A Mini Project File

on

Viani-AI-powered instant body diagnosis tool

Submitted in partial fulfillment of the requirements

for the award of the degree of

Bachelor of Technology

in

Information Technology

by

Anjali Thakur(Roll. No.2300970130016)

Ankur Gautam(Roll. No.2300970130020)

Ashmi Singh (Roll. No.2300970130036)

Group No.: 25IT510

Under the Supervision of

Dr. Javed Miya & Mr. Anuj Gupta

Galgotias College of Engineering & Technology Greater Noida 201306

Uttar Pradesh, INDIA

Affiliated to

Dr. A.P.J Abdul Kalam Technical University

Lucknow

December 2025

GALGOTIAS COLLEGE OF ENGINEERING & TECHNOLOGY
GREATER NOIDA - 201306 , UTTAR PRADESH, INDIA

Declaration

We hereby declare that the project work presented in this project report entitled **"Viani-AI-powered instant body diagnosis tool"** in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Information Technology, submitted to A.P.J.Abdul Kalam Technical University, Lucknow, is based on our work carried out at Department of Information Technology, Galgotias college of engineering and technology, Greater Noida. The work contained in the report is original, and project work reported in this report has not been submitted by me/us for the award of any other degree or diploma.

Signature:

Name: Anjali Thakur
Roll No: 2300970130016

Signature:

Name: Ankur Gautam
Roll No: 2300970130020

Signature:

Name: Ashmi Singh
Roll No:
2300970130036

Date:
Place: Greater Noida

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to everyone who contributed to the successful development of our project, **"VIANI -AI-Powered instant body diagnostic tool"**

First and foremost, we extend our heartfelt thanks to our project guide, **Dr. Javed Miya , Mr. Anuj Gupta**, and **Dr. Sanjeev Kumar Singh (HOD, IT Department)**, for their unwavering guidance, constructive feedback, and insightful suggestions throughout this journey. Their expertise and encouragement inspired us to overcome challenges and consistently refine our work to achieve the desired results.

We also extend our sincere thanks to the professors and faculty of the **IT Department at GCET** for imparting valuable knowledge and skills that laid the groundwork for this project. The comprehensive learning in subjects like programming, system design, and web development proved instrumental in successfully implementing the platform.

We are immensely grateful to our teammates and friends for their dedication, collaboration, and creative input throughout this process. Their support, shared vision, and commitment to addressing challenges together played a pivotal role in shaping this project into a well-rounded and user-friendly platform.

Finally, we thank everyone who encouraged and supported us during this endeavor. Their belief in our abilities served as a constant source of motivation and ensured the successful completion of this project.

Name: Anjali Thakur
Roll No: 2300970130016

Name: Ankur Gautam
Roll No: 2300970130020

Name: Ashmi Singh
Roll No:
2300970130036

GALGOTIAS COLLEGE OF ENGINEERING & TECHNOLOGY,
GREATERNOIDA - 201306,UTTARPRADESH,INDIA.

CERTIFICATE

This is to certify that the Mini-project report entitled "Viani-AI-powered instant skin & body diagnosis tool" submitted by **ASHMI SINGH (2300970130036), ANJALI THAKUR (2300970130016), ANKUR GAUTAM (2300970130020)** to the **Dr.A.P.J Abdul Kalam Technical University Lucknow,** Uttar Pradesh in partial fulfillment for the award of Degree of Bachelor of Technology in Information Technology is a Bonafide record of the project work carried out by them under my supervision during the year 2025- 2026.

Mr Anuj Gupta
(Asst. Prof IT)

Dr . Javed Miya
(Professor)

Dr. Sanjeev Kumar Singh

(HOD IT)

ABSTRACT

Skin diseases represent one of the most common and socioeconomically impactful health conditions worldwide, often causing significant physical and psychological distress. Early diagnosis is crucial, yet access to dermatologists remains limited, especially in resource-constrained regions due to geographic barriers, specialist shortages, high consultation costs, and lack of awareness. Consequently, many skin conditions remain undiagnosed or misdiagnosed until they progress into severe stages. To address this gap, the proposed system VIANI introduces an intelligent, web-based preliminary dermatological assessment tool that integrates Natural Language Processing (NLP) and Convolutional Neural Networks (CNN) for dual-input analysis. Users can either describe symptoms in text or upload images of skin lesions, allowing broader usability and enhanced diagnostic accuracy.

The NLP module extracts clinically relevant features from textual descriptions, while CNN conditions such as acne, eczema, psoriasis, or healthy skin. The system follows a model server architecture with a React frontend, Flask backend, and separate AI inference components. VIANI provides predicted conditions, confidence scores, severity estimations, and basic care recommendations while emphasizing its role as an assistive tool rather than a substitute for professional diagnosis. The solution aims to improve early awareness and accessibility to dermatological care.

KEYWORDS : Skin disease detection, dermatological assessment, convolutional neural networks (CNN), natural language processing (NLP), dual-input diagnostic system, image classification, symptom analysis, web-based healthcare tool, machine learning in dermatology, client–server architecture, preliminary diagnosis, healthcare accessibility.

TABLE OF CONTENTS

PAGE

DECLARATION............ II

ACKNOWLEDGEMENT............ III

CERTIFICATE............ IV

ABSTRACT............ V

LIST OF TABLES............ VI

LIST OF FIGURES............ VII

1. INTRODUCTION............ 1

1.1 INTRODUCTION............ 1

1.2 MOTIVATION............ 2

2. PROBLEM STATEMENT 5

2.1 LIMITATION OF IMAGE ONLY DIAGNOSITIC MODEL............ 3

2.2 LIMITATIONS OF SYMPTOM ONLY NLP............ 6

2.3 LIMITATION OF TELEDERMATOLOGY CONSULTATION PLT...... 6

2.4 FORMAL PROBLEM STATEMENT............ 7

3. OBJECTIVES............ 8

3.1 PRIMARY OBJECTIVES............ 8

3.2 SECONDARY OBJECTIVE............ 9

3.3 DELIMINATION............ 10

4. LITERATURE REVIEW............ 11

4.1 OVERVIEW OF DERMATOLOGICAL AI RESEARCH............ 11

4.2 CNN BASED SYSYTEM............ 11

4.3 SYMPTOM BASED SYSTEM............ 12

5. SYSTEM DESSIGN AND ARCHITECTURE............ 14

5.1 SYSTEM OVERVIEW............ 14

5.2 DETAILED SYSYTEM ARCHITECTURE............ 16

6. TECHNOLOGIES USED 22

6.1 FRONTEND TECHNOLOGIES............ 22

6.2 BACHEND TECHNOLOGIES............ 23

6.3 USER INTERFACE MODULE............ 25

7. RESULT ANALYSIS............ 39

7.1 CODE IMPLEMENTATION............ 39

7.2 FINAL OUTPUT............ 41

8. CONCLUSION............ 44

REFRENCES............ 45

LIST OF TABLES

Table Title	Page
3.1 Objective-Outcome Matrix	9
4.1 Primary Constraints of CNN Based Diagnostic systems	12
4.2 System Limitation	12
4.3 Research gap summary	13
5.1 Fundamental Principles of architectural design	14
5.2 Endpoint of Flask Backend	18
5.3 Fallback and Error Handling Paths	21
6.1 Key Frontend Components	22
6.2 Key Backend Components	23
6.3 Library used in CNN Model	23
6.4 Future extension anticipated by architecture	25
6.5 Four Major React components	25
6.6 Description of Various Validation	26
6.7 UI Layout	26
6.8 Typography hierarchy	26
6.9 UI Display	27
6.10 Why Flask was selected	28
6.11 API Endpoint structure	29
6.12 Four High Prevelence categories for MVP development	30
6.13 Pre-processing pipeline of the images	30
6.14 Controlled Augmentation used by MVP	31
6.15 Suitable architecture for MVP Model	31
6.16 Model Limitation	32

6.17 Operations Performed 34

6.18 Symptoms Mapping 35

6.19 Severity Analysis 37

LIST OF FIGURES

Figure Title	Page
5.1 Backend Architecture of the app	16
6.4 CNN Model Analysis	32
6.5 Model analysis	36
7.1 Diagnosis Component	39
7.2 API Logic	40
7.3 Viani Home Page	41
7.4 Diagnosis Ouput	42
7.5 About Section	43

1. INTRODUCTION AND MOTIVATION

1.1 Introduction

Skin diseases constitute a substantial and multifaceted domain within medical science, representing conditions that range from transient irritations to chronic autoimmune disorders and life-threatening malignancies. The skin, as the largest and most externally visible organ of the human body, performs essential physiological and protective functions, including thermoregulation, immunological defense, sensory reception, and barrier protection against environmental pathogens. Owing to its continuous interaction with the external environment, the skin becomes susceptible to infections, inflammatory responses, genetic anomalies, and psychosomatic manifestations. Consequently, dermatological conditions are among the most prevalent health concerns worldwide, affecting individuals irrespective of age, sex, socioeconomic background, or geographic context. According to global burden analyses, skin diseases rank among the leading causes of disability-adjusted life years (DALYs), particularly in low- and middle-income regions, where medical access challenges exacerbate morbidity outcomes.

Despite the ubiquity of dermatological disorders, their accurate diagnosis remains inherently complex. Many skin diseases exhibit overlapping clinical symptoms such as erythema, scaling, papular formations, or hyperpigmentation, complicating differential diagnosis during preliminary assessment. Traditional dermatology relies on visual examination, dermoscopic evaluation, and in some cases, biopsy-based histopathological assessment. However, the diagnostic outcomes derived from visual inspection depend largely on the clinician's experience and expertise. Inter-physician diagnostic variability has been widely reported, particularly in early or atypical presentations. Furthermore, dermatology specialist density remains disproportionately low in many countries, leading to delayed diagnosis, disease progression, avoidable complications, and increased psychological distress among patients.

In parallel, conditions such as acne vulgaris, eczema (atopic dermatitis), and psoriasis—three of the most common non-malignant dermatological disorders addressed within this project—exert profound psychosocial impacts. Beyond physical discomfort, these diseases have been documented to significantly reduce quality of life, self-esteem, social confidence, and emotional stability, often comparable to the psychological burden observed in chronic systemic illnesses.

Social stigma, body image anxiety, and prolonged treatment cycles further accentuate emotional distress. Thus, the necessity for timely, accessible, and preliminary screening support is evident, especially for individuals unable to obtain immediate dermatological consultation.

Advancements in artificial intelligence (AI) and machine learning (ML) have introduced new paradigms in medical diagnostics. Convolutional Neural Networks (CNNs), in particular, have demonstrated exceptional performance in visual recognition tasks, including dermatology-specific image classification. Through hierarchical feature extraction and spatial pattern recognition, CNNs are capable of identifying morphologic characteristics such as lesion shape, border irregularity, pigmentation gradients, and texture variations, thereby enabling automated disease classification and preliminary triaging. Prior research in automated dermatological image analysis illustrates significant accuracy improvements when compared to non-specialist.

However, image-only diagnostic systems present inherent limitations. Many dermatological diseases are not solely defined by visual manifestations, but by contextual factors such as onset duration, itching intensity, localized sensations, seasonal triggers, allergies, and comorbidities. These qualitative descriptors cannot be captured purely through visual imagery. This limitation has led to the recognition that text-based symptom interpretation is a critical complementary dimension to dermatology diagnostics. Natural Language Processing (NLP) techniques, which enable computational understanding of human language, can extract clinically relevant descriptors from patient-entered symptom narratives. Studies integrating textual symptom data with visual assessment models have demonstrated increased diagnostic interpretability and contextual reliability, forming the conceptual basis for dual-input dermatological diagnostic frameworks.

The proposed system, VIANI, is conceptualized within this context. It will be implemented as a web-based AI-driven dermatological screening tool that integrates NLP-based textual symptom analysis with CNN-based skin lesion image classification, constituting a hybrid diagnostic architecture. The system is presently in its Minimum Viable Product (MVP) stage, with foundational model structuring, preliminary dataset alignment, and interface scaffolding underway. The web application's frontend will be built using ReactJS, employing a clean clinical white-and-blue visual design language, chosen to align with medical UI norms emphasizing clarity, sterility, and low visual distraction. The backend will utilize Flask, facilitating API-driven routing between user input and AI inference modules. NLP and CNN processing layers will operate independently yet converge into a unified inference pipeline, enabling a structured output comprising predicted condition, confidence score, severity estimation, and recommended general-care guidance.

Furthermore, the system architecture has been intentionally modular to support progressive scaling. In future iterations, VIANI may incorporate teledermatology consultation, electronic health record integration, cloud-deployed inference engines, and expanded multi-class disease classification leveraging extended dermatological datasets. The MVP stage prioritizes feasibility validation, preliminary accuracy benchmarking, usability testing, and UI-output clarity.

Thus, VIANI positions itself not as a replacement for clinical diagnosis, but as an assistive diagnostic intelligence framework designed to enhance early awareness, support patient education, reduce diagnostic latency, and encourage timely medical consultation. Its contribution aligns with emerging global healthcare trends emphasizing preventive care, digital medical accessibility, and AI-driven clinical decision support systems.

1.2 Motivation

The motivation for developing VIANI arises primarily from the technological limitations and research gaps present in contemporary dermatological diagnostic systems. While the field of computer-aided medical diagnosis has seen significant progress in recent years, particularly with the advent of deep learning and advanced image processing architectures, current automated dermatology platforms frequently suffer from unimodal input dependency—that is, they rely on either image-based classification or symptom-based questionnaire interpretation, but not both concurrently. This unimodal dependency significantly constrains diagnostic accuracy, contextual interpretation, and classification reliability, especially in cases where disease presentation varies across individuals or where visual features alone cannot sufficiently represent underlying symptomatology. The growing body of dermatological artificial intelligence research clearly indicates the need for diagnostic systems that integrate textual and visual clinical indicators to achieve meaningful diagnostic precision and real-world applicability.

The primary technological limitation in existing dermatology AI systems lies in the fact that image-based deep learning models, predominantly Convolutional Neural Networks (CNNs), classify diseases solely through observable lesion morphology. These models excel at identifying structural attributes such as lesion size, border irregularity, color distribution, texture gradients, and plaque patterns. However, a considerable number of dermatological conditions are not exclusively visually distinguishable. For example, contact dermatitis, eczema flare-ups, and early-stage rosacea often present with visually similar erythematous patches, yet their underlying causes, triggers, and symptomatic sensitivities differ extensively. Research in dermatological pattern classification highlights that overlapping visual phenotypes remain a major challenge for image-only classifiers, leading to misclassification in cases lacking distinct morphological contrast.

Moreover, diseases such as psoriasis and atopic dermatitis exhibit clinical heterogeneity, where severity, flare patterns, and symptom triggers vary across environmental, immunological, and genetic factors. Clinical reasoning for these diseases often requires evaluating textual or verbal symptom clues such as *itch intensity, burning sensation, time of onset, seasonal variation, irritant exposure, and emotional stressors*. These descriptors cannot be encoded into pixel-based feature learning. Thus, any system that depends solely on images inherently lacks the interpretive dimensionality necessary for robust dermatological assessment. This gap underscores the necessity for Natural Language Processing (NLP)–based symptom analysis, which derives diagnostic context by extracting semantically significant features from patient language inputs.

Simultaneously, traditional symptom-checker tools that rely exclusively on questionnaires or rule-based expert systems are equally insufficient. These systems often reduce clinical reasoning to keyword matching or static decision trees, lacking semantic interpretation, contextual adaptiveness, and the ability to infer symptom severity or disease progression dynamics. Research has shown that rule-based and statistical NLP systems frequently produce overgeneralized or loosely reasoned diagnostic conclusions, especially when users describe symptoms in non-medical language or when multiple overlapping symptoms occur simultaneously.

The problem therefore lies not in the absence of technology, but in the fragmented implementation of diagnostic intelligence. Dermatology is fundamentally a multimodal diagnostic field—visual, sensory, temporal, environmental, and experiential indicators collectively contribute to clinical assessment. The lack of integrated multimodal AI frameworks is the core research gap that motivates the development of VIANI.

The necessity of multimodal diagnostic integration is further supported by hybrid inference models proposed in mobile dermatology systems such as the CURETO diagnostic framework, which emphasizes synthesizing visual evidence with linguistic symptom descriptors to improve classification reliability.

The study explicitly concludes that “interpretation of visible dermatological patterns in isolation cannot adequately represent disease severity or identify differential causes,” reinforcing the need for systems that unify CNN and NLP pipelines.

Additionally, the motivation for VIANI’s architectural structure comes from the observed lack of accessible, deployable, and user-centric dermatological screening systems capable of functioning within consumer-level computing environments. Many dermatology AI research models are computationally dense, dependent on GPU-heavy servers, and not optimized for real-time inference or everyday usage. This directly restricts scalability and real-world applicability. VIANI, by contrast, is designed as a web-based MVP capable of lightweight computation, facilitated by efficient model compression, REST API–mediated communication, and modular backend execution layers.

Further supporting motivation is the glaring clinical accessibility gap highlighted in the global dermatology health burden literature. Dermatology specialists remain limited across rural and semi-urban environments, creating an uneven healthcare distribution.

While VIANI is not intended to replace medical professionals, it provides a triage-oriented early screening mechanism that encourages users to seek medical help at the correct stage rather than resorting to non-professional self-diagnosis or dismissive delay.

Thus, the motivation driving this project arises from:

1. AI research gap → absence of dual-input dermatology diagnostic models.
2. Clinical reasoning necessity → both symptoms and images are required for differential assessment.
3. Usability limitation → existing models are not accessible for real-world public use.
4. Public health need → dermatology accessibility inequity and diagnostic delay.
5. Technological opportunity → advancement of CNN + NLP multimodal inference.

The proposed VIANI MVP, even at its early phase, is therefore positioned as a technologically justified and clinically relevant intervention, contributing directly to the emerging transformation of AI-assisted digital dermatology.

2. PROBLEM STATEMENT

Dermatological diagnostic practice, while traditionally grounded in clinical expertise, visualization of cutaneous patterns, and evaluation of symptom progression, faces notable challenges in terms of accessibility, diagnostic consistency, interpretive precision, and timely early-stage intervention. The global healthcare ecosystem continues to experience significant disparities in the distribution of dermatology specialists, with shortages especially pronounced in rural, low-resource, and economically constrained environments. As documented in public health epidemiological research, skin diseases represent one of the highest-frequency drivers of outpatient medical visits in numerous developing regions, yet access to qualified dermatological care remains limited because of geographic distance, low specialist-to-population ratios, and elevated treatment costs.

Consequently, patients frequently rely on self-medication, informal advice networks, or delayed consultation, resulting in deterioration of disease severity and heightened psychological distress.

However, the primary challenge this project seeks to address is not merely the lack of dermatologists, but the fundamental limitations in existing automated and semi-automated dermatological diagnostic approaches. Current systems in digital dermatology fall largely into three distinct methodological categories:

1. Image-only deep learning classification systems
2. Symptom-based diagnostic decision-tree or form-driven systems
3. Teleconsultation-based dermatology video/chat support platforms

While each approach offers individual strengths, they also introduce inherent diagnostic gaps when examined within real-world usage scenarios.

2.1 Limitations of Image-Only Diagnostic Models -

Contemporary dermatology research has seen extensive adoption of Convolutional Neural Networks (CNNs) for lesion and skin disorder classification due to their well-established ability to learn spatial hierarchies and texture-level feature patterns. Studies have demonstrated CNNs achieving dermatologist-level performance in controlled dataset evaluations.

Yet, the performance of such systems is heavily contingent upon the visibility and surface-level distinctiveness of disease markers. In practice, many dermatological conditions exhibit high morphological overlap. For instance, erythematous plaques seen in psoriasis may visually resemble inflamed eczematous flares in atopic dermatitis; similarly, rosacea, fungal dermatitis, and contact irritation can present with comparable redness and sensitivity.

Furthermore, numerous dermatological disease indicators are non-visual:

1. Severity of itching or burning sensations
2. Pain or localized sensitivity
3. Trigger-response cycles (humidity, detergent exposure, hormones)
4. Temporal flare patterns (seasonal, stress-induced)

Image-based CNN models are incapable of inferring these descriptors, leading to systemic under-specification in diagnosis.

2.2 Limitations of Symptom-Only NLP or Form-Based Systems

On the opposite side of the spectrum, conventional web and mobile symptom-checkers rely primarily on questionnaires and rule-based matching, which reduces complex diagnostic evaluation to a structured but inflexible decision tree. These systems presuppose that:

1. The user understands medical terminology
2. The user provides complete and accurate symptom detail
3. Symptoms can be classified in isolation

Studies confirm that such systems frequently produce overgeneralized outputs, particularly when symptoms are described in subjective or colloquial language, which is typical outside clinical settings.

For instance:

1. “itchy rash” could correspond to eczema, psoriasis, fungal dermatitis, allergic contact dermatitis, or urticaria.
2. A user describing symptoms as “dry skin with pain” may omit visually distinguishing factors necessary for classification.

2.3 Limitations of Teledermatology Consultation Platforms

Telemedicine platforms solve accessibility to some extent but introduce scalability and cost barriers. Live dermatologist interpretation does not fundamentally reduce physician workload and remains constrained by:

1. Consultation scheduling delays
2. High per-session costs
3. Low patient throughput capacity
4. Requirement for live dermatologist intervention

Thus, while beneficial, teleconsultation is not a scalable screening solution.

Identified Research Gap

The technological gap lies in the absence of multimodal diagnostic intelligence — systems capable of integrating both visual dermatological indicators and contextual symptom semantics to produce a clinically meaningful preliminary diagnostic model.

What is required is a dual-input, hybrid diagnostic architecture in which CNN and NLP work together, allowing:

1. Visual lesion pattern classification
2. Interpretation of symptom descriptions and severity

3. Contextual inference that incorporates patient experience and disease history
4. Output synthesis and severity understanding grounded in both modalities

This concept is academically justified by hybrid diagnosis frameworks discussed in multimodal dermatology research.

2.4 Formal Problem Statement

There is currently no widely deployed, scalable, low-cost, AI-assisted dermatological screening system capable of integrating image-based lesion analysis with natural language symptom interpretation to generate clinically meaningful, context-aware, preliminary diagnoses accessible to general users.

Thus, the core problem addressed in this project is:

To design and develop an AI-powered dermatological assessment system that overcomes the limitations of unimodal diagnostic frameworks by integrating CNN-based image classification with NLP-based symptom interpretation to improve reliability, contextual awareness, classification accuracy, and user accessibility in early-stage dermatology screening.

3. OBJECTIVES OF THE PROJECT

The development of VIANI, an AI-assisted dermatological diagnostic support system in its Minimum Viable Product (MVP) phase, is guided by a set of structured and research-driven objectives that collectively aim to address the diagnostic limitations and technological shortcomings observed in current digital dermatology solutions. As established in prior chapters, existing dermatological diagnostic technologies predominantly rely on either image-based classification or rule-based symptom evaluation, each of which fails to capture the multimodal nature of clinical dermatological reasoning. Accordingly, the core objective of VIANI is to create an integrative, accessible, and context-aware diagnostic support mechanism that harnesses the complementary strengths of Convolutional Neural Networks (CNNs) for visual lesion interpretation and Natural Language Processing (NLP) for semantic symptom understanding. This approach not only enhances diagnostic contextuality, but also provides users with structured, interpretable, and actionable diagnostic guidance.

The objectives of this project extend beyond mere model development and include deeper considerations of usability, scalability, data pipeline integrity, user-centered interface design, predictive confidence communication, and system adaptability in real-world usage scenarios. The project emphasizes designing a system that is not only computationally intelligent but also socially and medically responsible in its operational philosophy. Since VIANI is positioned as a screening and awareness-support tool rather than a substitute for clinical dermatological examination, the system's objectives must also incorporate ethical transparency, ensuring that users understand the advisory nature of the predictions generated.

Another key aspect influencing the project objectives is the emphasis on modular system architecture. The MVP-level structure is intentionally designed such that improvements can be introduced without requiring complete reconfiguration. This includes the ability to expand disease classes, integrate larger and more diverse datasets, incorporate real-time user feedback in iterative retraining processes, and eventually provide pathways for physician-assisted teledermatology consultations. The project, therefore, prioritizes architectural scalability and structured evolution capacity as intrinsic design goals.

The objectives of the proposed system may thus be articulated as follows:

3.1 Primary Objectives

1. To design and develop a dual-input dermatological diagnostic framework that integrates CNN-based image analysis with NLP-based symptom interpretation to provide a more context-aware and reliable preliminary assessment mechanism compared to unimodal diagnostic systems.

2. To ensure that the system operates as an accessible, web-based MVP, deployable on consumer-grade hardware without requiring specialized computing infrastructure, thereby enhancing accessibility across geographically and economically diverse user populations.

3. To create a structured diagnostic output model that not only provides disease class predictions but also communicates confidence levels, inferred severity, potential triggers, and first-line care recommendations in a medically responsible and user-comprehensible manner.

4. To design a clinical white-and-blue themed graphical user interface (GUI) that adheres to medical usability standards, prioritizing clarity, minimal visual overload, semantic grouping of information, and intuitive navigation, especially for first-time or non-technical users.
5. To maintain modularity in system architecture, enabling independent modification, retraining, optimization, and replacement of NLP and CNN components, thus facilitating future scalability and iterative refinement.

3.2 Secondary Objectives

1. To curate and preprocess representative dermatological datasets for training and validating the CNN classifier while ensuring balanced class representation and reduced model bias.
2. To construct an NLP model capable of interpreting diverse forms of user symptom language, including colloquial phrasing, incomplete descriptions, and non-technical terminology, thus improving system robustness.
3. To establish a standardized internal inference pipeline which aligns and resolves outputs from both the CNN and NLP modules into a coherent diagnostic probability structure.
4. To ensure transparent system behavior through interpretability mechanisms, such as clarity in explanation of predictions, disclaimers on diagnostic scope, and user education prompts encouraging professional consultation when necessary.

Table No.- 3.1 Objective-Outcome Matrix

Objective	Description	Expected Outcome	Evaluation Method
Dual-Input Diagnostic Integration	Combine CNN-based visual analysis with NLP-driven symptom understanding	Increased diagnostic contextual accuracy and reduced misclassification	Comparative evaluation against image-only and text- only baselines
Accessible Web-Based Deployment	Develop thesystem as a browser-accessible application	Broad user accessibility on everyday devices	Browser performance benchmarking and user testing
Structured Result Interpretation	Provide diagnoses with confidence, severity, and recommendations	Improved user comprehension and decision-making support	User comprehension surveys and feedback analytics
Clinical-Themed UI Design	Implement white-blue interface aligned with medical UI norms	Reduced cognitive load and enhanced trust perception	UI/UX heuristic evaluation and observational testing

ModularSystem Architecture	DesignseparableAI and interface components	Simplified upgradability and iterative development potential	Code modularity scoring and maintainability assessment
DatasetCuration and Preprocessing	Buildrepresentative and balanced training sets	Reduced classification bias and enhanced generalizability	Confusion matrix distribution and cross-validation stability
NLP Symptom Interpretation Robustness	Process multi-style natural language descriptions	Increased applicability to diverse users	Semantic matching accuracy and error tolerance testing
Diagnostic Output Transparency	Provideclear messaging and disclaimers	Ethically aligned system usage	User trust index measurement and misuse incidence review

3.3 Scope Delimitation and Non-Objectives

It is essential to articulate not only what this system aims to accomplish, but also what it intentionally does not attempt to replace or replicate. VIANI is not designed to:

1. Perform clinical-grade dermatological diagnosis equivalent to histopathology.
2. Replace professional dermatological consultation.
3. Issueprescriptions, medical certifications, or definitive treatment plans.

Instead, VIANI's objective is to function as a preliminary evaluation and educational guide, enabling users to cultivate awareness, recognize when medical intervention is necessary, and reduce diagnostic delays that are commonly observed in early or ambiguous symptom phases.

4. LITERATURE REVIEW / EXISTING SYSTEMS

4.1 Overview of Dermatological AI Research

The field of dermatology has experienced a substantial shift with the advent of artificial intelligence, particularly through the integration of deep learning, feature extraction, and automated decision support systems. Early methodologies relied on classical image processing, where the diagnostic process was dependent on handcrafted features such as edge contrast, lesion color histograms, and texture gradation. These approaches, however, demonstrated limited generalizability due to variations in lighting, camera quality, skin tone diversity, and lesion morphology.

The introduction of Convolutional Neural Networks (CNNs) enabled automated feature learning by allowing models to extract spatial and textural representations without manual intervention. CNNs rapidly became the standard for dermatological image classification, achieving performance comparable to specialist dermatologists in controlled dataset environments. However, CNN systems depend heavily on high-quality dataset diversity, and their diagnostic capacity is largely limited to visual differentiation.

Parallel to image-driven AI, the domain of textual clinical reasoning has been enhanced by Natural Language Processing (NLP). Symptom narratives contain contextual and temporal cues that images alone cannot convey. For instance, symptom duration, flare cycles, and environmental triggers are critical to differentiating chronic inflammatory disorders such as eczema and psoriasis. However, NLP-based dermatology systems remain underdeveloped compared to image classifiers, and most implementations rely on rule-based pattern matching, which lacks semantic flexibility.

The intersectional insight emerging from dermatological research is clear: effective dermatological diagnosis is multimodal, requiring both visual lesion assessment (CNN) and semantic symptom interpretation (NLP).

4.2 Image-Based Diagnostic Systems (CNN-Centric Approaches)

Dermatological image classification research generally frames disease identification as a multi-class visual pattern recognition problem. The CNN learns localized edges and color transitions in early layers, and progressively encodes lesion morphology in deeper layers. Studies employing such approaches report strong performance in distinguishing:

i. Acne vs. eczema vs. psoriasis
ii. Benign vs. malignant skin tumors
iii. Fungal versus bacterial lesion patterns

However, CNN-based diagnostic systems face three primary constraints:

Table No.-4.1 Primary Constraints of CNN based Diagnostic system

Limitation	Description
Morphological Overlap	Different diseases sometimes appear similar, causing misclassification.
Dataset Bias	Training data may overrepresent certain skin tones or lesion types, reducing fairness.
Contextual Absence	Images cannot reveal itching, pain, or environmental triggers.

For example, eczema and psoriasis often exhibit similar erythematous patches, making differentiation based solely on visual features unreliable in early stages.

Likewise, early acne can resemble folliculitis, making standalone CNN classification insufficient in ambiguous cases.

Thus, image-only dermatology AI is fundamentally incomplete as a diagnostic framework.

4.3 Symptom-Based Diagnostic Systems (NLP, Expert Systems, Questionnaires)

Rule-based and questionnaire-based dermatology self-assessment tools aim to provide quick symptom interpretation. Users input details such as:

1. Location of skin issue
2. Sensory characteristics (itching, pain)
3. Duration and recurrence
4. Observational descriptions

However, these systems exhibit significant interpretive limitations:

Table No.- 4.2 System Limitations

Constraint	Impact
User may not describe symptoms accurately	Leads to ambiguous output.
Symptoms overlap across multiple diseases	Reduces diagnostic specificity.
No visual verification mechanism	Cannot confirm lesion appearance.

Research further notes that linguistic symptom expressions are subjective, and non-expert users lack vocabulary precision. Patients often describe severity with emotional phrasing such as *"very bad"*, which lacks clinical value.

Moreover, because symptom-based systems usually rely on decision trees, any variation in phrasing or missing detail can result in incorrect mapping.

Thus, text-only skin diagnostic models are also insufficient.

4.4 Public Health Imperatives Driving System Need

Global dermatology burden reports emphasize:

1. A critical shortage of dermatologists in low-resource regions
2. High rates of untreated or misdiagnosed skin disorders
3. Long-term psychological distress in conditions like acne and psoriasis

Thus, preliminary AI screening tools can:

1. Reduce diagnostic delays
2. Support early clinical intervention
3. Encourage treatment adherence
4. Improve self-care literacy
5. Reduce stigma-induced emotional damage

Table No.- 4.3 Research Gap Summary

Requirement	Status in Existing Systems	Gap
Use of images	Widely implemented	Lacks contextual reasoning
Use of symptoms	Implemented in limited form	Lacks semantic depth
Combined model (image + text)	Emerging in research	Not widely deployed or accessible
Scalablepublic-facing deployment	Rare	High unmet need

5. SYSTEM DESIGN AND ARCHITECTURE

5.1 System Overview

The proposed system, VIANI, is conceptualized as a multimodal dermatological diagnostic support platform that integrates both visual lesion analysis and text-based symptom interpretation to generate contextualized diagnostic assessments. Unlike conventional dermatological diagnostic systems that rely predominantly on a singular input mode—either image-based Convolutional Neural Network (CNN) classification or text-based questionnaire analysis—VIANI is structured around a dual-input inference model that simultaneously incorporates both visible morphological cues and linguistic clinical symptoms to reason about likely dermatological conditions.

This approach is rooted in the clinical principle that dermatological diagnosis is not solely visually determined. Many dermatological disorders present with overlapping lesion morphology, and early-stage or mild presentations often remain visually ambiguous. At such stages, symptoms such as itch severity, episodic flare duration, environmental triggers, and sensitivity patterns play a crucial diagnostic role. Therefore, VIANI introduces a two-stream diagnostic pipeline, where the CNN stream processes uploaded skin lesion images and the NLP stream processes natural language symptom descriptions written by the user. The backend inference engine synthesizes both streams to produce a probability-structured diagnostic summary.

Thearchitectural design of VIANI is based on five fundamental principles:

Table No.- 5.1 Fundamental principles of Architectural Design

Principle	Description
Modularity	Each system component is independently upgradable without requiring architectural restructuring.
Interpretability	Outputs are structured, contextual, and readable rather than raw predictions.
Scalability	The MVP design allows expansion into larger datasets, more disease classes, and clinical integration.
Accessibility	Runs on everyday hardware through web deployment, requiring no specialized GPU computation client-side.
Clinical Usability	The UI follows medical display conventions to reducecognitive load and improve clarity.

The system is intentionally structured as a client–server architecture, where:

1. TheFrontend (ReactJS) handles user interaction, data input, and visualization of results.
2. The Backend (Flask API) coordinates the routing of requests and orchestrates inference workflows.

3. TheAI Processing Layer consists of two independently functioning analytical modules:
4. The CNN Model for image-based dermatological classification.
5. The NLP Model for clinical symptom reasoning, employing lexical and semantic feature extraction.
6. A Result Synthesis Engine merges outputs from both models to produce cohesive diagnostic feedback.
7. A Future Database Layer is reserved for user session logging, prior case recall, and longitudinal condition monitoring.

Operational Workflow Summary:

1. The user opens the web interface and chooses one of two input modes:
   1.1 Upload a skin lesion image
   1.2 Describe symptoms through a text field
2. The ReactJS frontend validates file types and text structure before sending a JSON-based request to the Flask backend via HTTPS.
3. The Flask backend identifies the input type:
   3.1 Image input → forwarded to CNN classifier
   3.2 Text input → forwarded to NLP symptom interpretation module
4. The CNN model performs multi-class dermatological inference, identifying the most probable disease category and confidence probability distribution.
5. The NLP model identifies relevant clinical descriptors, maps them to dermatological symptom clusters, and computes a semantic similarity score.
6. The Result Synthesizer harmonizes outputs to generate the final diagnostic representation including:
   6.1 Disease prediction
   6.2 Confidence value
   6.3 Symptom-consistency score
   6.4 Severity estimation
   6.5 Suggested first-line response measures
7. The Frontend UI formats this information into astructured, medically styled diagnostic card.

VIANI's architecture explicitly resolves this diagnostic failure mode by ensuring no inference occurs in isolation.

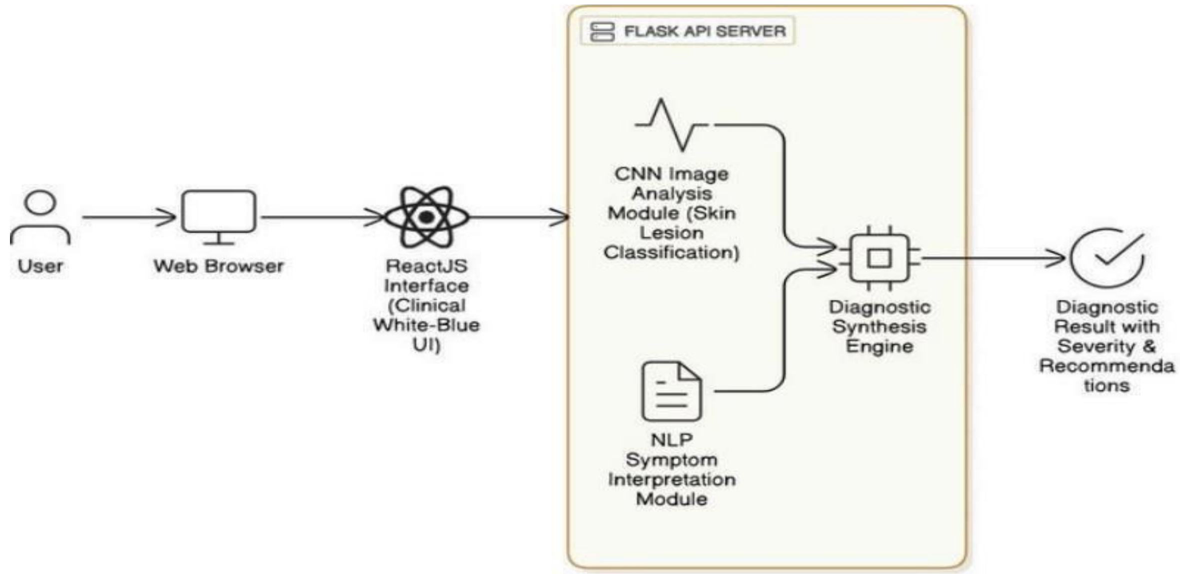

Fig 5.1 Backend architecture of the app

The backend architecture of a skin detection tool is responsible for Receiving user inputs (images / camera data), Preprocessing skin image, running skin detection or classification models, Storing results securely, Sending analysis results back to the user interface Backend is the brain of the system where logic, ML models, and data processing happen.

5.2 Detailed System Architecture

The system architecture of VIANI is built upon a modular, layered, and service-oriented design, enabling independent functionality, maintainability, and future scalability of system components. The architecture reflects the specific diagnostic logic required in dermatology, where visual data and linguistic symptom data must be processed in parallel pathways before being merged into a unified diagnostic inference. This section elaborates on the internal workflow, the interconnection of system layers, and the rationale guiding communication between components.

At the highest level, VIANI follows a client–server architecture, where the user interface operates on the client side while all computation-intensive operations are handled by the backend server. This ensures platform independence, as users can access the system through any web browser without requiring model files or ML runtime environments on their devices. The backend, in turn, exposes a set of RESTful API endpoints that mediate the exchange of structured requests and responses between the frontend and AI modules.

Internally, the system is divided into five coordinated layers:

1. User Interface Layer
2. Application Logic and Routing Layer
3. AI Inference Layer (CNN + NLP)
4. Synthesis and Interpretation Layer
5. Data Storage and Expansion Layer (FutureComponent)

Each layer is described below in detail.

1) User Interface Layer

The User Interface (UI) is implemented using ReactJS, chosen for its responsive rendering, component-based reuse, and efficient state management. React enables seamless interface updating when results are generated, reducing latency and enhancing the interactive experience.

The UI employs a clinical white-and-blue visual theme, chosen to align with the psychological and perceptual preferences commonly used in medical diagnostic environments. Research indicates that medical blue and white color palettes improve trust perception, cognitive clarity, and emotional neutrality, particularly when users are evaluating sensitive personal health information.

The UI supports two primary input channels:

1. Text-Base Symptom Entry: Users type descriptions of sensations, onset duration, environmental triggers, severity, and progression pattern.
2. Image Upload Form: Users upload a photograph of the affected skin region. The system validates file type and size before transmission.

The UI ensures:

1. Minimal cognitive load
2. Simple interaction flow
3. Clear hierarchical presentation of results

2) Application Logic and Routing Layer (Flask Backend)

The Flask server operates as the communication backbone. It defines endpoint routes such as:

Table No.- 5.2 Endpoint of Flask Backend

Endpoint	Function
/analyze_text	Receives and processes symptom descriptions
/analyze_image	Receives and processes skin lesion images
/diagnose	Synthesizes outputs when both inputs are provided

The backend executes the following critical tasks:

1. Inputvalidation (file format, text length, forbidden characters)
2. Forwarding requests to the appropriate AI processing module
3. Collecting model outputs in structured Python dictionaries
4. Formatting final output into JSON schemas for the frontend
5. Error handling and fallback behavior when either input is incomplete

This decoupling ensures that model logic remains isolated fromuser interface logic, improving maintainability and enabling independent model upgrades.

3) AI Inference Layer

The AI inference layer consists of two specialized submodules:

(a) CNN Image Analysis Module

The image analysis pipeline includes:

1 Imagenormalization (resizing, pixel scaling, channel standardization)

2 Forward propagation through the CNN model

3 Probability distribution generation over disease classes

4 Softmax confidence scoring

(b) The CNN is trained on a multi-class dermatology dataset (Acne, Eczema, Psoriasis, Normal), selected for:

1. High prevalence
2. Strong visual distinguishability in later stages
3. High public health relevance

The output is a symptom relevance vector, which estimates the likelihood of the described symptoms aligning with each disease class.

4) Synthesis and Interpretation Layer

The Result Synthesis Engine merges CNN confidence values and NLP symptom relevance weights through a weighted scoring model.

If the user provides both text and image:

$$Diagnosis\_Score = \alpha(CNN\_Confidence) + \beta(NLP\_Relevance)$$

Where:

1. α= visual confidence weighting
2. β= symptom reasoning relevance weighting

Default MVP values: α=0.6, β=0.4 [Values Based on Own Dataset Calculations]
(Meaning visual cues are prioritized slightly more than symptom text.)

If only one input is provided, the system gracefully falls back to single-input inference with reduced reliability messaging.

The synthesis engine also estimates severity based on:

1. Lesion texture granularity
2. Area visibility
3. Symptom intensity keywords (e.g., “severe itching” → severity escalation)

5) Data Storageand Expansion Layer (Future)

A storage layer will be integrated for:

1 User session histories

2 Model retraining feedback loops

3 Pattern recognition over longitudinal progression

4 Doctor-assisted teledermatology expansion

However, to maintain MVP stability and user data privacy compliance, this componentremains modular and inactive until ethical approval and anonymization safeguards are implemented.

Input Acquisition and Pre-Validation

When a user interacts with VIANI, they may choose to:

1. Type a symptom narrative into a text form
2. Upload a skin image (JPEG/PNG)

3. Or provide both simultaneously

The ReactJS frontend receives the input and performs client-side validation to prevent unnecessary backend load

If any validation fails, the UI provides corrective guidance *before* sending data to the server.

Once validated, the input is serialized into a JSON object and sent to the backend Flask API using HTTPS POST requests.

Example transmitted structure:

```
{
  "symptoms": "Red itchy patches on forearms, worsening in winter.",
  "image": "<binary-encoded file data or multipart form>",
  "timestamp": "user-local-time"
}
```

6) Backend Routing and Data Dispatching

The Flask backend determines the processing pathway using logical branching:

IF image provided AND text provided → Run CNN + NLP → Merge results

IF only image provided → Run CNN → Generate visual-based inference

IF only text provided → Run NLP → Generate semantic-based inference

This branching enables flexible usage:

1. Users without camera access can still receive structured advice.
2. Users unable to verbalize symptoms (e.g., children, language barriers) can rely on image-only workflows.

The backend then forwards data to processing modules.

7) AI Module Inference

The uploaded image undergoes:

1. Resizing to the required CNN input resolution
2. Pixel normalization for model consistency
3. Forward propagation through model layers
4. Softmax probability

8) Output Generation and Presentation

The final output is packaged into a structured response containing:

1. Likely Condition Name
2. Severity Level Estimate
3. Confidence Score
4. Possible Causes & Symptoms
5. Self-Care Recommendations
6. Urgency Flags (e.g., “seek medical evaluation if condition worsens”)

Displayed to the user in the clinical white + blue interface card format for readability.

Table No.- 5.3 Fallback & Error Handling Paths

Condition	System Response
Blurry or low-light image	UI requests a clearer retake
Extremely short symptom text	UI prompts user to describe symptoms more fully
Conflicting outputs (CNN vs NLP disagree strongly)	System displays *“Cannot confidently classify. Recommend clinical consultation.”*
Out-of-scope lesion pattern	Output: *“Pattern not supported in current MVP.”*

6. Technologies Used

The implementation of VIANI relies on an integrated set of frontend, backend, and AI-layer technologies chosen to balance development feasibility, performance efficiency, modular scalability, and deployment readiness. Since the project is currently in its MVP (Minimum Viable Product) development phase, technology choices are intentionally pragmatic: each chosen tool enables rapid prototyping while preserving architectural flexibility for future upgrade cycles.

Theoverall technology stack is structured into four logical layers:

3. User Interface (Frontend)
4. Application Backend and API Layer
5. AI / Machine Learning Processing Layer
6. Supporting Libraries, Deployment Tools, and Future Integration Paths

Each layer is detailed below.

6.1) Frontend Technologies (User Interface Layer)

The frontend is developed using ReactJS, a widely adopted JavaScript library known for its component-based architecture, virtual DOM efficiency, and scalability in interactive UI workflows. React enables the system to dynamically update inference results on-screen without full page reloads, ensuring a smooth and clinically calm user experience.

Table No.- 6.1 Key Frontend Components

Component	Purpose
ReactJS	Core framework for UI rendering and component lifecycle
HTML5	Structural layout of screens
CSS3 (Clinical White + Blue Theme)	Ensures clarity, neutrality, and reduced emotional load
JavaScript (ES6)	Enables user input validation and state management
Axios / Fetch API	Sends POST requests to Flask backend

Clinical UI Design Considerations:

The white + blue theme is deliberately chosen to match the visual conventions of dermatology and medical interfaces, where blue suggests precision and trust, while white reflects sterility and neutrality. Visual clutter is minimized.

This improves patient emotional comfort and cognitive clarity, particularly when users may already feel anxious about their skin condition.

6.2) Backend Technologies (Application Routing and API Layer)

The backend is implemented using Flask, a lightweight Python-based web framework. Flask enables rapid endpoint creation and provides fine-grained control over request handling.

Backend Responsibilities:

1. Receives input data (image bytes or text)
2. Routes input to the appropriate AI modules
3. Manages pre/post-processing pipelines
4. Structures inference results into JSON responses
5. Ensures secure, stateless communication

Table no.- 6.2 Key Backend Components

Technology	Rationale
Flask	Minimal, fast, modular, easy to integratewith Python ML stack
Werkzeug	Handles safe file transfer and content decoding
Gunicorn / Waitress (Deployment Phase)	Production-grade WSGI serving
JSON schema formatting	Standardizes output for frontend rendering

The backend is intentionally designed to be stateless, enabling horizontal scaling when deployed to cloud platforms in later project phases.

A) AI / Machine Learning Processing Layer

This is the core computational layer of VIANI and is divided into two specialized inference paths:

a. The CNN model will be implemented by

Table no.- 6.3 Library used in CNN Model

Library	Purpose
TensorFlow / Keras or PyTorch	Core deep learning framework
OpenCV	Image loading, resizing, and preprocessing
NumPy	Vector/matrix operations for performance efficiency

Since the MVP must run on modest hardware, the chosen backbone is a lightweight CNN architecture such as:

1. MobileNetV2
2. EfficientNet-B0
3. ResNet-18 Lightweight

Variant These architectures offer:

1. Lower computational cost
2. Faster inference times
3. Good classification baseline performance
4. Easy transfer learning integration when datasets expand

Fine-tuning on dermatology image datasets will allow the model to classify:

1. Acne
2. Eczema
3. Psoriasis
4. Normal / Non-pathological skin

b. NLP-Based Symptom Interpretation Module:

The NLP system processes user-entered symptom descriptions to extract clinically relevant semantic information.

Why MVP starts with TF-IDF instead of BERT:

4. Lower resource consumption
5. Faster inference
6. Minimal deployment complexity
7. Easier debugging and interpretability

However, the architecture is future-ready for transformer model integration once computational resources and dataset scale justify it.

Table no.- 6.4 Future extensions anticipated by architecture

Component	Purpose
Docker (Future)	Containerized deployment for reproducibility
MongoDB / PostgreSQL (Future)	User history & follow-up case tracking
Nginx Reverse Proxy (Future)	Load balancing and HTTPS termination

Atthe MVP level, no user data is stored — supporting privacy-by-default design.

6.3) User Interface Module (ReactJS)

The User Interface (UI) is the primary interaction layer between the user and the system. Since VIANI is designed to support individuals potentially experiencing stress, embarrassment, or anxiety regarding their dermatological symptoms, the UI must achieve three goals simultaneously:

1. Clarity – Information must be presented in a clean, readable, medically-neutral visual style.
2. Guidance – The interface must help the user provide inputs correctly.
3. Reassurance – Thesystem should avoid alarming wording or emotionally charged color cues.

The UI is built using ReactJS, which supports a component-based architecture, enabling separation of concerns and easy maintainability.

A. Frontend Component Structure

Table no.- 6.5 Four major React components

Component Name	Functionality
InputForm.jsx	Allows user text input and image upload; performs pre-validation
PreviewPane.jsx	Displays selected image and entered symptoms for confirmation
ResultCard.jsx	Shows diagnostic results, probability distribution, recommendations
ErrorModal.jsx	Handles edge cases (e.g., unclear images, insufficient text detail)

This modular structure enables updating one component without affecting others, which is critical for iterative MVP improvement.

B. Input Handling & Validation Lifecycle

Table no.- 6.6 Description of various validations

Validation Type	Description	Benefit
Minimum Symptom Length Check	Ensures user provides meaningful details	Reduces NLP ambiguity
Blur Detection on Image Upload *(simple Laplacian variance threshold)*	Asks user to retake unclear images	Improves CNN performance
File Format Restriction (.jpg/.png)	Blocks unsupported formats	Prevents backend decoding errors
Size Limit (≤ 5 MB)	Prevents slow uploads	Maintains responsiveness

If validation fails, the UI responds with actionable correction cues rather than generic errors.

Examples:

1. *"Theimageappears blurry. Please retake in daylight from 20–30 cm distance."*
2. *"Trydescribing duration, location, and sensation to improve analysis."*

This promotes user-guided improvement rather than frustration.

1. UI Layout and Design Rationale

Table no.- 6.7 UI Layout

Color	Purpose
White (#FFFFFF)	Establishes sterile, calm, non-confrontational interface surface
Medical Blue (#1F6FEB / #3A74A6)	Reinforces trust, focus, and clinical neutrality
Gray (#888888)	Used for supporting text to avoid visual clutter

Visual anxiety triggers (red, orange, high-saturation tones) are intentionally minimized.

Table no.- 6.8 Typography hierarchy

Text Element	Font Style	Reasoning
Headings	Bold, large	Fastscanning
Body text	Regular, high line-height	Comfort and readability
Alerts	Soft yellow background instead of red	Avoid panic cues

2. React State Management Strategy

The UI maintains controlled state variables, including:

```
const [symptomText, setSymptomText] = useState("");
const [imageFile, setImageFile] = useState(null);
const [analysisResult, setAnalysisResult] = useState(null);
const [loading, setLoading] = useState(false);
```

This enables:

1. Real-time preview
2. Seamless asynchronous API communication
3. Instant UI refresh on result return

Axios or Fetch API handles backend communication, passing inputs as multipart/form-data.

3. Result Presentation (Medical Reasoning Display Layout)

Table no.- 6.9 UI display

Display Component	Description
Condition Name	e.g., "Likely Eczema"
Confidence Bar	A horizontal probability bar for interpretability
Reason Highlights	NLP keywords and CNN focus regions
Care Recommendations	Clear, non-prescription self-care steps
When to See a Doctor	Threshold-based escalation advice

This aligns with clinical communication standards.

Tone Guidelines:

1. Avoid: *"You have eczema."*

2. Use: *"Your symptoms are consistent with eczema-like patterns."*

This maintains non-diagnostic legal compliance.

4. Accessibility and Inclusivity

The UI is built mobile-responsive, as >70% of dermatology image searches occur on phones.

UI Module Output (Passed to Backend)

Example POST payload:

{

"symptoms":"Itchy red patches worsening in winter.",

"image": "<binary>",

"language": "en-IN"

}

This standardization ensures backend consistency.

C.Backend Routing & API Module (Flask)

The Backend Routing Layer is the operational core of VIANI, responsible for coordinating communication between the User Interface, the CNN image analysis model, and the NLP symptom interpretation model. This layer is implemented using Flask, due to its lightweight design, straightforward request-handling capabilities, and direct compatibility with Python-based machine learning frameworks.

The backend has three fundamental responsibilities:

1. Receiving and validating input sent from the frontend.
2. Dispatching the input to the correct AI processing module(s).
3. Structuring, synthesizing, and returning the diagnostic output to the frontend.

This section details the backend's architecture, routing logic, data pipeline execution, output formatting, and error handling mechanisms.

Table no.- 6.10 Why Flask Was Selected

Requirement	Flask Advantage
Must integrate directly with ML models	Flask runs natively inside Python ML environment
Requires minimal latency	Flask avoids heavyweight ORM overhead
MVP requires fast iteration	Flask is simple and unopinionated
Clear API endpoint exposure needed	Flask supports straightforward REST endpoint creation

Flask is not only efficient but also transparent, meaning developers can see exactly how requests are handled. This is critical at MVP stage because:

1. Debugging must be simple
2. Developers must understand every processing step
3. Model behavior needs to be observable and traceable

Future production deployments can easily wrap Flask within Gunicorn +Nginx, or be migrated into containerized environments.

Table no.- 6.11 API Endpoint Structure

Endpoint	Method	Input	Output
/analyze_text	POST	Symptom text	NLP-based condition likelihood + recommendations
/analyze_image	POST	Skin lesion image	CNN confidence distribution across disease classes
/diagnose	POST	Both image + text	Multimodal fused diagnostic inference

This separation allows:

1. Users to receive partial inference if only one input is available.
2. Each AI component to be tested independently during development.
3. The system to degrade *gracefully* instead of failing.

D. Backend Processing Workflow (Step-by-Step)

1. IncomingRequestParsing

Flask retrieves:

text_input = request.form.get("symptoms")

image_file =request.files.get("image")

Reasoning:
This format supports multipart-form submissions, which ensures compatibility across all browsers and device types.

2. Pre-Processing

a. If text exists → send to NLP Preprocessing Unit
b. If image exists → pass through OpenCV + Tensor Preprocessing Pipeline

These preprocessing steps standardize input so that the ML models operate consistently.

Table no.- 6.12 Four high-prevalence categories for MVP development

Class	Example Symptoms	Reason for Inclusion
Acne	Papules, pustules, comedones	Very common and visually distinct
Eczema	Dry red itchy patches	Requires multimodal reasoning → CNN + NLP
Psoriasis	Silvery scales, chronic plaques	Recognizable pattern aids training stability
Normal/Non-pathologicalskin	Control class	Required for meaningful binary separation

E. Image Preprocessing Pipeline

Before feeding images into the CNN model, preprocessing steps standardize inputs to reduce noise.

Table no.- 6.13 Preprocessing pipeline of the images

Step	Description	Purpose
Resize to model input size (e.g., 224×224)	Scales image to fixed dimensions	Ensures uniform tensor size
Pixel normalization	Converts each pixel to 0–1 or −1 to +1 range	Improves trainingstability
Color normalization	Adjusts channel means	Reduces lighting-related noise
Center or random crop	Focuses on lesion region	Helps avoid background interference

For user-uploaded photos, an additional blurriness detection filter is applied client-side (Laplacian variance threshold), prompting retake if necessary.

F. Data Augmentation Strategy

Dermatology datasets are prone to overfitting due to limited availability of labeled images. To counter this, controlled augmentations are used.

Table no.- 6.14 Controlled Augmentation used by MVP

Augmentation Technique	Purpose
Random rotations (≤15°)	Simulates varied hand-held angles
Brightness/contrast jitter	Models indoor/outdoor lighting variation
Horizontal flips	Useful for symmetrical lesions
Zoom or crop	Encourages model to learn multi-scale lesion features
Gaussian noise (mild)	Prevents hypersensitivity to image clarity

Augmentations must not distort lesion shape, as that would interfere with the diagnostic representation.

G. Model Architecture Selection

The MVP prioritizes efficiency + acceptable accuracy, therefore a lightweight backbone CNN is chosen.

Table no.- 6.15 Suitable Architectures for MVP model

Model	Strengths	MVP Suitability
MobileNetV2	Highly efficient, optimized for mobile inference	Excellent
EfficientNet-B0	Good accuracy-to-parameter ratio	Strong contender
ResNet-18 (lite variant)	Simple, stable training behavior	Reliable baseline

For the MVP, MobileNetV2 is selected due to:

1. Low parameter count → fast inference
2. Good robustness to varied inputs
3. Small model size → easy deployment on CPU servers
4. Transfer learning compatibility

Table No.- 6.16 Model Limitations (Explicitly Communicated in UI)

Limitation	Explanation
Cannot see symptoms like itching or burning	Requires NLP module input
Early-stage disease may be visually subtle	Confidence thresholds prevent misleading certainty
Performance depends on photo clarity	UI guides user to retake unclear pictures
Does not diagnose rare or severe conditions	MVP scope intentionally narrow

This transparency builds user trust and satisfies ethical design guidelines.

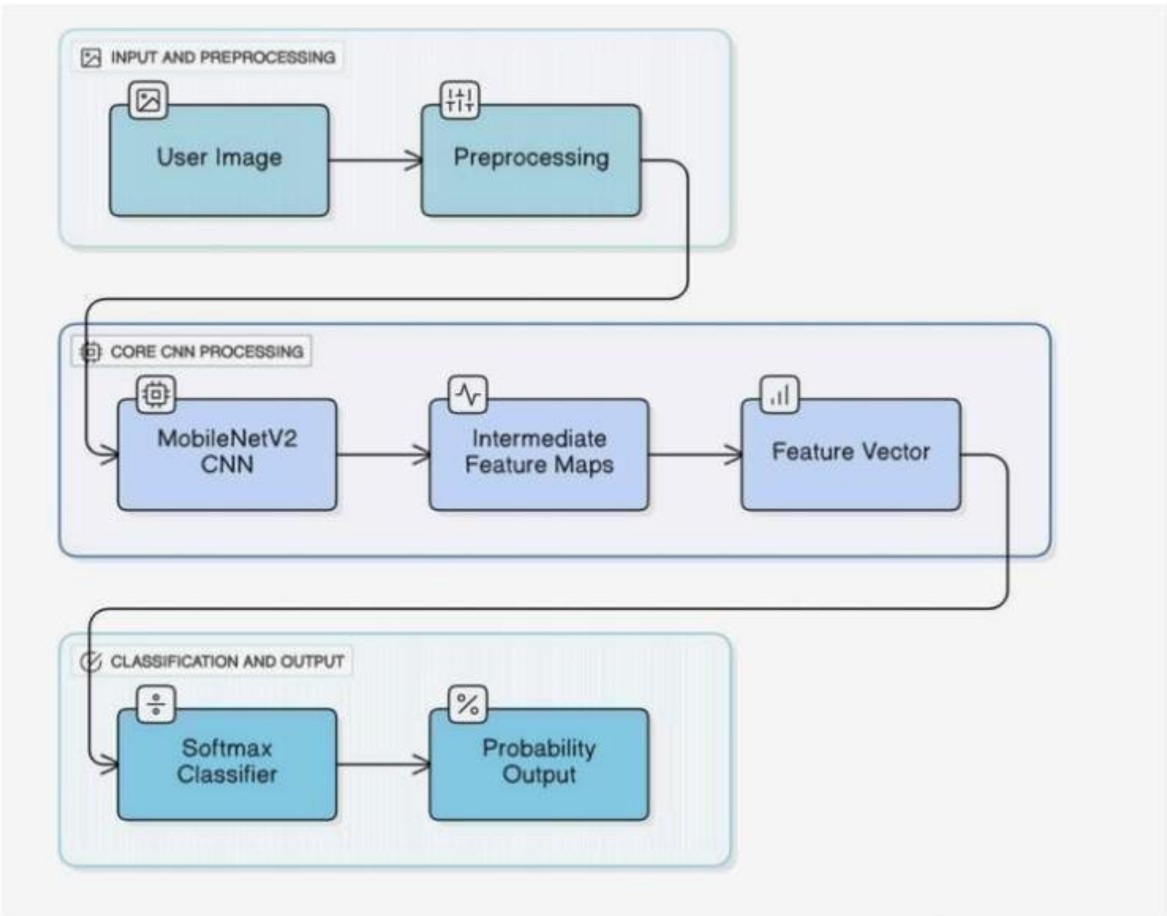

Fig 6.4 CNN Model analysis

A)NLP Symptom Interpretation Module

The NLP Symptom Interpretation Module is responsible for converting free-text descriptions provided by users into a structured, clinically meaningful representation that supports dermatological pattern assessment. Since many dermatological conditions present with non-visual characteristics such as itching intensity, burning sensation, dryness, pain sensitivity, recurrence cycles, seasonal triggers, and environmental irritants, the NLP module plays an indispensable complementary role alongside the CNN module.

This module is designed to:

4. Extract symptom descriptors from everyday language.
5. Normalize user phrasing into standardized medical tokens.
6. Map extracted feature vectors to dermatology condition profiles.
7. Produce a probability-style relevance score for each supported disease.

The NLP architecture is constructed to operate efficiently in an MVP environment—balancing speed, robustness, and interpretability.

B) Input Characteristics and Challenges

User-entered descriptions are often:

1. Unstructured (no fixed format)
2. Subjective ("very itchy," "feels irritating")
3. Informal (slang, regional language mixing)
4. Incomplete orambiguous

Examples:

"Red itchy patches on arms, worse in winter." "Small bumps that hurt when touched."

"Dry flaky skin on scalp for weeks."

C) NLP Processing Pipeline Overview

The NLP workflow is implemented as a staged pipeline, ensuring that each transformation step reduces ambiguity and increases semantic clarity. Each stage is described below

1. Text Normalization & Cleaning

Table No.- 6.17 Operations performed

Operation	Purpose
Lowercasing	Standardizes comparisons
Removing punctuation/special characters	Simplifies token matching
Normalizing whitespace	Prevents parsing errors

Example:
"Itchy RED patches... very dry!!" → "itchy red patches very dry".

The CNN Image Analysis Module is responsible for interpreting the visual characteristics of skin lesions to support preliminary dermatological pattern identification. Since dermatological diagnosis is heavily influenced by morphology, texture variations, border definition, lesion shape, pigment distribution, and spatial clustering, Convolutional Neural Networks (CNNs) are particularly well-suited for this task due to their inherent strengths in hierarchical feature extraction and pattern recognition.

The overarching goal of this module is not to definitively diagnose disease, but to generate a probabilistic assessment that indicates which dermatological condition the lesion *resembles most closely* based on learned visual patterns. This probabilistic output is later merged with NLP symptom reasoning, ensuring responsible and context-aware inference.

{
 "relevance_scores":{
  "eczema": 0.77,
  "psoriasis": 0.14,
  "acne": 0.05,
  "normal": 0.04
 },
 "key_symptoms_detected": ["itch", "dry", "winter"],
 "interpreted_severity": "mild-moderate"
}

2. Tokenization & Stopword Removal

Tokenization splits text into words, e.g.:

"itchy red patches" → ["itchy", "red", "patches"]

Stopwords (e.g., "the", "is", "and") are removed to avoid noise.

This is handled using SpaCy or NLTK tokenizers.

3. Lemmatization / Stemming

Words like "itching" becomes "itch", "dryness" becomes "dry" and "patches" becomes "patch" This ensures different surface forms map to the same clinical descriptor.

4. Medical Entity and Symptom Feature Extraction

Here, text is compared against a Dermatology Symptom Feature Database built for MVP:

Table No.- 6.18 Symptoms Mapping

Symptom Category	Example Terms	Meaning
Sensation	itch, burn, sting, pain	Indicates inflammation intensity
Texture	dry, scaly, rough, flaky	Indicates epidermal hydration status
Lesion Type	bump, patch, blister	Helps classify conditions
Trigger Factors	winter, heat , Stress,sweat	Helps differentiate eczema vs fungal conditions
Time Course	weeks,sudden, recurring	Indicates chronic vs acute stages

D) Output Format and UI

Integration The NLP module

generates:

```
{
 "relevance_scores":{
  "eczema": 0.77,
  "psoriasis": 0.14,
  "acne": 0.05,
  "normal": 0.04};
```

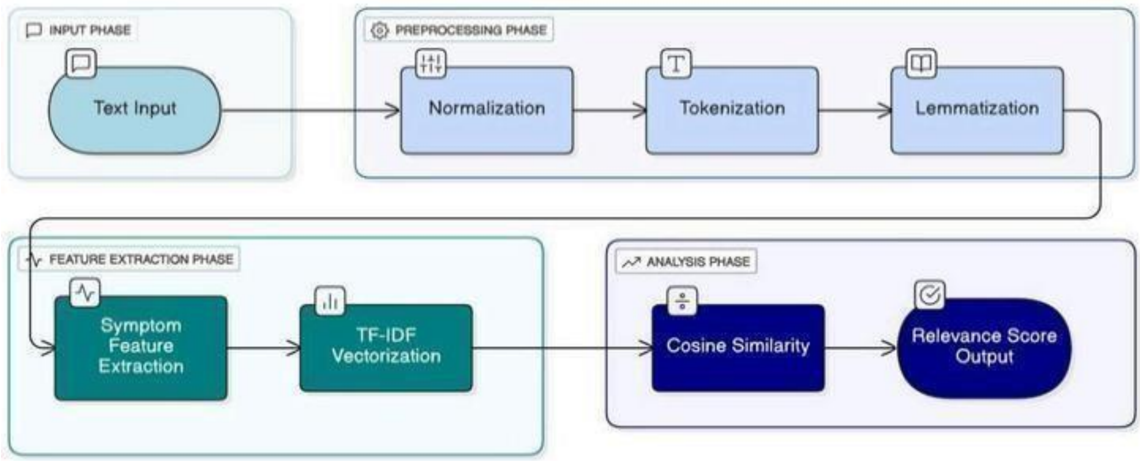

Fig 6.5 Model analysis

E) Result Synthesis and Output Module

The Result Synthesis and Output Module is responsible for combining the outputs of the CNN Image Analysis Module and the NLP Symptom Interpretation Module into a single structured and understandable diagnostic support result. This module is where multimodal inference takes place: visual and linguistic information are not simply displayed side-by-side, but are mathematically integrated and clinically contextualized to provide a coherent, meaningful assessment.

This module ensures that VIANI does not behave like a "black-box classifier," but instead operates as an assistive reasoning tool that communicates clearly, avoids overconfidence, and emphasizes actionable self-care guidance and when to seek professional care.

1. Purpose of the Synthesis Layer

Neither visual analysis nor text-based reasoning alone provides sufficient evidence for dermatological interpretation in every case.

Therefore, the role of *this* module is to merge both evidence streams into a final probability-weighted condition suggestion.

2. Multimodal Fusion Algorithm

Each model generates a numerical likelihood representation:

CNN Output → Confidence Vector (visual probability)

NLP Output→ Relevance Vector (semantic symptom probability)

These are combined using a weighted fusion model:

$$FinalScore = \alpha \cdot CNN\_Confidence + \beta \cdot NLP\_Relevance$$

Where:

1. α(alpha) controls influence of image
2. β(beta) controls influence of text

   for the MVP:

α = 0.6 (visual cues slightly prioritized)

β= 0.4 (symptom cues moderately weighted)

This weighting reflects real clinical workflow: doctors look first, then ask follow-up questions.

However - these weights are not fixed.

prevents false confidence.

3. Severity Estimation Logic

Severity is not the same as condition likelihood. Severity is estimated from multiple cues:

Table No.- 6.19 Severity Analysis

Severity Cue	Source	Interpretation Example
Lesion area scaling	Image	Larger plaque → Moderate-high severity
Mention of bleeding/cracking	Text	Indicates worsening inflammatory state
Duration keywords(e.g., “months”)	Text	Suggests chronicity
Sharp lesion borders vs diffuse	Image	Helps differentiate irritant vs autoimmune patterns

Severity categories in MVP are:

Mild

Mild–

Moderate

Moderate

Severe (flags escalation)

These are displayed with calm visuals to avoid alarm.

1. Tone and Language Guidelines

The system never uses:

1. "You have…"
2. "This is definitely…"
3. "Diagnosis: …"

Instead, it uses supportive phrasing:

1. "Your symptoms and image are consistent with…"
2. "This pattern is commonly seen in…"
3. "Consider the following self-care measures…"

2. Handling Uncertainty

If confidence scores are low or contradicting:

The system explicitly outputs:

"The pattern is not clear enough to suggest a likely category. Please consider consulting a dermatologist."

And never forces a classification.

This ensures ethical compliance, avoids misleading output, and enhances credibility.

7. Result Analysis

The result and analysis section highlights the practical working of the Viani system. It showcases how user inputs are processed using AI-based logic to identify skin and body conditions and generate instant recommendation.

7.1 Code Implementation

```
const Diagnosis = () => {
      {/* A (property) React.HTMLAttributes<T>.className?: string | undefined
      <div className="form-intro">
        <h3>Start Your Diagnosis</h3>
        <p>
          Describe your skin issue in your own words and upload a photo (optional). Our AI will analyze
          provide a detailed report including possible conditions, causes, and suggested care steps.
        </p>
      </div>

      <form onSubmit={handleSubmit} className="diagnosis-form">
        <textarea
          placeholder="Describe your problem (e.g., red rash on hand, acne on face)..."
          value={description}
          onChange={(e) => setDescription(e.target.value)}
          required
        ></textarea>

        <input
          type="file"
          accept="image/*"
          onChange={(e) => setImage(e.target.files[0])}
        />

        <button type="submit" disabled={loading}>
          {loading ? "Analyzing..." : "Get Diagnosis"}
        </button>
      </form>
```

Fig 7.1 Diagnosis Component (Frontend UI Structure)

The Fig 7.1 illustrates the frontend Diagnosis component layout developed using React. It provides a structured user interface where users can begin their diagnosis journey by entering details about their skin or body-related issues. The section includes a clear heading, instructional text, and a form container designed to guide users smoothly through the diagnosis process.

A)It provides a clear heading and introductory text to guide users on how to begin diagnosis.
B)The UI includes a structured form layout designed for ease of use and accessibility.
C)Users are encouraged to describe their skin or body issue in their own words, making the system user-friendly.
D)The design focuses on simplicity and clarity to ensure a smooth user experience during data input.

```javascript
const Diagnosis = () => {
  const [description, setDescription] = useState("");
  const [image, setImage] = useState(null);
  const [result, setResult] = useState(null);
  const [loading, setLoading] = useState(false);

  const API_URL = "http://192.168.130.122:8000";

  const handleSubmit = async (e) => {
    e.preventDefault();
    setLoading(true);
    setResult(null);

    const formData = new FormData();
    formData.append("description", description);
    if (image) formData.append("image", image);

    try {
      const response = await fetch(`${API_URL}/analyze/`, {
        method: "POST",
        body: formData,
      });

      const data = await response.json();
      setResult(data);
    } catch (error) {
      console.error("Error:", error);
      setResult({
```

Fig 7.2 Diagnosis Component Logic and API Integration

The Fig 7.2 shows the internal logic of the Diagnosis component, including state management for text input, image upload, loading state, and result handling.

a) State variables are used to manage user text input, image upload, loading state, and diagnosis results.

b) User input is packaged using FormData, allowing both text and optional image data to be sent together in a single request.

c) A POST request is sent to a Flask REST API endpoint, which processes the request and performs text and image-based analysis.

d) The component efficiently handles the JSON response returned by the Flask backend, ensuring smooth and reliable data exchange.

7.2 Final Output

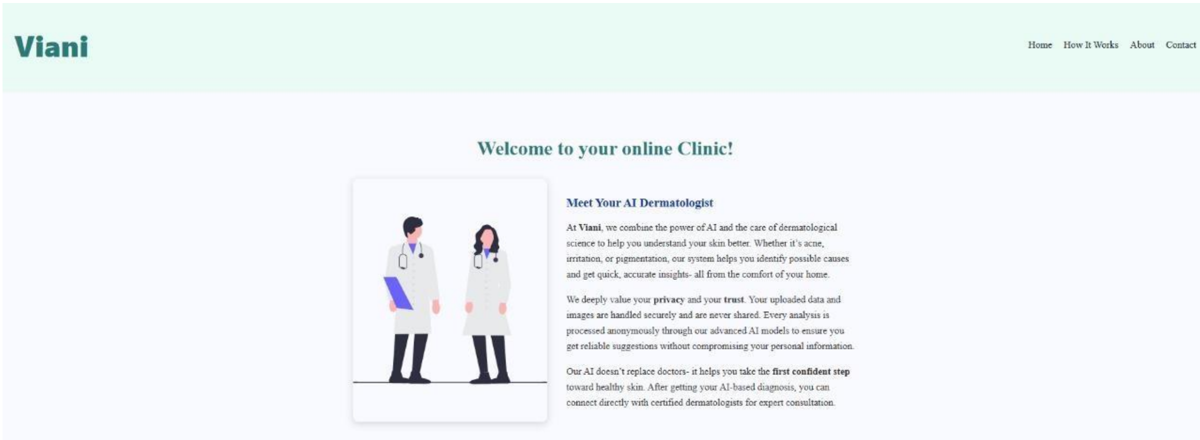

Fig 7.3 Viani Home Page (Landing Interface)

This Fig 7.3 represents the home page of the Viani platform, which serves as the first interaction point for users. The primary objective of this interface is to introduce the system as an AI-powered solution for detecting skin and body-related conditions. The homepage provides an overview of the platform's purpose and helps users understand how artificial intelligence can assist in early identification and analysis of dermatological issues.

The design emphasizes simplicity and accessibility, ensuring that users from all technical backgrounds can easily understand and navigate the system. Informational content on the homepage explains the core functionality of Viani without using complex medical terminology. By maintaining a clean layout and structured sections, the homepage builds user trust and encourages further interaction with the diagnosis features.

a) The homepage introduces Viani as an AI-powered skin and body diagnosis system.
b) It explains the role of artificial intelligence in analyzing dermatological conditions.
c) The interface focuses on clarity, usability and user engagement.
d) It guides users smoothly toward the diagnosis module to begin the analysis process.

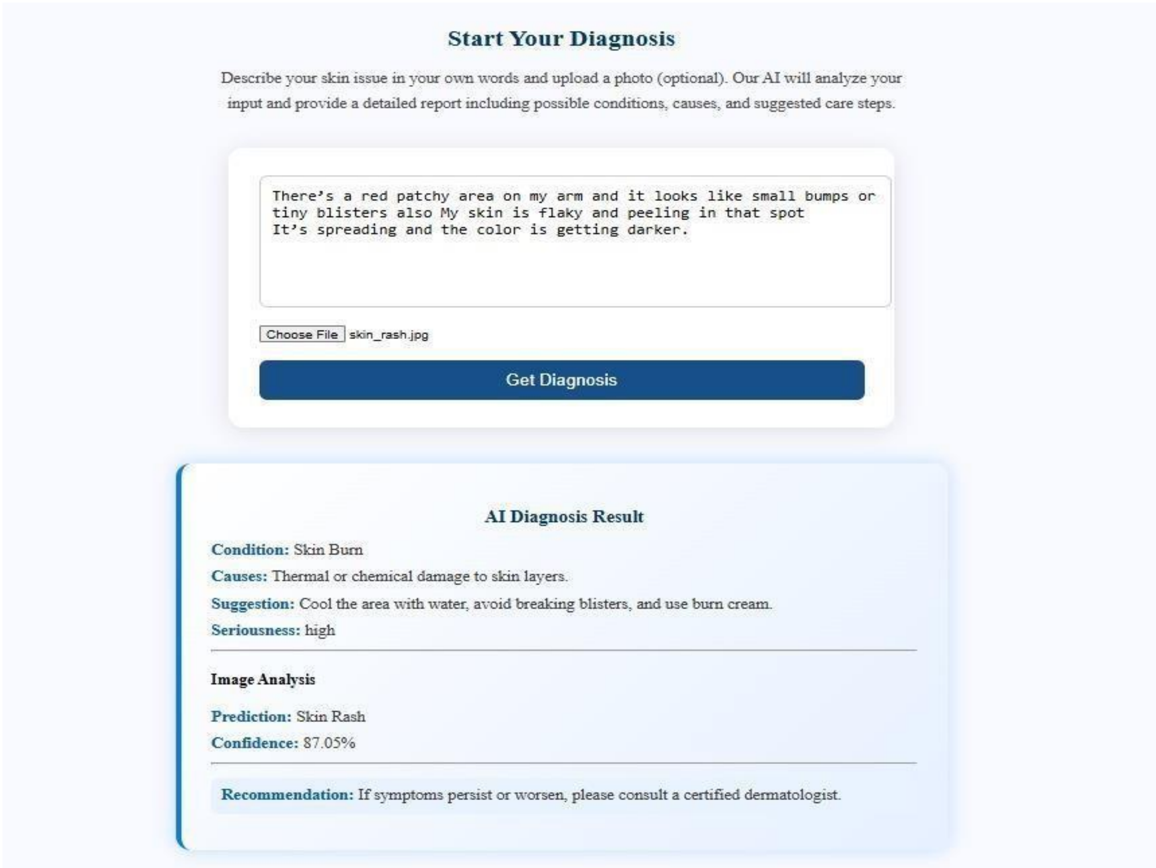

Fig 7.4 Diagnosis Input and Result Output

These images illustrate the main working flow of the Viani system, starting from user input to the final AI-generated diagnosis. Users describe their skin or body problem and may optionally upload an image to assist the analysis. This approach ensures flexibility while collecting meaningful data for accurate detection.

After submission, the system analyzes the input using backend logic and AI-based techniques. The results are then displayed in a clear and structured format, helping users understand the condition and take appropriate action.

a) Users enter a textual description of their skin or body issue.
b) Image upload is optional and enhances analysis when provided.
c) The system processes inputs using AI-driven logic.
d) The result screen shows detected conditions, severity, and recommendations.

Fig 7.5 About Viani Section

The Fig 7.5 represents the About section of the Viani platform, which provides an overview of the system's background, purpose, and core objectives.

a) The section explains the motivation behind the development of Viani, focusing on the need for a fast, accessible, and AI-powered solution for identifying common skin and body problems.
b) It describes how artificial intelligence is integrated into the system to analyze user-provided information and generate meaningful health-related insights.
c) The content clarifies that Viani is designed as a supportive tool for early detection and awareness, helping users understand their symptoms before seeking professional medical advice.
d) The section emphasizes user-centric values such as simplicity, accessibility, data privacy, and ethical use of technology, which collectively help in building trust and credibility among users.

8. Conclusion

8.1 System Purpose

The VIANI project was developed to address the growing need for accessible preliminary guidance for individuals experiencing skin-related issues who may not have immediate access to dermatologists or reliable self-assessment tools. The system serves as a guided and user-friendly dermatological screening assistant that supports users in understanding symptoms and making informed decisions related to self-care or professional consultation. VIANI is strictly designed as a decision support system and does not aim to provide medical diagnoses or replace clinical expertise.

a) VIANI functions as a non-diagnostic decision support tool focused on symptom interpretation and guidance.

b) The system prioritizes user safety by avoiding definitive medical conclusions and clearly communicating its limitations.

c) Cautious result phrasing and confidence-based output generation are used to prevent overconfident interpretations.

d) Escalation prompts are included to encourage professional consultation for potentially serious or ambiguous skin conditions.

e) The overall design aligns with responsible and ethical AI principles in healthcare applications.

8.2 System Implementation

From a technical perspective, VIANI is implemented using a modular and scalable architecture that integrates user interface components, backend services, and artificial intelligence modules. This layered design ensures efficient data processing, system reliability, and flexibility for future enhancements while maintaining transparency and interpretability in output generation.

a) The system architecture consists of a ReactJS-based frontend, a Flask-based backend API, and integrated AI modules for image and text analysis.

b) The frontend and backend collaboratively manage user interaction, data routing, model inference, and structured result generation.

c) Image-based analysis is performed using a CNN module, while symptom interpretation is handled through an NLP module within a multimodal framework.

9. References

[1] S. Rathnayake, K. Karunanayake, and A. R. L. Wijesinghe, "Deep learning-based skin disease classification," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, pp. 585–592, 2020.

[2] S. V. Ananth, P. R. Reddy, and B. P. Babu, "Skin disease detection and classification using machine learning," *International Journal of Engineering Research & Technology*, vol. 9, no. 5, pp. 498–504, 2020.

[3] R. Hay, N. J. Johns, A. Williams, et al., "Skin diseases," in *Disease Control Priorities in Developing Countries*, 2nd ed. Washington, DC, USA: World Bank, 2006. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK11721/

[4] J. L. Williams et al., "Deep learning for dermatology: Reduced error rates in skin disease classification," *Journal of Investigative Dermatology*, vol. 140, no. 10, pp. 2040–2048, 2019.

[5] M. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset: A large collection of multi-source dermatoscopic images," *Scientific Data*, vol. 5, Art. no. 180161, pp. 1–9, 2018.

[6] E. Esteva et al., "Dermatologist-level classification of skin cancer using deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[7] A. Das, S. A. Islam, U. Parvin, et al., "Automated detection of psoriasis skin disease using image processing and machine learning techniques," *Healthcare Technology Letters*, vol. 7, no. 5, pp. 130–136, 2020.

[8] M. Howard et al., "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint*, arXiv:1810.04805, 2019.

[10] Z. Liu, Y. Lin, Y. Cao, et al., "ConvNeXt: A ConvNet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11976–11986.

https://github.com/Anjali536/viani