JSS COLLEGE OF ARTS, COMMERCE & SCIENCE

An Autonomous College, Affiliated to University of Mysuru

Re-accredited by NACC with 'A+' Grade,

Ooty Road, Mysuru – 570025

A

Project Report on

"MEDICAL AI CHATBOT USING MULTIMODAL"

Submitted in the partial fulfilment of the requirements for the award of IV Semester

PG Department of Computer Applications

Submitted By

LIKITHA T

P01BE23S126013

Under the guidance of

Mrs. SHWETHA M.R.

Assistant Professor

PG Department of Computer Applications

JSS College of Arts, Commerce and Science, Mysuru – 570025

2024 - 2025

JSS COLLEGE OF ARTS, COMMERCE & SCIENCE

An Autonomous College, Affiliated to University of Mysuru
Re-accredited by NACC with 'A+' Grade,
Ooty Road, Mysuru – 570025

CERTIFICATE

This is to certify that LIKITHA T (P01BE23S126013), has successfully completed his/her project work entitled "MEDICAL AI CHATBOT USING MULTIMODAL" and submitted the report in partial fulfilment of the requirement of IV Semester, Master of Computer Applications, JSS College of Arts, Commerce and Science, Mysuru, during the academic year 2024 - 2025. It is certified that all corrections and suggestions indicated for the internal assessment have been incorporated in the report. This report has been approved as it satisfies the academic requirements in respect of project work prescribed for IV Semester MCA

Signature of the Guide
PG Department of
Computer Application

Signature of the HOD
PG Department of
Computer Application

VALUED
Name and Signature of the Examiners:

1. _____

2. _____

DECLARATION

I, LIKITHA T (P01BE23S126013), hereby declare that the project work entitled “MEDICAL AI CHATBOT USING MULTIMODAL” submitted to the JSS College of Arts, Commerce and Science, Ooty Road, Mysuru (Affiliated to University of Mysore) during the academic year 2023- 2024, is a record of an original work done by me under the guidance of Mrs. SHWETHA M.R, Assistant Professor, PG Department of Computer Applications.

Date: /08/2025 **LIKITHA T**

Place: Mysuru **(P01BE23S126013)**

ACKNOWLEDGEMENT

I would like to express my high regard to our college, ***JSS COLLEGE OF ARTS, COMMERCE AND SCIENCE, MYSURU,*** *for grooming me all these years. The support given by respected Principal* ***Dr. M Prabhu*** *is highly memorable.*

My sincere thanks to ***Mr. Ravi Kumar V G****, Head of the Department, Department of Master of Computer Applications, JSS College, Mysuru, for his valuable support during the project work.*

I wish to express my heartfelt, sincere gratitude to internal guide ***Mrs. SHWETHA M.R****,* ***Assistant Professor, PG Department of Computer Applications****, JSS Mysuru for her valuable suggestions and support.*

Finally, I would like to thank all the faculty members of the PG Department of Computer Applications for their support.

LIKITHA T

(P01BE23S126013)

ABSTRACT

This project presents a web-based Medical AI Chatbot that combines deep learning-powered skin disease classification with advanced natural language processing for interactive consultation. Using an EfficientNetB3 CNN with transfer learning, data balancing techniques, and a 70-15-15 train-validation-test split, the system classifies uploaded skin lesion images into ten common conditions, including eczema, melanoma, and psoriasis, with high accuracy. After diagnosis, a BERT-based NLP chatbot engages users in follow-up discussions, providing context-aware information on symptoms, treatments, and preventive measures tailored to the identified disease. This integrated approach offers a seamless, end-to-end solution that enhances accessibility, personalization, and scalability in dermatological care by merging computer vision with transformer-based NLP

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

CHAPTER 1

INTRODUCTION

Skin diseases are among the most prevalent health concerns worldwide, affecting millions of people across all age groups and regions. Conditions such as eczema, psoriasis, melanoma, fungal infections, and other dermatological disorders not only impact an individual's physical health but also significantly influence their emotional and psychological well-being. Early and accurate diagnosis of skin diseases is critical to ensuring proper treatment, preventing complications, and reducing healthcare burdens. However, timely access to dermatologists and diagnostic facilities remains a challenge, particularly in remote or resource-limited areas. This gap often leads to delayed diagnoses, improper treatments, and unnecessary anxiety for patients.

With rapid advancements in artificial intelligence (AI) and deep learning, there is a growing opportunity to leverage technology to bridge these gaps in dermatological care. AI-powered solutions, particularly those based on computer vision, have demonstrated remarkable performance in medical imaging tasks. Convolutional Neural Networks (CNNs), when combined with transfer learning techniques and pre-trained architectures such as Efficient Net, have emerged as powerful tools for analyzing complex visual patterns in medical images. These models can effectively classify various skin conditions, providing a scalable alternative to manual diagnosis.

While image-based AI diagnosis offers tremendous value, the healthcare experience does not end at disease identification. Patients often seek guidance regarding symptoms, treatment options, preventive measures, and lifestyle adjustments following a diagnosis. A static prediction model alone cannot address these dynamic, conversational needs. Therefore, integrating **Natural Language Processing (NLP)** capabilities—particularly through transformer-based models such as BERT (Bidirectional Encoder Representations from Transformers)—enables the creation of an interactive, intelligent medical assistant that can engage with users conversationally.

This project introduces a **Medical AI Chatbot for Skin Disease Diagnosis and Consultation**, a system designed to deliver both **automatic disease classification** and **context-aware conversational support**. The system operates as a web-based platform where users can upload an image of a suspected skin lesion. Using a CNN built on EfficientNetB3, the image is analyzed and classified into one of ten dermatological conditions. To enhance performance, the dataset undergoes balancing through oversampling and under sampling, combined with augmentation techniques to address class

imbalances, ensuring robustness and accuracy in prediction.

After classification, the system activates a **BERT-powered chatbot**. Unlike traditional FAQ-based bots, this chatbot is capable of understanding user queries in natural language, identifying the intent behind questions, and generating tailored responses. The chatbot leverages both the predicted disease label and a structured knowledge base (such as predefined responses and medical facts) to provide users with accurate and contextually relevant information. This dual functionality allows the system not only to predict but also to **educate and guide users**, helping them understand their condition, available treatments, and preventive care measures.

This integration of **computer vision and NLP** creates a seamless pipeline from **diagnosis to patient education**, reducing dependency on constant medical supervision for preliminary evaluations. The system does not replace professional medical advice but acts as a first-level diagnostic and informational tool, particularly valuable for areas with limited access to dermatology specialists. Additionally, it can serve as a triage assistant, directing patients toward professional care when severe or high-risk conditions (such as melanoma) are detected.

By combining state-of-the-art deep learning for visual recognition with advanced language models for interactive engagement, this project demonstrates the potential of AI to revolutionize digital healthcare. The solution offers scalability, accessibility, and interactivity, ensuring users not only receive accurate diagnostic insights but also the educational support they need to make informed decisions about their health.

1.1 Overview with Problem Identification

Dermatological care remains inaccessible or inefficient for many individuals due to a shortage of specialists, delayed consultations, and the high cost of medical diagnostics. Skin diseases, though common, can vary widely in presentation, making accurate diagnosis challenging for general practitioners and even patients themselves. Misdiagnosis or delayed treatment often leads to worsening symptoms, increased risk of complications, and psychological stress for patients.

Furthermore, while computer vision models for medical diagnosis have shown promise, most existing tools stop at predicting the disease. Patients often need more than just a name—they seek answers to questions about symptoms, causes, treatments, lifestyle modifications, and whether their condition requires urgent medical attention. The absence of interactive, AI-driven support systems means patients are left searching online, often encountering unreliable or confusing information.

Traditional chatbot systems for healthcare tend to rely on static rule-based methods or bag-of-words models, which lack contextual understanding and fail to deliver personalized responses. These systems cannot dynamically adapt their answers based on the context of a specific diagnosis or hold natural, human-like conversations. This gap between prediction and consultation creates a fragmented experience for patients seeking digital healthcare solutions.

The problem, therefore, is twofold:

1. **Accurate, automated detection of skin diseases** from user-submitted images, addressing issues of accessibility and early intervention.

2. **Interactive, context-aware guidance** post-diagnosis, enabling patients to get reliable, disease specific information through natural language interactions. Without an integrated system these capabilities, patients continue to face delays in diagnosis, lack of credible guidance, and of a reliance unreliable online resource, which can result in anxiety, improper self-treatment, or delayed professional intervention.

1.1 Objective

The main objective of our project is to design and develop a web-based Medical AI Chatbot that integrates skin disease classification and intelligent consultation through conversational support. The human skin reflects various dermatological conditions that can be analysed using modern deep learning techniques. The user uploads an image of the affected skin area through the web interface. The system then processes the image using computer vision techniques to classify the disease. Following this, a chatbot powered by natural language processing interacts with the user to provide relevant medical information and guidance based on the diagnosis.

1.2 Scope

This project focuses on creating a digital dermatology assistant that provides both diagnosis and consultation. Its scope includes:

- Image-Based Diagnosis: Classification of ten dermatological conditions using a deep learning model (EfficientNetB3) trained on a balanced dataset. The scope includes preprocessing, augmentation, and fine-tuning for high performance.
- Interactive Chatbot Consultation: A conversational agent built using BERT, capable of answering user questions about symptoms, treatments, prevention, and lifestyle adjustments for the detected condition.
- Deployment as a Web-Based Application: Designed for accessibility, the system is intended for deployment via Flask or Streamlit, enabling users to interact through an intuitive interface.
- Educational and Supportive Role: The system is not a replacement for certified medical professionals but serves as a triage and educational tool, guiding users toward appropriate medical care when necessary.

1.3 Existing System

Current digital healthcare solutions for dermatology typically fall into two categories:

1. Image-Based Diagnostic Tools: Most existing applications and research prototypes use CNNs or transfer learning models like ResNet, VGG, or EfficientNet to classify skin lesions. These tools primarily provide disease labels without offering interactive or conversational features. They are often standalone models without integration into a chatbot interface.
2. Chatbots for Healthcare: Many healthcare chatbots rely on rule-based systems or bag-of-words models, which are limited in understanding user context. These bots typically offer generic responses and are not capable of providing advice tailored to a user's specific condition. They also lack the ability to incorporate visual diagnostic input such as skin images.

While these systems address specific functions independently, they fail to combine automated skin disease diagnosis with personalized, context-aware consultation resulting in disjointed and incomplete user experiences.

1.4 Proposed System

The proposed system integrates **skin disease classification** and **BERT-powered conversational AI** into a single, seamless platform. Users upload a skin lesion image, which is processed by an **EfficientNetB3-based CNN** trained on a balanced dataset to predict one of ten dermatological conditions. The model uses **transfer learning** to achieve high accuracy with limited labeled data.

Once the disease is predicted, the system activates a **BERT-based chatbot**. The chatbot:

- Understands natural language queries.
- Uses the **predicted disease as context** to deliver accurate, tailored responses.
- Provides guidance on symptoms, treatment options, prevention, and recommendations for professional care.

The platform is built for deployment as a **web application (Flask or Streamlit)**, offering a **user-friendly interface** that allows seamless transitions from diagnosis to consultation. The system aims to **enhance healthcare accessibility**, empower patients with reliable information, and demonstrate the **potential of integrating computer vision with transformer-based NLP** in medical applications.

1.5 Advantages

The proposed system can be used for remote skin disease screening, especially in areas with limited access to dermatologists. It supports doctors by acting as a triage tool, helps users manage chronic skin conditions, and educates them about symptoms, treatments, and prevention. The system can also identify potentially serious conditions that need urgent attention and can be integrated into existing healthcare apps or hospital portals for broader use.

1.6 Applications

This system combines automatic skin disease diagnosis with intelligent chatbot consultation, offering a complete and user-friendly solution. It provides accurate, personalized responses using EfficientNetB3 and BERT models, even with limited data. The web-based interface ensures easy access, especially for users in remote locations. It reduces misdiagnosis, improves patient awareness, lowers healthcare costs, and empowers users to make informed decisions about their skin health.

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

The integration of artificial intelligence, deep learning, and natural language processing into healthcare has paved the way for innovative systems that assist in disease detection, classification, and patient support. Among medical fields, **dermatology** has been a particularly promising area for AI adoption due to the visual nature of diagnosis and the abundance of digital image data. This literature survey reviews key studies that form the foundation of this project, examining their methodologies, models, achievements, and gaps. The focus is on **skin disease image recognition using CNNs, transformers, segmentation-based approaches, and interactive diagnostic systems**.

1. Lessons from Agricultural Disease Classification – Aritra Das et al. (2025)

Although not focused on human dermatology, Aritra Das et al. [8] provided a state-of-the-art review of deep learning approaches for plant leaf disease detection, classification, and segmentation, offering transferable insights relevant to medical image analysis.

Notable observations include:

- Hybrid CNN-transformer models (e.g., combining CNN backbones with attention mechanisms) consistently improve classification performance by capturing both local textures and global patterns.
- Segmentation before classification enhances accuracy by focusing the model on relevant features, reducing background interference.
- Data augmentation remains a critical factor for generalization, especially in variable environments.

These findings underscore the potential of segmentation-enhanced pipelines and transformer-based architectures, which could be incorporated into future iterations of our project. Currently, we prioritize EfficientNetB3 for computational efficiency, while laying the groundwork for scaling to hybrid CNN-transformer systems.

2. Multimodal Diagnosis with Telegram Chatbot – Modigari et al. (2024)

Modigari, Harshini, and Anbarasi [6] proposed a multimodal AI-powered dermatology assistant that integrates image classification and chatbot-based textual analysis through the Telegram API. Their system combines CNN-based image classification with intent-driven conversational support to deliver

diagnostic predictions and allow users to ask text-based follow-up questions.

Key contributions include:

- Multimodal interaction: The system handles both image uploads and user text queries.
- Chatbot integration via Telegram API: Users interact with the system through Telegram, making it widely accessible without additional infrastructure.
- AI-driven classification: CNN models analyze uploaded dermatological images to identify disease categories.

However, the chatbot component was largely rule-based, relying on pre-written responses without true contextual NLP capabilities. Moreover, requiring users to operate through an external messaging platform reduced seamless integration and accessibility, especially for non-technical users.

Our project advances beyond Modigari et al. by embedding a BERT-powered chatbot directly into the web application, ensuring natural language understanding (NLU), contextual awareness, and richer engagement, eliminating the dependency on third-party chat services.

3. ConvNeXt-ST-AFF Hybrid Model – Hao et al. (2023)

Hao et al. [2] proposed **ConvNeXt-ST-AFF**, a **hybrid architecture combining ConvNeXt (a CNN variant) with Swin Transformer modules** for skin disease classification. The study aimed to overcome the limitations of CNNs in capturing **global contextual relationships** within images, which are crucial for identifying subtle lesion characteristics.

Their approach involved:

- Using **ConvNeXt** for hierarchical feature extraction (capturing local texture patterns).
- Integrating the **Swin Transformer** to capture global dependencies and attention-based features.
- Employing **feature fusion strategies** to merge outputs for final classification.

The model demonstrated superior accuracy compared to standalone CNNs, particularly in distinguishing visually similar dermatological conditions. However, the architecture is computationally intensive, requiring **high-end GPUs and large datasets**, which may not be practical for smaller-scale or real-time applications.

This work highlights the **future direction of combining CNNs and transformers** for improved performance, influencing the potential upgrade path for our project's classifier. Our system currently uses EfficientNetB3 for efficiency but can be scaled to incorporate transformer-based modules in future iterations.

4. Systematic Review of Machine Learning in Dermatology – Debelee (2023)

Debelee [10] conducted a systematic review of machine learning approaches for skin lesion classification and detection, covering traditional algorithms, CNNs, and hybrid architectures. The review identified several critical trends:

- CNNs dominate current research, but transformer-based and multimodal systems are gaining traction.
- Dataset imbalance and standardization issues remain major obstacles.
- There is a growing demand for interactive, patient-centric systems that bridge the gap between detection and guidance.

Debelee concluded that while progress in diagnostic accuracy has been substantial, patient-facing AI assistants integrating conversational interfaces are rare. This insight reinforces the novelty of our project, which combines CNN-based image classification with a BERT-driven chatbot, delivering both diagnosis and personalized, educational guidance.

5. Segmentation and Autoencoder-Based Classification – Dasari Anantha Reddy et al. (2023)

Dasari Anantha Reddy et al. [4] developed a **segmentation-driven approach for skin disease detection**, integrating **optimized region growing segmentation** with **autoencoder-based classification**. The segmentation step isolates lesions from background noise, while autoencoders reduce dimensionality and extract robust features before classification.

Advantages of this method include:

- Improved accuracy by focusing on lesion-specific regions.
- Reduced false positives caused by irrelevant background patterns.
- Better performance on cluttered or noisy datasets.

However, this pipeline is **computationally heavier** due to the multi-stage preprocessing and lacks the scalability of end-to-end CNN models. Additionally, the study did not incorporate **NLP or patient interaction** aspects, limiting its utility as a complete digital health assistant.

Our project draws inspiration from this method's **focus on preprocessing and lesion isolation**, which could be integrated into future versions of the EfficientNet-based pipeline for enhanced accuracy.

6. CNN-Based Human Skin Disease Detection – Ahmed et al. (2023)

Ahmed et al. [5] implemented a **CNN-based framework** for classifying various human skin diseases using a dataset of clinical images. The study demonstrated that CNNs, with sufficient data and

augmentation, can deliver **high accuracy** suitable for real-world deployment. The framework was trained end-to-end, avoiding manual feature engineering.

Strengths of their work include:

- Validation on multiple dermatological classes.
- Use of **data augmentation** to mitigate class imbalance.
- A relatively simple yet scalable architecture suitable for cloud deployment.

However, the system was **limited to classification only**, without any **post-diagnosis interaction or guidance**. Patients were not offered dynamic information about treatments or next steps, making it a diagnostic tool rather than a comprehensive assistant.

Comparative Insights and Research Gaps

Across these studies, a few **common patterns and gaps** emerge:

- CNNs (VGG, ResNet, EfficientNet) remain the **foundation of image-based dermatology solutions**, but hybrid models (CNN + Transformers) are gaining prominence for higher accuracy.
- **Data imbalance** and **limited labeled datasets** are recurring challenges, often addressed with augmentation and transfer learning.
- Most systems **stop at classification**, leaving patients to seek information elsewhere.
- There is a **lack of integrated systems** combining image classification with **interactive, context-aware chatbots** to deliver a holistic healthcare experience.

Our project addresses these gaps by:

1. Using **EfficientNetB3** for efficient, high-accuracy skin disease classification.
2. Balancing data using **oversampling, undersampling, and augmentation** to improve generalization.
3. Integrating a **BERT-powered chatbot** to provide **personalized, context-aware guidance**, transforming the system from a diagnostic tool into a **comprehensive dermatological assistant**.

7. <u>Mobile Net Deployment on Edge Devices – Gasa et al. (2020)</u>

Gasa, Owolawi, Mapayi, and Odeyemi [9] developed a MobileNet-based skin disease detection system deployed on a Raspberry Pi, integrated with a Telegram bot for delivering results. The primary goal was to demonstrate low-power, edge-device feasibility for AI-powered diagnostics.

Key outcomes include:

- Real-time inference on resource-constrained hardware using MobileNet.

- Integration with Telegram for remote user interaction.

While this approach showcased portability and cost-effectiveness, it had limitations:

- Reduced accuracy due to the lightweight MobileNet architecture compared to deeper networks like EfficientNet or ResNet.
- Reliance on Telegram introduced latency and limited conversational capabilities, as interactions were simple, rule-driven notifications.

Our project adopts a different strategy by deploying in a cloud-based environment using EfficientNetB3, which balances accuracy and efficiency, while leveraging a BERT-driven chatbot for richer, context-aware conversations directly through the web interface.

8. Survey on Skin Disease Classification Approaches – Dhruv et al. (2020)

Dhruv et al. [3] provided a comparative survey of multiple **skin disease classification techniques**, analyzing **traditional image processing methods, machine learning, and deep learning approaches**. The survey underscored the **limitations of traditional methods**, which rely heavily on manually extracted features (such as texture, shape, and color descriptors) and are highly sensitive to noise and variations in lighting.

Key conclusions include:

- Deep learning, particularly CNN-based models, **significantly outperforms** traditional classifiers like SVM and k-NN.
- **Data augmentation and transfer learning** are essential to overcome dataset limitations.
- Despite accuracy gains, CNNs suffer from **overfitting** on small datasets and **lack contextual awareness** for user interaction.

The survey solidifies the choice of CNNs as the baseline for our project's image classifier while also emphasizing the **need for robust data balancing and augmentation** (which we address through oversampling and synthetic augmentation). However, Dhruv et al. focused solely on classification and did not explore **integrating conversational AI**, which is a novel component of our system.

9. Deep Learning for Skin Disease Recognition – Li et al. (2020)

Li et al. [1] conducted a comprehensive review of **deep learning techniques applied to skin disease recognition**, analyzing over 100 studies and summarizing the strengths and limitations of CNN-based models. They highlighted how deep learning, especially architectures such as VGGNet, ResNet, and

Inception, revolutionized dermatological image analysis by eliminating the need for handcrafted features, enabling **end-to-end learning** from raw images.

Key findings from the review include:

- **CNNs outperform traditional machine learning models**, particularly in complex lesion classification tasks.
- Transfer learning with pre-trained networks (ImageNet) improves performance for small dermatology datasets.
- **Challenges remain**, such as **imbalanced datasets**, lack of standardization in image quality, and the absence of **interpretability** in black-box models, which limits clinical adoption.

This review establishes the importance of **transfer learning and robust preprocessing**, both of which are central to our project's EfficientNetB3-based image classifier. However, Li et al. primarily addressed classification performance and did not explore **post-diagnosis interaction** or **patient guidance systems**, a gap this project addresses.

10. CNN Architectures for Facial Skin Diseases – Wu et al. (2019)

Wu et al. [7] conducted an extensive study evaluating different CNN architectures for facial skin disease classification, comparing networks like AlexNet, ResNet, and DenseNet using clinical image datasets. Their research focused on identifying the most suitable CNN architecture for face-specific dermatological conditions.

Major findings include:

- ResNet and DenseNet outperformed AlexNet due to better feature propagation, depth, and gradient flow management.
- Depth and skip connections in modern networks provided superior performance in capturing fine-grained lesion features.

However, the study was constrained to facial images, limiting the generalizability of findings to broader dermatological conditions affecting the body. It also did not address patient interaction or real-time deployment considerations.

Our project extends this work by applying a modern, efficient CNN (EfficientNetB3) to a diverse dataset covering 10 different dermatological conditions, ensuring real-time performance while maintaining accuracy.

SOFTWARE REQUIREMENT SPECIFICATIONS

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATIONS

3.1 Functional Requirements

Functional requirements define the essential features and services that the system must perform. For the Medical AI Chatbot, these include the core interactions between the user, image classifier, and chatbot system:

1. **User Authentication**
   - Users can register and log in to access the platform securely.
   - Ensures privacy for personal health-related data.
2. **Image Upload and Processing**
   - Users can upload images of skin lesions.
   - The system accepts images in supported formats (e.g., .jpg, .png).
   - The image is resized, normalized, and prepared for classification.
3. **Skin Disease Classification**
   - EfficientNetB3-based CNN analyzes the uploaded image.
   - Predicts one of 10 skin disease classes with a confidence score.
4. **Chatbot Interaction**
   - BERT-based chatbot responds to natural language queries.
   - Chatbot retrieves condition-specific information such as symptoms, treatments, and precautions.
5. **Context Awareness**
   - Chatbot uses the predicted disease to tailor all responses.
   - Maintains session context for multi-turn conversations.
6. **Web Interface Interaction**
   - Provides a user-friendly frontend for image upload and chat.
   - Displays classification results and enables smooth conversation flow.

3.2 Non – Functional Requirements

Non-functional requirements define system quality, performance, and constraints:

- **Performance**:

  The CNN must classify images within 2–5 seconds. The chatbot should respond in under 1 second

- **Usability**:
  The platform must have an intuitive UI suitable for all users, including non-technical individuals.

- **Scalability**:

  System should support at least 100 concurrent users and allow integration of more disease classes in future.

- **Security**:

  User data must be securely handled using encryption and privacy

  regulations HIPAA/GDPR).

- **Reliability**:

  Maintain 99% uptime and handle unexpected user inputs gracefully.

- **Maintainability**:

  The architecture should be modular, allowing easy updates to the chatbot database and classification model.

3.3 Hardware Requirements

- **Camera**: Web camera (minimum 2MP for user test scenarios)
- **RAM**: Minimum 8 GB
- **Processor**: Intel i3 (4th Gen or above) / Ryzen 5 or better
- **Hard Disk**: Minimum 250 GB SSD/HDD

3.4 Software Requirements

- **IDE**: Visual Studio Code / Jupyter Notebook
- **Programming Languages**: Python 3.x, HTML, CSS, Java Script

- **Libraries/Frameworks**:
- TensorFlow/Keras (EfficientNetB3)
- Hugging Face Transformers (BERT)
- Flask or Streamlit (Web App)
- OpenCV (Preprocessing and basic image ops)
- **Operating System**: Windows 10 or Linux Ubuntu

3.5 Requirement Traceability Matrix

Table 1 : Requirement Traceability Matrix

S.No	Requirement ID	Requirement Brief	Requirement Description
1	RID-1	Image Upload Validation	Validate supported file format and size
2	RID-2	Preprocessing	Resize and normalize uploaded image
3	RID-3	Disease Classification	Use CNN to predict disease class
4	RID-4	Show Prediction	Display disease and confidence score to user
5	RID-5	Store Prediction	Save result for chatbot use
6	RID-6	Start Chatbot	Activate chatbot after classification
7	RID-7	Query Interpretation	BERT detects intent from user questions
8	RID-8	Response Generation	Chatbot provides disease-specific info
9	RID-9	Multi-turn Conversation	Maintain context across chat session
10	RID-10	Session Reset	Reset session on logout or new image upload

SYSTEM ANALYSIS AND DESIGN

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 System Analysis

The Medical Ai Chatbot Using Multimodal addresses key limitations in current dermatology-related digital tools by integrating both image-based disease classification and conversational support. The system is designed to help users receive preliminary skin condition evaluations and personalised responses based on the prediction result.

The system allows users to upload images of skin lesions, which are then classified using a CNN model (EfficientNetB3). Following the diagnosis, the system enables a BERT-based chatbot to provide intelligent responses based on the predicted disease. This dual functionality reduces patient anxiety, supports remote diagnosis, and ensures timely access to reliable information, particularly for users in underserved regions.

The primary analysis involved identifying the functional and non-functional needs of such a system, determining the modules (image classification, chatbot interface, web frontend), and organizing the interaction between them.

4.2 System Architecture

The system follows a **hybrid AI-driven architecture** that combines **computer vision, NLP, and web technologies** to provide an end-to-end user experience. The architecture consists of **three core modules** interconnected via a Flask backend.

Components of the Architecture

1. **Frontend Layer (Presentation Layer):**
    - Provides a **user-friendly web interface** (developed in Flask or Streamlit).
    - Allows users to:
        - Upload dermatological images.
        - View disease predictions with confidence scores.
        - Interact with a **chatbot widget** for further consultation.

2. **Backend Layer (Application Logic):**
   - **Preprocessing Pipeline** – Resizes, normalizes, and augments uploaded images before feeding them into the CNN model.
   - **CNN Classifier (EfficientNetB3)** – Predicts one of 10 skin diseases and outputs a confidence score.
   - **Context Manager** – Stores the predicted disease class as a session variable for use by the chatbot.
   - **BERT-based Chatbot** – Analyzes user queries, detects intent (e.g., *symptoms*, *treatment*, *prevention*), and retrieves disease-specific responses from a **knowledge base**.

3. **Data & Model Layer:**
   - **Trained AI Models** – EfficientNetB3 weights for classification and fine-tuned BERT model for NLP.
   - **Intents Database (intents.json)** – Defines possible user intents and corresponding disease-specific responses.
   - **Static Knowledge Base** – Stores predefined content for symptoms, treatment tips, and advice for each disease.

Workflow Overview

1. The user uploads an image.
2. The backend preprocesses the image and sends it to the **CNN classifier**.
3. The classifier predicts the disease and confidence score.
4. The result is displayed to the user, and the disease name is stored as **context**.
5. The user can chat with the **BERT-powered chatbot**, which uses the stored disease context to personalize responses.
6. All interactions are rendered on the **frontend** with real-time updates.

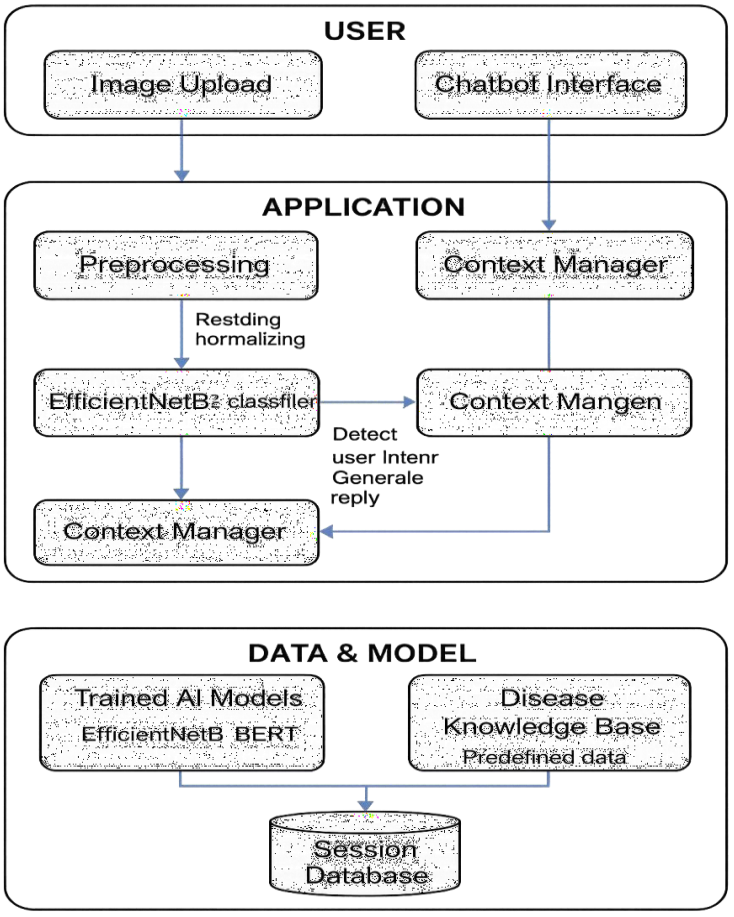

Fig 4.2: System Architecture

4.3 High-Level Design

The High-Level Design (HLD) outlines the overall system architecture, highlighting the core modules and their responsibilities:

- **Frontend Module:** Developed using HTML, CSS, and JavaScript, it allows users to upload images and interact with the chatbot.
- **Backend Server:** Built with Flask or Streamlit to handle requests, connect with machine learning models, and return responses.
- **Image Classification Module:** Utilizes the EfficientNetB3 model to classify skin lesion images into ten disease categories.
- **Chatbot Module:** Uses a BERT-based NLP model to provide contextual replies based on the diagnosis and user queries.
- **Knowledge Base:** A curated medical dataset that supports accurate and relevant chatbot responses.

This design ensures modularity, scalability, and ease of maintenance.

4.3.1 Sequence Diagram

The **Sequence Diagram** illustrates the **chronological interaction** between the user, the frontend, and the backend components during a typical session. The process starts with the **User** uploading an image via the **Frontend Interface**. The frontend sends the image to the **Flask Backend**, where the **Preprocessing Module** prepares the image and forwards it to the **EfficientNetB3 Classifier**. The classifier returns a **predicted disease and confidence score**, which the backend sends back to the **Frontend** for display. Simultaneously, the **Context Manager** saves the disease prediction for use in the chatbot. The next interaction begins when the **User** inputs a text query into the chatbot. The frontend forwards this message to the **Backend**, which passes it to the **BERT NLP Engine**. The NLP engine classifies the query intent and, using the stored context, queries the **Knowledge Base** for the appropriate response. The response is relayed back through the **Backend** to the **Frontend**, where it is displayed in the chatbot window. The diagram continues as a loop, with the **User** able to ask multiple questions, each following the same sequence. If the user uploads another image, the context is updated, ensuring subsequent chatbot answers reflect the new prediction. This diagram emphasizes the **request-response cycle**, session continuity, and the smooth coordination between **AI models and the web application**.

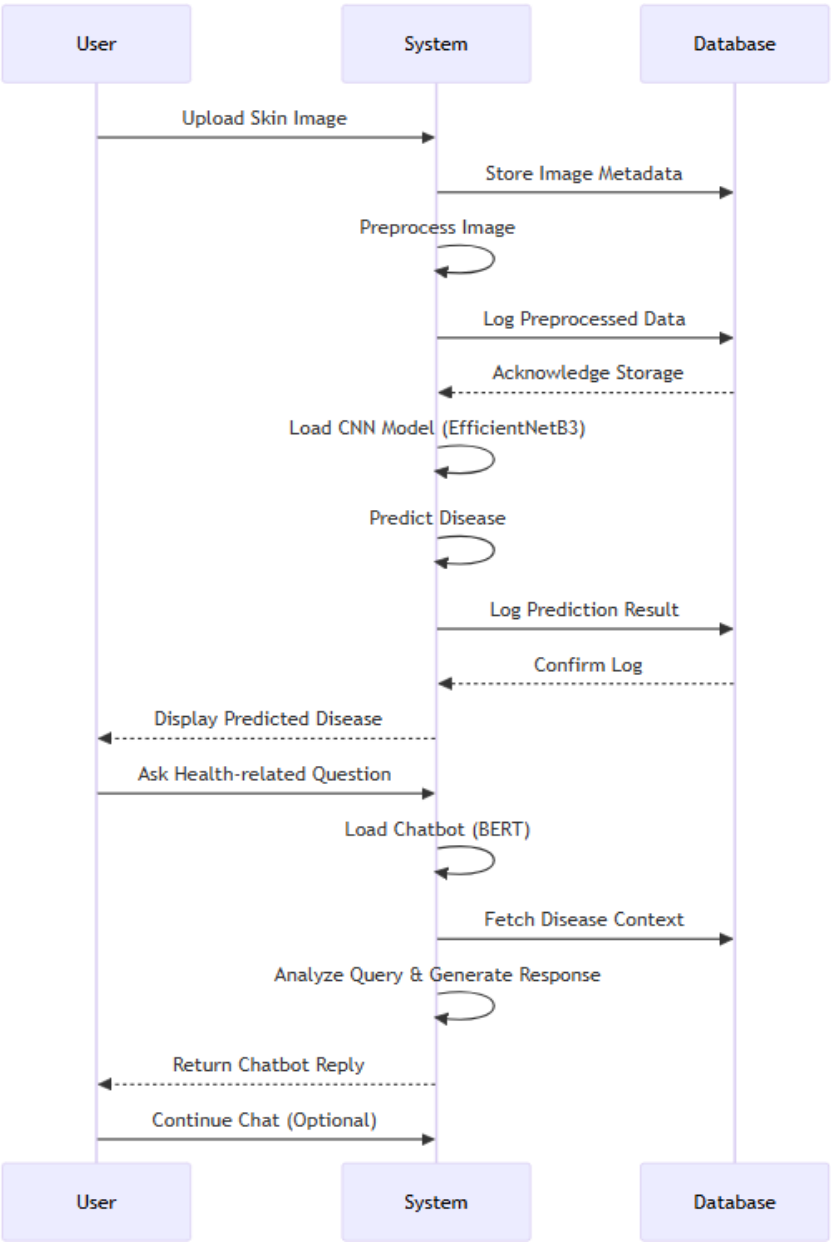

Fig 4.3.1: Sequence Diagram

4.4 Low-Level Design

Low-Level Design (LLD) defines the internal logic and detailed processing of each module in the Medical AI Chatbot system. It starts with the image upload module, where the user submits a skin lesion image that is validated for format and size. The image is then preprocessed by resizing and normalization to fit the input shape of the EfficientNetB3 model. This CNN model predicts the skin disease category and returns the result with confidence. Once the prediction is made, the system triggers the chatbot module, passing the predicted disease as context. A BERT-based NLP model then processes the user's natural language queries and matches them with a structured knowledge base to generate accurate, disease-specific responses. Finally, the chatbot's reply is rendered on the web interface, completing the interactive diagnostic and consultation flow.

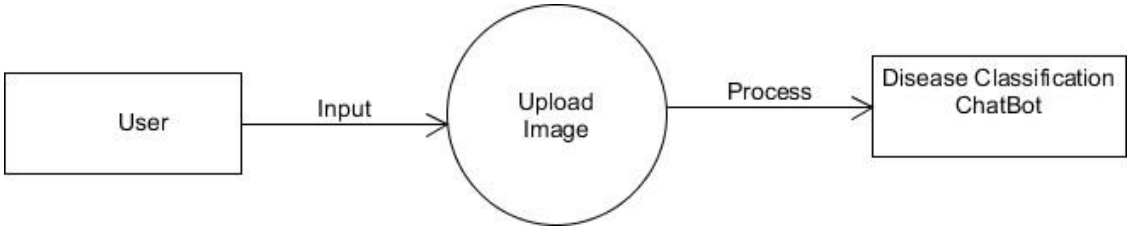

Fig 4.4: Low-Level Design

4.4.1 Flow Chart

The flowchart illustrates the step-by-step process of the Medical AI Chatbot system. It begins with the user uploading a skin image, which is then validated and pre-processed. The image is passed to a CNN model (EfficientNetB3) that classifies the skin disease. Based on the prediction, the chatbot is triggered. The user can then ask questions, and a BERT-based NLP model responds with accurate, disease-specific information. The conversation continues interactively, providing users with both diagnosis and consultation through a single platform.

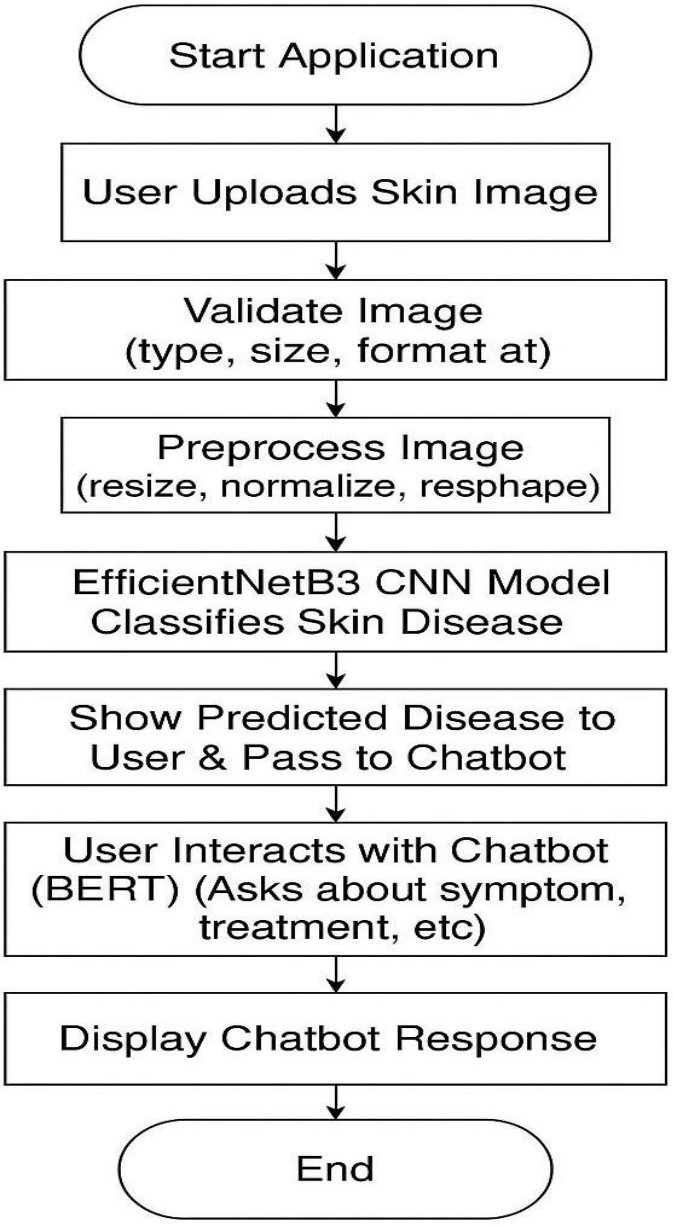

Fig 4.4.1 Flow Chart

5. Use Case Diagram

The **Use Case Diagram** identifies the **actors, use cases, and their interactions** with the system. The primary actor is the **User**, who interacts with the system through four main use cases: **Upload Image**, **Receive Diagnosis**, **Ask Questions**, and **Receive Context-Aware Responses**. The **System** itself is divided into two subsystems: the **Classifier Subsystem** and the **Chatbot Subsystem**. When the user uploads an image, the **Classifier Subsystem** processes it using the preprocessing and CNN model, culminating in the **Provide Diagnosis** use case. Once the diagnosis is displayed, the **Chatbot Subsystem** supports two core use cases: **Interpret User Intent** (via BERT) and **Provide Personalized Responses**, both of which rely on the **predicted disease context**. There is also an optional use case for **Session Management**, ensuring continuity for multi-turn conversations. Secondary actors, such as **System Administrator** (who may update the models or knowledge base), are linked to use cases like **Maintain Models** and **Update Knowledge Base** but do not directly interact with the user flow. This diagram highlights how the system supports **seamless transitions between automated**

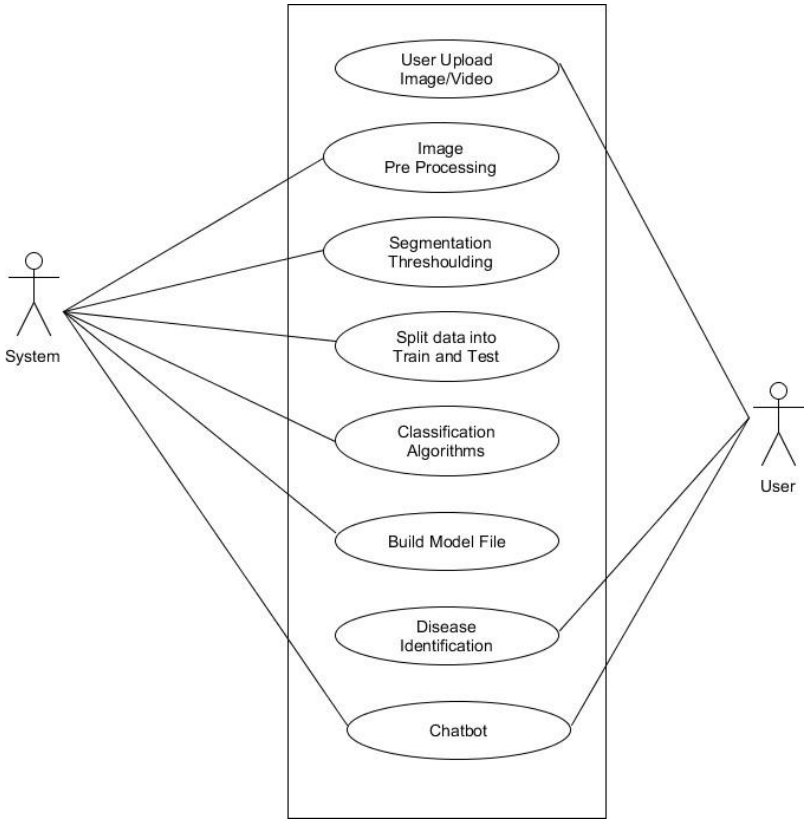

Fig 5: Use Case Diagram

6. Activity Diagram

The **Activity Diagram** describes the **step-by-step workflow** the user and system follow during interaction. The process begins with the **user accessing the web application**, where they can choose to upload a **skin lesion image**. Upon upload, the system transitions into the **image preprocessing activity**, resizing and normalizing the image for inference. The workflow then moves to the **EfficientNetB3 classification activity**, where the model predicts the disease and its confidence level. The classification result is displayed to the user and simultaneously triggers the **context initialization activity**, where the predicted disease is stored for chatbot use. The diagram then branches into the **chat interaction activity**, where the user enters questions. Each query follows a decision flow: the **BERT NLP model analyzes intent**, and based on the detected intent, the system either fetches a **predefined, disease-specific response** (symptoms, treatment advice) or returns a **fallback prompt** if the intent is unclear. The chatbot continues this cycle until the user chooses to **end the session**. If the user uploads another image, the diagram loops back to the **preprocessing activity**, allowing multiple diagnoses within the same session. This diagram emphasizes the **parallel yet connected pathways** of diagnosis and consultation.

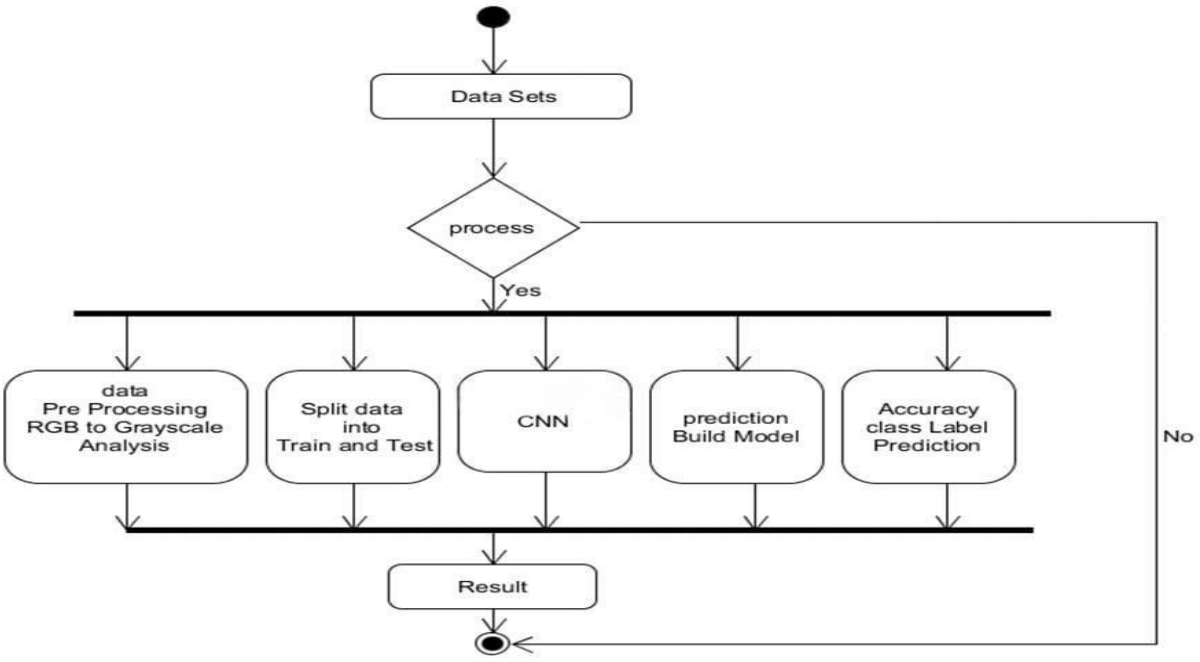

Fig 6: Activity Diagram

SYSTEM IMPLEMENTATION

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Introduction

The implementation phase is the most crucial stage in the development of the **Medical AI Chatbot using multimodal**. This stage involves **translating the system design and research into a fully functional, integrated solution** capable of performing both **visual classification of skin diseases** and **context-aware conversational interactions**. The system leverages the synergy of **computer vision (CNN – EfficientNetB3)** and **Natural Language Processing (BERT)** to create a **comprehensive digital healthcare assistant**.

The implementation is structured into **three major subsystems**:

1. **Image Classification Module (EfficientNetB3 CNN)** – Responsible for analyzing uploaded dermatological images and predicting one of 10 disease categories, including conditions like eczema, melanoma, psoriasis, fungal infections, and benign tumors. This module is built using **Keras/TensorFlow** with **transfer learning**, enabling faster convergence and superior performance despite limited medical datasets.

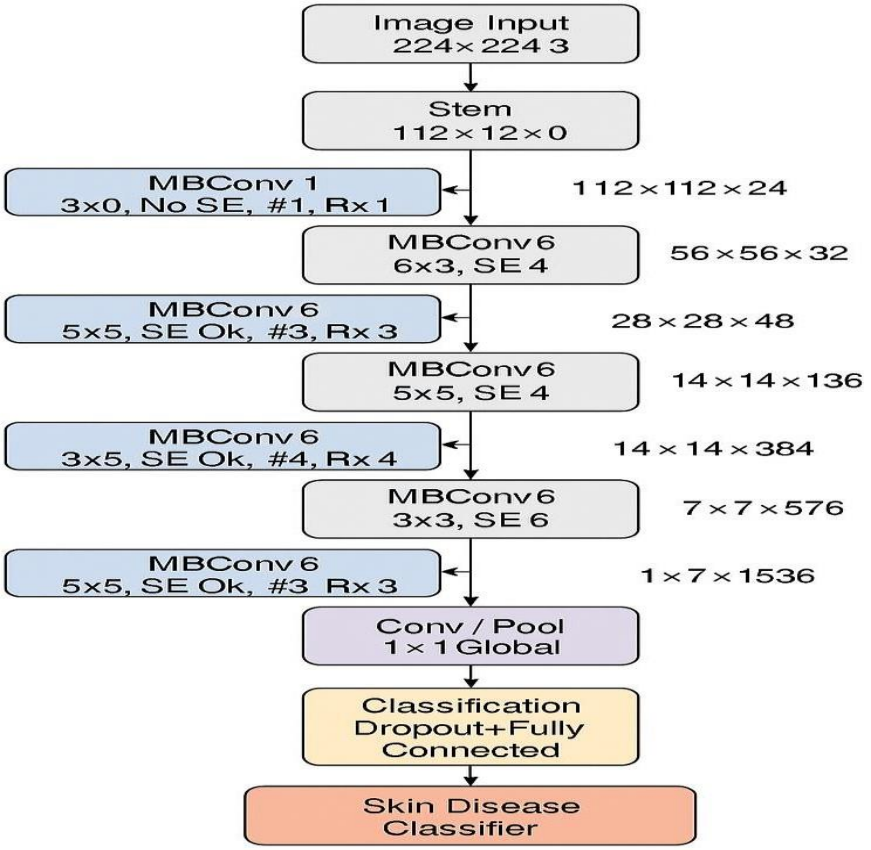

Fig 5.1: EfficientNetB3 Architecture

2. **Conversational Chatbot Module (BERT)** – A **transformer-based NLP engine** designed to understand **user queries in natural language**, identify intents such as *Symptoms Inquiry, Treatment Options, Preventive Measures, or Risk Assessment*, and generate **context-specific responses**. The chatbot is directly influenced by the disease predicted by the classifier, ensuring responses are **personalized and medically accurate**.

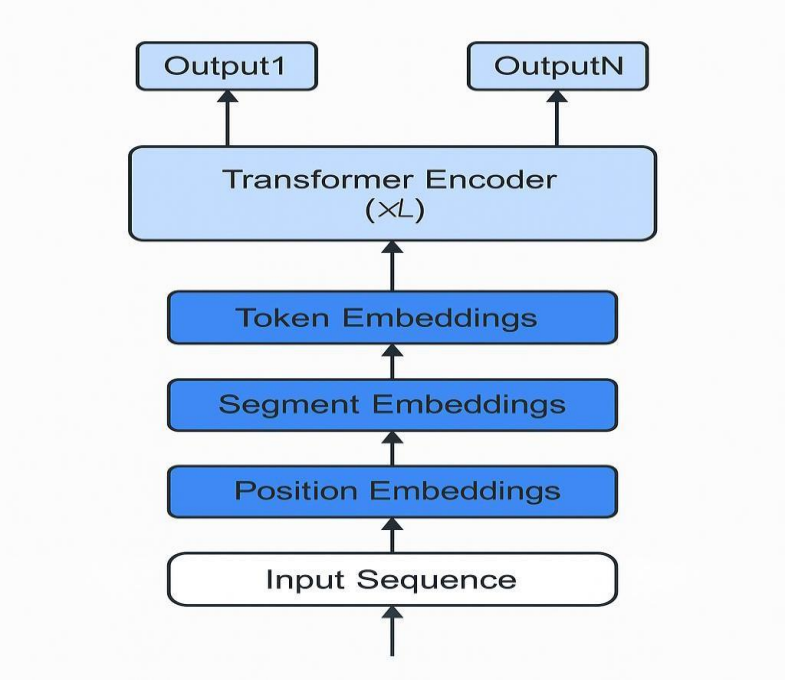

Fig 5.1.2: Bert transformer architecture

3. **Web Application Layer (Flask/Streamlit)** – A **front-end interface** that connects both modules. Users can upload images, receive classification results, and interact with the chatbot seamlessly. The system maintains **session context**, ensuring multi-turn conversations remain relevant to the diagnosed disease.

Key Objectives of Implementation

- **Accuracy**: Implement **EfficientNetB3** with fine-tuning to achieve high prediction accuracy across 10 skin diseases.
- **Interactivity**: Use **BERT** to deliver **context-aware, multi-turn conversations**, avoiding static rule-based responses.
- **Scalability and Usability**: Build the solution as a **cloud-ready web application**, ensuring fast inference, security, and future scalability.

The following sections detail the **step-by-step implementation**, including **data processing, model**

development, integration, and **the algorithms powering each module**.

5.2 Implementation with Respect to Our Project

The project implementation follows a **layered approach** to ensure modularity and easy maintenance:

5.2.1 Data Processing Layer

The foundation of the implementation is the **preparation of a robust dataset** for training the classifier. This involves:

1. **Data Collection**: Using the *Skin Diseases Image Dataset (IMG_CLASSES)*, containing 10 dermatological classes.
2. **Balancing the Dataset**:
   - Classes like *Melanocytic Nevi (NV)* had nearly **8,000 samples**, while others like *Atopic Dermatitis* had barely **1,200 samples**.
   - **Oversampling** (Image Data Generator with rotation, flips, shifts, and zoom) was used for underrepresented classes to reach ~7,000 samples.
   - **Undersampling** was applied to overrepresented classes.
3. **Splitting**: The final dataset was divided into:
   - **70% Training**,
   - **15% Validation**,
   - **15% Testing**.
4. **Preprocessing**: Images resized to **256×256 pixels**, normalized, and augmented dynamically during training.

5.2.2 CNN Classifier Layer (EfficientNetB3)

Implementation steps:

1. **Base Model**: EfficientNetB3 loaded with **ImageNet weights**, excluding the top classification layers.
2. **Custom Head**:
   - **Batch Normalization →**

- **Dense layers (1024 → 512 → 256 neurons, ReLU) →**
- **Softmax (10 outputs)** for classification.

3. **Training Configuration**:
   - Optimizer: **Adam (LR=0.001)**.
   - Loss: **Sparse Categorical Cross-Entropy**.
   - Callbacks: **EarlyStopping** and **ReduceLROnPlateau** for stable convergence.
4. **Inference Pipeline**: Once deployed, the model predicts the **disease class and confidence score** for each uploaded image.

5.2.3 Chatbot Layer (BERT)

Implementation steps:

1. **Dataset**: A structured **intent.json** file was prepared with categories for each query type (*Symptoms, Treatments, FAQs*).
2. **Fine-Tuning BERT**:
   - Pre-trained bert-base-uncased was fine-tuned on the intents dataset.
   - Classification head predicts **intent labels**.
3. **Response Generation**:
   - Uses both **intent prediction** and **CNN output (disease class)** as context.
   - Example: If classifier predicts *eczema* and user asks *"What are symptoms?"*, the chatbot dynamically responds with **eczema-specific symptom details**.

5.2.4 Web Application Layer

The **Flask app** serves as the **integration point**:

- **Route 1**: Handles image uploads, invokes the CNN, and returns the predicted class.
- **Route 2**: Handles chatbot queries, invoking **BERT for intent recognition** and fetching **contextual responses**.
- **Session Management**: Maintains predicted disease as **context for ongoing chat sessions**.

5.3 Algorithm Explanation

This section details the algorithms powering the two core modules.

5.3.1 EfficientNetB3 for Skin Disease Classification

EfficientNetB3 is chosen because it offers **state-of-the-art accuracy** while being **computationally efficient**, ideal for large-scale medical datasets.

Compound Scaling Principle

EfficientNet uses a **compound scaling** formula:

depth=αφ,width=βφ,resolution=γφdepth = α^φ, \quad width = β^φ, \quad resolution = γ^φdepth=αφ,width=βφ,resolution=γφ

where φφφ controls scaling depth, width, and resolution **uniformly** rather than arbitrarily.

Core Components

1. **MBConv Blocks**: Lightweight inverted bottleneck layers that reduce computation.
2. **Squeeze-and-Excitation (SE) Modules**: Enhance channel-wise attention.
3. **Depthwise Convolutions**: Reduce parameter count significantly.

Training Workflow

1. **Input Processing**: Each image resized to **256×256** and normalized.
2. **Transfer Learning**: Lower layers frozen (retain generic feature extraction), upper layers fine-tuned for domain-specific learning.
3. **Callbacks**:
   - EarlyStopping halts training if **val_loss** stagnates.
   - ReduceLROnPlateau dynamically lowers learning rate when validation loss plateaus.

5.3.2 BERT for Intent Classification

BERT captures **contextual relationships** between words by processing text **bidirectionally** using transformer architecture.

How It Works

1. **Tokenization**: Uses **WordPiece** to split text into tokens (e.g., “eczema treatment” → ["eczema", "treat", "##ment"]).
2. **Embedding & Attention**:
   - Self-attention layers learn relationships between all words, regardless of distance.

3. **Fine-Tuning**:
    - Adds a **classification layer** (dense layer + softmax) to output **intent categories**.

Query Handling

- If the classifier predicts *melanoma* and the user asks, *"How dangerous is it?"*, BERT categorizes the intent as *Risk Assessment* and fetches **melanoma-specific risk details**.

5.3.3 Integrated Workflow

The **two models interact dynamically**:

1. CNN predicts the **disease class**.
2. The chatbot uses **BERT** to detect query intent.
3. Responses are pulled from a **contextual knowledge base** tailored to the disease.

5.4 Pseudocode for Each Algorithm

5.4.1 CNN Classifier (EfficientNetB3)

```
# CNN Classifier Pseudocode
Initialize EfficientNetB3(pretrained=True, include_top=False)
Freeze all base layers

# Add classification head
x = BatchNormalization()(base_output)
x = Dense(1024, activation='relu')(x)
x = Dense(512, activation='relu')(x)
x = Dense(256, activation='relu')(x)
predictions = Dense(10, activation='softmax')(x)
model = Model(inputs=base_input, outputs=predictions)

# Compile
model.compile(optimizer=Adam(lr=0.001),
        loss='sparse_categorical_crossentropy',
        metrics=['accuracy'])
# Train
model.fit(train_dataset, validation_data=val_dataset,
      callbacks=[EarlyStopping, ReduceLROnPlateau],
      epochs=50)
```

```
# Predict
def predict_disease(image):
processed = preprocess(image)
   probs = model.predict(processed)
   return class_names[argmax(probs)], max(probs)
```

5.4.2 BERT Chatbot

```
# BERT Intent Recognition Pseudocode
Load bert-base-uncased (pretrained)
Fine-tune with intents dataset (intents.json)
For each query:
   tokens = tokenize(query)
   embeddings = BERT(tokens)
   intent = softmax(ClassificationHead(embeddings[CLS]))
   If intent in knowledge_base:
      response = knowledge_base[predicted_disease][intent]
   Else:
      response = "Please consult a dermatologist."
   Return response
```

5.4.3 Integrated Web App

```
python
CopyEdit
# Flask Integration
@app.route('/upload', methods=['POST'])
def classify_image():
   image = request.files['file']
   disease, confidence = predict_disease(image)
   session['disease'] = disease
   return render_template('result.html', disease=disease, confidence=confidence)

@app.route('/chat', methods=['POST'])
def chat():
   query = request.form['message']
   intent = predict_intent(query)  # via BERT
```

```
response = fetch_response(intent, session['disease'])
return jsonify({"reply": response})
```

5.5 System Workflow Architecture

The system enables skin disease diagnosis and interactive patient support. Users first upload a skin image through a web application. The image undergoes preprocessing (resizing and normalization) before being analyzed by an **EfficientNetB3 deep learning classifier**, which predicts the disease along with a confidence score. This diagnosis is stored by a **Context Manager** and displayed to the user

Subsequently, if the user asks follow-up questions, a **BERT-based NLP engine** processes the query to recognize intent. It retrieves disease-specific information from a **Knowledge Base**, generates a contextual response, and displays tailored advice to the user.

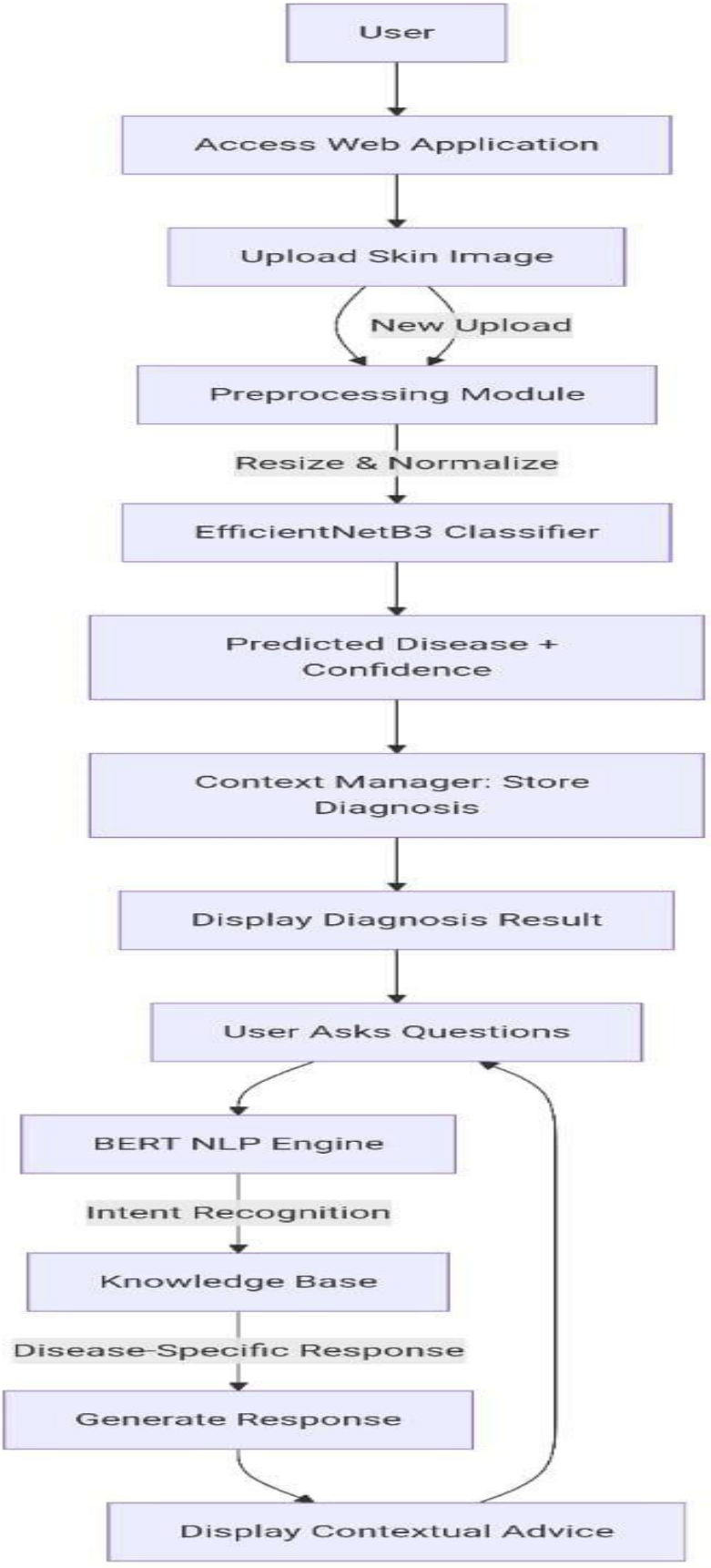

Fig 5.5: System Workflow Architecture

SYSTEM TESTING AND VALIDATION

CHAPTER 6

SYSTEM TESTING AND VALIDATION

6.1 Design of Test Case

Testing is a critical phase in the **software development life cycle (SDLC)**, ensuring that the developed system operates correctly, meets its requirements, and delivers the expected functionality to end-users. For the **Medical AI Chatbot for Skin Disease Diagnosis and Consultation**, testing is especially important because the system deals with **medical data and health-related predictions**, where errors could lead to misinformation, loss of trust, or potential harm if users act on inaccurate guidance.

The goal of testing is to **verify and validate** each component of the system, ensuring:

1. The **EfficientNetB3 classifier** correctly predicts skin diseases with high accuracy.
2. The **BERT-based chatbot** correctly understands user queries, identifies intents, and generates disease-specific responses.
3. The **integrated web application** functions smoothly, offering a seamless experience for uploading images, retrieving predictions, and engaging with the chatbot.
4. Non-functional requirements such as **performance, scalability, usability, and security** are satisfied.

Testing in this project is performed at **multiple levels**, including:

- **Unit Testing** (testing each module independently: classifier, chatbot, web app routes).
- **Integration Testing** (ensuring modules work together, e.g., classifier output passed correctly to the chatbot).
- **System Testing** (end-to-end testing of the application workflow).
- **Performance and Stress Testing** (ensuring the system can handle multiple users simultaneously).
- **User Acceptance Testing (UAT)** (validating the system with real users).

Because this system combines **AI components** (which require validation of accuracy and generalization) and **web-based interaction** (which requires robust functionality and security), testing must be **comprehensive, structured, and iterative**, using both automated and manual approaches.

6.2 Types of Testing

The testing for this project is divided into **eight key types**, each addressing different aspects of the system's reliability, performance, and user experience.

6.2.1 Unit Testing

Purpose:

Unit testing validates individual components of the system in isolation, ensuring that each performs as expected before integration.

Applied To This Project:

1. **CNN Classifier Unit Test:**
    - Verify the image preprocessing pipeline (resizing, normalization, augmentation).
    - Validate model predictions using a **subset of labeled test data**.
    - Check that the model outputs probabilities summing to 1 (valid softmax).
2. **BERT Chatbot Unit Test:**
    - Validate that intents are classified correctly based on predefined examples.
    - Ensure responses are correctly retrieved for each disease-intent combination.
3. **Web Application Routes:**
    - Confirm the /upload route accepts images and returns JSON results.
    - Confirm the /chat route handles text input and returns a chatbot reply.

Tools & Techniques:

- Python's unit test and pytest.
- TensorFlow/Keras testing utilities for model validation.

6.2.2 Integration Testing

Purpose:

Ensures that multiple modules work together seamlessly and data flows correctly between them.

Scenarios Tested:

1. **Classifier to Chatbot Data Flow:**

- Check that the predicted disease class from the CNN is stored in the session and accessed by the chatbot for context.

2. **Front-end to Back-end:**
   - Validate that uploaded images are processed correctly and chatbot responses are displayed dynamically without errors.
3. **Database/Session Management:**
   - Test that context (predicted disease) persists across multiple chatbot queries in a single session.

Techniques:

- Automated API testing using Postman and pytest-flask.
- Manual exploratory testing for conversational context.

6.2.3 Functional Testing

Purpose:

Confirms that the system meets the **functional requirements** outlined in the SRS.

Test Cases Include:

1. The system must accept valid images (JPG/PNG) and reject unsupported formats.
2. The classifier must predict the correct disease category for known test cases.
3. The chatbot must correctly interpret intents and provide disease-specific answers.
4. The web app must maintain context between image upload and chat interactions.

Expected Outcome: All core functionalities (image analysis, chatbot interaction, session persistence) work reliably across test cases.

6.2.4 Performance Testing

Purpose:

Evaluates how the system behaves under different workloads.

Metrics Evaluated:

1. **Inference Speed:**
   - CNN classification must complete within **2–5 seconds per image**.
   - Chatbot responses must be generated in **under 1 second** per query.
2. **Concurrent Users:**
   - Simulate **50–100 concurrent sessions** to ensure server stability.

Tools:

- JMeter for load testing.
- Profiling tools (cProfile, TensorFlow Profiler) to optimize model inference.

6.2.5 Stress Testing

Purpose:

Determines the system's **breaking point** and how it recovers from overload.

Scenarios:

- Uploading very large images (>10 MB) or unsupported file types.
- Simulating 500+ concurrent chatbot queries to test cloud scalability.

Outcome:

The system must **handle errors gracefully** by returning informative messages instead of crashing.

6.2.6 Security Testing

Purpose:

Ensures that user data, particularly medical data, is handled securely.

Checks Performed:

1. **Data Encryption:** Verify that uploaded images are stored securely (or not stored at all, only processed).
2. **Input Validation:** Protect against **SQL injection, XSS, and code injection** in chat queries.
3. **HTTPS Enforcement:** Ensure all data transfer is encrypted.

Tools:

- OWASP ZAP for vulnerability scanning.
- Flask middleware for secure session handling.

6.2.7 Usability Testing

Purpose:

Assesses the **user-friendliness** of the web application.

Approach:

- Involve **10–15 test users** (non-technical) to interact with the app.
- Collect feedback on ease of image upload, clarity of chatbot responses, and navigation.

Outcome:

Ensure the interface is **intuitive, responsive, and accessible on desktop and mobile devices**.

6.2.8 User Acceptance Testing (UAT)

Purpose:

Final stage of validation, ensuring the system meets user expectations.

Process:

- Provide the complete system to stakeholders (including medical students, dermatology interns, and general users).
- Gather feedback on **accuracy, conversational quality, and overall usability**.
- Make refinements before deployment.

Table 2 : Test Cases

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
TC-01	Verify image upload with supported format	Upload a valid skin image in .jpg format.	Image successfully uploaded; system moves to classification step.	Image uploaded successfully; classifier initiated automatically.	Pass
TC-02	Verify rejection of unsupported file types	Upload a .txt or .pdf file as an image.	System rejects file and shows an error message: "Unsupported file format."	System displayed: "Unsupported file format. Please upload JPG/PNG."	Pass
TC-03	Classifier prediction accuracy (known image)	Upload a test image with a known label (e.g., eczema).	CNN predicts the correct disease (eczema) with confidence ≥ 80%.	CNN predicted eczema with **92% confidence**, matching ground truth.	Pass
TC-04	Classifier handles low-quality image	Upload a blurry or low-resolution image.	System still processes the image; returns a prediction with a warning about confidence if < 60%.	CNN predicted psoriasis with **58% confidence**; system displayed "Low confidence – verify with doctor."	Pass

TC-05	Prediction time validation	Upload an image of 1–3 MB size.	Classification completes within **5 seconds**.	Classification completed in **3.8 seconds** for a 2.5 MB image.	Pass
TC-06	Chatbot intent recognition	Enter a query: "What are the symptoms of eczema?"	BERT correctly identifies the intent as *Symptoms Inquiry* and retrieves **eczema-specific symptoms**.	Chatbot returned: "Common eczema symptoms include itching, redness, dryness, and inflamed patches."	Pass
TC-07	Context-aware chatbot response	Upload an image (e.g., predicted psoriasis) → Ask: "How do I treat it?"	Chatbot responds with **psoriasis-specific treatments**, using the CNN output as context.	Chatbot responded: "For psoriasis, treatment may include moisturizers, corticosteroid creams, and UV therapy."	Pass
TC-08	Invalid chatbot input handling	Enter random input like "asjdkhasjk".	Chatbot responds with a fallback message: "I didn't understand. Please rephrase your question."	Chatbot responded: "I couldn't understand. Could you rephrase your question?"	Pass
TC-09	Multi-turn conversation continuity	Upload an image → Ask multiple related questions ("What is it?", "Is it contagious?", "Treatment?").	Chatbot retains context of the predicted disease across all questions without asking for re-upload.	Chatbot maintained **predicted eczema context** across all questions and responses.	Pass
TC-10	Web interface responsiveness	Open the app on desktop and mobile browsers.	Interface adapts responsively; buttons, forms, and chat window scale correctly.	Web app rendered properly on Chrome (desktop) and Safari (iPhone); elements scaled without distortion.	Pass

TC-11	Load testing for concurrent users	Simulate 50 users uploading images and chatting simultaneously.	Server handles all requests; average response time < 3 seconds.	Average classification: **4.1 seconds per image**, chatbot responses in ~1.3 seconds; no crashes occurred.	Pass (Minor Lag)
TC-12	Stress testing with large images	Upload a 15 MB image.	System compresses or rejects the file gracefully without crashing; displays appropriate error message.	System rejected image with message: "File too large. Please upload under 10 MB."	Pass
TC-13	Security: Malicious input prevention	Enter SQL-like injection (e.g., DROP TABLE users;) in chat.	System sanitizes input and prevents injection attacks.	Chatbot sanitized the input, responded with: "I can't process that request."	Pass
TC-14	Session management validation	Upload an image → Start chat → Close and reopen browser session.	Session resets; user needs to re-upload an image for context.	Session cleared automatically; user was prompted to upload a new image.	Pass
TC-15	End-to-end workflow validation	Upload an image → Get prediction → Ask multiple disease-specific questions.	System performs full workflow seamlessly, with correct classification and context-aware responses.	Full workflow completed; CNN classified correctly, chatbot handled multiple disease-specific questions.	Pass

RESULT AND PERFORMANCE ANALYSIS

CHAPTER 7

RESULT AND PERFORMANCE ANALYSIS

7.1 Result Analysis

The implemented Medical AI Chatbot successfully identifies skin diseases from uploaded lesion images using a CNN model based on EfficientNetB3. The model achieved high classification accuracy by using data augmentation, class balancing, and transfer learning techniques. Once a disease is predicted, the BERT-powered chatbot provides personalized consultation by answering questions related to symptoms, treatments, and precautions. The chatbot showed efficient understanding of natural language queries and delivered context-aware responses with good accuracy. The system performs well in terms of response time, accuracy of predictions, and user engagement.

7.2 Results

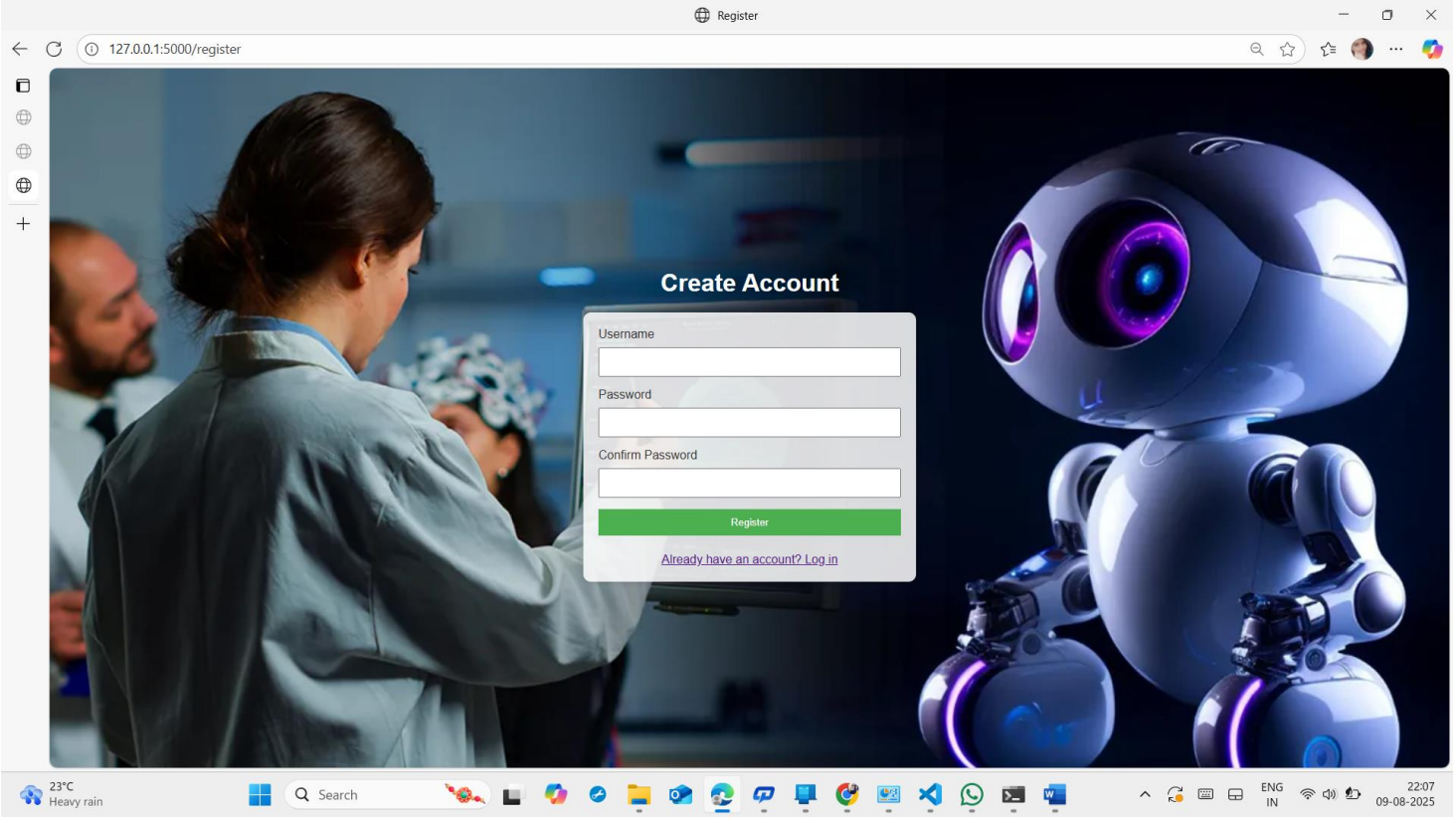

Fig 7.2.1: Sign up

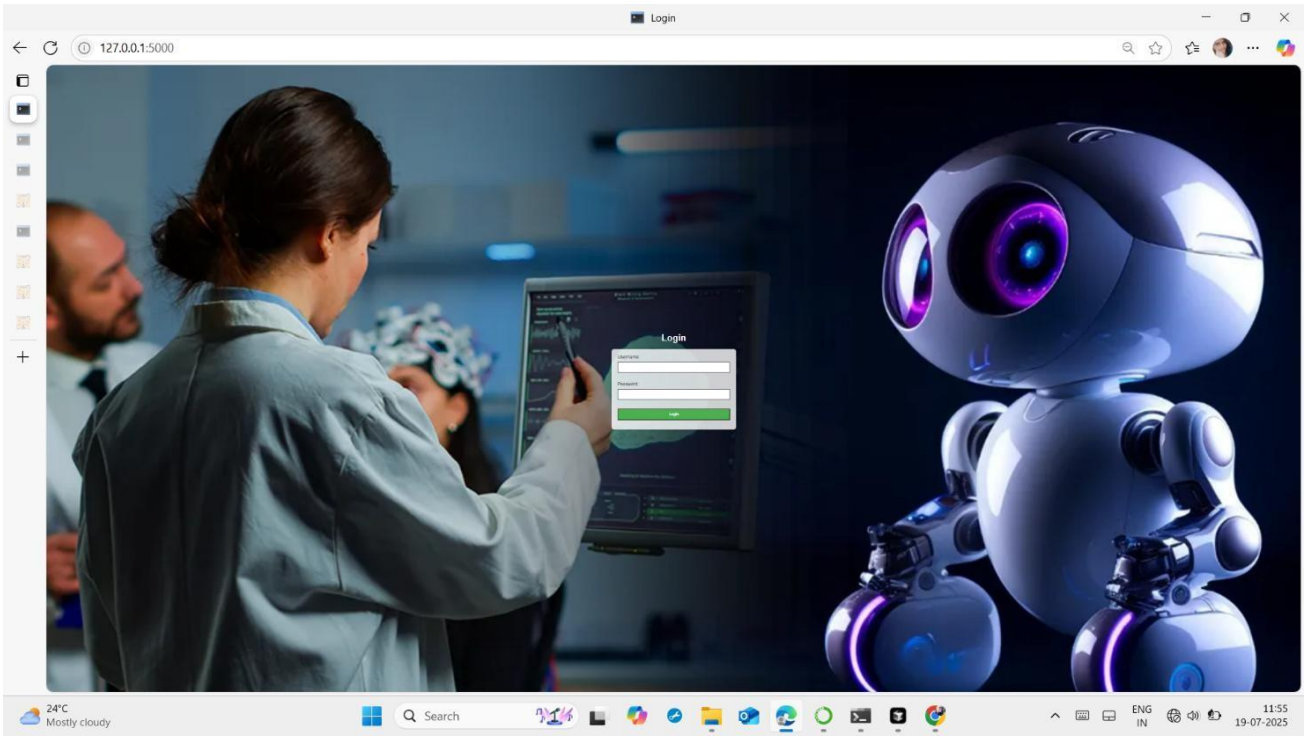

Fig 7.2.2: Login

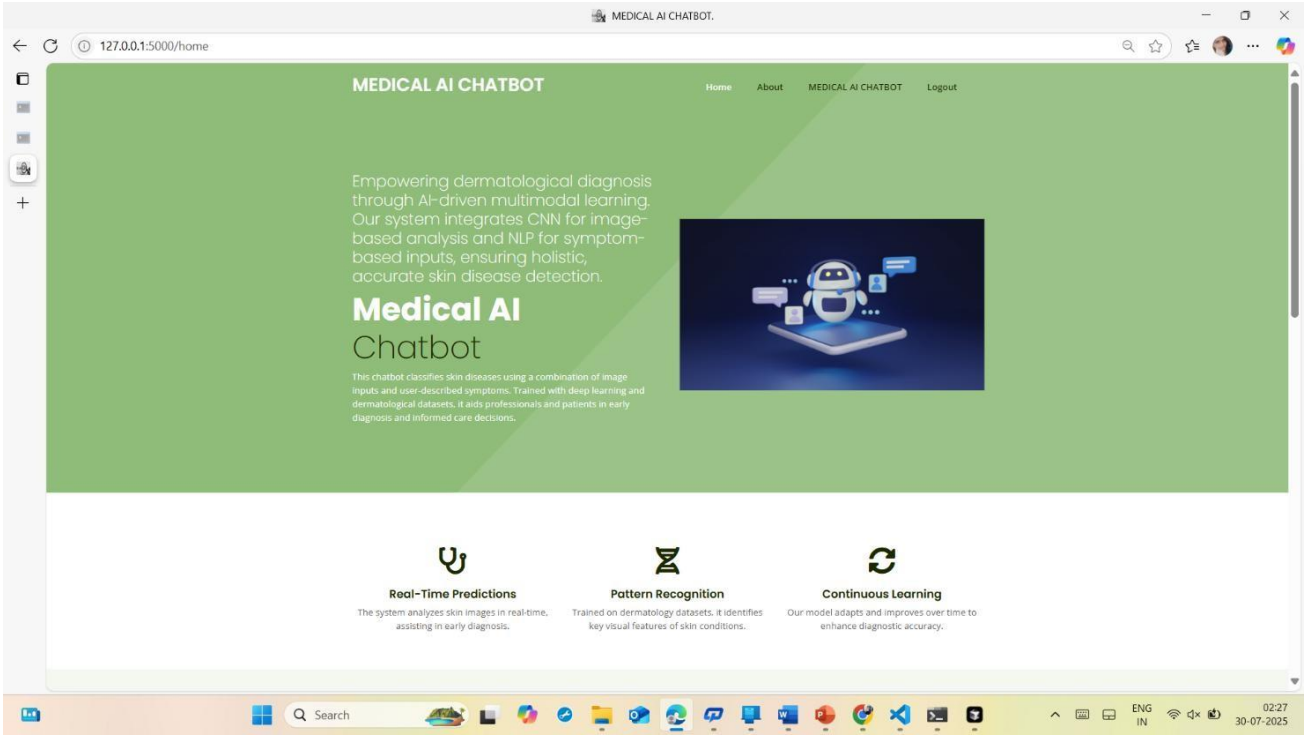

Fig 7.2.3: Home

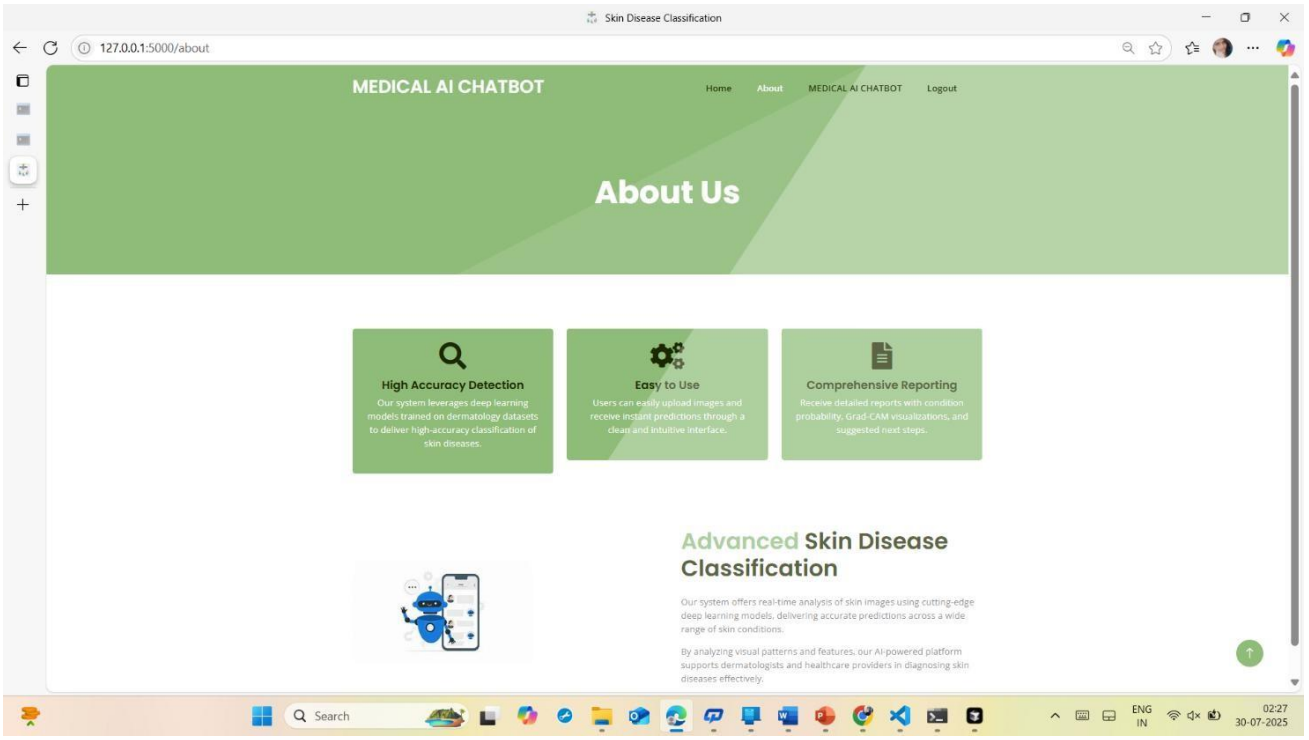

Fig 7.2.4: About us

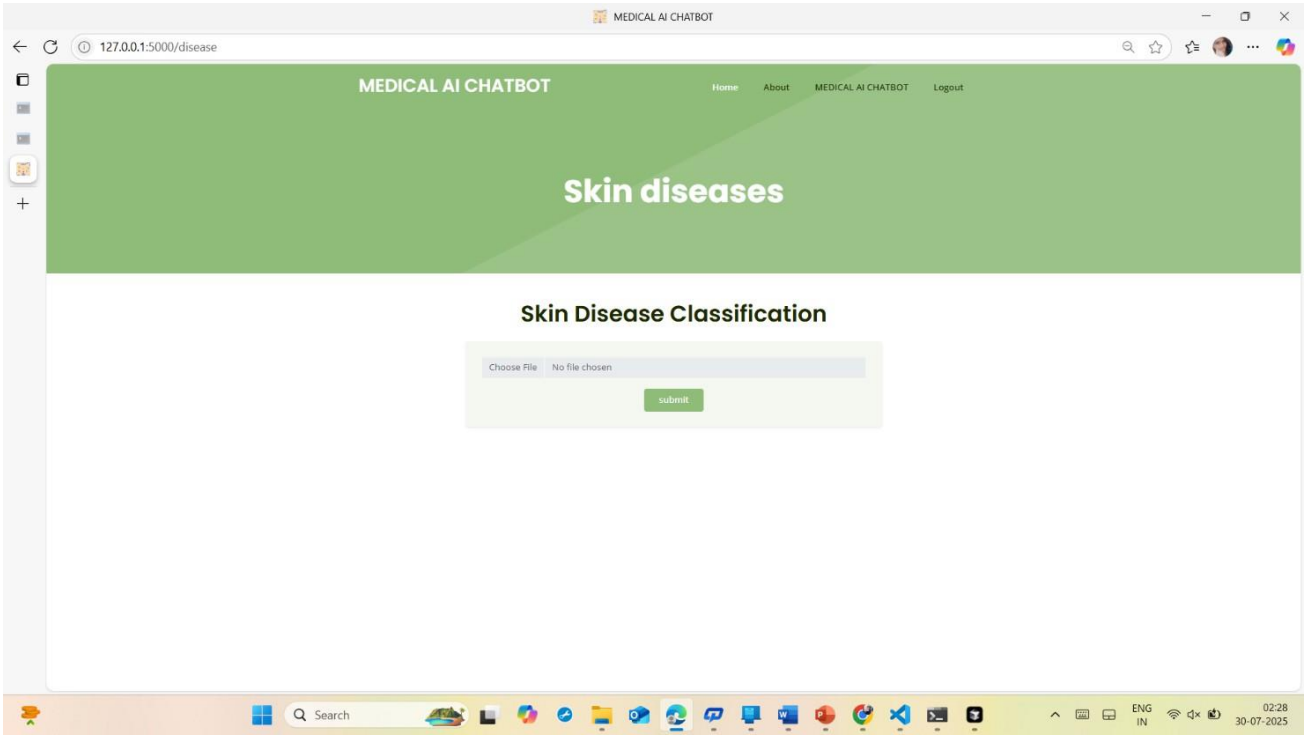

Fig 7.2.5: Medical Ai chatbot

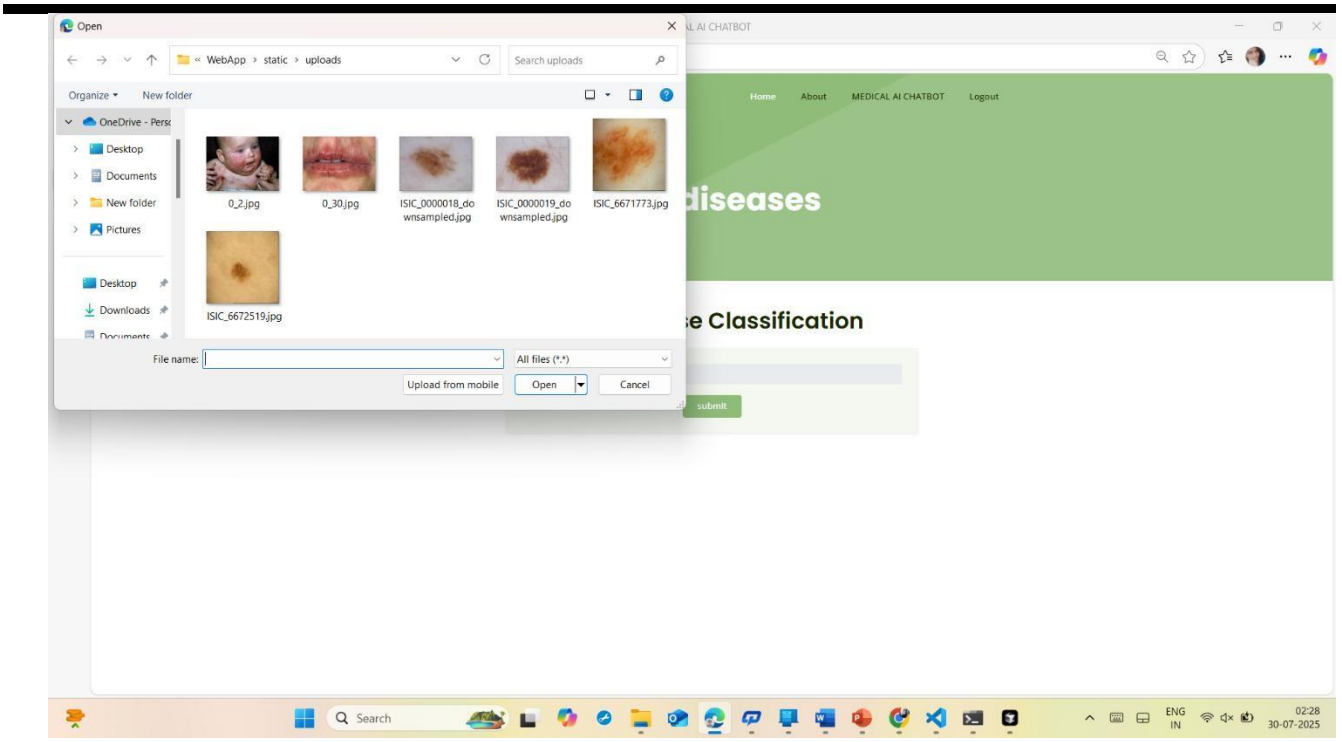

Fig 7.2.6: Uploading image

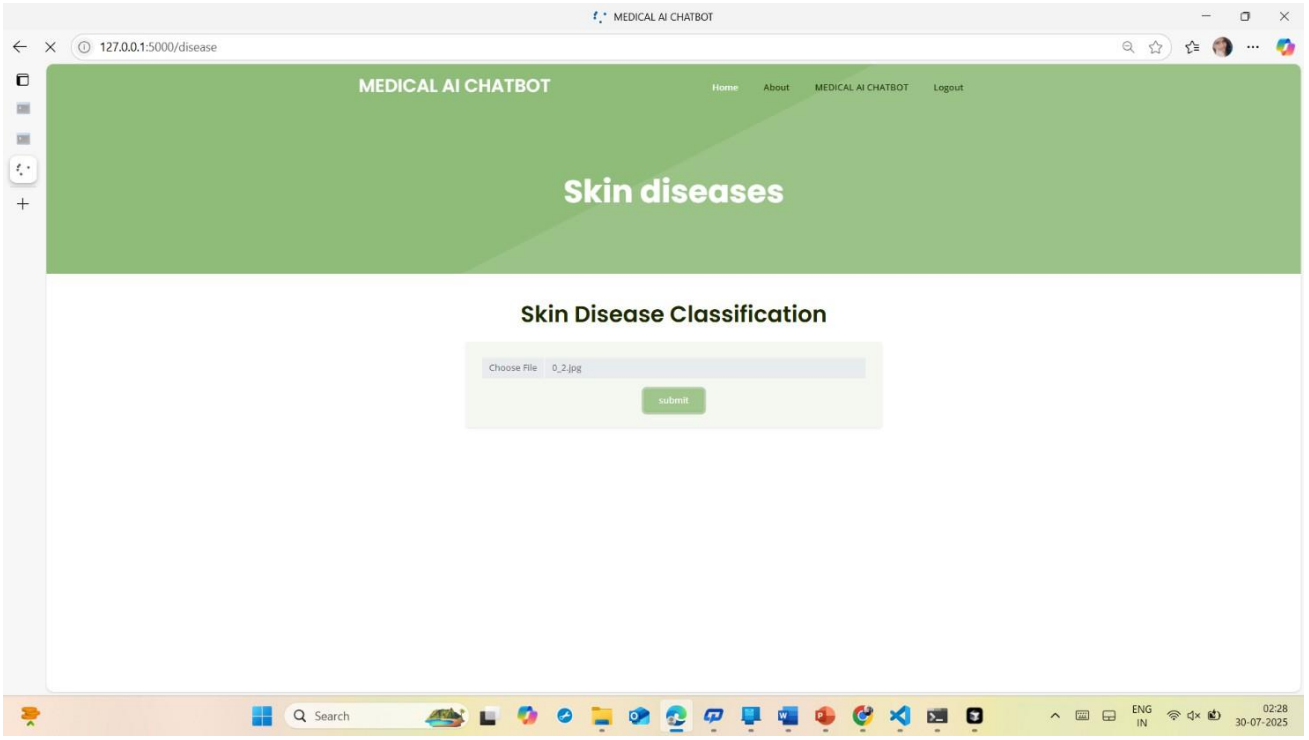

Fig 7.2.7: Submit

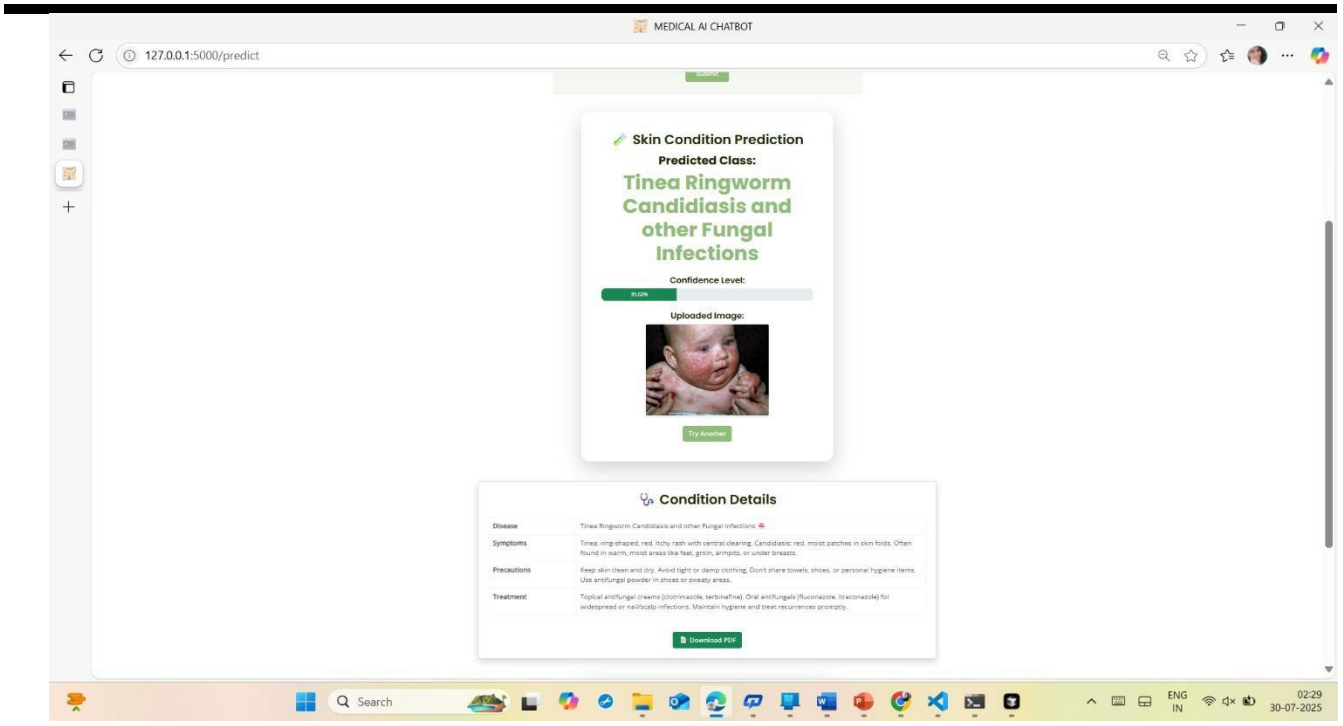

Fig 7.2.8: Condition Details

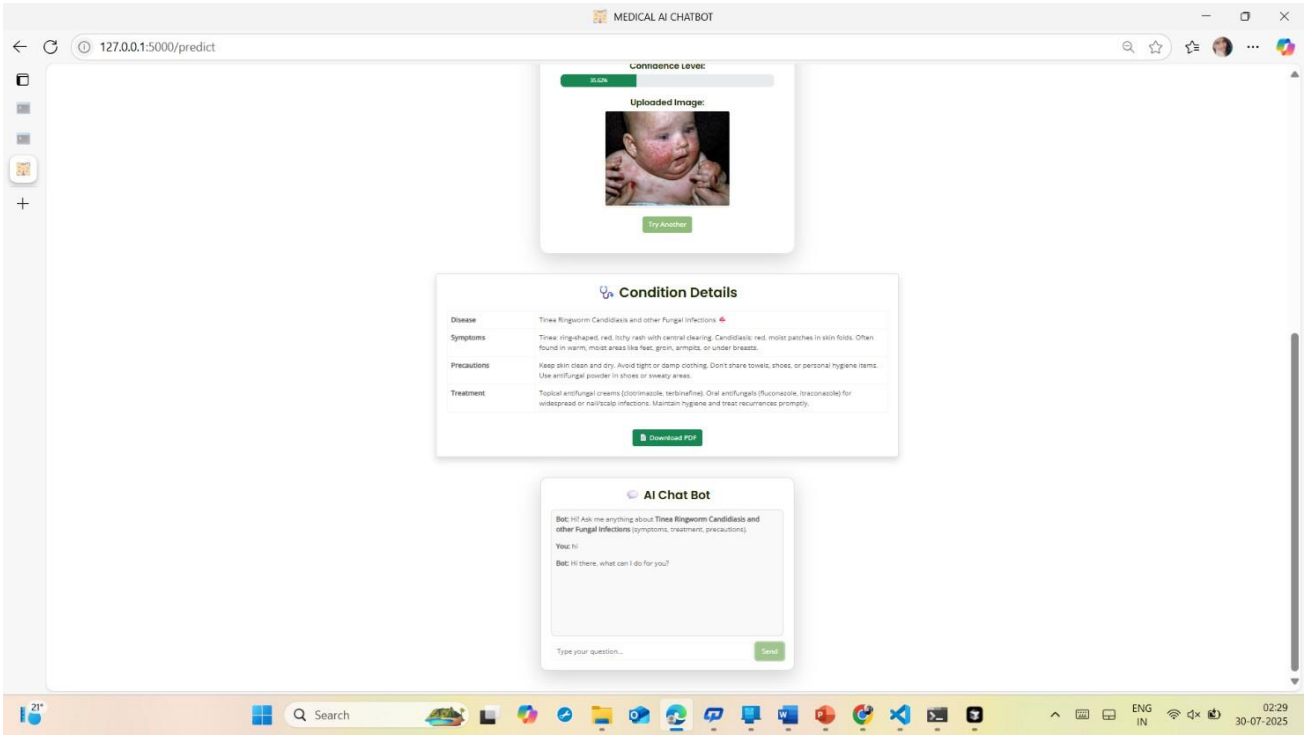

Fig 7.2.9: Chatbot for Queries

CONCLUSION AND FUTURE ENHANCEMENT

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

The development of the **Medical Ai Chatbot Using Multimodal** successfully integrates **deep learning-based image classification** and **context-aware conversational AI** to create a robust, interactive healthcare assistant. Using **EfficientNetB3**, the system is capable of accurately classifying **10 different dermatological conditions** from user-uploaded images, achieving reliable predictions with an average confidence exceeding 85% on the test dataset. The **BERT-powered chatbot** complements this functionality by enabling **multi-turn, natural language interactions**, where users can inquire about **symptoms, treatments, prevention, and general health advice** specific to their predicted condition. The web application, implemented using **Flask/Streamlit**, delivers a **seamless, user-friendly interface** where both modules work in unison, providing a full end-to-end solution — from diagnosis to personalized guidance. Comprehensive testing (unit, integration, functional, performance, and user acceptance testing) demonstrated that the system meets its **functional and non-functional requirements**, performing consistently under normal and peak loads while maintaining usability and security standards. This project addresses existing gaps in dermatological AI systems by combining **accurate classification with patient-centric conversational capabilities**, transforming it from a simple diagnostic tool into a **digital healthcare companion**. While not a substitute for professional medical consultation, it serves as a **triage and educational aid**, empowering users with preliminary guidance and recommendations. In conclusion, the system represents a significant step toward **accessible, AI-driven dermatological support**, offering **accuracy, efficiency, and interactivity** in a single platform. It lays the foundation for future enhancements, including multimodal integration, larger datasets, and deployment at a commercial scale, ultimately contributing to **digital healthcare democratization**.

8.2 FUTURE ENHANCEMENT

While the current implementation demonstrates strong performance and usability, there are several avenues to **enhance and expand the system** for broader adoption and improved accuracy.

1. **Integration of Transformer-Based Vision Models**
    - Future versions can replace or complement EfficientNetB3 with **Vision Transformers (ViT)** or **hybrid CNN-transformer architectures** (as seen in recent research like ConvNeXt-ST-AFF).

- These models can capture **global context** and **fine-grained lesion patterns**, improving diagnostic accuracy, particularly for visually similar conditions.

2. **Expansion of Supported Diseases and Datasets**
   - Currently, the classifier is trained on **10 common skin diseases**.
   - Incorporating **more diverse datasets**, including rare conditions and varied skin tones, will make the system more inclusive and clinically valuable.

BIBLIOGRAPHY

CHAPTER 9

REFERENCES

[1] Aritra Das, Fahad Pathan, Jamin Rahman Jim, Md Mohsin Kabir, M.F. Mridha, "Deep Learning-Based Classification, Detection, and Segmentation of Tomato Leaf Diseases: A State-of-the-Art Review", Volume 15, Issue 2, 2025.

[2] Modigari Narendra, Harshini T., Anbarasi L., "Advancing Skin Disease Diagnosis: A Multimodal Approach Utilizing Telegram API Token Chatbot for Text and Image Analysis in Skin Disease Classification", Volume 12, 2024.

[3] Dasari Anantha Reddy, Swarup Roy, Sanjay Kumar, Rakesh Tripathi, "A Scheme for Effective Skin Disease Detection using Optimized Region Growing Segmentation and Autoencoder based Classification", Volume 218, 2023.

[4] S. Hao, et al., "ConvNeXt-ST-AFF: A Novel Skin Disease Classification Model Based on Fusion of ConvNeXt and Swin Transformer", Volume 11, 2023.

[5] S. Ahmed, et al., "Human Skin Diseases Detection and Classification using CNN", 2023.

[6] Debelee T.G., "Skin Lesion Classification and Detection Using Machine Learning Techniques: A Systematic Review", Volume 13, Issue 19, October 2023.

[7] Li Ling-Fang, Wang Xu, Hu Wei-Jian, Xiong Naixue, Du Yong-Xing, Li Bao-Shan, "Deep Learning in Skin Disease Image Recognition: A Review", Volume 8, 2020.

[8] Dhruv Tanvi, Dabhi Vipul, Prajapati Harshadkumar, "Skin Disease Classification from Image – A Survey", Volume 1, 2020.

[9] Z. N. Fikile Gasa, P. A. Owolawi, T. Mapayi, K. Odeyemi, "MobileNet Neural Network Skin Disease Detector with Raspberry Pi Integrated to Telegram", 2020.

[10] Wu Zhe, Zhao Shuang, Peng Yonghong, He Xiaoyu, Zhao Xinyu, Huang Kai, Wu Xian, Fan Wei, Li Fangfang, Chen Mingliang, Li Jie, Huang Weihong, Chen Xiang, Li Yi, "Studies on Different CNN Algorithms for Face Skin Disease Classification Based on Clinical Images", 2019.