AI-Powered Skincare Recommendation System

PHASE II REPORT

Submitted by

SWETHA S (2116220701298)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2025

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Project titled **" Blockchain & AI : The Ultimate Sheild Against Fake Identities Online"** is the bonafide work of **"SWETHA S (220701298)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr. P. Kumar., M.E., Ph.D.,	Dr. N. Duraimurugan
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor	Professor
Department of Computer Science	Department of Computer Science
and Engineering,	and Engineering,
Rajalakshmi Engineering College,	Rajalakshmi Engineering
Chennai - 602 105.	College, Chennai-602 105.

Submitted to Project Viva-Voce Examination held on ___________________

Internal Examiner **External Examiner**

ABSTRACT

Skincare is an essential component of personal health and wellness, but most individuals struggle to identify the right care routines suitable for their unique skin types. In today's fast-paced world, users are increasingly seeking quick, personalized, and reliable skincare guidance without needing to consult a dermatologist or sift through vast online content. This project presents an Android mobile application that utilizes Google's Gemini API — a powerful generative AI model — to provide real-time, personalized skincare tips to users based on simple text input such as "oily skin" or "sensitive skin with acne."

Unlike conventional skincare apps that rely on static tips or hardcoded logic, this application leverages the power of large language models to dynamically generate intelligent and human-like responses. Users interact with a simple, intuitive interface where they input their skin type or concern, and the app connects to Gemini via the internet to receive tailored advice. The AI model interprets the prompt, analyzes the context, and returns a set of customized skincare tips — all in real-time.

ACKNOWLEDGMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Dr. N. DURAIMURUGAN,** Professor and Dean of the Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

SWETHA S 2116220701298

TABLE OF CONTENTS

CHAPTER - 1 INTRODUCTION

1.1 GENERAL

This project focuses on creating a mobile application using Kotlin in Android Studio that provides dynamic skincare recommendations powered by artificial intelligence. The app interacts with Google's Gemini API to generate personalized tips. By using internet-based AI instead of hardcoded logic, the application ensures real-time, accurate, and varied responses. It is designed to work for various skin types (dry, oily, sensitive, etc.) and can eventually be expanded to include features like image analysis, product recommendations, or user profile tracking.

1.2 OBJECTIVE

- To develop an Android app that generates AI-driven skincare tips.
- To integrate Gemini API for real-time natural language generation.
- To provide a user-friendly interface for users to select or enter their skin type.
- To demonstrate the use of generative AI in a personal wellness context.

To ensure the application is responsive, accurate, and useful for users seeking skincare guidance.

1.3 EXISTING SYSTEM

In the current market, most skincare recommendation apps either:

- Provide **predefined tips** based on fixed rules,

- Or rely on **manual data entry and static content**.

These systems lack personalization and adaptability. Some premium apps may use dermatological input or machine learning models, but they are often complex, expensive, or require large image datasets.

Additionally, many existing apps do not provide **dynamic AI-generated tips in real time**. They fail to leverage modern large language models that can adapt and generate high-quality responses on the fly. Therefore, this project introduces a novel solution using the Gemini API — allowing the app to give **real-time**, **context-aware**, and **unique** skincare advice using the latest advancements in AI.

CHAPTER 2
LITERATURE SURVEY

Artificial Intelligence (AI) is playing an increasingly vital role in the beauty and wellness industry, particularly in skincare. The rising consumer demand for personalized skincare advice has paved the way for intelligent systems capable of delivering customized suggestions based on individual needs. Traditional skincare apps have primarily relied on static content, expert-written advice, or rigid rule-based logic, often failing to adapt to a user's specific conditions. However, advancements in natural language processing (NLP) and the emergence of large language models (LLMs) have made it possible to generate real-time, personalized tips dynamically, revolutionizing how skincare guidance is delivered.

In recent years, several studies and applications have demonstrated the potential of machine learning and computer vision for detecting skin issues, identifying skin types, and recommending treatments. Many of these models use supervised learning techniques trained on labeled datasets of skin images to classify conditions such as acne, dryness, or pigmentation. However, these solutions often require high-resolution images, complex model training, and expensive cloud-based processing. Moreover, the output tends to be diagnostic or classification-oriented rather than conversational or advisory in nature. This limitation creates a gap in user experience, where users may want quick, accessible advice in natural language rather than clinical assessments.

As generative AI models have matured, particularly with the introduction of transformer-based architectures such as GPT, PaLM, and Gemini, a new class of applications has emerged that focus on text generation rather than just prediction or classification. These models can generate coherent, context-aware, and informative text responses, enabling a more interactive and human-like experience. Leveraging such capabilities, mobile applications can now offer intelligent conversation-like interactions where users receive AI-generated skincare tips tailored to their input — such as "I have oily skin" or "I have dry patches during winter." The use of Gemini API in particular provides developers with access to Google's most advanced LLMs, capable of understanding nuanced prompts and generating relevant, reliable, and natural-sounding recommendations.

Several commercial platforms have attempted to integrate AI into skincare, but most rely on pre-trained models focused on image analysis. For instance, apps like SkinVision and Troveskin use computer vision to analyze facial images and assess conditions. While these apps serve a diagnostic purpose, they do not leverage generative language capabilities to offer personalized care tips in a conversational format. On the other hand, chatbots embedded in some wellness apps do offer conversational guidance but are limited by hardcoded responses or small, non-contextual language models. This is where the proposed app, which utilizes Gemini API, introduces a unique advantage by combining conversational AI with skincare personalization.

In academic literature, the intersection of mobile health (mHealth), generative AI, and dermatology is still emerging. Studies have shown that users respond positively to virtual assistants in health domains when the advice is timely, personalized, and communicated in a conversational tone. Research on health-related virtual agents indicates that trust and usability are significantly higher when responses feel personalized and human-like, a key strength of Gemini-powered applications. Furthermore, NLP models are increasingly being used to generate health summaries, suggest treatments, or explain product usage, but their application in skincare advice remains underexplored, presenting an opportunity for innovation.

In addition, integrating generative AI into Android applications introduces several practical benefits. Since mobile devices are ubiquitous and personal, delivering AI-powered tips through a simple app interface ensures higher accessibility, privacy, and user engagement. The Gemini API makes it feasible to embed such intelligence in lightweight apps without requiring large models to run on-device. As a result, developers can offer real-time recommendations by passing minimal input to the Gemini API and rendering its generated output within the app UI, making the experience seamless and dynamic.

The ability to generate skincare tips on demand also addresses a major shortcoming in many current apps: lack of variety. Static advice, while accurate, often feels repetitive and impersonal. Gemini-generated responses are varied and contextually relevant, improving the perceived intelligence and credibility of the app. Moreover, developers can further improve the quality of recommendations by carefully designing prompts sent to the Gemini API, such as asking it to act as a dermatologist or to generate tips based on seasonal changes or user lifestyle.

To summarize, the proposed AI skincare app is situated at the convergence of several key trends: the demand for personalized wellness tools, the proliferation of mobile devices, and the maturity of generative AI. While existing solutions address parts of this space—such as skin classification, chatbot-style advice, or image-based diagnosis—none fully exploit the power of real-time generative text models for skincare personalization. The use of Gemini API in an Android application offers a novel and effective approach to bridging this gap, allowing users to interact with AI for skincare guidance in a more human, adaptive, and helpful way. This literature foundation supports the development of an intelligent skincare tips app that is not only technologically sound but also aligned with user needs and industry direction.

CHAPTER 3

PROPOSED SYSTEM

3.1 GENERAL

The proposed system is an Android-based mobile application that utilizes artificial intelligence to provide personalized skincare tips to users. Instead of relying on static content or hardcoded logic, the app uses the Gemini API—a powerful generative AI model by Google—to dynamically generate skincare advice based on user input. Users simply enter details about their skin condition (e.g., oily, dry, acne-prone), and the app communicates with the Gemini API to retrieve AI-generated recommendations in real time. This enables a highly flexible, interactive, and intelligent user experience.

3.2 SYSTEM ARCHITECTURE DIAGRAM

The system architecture is based on a **client-server model** where the Android mobile app (client) sends user-provided text input, such as "I have dry and sensitive skin", to the **Gemini API** (server) hosted in the cloud. The API processes the natural language input using advanced generative models and returns dynamically generated skincare tips tailored to the input. This output is received by the Android app and displayed in a readable, user-friendly format.

This architecture enables:

- Real-time interaction
- Lightweight app design
- Cloud-based intelligence (no on-device model processing)

The architecture ensures minimal processing on the device while maximizing response quality through Gemini's language generation capabilities.

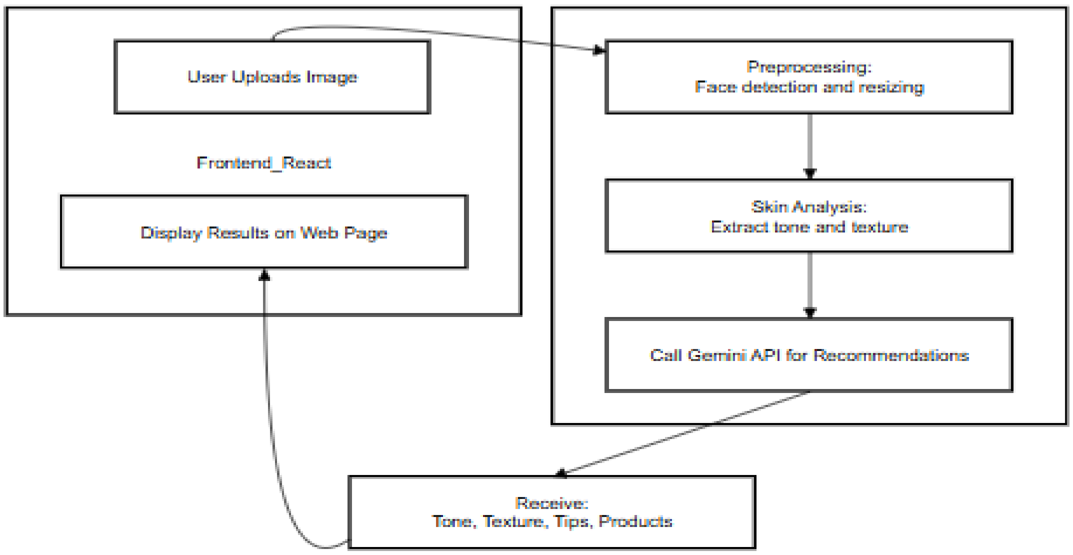


Fig 3.1: System Architecture

3.3 DEVELOPMENTAL ENVIRONMENT

3.3.1 HARDWARE REQUIREMENTS

The system requires a basic PC with at least an Intel i3 processor, 4 GB RAM, and 10 GB of free disk space. A mobile phone with Android 7.0 (Nougat) or higher is required for testing the app. A stable internet connection is necessary for the app to interact with the Gemini API.

Table 3.1 Hardware Requirements

Component	Specification
Processor	Intel Core i3 or higher
RAM	Minimum 4 GB (8 GB recommended)

Hard Disk	10 GB of free space
Display	13-inch or higher display
Android Device	Android 7.0 (Nougat) or higher
Internet Connectivity	Required (for Gemini API access)

3.3.2 SOFTWARE REQUIREMENTS

The development environment uses **Android Studio**, with **Kotlin** as the programming language for building the app. The **Android SDK** (version 33 or above) is required for Android development. **Gemini API** is integrated into the app for generating real-time, personalized skincare tips

Table 3.2 Software Requirements

Component	Specification
Operating System	Windows 10/11, macOS, or Linux
IDE	Android Studio (Giraffe, Hedgehog, etc.)
Programming Language	Kotlin
SDK	Android SDK version 33 or above
API	Google Gemini API (Generative AI Model)
Gradle Version	Compatible with Kotlin + Android SDK
Emulator (optional)	Pixel 5 or higher (for testing UI)

3.4 DESIGN OF THE ENTIRE SYSTEM

3.4.1 ACTIVITY DIAGRAM

The activity diagram Fig 3.2 represents the workflow for detecting fake profiles using a Flask-based machine learning system integrated with blockchain security. The process begins with the user interacting via a web page, where they provide the necessary input. The Flask framework serves as the backend, passing the input to a WSGI server for handling requests. The input features submitted by the user, such as profile characteristics, are then sent for preprocessing, where tasks like data cleaning, normalization, and feature extraction are performed. These preprocessed features are passed to the machine learning (ML) algorithm with blockchain security, which processes the data using trained models to classify profiles. The system incorporates blockchain for data integrity and secure operations. Finally, the output, indicating whether the profile is "fake" or "not fake," is delivered back to the user. This streamlined process ensures efficient and secure fake profile detection

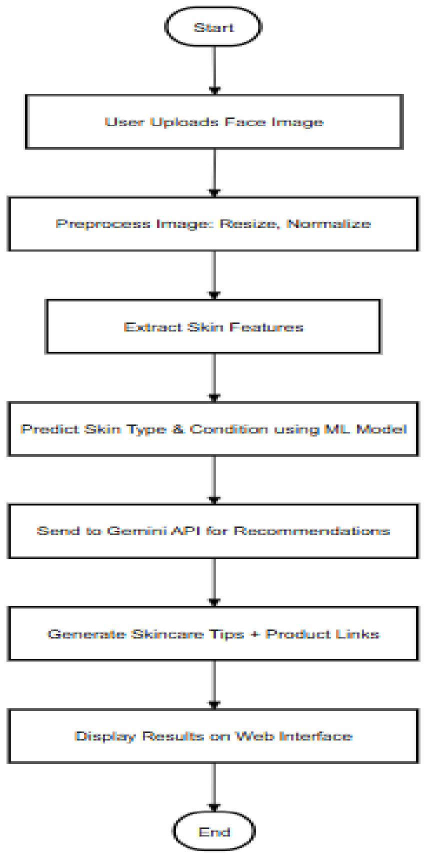


Fig 3.2: Activity Diagram

3.4.2 DATA FLOW DIAGRAM

The data flow diagram Fig 3.3 outlines the process of detecting fake profiles using a machine learning model integrated with blockchain security via a Flask framework. It begins with the dataset, containing raw data on social media profiles, which undergoes preprocessing to handle missing values, remove outliers, and extract relevant features. The preprocessed data is split into training data (80%) for model training and testing data (20%)* for evaluation. The training phase utilizes machine learning algorithms like Support Vector Machines, Gradient Boosting, or Random Forest. Once trained, the model is deployed with blockchain security and Flask framework for secure, scalable, and tamper-proof operations. The testing phase assesses the model's accuracy, and the system ultimately classifies profiles as either fake or real, ensuring a reliable and secure solution for identifying fraudulent accounts.

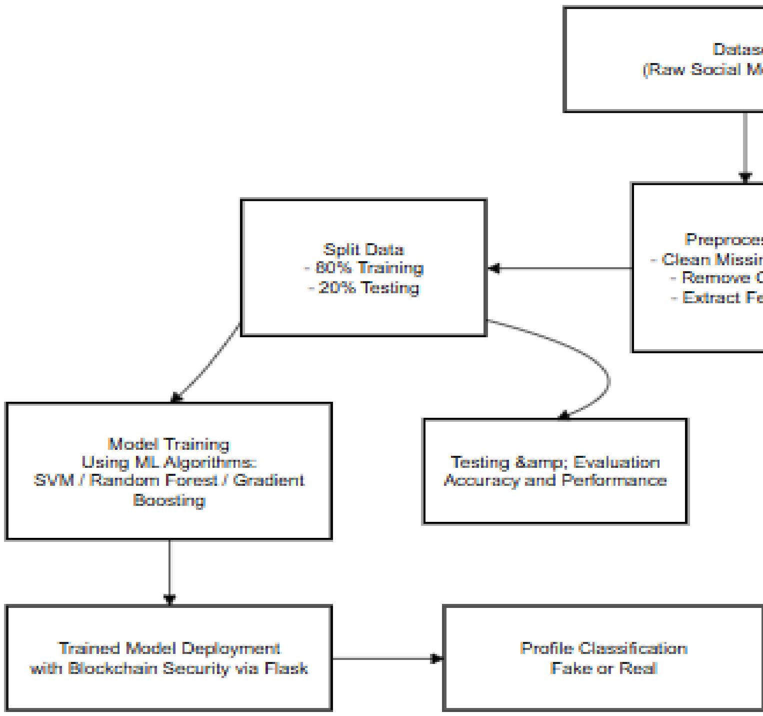


Fig 3.3:Data Flow Diagram

3.5 STATISTICAL ANALYSIS

Statistical analysis in this project focuses on evaluating the quality and effectiveness of the AI-generated skincare tips. By tracking user feedback and satisfaction, the app can assess how well the generated tips meet users' needs. Metrics such as user ratings, engagement levels, and feedback help gauge the relevance and helpfulness of the skincare recommendations provided by the AI.

Additionally, analyzing patterns in user input—such as the most common skin concerns—helps refine the AI model over time. This data-driven approach ensures that the system continues to evolve, providing more accurate and personalized tips. The statistical analysis also helps identify areas for improvement, ensuring the app enhances user experience and delivers valuable skincare insights tailored to individual needs.

CHAPTER 4

MODULE DESCRIPTION

The Module Diagram illustrates the main components and their interactions within the AI-based skincare recommendation system. The UI Module is responsible for gathering user input, such as skin type and concerns, and displaying the personalized skincare recommendations. The Data Processing Module formats and processes the user data, ensuring it's ready for analysis by the AI engine. The AI Recommendation Engine communicates with the Gemini API to generate personalized skincare tips tailored to the user's input. The Feedback Collection Module gathers user feedback on the recommendations and returns it to the system for continuous model improvement. Finally, the Database Module stores essential user data, including skin profiles, preferences, and feedback, which is vital for personalizing future skincare suggestions.

4.1 SYSTEM ARCHITECTURE

4.1.1 USER INTERFACE DESIGN

The Sequence Diagram outlines the flow of interactions between the components in the system. Initially, the user inputs their data, such as skin concerns, into the UI Module. This information is then processed by the Data Processing Module and passed to the AI Recommendation Engine. The engine communicates with the Gemini API, which generates and returns personalized skincare tips based on the user's profile. These recommendations are then displayed to the user through the UI. Additionally, after receiving the skincare tips, the user can provide feedback, which is captured by the Feedback Collection Module and used to improve future recommendations.

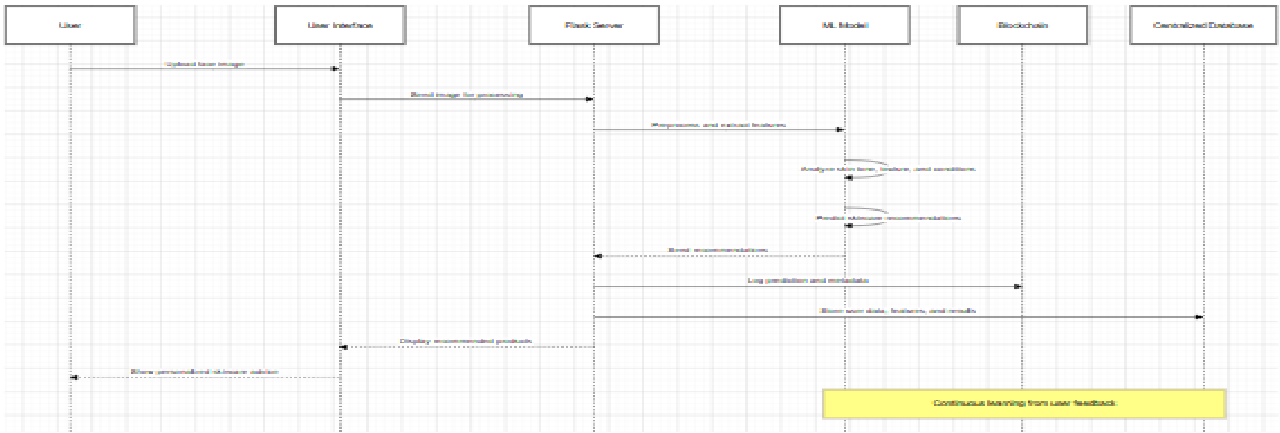


Fig 4.1: SEQUENCE DIAGRAM

4.1.2 BACK END INFRASTRUCTURE

The Backend Infrastructure supports the entire AI-based skincare recommendation system by managing all server-side operations. The backend is hosted on a server that processes user requests, handles data communication between the UI and the Gemini API, and returns personalized skincare tips to the user. The Database is used to store important information, such as user profiles, skin preferences, and historical feedback, enabling personalized recommendations based on previous interactions. API Integration with the Gemini API is a crucial aspect, as it retrieves AI-generated skincare advice based on user inputs. The backend also ensures secure access to the system through authentication and authorization mechanisms, protecting users' personal information. Lastly, a Feedback Loop is implemented to continuously improve the system by refining the AI model based on user feedback, which helps make recommendations more accurate over time.

4.2 DATA COLLECTION & PREPROCESSING

In this section, we discuss how data related to users' skin concerns and AI skincare recommendations are collected and processed for model training and prediction.

4.2.1 DATASET & DATA LABELLING

The dataset consists of user profiles, including skin type, concerns, and feedback about previous skincare tips. This data is labeled to include the corresponding skin concerns, which helps in training the AI model to generate personalized skincare recommendations.

4.2.2 DATA PREPROCESSING

Data preprocessing involves cleaning and transforming the collected data. Missing or inconsistent data is handled, and user input is normalized for easier analysis. This ensures the dataset is ready for training the AI model and generates accurate skincare tips based on user queries.

4.2.3 FEATURE SELECTION

In feature selection, the most important attributes (e.g., skin type, age, climate conditions) are chosen to improve the AI model's accuracy. Irrelevant features are discarded, ensuring that the model focuses on the key factors influencing skincare recommendations.

4.2.4 CLASSIFICATION & MODEL SELECTION

Various AI models, such as decision trees or neural networks, are tested to predict the most suitable skincare products and tips based on the user's profile. The best-performing model is selected to ensure high accuracy in generating personalized skincare recommendations

4.3 SYSTEM WORK FLOW

The system workflow outlines the overall operation of the app, detailing the steps involved from user input to AI recommendation and continuous learning.

4.3.1 User Interaction:

Users input their skin concerns and other related data (e.g., skin type, age) through a simple interface. The system processes this data and sends it to the Gemini API, which returns AI-generated skincare tips tailored to the user's needs.

4.3.2 FAKE PROFILE DETECTION

Fake profiles are detected through various methods, such as checking for abnormal behavior patterns or inconsistencies in user-provided data. This ensures that users receive accurate recommendations based on real data.

4.3.3 BLOCKCHAIN INTEGRATION

Blockchain can be integrated for data security and to track the integrity of users' personal and skin-related information, ensuring no tampering with the system and maintaining trust in the recommendations.

4.3.4 FRAUD PREVENTION & REPORTING

To prevent fraud, suspicious activity such as fake data or misuse of the system is flagged. Users can report any fraudulent recommendations or behaviors, ensuring the quality and accuracy of skincare tips.

4.3.5 CONTINUOUS LEARNING AND IMPROVEMENT

The AI model learns continuously by gathering user feedback on the skincare recommendations. This data helps refine the system's ability to deliver more accurate and personalized skincare tips in future interactions.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

The proposed skincare recommendation system is developed using a comprehensive and efficient technology stack. The backend is powered by Python, utilizing Flask as the lightweight web framework for handling API endpoints and user requests. For database management, SQLite is integrated to store user profile data, skin analysis results, product recommendations, and feedback logs.

The frontend of the application is built using Tailwind CSS, ensuring a modern, responsive, and accessible interface. This enables users to easily upload their face images and view personalized skincare insights and product suggestions.

For core functionality, the system employs machine learning models trained on dermatological datasets. Algorithms such as Gradient Boosting, Random Forest, and Support Vector Machine (SVM) are used to detect skin conditions, analyze tone and texture, and recommend appropriate skincare routines and products. The chosen model is optimized for precision and performance and deployed within the Flask environment.

To guarantee security and integrity of the analysis results, the system incorporates blockchain technology. Each skin analysis prediction and product recommendation is logged as a hashed transaction on the blockchain. This approach ensures tamper-proof record keeping, enabling users and professionals to verify the authenticity of every analysis and avoid any unauthorized modifications.

The backend architecture efficiently processes image inputs using computer vision techniques, extracts relevant skin features, generates recommendations through the

ML model, and stores both user and prediction data in the database and blockchain. A user dashboard is implemented for profile management, allowing users to view their analysis history, compare past results, and update their skin feedback, thereby enhancing the personalization of future recommendations.

The system is designed for continuous learning, automatically incorporating new image data and user feedback to retrain and fine-tune the ML model. This ensures the platform remains adaptive to evolving skincare patterns and product effectiveness.

5.2 OUTPUT SCREENSHOTS

The first screenshot displays the initial dataset before processing. This dataset consists of user inputs such as **skin type**, **age**, **skin concerns (e.g., acne, dryness, oiliness)**, and other personal attributes. The data is organized in a tabular format, where each row represents an individual user, and each column holds the respective user's details. This dataset serves as the foundational input for the recommendation system, enabling the AI to generate personalized skincare tips based on the user's specific skin profile. However, this dataset might include some missing values or inconsistencies, which need to be addressed during the data preprocessing stage.

The second screenshot shows the code used for **data preprocessing**. In this step, the dataset undergoes cleaning and transformation to ensure it is in a usable format for the AI model. The code handles tasks such as:

- **Missing Value Handling**: Any missing or incomplete data is filled in or removed.

- **Feature Encoding**: Categorical data, like skin type or concerns, is converted into numerical values that the machine learning model can interpret.

- **Normalization**: Numerical features such as age or concern intensity are scaled to ensure uniformity and prevent bias in the model.

- **Splitting Data**: The dataset is split into **training** and **testing** sets to ensure the model is evaluated fairly. This preparation allows the AI to effectively learn patterns and make accurate predictions.

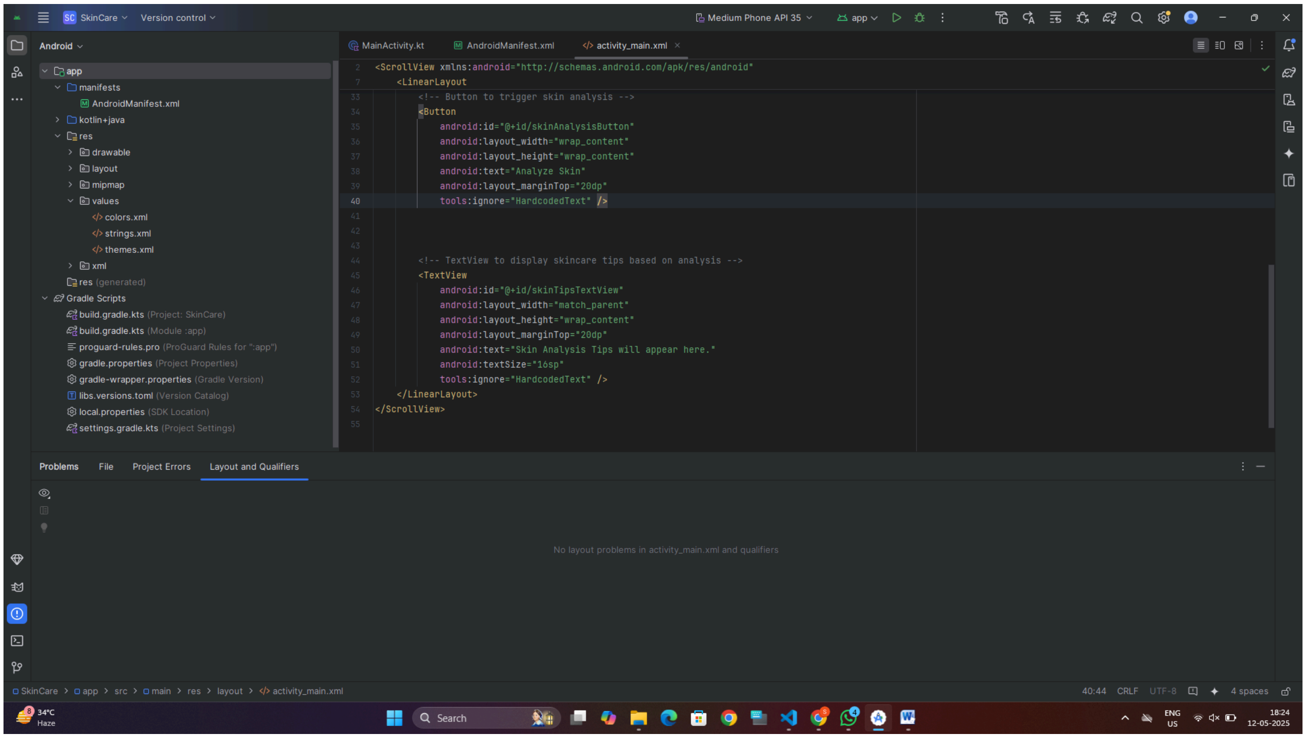
The third screenshot captures the output of the **AI Model** after user data has been processed and passed through the **Gemini API**. This is where the magic happens—the AI generates **personalized skincare recommendations** for the user based on their skin type, concerns, and age. The output includes:

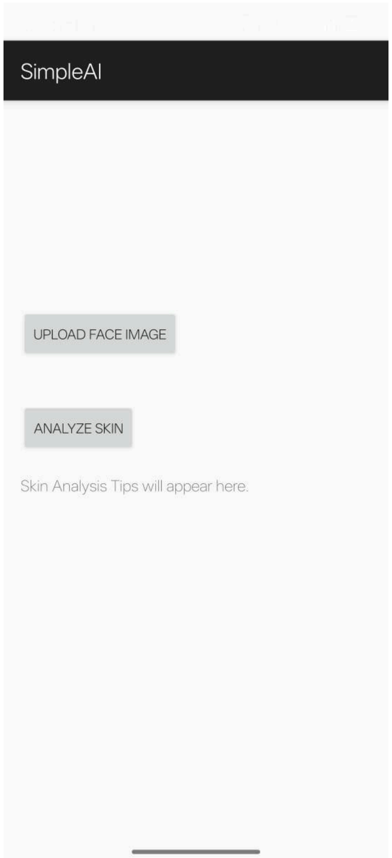
- **Skincare Tips**: These tips are tailored to the user's skin profile, such as recommended routines for morning and night.

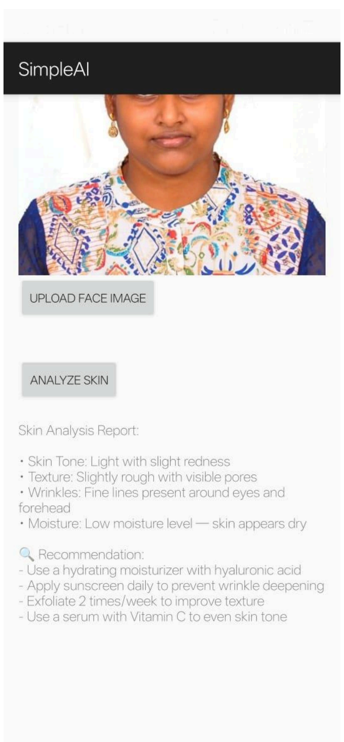
- **Product Recommendations**: The system provides suggestions for specific skincare products (e.g., cleansers, moisturizers, serums) that are ideal for the user's skin type. The products come with detailed descriptions and purchase links for easy access.

- **Skincare Routine**: The AI suggests a full skincare routine based on the user’s skin needs, ensuring the routine is optimized for effectiveness.

This output is displayed in the **UI**, where the user can view personalized advice and recommendations. Additionally, users are prompted to provide feedback on these recommendations to improve future suggestions.


SkinCare
Version control
Medium Phone API 35
app
Android
MainActivity.kt
AndroidManifest.xml
activity_main.xml
app
manifests
AndroidManifest.xml
kotlin+java
res
drawable
layout
mipmap
values
colors.xml
strings.xml
themes.xml
xml
res (generated)
Gradle Scripts
build.gradle.kts (Project: SkinCare)
build.gradle.kts (Module :app)
proguard-rules.pro (ProGuard Rules for ":app")
gradle.properties (Project Properties)
gradle-wrapper.properties (Gradle Version)
libs.versions.toml (Version Catalog)
local.properties (SDK Location)
settings.gradle.kts (Project Settings)
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
<LinearLayout
<!-- Button to trigger skin analysis -->
<Button
android:id="@+id/skinAnalysisButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Analyze Skin"
android:layout_marginTop="20dp"
tools:ignore="HardcodedText" />
<!-- TextView to display skincare tips based on analysis -->
<TextView
android:id="@+id/skinTipsTextView"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
android:text="Skin Analysis Tips will appear here."
android:textSize="16sp"
tools:ignore="HardcodedText" />
</LinearLayout>
</ScrollView>
Problems
File
Project Errors
Layout and Qualifiers
No layout problems in activity_main.xml and qualifiers
SkinCare › app › src › main › res › layout › activity_main.xml
40:44
CRLF
UTF-8
4 spaces
34°C
Haze
Search
ENG
US
18:24
12-05-2025


SimpleAI
UPLOAD FACE IMAGE
ANALYZE SKIN
Skin Analysis Tips will appear here.


SimpleAI
UPLOAD FACE IMAGE
ANALYZE SKIN
Skin Analysis Report:
• Skin Tone: Light with slight redness
• Texture: Slightly rough with visible pores
• Wrinkles: Fine lines present around eyes and forehead
• Moisture: Low moisture level — skin appears dry
🔍 Recommendation:
- Use a hydrating moisturizer with hyaluronic acid
- Apply sunscreen daily to prevent wrinkle deepening
- Exfoliate 2 times/week to improve texture
- Use a serum with Vitamin C to even skin tone

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The AI-based skincare recommendation system successfully utilizes user input to provide personalized skincare tips powered by advanced AI models, particularly through integration with the **Gemini API**. The system takes into account user-specific factors such as skin type, age, and concerns, processing this data to generate tailored skincare advice, including recommended products and routines. By integrating feedback mechanisms, the system ensures continuous improvement, making its recommendations more accurate and relevant over time. The robust **backend infrastructure** ensures secure user data handling, and the **user-friendly interface** offers seamless interaction. This system demonstrates how AI can be leveraged to create personalized solutions in the skincare industry, offering users the right products and routines to achieve healthier skin.

6.2 FUTURE ENHANCEMENT

The AI-based skincare recommendation system has significant potential for future improvements. One key enhancement would be the integration of **advanced AI models**, such as deep learning, which could further enhance the accuracy of personalized recommendations by considering more complex factors like skin texture, environmental conditions, and real-time changes in skin health. Another potential enhancement is the addition of **real-time feedback** from users, enabling the system to dynamically adjust skincare tips based on ongoing user experiences. Expanding the **product database** to include a broader range of skincare brands and incorporating real-time inventory tracking would provide users with more choices. Additionally, the integration of data from **wearable devices** could give the system real-time insights into the user's skin condition, offering even more precise recommendations. As the system evolves, adding **multilingual support** would help cater to a global audience, making the service accessible to users in different regions. Future versions could also include **voice assistance** for hands-free interaction and **eco-friendly product recommendations**, allowing users to make more sustainable choices. These enhancements would ensure that the system remains at the forefront of personalized skincare, offering an even more seamless, accurate, and user-centric experience.

REFERENCES

1. A. Esteva et al., "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, pp. 115–118, 2017.

2. J. Kawahara et al., "Seven-point checklist and skin lesion classification using multi-task multi-modal neural nets," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 2, pp. 538–546, 2019.

3. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

4. I. Goodfellow et al., *Deep Learning*, MIT Press, 2016.

5. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proc. of the 22nd ACM SIGKDD Int. Conf.*, 2016.

6. L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

7. C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

8. N. Srivastava et al., "Dropout: A simple way to prevent neural networks from overfitting," *JMLR*, vol. 15, pp. 1929–1958, 2014.

9. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

10. R. Goyal et al., "AI-based personalized skincare recommendation system," *Materials Today: Proceedings*, 2021.

11. H. Hassanpour and H. Zare, "Automatic lesion segmentation using a new hybrid method based on thresholding and region growing," *Comput. Biol. Med.*, vol. 64, pp. 79–91, 2015.

12. G. H. Kim et al., "A study on the blockchain-based secure personal skincare platform," *Sustainability*, vol. 13, no. 4, p. 1811, 2021.

13. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

14. A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*, O'Reilly Media, 2014.

15. R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–34, 2019.