. For the development of the Recommdy application, we have made the following assumptions and identified dependencies: .

Assumptions:

- The users have access to a stable internet connection and a compatible device to use the application.
- The users will be willing to provide their personal information such as their preferences and interests to receive personalized recommendations.
- The data provided by the users is accurate and reliable.

.

Dependencies:

- The availability and reliability of third-party APIs and services used by the app, such as email providers and analytics tools.

- The success of the app is dependent on the number of users and their engagement with the platform.
- The app's performance is dependent on the quality of the internet connection and the device capabilities of the users.
- The development team is dependent on the timely delivery of hardware and software components from vendors.

Identifying these assumptions and dependencies is crucial in order to plan and mitigate any potential risks or issues that may arise during the development and deployment of the Recommdy application.

3.3 Functional Requirements

1. **User registration and login:** Users should be able to create a new account and log in to the app using their email addresses or social media accounts.
2. **Profile creation and management:** Users should be able to create and manage their profiles, which should include personal information, preferences, and settings.
3. **Product management:** Advertisers should be able to manage their products, which includes adding new products, updating product details, and removing products from the platform.
4. **Advertisement management:** Advertisers should be able to create and manage their advertisements, which includes specifying the target audience, setting the ad budget, and selecting the ad placement.
5. **Affiliate management:** Affiliates should be able to manage their affiliate accounts, which includes viewing their earnings, tracking their performance, and managing their referrals.
6. **Recommendation engine:** The app should include a recommendation engine that can suggest products to users based on their purchase history and other factors.
7. **User support:** The app should provide user support, such as a help center, FAQs, and customer service, to assist users with their inquiries and issues.
8. **Security and privacy:** The app should ensure the security and privacy of user data and transactions through various measures, such as encryption, authentication, and compliance with data protection regulations.

3.4 External Interface Requirements

3.4.1 User Interface

The user interface (UI) of the Recommdy application should be intuitive, user-friendly, and visually appealing. Some of the UI requirements for the application are:

1. **Login and sign-up interface for affiliate marketers:** The interface should allow users to create an account or log in to their existing account. The interface should also have options for password reset and social login.
2. **Registration and login for advertisers/stores:** The interface should allow advertisers to register their store and create a new account or log in to their existing account.
3. **Category view of all ads:** The interface should display all the available ads in a categorized manner for the users to browse through.
4. **In-app push notification:** The app uses OneSignal API for pushing the notification from the advertiser's end i.e., from the admin panel to the affiliates.
5. **KYC verification:** To ensure that legitimate affiliates join the program, the app uses KYC verification and then assigns a verified badge to the legitimate affiliates.
6. **History of all previous activities:** The interface should provide a comprehensive history of all the user's previous activities, including purchases, ad clicks, and other interactions with the app.

3.4.2 Software Interface

Software interface requirements specify the interactions between the different software components and systems that are part of the application. These include:

1. **Application Programming Interface (API):** The API should allow external systems to interact with the application, such as for integrating with third-party systems or for providing data to other applications.
2. **Database Interface:** The application should be able to connect and interact with the database, including the ability to read, write, and update data.
3. **Social Media Integration:** The application should allow for social media integration, such as the ability for users to share their affiliate links on social media platforms.
4. **Reporting Interface:** The application should provide a reporting interface for users to view their performance metrics and earnings, as well as for the platform to track the overall performance and revenue generated.

Following are the software interfaces used for the application.

Table 3.1: Software Interfaces

Software Used	Descriptions
Operating system	In order to provide the best support and user-friendliness, we have chosen Windows as the operating system.
Database	To save the advertiser records, Affiliate's records we have chosen SQL database.
Figma	Figma is a collaboration tool for teams and individuals to create and share high-quality work.
Android Studio	The Android Studio integrated development environment (IDE) is Google's official tool for developing Android apps. Software developers can easily design, build, run, and test apps for Android using an IDE.

3.4.3 Hardware Interface

Hardware interfaces refer to the physical components of a system that are used to interact with the software. In the case of the Recommdy app, the minimum hardware requirements are a device running Android version 7.0 (Nougat) or higher. This includes smartphones and tablets that can support the app's functionality. For optimal performance, we recommend using an HP Envy Laptop with an i5 processor and 8GB RAM. These hardware requirements ensure that the app runs smoothly and efficiently, providing users with an optimal experience. By providing clear hardware requirements, we aim to ensure that users can access and use the Recommdy app effectively, regardless of the device they are using.

1. Laptop with i5 processor and 8 GB RAM (Recommended)
2. Android version 7.0(Nougat) and above.

3.4.4 Communication Interface

The communication interfaces for the Recommdy app include:

1. **Android version 7 and above:** The app is designed to be compatible with Android devices running version 7 or later, which allows for a wide range of users to access the app.

2. **User-friendly UI:** The app has a simple and easy-to-use interface that enables users to interact with the system and access its services easily.

3. **In-app push notifications:** Users can receive daily updates, new trends, and deals through push notifications in the app, which is a widely used messaging platform.

Overall, the communication interfaces are designed to enhance the user experience and make it easy for users to stay informed and engaged with the app's content.

3.5 Non Functional Requirements

3.5.1 Performance Requirements

- The app should have a fast response time to user actions, with no lag or delay in loading data.
- The app should be able to handle a large user base, with a minimum capacity of 100k users.
- The app should be able to handle a large number of concurrent users without crashing or slowing down.

3.5.2 Safety Requirements

- **Secure user authentication and authorization:** The app should ensure that only authorized users have access to the application and its features. This can be achieved by implementing secure authentication methods like two-factor authentication, strong password policies, and session management techniques.
- **Secure data storage and transmission:** The app should ensure that user data, including personal and financial information, is stored and transmitted securely. This can be achieved by using encryption techniques, secure communication protocols like HTTPS, and secure data storage mechanisms.
- **Protection against malware and viruses:** The app should be designed to protect against malware and viruses that can harm the user's device or compromise their data. This can be achieved by implementing anti-malware and anti-virus measures, like regular software updates, antivirus software, and firewalls.
- **Error handling and recovery:** The app should be designed to handle errors and recover gracefully from failures. This can be achieved by implementing robust error-handling mechanisms and providing users with clear instructions on how to recover from errors.
- **Compliance with regulatory requirements:** The app should comply with all relevant regulatory requirements, such as data protection and privacy laws. This can be achieved by implementing appropriate security measures and adhering to industry

standards and best practices.

3.5.3 Security Requirements

- The app should ensure the privacy and security of user data, including personal information, payment information, and transaction history.
- The app should use secure payment gateways to protect against fraud and unauthorized transactions.
- The app should be regularly updated and maintained to address any potential security vulnerabilities.

3.5.4 Software Quality Attributes

- **Usability:** The app should be easy to navigate and understand for all types of users. The UI should be intuitive and user-friendly, with clear labels and instructions.
- **Performance:** The app should be responsive and quick, with fast loading times and smooth transitions between screens. It should also be able to handle a large number of users without experiencing any major slowdowns.
- **Reliability:** The app should be reliable and stable, with minimal crashes or bugs. It should also have a system in place for handling errors and exceptions.
- **Security:** The app should be secure, with strong encryption and secure data storage. It should also have user authentication and authorization features to prevent unauthorized access.
- **Scalability:** The app should be scalable, with the ability to handle increasing numbers of users and data without experiencing any major issues.
- **Maintainability:** The app should be easy to maintain and update, with a clear code structure and documentation. It should also be modular to allow for easy modification of individual components.
- **Compatibility:** The app should be compatible with a wide range of devices and platforms, including different operating systems and screen sizes.

3.6 System Requirements

3.6.1 Database Requirements

For the "Recommdy" application, we have chosen Firebase as the database for user login and authorization. Firebase offers an easy-to-use and secure authentication system that can

be seamlessly integrated into our app. Additionally, we will be using an SQL database for the admin panel where the advertiser and affiliate marketer records will be stored. SQL offers a robust and scalable solution for handling large datasets and is ideal for managing user information and transactions. By using both Firebase and SQL databases, we can ensure that our app is reliable, fast, and secure, providing a seamless experience to our users.

3.6.2 Software Requirements

The software requirements for the system include the use of Android Studio for the development of the mobile application, the Android Gradle plugin 7.3.1 and Firebase CLI v10.1.0 for the integration of Firebase services, and Google's fused location API for Android for the location-based services. The mobile application must be compatible with Android version 7.0 (Nougat) and above. Additionally, Figma will be used as the collaboration tool for designing the user interface, and the system must be able to interact with a SQL database for the management of advertiser and affiliate records. The system must also be able to support secure login and authorization through Firebase and an admin panel using SQL. The communication interfaces include in-app push notifications for daily updates and new trends and deals. The system must be able to provide high performance, reliability, and security, as well as be easily maintainable and scalable.

3.6.3 Hardware Requirements

Here are the hardware requirements for the system:

1. **Processor:** Intel Core i5 or equivalent
2. **RAM:** 8GB or higher
3. **Hard Disk Space:** 500GB or higher
4. **Graphics Card:** NVIDIA or AMD Graphics Card
5. **Display:** 15.6-inch display with a minimum resolution of 1366x768
6. **Operating System:** Windows 10 or higher
7. **Mobile Device:** Android version 7.0 (Nougat) or higher

It is important to note that these are minimum requirements, and higher specifications may be needed for optimal performance, depending on the workload and usage patterns of the system. Additionally, some components may need to be upgraded over time as technology advances and new software versions are released.

3.7 Analysis Models

3.7.1 User Classes and Characteristics

For the development of our project, we have followed the waterfall model of the software development life cycle. Below given is a brief description of the same.
Waterfall Model: The waterfall Model also called a classical model or Standard Development Cycle (SDLC) model is used in the system development life cycle to create a system with a linear and sequential approach. The model completes one phase and then moves on to the next phase, all of which is done in a downward fashion.

1. **Planning:** It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys, and domain experts. This information is then used to plan the basic project approach and to conduct product feasibility studies in the economical, operational, and technical areas. Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage.

2. **Defining:** Next step is to clearly define and document the product requirements and get them approved by the customer or the market analysts.

3. **Designing:** Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented. This is reviewed by all the important stakeholders and based on various parameters such as risk assessment, product robustness, design modularity, budget, and time constraints, the best design approach is selected for the product. A design approach clearly defines all the architectural modules of the product.

4. **Building:** The actual development starts and the product is built. The programming code is generated. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle. The programming language is chosen with respect to the type of software being developed.

5. **Testing:** This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing-only stage of the product where the product defects are reported, tracked, fixed, and retested until the product reaches the quality standards defined in the SRS.

6. **Deployment:** Once the product is tested and ready to be deployed it is released formally.

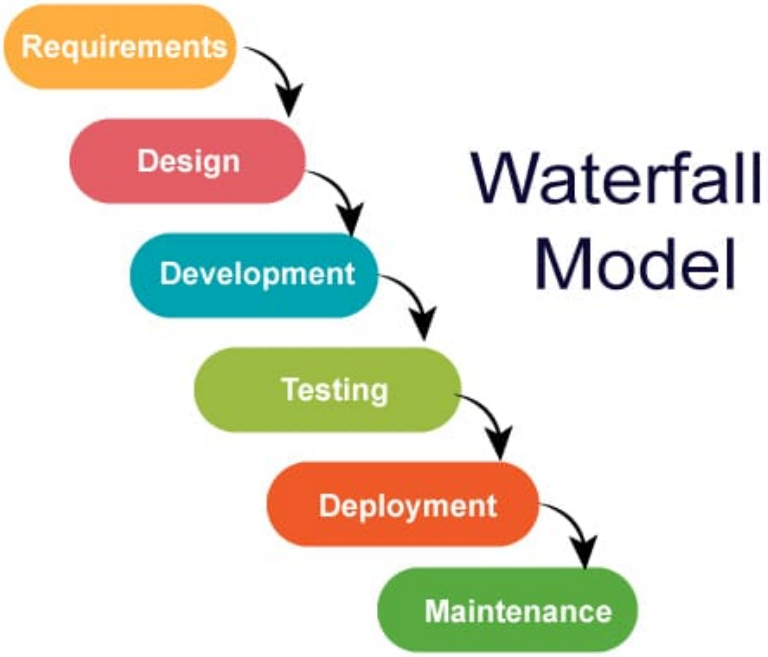


Figure 3.1: Waterfall Model

Chapter 4

System Design

4.1 System Architecture

Diagrams of system architecture represent component architecture in an abstract manner. Component-component relationships and system functioning can be facilitated by providing a succinct description of the system's component architecture. A system architecture diagram shows how the system architecture is organized. In this diagram, the system components are shown in relation to each other and what functions they perform. In the general system representation, the major functions of the system and the relationships between its components are shown. When designing their new system, the team will use a common system architecture diagram as a reference. Creating a common architecture diagram gives the team a good starting point. Besides providing a common language for communicating system design, it also allows for tracking system status. This proposed system has multiple components, including a database, an advertiser, a similarity analysis, and an ad recommendation system.

To get a complete view of entities, this recommendation system and application consists of multiple components. In this, the advertiser uploads the ads in the application via the interface into the database. Data from the database is sent to the Similarity score determiner. In this case, the similarity score is then calculated based on the Euclidean distance and the cosine method.

This similarity score is then fed as a parameter to the recommendation system along with other parametric attributes. In the end, a recommendation is given to the affiliates as a result of this whole process.

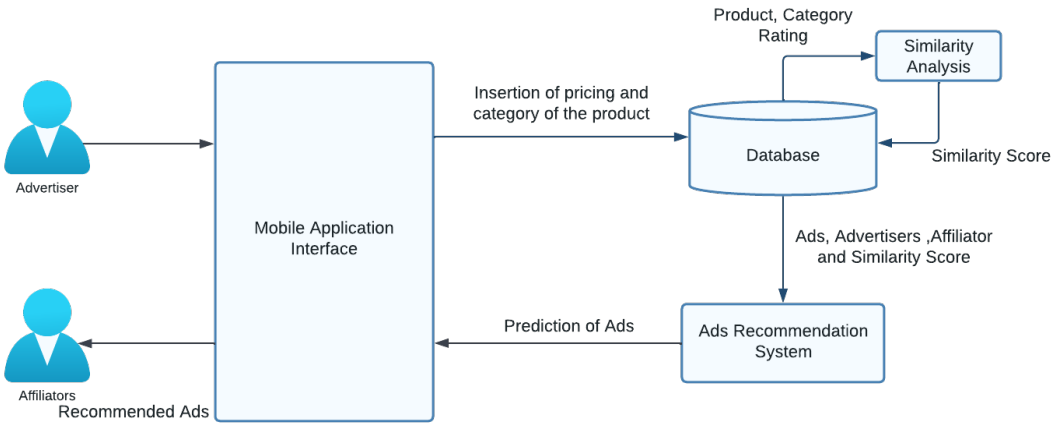


Figure 4.1: System Architecture

4.2 Mathematical Model

In order to estimate unknown user-item ratings in a collaborative filtering algorithm, a mathematical model can be used.
The first step is to represent user-item interactions as a matrix R, where m is the number of users and n is the number of items. Next, the similarity between users must be computed using a similarity matrix S, where each element S_{ij} represents the similarity between users i and j.

The cosine similarity measure is commonly used, defined as

$$S_{ij} = (r_i.r_j)/(||r_i|| * ||r_j||)$$

where r_i and r_j are the rating vectors of users i and j, respectively, and $||r_i||$ and $||r_j||$ are their Euclidean norms. The k-nearest neighbors are then found for each user i, with the k most similar users based on the values in S.
This set of k-nearest neighbors is denoted as N(i). Predicting user-item ratings involves using the ratings of the nearest neighbors to estimate unknown ratings. Specifically, for each user-item pair (i,j) where r_{ij} is unknown, the rating is predicted using the formula

$$r_{ij} = (1/|N(i)|) * SUM\{uinN(i)\}(s\{iu\} * r\{uj\})$$

where $|N(i)|$ is the number of nearest neighbors for user i, and SUM{u in N(i)} represents the sum of the ratings of user u for item j, weighted by their similarity to user i.

Finally, the accuracy of the predicted ratings can be evaluated using suitable evaluation metrics such as the root mean squared error (RMSE).

4.3 Data Flow Diagram

An information system's data flow is represented graphically using data flow diagrams. Data flow diagrams illustrate the processes involved in moving data from the input to the file storage and reporting systems. A data flow diagram can be logical or physical. In a logical data flow diagram, data flow through a system to perform certain business functions. A logical data flow diagram is illustrated in a physical data flow diagram.

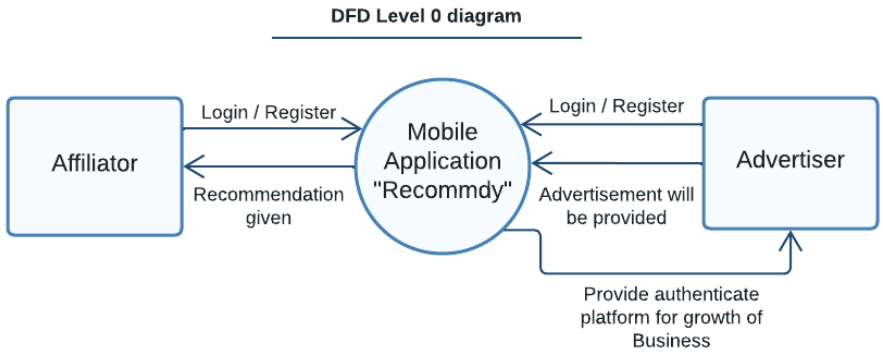


Figure 4.2: DFD Level- 0 Diagram

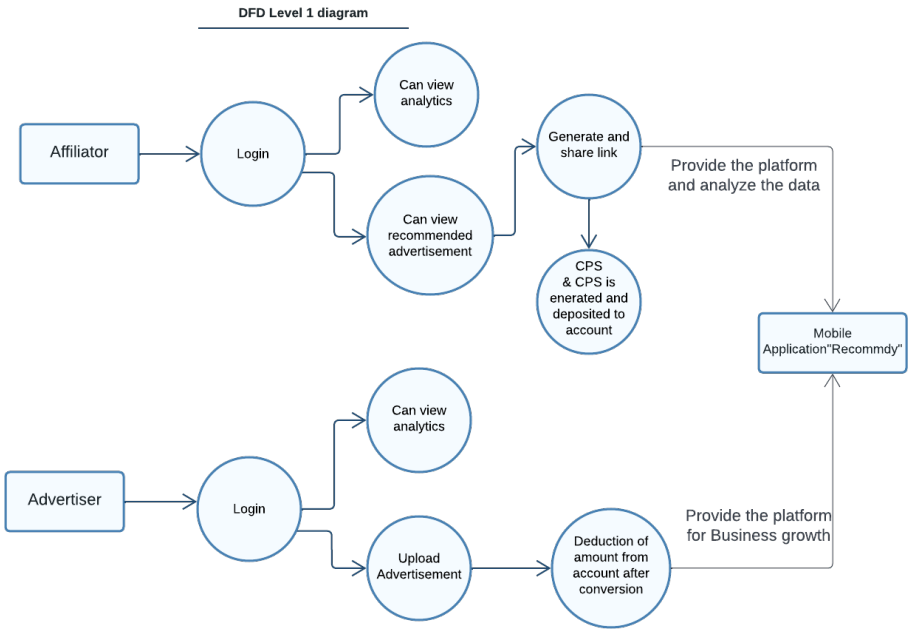


Figure 4.3: DFD Level- 1 Diagram

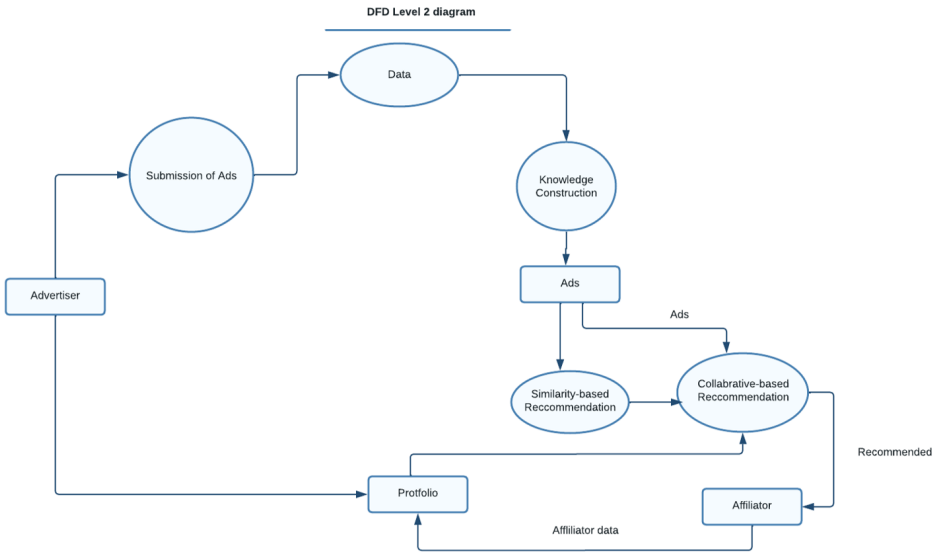


Figure 4.4: DFD Level- 2 Diagram

4.4 UML Diagrams

4.4.1 Activity Diagram

An activity diagram is based on the state of flow and the order in which it occurs. Using an activity diagram, we can explain or show what leads to a specific event. Structure diagrams, interaction diagrams, and behaviour diagrams are the three main types of diagrams that UML describes.
An activity diagram is a behavioural diagram, meaning it shows how a system behaves. An activity diagram shows the numerous decision routes that are available throughout the execution of an activity. It depicts the control flow from a start point to an endpoint.
Using an activity diagram, we can show both concurrent and sequential processing of activities. They are mostly used to represent the dynamic elements of a system in business and process modelling.
An activity diagram is essentially a flowchart that shows how one activity leads to another. System operation refers to action. One operation leads to the next in the control flow. This flow may be parallel, contemporaneous, or branched. This diagram shows how a user logs into the system as an authorized user and progresses through logging out of the system. This is a diagram showing the steps a user takes to interact with the system. It is a diagrammatic representation of the features that the user can access until the session expires or he logs off.

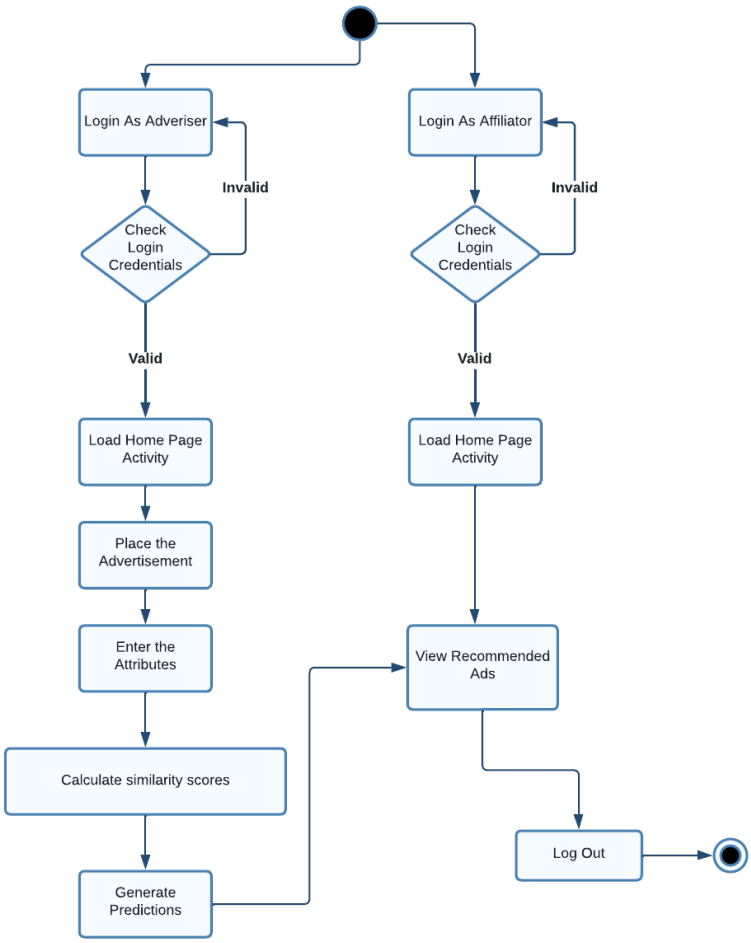


Figure 4.5: Activity Diagram

4.4.2 Object Diagram

Since object diagrams are derived from class diagrams, they are dependent upon them. An object diagram is an instance of a class diagram [7]. A class diagram and an object diagram have similar concepts. Object diagrams also depict a static view of a system, but this static view is a snapshot of the system at a particular point. We have considered four classes derived from the class diagram and given the latest instance of each class. Object attributes include advertisement id, product name, etc.

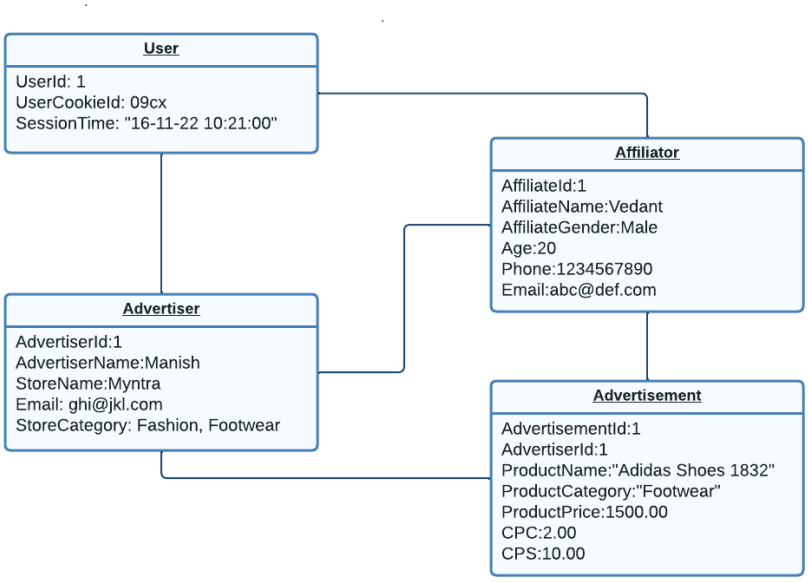


Figure 4.6: Object Diagram

4.4.3 Class Diagram

Class diagrams show the static view of an application. The class diagram is not only used for visualizing, describing, and documenting the various aspects of a system but is also used for building the executable code of the software application. Diagrams describe the attributes and operations of classes as well as the constraints that apply to them. Since class diagrams are the only UML diagrams that can be directly mapped to object-oriented languages, they are widely used in the modeling of object-oriented systems. A class diagram shows a set of classes, interfaces, associations, collaborations, and constraints. Diagrams of this type are also called structural diagrams. There are four major diagrams in this, namely user, affiliate, advertiser, and advertisement. Every class has definite methods allotted, and the functionality is determined by the way and sequence in which these methods are executed.

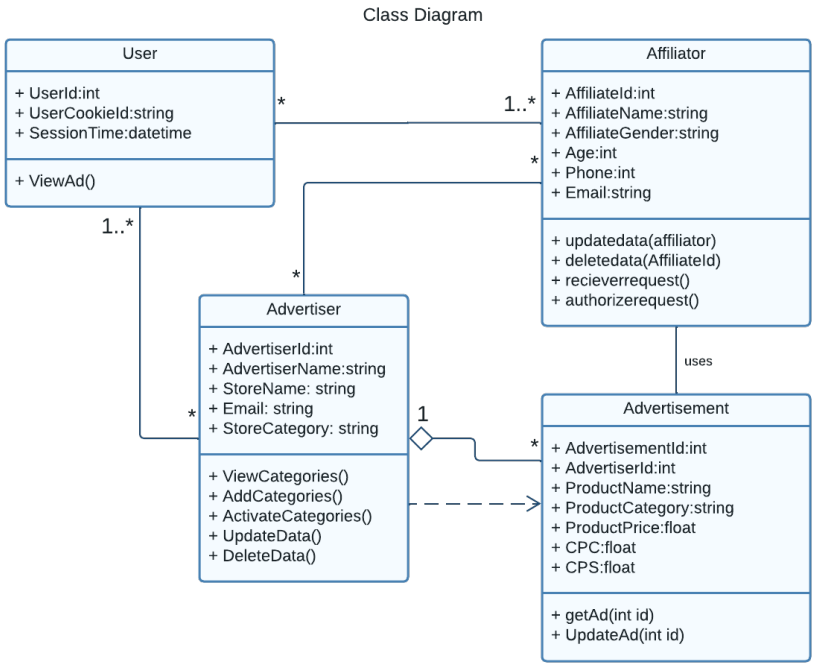


Figure 4.7: Class Diagram

4.4.4 Package Diagram

Package diagrams are structural diagrams that display how different model pieces are arranged and organized into packages. An assortment of similar UML elements, such as diagrams, documents, classes, or even other packages, are grouped together as a package. Each piece is nestled within the package, which is shown in the figure as a file folder, and then placed hierarchically. To visually organize the layered architecture within any UML classification, such as a software system, package diagrams are most frequently employed. A package diagram is a UML (Unified Modeling Language) design for grouping elements and specifying their interdependencies (packages). Package diagrams' major objective is to decompose the intricate class diagrams that can be used to organize classes into packages. The hierarchy is defined in part by these groups. It is important to note that the package's UML parts are founded on logical relationships. A package diagram is a collection of predefined pieces that are connected semantically and may be altered collectively. It is the classification of the relationships between model components and their grouping. A package diagram functions somewhat as a container for model components. However, the package might only include a component name or even the full diagram.

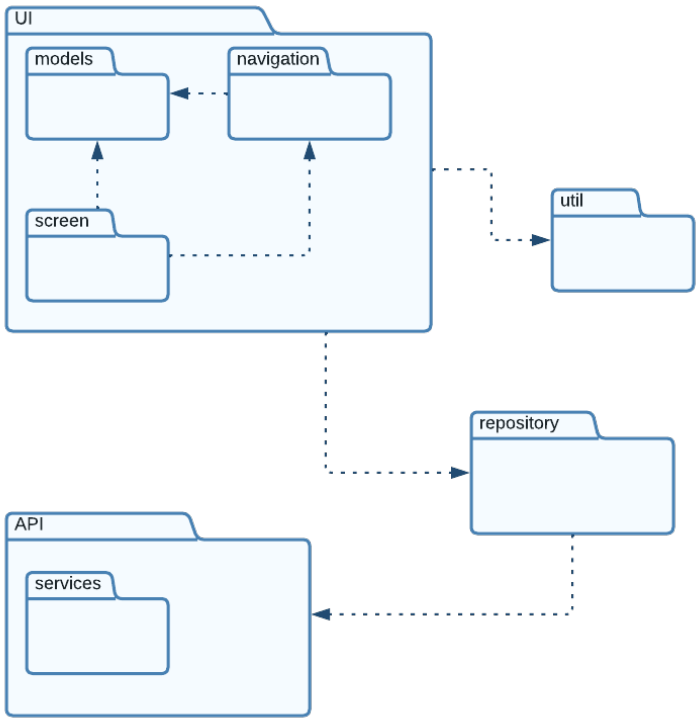


Figure 4.8: Package Diagram

4.4.5 Use Case Diagram

A use case diagram illustrates the dynamic behavior of a system. Using use cases, actors, and their relationships, it encapsulates the system's functionality. Systems/subsystems of an application are represented by tasks, services, and functions. The diagram depicts the high-level functionality of a system as well as how the user interacts with it. There are three actors in our system: the advertiser, the affiliate, and the user. There are multiple functions encapsulated in them, such as login, authentication, session management, and role checking.

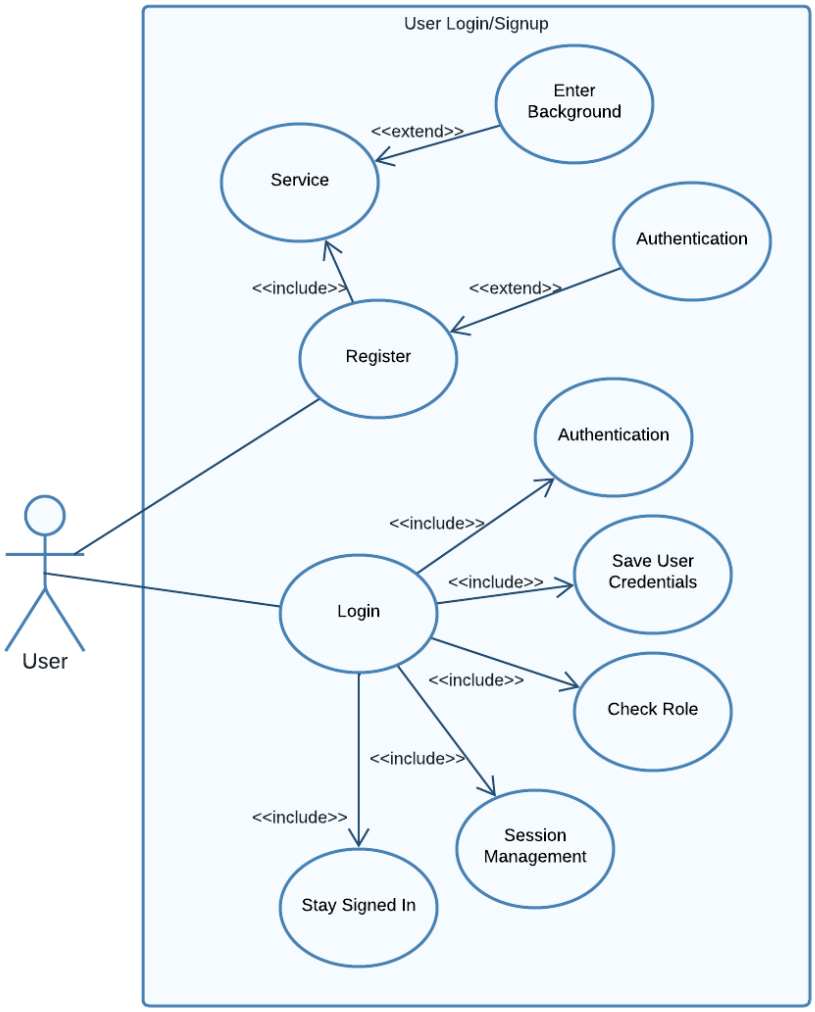


Figure 4.9: Use Case Diagram

4.4.6 Sequence Diagram

The most used interaction diagram is a sequence diagram. The interaction diagram illustrates the interactive behavior of a system. It is not easy to visualize the interactions in a system, so we use different types of interaction diagrams to capture the various features and aspects of interaction. In a sequence diagram, interactions between objects are shown in sequential order. A sequence diagram can also be referenced as an event diagram or event scenario. Sequence diagrams show the actions taken by the components of a system in chronological order. Business people and software engineers frequently use these diagrams to record and analyze the requirements for new and current systems. The sequence diagram, also known as an event diagram, shows how messages move through the system. It aids in creating a variety

of dynamic settings. It depicts communication between any two lifelines as a chronologically ordered series of activities, implying that these lifelines were active now of communication. In UML, a message flow is represented by a vertical dotted line that crosses the lifeline, which is represented by a vertical bar.

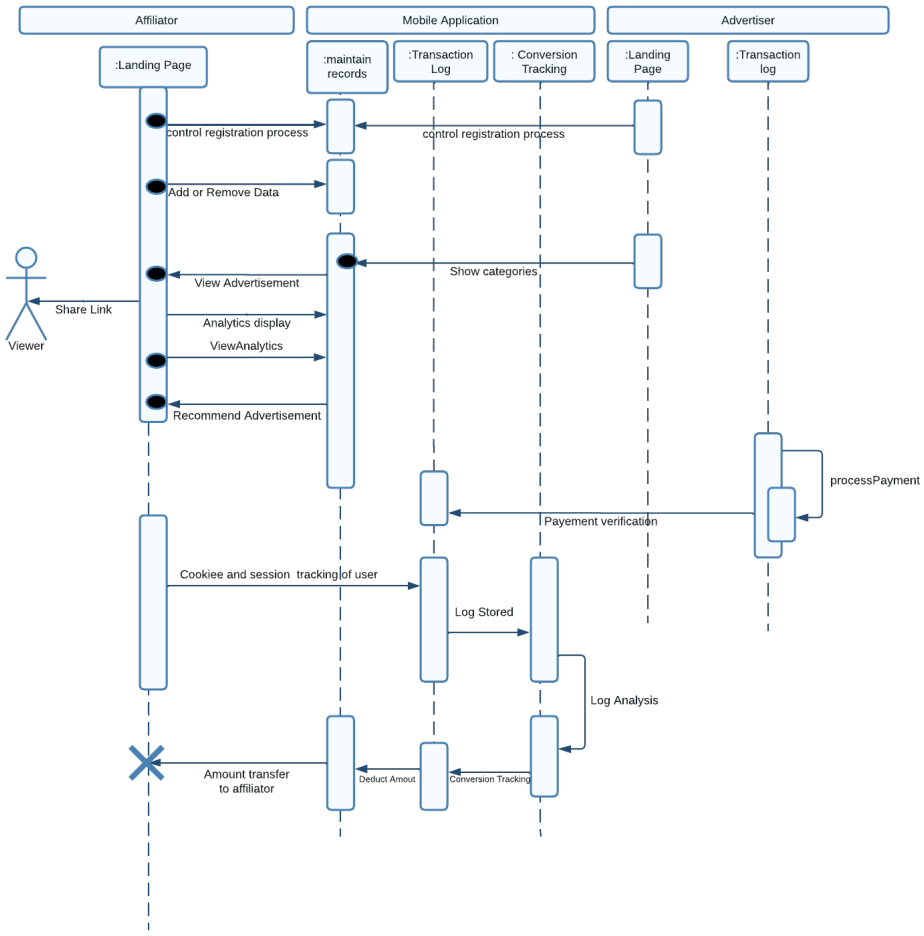


Figure 4.10: Sequence Diagram

4.4.7 State Diagram

A state diagram, also known as a state machine diagram or a state chart diagram, is a representation of the states that an object in the Unified Modelling Language can reach as well as the transitions between those states. A state, which in this context refers to a particular entity in a program or the unit of code that represents that entity, defines a stage in the evolution or behaviour of an object. A state diagram is used to depict the status of a system or a component of a system at specific points in time. It is a behavioural diagram that uses finite state transitions to depict the behaviour. State machines and state-chart diagrams

are other names for state diagrams. These words are frequently used in the same sentence. Simply said, a state diagram is used to represent how a class will behave dynamically over time and in reaction to changing external stimuli. While we can claim that every class has a state, not all classes are modelled using state diagrams. The structure of a state diagram is similar to that of a flowchart, although a flowchart depicts system operations rather than actual state changes that affect an object's state. Finding a system's beginning and end states is the first step in producing a state chart diagram. Then, all potential existing states are positioned in reference to the start and finish. Last but not least, transition components are used to describe all of the events that result in state changes.

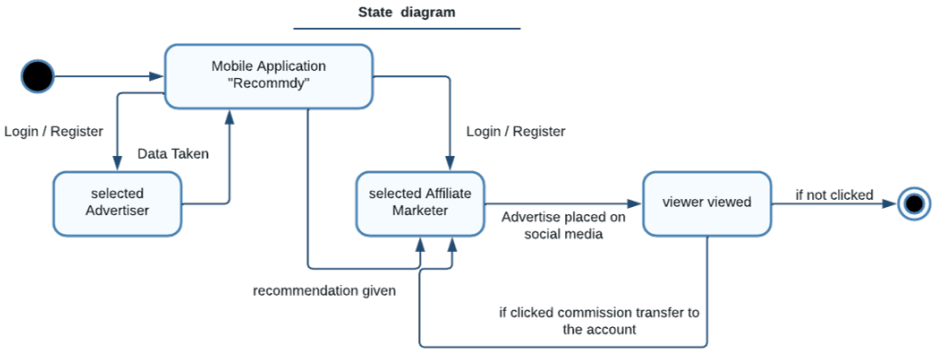


Figure 4.11: State Diagram

4.5 Component Diagram

The nature and behaviour of component diagrams vary. Component diagrams are used to represent the system's physical components. What exactly are these physical aspects, then? The things that are physically present in a node include executable, libraries, files, documents, etc. The arrangement and connections between the components in a system are depicted using component diagrams. Systems that can be executed are also created using these diagrams. To make a complex object-oriented system more understandable, the smaller components are separated out using a component diagram. It simulates the physical view of a system, including its internal node's executables, files, libraries, etc. It depicts the connections and hierarchies that exist between the system's components. It aids in creating an operational system. An individual, replaceable, and executable system unit is referred to as a component. A component's implementation details are concealed, therefore an interface is required to carry out a function. It functions like a "black box," with the provided and necessary interfaces explaining its behavior.

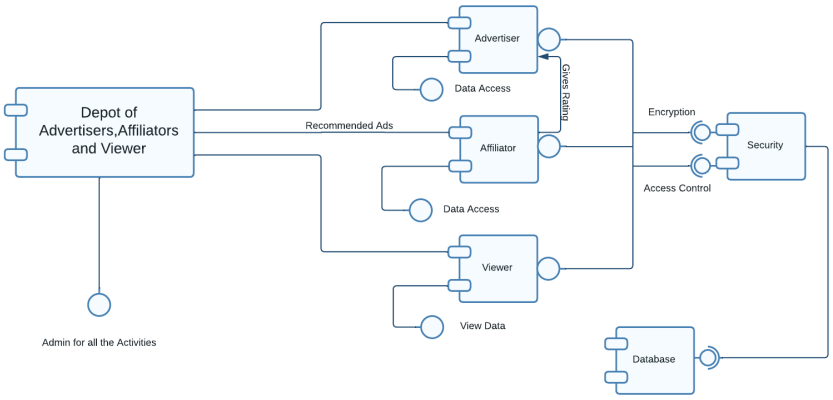


Figure 4.12: Component Diagram

4.6 Deployment Model

The topology of the physical parts of a system, where the software components are installed, is depicted using deployment diagrams. The static deployment view of a system is described using deployment diagrams. Nodes and their connections are the main components of deployment diagrams. The physical gear that will be used to run the program is shown in the deployment diagram. It depicts a system's static deployment view. It involves the nodes and the connections between them. It determines the software deployment strategy on the hardware. It connects the design-created software architecture to the actual system architecture, where the software will run as a node. Communication channels are used to demonstrate the link because there are numerous nodes involved.

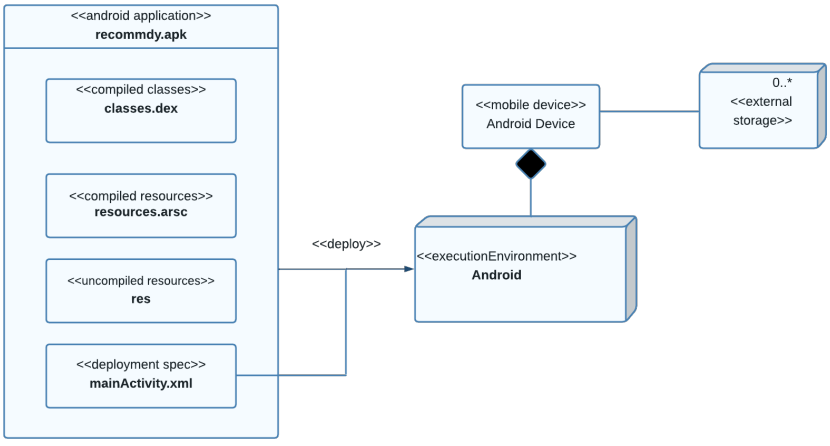


Figure 4.13: Deployment Diagram

Chapter 5

Project Plan

5.1 Project Estimates

Table 5.1: Project Estimates

Project Task	Estimate Date
Literature Survey	30 days
Requirement gathering	15 days
Base paper study	5 days
Designing of project	15 days
UML diagrams	15 days
Backend development	60 days
Frontend development	60 days
Validation and Testing	30 days
Deployment	7 days
Documentation	20 days

5.1.1 Reconciled Estimates

Now, breaking down the estimates into more detail we will see reconciled estimates first. Using the Constructive Cost Model (COCOMO) for estimating the efforts required in completing the project. Like all the estimation models COCOMO first model requires the sizing

of the information.
The information can be specified in the form of:

- Object point
- Function Point (FP)
- Lines of Source Code (KLOC)

There are different models of COCOMO but we are using Organic model.
Organic Model: A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem. [8]

1. **Effort Calculation:**
   Effort = a(KLOC) $\wedge b$
   Effort = 2.4(15) $\wedge$1.05
   Effort = 41.2199 Person-Month

2. **Time Calculation:**
   Time = c(Effort) $\wedge d$
   Time = 2.5(41.2199) $\wedge$0.38
   Time = 10.2726 Months

3. **Person Required:**
   Person Required = Effort/Time
   Person Required = 41.2199 / 10.2726
   Person Required = 4.012 Persons

4. **Project Cost:**
   Project Cost = Assumed Salary * Person Required
   Project Cost = 15,000 * 4
   Project Cost = Rs. 60,000

5.1.2 Project Resources

1. **Human Resources**
   (a) **Development Team:** Depending on the size and complexity of your project, you may need software engineers, mobile app developers, backend developers, UI/UX designers, and QA/testers.
   (b) **Affiliate Marketing Specialist:** If you don't have expertise in affiliate marketing, consider bringing in a specialist who can guide you in establishing partnerships and optimizing your affiliate marketing strategy.

2. **Hardware and Software**
   (a) **Development Machines:** Each team member will need a reliable computer or laptop with the necessary hardware specifications to support the development environment.
   (b) **Development Tools:** Depending on your chosen technology stack, you may require integrated development environments (IDEs), code editors, version control systems (e.g., Git), and collaboration tools (e.g., project management software, communication tools).
   (c) **Push Notifications:** Consider using push notification services like Firebase Cloud Messaging or One Signal to engage with your app users.

3. **Designing and Assets**
   (a) **Design Software:** Tools like Adobe Photoshop, Sketch, or Figma can be used for creating UI/UX designs, wireframes, and prototypes.
   (b) **Graphic and Media Assets:** Depending on your app's requirements, you may need access to stock images, icons, fonts, or graphic design services.

5.2 Risk Management

Risk management is an essential aspect of any project to identify, assess, and mitigate potential risks that could impact the success and progress of your affiliate marketing mobile application. The risk management process can be broken down into two interrelated phases, risk assessment, and risk control. These phases are further broken down. Risk assessment involves risk identification, risk analysis, and risk prioritization. Risk control involves risk planning, risk mitigation, and risk monitoring. It is essential that risk management can be done iteratively, throughout the project, as a part of the team's project management routine.

5.2.1 Risk Identification

Risk identification is the process of determining which risks may affect the project and documenting their characteristics. Risk assessment is an important factor to prevent them. Some are listed below:

1. **Technical Risks**
   - **Risk:** Integration challenges with affiliate network APIs, compatibility issues with different mobile devices and operating systems, or performance bottlenecks.
   - **Mitigation:** Thoroughly research and test the APIs and ensure compatibility with your chosen development framework. Conduct comprehensive testing on various devices and use performance optimization techniques during development.

2. **Security Risks**
   - **Risk:** Potential vulnerabilities in data storage, payment processing, or user authentication mechanisms that could lead to data breaches or unauthorized access.
   - **Mitigation:** Implement industry-standard security measures such as encryption, secure communication protocols, robust user authentication mechanisms, and regular security audits. Stay updated with security best practices and address any identified vulnerabilities promptly.

3. **Legal and Compliance Risks**
   - **Risk:** Non-compliance with relevant laws and regulations, such as data privacy (GDPR, CCPA), advertising guidelines, or affiliate program terms.
   - **Mitigation:** Stay updated with legal requirements and regulations relevant to your application. Comply with data protection regulations, disclose affiliate partnerships transparently, and adhere to advertising guidelines. Consult with legal experts if necessary.

4. **Affiliate Partner Risks**
   - **Risk:** Unreliable or non-compliant affiliate partners, delayed commission payments, or disagreements on tracking and attribution.
   - **Mitigation:** Conduct thorough due diligence on potential affiliate partners, including their reputation, track record, and terms of the agreement. Clearly define expectations, establish strong communication channels, and have proper tracking and reporting mechanisms in place.

5.2.2 Risk Analysis

A risk analysis performed during software testing helps to identify areas where software flaws could result in serious issues in the product. By identifying areas of concern early, we will be able to proactively remediate and reduce the overall risk of production defects. Risk analysis involves examining how project outcomes and objectives might change due to the impact of the risk event. Once the risks are identified, they are analyzed to identify the qualitative and quantitative impact of the risk event. Once the risks are identified, they are analyzed to identify the qualitative and quantitative impact of the risk on the project so that appropriate steps can be taken to mitigate them.

1. **Technical Risks**
   - **Impact:** Delays in development, limited functionality, or compromised user experience.
   - **Likelihood:** Moderate, as integrating with third-party APIs can pose technical complexities.

2. **Security Risks**
   - **Impact:** Damage to reputation, loss of user trust, and potential legal consequences.
   - **Likelihood:**High, as security threats are prevalent in today's digital landscape.

3. **Affiliate Partner Risks**
   - **Impact:** Inaccurate tracking, delayed payments, or damage to the user experience.
   - **Likelihood:** Moderate, as the reliability of affiliate partners may vary

Table 5.2: Risk Prioritization

Risk Name	Probability of Occurrence	Priority
Data confidentiality/breach issues	High	1 (Highest)
Database connectivity failure	High	2
Improper integration of modules	High	3
Data hacking/malware	High	4
The constant change in requirements	Medium	5
JAVA SDK incompatibility/fatal bugs	Medium	6
Cascading delays in development	Medium	7
Development Tool issues	Low	8
Asynchronous DOM loading	Low	9
Insufficient hard disk space	Low	10 (Lowest)

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Risk mitigation, monitoring, and management are essential components of a successful project.

Risk Mitigation:

1. Risk mitigation involves taking proactive measures to reduce the likelihood and impact of identified risks.
2. It aims to minimize the occurrence of risks or their potential consequences.
3. Some common risk mitigation strategies include:
   (a) Implementing preventive controls and security measures.
   (b) Conducting thorough testing and quality assurance to identify and resolve issues early.
   (c) Diversifying partnerships and resources to reduce dependency on a single entity.
   (d) Establishing contingency plans to address potential risks.
   (e) Ensuring compliance with legal and regulatory requirements.
   (f) Regularly updating and patching software systems to address vulnerabilities.
   (g) Conducting risk workshops or brainstorming sessions to identify and address potential risks proactively.

Risk Monitoring:

1. Risk monitoring involves ongoing tracking and evaluation of identified risks throughout the project lifecycle.
2. It ensures that risks are monitored, and their status is regularly assessed.
3. Key activities in risk monitoring include:
   (a) Regularly reviewing the risk register to assess the status of identified risks.
   (b) Monitoring key risk indicators to detect early warning signs.
   (c) Conducting periodic risk assessments to identify new risks or changes in existing risks.

Risk Management:

1. Risk management encompasses the overall process of identifying, analyzing, mitigating, and monitoring risks.
2. It involves a systematic approach to address risks and ensure project success.
3. Key activities in risk management include:
   (a) Creating a risk management plan that outlines the risk management process and responsibilities.
   (b) Identifying and documenting risks, including their likelihood, impact, and potential mitigation strategies.

(c) Developing and implementing risk mitigation strategies and action plans.
(d) Assigning responsibilities for risk management activities to team members.

5.3 Project Schedule

5.3.1 Project Task Set

Following is the project task set:

1. Carrying out a literature survey
2. Study of base Paper
3. Implementation of Project with base project functionalities and added-on features.
4. Affiliate Network Integration
5. Continuous error discovery and debugging
6. Testing the system for various test cases
7. Documentation and report generation

5.3.2 Task Network

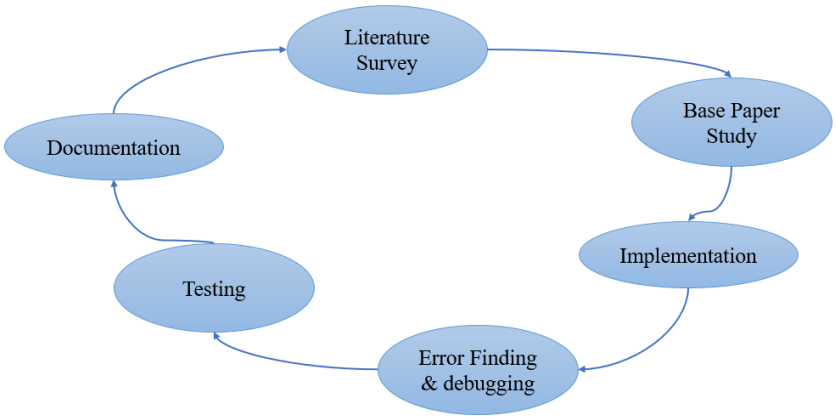


Figure 5.1: Task network

5.3.3 Timeline Chart

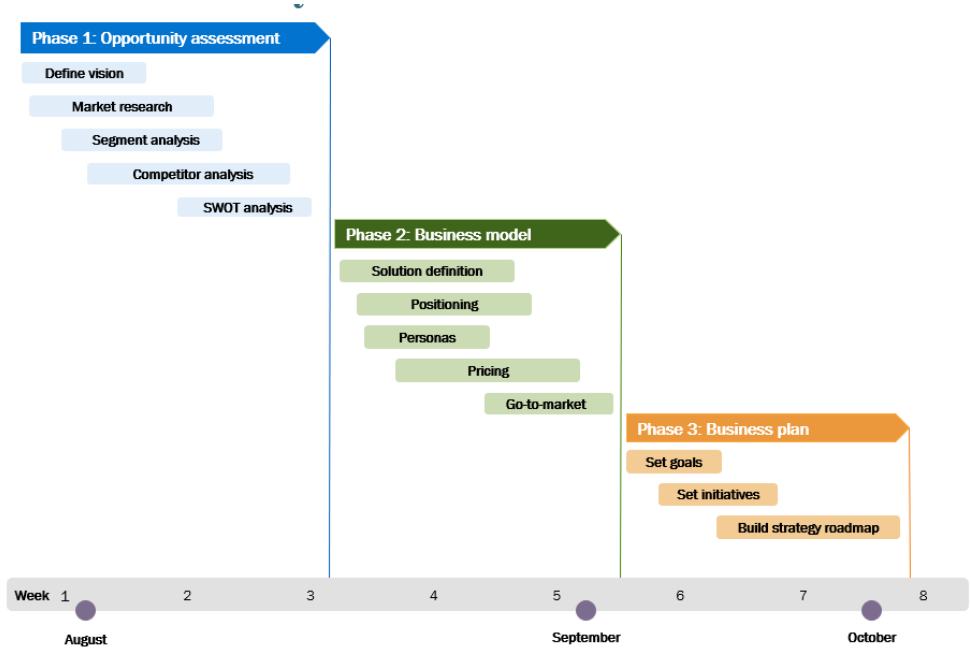


Figure 5.2: Timeline Chart 1

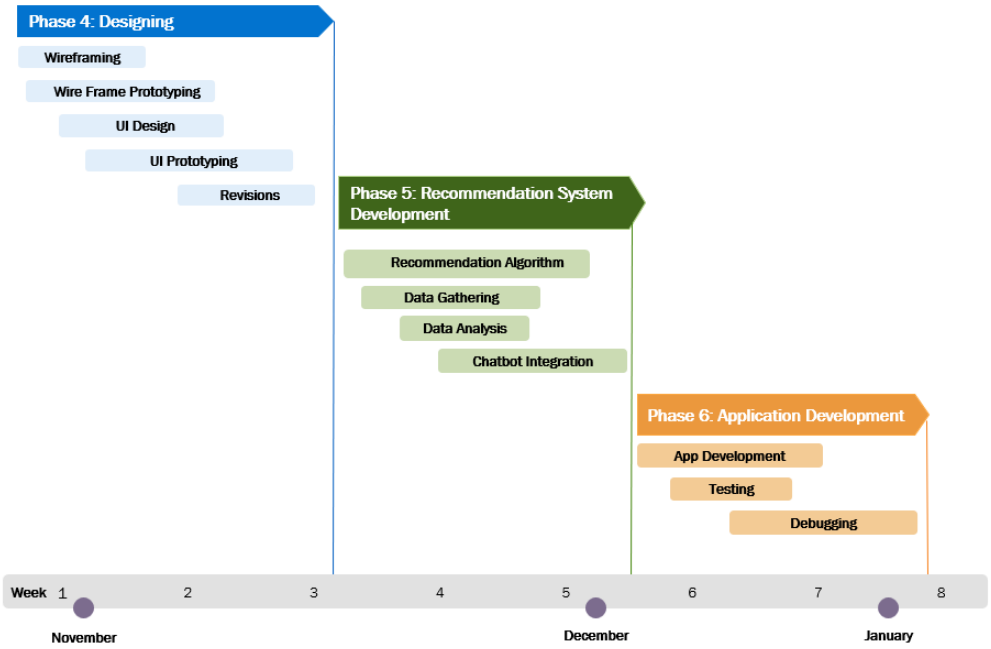


Figure 5.3: Timeline Chart 2

5.4 Team Organizations

Team organization is crucial for the successful development of your affiliate marketing mobile application. Proper project team organization is one of the key constraints to project success. If the project has no productive and well-organized team, there is an increased probability that this project will be failed at the very beginning because initially, the team is unable to do the project in the right manner. Without the right organization of teamwork, people who form the team will fail with performing several specific roles and carrying out a variety of group/individual responsibilities. Hence, when you plan for a new project, first you must take care of the best project team organization through team-building activities. A Project Team is an organized group of people who are involved in per- forming shared/individual tasks of the project as well as achieving shared/individual goals and objectives for the purpose of accomplishing the project and producing its results. Two Conventional Roles Every team, regardless of the project type, size, and nature, has two roles (defined as "conventional"). These roles are:

1. **Backend Developer:**
   - **Responsibilities:** Design and develop the backend infrastructure of the mobile application. Set up the server environment and database systems. Implement APIs for data retrieval, user authentication, and affiliate network integration. Handle data modeling and database management. Integrate payment gateways for transactions and commissions. Conduct thorough testing and debugging of backend functionalities.

2. **UI/UX Designers:**
   - **Responsibilities:** Develop test plans, test cases, and test scenarios. Perform functional and non-functional testing of the application. Conduct compatibility testing across various devices and operating systems. Validate affiliate tracking and commission calculations. Identify and report bugs and issues, and work with developers to resolve them. Conduct regression testing after bug fixes and updates.

3. **QA/Testers:**
   - **Responsibilities:** Create visually appealing and user-friendly UI/UX designs. Develop wireframes, mock-ups, and interactive prototypes. Ensure consistent branding and design elements throughout the application. Collaborate with developers to implement the UI designs. Conduct user testing and gather feedback to improve the user experience. Iterate designs based on user feedback and project requirements.

4. **Leader:** A project team leader is a person who provides leadership and guidance to the team and takes responsibility for the results of teamwork. The team leader role involves the development and encouragement of the team through training, leading,

motivation, recognition, rewarding and other activities that stimulate or force team members to do the required tasks.

5. **Member:** A project team member is a person who is involved in doing assigned tasks. Team members directly access the project and actively evolve its processes. They are subordinate to the team leader.

5.4.1 Management Reporting and Communication

Management reporting and communication are essential aspects of keeping stakeholders informed about the progress, status, and key metrics of your affiliate marketing mobile application project. We focused on our intra communication which was by far fluent in order to interact while gathering the information. Each member voiced their problems and these were solved with the help of each member. We distributed the task among us equally. Our teamwork is evident from the report we share. Our communication with our guide Prof R.R Shevale is good. We reported timely to show our project process. We consult them when in doubt and she guided us every well.

5.4.2 Team Structure

Following is the team structure:

Table 5.3: Risk Prioritization

Name	Designation	Work Done
Aditya Kishor Shinde	Team Leader	• Requirement analysis • Base paper study • Component diagram • Wireframing • UI designing • Frontend App development • Error Debugging • Report generation

Table 5.4: Risk Prioritization

Name	Designation	Work Done
Ankita Ajit Kulkarni	Team Member	• Requirement analysis • Design analysis • System Architecture • Activity Diagram • Admin panel development • Documentation • Error Debugging • Report generation
Tanaya Bhushan Gavande	Team Member	• Literature survey • Mood board designing • PPT Presentation • Use Case Diagram • Backend development • Recommendation system • Testing • Report generation
Krutika Ravindra Rawal	Team Member	• Literature survey • Mood board analysis • Base paper study • Sequence Diagram • DFD diagram • Testing • Report generation • Documentation

Chapter 6

Project Implementation

6.1 Overview of Project modules

1. **Authentication Module:** This module is responsible for user registration, login, and authentication. It includes features such as email verification and password reset.
   (a) **User Registration:** Allows users to create new accounts by providing necessary information.
   (b) **Login:** Provides a secure login mechanism for users to access the application, with email or Google account.
   (c) **Email Verification:** Verifies the user's email address to ensure account authenticity.
   (d) **Password Reset:** Allows users to reset their passwords in case they forget or need to change them.
2. **Advertiser Module:** This module is designed for advertisers who want to post advertisements for their products or services. It includes features such as advertisement posting.
   (a) **Advertisement Posting:** Enables advertisers to create and publish advertisements for their products or services.
   (b) **Category Management:** Advertisers can categorize their products or services within the advertiser module. They can create and manage custom categories and subcategories that align with their business offerings, allowing for easier organization and navigation of products.
   (c) **Dashboard:** The dashboard provides an overview of key metrics and statistics related to the advertiser's account, such as total products, websites, coupons, users, and campaign performance. It serves as a centralized hub for monitoring and analyzing advertising activities.

3. **Affiliate Marketer Module:** This module is designed for affiliate marketers who want to promote products or services through their network. It includes features such as affiliate link generation and uses history.
   (a) **Affiliate Link Generation:** Generates unique affiliate links for affiliate marketers to promote products or services.
   (b) **Featured Products Based on User Share History:** The module incorporates a recommendation system that analyzes the user's share history and identifies trending or popular products that align with their interests.
   (c) **Integration with Various Platforms:** The module facilitates seamless integration with various platforms such as news websites, social media platforms, food delivery apps, and more.
   (d) **Support Chat:** The Support Chat sub-module provides affiliate marketers with a direct communication channel to connect with the support team. It enables them to seek assistance, ask questions, and receive timely support for any issues or queries they may have.
   (e) **Real-time Messaging:** The Group Chat enables instant messaging with the admin, allowing affiliate marketers to have discussions, seek advice, and share valuable insights. They can exchange information, tips, and best practices, fostering a collaborative environment.
   (f) **KYC (Know Your Customer):** The KYC sub-module is designed to ensure compliance with regulatory requirements and maintain the integrity of the affiliate marketing system. It involves a verification process where affiliate marketers are required to provide valid identification documents and personal information to verify their identity and legitimacy.
4. **Advertisement Module:** This module manages all the advertisements posted by the advertisers. It includes features such as ad categorization, search functionality, and a recommendation engine.
   (a) **Ad Categorization:** Organizes advertisements into different categories for easy navigation and filtering.
   (b) **Search Functionality:** Enables users to search for specific advertisements based on keywords.
   (c) **Recommendation Engine:** Utilizes user historical data to recommend personalized advertisements to users.
5. **Notification Module:** This module sends notifications to users regarding new advertisements and other important updates. It includes features such as push notifications.
   (a) **Push Notifications:** Sends push notifications to users' devices to alert them about new advertisements, earnings, and important updates.

6. **History Module:** This module maintains a record of all the user's past activities such as ad clicks, ad views, and transactions. It includes features such as search and filter functionality and data export.

6.2 Tools and Technologies Used

Our project makes use of several tools and technologies to deliver a robust and user-friendly application. For the front end, we use the Android Studio IDE, which is the official IDE for Android app development. It offers a wide range of features and tools that make the development process efficient and effective. For the backend, we use Firebase, which is a comprehensive mobile development platform that offers a variety of tools and services such as authentication, real-time databases, and cloud storage.

6.2.1 Tools

- **Android Studio:** This is the official integrated development environment (IDE) for Android app development built and distributed by Google. It contains tools that make it easy for software developers to design, build, run, and test software, in this case, apps for the Android platform.
- **Figma:** Figma is a collaboration tool for teams and individuals to create and share high-quality work. It is used for designing and prototyping user interfaces.
- **SQL database:** SQL is a relational database management system used for storing and retrieving data. It is widely used for web applications and is known for its scalability, security, and performance.
- **GitHub:** GitHub is a web-based platform used for version control and collaborative software development. It allows developers to store and manage their code, track changes, and collaborate with other developers.

6.2.2 Technologies

- **Java:** a programming language used for developing the Android mobile application.
- **XML:** a markup language used for designing the user interface of the Android application.
- **Firebase:** a mobile and web application development platform developed by Google. It provides a variety of services including real-time databases, authentication, cloud messaging, and analytics.
- **PHP:** This is a programming language that is used for web-based applications. We have used this technology to develop the admin panel for the app.

Chapter 7

Software Testing

7.1 Type of Testing

- **Unit Testing**
  Testing a single unit or a group of related units is known as unit testing. It belongs to the category of white box testing. It is frequently used by programmers to ensure that the unit they have created is producing the expected output from supplied input. We have performed unit testing by testing each unit of our project.

- **Functional Testing**
  Functional testing is a type of testing that seeks to establish whether each application feature works as per the software requirements. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user's expectations.

- **Performance Testing**
  It is designed to test the run-time performance of software within the context of an integrated system. It is used to test the speed and effectiveness of the program. It is also called load testing. In it, we check, what is the performance of the system in the given load.

- **Usability Testing**
  Usability testing refers to evaluating a product or service by testing it with representative users. Typically, during a test, participants will try to complete typical tasks while observers watch, listen, and take notes.

- **Acceptance Testing**
  Acceptance testing is a quality assurance (QA) process that determines to what degree an application meets end users' approval. Depending on the organization, acceptance testing might take the form of beta, application, field, or end-user testing.

7.2 Test Cases and Test Result

Table 7.1: Test Cases and Test Result

Test Case Id	Test Case	Expected Result	Actual Result	Status
TC1	**User Signup Test Cases:**			
TC 1.1	User Profile Creation	Verify that users are able to create a profile successfully and input relevant information such as name, email address, and interests.	It should be logged in Successfully.	Pass
TC 1.2	Verify the user	Users are required to confirm their email address before they can log in.	It is sending mail.	Pass
TC 1.3	Poping the message in case of the profile is incomplete	Verify that users are prompted to complete their profile upon signing up.	It is giving the message.	Pass
TC2	**Affiliate Program Test Cases:**			
TC 2.1	Affiliate Link Generation	Check that affiliates are able to generate their unique affiliate link successfully.	It is generating.	Pass

Test Case Id	Test Case	Expected Result	Actual Result	Status
TC 2.2	Verify that users can access their affiliate dashboard.	It should display the user dashboard.	It is displaying.	Pass
TC 2.3	Verify that users can see their unique referral link.	It should show a unique link for each affiliate.	It is showing.	Pass
TC3	**Recommendation System Test Cases:**			
TC 3.1	Product Recommendation	Test that the app is able to suggest relevant products to users based on their interests and previous purchases.	It is suggesting.	Pass
TC 3.2	Verify that users can search for recommended products based on specific criteria, such as category.	It should give recommendations according to the category.	It is suggesting.	Pass
TC4	Affiliate Coins	Verify that the app is accurately granting the coins on sharing affiliate links.	It is showing earned coins on the profile.	Pass

Test Case Id	Test Case	Expected Result	Actual Result	Status
TC5	Push Notifications	Test that users are receiving push notifications for new product recommendations and special deals.	It is giving notification.	Pass.
TC6	Social Sharing	Verify that users are able to share affiliate links on social media platforms.	It is showing various social media.	Pass
TC7	Search Functionality	Test that users are able to search for products and services within the app.	Through keywords, it is providing results.	Pass
TC8	**User Engagement:**			
TC 8.1	Verify that the application is user-friendly and easy to navigate.	User should easily understand the application and get an idea about it.	Icons are user-friendly so it is easily understood.	Pass
TC 8.2	Verify that the application is responsive and works well on different mobile devices.	It should support all Android versions.	It is supported till Android version 12.	Pass

Test Case Id	Test Case	Expected Result	Actual Result	Status
TC 8.3	Verify that the application is fast and does not have any major bugs or glitches.	It should not crash in the middle.	It executes fully the function which is in the executing state.	Pass
TC 8.4	Verify that the application provides helpful notifications and alerts to users.	It should give various notifications.	It provides alerts regarding new deals and offers.	Pass
TC9	**Performance and Usability:**			
TC 9.1	Verify that the application loads quickly and performs well under different network conditions.	It is expected that the application should handle multiple requests.	Application is responding to all the requests in heavy traffic is there.	Pass
TC 9.2	Verify that the application displays appropriate error messages for any failures or issues encountered.	It should give an error in case any wrong input is given.	It is not allowing wrong input.	Pass

Test Case Id	Test Case	Expected Result	Actual Result	Status
TC10 TC 10.1	**Security:** Verify that user authentication and authorization mechanisms are in place and working correctly.	It should not allow the user without email verification.	Email is received ion mailbox.	Pass

Chapter 8

Result

8.1 Outcomes

The outcomes of the developed application are significant and cater to the needs of both advertisers and affiliate marketers. Based on user history, the recommendation system provides personalized recommendations to the affiliates, increasing the chances of conversion. The affiliate-generated custom links for each affiliate marketer are unique. The dynamic nature of the application ensures that the advertisements uploaded on the admin panel are displayed in the application and can be shared accordingly.

The application has a user-friendly interface that facilitates easy navigation and interaction. The category view of all the ads allows users to browse through the ads of their interest, and the payment interface ensures secure payment transactions.

The developed application meets the functional and non-functional requirements specified and provides a seamless experience to the users. It is an effective tool for advertisers to promote their products and services and for affiliate marketers to earn commissions by promoting those products and services. The application has the potential to scale and expand its user base, which can lead to increased revenue for advertisers and affiliate marketers.

8.2 Advantages

1. **Personalized recommendations:** Recommdy provides personalized recommendations based on user preferences and history, enhancing the user experience and increasing the chances of finding relevant and appealing products or services.

2. **Efficient affiliate marketing:** The app facilitates affiliate marketing by connecting advertisers with affiliate marketers, enabling them to reach a wider audience and generate more conversions and sales.

3. **Enhanced user engagement:** With features like wishlists, history tracking, and performance analytics, Recommdy enhances user engagement by allowing users to track

their activities, manage their preferences, and stay informed about their affiliate marketing performance.

4. **Streamlined ad placement:** Advertisers can easily place their advertisements through the admin panel, making it convenient and efficient to reach their target audience.

5. **User-friendly interface:** Recommdy offers a user-friendly interface that is intuitive and easy to navigate, ensuring a seamless user experience for both affiliate marketers and viewers.

6. **Increased revenue opportunities:** By leveraging the power of affiliate marketing and personalized recommendations, Recommdy creates new revenue opportunities for both advertisers and affiliate marketers, leading to potential business growth and increased profits.

7. **Cost-effective advertising:** Advertisers can benefit from the cost-effectiveness of affiliate marketing, paying commissions only when conversions or sales occur, resulting in optimized advertising budgets and increased return on investment.

8. **Convenient payment gateway:** Recommdy incorporates a payment gateway that simplifies transaction processes for both advertisers and affiliate marketers, ensuring secure and hassle-free payment transactions.

9. **Continuous improvement:** Through data analysis and feedback collection, Recommdy can continuously improve its recommendation algorithms and enhance the overall functionality and user experience of the app.

8.3 Applications

1. **Marketing industry:** Effective targeting and personalized advertisements.

2. **Customer experience:** Enhanced shopping experience through personalized ads.

3. **Affiliate marketing optimization:** Custom link generation for better tracking and optimization.

4. **E-learning platforms:** Personalized course recommendations for learners.

5. **Entertainment industry:** Personalized content recommendations for movies, TV shows, and music.

6. **Improved efficiency:** Valuable insights for various industries to enhance their operations and strategies.

8.4 Screenshots of Admin panel and Affiliate app

8.4.1 Admin panel

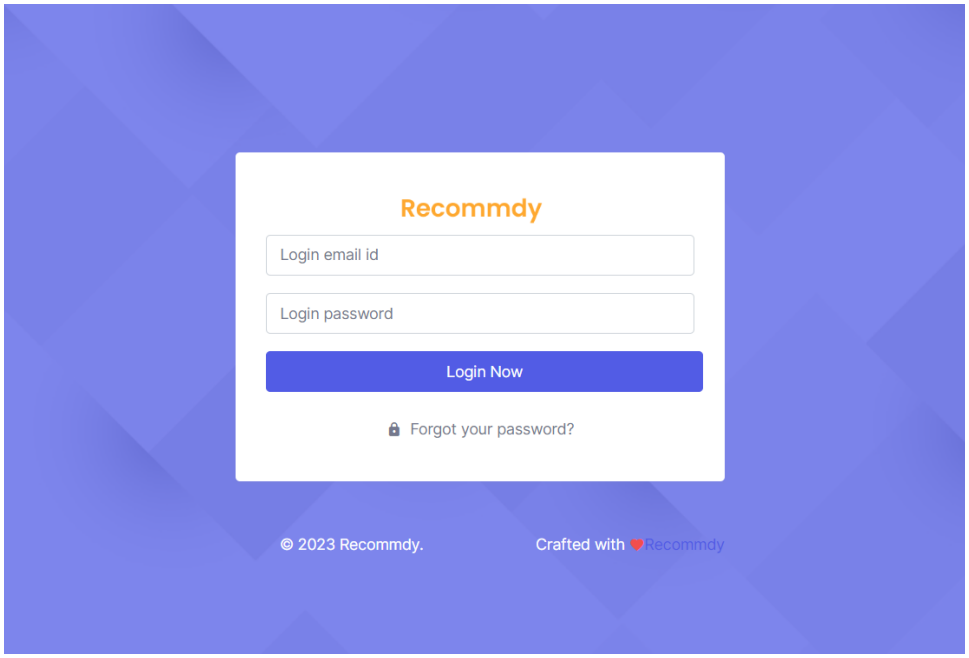


Figure 8.1: Admin panel Login Screen

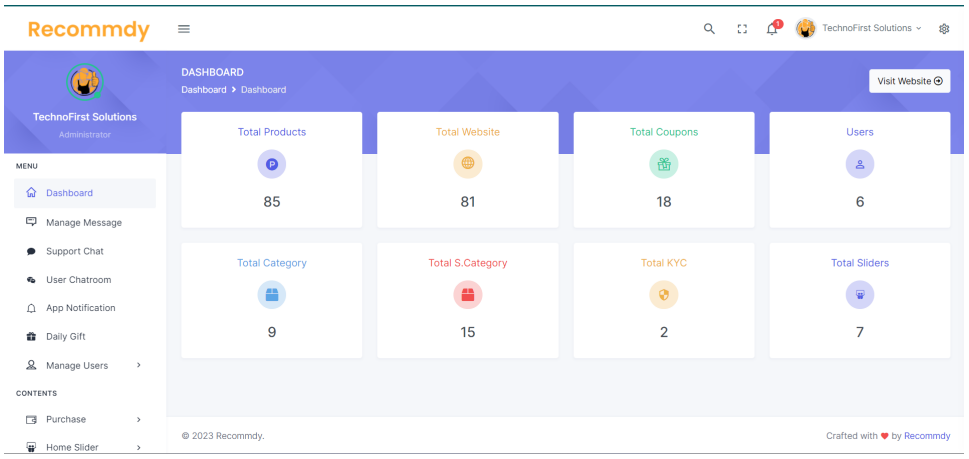


Figure 8.2: Dashboard

.

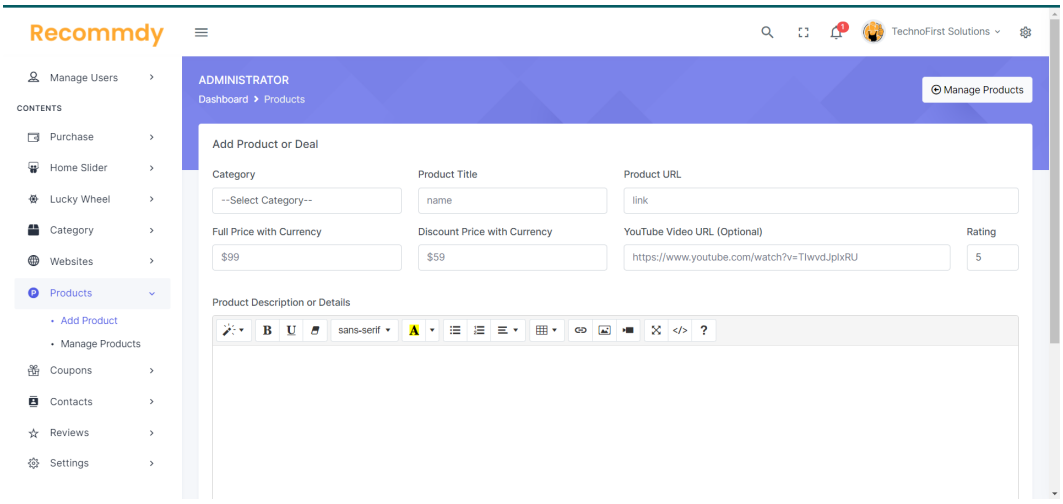


Figure 8.3: Screen to add products

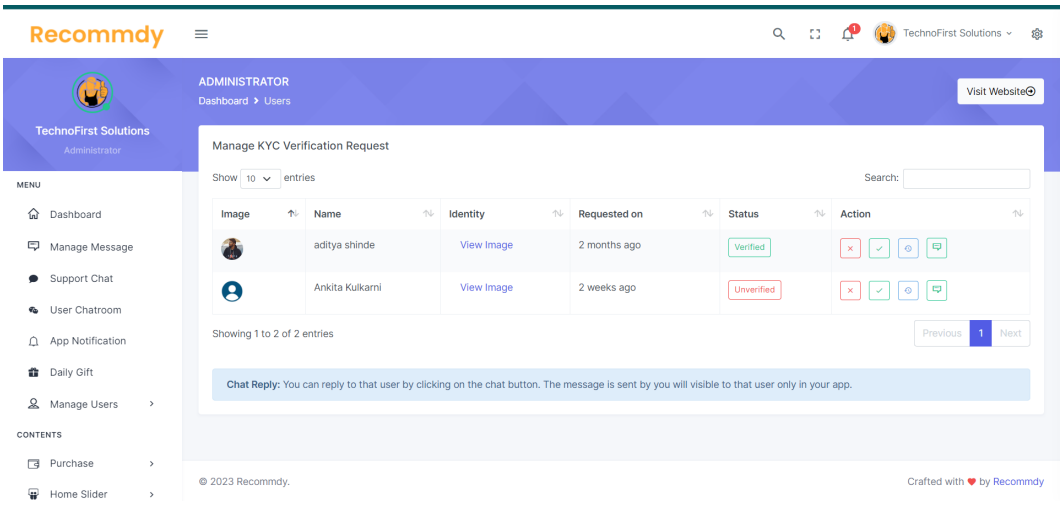


Figure 8.4: List of KYC verified Affiliates

8.4.2 Affiliates Application - Recommdy

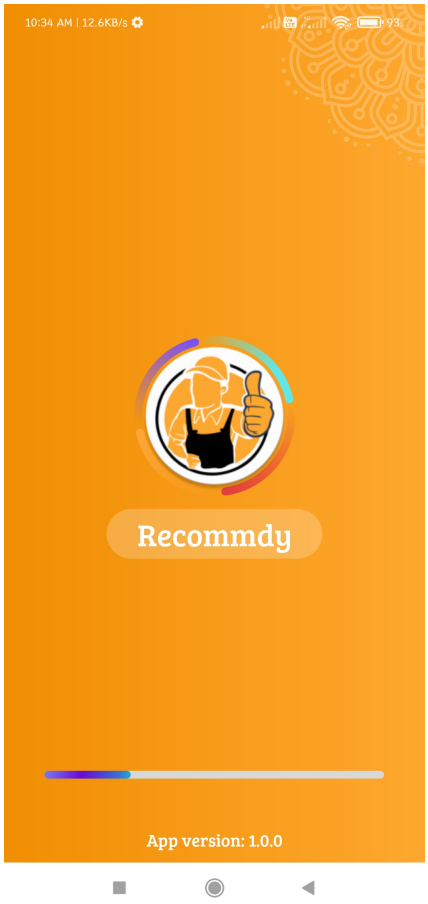


Figure 8.5: Splash Screen

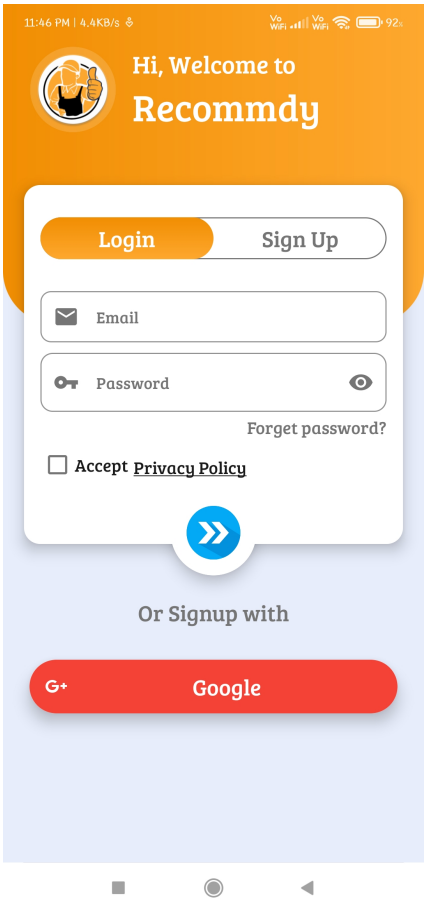


Figure 8.6: Login Screen

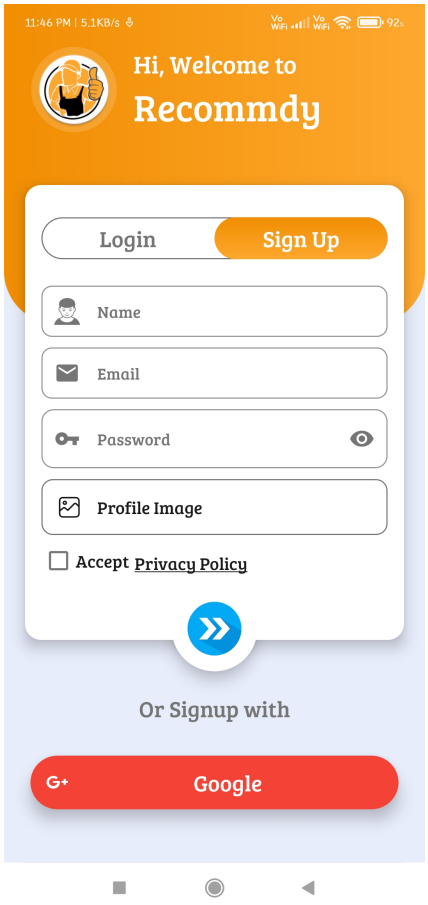


Figure 8.7: Sign Up Screen

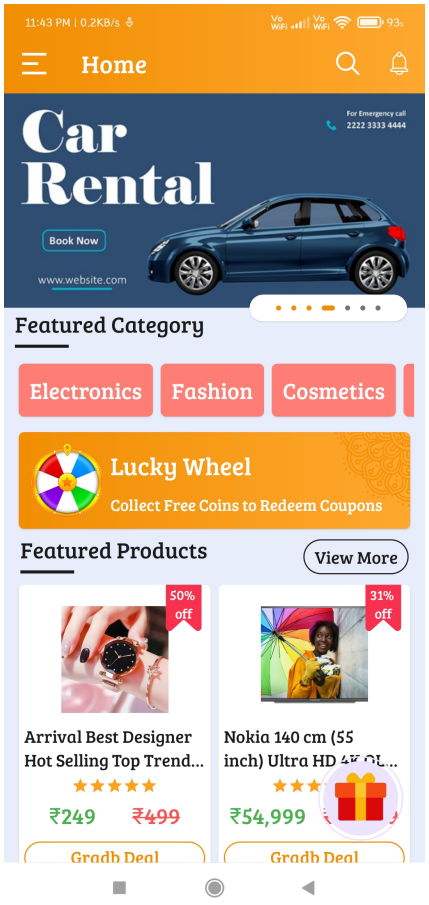


Figure 8.8: Home Screen

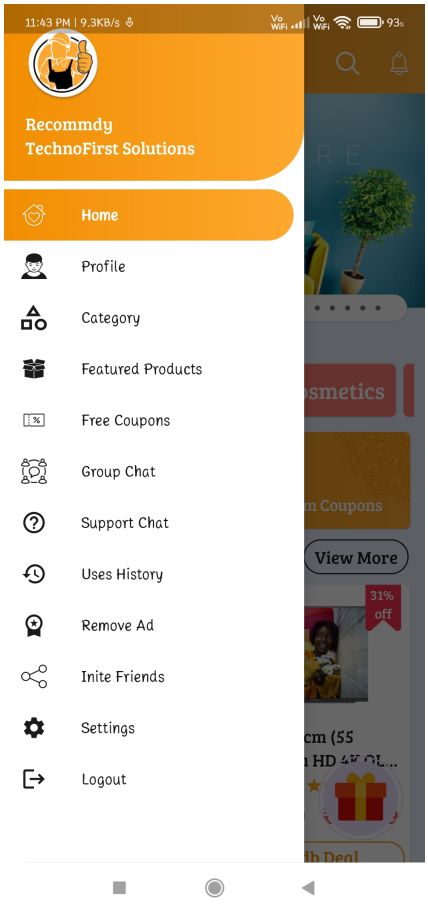


Figure 8.9: Side menu

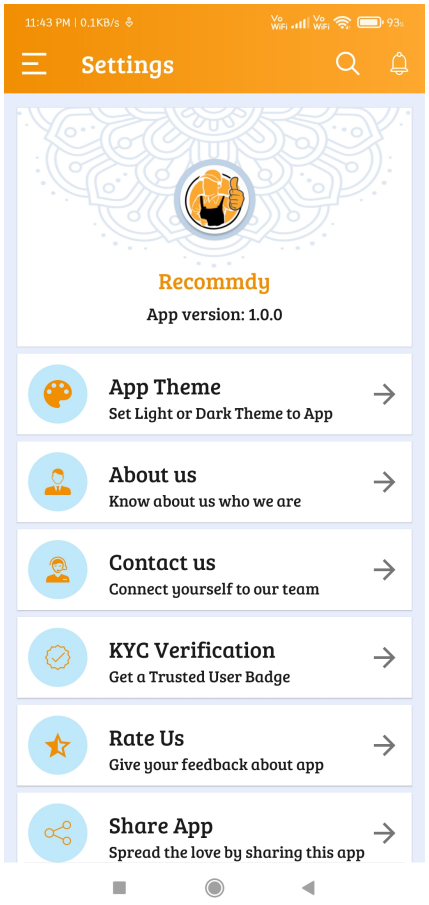


Figure 8.10: App Settings

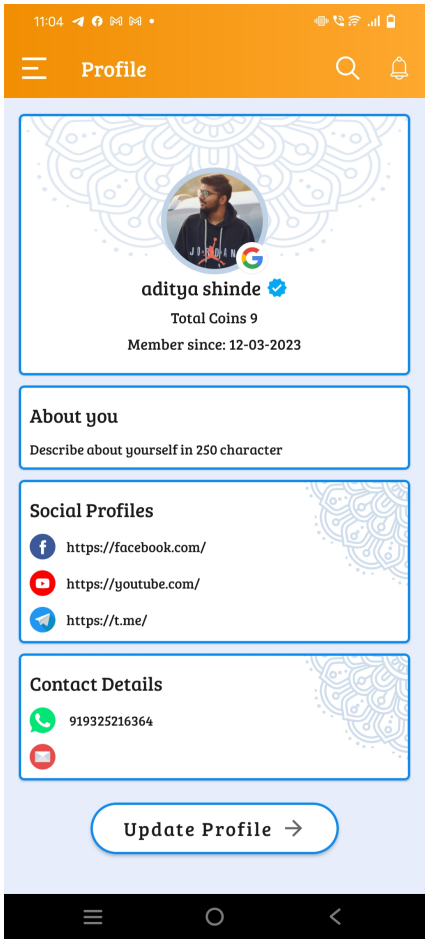


Figure 8.11: User Profile

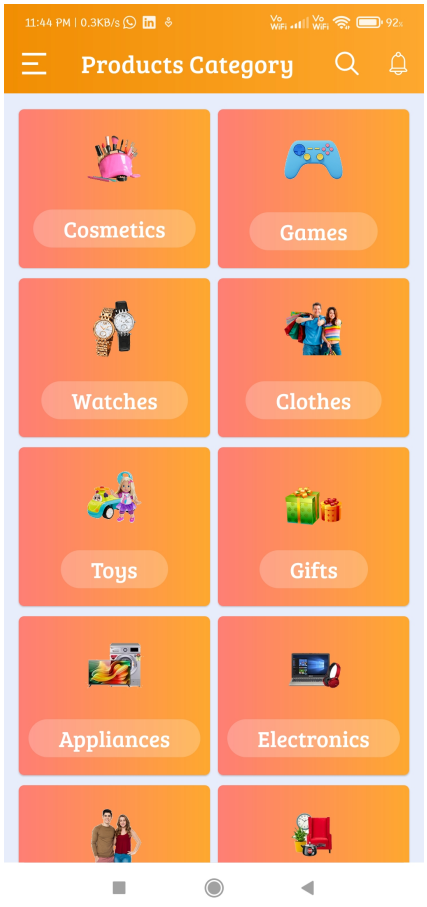


Figure 8.12: Products Category

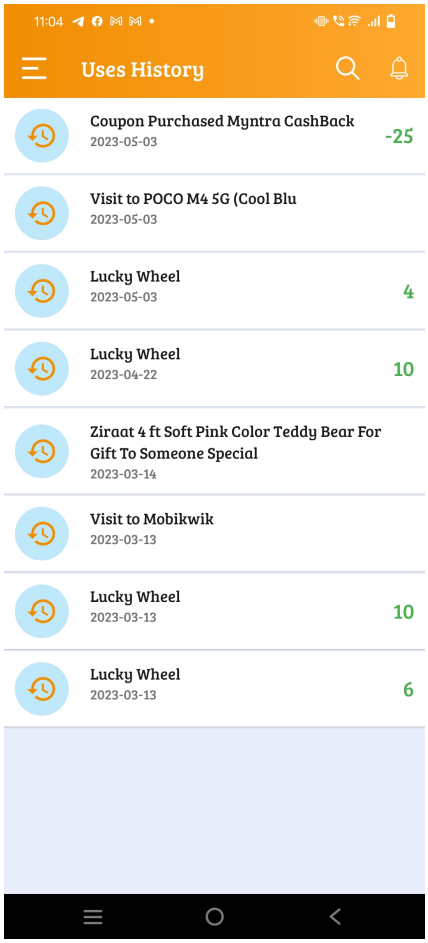


Figure 8.13: Uses History

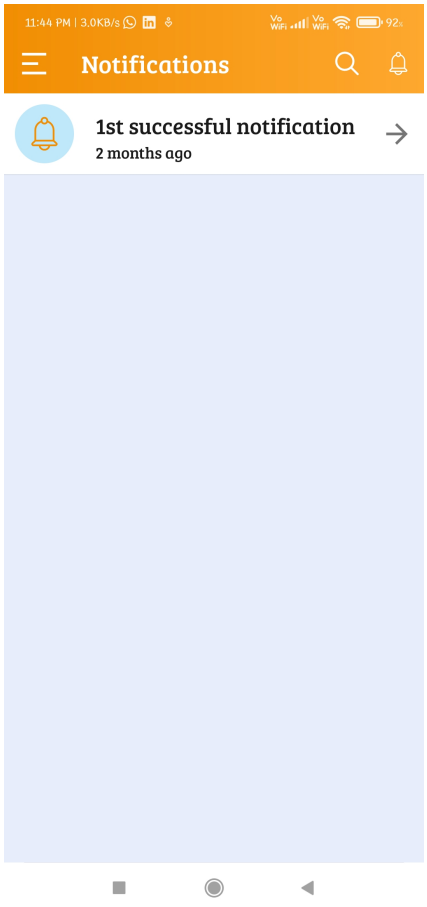


Figure 8.14: In-app push notification

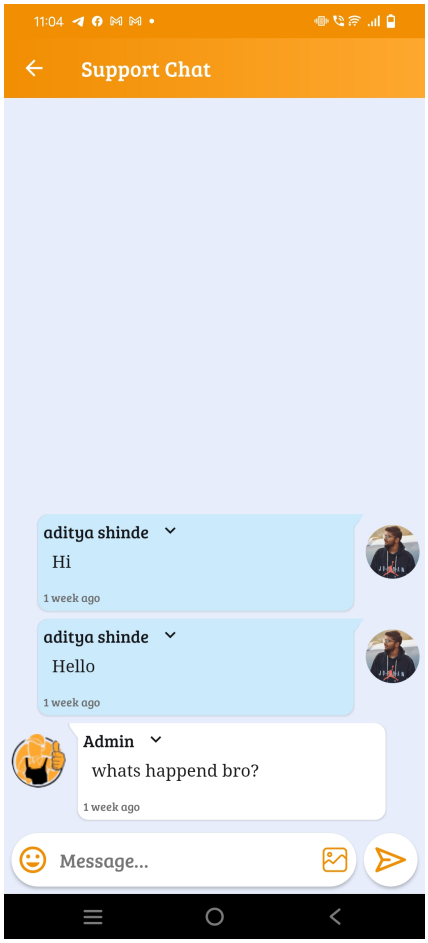


Figure 8.15: Support Chat

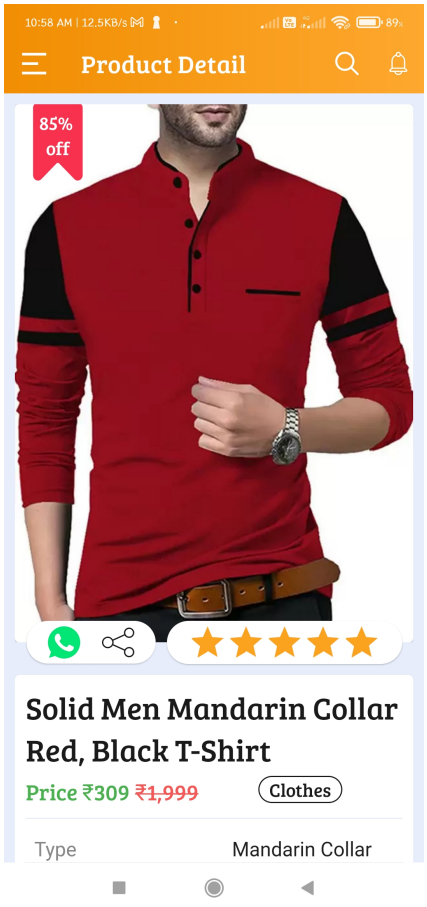


Figure 8.16: Products Description

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

Our project aimed to develop an advertisement recommendation system with a user-friendly interface for advertisers and affiliate marketers. The project has been successful in achieving its objectives and has provided a reliable, efficient, and effective solution for advertisers and affiliate marketers to promote their products and services. In conclusion, Recommdy emerges as a promising application with its advanced recommendation system and custom link generation feature. It provides advertisers and affiliate marketers with the means to connect with their target audience more effectively, enabling personalized advertising campaigns and increased conversion rates. Additionally, users benefit from tailored recommendations that enhance their shopping experience. With its potential applications across industries, including e-learning and entertainment, Recommdy has the capacity to revolutionize the way businesses engage with their customers and deliver relevant content. By harnessing the power of data-driven recommendations, Recommdy empowers businesses to make informed decisions, drive customer satisfaction, and achieve greater success in the dynamic digital landscape.

9.2 Future Scope

The developed application has a strong foundation, and there are a few areas where it can be improved upon in future work. One of the areas is enhancing the recommendation system by integrating machine learning algorithms that can improve the accuracy of the recommendations provided to users. This can be achieved by analyzing user behavior and generating personalized recommendations based on their interests and preferences.

Another potential area for future work is implementing additional payment gateways to the system to provide more payment options to users. This can help to increase the user base of the application by catering to a broader audience. Additionally, the application can benefit from incorporating more social media platforms into the affiliate marketing program to allow affiliates to share custom links on other platforms besides WhatsApp.

Moreover, incorporating a feature to allow users to provide feedback and ratings for advertisements can help to improve the quality of advertisements shown on the platform. This can help advertisers to improve their marketing strategies and generate better results.

Finally, improving the user interface to make it more user-friendly and attractive can enhance the user experience and increase the app's usability. Incorporating additional features such as push notifications for new advertisements or exclusive deals can also help to keep users engaged and returning to the application. Overall, there are many areas for future work that can further improve the application's functionality, usability, and user experience.

Appendix A

Computational Complexity

The computational complexity of collaborative filtering algorithms can vary depending on the specific approach and implementation used. Here's a general overview of the computational complexity of collaborative filtering:

1. **User-Based Collaborative Filtering:**
   - Computational complexity for predicting ratings for a user: $O(n^2)$, where n is the number of users.
   - Generating recommendations for a user: $O(n^2 * m)$, where n is the number of users and m is the number of items.

2. **Item-Based Collaborative Filtering:**
   - Computational complexity for predicting ratings for a user: O(n * m), where n is the number of users and m is the number of items.
   - Generating recommendations for a user: O(n * m), where n is the number of users and m is the number of items.

Appendix B

Certificates of Participation/Prize winning

Maratha Vidya Prasarak Samaj's
Karmaveer Adv. Baburao Ganpatrao Thakare
College of Engineering, Nashik
KBTCOE
NAAC
Permanently Affiliated to Savitribai Phule Pune University and Approved by AICTE, New Delhi

Certificate

This is to certify that
Mr. Aditya Shinde
Ms. Ankita Kulkarni
Ms. Tanaya Gavandel
Ms. Krutika Rawal
have actively participated and presented their innovative idea entitled
"Recommendation Based Affiliate Marketing Mobile Application" in
"Idea Spark 2022" - An Institute Level Ideation Competition
Organized by ED & Startup Cell, MVPS's KBTCOE, Nashik on 5th November 2022 in association with
SIVANANDA ELECTRONICS, Devlali, Nashik

Dr. A. A. Kapse
Coordinator, ED & Startup Cell

Dr. S. R. Devane
Principal

Society of Computer Technology and Research's
PUNE INSTITUTE OF COMPUTER TECHNOLOGY

Impetus and Concepts '23

CERTIFICATE
OF PARTICIPATION

This certificate is awarded to Ankita Kulkarni
of MVP KBTCOE has participated in
Concepts at Impetus and Concepts 2023, an
International Level Technical Symposium

Dr. G. P. Potdar
InC 2023
Convenor

Dr. S. T. Gandhe
Principal
PICT

Dr. P. T. Kulkarni
Director
PICT

Title Sponsors: UPTIQ, eQ Technologic
Co-Sponsor and Concepts Sponsor: BAJAJ FINSERV
Co-Sponsors: iMocha, Sarvatra Technologies, AIRAVANA
Impetus Sponsors: AGCO
Pradnya Sponsors: VERITAS
Other Sponsors: zbyte, BNY MELLON, Foldhealth, AlgoAnalytics, DIACTO
In Association with: PICT IEEE Student Branch, CSI

Society of Computer Technology and Research's
PUNE INSTITUTE OF COMPUTER TECHNOLOGY

Impetus and Concepts '23

CERTIFICATE
OF PARTICIPATION

This certificate is awarded to Aditya Shinde
of MVPK BTCOE has participated in
Concepts at Impetus and Concepts 2023, an
International Level Technical Symposium

Dr. G. P. Potdar
InC 2023
Convenor

Dr. S. T. Gandhe
Principal
PICT

Dr. P. T. Kulkarni
Director
PICT

Title Sponsors: UPTIQ, eQ Technologic
Co-Sponsor and Concepts Sponsor: BAJAJ FINSERV
Co-Sponsors: iMocha, Sarvatra Technologies, AIRAVANA
Impetus Sponsors: AGCO
Pradnya Sponsors: VERITAS
Other Sponsors: zbyte, BNY MELLON, Foldhealth, AlgoAnalytics, DIACTO
In Association with: PICT IEEE Student Branch, CSI


Society of Computer Technology and Research's
PUNE INSTITUTE OF COMPUTER TECHNOLOGY
Impetus and Concepts '23
CERTIFICATE
OF PARTICIPATION
This certificate is awarded to Krutika Rawal
of MVP's KBTCoE has participated in
Concepts at Impetus and Concepts 2023, an
International Level Technical Symposium
Dr. G. P. Potdar
InC 2023
Convenor
Dr. S. T. Gandhe
Principal
PICT
Dr. P. T. Kulkarni
Director
PICT
Title Sponsors
UPTIQ
eQ Technologic
Co-Sponsor and Concepts Sponsor
BAJAJ FINSERV
Co-Sponsors
iMocha
Sarvatra TECHNOLOGIES
AIRAVANA
Impetus Sponsors
AGCO
Pradnya Sponsors
VERITAS
Other Sponsors
zbyte
BNY MELLON
Foldhealth
DIACTO
In Association with
PICT IEEE Student Branch
CSI


Society of Computer Technology and Research's
PUNE INSTITUTE OF COMPUTER TECHNOLOGY
Impetus and Concepts '23
CERTIFICATE
OF PARTICIPATION
This certificate is awarded to Tanaya Gavande
of MVP's KBTCoE has participated in
Concepts at Impetus and Concepts 2023, an
International Level Technical Symposium
Dr. G. P. Potdar
InC 2023
Convenor
Dr. S. T. Gandhe
Principal
PICT
Dr. P. T. Kulkarni
Director
PICT
Title Sponsors
UPTIQ
eQ Technologic
Co-Sponsor and Concepts Sponsor
BAJAJ FINSERV
Co-Sponsors
iMocha
Sarvatra TECHNOLOGIES
AIRAVANA
Impetus Sponsors
AGCO
Pradnya Sponsors
VERITAS
Other Sponsors
zbyte
BNY MELLON
Foldhealth
DIACTO
In Association with
PICT IEEE Student Branch
CSI

MARATHA VIDYA PRASARAK SAMAJ'S
KARMAVEER ADV. BABURAO GANPATRAO THAKARE COLLEGE OF ENGINEERING

KBTCOE

NAAC A++

DEPARTMENT OF COMPUTER ENGINEERING

ACCREDITED BY NBA NATIONAL BOARD OF ACCREDITATION
- MECHANICAL ENGINEERING
- COMPUTER ENGINEERING
- INSTRUMENTATION AND CONTROL ENGINEERING
- CIVIL ENGINEERING
- INFORMATION TECHNOLOGY (IT)

ACES

CERTIFICATE
OF PARTICIPATION

Ankita Kulkarni

has actively participated in the MVPS's - KSHITIJ 2023 "Project Competition" organized by Department of Computer Engineering (ACES), KBTCOE, Nashik on 8th May, 2023.

Ms.P.P.Boraste	Dr. D.V. Medhane	Dr.V.M.Birari	Dr. S.R. Devane
Event Coordinator	Head of Department	Vice-Principal	Principal

MARATHA VIDYA PRASARAK SAMAJ'S
KARMAVEER ADV. BABURAO GANPATRAO THAKARE COLLEGE OF ENGINEERING

KBTCOE

NAAC A++

DEPARTMENT OF COMPUTER ENGINEERING

ACCREDITED BY NBA NATIONAL BOARD OF ACCREDITATION
- MECHANICAL ENGINEERING
- COMPUTER ENGINEERING
- INSTRUMENTATION AND CONTROL ENGINEERING
- CIVIL ENGINEERING
- INFORMATION TECHNOLOGY (IT)

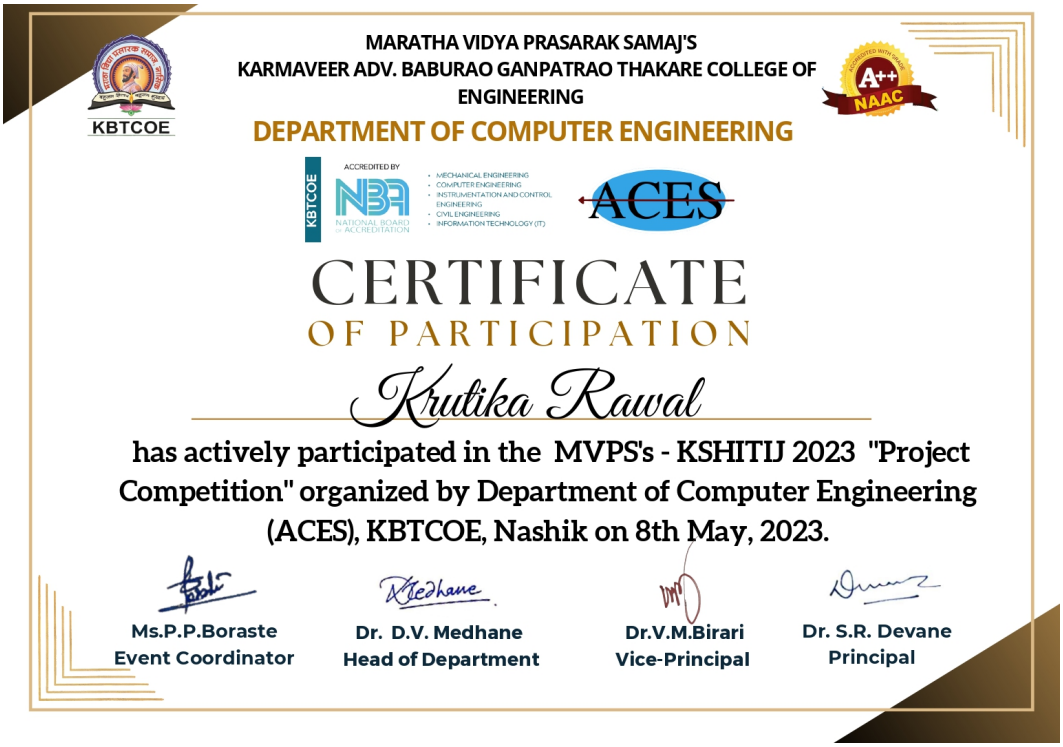
ACES

CERTIFICATE
OF PARTICIPATION

Aditya Shinde

has actively participated in the MVPS's - KSHITIJ 2023 "Project Competition" organized by Department of Computer Engineering (ACES), KBTCOE, Nashik on 8th May, 2023.

Ms.P.P.Boraste	Dr. D.V. Medhane	Dr.V.M.Birari	Dr. S.R. Devane
Event Coordinator	Head of Department	Vice-Principal	Principal



MARATHA VIDYA PRASARAK SAMAJ'S
KARMAVEER ADV. BABURAO GANPATRAO THAKARE COLLEGE OF ENGINEERING

KBTCOE

NAAC A++

DEPARTMENT OF COMPUTER ENGINEERING

ACCREDITED BY NBA NATIONAL BOARD OF ACCREDITATION
- MECHANICAL ENGINEERING
- COMPUTER ENGINEERING
- INSTRUMENTATION AND CONTROL ENGINEERING
- CIVIL ENGINEERING
- INFORMATION TECHNOLOGY (IT)

ACES

CERTIFICATE
OF PARTICIPATION

Krutika Rawal

has actively participated in the MVPS's - KSHITIJ 2023 "Project Competition" organized by Department of Computer Engineering (ACES), KBTCOE, Nashik on 8th May, 2023.

Ms.P.P.Boraste	Dr. D.V. Medhane	Dr.V.M.Birari	Dr. S.R. Devane
Event Coordinator	Head of Department	Vice-Principal	Principal

MARATHA VIDYA PRASARAK SAMAJ'S
KARMAVEER ADV. BABURAO GANPATRAO THAKARE COLLEGE OF ENGINEERING

KBTCOE

NAAC A++

DEPARTMENT OF COMPUTER ENGINEERING

ACCREDITED BY NBA NATIONAL BOARD OF ACCREDITATION
- MECHANICAL ENGINEERING
- COMPUTER ENGINEERING
- INSTRUMENTATION AND CONTROL ENGINEERING
- CIVIL ENGINEERING
- INFORMATION TECHNOLOGY (IT)

ACES

CERTIFICATE
OF PARTICIPATION

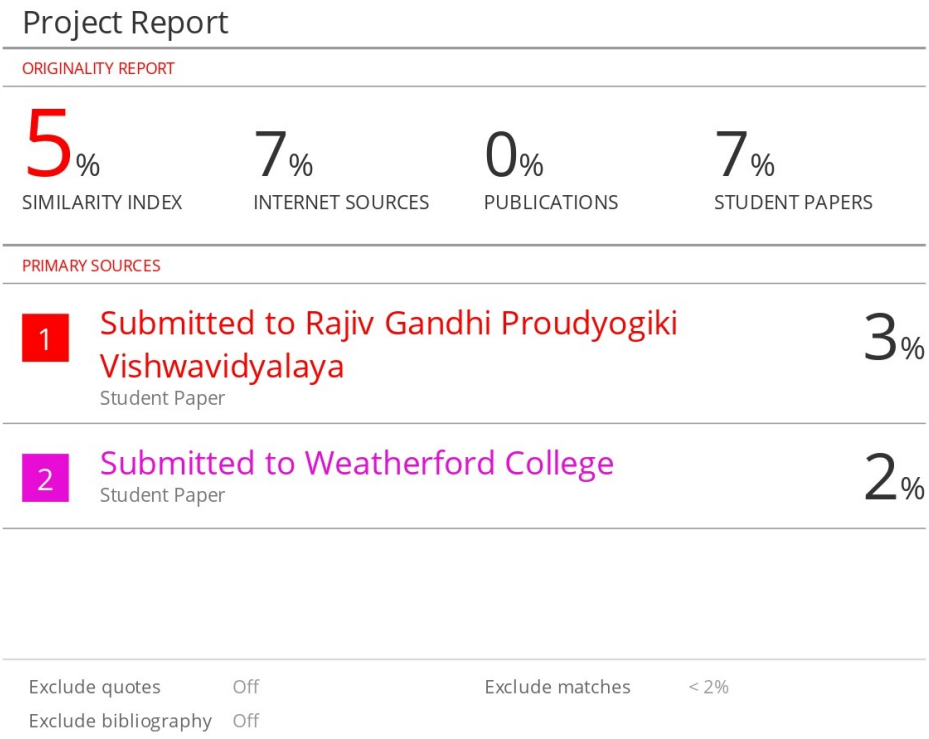
Tanaya Gavande

has actively participated in the MVPS's - KSHITIJ 2023 "Project Competition" organized by Department of Computer Engineering (ACES), KBTCOE, Nashik on 8th May, 2023.

Ms.P.P.Boraste	Dr. D.V. Medhane	Dr.V.M.Birari	Dr. S.R. Devane
Event Coordinator	Head of Department	Vice-Principal	Principal

Appendix C

Plagiarism Check Report



Project Report

ORIGINALITY REPORT

5%	7%	0%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Rajiv Gandhi Proudyogiki Vishwavidyalaya Student Paper	3%
2	Submitted to Weatherford College Student Paper	2%

Exclude quotes	Off	Exclude matches	< 2%
Exclude bibliography	Off		

Bibliography

[1] Nhi Luu "A CRITICAL LOOK AT AFFILIATE MARKETING IN VIETNAM", 2022

[2] Shun Li et. al. "A New QoS-Aware Web Service Recommendation System Based on Contextual Feature Recognition at Server-Side", 2017

[3] Pegah Malekpour Alamdari et. al. "A systematic study on the recommender systems in the e-commerce", 2017

[4] Senay Kocakoyun "Developing of Android Mobile Application using Java and Eclipse: An Application", 2017

[5] Hwansoo Kang et. al. "Case Study on Efficient Android Programming Education using Multi Android Development Tool", 2015

[6] Jianying Mai et. al. "A Neural Networks-based Clustering Collaborative Filtering Algorithm in E-commerce Recommendation System", 2009

[7] https://docs.nomagic.com/display/MD2022xR1/Object+diagram

[8] https://www.geeksforgeeks.org/software-engineering-cocomo-model/amp/