

Team Biased



SkinLens

Your Personal Dermatologist

AGENDA

01

Roles and
Responsibilities
Improvements

02

Problem Statement
Project Description
Working Agreement

03

Personas
MVP
Project Design

04

Technologies
Algorithms
Diagrams

05

Project backlog
Sprint 1 Summary
Sprint 2 Summary

06

Sprint 3:

- Stories
- Test cases
- Completed
- Not Completed

07

Sprint 3 Metrics:

- Velocity
- Burndown Chart
- Committed Ratio

08

Sprint 3 Retrospective
Project Demo and APIs

Roles and Responsibilities



MOHAMMAD ZAID

TEAM LEADER/DEVELOPER



ZHUOWEN YAN

SCRUM MASTER/DEVELOPER



NAGANJALI PUJITHA KALLI

DBA/DEVELOPER

Roles and Responsibilities



MOHAMMED TANZIL
DEVELOPER/TESTER



BANGLING YIN
DEVELOPER



ABHIKUMAR PATEL
DEVELOPER/TESTER

IMPROVEMENTS

- Historical Velocity Chart (slide 72) and Committed Ratio (slide 74)
- Removed a technical story of creating user stories from Product Backlog. (slide 53)



WHAT PROBLEM ARE WE SOLVING?

Accurate and timely diagnosis of skin diseases is a challenge because:

- Limited availability of dermatologists in remote areas
- Traditional methods primarily involve only visual inspection of the skin which may be prone to human error or variability in interpretation.

What does it affect?

This results in delayed diagnosis, limited personal care and possibly worsening of skin diseases.

PROJECT DESCRIPTION

- A web-based platform to help patients diagnose skin conditions using deep learning.
- Intended for patients who wants to quickly access their skin conditions, skinLens integrates image analysis to provide accurate diagnosis of skin diseases and recommend treatments.
- Intended for dermatologists who wants to efficiently access patient's skin conditions.
- Unlike traditional visual inspections, SkinLens uses advanced image-based deep learning model and offers accurate predictions and recommend treatments along with real-time consultation with dermatologists.
- Benefits - Improved diagnose accuracy, faster and personalized care, increased accessibility, reduced costs and real-time access to consultations with dermatologists.

PROJECT DESCRIPTION

Project Name:	SkinLens
Team:	Team Biased
Project Description:	<p>For patients who wants to quickly access their skin conditions, for dermatologists who wants to quickly access patient's skin conditions</p> <p>SkinLens is a web application that integrates image analysis to provide accurate diagnosis of skin conditions and recommend treatments along with real-time consultation with dermatologists.</p> <p>Unlike traditional visual inspections, SkinLens uses advanced image-based deep learning model and offers accurate predictions and recommend treatments.</p>
Project Details	
Benefit Outcomes:	Improved diagnose accuracy, Faster and personalized care, Increased accessibility, Reduced costs, Real Time consultation with Dermatologist.
GitHub Link:	https://github.com/htmw/2024F-Biased/wiki

Team Working Agreement

Communication expectations

- All communications will be through Slack.
- Opinions and ideas of all team members will be respected.
- Changes, if any, should be discussed and agreed within the team.
- Conflicts should be focused on issues and not individuals.
(Strictly avoid blame game)
- Unresolved conflicts among the team should be reported to the team leader.
- We encourage everyone to express their frustrations and discomfort early.
- Clear understanding and listening to others are the most important things to resolve conflicts.

Meeting structure and frequency

- Meetings will be held via zoom
- Daily scrum at 8pm for 15 minutes.
- A follow-up meeting should be planned after the daily scrum if there are any blockers that can't be addressed quickly.
- A meeting for Sprint Planning will be planned at the beginning of the Sprint.
- A meeting for Sprint Retrospective will be planned before the Sprint end date.
- Encourage everyone to actively participate in the meeting.
- We will adhere to agreed meeting time and will inform the team of any absences.
- Any changes in the timings will be mutually decided.
- Meeting minutes with objectives and decisions will be recorded in a Google Sheet.
- If you miss a meeting, support decisions made in your absence.
- Scrum Master should host the meeting and control time.

Team Working Agreement

Team norms and values

- Each team member understands their roles and responsibilities.
- We divide tasks transparently and help each other when necessary.
- Each task should have a clear due date and mutually accepted definition of “Done”.
- Each team member takes responsibility for their tasks and for achieving the team's goals.
- Team members should report obstacles immediately if they cannot complete their work on time, and actively contribute to the solution.
- Team members should update their task's process via Jira before the daily scrum meeting.
- Team members should upload their work to prescribed places like GitHub or Google Drive before the daily scrum meeting.

Continuous Improvement

- Encourage teams to research, learn, come up with new ideas and share with the team.
- We will reflect on areas of improvement during each Sprint Retrospective so that our process is better for the next sprint.

Definition of DONE

- Work has been fully reviewed by another team members.
- Code can run and no errors.
- All tests are successfully completed.
- No critical defects.
- Work meets the acceptance criteria.
- Code pushed to GitHub.
- Documentation has been updated.

Project Schedule

Sprint 0

- Sep 6 - Sep 26

Sprint 1

- Sep 26 - Oct 24
- Home page
- Upload image
- AI prediction

Sprint 2

- Oct 24 - Nov 21
- More conditions
- More accuracy
- Disease description
- Treatment recommendations
- Generate report
- Users account
- Technical paper
- Nov 21 - Dec 12
- More accuracy
- Patient records
- Dermatologist dashboard
- Dermatologist review
- Chat
- User Manual
- Deployment Manual
- API Manual

Sprint 3

PATIENT PERSONA

Nashley is a busy professional person, living in a city and has sensitive skin prone to rash and acne.

Challenges:

- She has skin issues and often experiences delays in scheduling appointments with dermatologists.
- Finds very difficult to describe disease accurately and fears misdiagnosis due to lack of knowledge.
- Concerned about long-term skin damage and wants fast treatment options.

Goals:

- She wants a reliable, easy-to-use platform to self-access skin issues quickly and get initial feedback without needing to wait for appointments.
- Hopes to receive personalized treatment recommendations based on her skin type, age, and medical history.



Nashley Pagiroli

Age: 30

Gender: Female

Occupation: IT Employee

PARENT PERSONA

Gloria is a housewife balancing taking care of four children, household duties, and her husband, leaving her with little time to focus on individual healths.

Challenges:

- Frequently deals with minor skin issues like rashes or eczema among her kids.
- Finds it gets confused especially when it comes to skincare for different ages and skin types.
- Concerned about the cost and time required for multiple doctor visits, particularly for non-serious conditions.

Goals:

- Seeks personalized advice for each child's skincare with their unique needs like skin type, allergies, or past treatments.
- Hopes to save time by using images and she uploads.
- Wants peace of mind by receiving accurate, clear advice and thinks homecare is sufficient.



Gloria Garson

Age : 40

Gender : Female

Occupation : House wife

DOCTOR PERSONA

Dr. Robert is working in both a hospital and private clinic, treating a high volume of patients with various skin conditions.

Challenges:

- Manages a large patient load daily, leading to limited time for each consultation and difficulty providing personalized care.
- Often receives incomplete descriptions of diseases, making it challenging to diagnose accurately without further testing or follow-ups.
- Struggles to track patient treatment outcomes over time, like eczema or psoriasis, due to irregular follow-ups or inconsistent data from patients.

Goals:

- Wants a platform of patient information, including images, and medical history, allowing for quick.
- Hopes to tracks patient progress over time, helping to adjust treatments.
- Aims to reduce the patient times.



Dr. Robert Smith

Age : 35

Gender : Male

Occupation : Dermatologist

MVP (Minimum Viable Product)

SkinLens MVP Features

AI Diagnosis

Condition Descriptions

Recommended Treatments

Reports

Real-Time Messaging

Chat History and Notifications

Manage Cases

Dermatologist Review

MVP (Minimum Viable Product)

Web-Based Interface:

- User-friendly platform accessible to patients and dermatologists.

Patient Image Upload:

- Secure and intuitive feature for patients to upload skin condition images.
- Supports multiple images per case for comprehensive analysis.

AI-Powered Diagnosis:

- Utilizes a pre-trained image classification model to process uploaded images.
- Provides fast and accurate diagnosis results for 20+ skin conditions.

Diagnosis Display:

- Clear and concise results displayed to the patient with an option to generate report, including:
 - Predicted skin condition.
 - Brief description of the diagnosis.
 - Actionable treatment recommendations.

Reports

- Generate PDF reports for cases.
- Downloadable and secure report access.

Chat

- Real-time messaging with dermatologists.
- Manage patient chats seamlessly.

Project Design (Figma)

SkinLens

Upload Appointment Chat Info

Sign Up

Take Your Skin Test Now

SkinLens uses AI to analyze skin conditions, detect issues, and provide personalized recommendations.

Start Your Test

How it works

- 1 Upload Image
- 2 AI Analysis
- 3 Get Results

Take a clear photo of your skin concern and securely upload it.

Our advance AI analyze your image to identity potential skin conditions.

Receive a detailed report with insights and recommendations.

Project Design (Figma)

SkinLens

Upload Records Info

Upload Your Skin Image



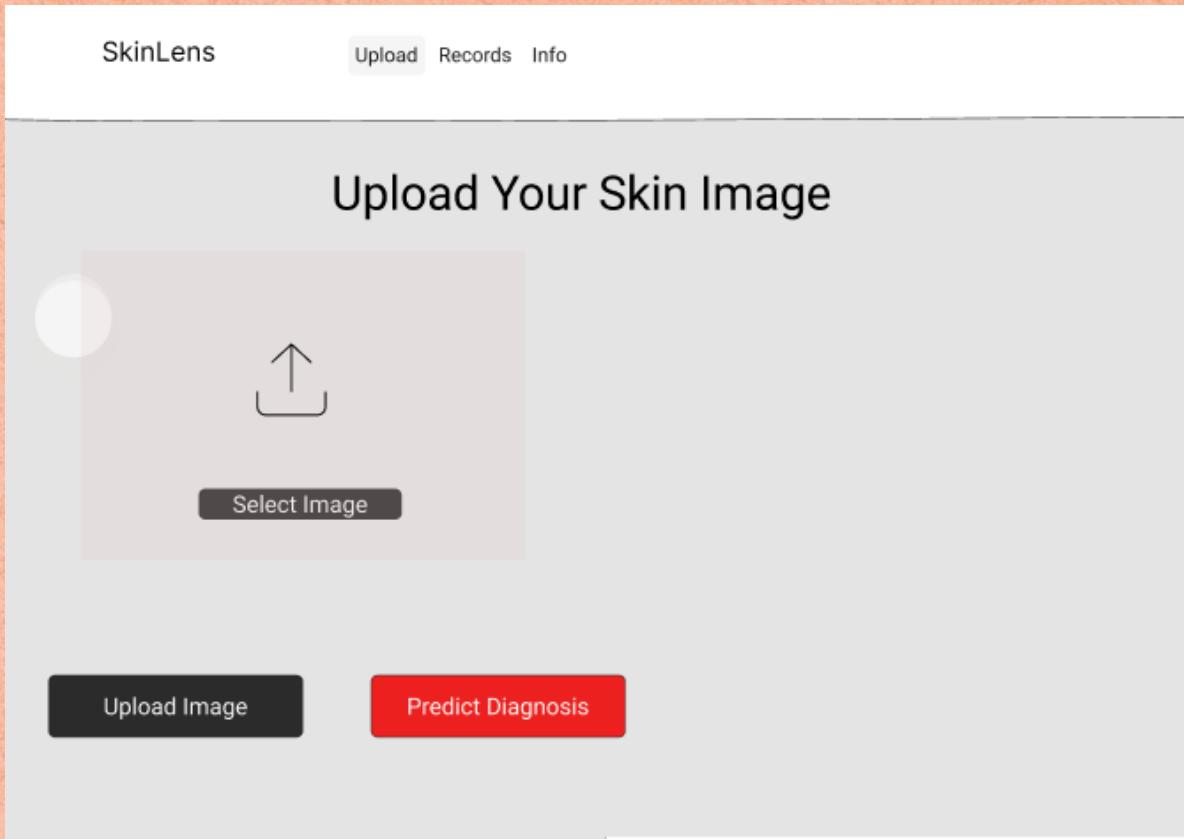
Select Image

Upload Image

Upload Instructions

- Ensure the image is clear and well-lit
- Focus on the specific skin area you want analyzed
- Remove any accessories or clothing that might obstruct the view
- Accepted file format: JPEG, PNG
- Maximum file size: 10MB

Project Design (Figma)



Technologies

Front-end & Back-end & Database



AI Model



Google Cloud

Tools



REACT+VITE AND TAILWIND CSS

REACT: It is an open source JavaScript library.

It used to create interactive and reactive user interfaces. We can split UI's into reusable components, which can make it easier to maintain and build large web applications.

VITE: It is a fast and good web application build tool. It stands out for its fast startup and instant compilation times during development.

TAILWIND CSS: It is an open source css framework.

It encourages a more direct approach to styling.



FLASK

SkinLens utilizes a flask-backend server.

1. Integration with AI Model

- Flask is used exclusively to host and manage the TensorFlow-based image classification model.
- Processes image data sent from the frontend and returns accurate diagnoses for over 20 skin conditions.
- Acts as a middleware to handle communication between the React frontend and the AI model.

2. REST API for Predictions

- Provides a dedicated API endpoint for uploading images and receiving AI-generated diagnoses.
- Handles data preprocessing before sending it to the model and ensures smooth response delivery.

3. Lightweight and Efficient

- Flask's lightweight framework ensures quick deployment and efficient handling of AI model requests.
- Keeps the architecture modular by separating AI-related operations from other backend functionalities.



FIREBASE INTEGRATION IN SKINLENS

5. Real-Time Features

- i. Provides real-time updates for patient case statuses and dermatologist comments.
- ii. Supports real-time chat between patients and dermatologists for instant consultation.
- iii. Ensures real-time notification alerts for case updates and new messages.

Why Firebase?

- Simplifies backend management with an all-in-one platform for authentication, storage, and database operations.
- Provides robust security measures for healthcare data while meeting privacy standards.
- Offers seamless real-time data updates for enhanced user experience.
- Ensures high reliability, scalability, and minimal maintenance effort, making it ideal for healthcare applications.

CORS

Cross Origin Resource Sharing

Security Feature: Prevents web pages from making requests to a different domain than the one serving the web page.

Handling Cross-Origin Requests: Needed to enable requests from the frontend (React) to the backend (Flask), especially when hosted on different servers or ports.

Flask-CORS: A Flask extension used to handle CORS settings, allowing the frontend to securely access the API.

Configuration: Configured in the Flask app to allow requests from the frontend domain, specifying allowed HTTP methods (GET, POST, etc.)

Axios

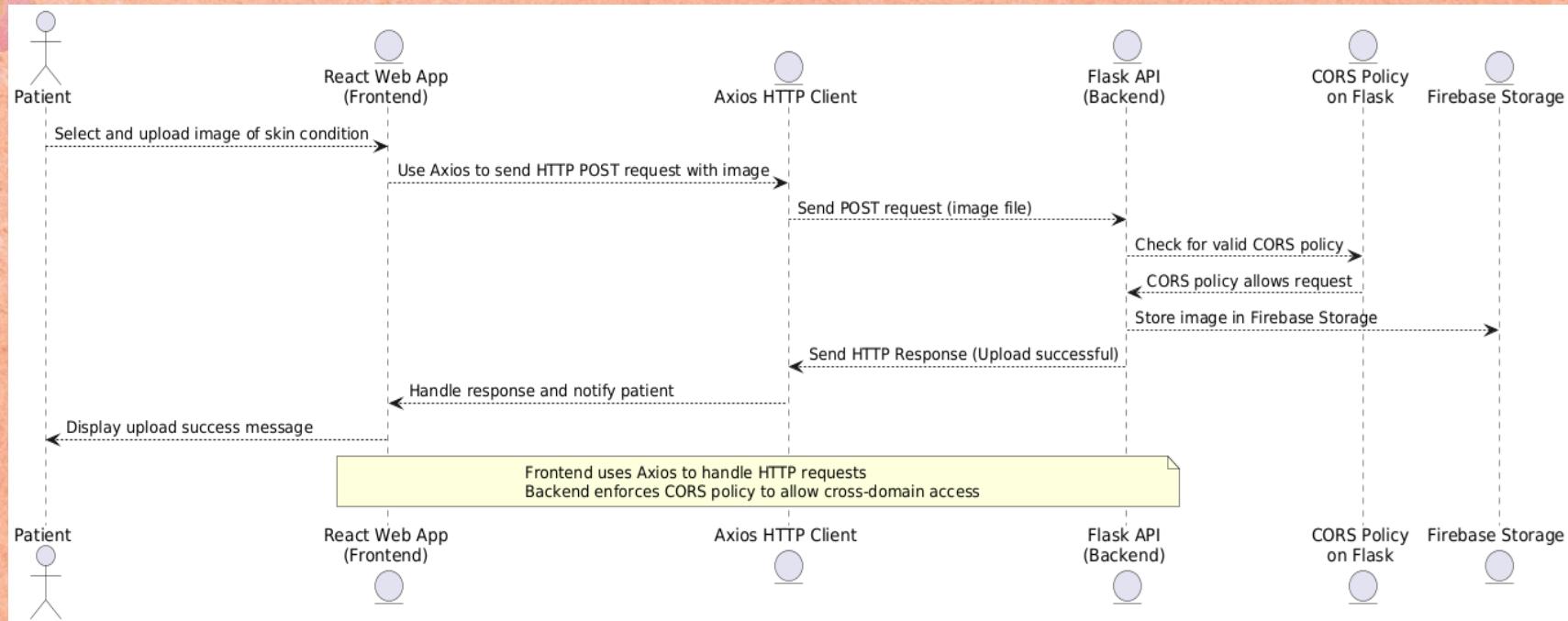
JavaScript HTTP Client: Axios is a promise-based library used to make HTTP requests from the frontend (React).

Simplifies API Calls: Handles API requests (GET, POST) and responses for operations like uploading images and retrieving diagnoses.

Error Handling: Provides built-in support for handling errors and network issues in the API requests.

Asynchronous Operations: Supports asynchronous requests, allowing the app to remain responsive while waiting for the backend to process data.

CORS and Axios functioning



FIREBASE INTEGRATION IN SKINLENS

Key Features of Firebase in SkinLens:

1. Authentication

- Enables secure login and signup for patients and dermatologists.
- Role-based authentication controls access to specific dashboards, cases, and reports.
- Ensures seamless user management with email/password and token-based authentication.

2. Firestore Database

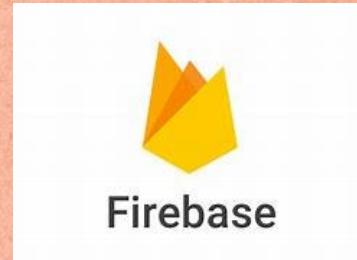
- Stores user profiles, including name, email, role, and unique user IDs (UIDs).
- Maintains detailed case records, including metadata, AI predictions, descriptions, and treatment recommendations.
- Tracks case statuses (e.g., open, reviewed) and supports real-time data synchronization for accurate updates.

3. Firebase Storage

- Securely stores patient-uploaded skin condition images.
- Organizes files by unique case IDs, ensuring easy access and retrieval for analysis and report generation.

4. Security & Scalability

- Leverages Firebase's built-in security rules to protect sensitive patient and dermatologist data.
- Automatically scales to handle high traffic, large datasets, and concurrent user sessions without performance issues.



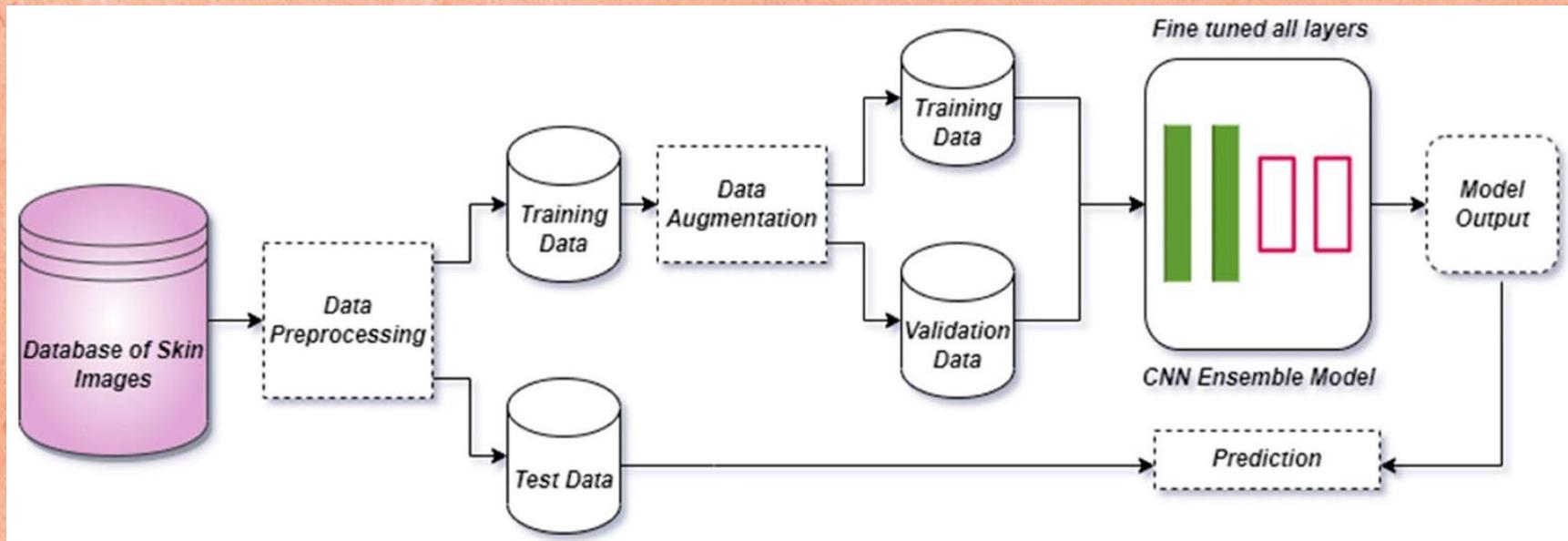
ALGORITHMS:

Used TensorFlow and keras for the CNN Model and Data Augmentation.

OpenCv library for image preprocessing.

Matplotlib for visualization.

System Architecture



CNN Architecture:

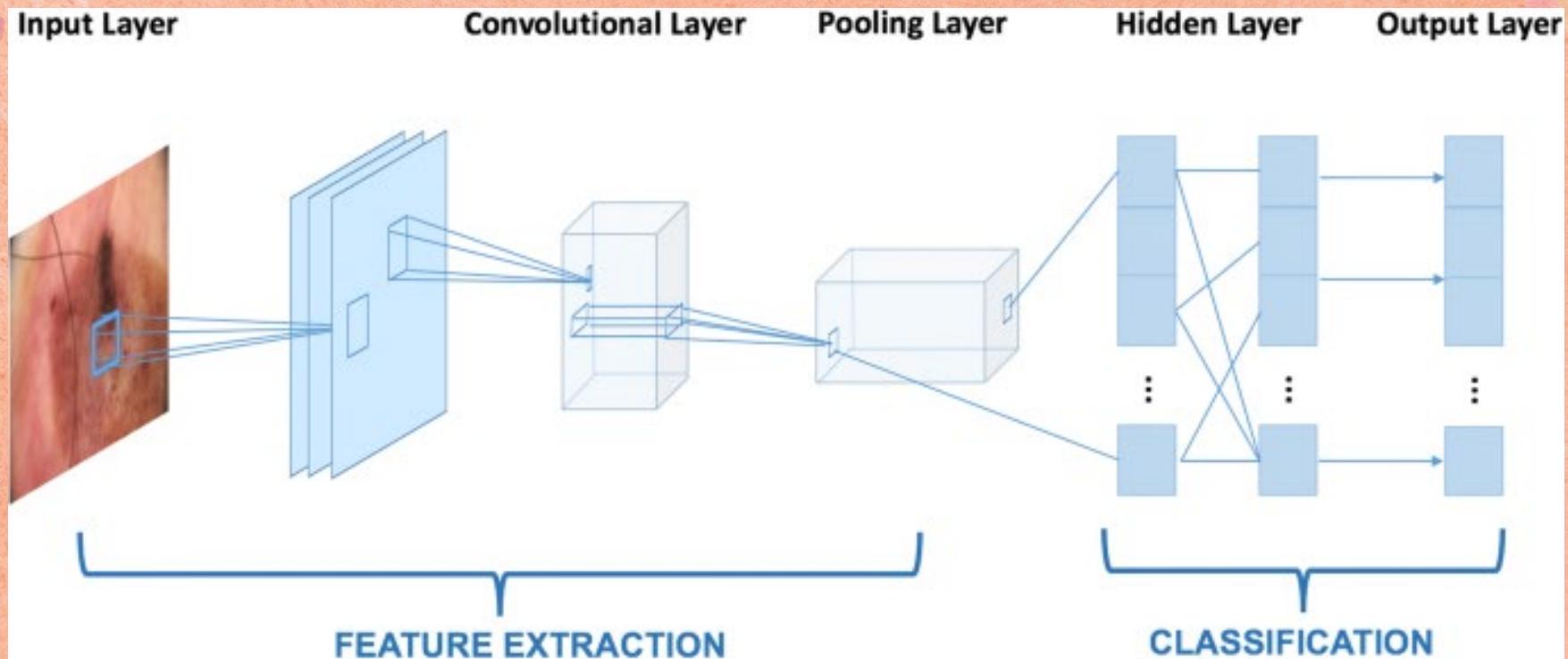
Model Layers:

- **Convolutional Layers:** These layers extract features like edges, textures, and shapes from the images.
- **Max Pooling:** These layers reduce the spatial dimensions of the feature maps, making the model more efficient and reducing overfitting.
- **Dropout:** Randomly sets some neurons to zero during training to prevent overfitting.
- **Flatten Layer:** Converts the 3D feature maps into a 1D vector, preparing it for the fully connected layers.
- **Dense Layer:** The final output layer that classifies the images into one of the 5 classes using softmax activation.
- **Batch Normalization:** Normalizes the output from the convolutional layer to stabilize and accelerate training.
- **Softmax Activation :**The softmax function converts the raw output scores into probabilities. It ensures that the sum of the output probabilities for all classes equals 1, making it suitable for multi-class classification.

Key Parameters:

- **Optimizer:** Adam.
- **Loss Function:** Categorical Cross-entropy (for multi-class classification).

Model Structure



Model Testing:



Choose Files

WhatsApp I....51 PM.jpeg

- WhatsApp Image 2024-11-20 at 9.30.51 PM.jpeg(image/jpeg) - 57624 bytes, last modified: 11/20/2024 - 100% done

Saving WhatsApp Image 2024-11-20 at 9.30.51 PM.jpeg to WhatsApp Image 2024-11-20 at 9.30.51 PM.jpeg

Uploaded file: WhatsApp Image 2024-11-20 at 9.30.51 PM.jpeg

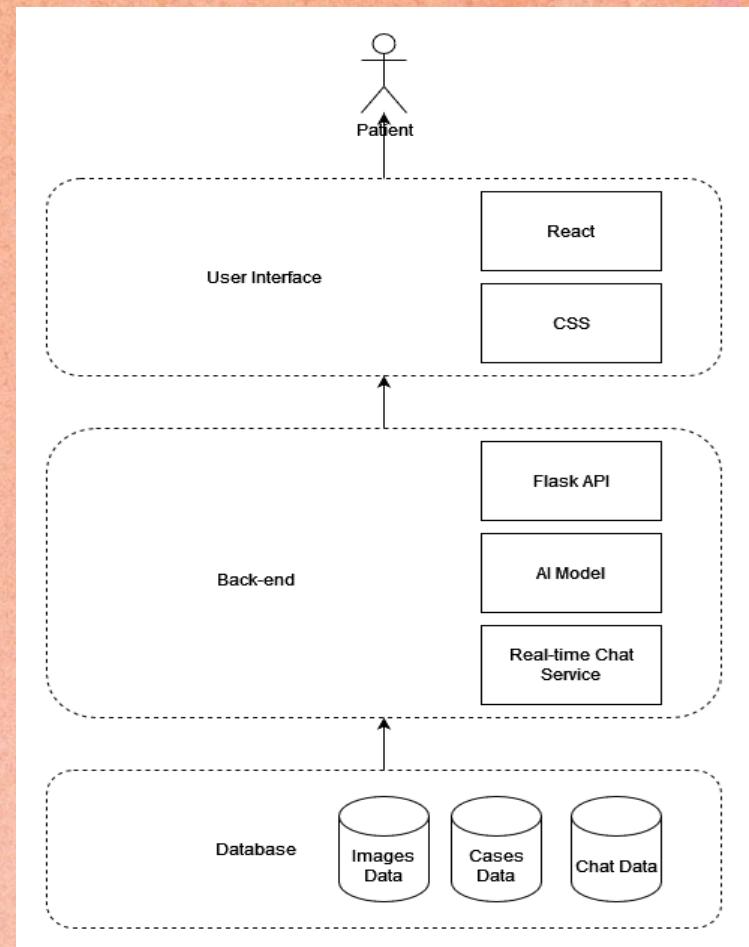
1/1 ————— 0s 110ms/step

Predicted Class Index: 10

Predicted Disease Name: Melanoma

Diagrams

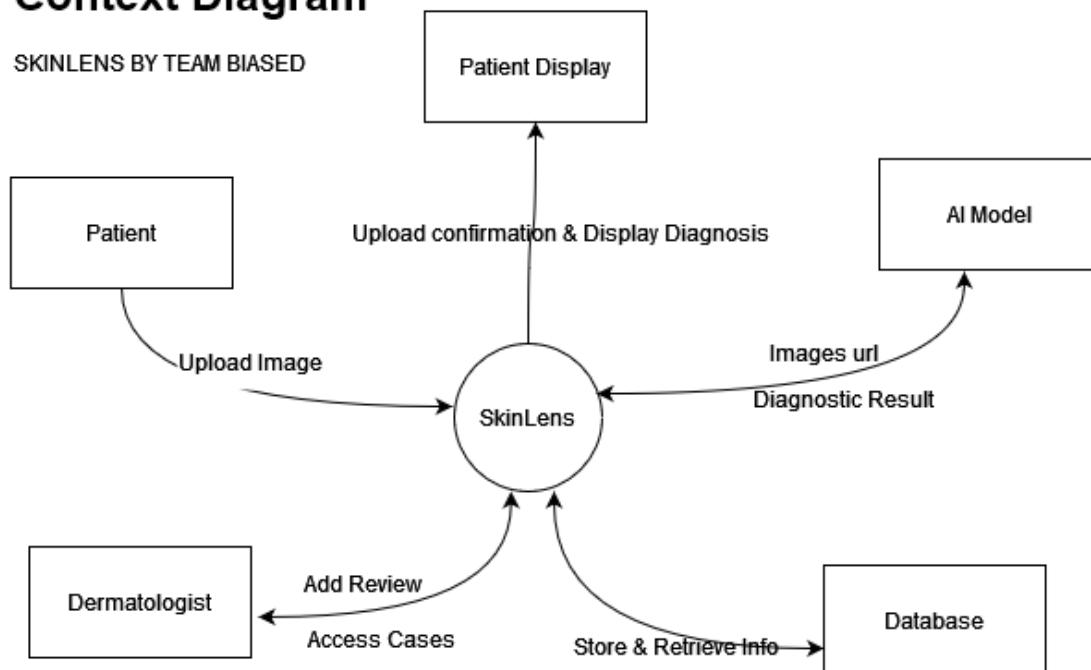
Architecture Diagram



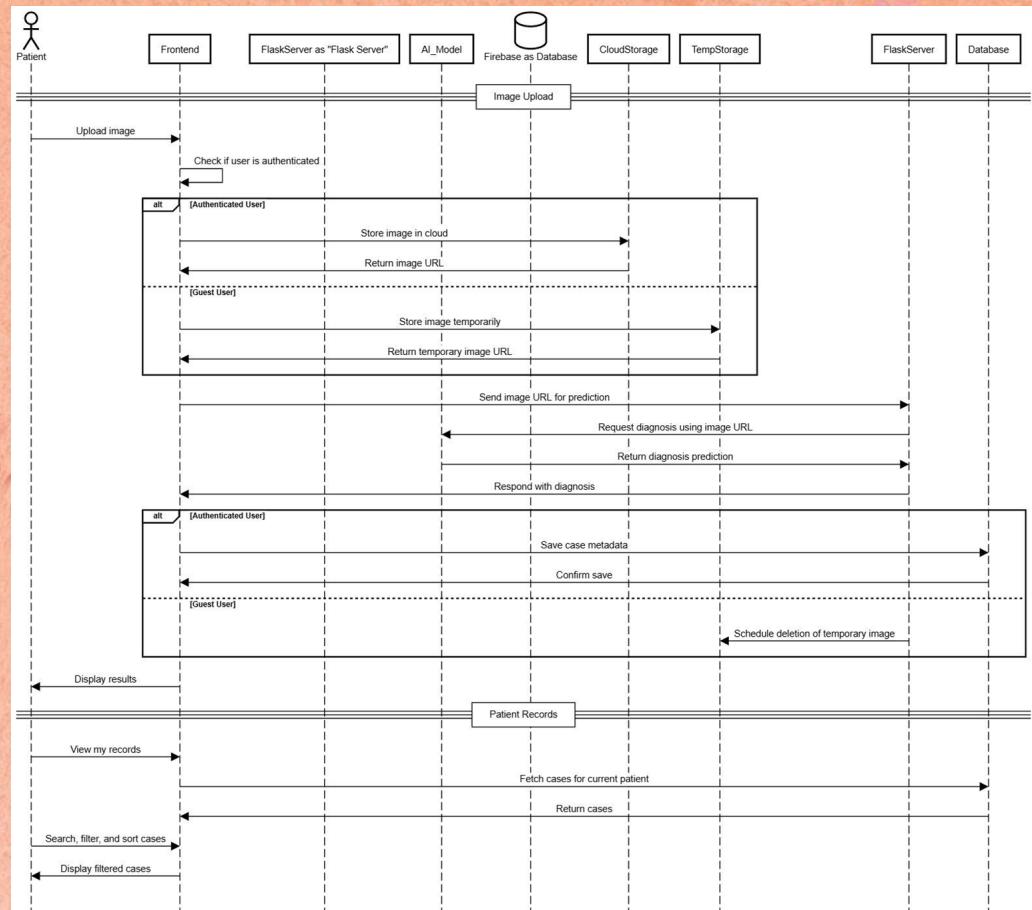
Context Diagram

Context Diagram

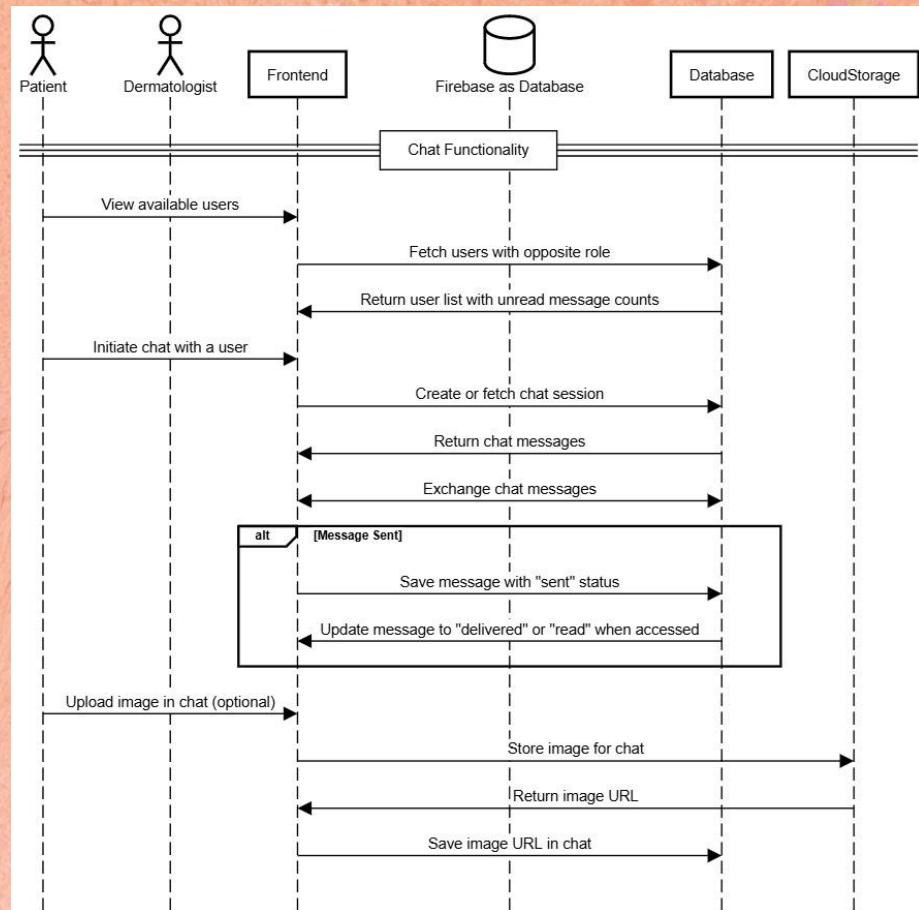
SKINLENS BY TEAM BIASED



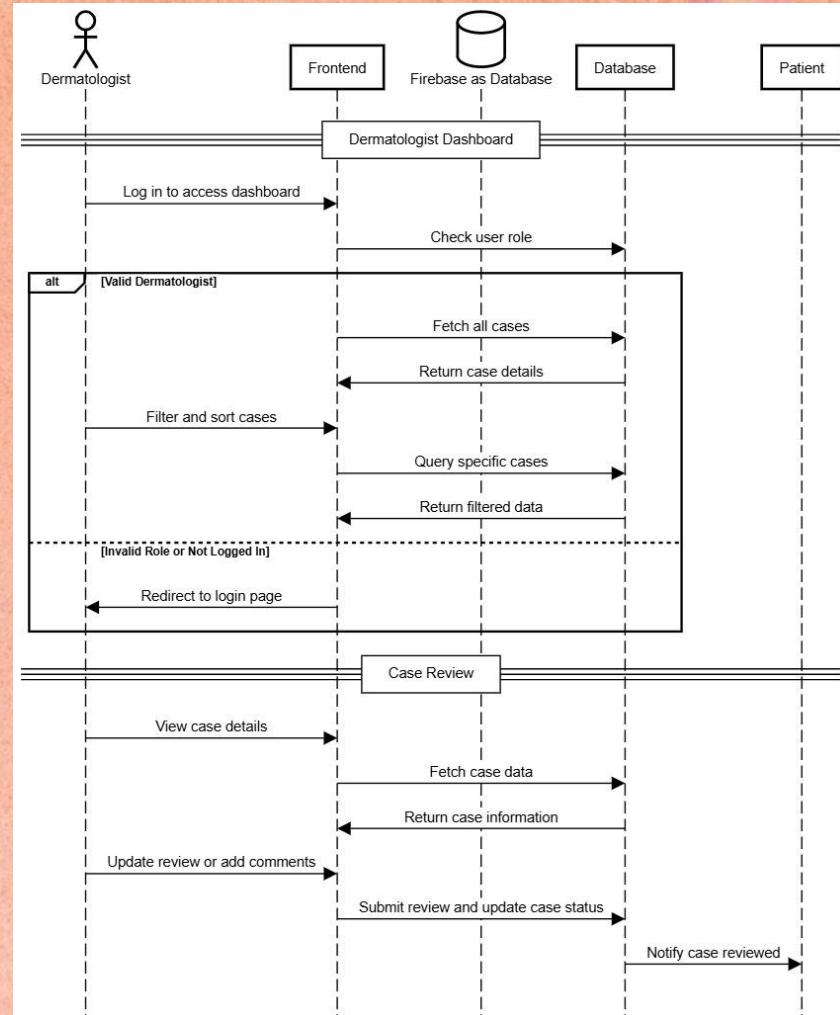
Sequence Diagram



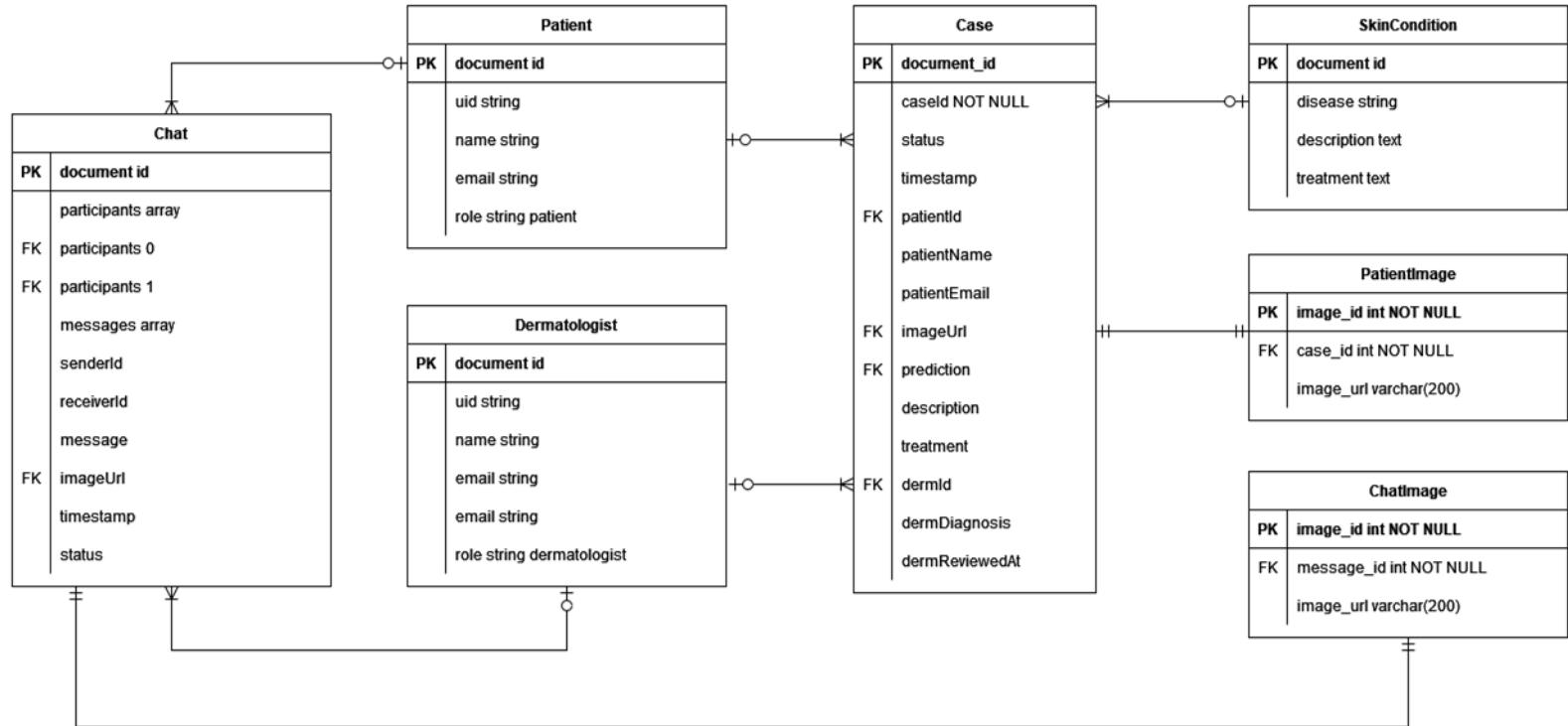
Sequence Diagram



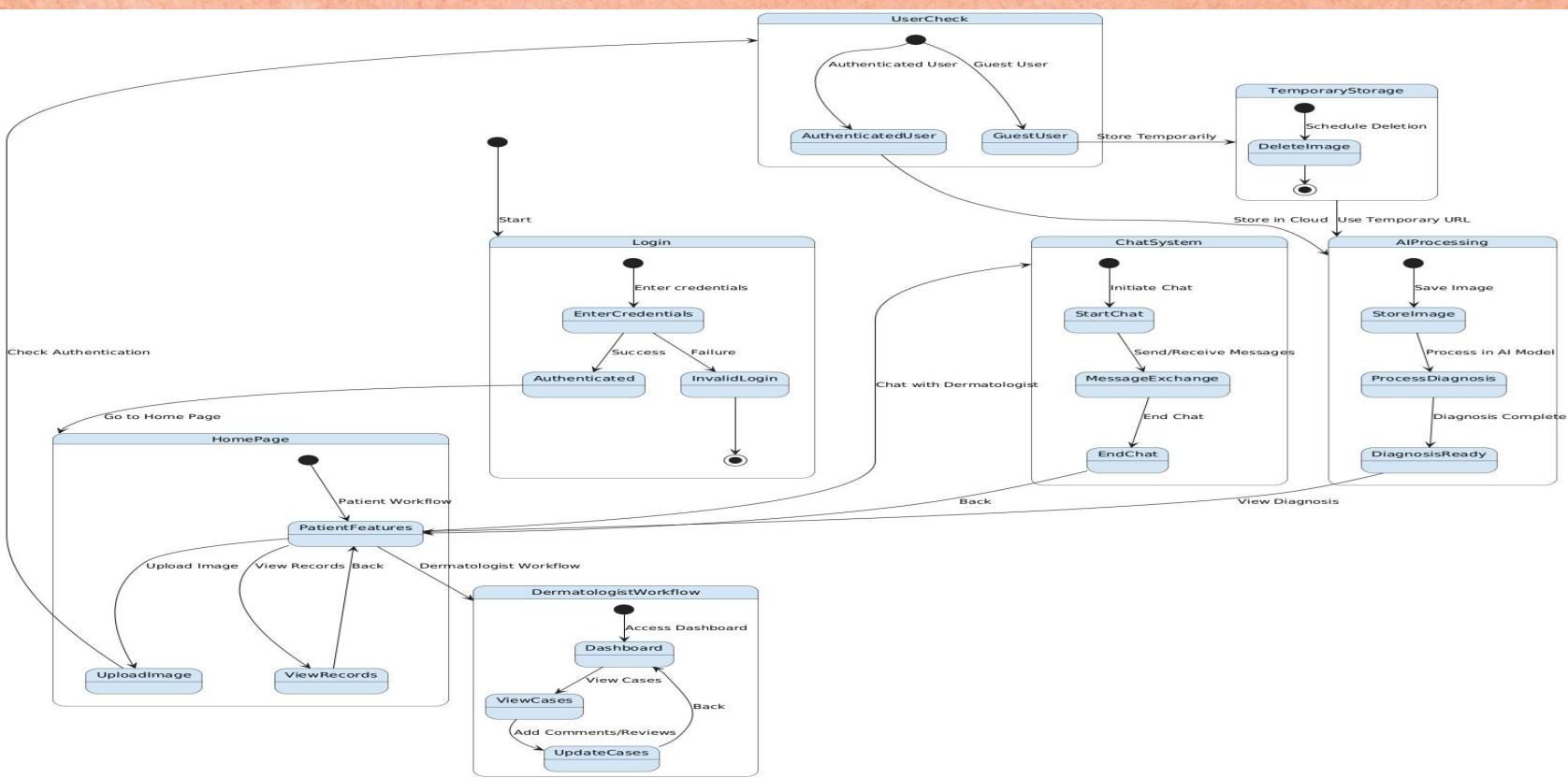
Sequence Diagram



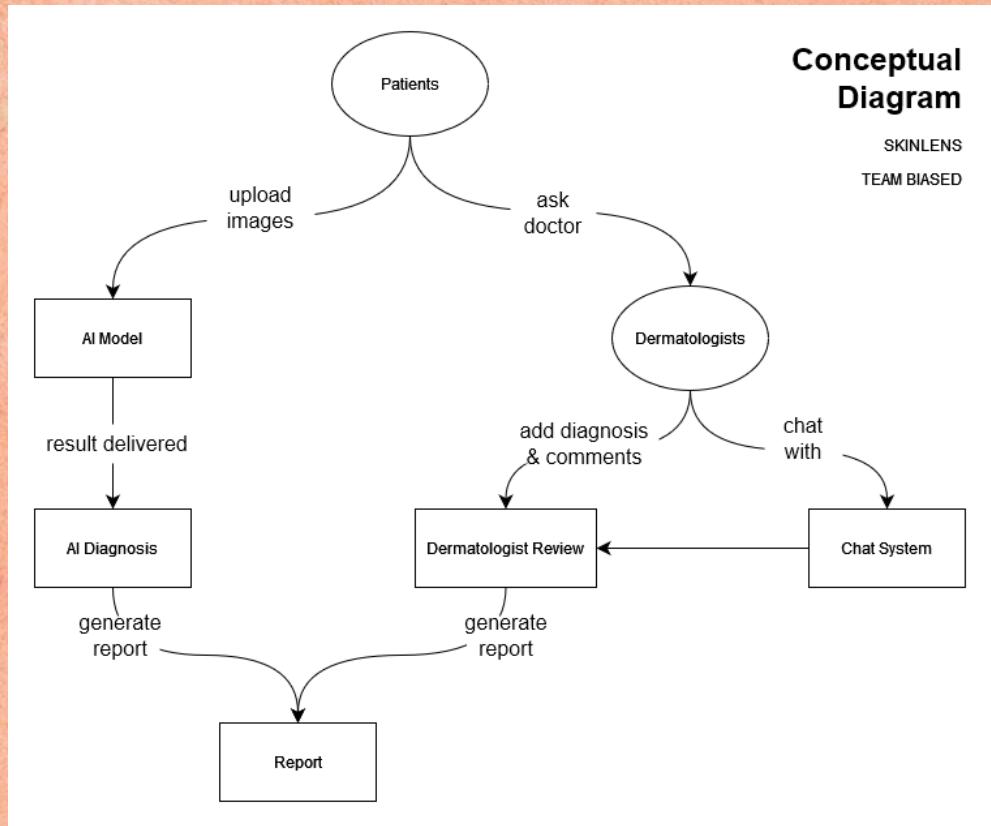
ER Diagram



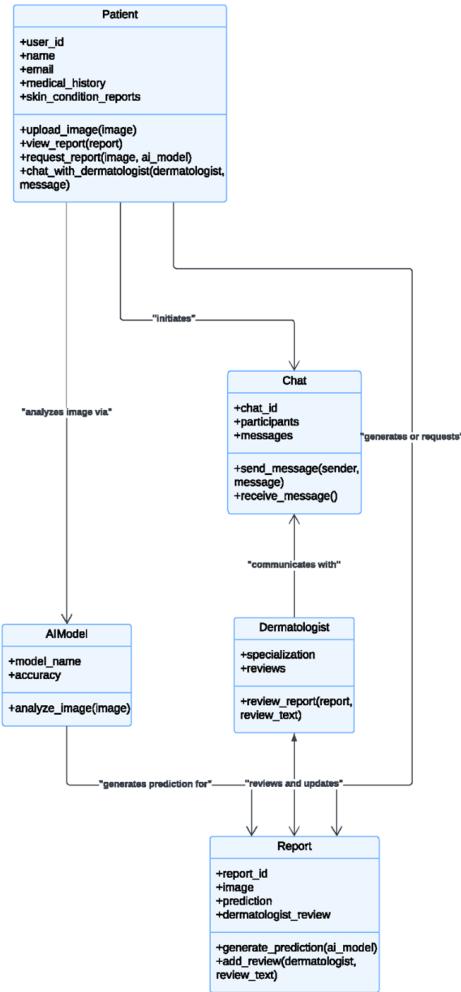
State Diagram



Concept Diagram



Class Diagram



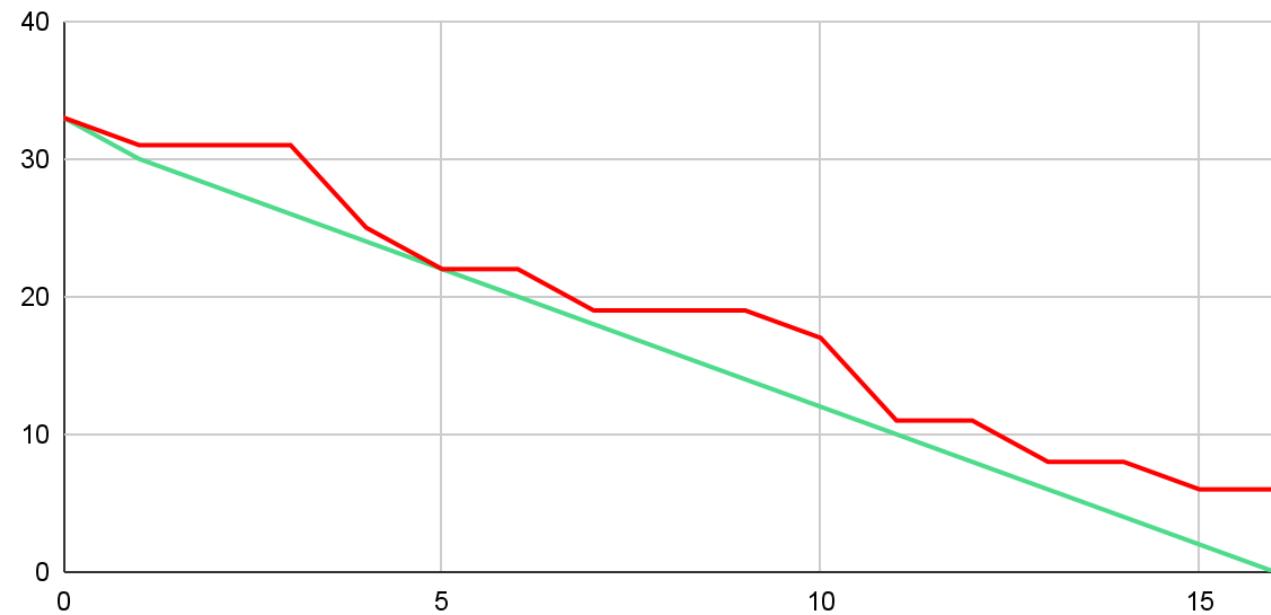
Sprint 2 Recap

- ❖ Extended the number of the skin conditions APP can recognize.
- ❖ Created Firebase Storage to store images.
- ❖ Created Firestore Database to store cases.
- ❖ Added brief description and treatment recommendations when patient got the diagnosis.
- ❖ Developed the report feature, so that patient can review and download report.
- ❖ Developed the sign up and log in pages, and created Firebase to store users account.

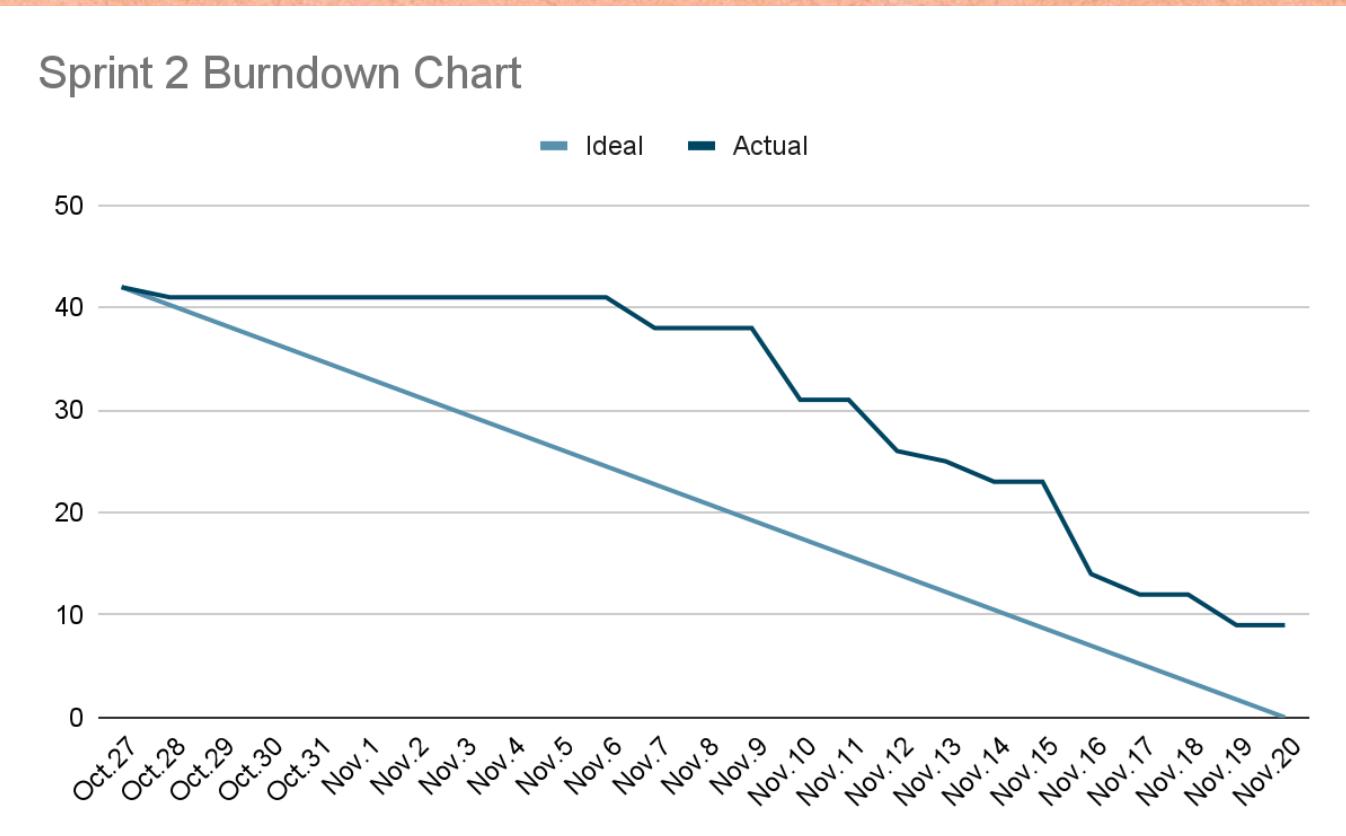
Sprint 1 Summary - BURNDOWN CHART

Burndown Chart for Sprint 1

■ Expected ■ Real



Sprint 2 Summary - BURNDOWN CHART



PRODUCT BACKLOG

Sprint 1

Sprint 2

Sprint 3

No.	User Stories	Acceptance Criteria	Feature	Story Points
US_01	As a patient, I want to access a home page and start a new case so that I can begin my skin condition assessment.	Access Home Page: Given the patient opens the application, When they navigate to the home page, Then the home page is displayed with options to start a new case and clear instructions on how to use the app. Start New Case Button: Given the patient is on the home page, When they click the "Start New Case" button, Then they are directed to the new case interface.	Image Upload	2
US_02	As a patient, I want to access an interface to upload images so that I can provide information about my skin condition.	Access Upload Interface: Given the patient starts a new case, When they proceed to the image upload step, Then an interface appears where they can upload images. Image Selection Options: Given the patient is on the image upload interface, When they choose to upload an image, Then they can either select an image from their device or drag and drop it onto the upload section.	Image Upload	5
US_03	As a patient, I want to select an image from my device and upload it so that it can be used for diagnosis.	The patient can select a single image file from their device. The system accepts specified formats (e.g., JPEG, JPG, PNG).	Image Upload	5
US_04	As a patient, I want to receive a confirmation message and see a preview of my uploaded image so that I know it was uploaded successfully.	Confirmation Message: Given the patient uploads an image successfully, When the system completes the upload process, Then a confirmation message is displayed on the screen. Image Preview: Given the image is uploaded successfully, When the system processes the image, Then a thumbnail preview of the uploaded image is displayed on the interface.	Image Upload	2

PRODUCT BACKLOG

Sprint 1

Sprint 2

Sprint 3

No.	User Stories	Acceptance Criteria	Feature	Story Points
US_05	As a patient, I want the app to analyze my image and recognize basic types of skin conditions so that I can receive a preliminary diagnosis.	Model Triggers Analysis: Given the patient has successfully uploaded an image, When they click the "Analyze" button, Then the app uses the trained AI model to classify the skin condition. Initial Classification Accuracy: Given the model is trained to recognize 5 common skin conditions, When an image is analyzed, Then the prediction results are displayed based on the model's classification accuracy.	AI Diagnosis	8
US_06	As a patient, I want to receive the diagnosis as the name of the condition so that I can understand the skin issue.	Display Diagnosis Name: Given the model completes the analysis, When a diagnosis is generated, Then the name of the skin condition is displayed clearly on the patient's screen. Readable Format: Given the diagnosis is presented to the patient, When it is displayed, Then the name of the condition is shown in an easy-to-read and user-friendly format.	AI Diagnosis	5
US_07	As a patient, I want the uploaded image to be securely stored so that I can access my report anytime.	Secure Image Storage: Given the patient uploads an image, When the upload process is successful, Then the image is securely stored in Firebase Storage with a unique case ID. Image Accessibility: Given the patient has logged into their account, When they access their case details, Then all stored images are displayed and linked to the relevant case.	Image Handling	3
US_08	As a guest user, I want my uploaded image to be deleted after processing so that it doesn't remain stored unnecessarily.	Automatic Deletion: Given the guest uploads an image, When 5 minutes have passed after upload, Then the image is automatically deleted from Firebase Storage.	Image Handling	2

PRODUCT BACKLOG

Sprint 1

Sprint 2

Sprint 3

No.	User Stories	Acceptance Criteria	Feature	Story Points
US_09	As a guest user, I want the system to alert me if my temporary image is unavailable after the session so that I can re-upload it.	<p>Image Unavailable Alert: Given the guest's temporary image is deleted due to timeout, When they attempt to proceed, Then the system displays an alert notifying them about the unavailability.</p> <p>Re-Upload Option: Given the guest's image is deleted, When they choose to upload a new image, Then the system provides an option to re-upload.</p>	Image Handling	1
US_10	As a patient, I want clear error messages if something goes wrong during upload or prediction so that I can take corrective actions.	<p>Error Messages for Uploads: Given the patient uploads an invalid file or encounters a failed upload, When the system detects an issue, Then an error message is displayed explaining the problem and steps to resolve it.</p> <p>Retry Options: Given an error occurs during upload or prediction, When the patient chooses to retry, Then the system provides a retry button to reattempt the process.</p>	Image Handling	1
US_11	As a patient, I want the app to recognize a wide range of skin conditions so that I can receive a diagnosis for diverse issues.	Wide Range of Conditions: Given the patient submits an image, When the diagnosis is completed, Then the app provides predictions for 20+ skin conditions with clear names.	AI Diagnosis	5
US_12	As a patient, I want the diagnosis accuracy to be improved so that I can trust the AI's results.	Model Accuracy Improvement: Given the AI model is retrained with additional data, When new predictions are made, Then the accuracy threshold meets or exceeds 80% for common conditions.	AI Diagnosis	5

PRODUCT BACKLOG

Sprint 1

Sprint 2

Sprint 3

No.	User Stories	Acceptance Criteria	Feature	Story Points
US_13	As a patient, I want to receive the diagnosis result as soon as possible so that I can act quickly if needed.	Fast Processing Time: Given the patient uploads an image for diagnosis, When the system processes the request, Then the results are displayed within 1 minute. Delay Notification: Given processing exceeds the expected time, When the patient waits, Then the system informs them of the delay and provides an estimated wait time.	AI Diagnosis	2
US_14	As a patient, I want to receive a brief description of the diagnosed condition so that I can understand about the skin issue I am facing.	Brief Description: Given the diagnosis is completed, When the result is displayed, Then a brief description is provided in clear, understandable language, including common symptoms and causes.	AI Diagnosis	2
US_15	As a patient, I want to receive treatment recommendations so that I can take action towards treating my skin condition.	Treatment Recommendations: Given the patient receives a diagnosis, When the system displays the result, Then evidence-based treatment recommendations are provided, including over-the-counter remedies or advice to seek professional care.	AI Diagnosis	2
US_16	As a patient, I want to view a comprehensive report of a case so that I have a complete record of my skin condition assessment.	View Comprehensive Report: Given the patient accesses their case, When they open the report, Then it includes metadata, patient details, uploaded images, diagnosis, descriptions, and treatment recommendations.	Report	5

PRODUCT BACKLOG

Sprint 1

Sprint 2

Sprint 3

No.	User Stories	Acceptance Criteria	Feature	Story Points
US_17	As a patient I want to download a detailed report as a PDF so that I can share it with a dermatologist or keep it for personal records..	Downloadable Report: Given the patient opens their report, When they click the "Download PDF" button, Then a PDF version of the report is generated and downloaded, matching the layout of the web application.	Report	1
US_18	As a patient , I want the ability to print my report directly from the app so that I can have a hard copy when needed.	Print Report: Given the patient opens their report, When they click the "Print" button, Then a printable version of the report is generated with a clear and professional layout.	Report	1
US_19	As a patient , I want my reports to be secure and accessible only by me so that my medical data remains private.	Secure Access: Given the patient logs into their account, When they access their reports, Then the system verifies their identity and ensures only they can view the data. Data Security: Given the patient's reports are stored, When the system handles data, Then secure storage methods like Firebase and Firestore encryption are used to ensure privacy.	Report Security and Privacy	2
US_20	As a patient , I want to view my past cases and track their statuses, so that I can monitor my health progress and stay informed about the evaluations.	Case History Access: Given the patient logs into their account, When they navigate to the "Case History" section, Then they can view a chronological list of all past cases. Real-Time Status Updates: Given the patient views their case list, Then the status of a case changes, Then the system updates the status in real time without requiring a refresh. Navigation to Report: Given the patient clicks on a case entry, When the system processes the request, Then the patient is navigated to that case's report.	User Account Management	2

PRODUCT BACKLOG

Sprint 1

Sprint 2

Sprint 3

No.	User Stories	Acceptance Criteria	Feature	Story Points
US_21	As a dermatologist, I want to access and view patient reports so that I can assess their conditions and provide professional feedback.	Access Reports Dashboard: Given the dermatologist logs into their account, When they navigate to the dashboard, Then a list of all assigned patient reports is displayed. Report Sorting Options: Given the dermatologist is on the reports dashboard, When they apply sorting filters (e.g., by date, status, or patient ID), Then the system dynamically updates the displayed list.	User Account Management	2
US_22	As a dermatologist, I want to add comments to patient reports so that I can provide additional guidance if necessary.	Add Comments: Given the dermatologist opens a patient report, When they enter comments in the designated text field, Then the comments are saved and associated with the report. Visibility of Comments: Given the dermatologist submits comments, When the patient views the report, Then the comments are displayed in the relevant section.	Derm Review	2
US_23	As a patient, I want to be notified as soon as my report is reviewed by a dermatologist so that I can stay informed about my case status.	Notification Delivery: Given the dermatologist submits their diagnosis and comments, When the report status changes to "Reviewed," Then the patient receives a notification. Notification Content: Given a notification is sent, When the patient opens it, Then it includes a brief message indicating the report has been reviewed and instructions on how to view the updated report.	Derm Review	2
US_24	As a patient, I want to access a chat interface from my report page so that I can communicate directly with a dermatologist.	Open Chat Interface: Given the patient is on their report page, When they click the "Chat" button, Then a chat window opens displaying active chat requests.	Chat	2

PRODUCT BACKLOG

Sprint 1

Sprint 2

Sprint 3

No.	User Stories	Acceptance Criteria	Feature	Story Points
US_25	As a patient, I want to message my dermatologist in real time so that I can get quick answers to my questions.	Real-Time Messaging: Given a patient initiates a chat with a dermatologist, When they send a message, Then the dermatologist receives the message in real time. Chat Features: Given a chat session is active, When messages are exchanged, Then each message includes a time stamp and typing indicators.	Chat	4
US_26	As a patient, I want to send images to dermatologist so that I can provide additional visual information about my skin condition.	Send Images: Given the patient is in a chat session, When they select and send an image, Then the image is uploaded and displayed in the chat interface. Supported Formats: Given the patient sends an image, When the system processes the file, Then only supported formats (JPEG, JPG, PNG) are accepted.	Chat	2
US_27	As a patient, I want to review the chat history with my dermatologist so that I can refer back to previous discussions and recommendations.	Access Chat History: Given the patient is in an active chat session, When they open the chat interface, Then all past messages are displayed in chronological order with time stamps.	Chat	3
US_28	As a dermatologist, I want to chat with patients in real-time so that I can provide personalized consultation and answer their questions.	Access Active Chat Requests: Given the dermatologist logs into their account, When they navigate to the chat dashboard, Then all active chat requests from patients are displayed. Real-Time Chat: Given the dermatologist accepts a chat request, When Messages are exchanged, Then all messages include timestamps and attachments are supported.	Chat	4

PRODUCT BACKLOG

Sprint 1

Sprint 2

Sprint 3

No.	User Stories	Acceptance Criteria	Feature	Story Points
US_29	As a patient, I want to sign up an account so that I can store my basic information and access personalized services.	Patient Registration: Given the patient accesses the registration page, When they enter their full name, email, password, and date of birth, Then the system creates an account upon form submission.	User Account Management	2
US_30	As a dermatologist, I want to sign up so that I can access patient reports and provide professional feedback.	Dermatologist Registration: Given the dermatologist accesses the registration page, When they enter their full name, email, password, and medical license number, Then the system verifies their details and creates an account.	User Account Management	2
US_31	As a patient, I want to log in to my account so that I can manage my cases and view diagnoses.	Patient Login: Given the patient accesses the login page, When they enter their email and password, Then the system authenticates their credentials and directs them to their dashboard.	User Account Management	1
US_32	As a dermatologist, I want to log in to my account so that I can manage patient reports and update case statuses.	Dermatologist Login: Given the dermatologist accesses the login page, When they enter their email and password, Then the system authenticates their credentials and directs them to their dashboard.	User Account Management	1

Technical User Stories

Sprint 1

Sprint 2

Sprint 3

US_ID	User Stories	Acceptance Criteria	Story Points
TU_01	As a developer, I want to finalize project diagrams so that I have a comprehensive visual representation of the system and its components.	Architecture Diagrams, Context Diagram, ER Diagrams, Sequence Diagram, State Diagrams, Class Diagrams.	1
TU_02	As a developer, I want to create test cases for Sprint 1 user stories so that I can verify the initial functionality is implemented correctly.	Test cases are written for each user story included in Sprint 1. Each test case includes test objectives, preconditions, test steps, expected results, and postconditions.	1
TU_03	As a developer, I want to write a technical paper on the project so that I can document its technical aspects and design decisions for future reference.	The technical paper is reviewed by team members.	3
TU_04	As a developer, I want to create test cases for Sprint 2 user stories so that I can verify the new functionality introduced in the Sprint 2.	Test cases are written for each user story included in Sprint 2. Each test case includes test objectives, preconditions, test steps, expected results, and postconditions.	1



Technical User Stories

Sprint 1

Sprint 2

Sprint 3

US_ID	User Stories	Acceptance Criteria	Story Points
TU_05	As a developer, I want to create an installation manual for the project so that users can easily set up the application in their environment.	The Installation Manual should assist other developers to deploy this application to the production/development environment, and indicate components and tools required to setup the environments. The installation manual is tested by a team member or tester following the instructions to ensure accuracy.	2
TU_06	As a developer, I want to create comprehensive API documentation so that other developers can integrate with and use the API effectively.	Documentation covers all endpoints, including request methods (GET, POST, etc.), parameters, request/response formats, and example requests.	2
TU_07	As a developer, I want to create test cases for Sprint 3 user stories so that I can verify the new functionality introduced in the Sprint 3.	Test cases are written for each user story included in Sprint 3. Each test case includes test objectives, preconditions, test steps, expected results, and postconditions.	1

Sprint 3

SPRINT 3 BACKLOG

US carry over from Sprint 2

User Stories for Sprint 3

Technical User Stories

US_ID	User Stories	Feature	Story Points
US_12	As a patient, I want the diagnosis accuracy to be improved so that I can trust the AI's results.	AI Diagnosis	5
US_17	As a patient, I want to download a detailed report as a PDF so that I can share it with a dermatologist or keep it for personal records..	Report	1
US_31	As a dermatologist, I want to sign up so that I can access patient reports and provide professional feedback.	User Account Management	2
US_33	As a dermatologist, I want to log in to my account so that I can manage patient reports and update case statuses.	User Account Management	1
US_20	As a patient, I want to view my past cases and track their statuses, so that I can monitor my health progress and stay informed about the evaluations.	User Account Management	2
US_21	As a dermatologists, I want to access and view patient reports so that I can assess their conditions and provide professional feedback.	User Account Management	2
US_22	As a dermatologist, I want to add comments to patient reports so that I can provide additional guidance if necessary.	Derm Review	2
US_23	As a patient, I want to be notified as soon as my report is reviewed by a dermatologist so that I can stay informed about my case status.	Derm Review	2

SPRINT 3 BACKLOG

US carry over from Sprint 2

User Stories for Sprint 3

Technical User Stories

US_ID	User Stories	Feature	Story Points
US_24	As a patient, I want to access a chat interface from my report page so that I can communicate directly with a dermatologist.	Chat	2
US_25	As a patient, I want to message my dermatologist in real time so that I can get quick answers to my questions.	Chat	4
US_26	As a patient, I want to send images to dermatologist so that I can provide additional visual information about my skin condition.	Chat	2
US_27	As a patient, I want to review the chat history with my dermatologist so that I can refer back to previous discussions and recommendations.	Chat	3
US_28	As a dermatologist, I want to chat with patients in real-time so that I can provide personalized consultation and answer their questions.	Chat	4
TU_05	As a developer, I want to create an installation manual for the project so that users can easily set up the application in their environment.	Technical	2
TU_06	As a developer, I want to create comprehensive API documentation so that other developers can integrate with and use the API effectively.	Technical	2
TU_07	As a developer, I want to create test cases for Sprint 3 user stories so that I can verify the new functionality introduced in the Sprint 3.	Technical	1

SPRINT 3 TEST CASES

US_12 As a patient, I want the diagnosis accuracy to be improved so that I can trust the AI's results.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_12_01	Validate AI Model Accuracy	1. Input a diverse dataset with known conditions. 2. Test the model on unseen cases.	The AI model achieves a predefined accuracy threshold (e.g., >90%) on the validation dataset.	pass	Dec 12	Tanzil
TC_12_02	Validate Rare Case Predictions	1. Input cases with rare conditions. 2. Check the model's predictions.	The AI model identifies rare conditions accurately within the dataset.	pass	Dec 12	tanzil

SPRINT 3 TEST CASES

US_17 As a patient, I want to download a detailed report as a PDF so that I can share it with a dermatologist or keep it for personal records..

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_17_01	Generate PDF Report	1. Navigate to the report section. 2. Click on the "Download Report" button.	A PDF report is generated with all the relevant details (patient info, diagnosis, and recommendations).	fail	Dec 12	Tanzil
TC_17_02	Verify PDF Download Functionality	1. Attempt to download the PDF report.	The PDF is downloaded successfully to the user's device.	fail	Dec 12	tanzil

SPRINT 3 TEST CASES

US_20 As a patient, I want to view my past cases and track their statuses, so that I can monitor my health progress and stay informed about the evaluations.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_20_01	View Past Cases List	1. Log in as a patient. 2. Navigate to the "My Cases" section.	A list of past cases with statuses is displayed.	Pass	Dec 12	Tanzil
TC_20_02	View Case Status	1. Select a past case.	The case displays statuses such as "Pending", "Reviewed", or "In Progress".	pass	Dec 12	Tanzil
TC_20_03	Filter Cases by Status	1. Apply a status filter (e.g., "Reviewed").	Only cases with the selected status are shown.	pass	Dec 12	Tanzil

SPRINT 3 TEST CASES

US_21 As a dermatologists, I want to access and view patient reports so that I can assess their conditions and provide professional feedback.						
TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_21_01	Access Patient Reports	1. Log in as a dermatologist. 2. Navigate to the "Patient Reports" section.	A list of patient reports is displayed.	Pass	Dec 12	Tanzil
TC_21_02	View Patient Report Details	1. Open a patient report.	Report displays patient details, uploaded images, and symptoms.	Pass	Dec 12	Tanzil

SPRINT 3 TEST CASES

US_22 As a dermatologist, I want to add comments to patient reports so that I can provide additional guidance if necessary.						
TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_22_01	Add Comments to Report	1. Open a patient report. 2. Add a comment. 3. Save the changes.	Comments are saved and visible in the report.	Pass	Dec 12	Tanzil
TC_22_02	Validate Comment Input	1. Attempt to save an empty comment.	An error message prompts for valid input.	pass	Dec 12	Tanzil

SPRINT 3 TEST CASES

US_23 As a patient, I want to be notified as soon as my report is reviewed by a dermatologist so that I can stay informed about my case status.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_23_01	Receive Report Reviewed Notification	1. Dermatologist reviews a patient's report.	Patient receives a notification about the status update.	fail	Dec 12	Tanzil
TC_23_02	Notification Links to Report	1. Click on the notification.	The corresponding report opens.	fail	Dec 12	Tanzil

US_24 As a patient, I want to access a chat interface from my report page so that I can communicate directly with a dermatologist.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_24_01	Access Chat Interface from Report Page	1. Open a patient report. 2. Click on the "Chat with Dermatologist" button.	Chat interface is displayed.	Pass	Dec 12	Tanzil

SPRINT 3 TEST CASES

US_25 As a patient, I want to message my dermatologist in real time so that I can get quick answers to my questions.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_25_01	Send Messages in Real Time (Patient)	1. Open the chat interface. 2. Send a message.	Message is delivered instantly.	pass	Dec 12	Tanzil
TC_25_02	Receive Messages in Real Time (Patient)	1. Dermatologist replies to a message.	Patient sees the reply instantly.	pass	Dec 12	Tanzil

SPRINT 3 TEST CASES

US_26 As a patient, I want to send images to dermatologist so that I can provide additional visual information about my skin condition.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_26_01	Upload and Send Images	1. Open the chat interface. 2. Upload an image. 3. Send the image.	Image is sent and visible to the dermatologist.	Pass	Dec 13	Tanzil
TC_26_02	Validate Supported File Formats	1. Attempt to upload an unsupported file type.	An error message prompts to upload a valid image file.	pass	Dec 13	Tanzil

SPRINT 3 TEST CASES

US_27 As a patient, I want to review the chat history with my dermatologist so that I can refer back to previous discussions and recommendations.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_27_01	View Chat History	1. Open the chat interface. 2. Scroll through the chat history.	Past messages and images are displayed chronologically.	Pass	Dec 13	Tanzil
TC_27_02	Access Chat History from Report Page	1. Open a specific report. 2. Access the chat interface.	Chat history for the corresponding report is displayed.	Pass	Dec 13	Tanzil

SPRINT 3 TEST CASES

US_28 As a dermatologist, I want to chat with patients in real-time so that I can provide personalized consultation and answer their questions.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_28_01	Send Messages in Real Time (Dermatologist)	1. Open the chat interface. 2. Send a message to the patient.	Patient receives the message instantly.	Pass	Dec 13	Tanzil
TC_28_02	Receive Messages in Real Time (Dermatologist)	1. Patient sends a message.	Dermatologist sees the message instantly.	pass	Dec 13	Tanzil

SPRINT 3 TEST CASES

US_31 As a dermatologist, I want to sign up so that I can access patient reports and provide professional feedback.						
TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_31_01	Verify Sign-Up Form Accessibility	1. Navigate to the dermatologist sign-up page. 2. Check the accessibility of the form.	The sign-up form is accessible and visible.	Pass	Dec 12	Tanzil
TC_31_02	Validate Mandatory Fields	1. Submit the form with missing mandatory fields (e.g., name, email, password).	An error message is displayed for missing fields.	Pass	Dec 12	tanzil
TC_31_03	Verify Account Creation	1. Fill in the sign-up form with valid details. 2. Submit the form.	A new dermatologist account is successfully created.	Pass	Dec 12	tanzil

SPRINT 3 TEST CASES

US_33 As a dermatologist, I want to log in to my account so that I can manage patient reports and update case statuses.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_33_01	Verify Login Form Accessibility	1. Navigate to the dermatologist login page. 2. Check if the login form is displayed.	The login form is accessible and visible.	Pass	Dec 12	Tanzil
TC_33_02	Validate Login with Correct Details	1. Enter valid credentials. 2. Click on the "Login" button.	The dermatologist is successfully logged in and redirected to their dashboard.	pass	Dec 12	Tanzil
TC_33_03	Verify Role-Based Access Control	1. Log in as a dermatologist. 2. Attempt to access patient-specific features (e.g., report editing, comments).	Dermatologists can access features related to managing reports and updating case statuses.	pass	Dec 12	Tanzil

SPRINT 3 STORIES COMPLETED OR NOT

US_ID	User Stories	Feature	Story Points	Completed
US_12	As a patient, I want the diagnosis accuracy to be improved so that I can trust the AI's results.	AI Diagnosis	5	Y
US_17	As a patient, I want to download a detailed report as a PDF so that I can share it with a dermatologist or keep it for personal records..	Report	1	N
US_31	As a dermatologist, I want to sign up so that I can access patient reports and provide professional feedback.	User Account Management	2	Y
US_33	As a dermatologist, I want to log in to my account so that I can manage patient reports and update case statuses.	User Account Management	1	Y
US_20	As a patient, I want to view my past cases and track their statuses, so that I can monitor my health progress and stay informed about the evaluations.	User Account Management	2	Y
US_21	As a dermatologists, I want to access and view patient reports so that I can assess their conditions and provide professional feedback.	User Account Management	2	Y
US_22	As a dermatologist, I want to add comments to patient reports so that I can provide additional guidance if necessary.	Derm Review	2	Y
US_23	As a patient, I want to be notified as soon as my report is reviewed by a dermatologist so that I can stay informed about my case status.	Derm Review	2	N

SPRINT 3 STORIES COMPLETED OR NOT

US_ID	User Stories	Feature	Story Points	Completed
US_24	As a patient, I want to access a chat interface from my report page so that I can communicate directly with a dermatologist.	Chat	2	Y
US_25	As a patient, I want to message my dermatologist in real time so that I can get quick answers to my questions.	Chat	4	Y
US_26	As a patient, I want to send images to dermatologist so that I can provide additional visual information about my skin condition.	Chat	2	Y
US_27	As a patient, I want to review the chat history with my dermatologist so that I can refer back to previous discussions and recommendations.	Chat	3	Y
US_28	As a dermatologist, I want to chat with patients in real-time so that I can provide personalized consultation and answer their questions.	Chat	4	Y
TU_05	As a developer, I want to create an installation manual for the project so that users can easily set up the application in their environment.	Technical	2	Y
TU_06	As a developer, I want to create comprehensive API documentation so that other developers can integrate with and use the API effectively.	Technical	2	Y
TU_07	As a developer, I want to create test cases for Sprint 3 user stories so that I can verify the new functionality introduced in the Sprint 3.	Technical	1	Y

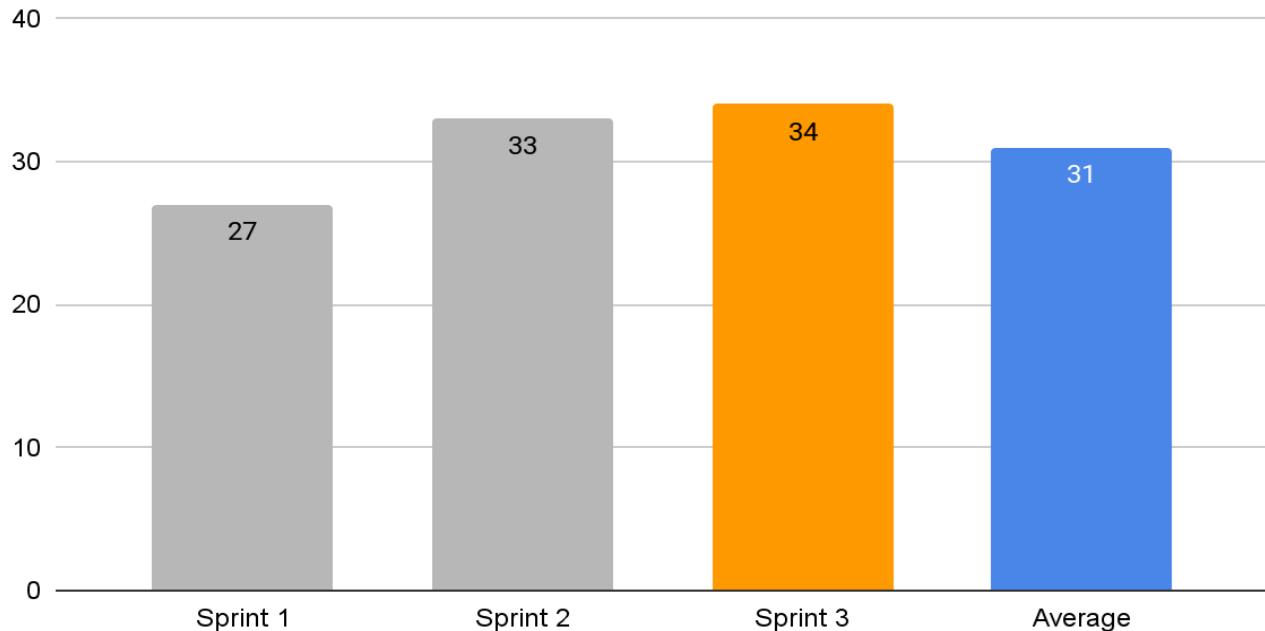
Sprint 3 METRICS - Team Velocity



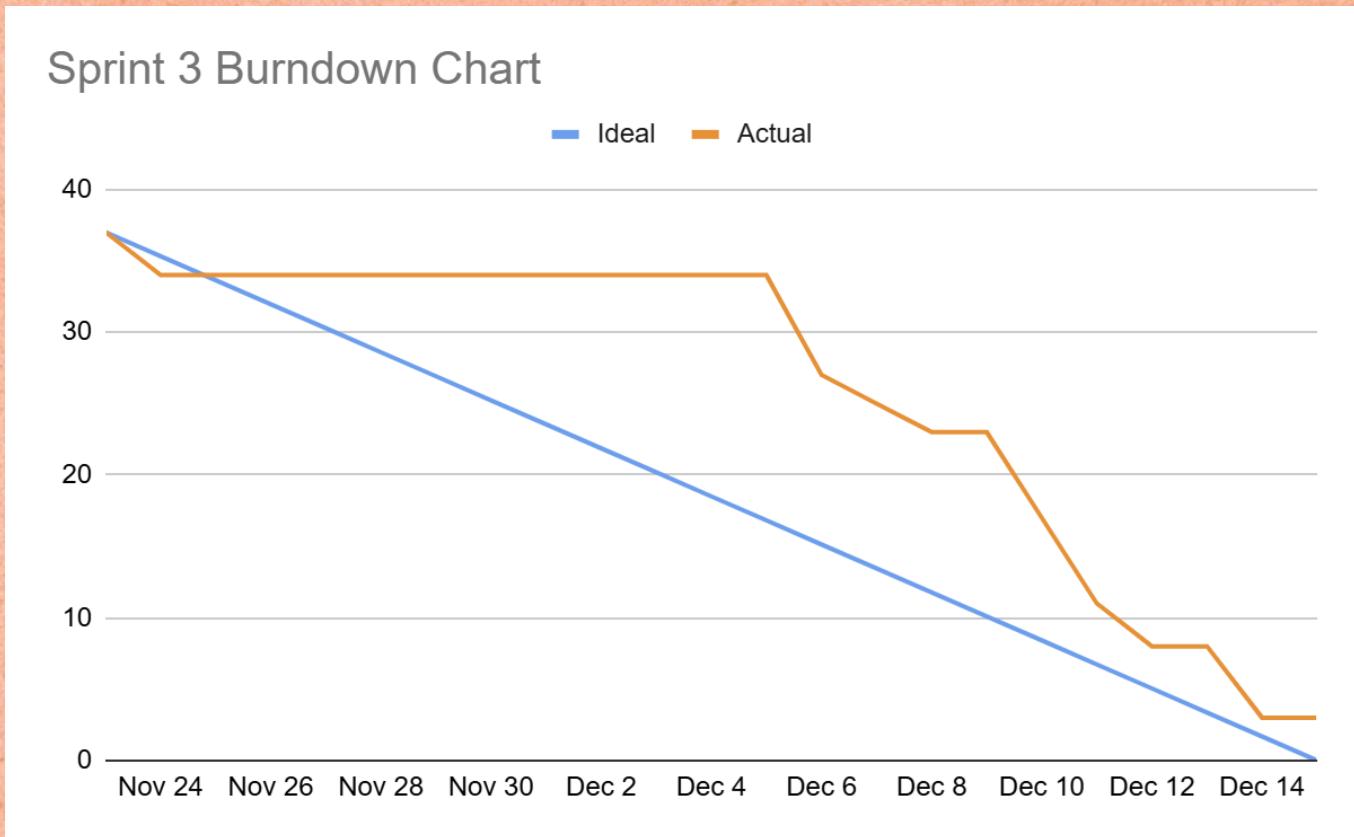
Team velocity of Sprint 3: 34 story points

Team velocity of Average: 31 story points

Sprint 3 Team Velocity



Sprint 3 METRICS - BURNDOWN CHART

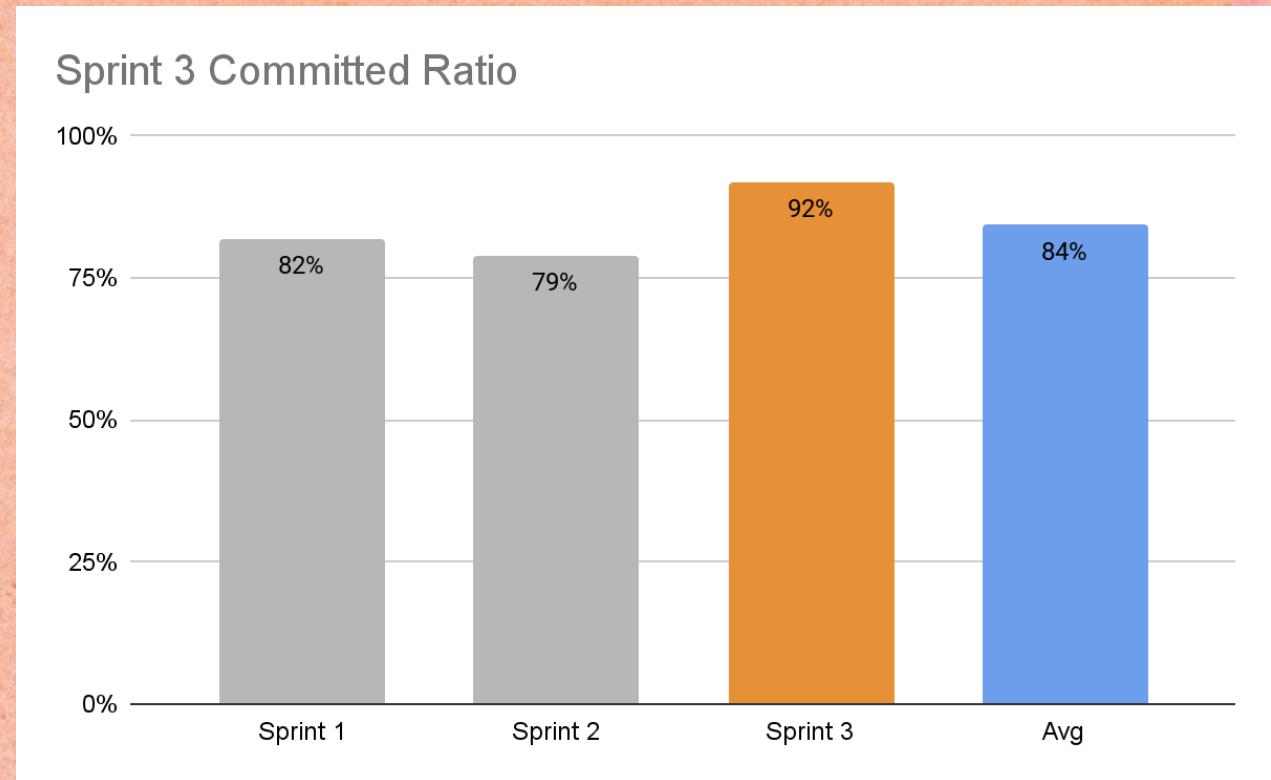


Sprint 3 METRICS - COMMITTED RATIO



Committed Ratio of Sprint 3 is **92%**.

We completed **34** story points as **37** committed.



Sprint 3 Retrospective

Few important points from the Retrospective

What did we do right?

- Brainstorming ideas in each step of project went well
- User stories and tasks were well-defined, so each team member understood their responsibilities and deliverables
- Implemented the user friendly interface for all users, guest, patient and dermatologist.
- Everybody work really hard even to midnight /before dawn.
- Willing to accept advice and changes
- Good management of work from team leader and scrum master led to success of MVP

Few important points from the Retrospective

If we were to do it again, what would we do differently?

- If we were to do it again, we would have prioritized the task on deciding the final dataset
- We would take up responsibility of the assigned part, that is propose at least an alternative if the assigned task cannot be done.
- Watch Demo together and ask for everyone's feedback after the new feature development.
- Actively seeking for solution and actions together when project facing obstacles, even if not their tasks.
- Do research more deeply, download previous model and test at the beginning, learn some deeper algorithms and principle.

Few important points from the Retrospective

Action items

- If we were to do it again, we would have prioritized the task on deciding the final dataset. Assigned to Zaid - prioritize the tasks in the planning meeting
- We would take up responsibility of the assigned part, that is propose at least an alternative if the assigned task cannot be done. Assigned to everyone- Come with alternate solutions and discuss with the team
- Watch Demo together and ask for everyone's feedback after the new feature development. Assigned to Everyone: Get feedback from the team after developing a feature
- Actively seeking for solution and actions together when project facing obstacles, even if not their tasks. Assigned to Everyone: Let the team know if facing any issues as soon as possible
- Do research more deeply, download previous model and test at the beginning, learn some deeper algorithms and principle. Assigned to everyone- Get familiar with tech stack and do more research

SPRINT 3 RETROSPECTIVE



start typing to filter stickies

View Section

All Sections

Sort By

votes

Sprint 3 Retrospective

What did we do right? +

brainstorming ideas in each step of project went well	User stories and tasks were well-defined, so each team member understood their responsibilities and deliverables
+ 5	+ 5
Implemented the user friendly interface for all users, guest, patient and dermatologist.	Everybody work really hard even to midnight/before dawn.
+ 4	+ 4
Willing to accept advise and changes	Good management of work from team leader and scrum master led to success of MVP
+ 4	+ 4
Scrum meetings went as planned	Good atmosphere to work together because of effective communication
+ 3	+ 1
Clearly tasks division and assignment in Sprint Planning.	Everyone did WONDERFUL in Agile.
+ 1	+ 1
doing their works in time	Team Leader coordinated everything well

If we were to do it again, what would we do differently? +

Update their work on time, even there is no output or positive result, that others could know the progress and discussing solution together.	We would collaborate more while coding a same feature so that there are no conflicts in the code
+ 2	+ 1
more clear instructions.	we would not be shy and overacted, just saying whenever and whatever we wanna say, don't be afraid to interrupt in a respectful way
+ 1	+ 1
Get familiar with Github	We would start writing the technical paper parallelly
+ 1	+ 0
interruptions are going on during meetings	To update timely.
+ 0	+ 0

Action items +

<p>we would take up responsibility of the assigned part, that is propose at least an alternative if the assigned task cannot be done. Assigned to everyone- Come with alternate solutions and discuss with the team</p> <p>if we were to do it again, we would have prioritized the task on deciding the final dataset. Assigned to Zaid - prioritize the tasks in the planning meeting</p> <p>+ 5</p>	<p>Watch Demo together and ask for everyone's feedback after the new feature development. Assigned to Everyone: Get feedback from the team after developing a feature</p> <p>+ 4</p> <p>Actively seeking for solution and actions together when project facing obstacles, even if not their tasks. Assigned to Everyone: Let the team know if facing any issues as soon as possible</p>
<p>Do research more deeply, download previous model and test at the beginning, learn some deeper algorithms and principle. Assigned to everyone- Get familiar with tech stack and do more research</p> <p>+ 0</p>	<p>Before jumping to alternate solutions we could find the underlying cause of the problem by dedicating a special meeting for the same. Assigned to everyone- Let the team know if facing any difficulties. Assigned to</p> <p>+ 0</p>

PROJECT DEMO- SCRENSHOT

SkinLens

Upload

Records

Chat

Info

Hello, MdZaid

Logout

Prediction Result



Actinic Keratoses and Bowen's Disease

Description

Rough, scaly patches or plaques on sun-exposed skin, often precancerous. May appear as red or brown lesions with crusting. Can lead to squamous cell carcinoma if untreated.

Treatment Recommendations

Cryotherapy, topical fluorouracil, and photodynamic therapy.

[View Report](#)

PROJECT DEMO- SCREENSHOT

SkinLens

Reports

Chat

Info

Hello, Admin@Zaid

Logout

Dermatologist Dashboard



zaid

Patient Name ▾

CASE ID	PATIENT NAME	DATE ▾	DIAGNOSIS	STATUS	ACTIONS
SKIN-60	MdZaid zaidakhtar313@gmail.com	12/13/2024, 10:01:50 PM	Squamous Cell Carcinoma	Pending Review	<button>View Report</button> <button>Take Action</button>
SKIN-48	MdZaid zaidakhtar313@gmail.com	12/12/2024, 7:38:58 AM	Actinic Keratoses and Bowen's Disease	Reviewed	<button>View Report</button> <button>Update Review</button>
SKIN-47	MdZaid zaidakhtar313@gmail.com	12/12/2024, 3:21:25 AM	Pending	Pending Review	<button>View Report</button> <button>Take Action</button>
SKIN-46	MdZaid zaidakhtar313@gmail.com	12/12/2024, 3:20:22 AM	Melanoma	Pending Review	<button>View Report</button> <button>Take Action</button>
SKIN-45	MdZaid zaidakhtar313@gmail.com	12/12/2024, 2:35:03 AM	Melanoma	Pending Review	<button>View Report</button> <button>Take Action</button>
SKIN-44	MdZaid zaidakhtar313@gmail.com	12/11/2024, 11:29:33 PM	Squamous Cell Carcinoma	Reviewed	<button>View Report</button> <button>Update Review</button>

PROJECT DEMO- SCREENSHOT

SkinLens

Reports Chat Info

Hello, Admin@Zaid

Logout

Dermatologist Dashboard



SKINLENS COMPREHENSIVE ANALYSIS REPORT

Patient Name: MdZaid
Email: zaidakhtar313@gmail.com

Case #: SKIN-48
Date: 12/12/2024, 7:38:58 AM

Uploaded Image



AI Analysis

AI analysis suggests characteristics consistent with:

Actinic Keratoses and Bowen's Disease

Rough, scaly patches or plaques on sun-exposed skin, often precancerous. May appear as red or brown lesions with crusting. Can lead to squamous cell carcinoma if untreated.

Dermatologist Review

Comment:

correct

Reviewed on: 12/12/2024, 7:40:22 AM

Reviewed By: Admin@Zaid

Treatment recommendations

Cryotherapy, topical fluorouracil, and photodynamic therapy.

PROJECT DEMO- SCREENSHOT

SkinLens

Upload

Records

Chat

Info

Hello, MdZaid

Logout

My Records

 Search by Case ID...

Sort by Date ▾

All ▾

CASE ID	DATE	DIAGNOSIS	STATUS	ACTIONS
SKIN-32	12/7/2024, 10:25:15 PM	Melanoma	Pending Review	<button>View Report</button>
SKIN-27	12/7/2024, 9:44:10 PM	Squamous Cell Carcinoma	Pending Review	<button>View Report</button>
SKIN-31	12/7/2024, 10:22:00 PM	Melanoma	Pending Review	<button>View Report</button>
SKIN-28	12/7/2024, 9:54:25 PM	Squamous Cell Carcinoma	Pending Review	<button>View Report</button>
SKIN-37	12/7/2024, 10:51:23 PM	Melanoma	Pending Review	<button>View Report</button>

Previous

Page 1 of 6

Next

PROJECT DEMO- SCREENSHOT

SkinLens

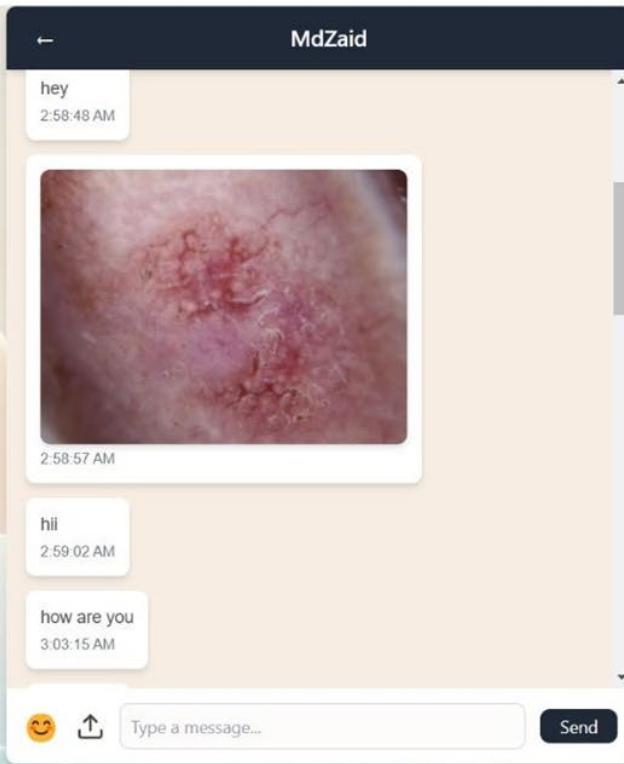
Reports

Chat

Info

Hello, Admin@Zaid

Logout



PROJECT - APIs

1. Authentication APIs

1.1 Sign-Up API

Purpose: Creates a new user account with email, password, and additional user details in Firebase Authentication and Firestore.

Method: **POST** via Firebase SDK.

Process:

1. Input: User's email, password, name, and role (e.g., patient or dermatologist).
2. Output: A new user account is created in Firebase Authentication, and corresponding user details are stored in Firestore.

Key Code:

```
const userCredential = await createUserWithEmailAndPassword(auth, email, password);
await setDoc(doc(db, "users", userCredential.user.uid), {
  name: name,
  email: email,
  role: role,
  uid: userCredential.user.uid,
});
```

PROJECT - APIs

1.2 Login API

Purpose: Authenticates the user and grants access to the application.

Method: `POST` via Firebase SDK.

Process:

1. Input: User's email and password.
2. Output: Authenticated user session and Firebase token.

Key Code:

```
await signInWithEmailAndPassword(auth, email, password);
```

PROJECT - APIs

2. Image Handling APIs

2.1 Image Upload API

Purpose: Uploads the user's image to Firebase Storage and generates a unique URL for accessing it.

Method: Indirectly through Firebase Storage SDK.

Process:

1. Input: Image file selected by the user.
2. Output: URL of the uploaded image.

Key Code:

```
const storage = getStorage();
const storageRef = ref(storage, `cases/${user.uid}/${uuidv4()}/${file.name}`);
await uploadBytes(storageRef, file);
const downloadURL = await getDownloadURL(storageRef);
```

PROJECT - APIs

2.2 Prediction API

Purpose: Sends the uploaded image to the backend for AI-based prediction and fetches the diagnosis result.

Method: **POST** via Axios.

Process:

1. Input: URL of the uploaded image.
2. Output: Predicted disease name, description, and treatment recommendations.

Key Code:

```
const response = await axios.post("http://localhost:5000/predict", { imageUrl });
```

PROJECT - APIs

2.3 Temporary Image Deletion API

Purpose: Deletes temporary images uploaded by unauthenticated users from Firebase Storage.

Method: Indirectly through Firebase Storage SDK.

Process:

1. Input: URL of the temporary image.
2. Output: Deletes the image from Firebase Storage.

Key Code:

```
const imageRef = ref(storage, imageUrl);
await deleteObject(imageRef);
```

PROJECT - APIs

3. Case Management APIs

3.1 Generate Case ID API

Purpose: Generates a unique case ID for each new case.

Method: Programmatic generation using UUID and sequential ID logic.

Process:

1. Input: None.
2. Output: A unique case ID in the format **SKIN-<incremental_number>**.

Key Code:

```
let newCaseId = "SKIN-1";
const casesSnapshot = await getDocs(collection(db, "cases"));
const maxId = Math.max(0, ...casesSnapshot.docs.map(doc =>
  parseInt(doc.data().caseId.split("-")[1])));
newCaseId = `SKIN-${maxId + 1}`;
```

PROJECT - APIs

3.2 Save Case Details API

Purpose: Saves case details to Firestore, including patient info, uploaded image URL, and timestamp.

Method: **POST** via Firestore SDK.

Process:

1. Input: Case details, including patient ID, email, image URL, and timestamp.
2. Output: Case document saved in Firestore.

Key Code:

```
await setDoc(doc(collection(db, "cases"), newCaseId), {  
  caseId: newCaseId,  
  patientId: user.uid,  
  patientEmail: user.email,  
  imageUrl: downloadURL,  
  timestamp: new Date().toISOString(),  
  status: "Open",  
});
```

PROJECT - APIs

3.3 Fetch Case Details API

Purpose: Retrieves details of a specific case from Firestore.

Method: GET via Firestore SDK.

Process:

1. Input: Case ID.
2. Output: Case details, including patient info, image URL, and status.

Key Code:

```
const caseRef = doc(db, "cases", caseId);
const caseData = (await getDoc(caseRef)).data();
```

PROJECT - APIs

3.4 Update Case Status API

Purpose: Updates the status of a case in Firestore (e.g., from "Pending Review" to "Reviewed").

Method: **PATCH** via Firestore SDK.

Process:

1. Input: Case ID and new status.
2. Output: Updated case document in Firestore.

Key Code:

```
await updateDoc(doc(db, "cases", caseId), { status: "Reviewed" });
```

PROJECT - APIs

3.5 Dermatologist Comment API

Purpose: Allows dermatologists to add comments to a case.

Method: [PATCH](#) via Firestore SDK.

Process:

1. Input: Case ID and dermatologist comment.
2. Output: Updated case document with dermatologist's comment.

Key Code:

```
await updateDoc(doc(db, "cases", caseId), { dermDiagnosis: diagnosis });
```

PROJECT - APIs

3.6 Fetch Reviewed By API

Purpose: Retrieves the name of the dermatologist who reviewed a case.

Method: **GET** via Firestore SDK.

Process:

1. Input: Case ID.
2. Output: Dermatologist's name.

Key Code:

```
const reviewedBy = (await getDoc(doc(db, "cases", caseId))).data().reviewedBy;
```

PROJECT - APIs

4. Chat APIs

4.1 Fetch User List API

Purpose: Retrieves a list of users available for chatting based on roles.

Method: `GET` via Firestore SDK.

Process:

1. Input: Current user's role.
2. Output: List of users with the opposite role (e.g., patients for dermatologists).

Key Code:

```
const userQuery = query(collection(db, "users"), where("role", "==", roleFilter));
const usersList = (await getDocs(userQuery)).docs.map(doc => doc.data());
```

PROJECT - APIs

4.2 Fetch Chat Messages API

Purpose: Retrieves chat messages between two users.

Method: [GET](#) via Firestore SDK.

Process:

1. Input: Chat ID (generated by combining both user IDs).
2. Output: List of messages in chronological order.

Key Code:

```
const chatRef = doc(db, "chats", chatId);
const messages = (await getDoc(chatRef)).data()?.messages || [];
```

PROJECT - APIs

4.3 Send Message API

Purpose: Sends a message (text or image) to another user.

Method: **POST** via Firestore SDK.

Process:

1. Input: Sender ID, receiver ID, message text or image URL, and timestamp.
2. Output: Message appended to the chat document in Firestore.

Key Code:

```
const messageData = {  
  senderId: currentUser.uid,  
  receiverId: otherUserId,  
  message: newMessage.trim(),  
  imageUrl,  
  timestamp: new Date().toISOString(),  
  status: "sent",  
};  
await updateDoc(doc(db, "chats", chatId), { messages: arrayUnion(messageData)  
});
```

PROJECT - APIs

4.4 Upload Chat Image API

Purpose: Uploads an image for a chat and generates a downloadable URL.

Method: Indirectly through Firebase Storage SDK.

Process:

1. Input: Image file selected by the user.
2. Output: URL of the uploaded image.

Key Code:

```
const fileRef = ref(storage, `chat-images/${chatId}/${file.name}`);
await uploadBytes(fileRef, file);
const imageUrl = await getDownloadURL(fileRef);
```

PROJECT - APIs

4.5 Update Message Status API

Purpose: Updates the status of a message (e.g., from "sent" to "delivered" or "read").

Method: **PATCH** via Firestore SDK.

Process:

1. Input: Chat ID and message ID.
2. Output: Updated message status in Firestore.

Key Code:

```
const updatedMessages = messages.map(msg =>
  msg.receiverId === currentUser.uid && msg.status === "sent"
    ? { ...msg, status: "delivered" }
    : msg
);
await updateDoc(doc(db, "chats", chatId), { messages: updatedMessages });
```

PROJECT - APIs

4.6 Fetch Unread Message Count API

Purpose: Fetches the count of unread messages for each chat.

Method: **GET** via Firestore SDK.

Process:

1. Input: Current user ID.
2. Output: Count of unread messages for each user.

Key Code:

```
const unreadCounts = messages.filter(  
  msg => msg.receiverId === currentUser.uid && msg.status !== "read"  
).length;
```

PROJECT - APIs

5. Report APIs

5.1 View Report API

Purpose: Fetches and displays a detailed case report.

Method: **GET** via Firestore SDK.

Process:

1. Input: Case ID.
2. Output: Case details, including patient info, uploaded image, prediction, dermatologist comment, and treatment plan.

Key Code:

```
const reportData = (await getDoc(doc(db, "cases", caseId))).data();
```

PROJECT - APIs

5.2 Generate PDF Report API

Purpose: Generates a downloadable PDF of the case report.

Method: Client-side rendering with `html2canvas` and `jsPDF`.

Process:

1. Input: HTML element containing report details.
2. Output: PDF file downloaded to the user's device.

Key Code:

```
const canvas = await html2canvas(document.getElementById("report-content"));
const pdf = new jsPDF();
pdf.addImage(canvas.toDataURL("image/png"), "PNG", 10, 10, 190, canvas.height * 190 /
    canvas.width);
pdf.save(`SkinLens_Report_${caseId}.pdf`);
```

PROJECT - APIs

5.3 Fetch Disease Details API

Purpose: Retrieves the description and treatment for a specific disease.

Method: **GET** via Firestore SDK.

Process:

1. Input: Disease name.
2. Output: Disease description and treatment details.

Key Code:

```
const diseasesRef = collection(db, "diseases");
const querySnapshot = await getDocs(query(diseasesRef, where("disease", "==",
diseaseName)));
const diseaseDetails = querySnapshot.docs[0]?.data();
```

PROJECT - APIs

6. Dashboard APIs

6.1 Fetch Cases for Dermatologist API

Purpose: Retrieves all cases visible to the logged-in dermatologist.

Method: `GET` via Firestore SDK.

Process:

1. Input: None.
2. Output: List of cases assigned to or available for the dermatologist.

Key Code:

```
const casesRef = collection(db, "cases");
const casesSnapshot = await getDocs(casesRef);
const casesList = casesSnapshot.docs.map(doc => doc.data());
```

PROJECT - APIs

6.2 Filter and Sort Cases API

Purpose: Filters and sorts cases based on selected criteria (e.g., date, status).

Method: Client-side filtering and sorting.

Process:

1. Input: Filter attribute, search term, and sort key.
2. Output: Filtered and sorted list of cases.

Key Code:

```
const filteredCases = cases.filter(case =>
  case[filterAttribute]?.toLowerCase().includes(searchTerm.toLowerCase())
).sort((a, b) => (a[sortKey] > b[sortKey] ? 1 : -1));
```

PROJECT - APIs

6.3 Pagination API

Purpose: Paginates the list of cases displayed on the dashboard.

Method: Client-side pagination logic.

Process:

1. Input: Current page number and items per page.
2. Output: Cases for the current page.

Key Code:

```
const currentData = filteredCases.slice((currentPage - 1) * itemsPerPage, currentPage *  
itemsPerPage);
```

PROJECT - APIs

7. Patient Dashboard APIs

7.1 Fetch Cases for Patients API

Purpose: Retrieves all cases for the logged-in patient.

Method: **GET** via Firestore SDK.

Process:

1. Input: Patient ID.
2. Output: List of cases associated with the patient.

Key Code:

```
const userCases = casesSnapshot.docs.filter(doc => doc.data().patientId === user.uid);
```

PROJECT - APIs

7.2 Filter and Search Cases API

Purpose: Filters and searches cases for the logged-in patient.

Method: Client-side filtering.

Process:

1. Input: Search term and status filter.
2. Output: Filtered list of cases.

Key Code:

```
7.2 Filter and Search Cases API
Purpose: Filters and searches cases for the logged-in patient.
Method: Client-side filtering.
Process:

Input: Search term and status filter.
Output: Filtered list of cases.
Key Code:
```

PROJECT - APIs

7.3 Fetch Logged-in User Data API

Purpose: Retrieves the details of the logged-in user (name, email, role).

Method: [GET](#) via Firestore SDK.

Process:

1. Input: User ID.
2. Output: User details from Firestore.

Key Code:

```
const userData = (await getDoc(doc(db, "users", user.uid))).data();
```

PROJECT- APIs

Flask API and CORS

```
# Initialize Flask app
app = Flask(__name__)
CORS(app) # Allow cross-origin requests from React frontend

# Load pre-trained model
model = keras.models.load_model("models/skinlensAI.keras")

# Define route for prediction
@app.route('/predict', methods=['POST'])
def predict():
    if 'image' not in request.files:
        return jsonify({'error': 'No image uploaded'}), 400

    # Get the uploaded image
    f = request.files['image']

    # Save the file temporarily
    basepath = os.path.dirname(__file__)
    filepath = os.path.join(basepath, 'uploads', f.filename)
    f.save(filepath)
```

Axios

```
const handleSubmit = async () => {
  if (!file) {
    alert("Please upload an image first.");
    return;
  }

  // Create FormData object and append the image file
  const formData = new FormData();
  formData.append("image", file);

  try {
    // Send POST request to Flask backend
    const response = await axios.post("http://localhost:5000/predict", formData, {
      headers: {
        "Content-Type": "multipart/form-data",
      },
    });

    // Set the prediction result to state
    setPrediction(response.data.prediction);
  } catch (error) {
    console.error("Error uploading file:", error);
  }
};
```

Future Enhancement

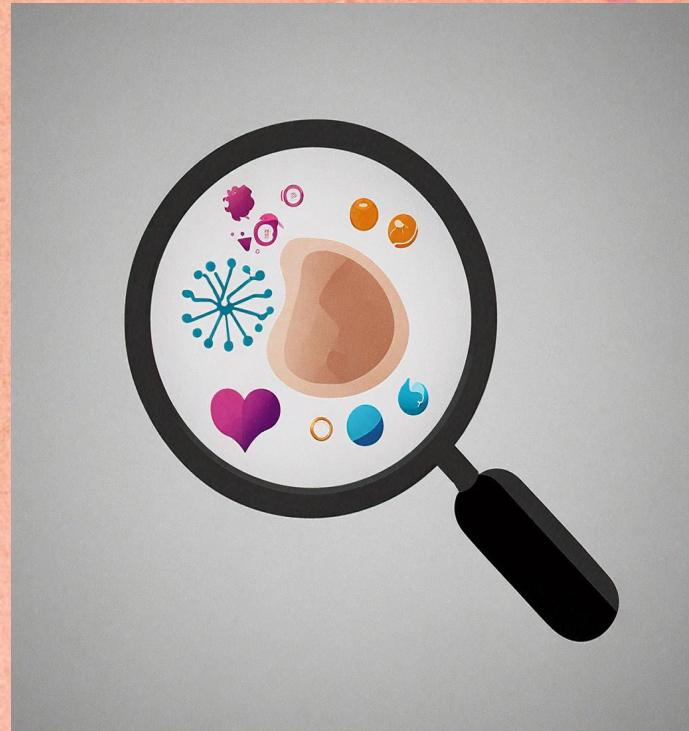
1. Integrate multi-modal learning by combining image data with text-based inputs, such as patient-entered symptoms and clinical notes, for enhanced diagnostic accuracy.
2. Enhance chat functionality with AI-powered conversational agents to automate routine queries and improve patient response times.
3. Integrate an appointment scheduling system.

WIKI PAGE LINK

To see our final work,

Click on the link below.

<https://github.com/htmw/2024F-Biased/wiki>



DEMO

THANK YOU!