Team Biased

SkinLens

Your Personal Dermatologist

AGENDA

01

Roles &
Responsibilities
Improvements

02

Problem Statement
Project Description
Working Agreement

03

Personas
MVP
Project Design

04

Technologies
Algorithms
Diagrams

05

Project backlog
Sprint 1 Summary

06

Sprint 2:

- Stories
- Test cases
- Completed
- Not completed

07

Sprint 2 Metrics:

- Velocity
- Burndown Charts
- Committed Ratio

08

Sprint 2 Retrospective
Sprint 3 Planning
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

- Story Points according to fibonacci number
- Headings on all pages
- Split bigger user stories into small user stories
- Separate Tasks from user stories
- Use Given, When, Then in acceptance Criteria
- Clear distinctions between user stories completed and not completed

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:	For patients who wants to quickly access their skin conditions, for dermatologists who wants to quickly access patient's skin conditions 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. Unlike traditional visual inspections, SkinLens uses advanced image-based deep learning model and offers accurate predictions and recommend treatments.
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.

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 to 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 children 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)

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.

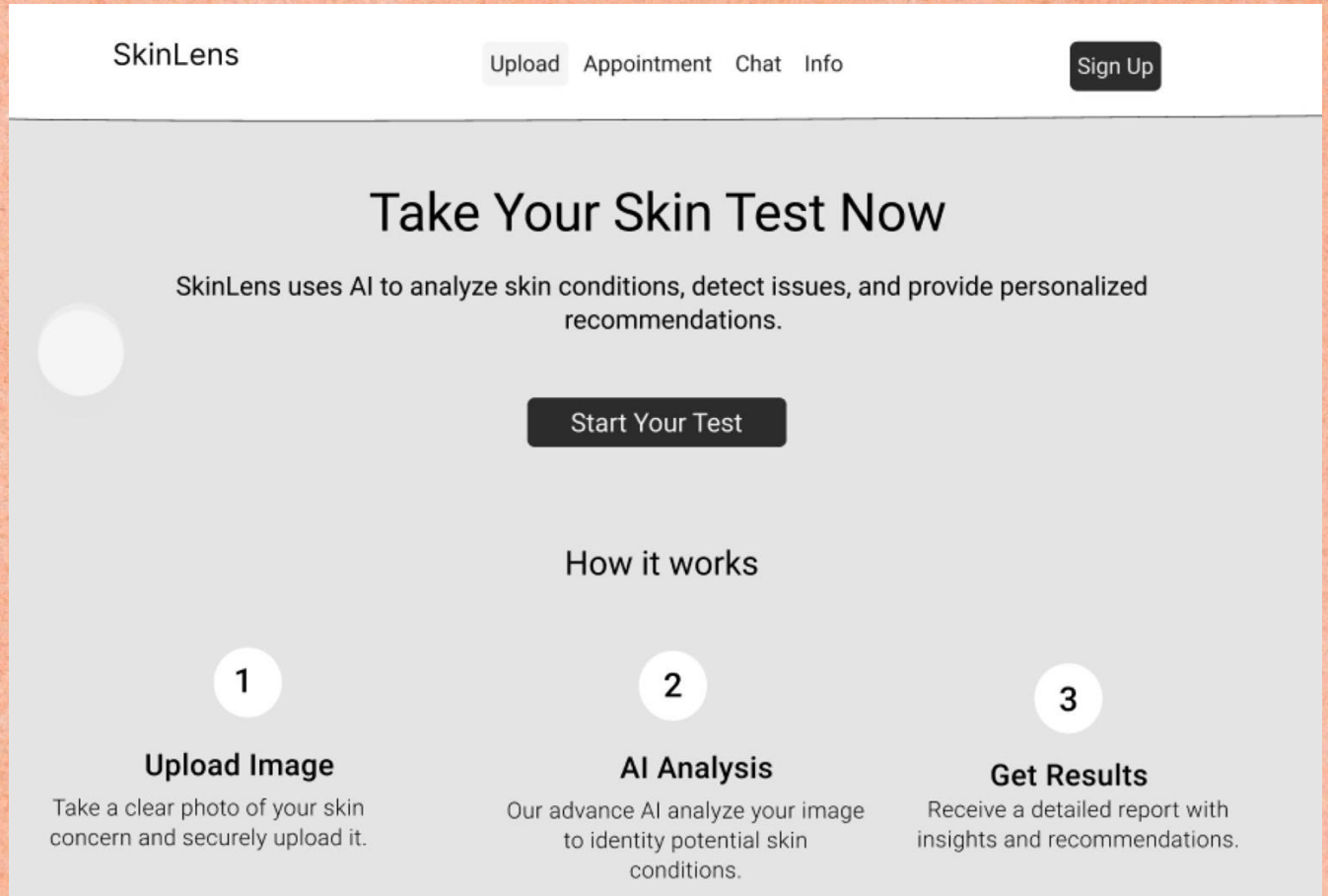
AI-Powered Diagnosis:

- 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.

Project Design (Figma)

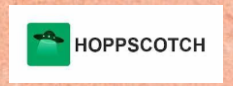


Technologies

Front-end & Back-end & Database

AI Model

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 UIs 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.

Lightweight Python Web Framework: Flask is a micro-framework that allows for easy and flexible development of web applications and APIs.

API development: Used to create RESTful API endpoints for handling requests from the frontend.

Routing: Flask handles routes that correspond to different functionalities (e.g., image upload, retrieving diagnoses).

Integration with AI Models: Flask acts as a middleware between the frontend and the ResNet50 image classification model for handling requests and returning diagnoses.

Easy Integration with Databases: Flask communicates with Firebase (Firestore and Storage) to store and retrieve data such as images and diagnosis reports.

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

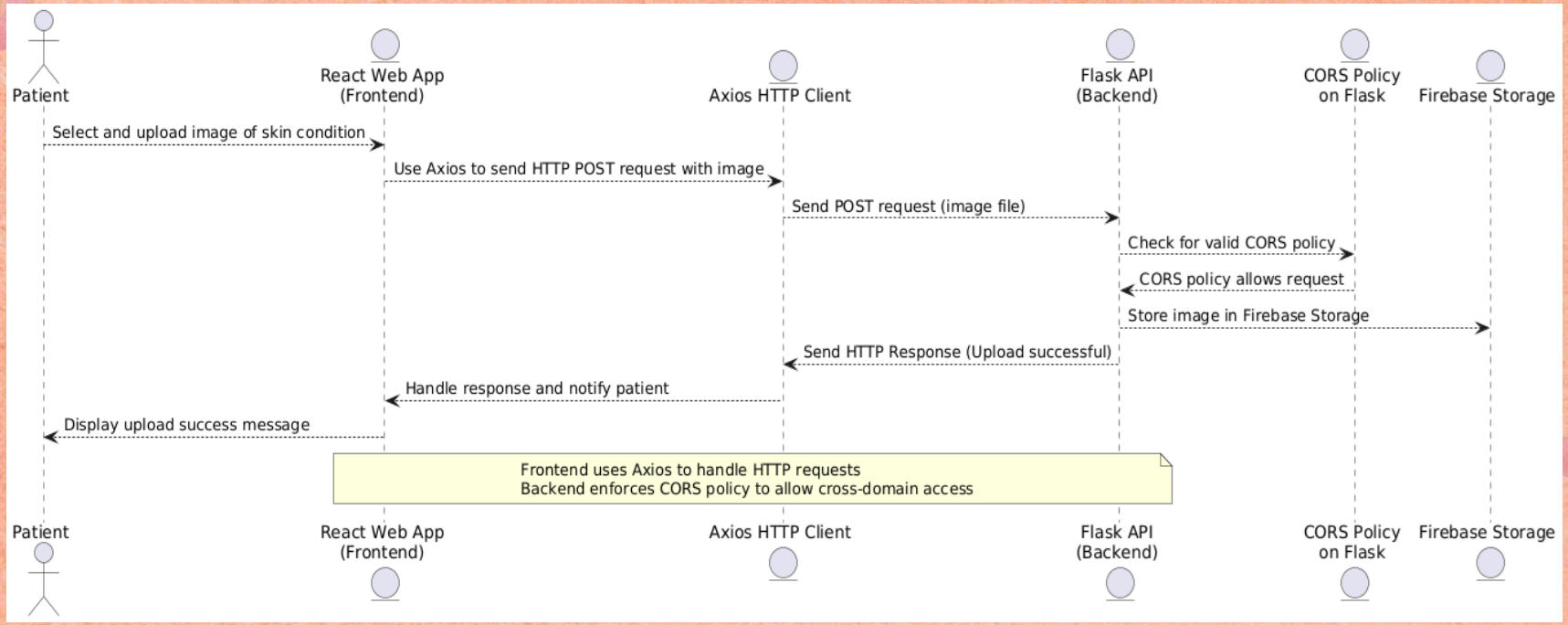
JavaScript HTTP Client: Axios is a promise-based library used to make HTTP requests from the frontend (React).

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:

- **Authentication**:
  - Enables secure login and signup for **patients** and **dermatologists**.
  - Role-based authentication to manage access to dashboards and reports.
- **Firestore Database**:
  - Stores user profiles (name, email, role, UID) securely.
  - Maintains **cases and reports**, including metadata, predictions, descriptions, and treatment recommendations.
- **Firebase Storage**:
  - Securely stores uploaded skin condition images.
  - Organizes images by unique **case IDs** for easy retrieval.
- **Security & Scalability**:
  - Implements Firebase's built-in security rules to ensure **data privacy**.
  - Scales effortlessly to handle multiple users and large datasets.

Why Firebase?

- Simplifies backend management.
- Provides seamless integration for authentication, storage, and real-time updates.
- Ensures high reliability and scalability for a healthcare-focused application.

ALGORITHMS:

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

OpenCv library for image preprocessing.

Matplotlib for visualization.

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:

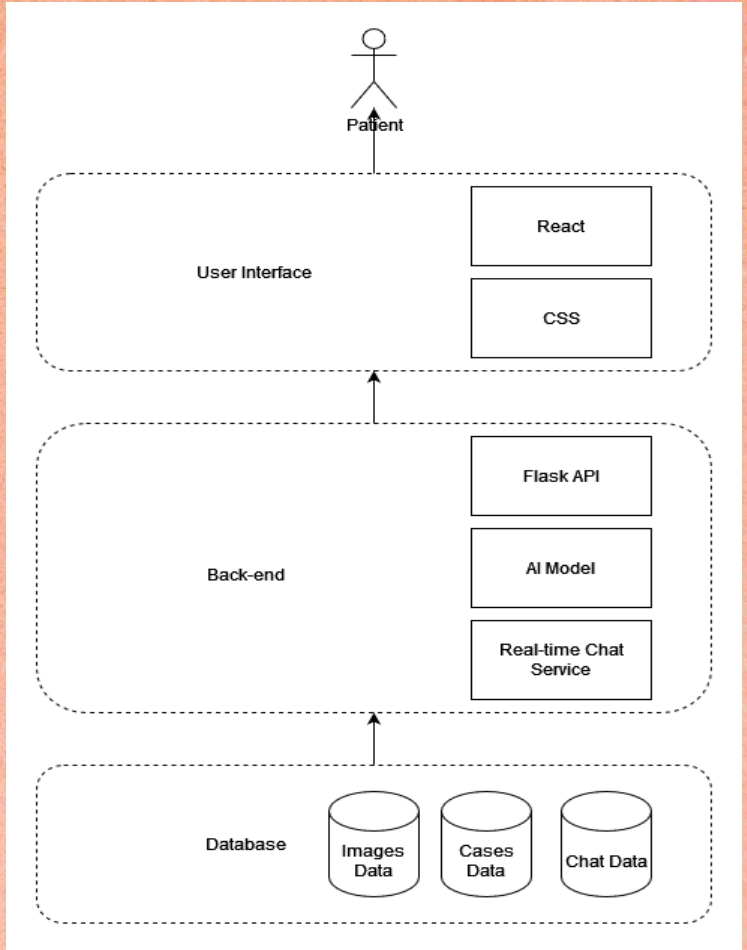
- **Optimizer**: Adam.
- **Loss Function**: Categorical Cross-entropy (for multi-class classification).

Project Schedule

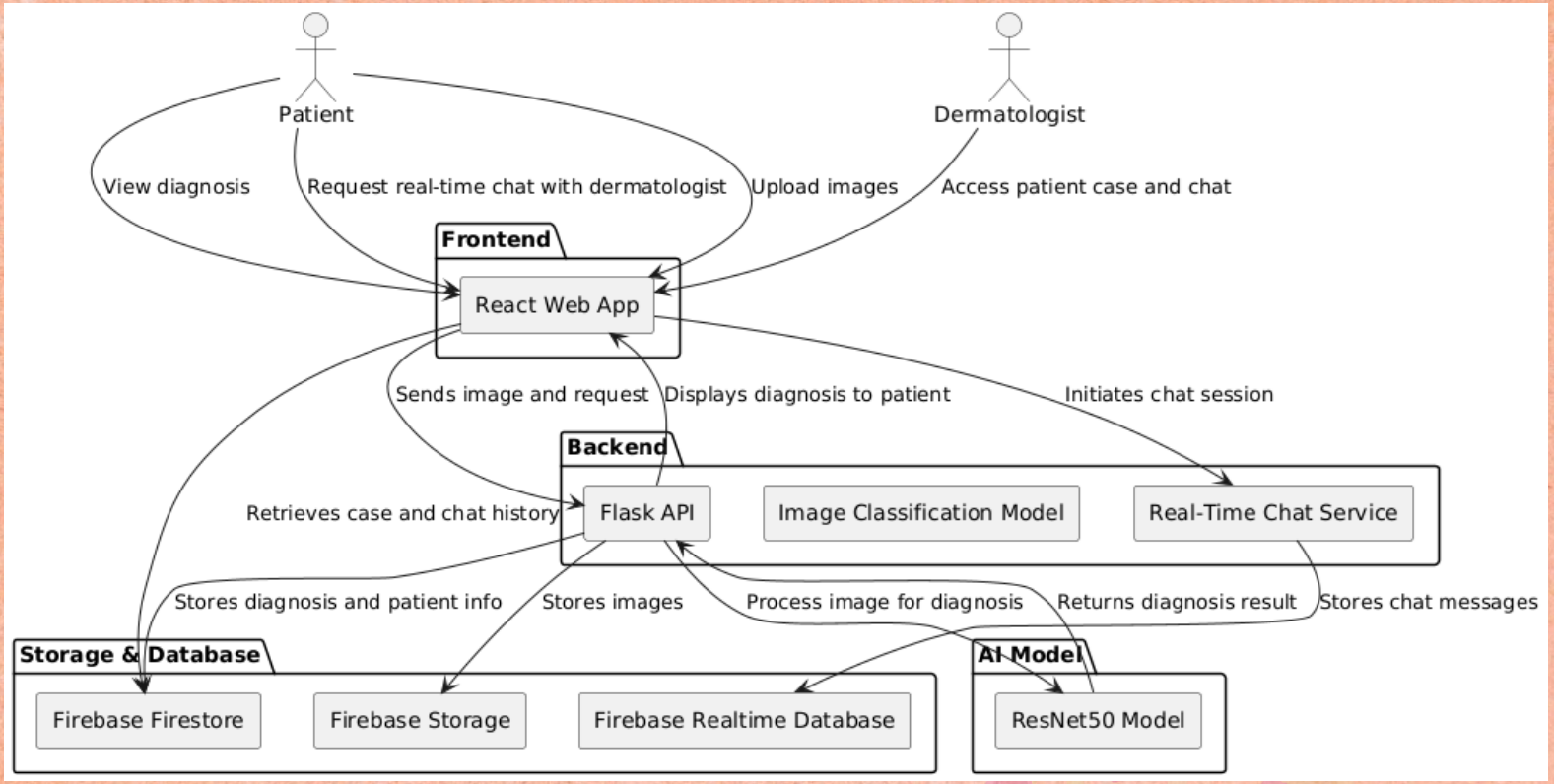
Sprint 0	Sprint 1	Sprint 2	Sprint 3
• Sep 6 - Sep 26	• Sep 26 - Oct 24 • Image upload • AI diagnosis	• Oct 24 - Nov 21 • More conditions • More accuracy • Brief description • Treatments • Report feature • Database	• Nov 21 - Dec 12 • More accuracy • User account management • Derm review • Real-time chat

Diagrams

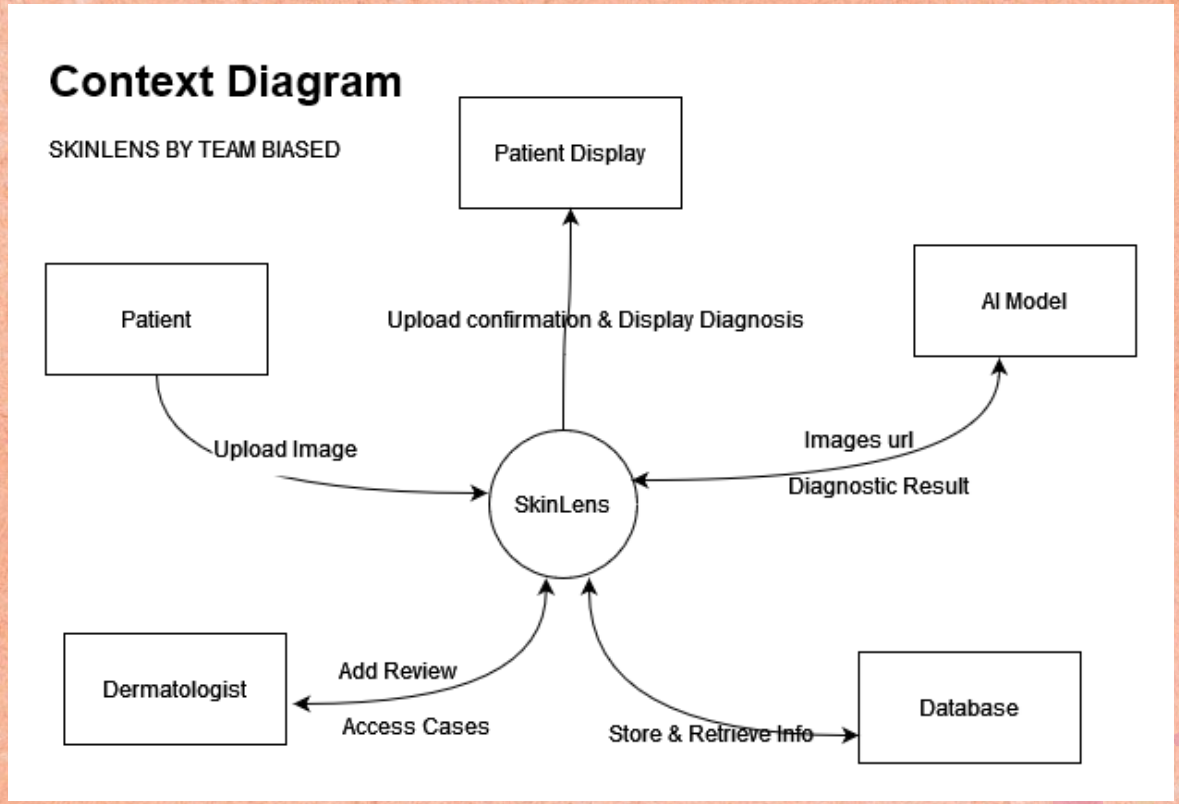
Architecture Diagram



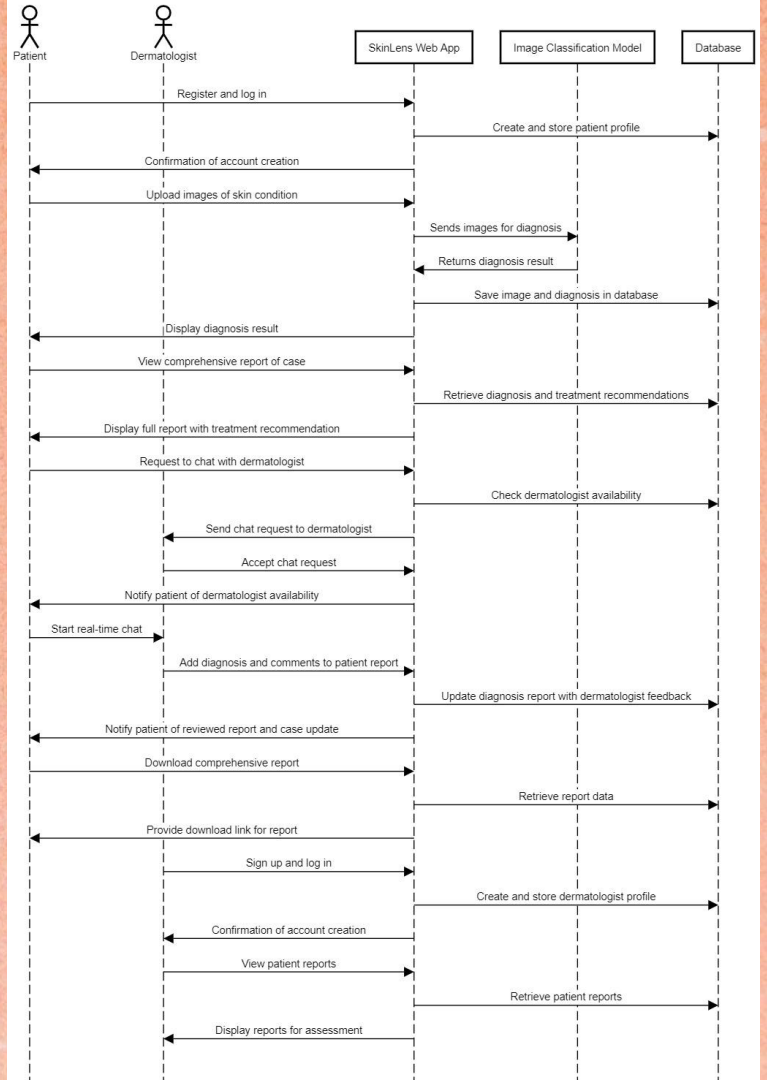
Architecture Diagram



Context Diagram



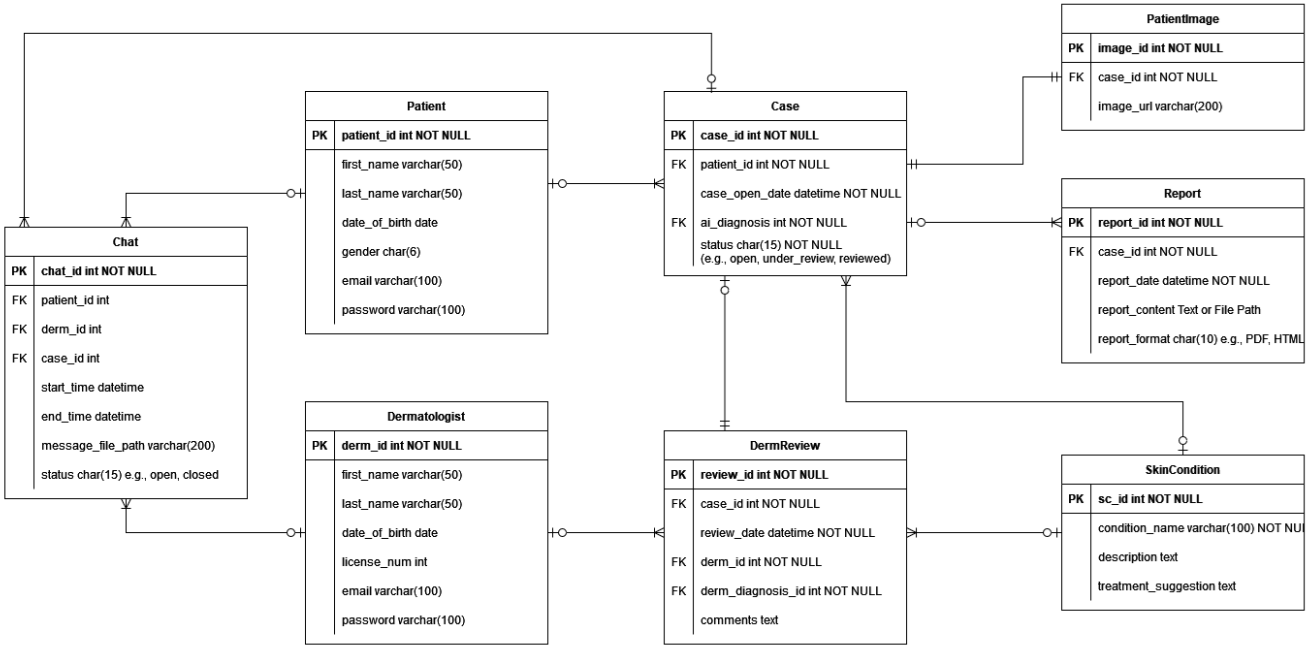
Sequence Diagram



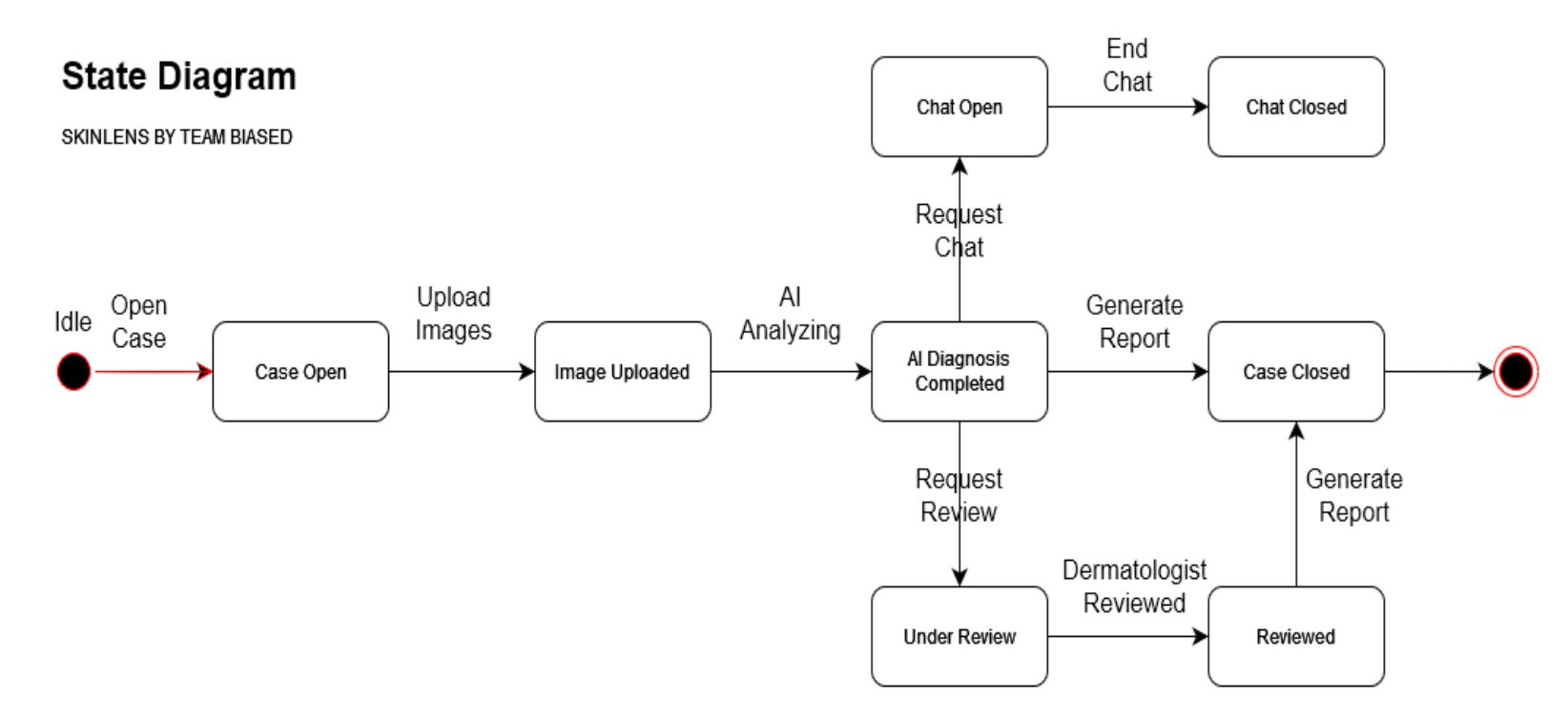
ER Diagram

Entity Relationship Diagram (ERD)

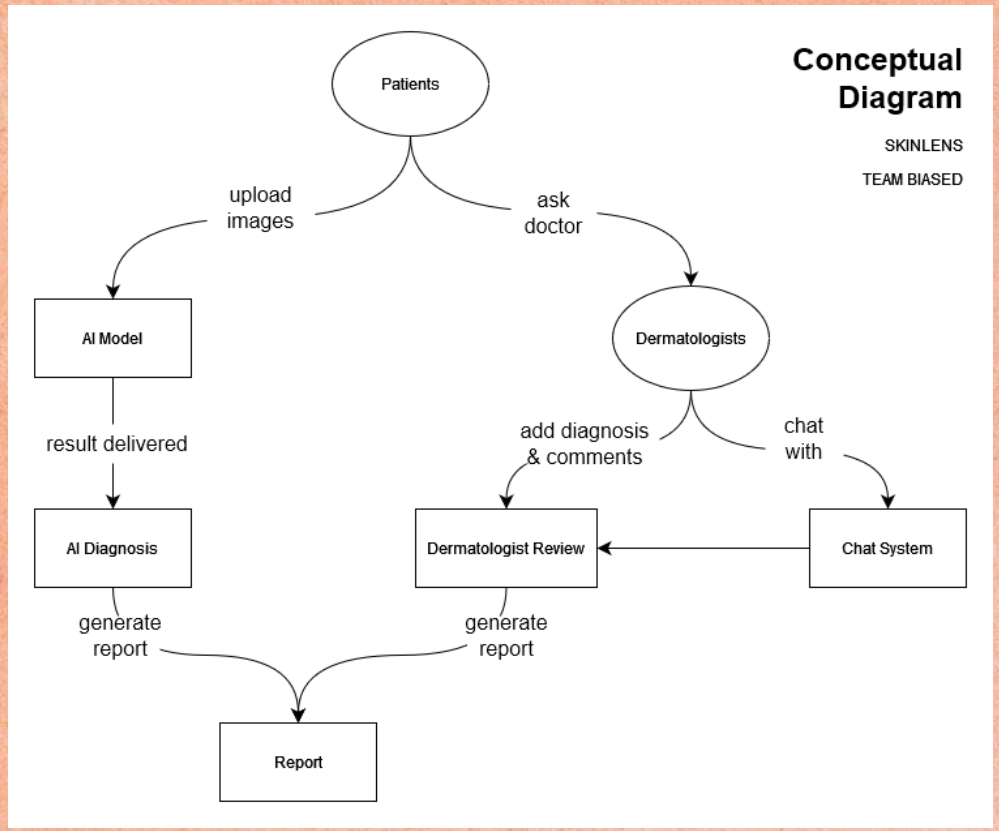
SKINLENS BY TEAM BIASED



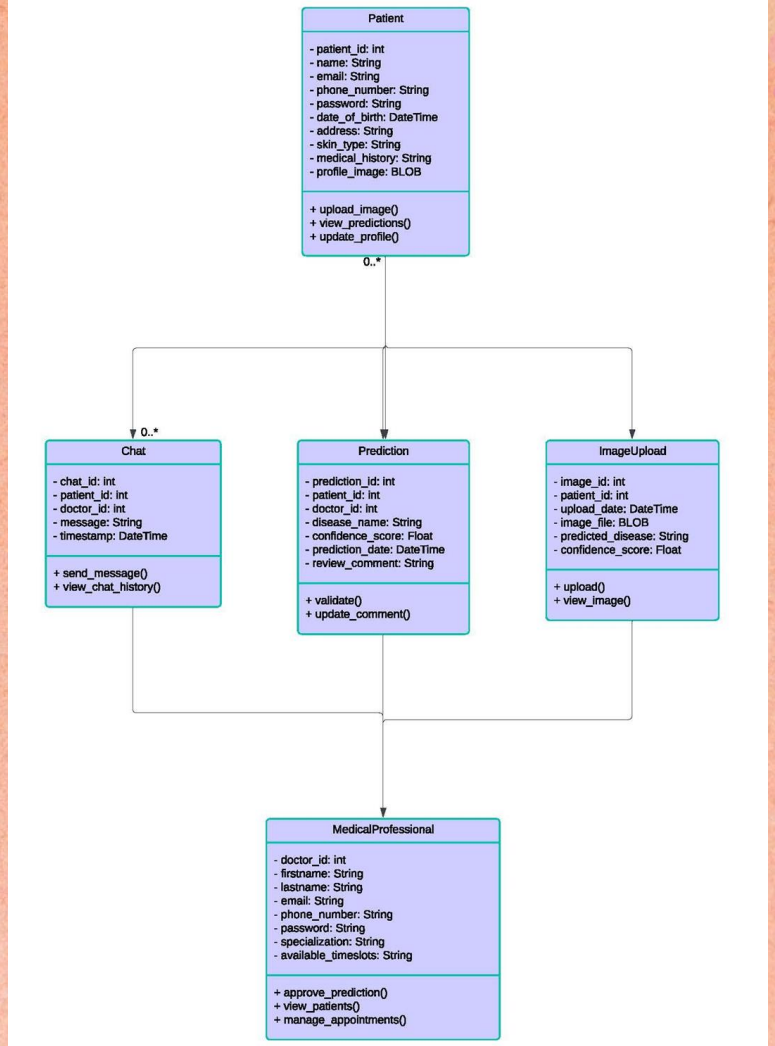
State Diagram



Concept Diagram



Class Diagram



Sprint 1 Summary

- Set up React + VITE front-end server
- Set up Flask back-end server
- Finalized a Figma design
- Created React front-end for homepage
- Created React front-end for image-upload page
- Queried dataset and formatted labels
- Developed the training file and executed it on Colab

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.	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. 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.	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.	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. 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.	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 my diagnosis to patient reports so that I can contribute my professional assessment to the patient's case.	Select Diagnosis: **Given** the dermatologist opens a patient report, **When** they select a diagnosis from the predefined skin condition list, **Then** the selected diagnosis is added to the report. Submit Diagnosis: **Given** the dermatologist enters their diagnosis, **When** they click the "Submit" button, **Then** the report status changes to "Reviewed," and the diagnosis is visible to the patient.	Derm Review	2
US_23	**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	1
US_24	**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

PRODUCT BACKLOG

Sprint 1 | Sprint 2 | Sprint 3

No.	User Stories	Acceptance Criteria	Feature	Story Points
US_25	**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
US_26	**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_27	**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_28	**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

PRODUCT BACKLOG

Sprint 1 | Sprint 2 | Sprint 3

No.	User Stories	Acceptance Criteria	Feature	Story Points
US_29	**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 time stamps and attachments are supported. End Chat Session: **Given** the dermatologist finishes the consultation, **When** they click "End Chat," **Then** the session is closed, and the patient receives a notification.	Chat	4
US_30	**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_31	**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_32	**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

PRODUCT BACKLOG

Sprint 1 | Sprint 2 | Sprint 3

No.	User Stories	Acceptance Criteria	Feature	Story Points
US_33	**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 define user stories and acceptance criteria so that I have a clear understanding of the project requirements and can prioritize tasks.	All user stories are written in the standard format: As a [user role], I want [goal] so that [benefit]. Acceptance criteria are defined for each user story. Story points is defined for each user story. User Stories, Acceptance criteria, and story points are setted in Jira.	2
TU_02	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_03	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_04	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_05	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_06	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 can assist other developers to deploy this application to the production/development environment, and indicate components and tools required to setup up the environments. The installation manual is tested by a team member or tester following the instructions to ensure accuracy.	2
TU_07	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. API documentation is reviewed by developers and testers to ensure clarity and completeness.	2
TU_08	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 2 BACKLOG

US carry over from Sprint 1
Technical User Stories
Normal User Stories for Sprint 2

US_ID	User Stories	Feature	Story Points
US_07	As a patient, I want the uploaded image to be securely stored so that I can access my report anytime.	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.	Image Handling	2
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.	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.	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.	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.	AI Diagnosis	5
US_13	As a patient, I want to receive the diagnosis result as soon as possible so that I can act quickly if needed.	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.	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.	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.	Report	5

SPRINT 2 BACKLOG

US carry over from Sprint 1
Technical User Stories
Normal User Stories for Sprint 2

US_ID	User Stories	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..	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.	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.	Report Security and Privacy	2
US_30	As a patient, I want to sign up an account so that I can store my basic information and access personalized services.	User Account Management	2
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_32	As a patient, I want to log in to my account so that I can manage my cases and view diagnoses.	User Account Management	1
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
TU_04	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.	Technical	3
TU_05	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.	Technical	1

SPRINT 2 STORIES WITH TASK

Task	User Stories	Assignee
US_07	As a patient, I want the uploaded image to be securely stored so that I can access my report anytime.	
TK_01	Configure Firebase Storage to organize images by patient ID and case ID.	zaid
TK_02	Save image metadata (URL, case ID, timestamp, etc.) to Firestore.	zaid
TK_03	Develop functionality to retrieve and display stored images on the patient's dashboard	zaid
US_08	As a guest user, I want my uploaded image to be deleted after processing so that it doesn't remain stored unnecessarily.	
TK_04	Use temp/ directory for guest user images in Firebase Storage.	zaid
TK_05	Schedule automatic deletion of images 5 minutes after upload using a setTimeout in the frontend or Firebase Cloud Function.	zaid
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.	
TK_06	Implement checks for image validity after timeout.	zaid
TK_07	Provide alerts and re-upload options if the image is unavailable.	Pujitha
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.	

SPRINT 2 STORIES WITH TASK

Task	User Stories	Assignee
TK_08	Add validation for file types and sizes during upload.	zaid
TK_09	Display clear error messages for failed uploads or predictions.	zaid
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.	
TK_10	Create a dataset	Lesley
TK_11	Update the AI model to predict 20+ skin conditions	Abhi
TK_12	Map predictions to readable condition names in the backend.	Tanzil
US_12	As a patient, I want the diagnosis accuracy to be improved so that I can trust the AI’s results.	
TK_13	Retrain the AI model with additional datasets.	Tanzil
TK_14	Test the model to ensure >80% accuracy for common conditions.	Abhi
TK_15	Replace the existing model in the backend with the optimized version.	zaid
US_13	As a patient, I want to receive the diagnosis result as soon as possible so that I can act quickly if needed.	

SPRINT 2 STORIES WITH TASK

Task	User Stories	Assignee
TK_16	Deploy the model to cloud infrastructure with GPU support for faster processing	Abhi
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.	
TK_17	Create a dataset of symptoms and causes	Bangling
TK_18	Store condition descriptions (symptoms, causes) in Firestore	Zaid
TK_19	Fetch and display descriptions alongside diagnosis results.	Zaid
US_15	As a patient, I want to receive treatment recommendations so that I can take action towards treating my skin condition.	
TK_20	Create a dataset of recommended treatments	Bangling
TK_21	Store treatment recommendations for each condition in Firestore.	Zaid
TK_22	Fetch and display recommendations alongside diagnosis results.	Zaid
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.	
TK_23	Create a report modal	Pujitha
TK_24	Fetch and display all required data in the report.	Zaid

SPRINT 2 STORIES WITH TASK

Task	User Stories	Assignee
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..	
TK_25	Ensure the downloadable report matches the layout of the web report.	Zaid
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.	
TK_26	Implement a "Print" button that uses window.print() to print reports.	Pujitha
US_19	As a patient, I want my reports to be secure and accessible only by me so that my medical data remains private.	
TK_27	Restrict report access to authenticated users using Firebase Authentication.	Zaid
TK_28	Use Firestore security rules and encryption to protect patient data	Zaid
US_30	As a patient, I want to sign up an account so that I can store my basic information and access personalized services.	
TK_29	Create a signup form and integrate Firebase Authentication to register patients, saving their details (name, email, UID) in Firestore.	zaid
TK_30	Redirect successfully registered patients to their dashboard and display success or error messages.	zaid
US_31	As a dermatologist, I want to sign up so that I can access patient reports and provide professional feedback.	
TK_31	Extend the signup form to allow dermatologist registration using Firebase Authentication and save their details in Firestore with the role as "dermatologist."	zaid

SPRINT 2 STORIES WITH TASK

TK_32	Redirect successfully registered dermatologists to their dashboard and display appropriate messages.	zaid
US_32	As a patient, I want to log in to my account so that I can manage my cases and view diagnoses.	
TK_33	Implement a login form and validate patient credentials using Firebase Authentication, fetching role-based data from Firestore.	zaid
TK_34	Redirect patients to their dashboard upon successful login and display relevant error messages if login fails.	zaid
US_33	As a dermatologist, I want to log in to my account so that I can manage patient reports and update case statuses.	
TK_36	Implement role-based authentication for dermatologists, validating credentials and retrieving their data from Firestore.	zaid
TK_37	Redirect logged-in dermatologists to their dashboard to manage patient reports and display success/error feedback.	zaid
TU_04	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.	lesley/zaid
TU_05	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.	tanzil

SPRINT 2 TEST CASES

US_07 As a patient, I want the uploaded image to be securely stored so that I can access my report anytime.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_07_01	Verify Secure Storage of Uploaded Image	1. Upload an image. 2. Check the storage backend for the image file. 3. Verify the image is stored in the user's folder.	The image is stored securely in a user-specific folder and accessible only by the user.	Pass	Nov 20	Tanzil
TC_07_02	Verify Image Accessibility in Reports after downloading	1. Open the report. 2. Check if the uploaded image is visible in the report after downloading	The uploaded image is displayed in the user’s report after downloading	Fail	Nov 20	Tanzil

US_08 As a guest user, I want my uploaded image to be deleted after processing so that it doesn’t remain stored unnecessarily.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_08_01	Verify Temporary Image Deletion for Guest Users	1. Upload an image. 2. Wait for processing to complete. 3. Check storage for the uploaded image.	The uploaded image is deleted after processing.	Pass	Nov 20	Tanzil
TC_08_02	Verify Guest User's Image Is Not Retained	1. Upload an image. 2. Wait for session expiry. 3. Check storage backend for the image.	The image is no longer stored in the system	Pass	Nov 20	Tanzil

SPRINT 2 TEST CASES

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.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_09_01	Verify Alert for Missing Temporary Image	1. Attempt to retrieve a deleted image. 2. Observe the alert message.	An alert message is displayed: "The image is no longer available. Please re-upload."	Pass	Nov 20	Tanzil

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.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_10_01	Verify File Type and Size Validation	1. Attempt to upload an invalid file (e.g., PDF or a file >10MB). 2. Observe the validation message.	An error like "Invalid file type or size" is displayed.	Pass	Nov 20	Tanzil
TC_10_02	Verify Clear Error Messages for Failed Uploads	1. Simulate an upload error (e.g., network disconnection). 2. Observe the error message.	An error like "Upload failed. Please check your network connection." is displayed.	Pass	Nov 20	Tanzil

SPRINT 2 TEST CASES

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.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_11_01	Verify Wide Range of Skin Conditions	1. Upload images for various skin conditions. 2. Observe the diagnosis results.	The system recognizes all supported skin conditions.	Pass	Nov 20	Tanzil

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	Verify Model Retraining Improves Accuracy	1. Test the retrained model with a validation dataset. 2. Compare accuracy before and after retraining.	Accuracy improves after retraining.	Fail	Nov 20	Tanzil

US_13 As a patient, I want to receive the diagnosis result as soon as possible so that I can act quickly if needed.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_13_01	Verify Quick Diagnosis Time	1. Upload an image. 2. Measure the time taken for the diagnosis result.	The result is displayed within the expected time limit (e.g., 1 minute).	Pass	Nov 20	Tanzil

SPRINT 2 TEST CASES

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.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_14_01	Verify Description Display for Diagnosed Condition	1. Upload a valid skin condition image. 2. Click "Predict Diagnosis". 3. Check the "Description" section.	A brief description of the diagnosed condition is displayed.	Pass	Nov 20	Tanzil

US_15 As a patient, I want to receive treatment recommendations so that I can take action towards treating my skin condition.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_15_01	Verify Treatment Recommendations Display	1. Upload a valid image. 2. Click "Predict Diagnosis". 3. Check the "Treatment Recommendations" section.	1. Upload a valid image. 2. Click "Predict Diagnosis". 3. Check the "Treatment Recommendations" section.	Pass	Nov 20	Tanzil

SPRINT 2 TEST CASES

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.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_16_01	Verify Report Content for a Case	1. Navigate to the reports section. 2. Open a specific case report. 3. Verify the report content.	The report includes details such as the uploaded image, diagnosis, description, treatment recommendations, and timestamps.	Pass	Nov 20	Tanzil

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	Verify PDF Download Functionality	1. Open a case report. 2. Click the "Download PDF" button. 3. Verify the downloaded file.	The patient is directed to their dashboard.	Fail	Nov 20	Tanzil

SPRINT 2 TEST CASES

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.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_18_01	Verify Print Functionality	1. Open a case report. 2. Click the "Print" button. 3. Verify the printer output.	The report is printed successfully.	Pass	Nov 20	Tanzil

US_19 As a patient, I want my reports to be secure and accessible only by me so that my medical data remains private.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_19_01	Verify Personal Details in Report	1. Open a case report. 2. Check for the user’s name, email, and case details.	The report includes the user’s personal details such as name, email, and timestamp.	Pass	Nov 20	Tanzil

SPRINT 2 TEST CASES

US_30 As a patient, I want to sign up an account so that I can store my basic information and access personalized services.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_30_01	Verify Patient Sign-Up Functionality	1. Navigate to the sign-up page. 2. Enter valid details (e.g., name, email, password). 3. Click "Sign Up". 4. Verify account creation.	The account is created successfully, and the patient is redirected to the home page.	Pass	Nov 20	Tanzil
TC_30_02	Handle Sign-Up with Existing Email	1. Navigate to the sign-up page. 2. Enter an already registered email. 3. Click "Sign Up". 4. Observe the error message.	An error message "Email already in use" is displayed.	Pass	Nov 20	Tanzil

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 Dermatologist Sign-Up Functionality	1. Navigate to the dermatologist sign-up page. 2. Enter valid details (e.g., name, email, password). 3. Click "Sign Up". 4. Verify account creation.	The account is created successfully, and the dermatologist is redirected to the dashboard.	Fail	Nov 20	Tanzil

SPRINT 2 TEST CASES

US_32 As a patient, I want to log in to my account so that I can manage my cases and view diagnoses.

TC	TC Title	Step Description	Expected Results	Execution Status	Execution date	Tester
TC_32_01	Verify Patient Sign-Up Functionality	1. Navigate to the sign-up page. 2. Enter valid details (e.g., name, email, password). 3. Click "Sign Up". 4. Verify account creation.	The account is created successfully, and the patient is redirected to the home page.	Pass	Nov 20	Tanzil
TC_32_02	Handle Invalid Login Credentials	1. Navigate to the login page. 2. Enter an invalid email or password. 3. Click "Log In". 4. Observe the error message.	An error message like "Invalid email or password" is displayed.	Pass	Nov 20	Tanzil

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 Dermatologist Login Functionality	1. Navigate to the login page. 2. Enter valid email and password. 3. Click "Log In". 4. Verify successful login.	The dermatologist is logged in successfully and redirected to the dashboard.	Fail	Nov 20	Tanzil

SPRINT 2 STORIES COMPLETED AND NOT

US_ID	User Stories	Feature	Story Points	Completed
US_07	As a patient, I want the uploaded image to be securely stored so that I can access my report anytime.	Image Handling	3	Y
US_08	As a guest user, I want my uploaded image to be deleted after processing so that it doesn't remain stored unnecessarily.	Image Handling	2	Y
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.	Image Handling	1	Y
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.	Image Handling	1	Y
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.	AI Diagnosis	5	Y
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	N
US_13	As a patient, I want to receive the diagnosis result as soon as possible so that I can act quickly if needed.	AI Diagnosis	2	Y
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.	AI Diagnosis	2	Y
US_15	As a patient, I want to receive treatment recommendations so that I can take action towards treating my skin condition.	AI Diagnosis	2	Y
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.	Report	5	Y

SPRINT 2 STORIES COMPLETED AND NOT

US_ID	User Stories	Feature	Story Points	Completed
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_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.	Report	1	Y
US_19	As a patient, I want my reports to be secure and accessible only by me so that my medical data remains private.	Report Security and Privacy	2	Y
US_30	As a patient, I want to sign up an account so that I can store my basic information and access personalized services.	User Account Management	2	Y
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	N
US_32	As a patient, I want to log in to my account so that I can manage my cases and view diagnoses.	User Account Management	1	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	N
TU_04	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.	Technical	3	Y
TU_05	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.	Technical	1	Y

METRICS - TEAM VELOCITY

Sprint 2 Velocity

Total Story Points: 42

Completed Story Points: 33

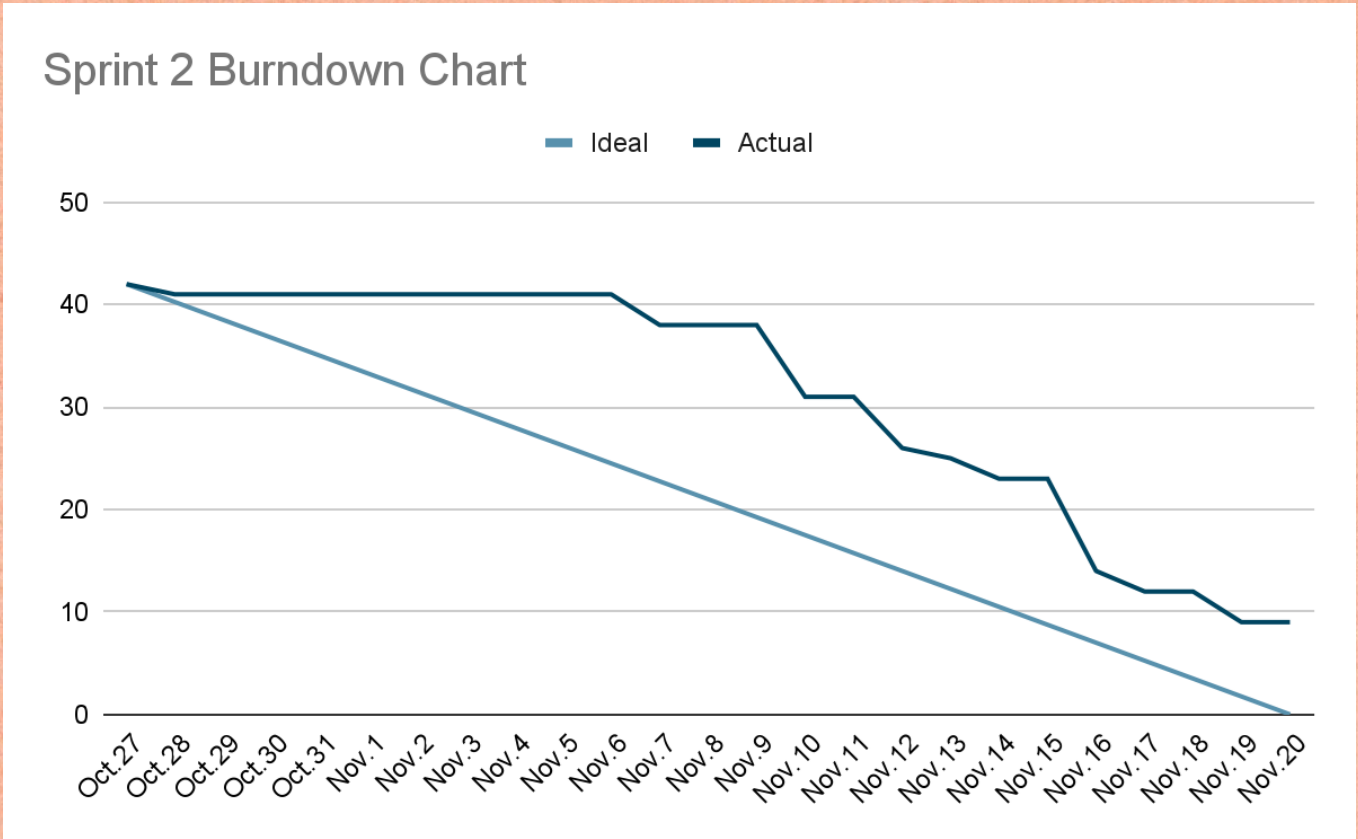
Velocity:33 story points

METRICS - Team's historical velocity (average)

The average of the previous sprints' velocities:

31 story points

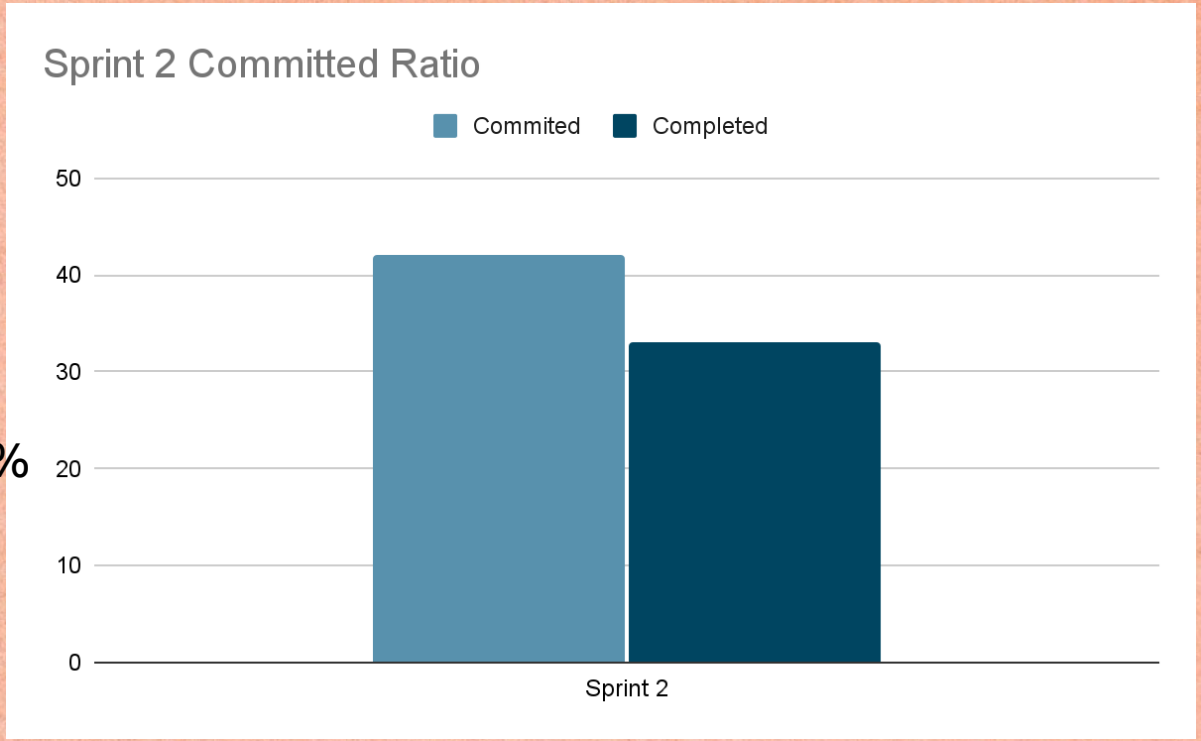
METRICS - BURNDOWN CHART



METRICS - COMMITTED RATIO

The committed ratio of Sprint 2 is 79%.

The average of the previous sprints' ratio: 83%



Sprint 2 Retrospective

Few important points from the Retrospective

What did we do right?

- Finding sources and preparing a dataset was one hell of a task, and we did it
- good team communication for resolving issues
- Finding sources and preparing a dataset was one hell of a task, and we did it
- Deadlines were set 1-2 days prior to the actually deadlines because of this we had time to encounter last minute issues
- good atmosphere to learn, work and communicate. team members are patient and nice.
- we began to code as soon as the planning meeting was done, giving enough development time to work on tasks

Few important points from the Retrospective

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

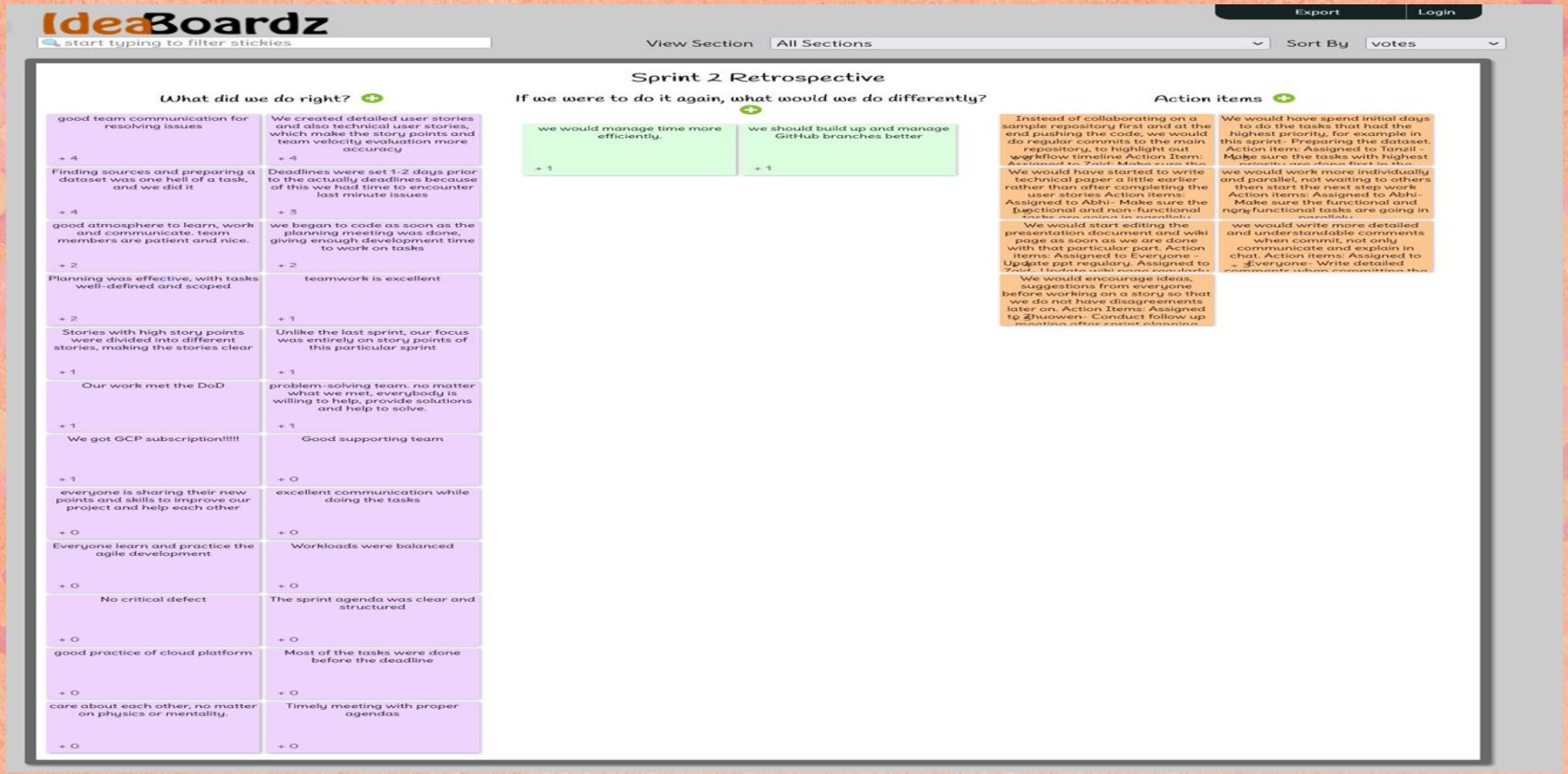
- Instead of collaborating on a sample repository first and at the end pushing the code, we would do regular commits to the main repository, to highlight out workflow timeline
- We would have started to write technical paper a little earlier rather than after completing the user stories
- We would have spend initial days to do the tasks that had the highest priority, for example in this sprint- Preparing the dataset.
- we would work more individually and parallel, not waiting to others Then start the next step work
- We would start editing the presentation document and wiki page as soon as we are done with that particular part.

Few important points from the Retrospective

Action items

- Instead of collaborating on a sample repository first and at the end pushing the code, we would do regular commits to the main repository, to highlight out workflow timeline Action Item: Assigned to Zaid: Make sure the code is on the main repository.
- We would have started to write technical paper a little earlier rather than after completing the user stories Action items: Assigned to Abhi- Make sure the functional and non-functional tasks are going in parallely
- We would have spend initial days to do the tasks that had the highest priority, for example in this sprint- Preparing the dataset. Action item: Assigned to Tanzil - Make sure the tasks with highest priority are done first in the planning meeting
- we would work more individually and parallel, not waiting to others Then start the next step work Action items: Assigned to Abhi- Make sure the functional and non-functional tasks are going in parallely
- We would start editing the presentation document and wiki page as soon as we are done with that particular part. Action items: Assigned to Everyone - Update ppt regulary. Assigned to Zaid- Update wiki page regularly

SPRINT 2 RETROSPECTIVE



SPRINT 3 – PLANNING AND BACKLOG

Sprint 3 User Stories
Technical user stories
US Carry Over From Sprint 2

US_ID	User Stories	Feature	Story Points	Completed	Date
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 my diagnosis to patient reports so that I can contribute my professional assessment to the patient's case.	Derm Review	2		
US_23	As a dermatologist, I want to add comments to patient reports so that I can provide addditional guidance if necessary.	Derm Review	1		

SPRINT 3 – PLANNING AND BACKLOG

Sprint 3 User Stories
Technical user stories
US Carry Over From Sprint 2

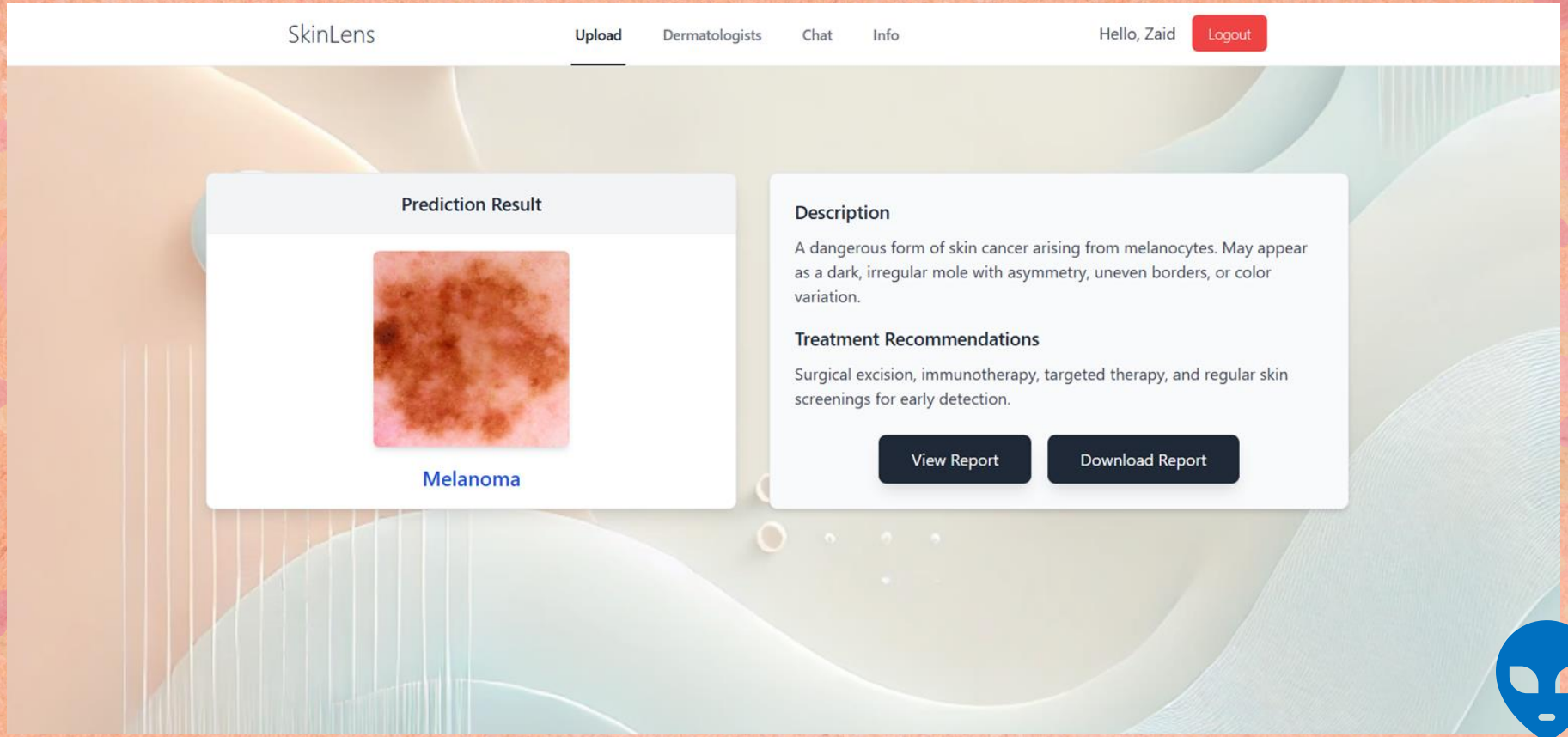
US_ID	User Stories	Feature	Story Points	Completed	Date
US_24	As a patient, I want to be notified as soon as my report is reviewed by a dermatologist so that I can stay informed about my case status.	Derm Review	2		
US_25	As a patient, I want to access a chat interface from my report page so that I can communicate directly with a dermatologist.	Chat	2		
US_26	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_27	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_28	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_29	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		

SPRINT 3 - TECHNICAL USER STORIES

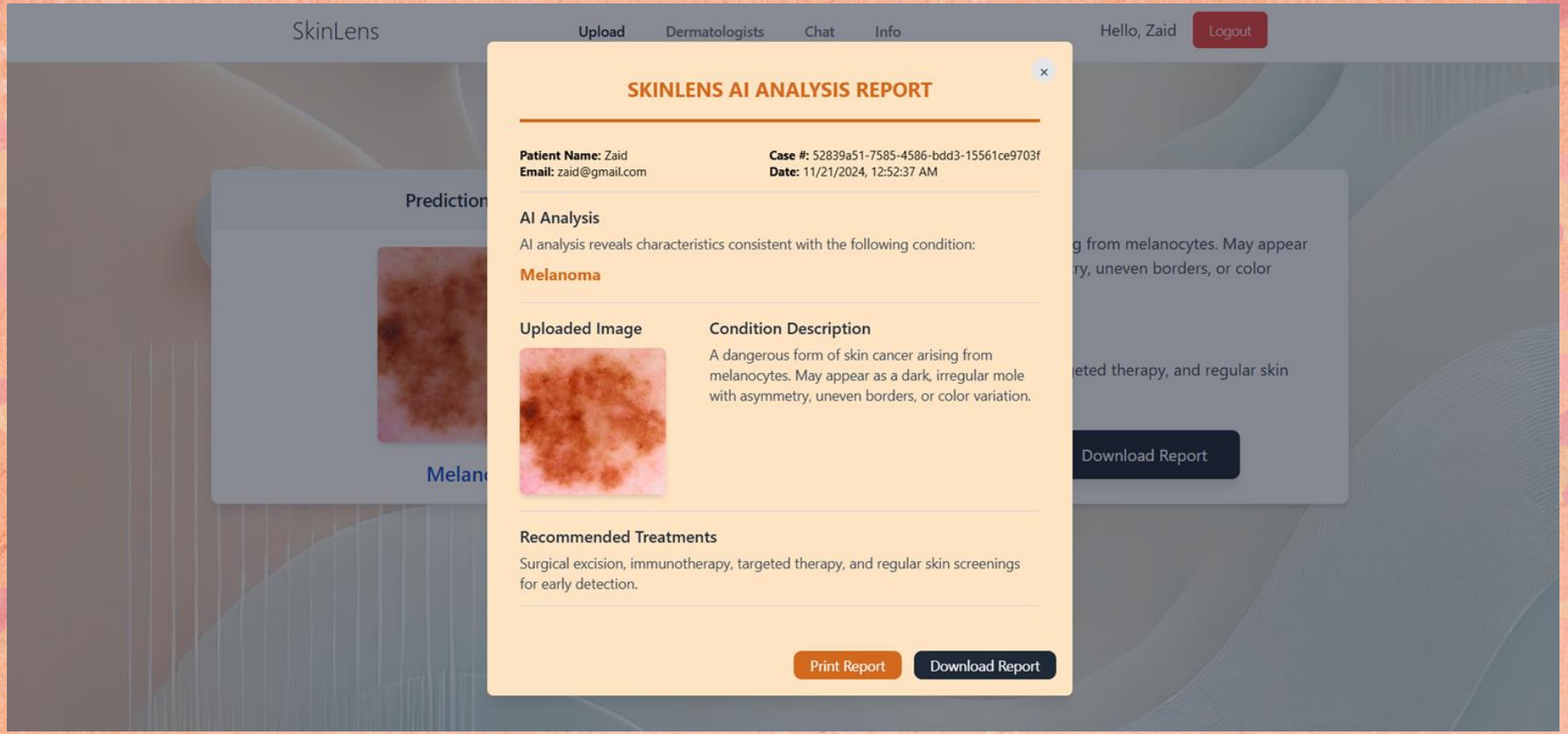
Sprint 3 User Stories
Technical user stories
US Carry Over From Sprint 2

US_ID	User Stories	Feature	Story Points	Completed	Date
TU_06	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_07	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_08	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		

PROJECT DEMO - SCREENSHOT



PROJECT DEMO – SCREENSHOT



PROJECT - APIs

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**:
    - **Input**: Image file selected by the user.
    - **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. AI Prediction API

- **Endpoint**: /predict
- **Purpose**: Sends the uploaded image URL to the backend AI model for processing and returns a predicted diagnosis.
- **Method**: POST
- **Input**:
  - imageUrl (String): The URL of the uploaded image.
- **Output**:
  - prediction (String): The predicted skin condition.
  - Example response: { "prediction": "Melanoma" }

Key Code:

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

PROJECT - APIs

3. Fetch Disease Details API

- **Purpose**: Fetches description and treatment recommendations for a specific skin condition from Firestore.
- **Method**: Firebase Firestore Query
- **Input**:
  - predictedDisease (String): The name of the predicted skin condition.
- **Output**:
  - description (String): A brief description of the condition.
  - treatment (String): Recommended treatment.

Key Code:

```
const diseasesRef = collection(db, "diseases");
const q = query(diseasesRef, where("disease", "==", predictedDisease));
const querySnapshot = await getDocs(q);
```

PROJECT – APIs

4. Save Case Details API

- **Purpose**: Saves case details (uploaded image, prediction, description, and treatment) in Firestore for logged-in users.
- **Method**: Firebase Firestore setDoc
- **Input**: Case details (JSON)
- **Output**: None

Key Code:

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

PROJECT - APIs

5. Retrieve Case Details API

- **Purpose**: Retrieves case details for a logged-in user, including image URL, prediction, description, and treatment.
- **Method**: Firebase Firestore Query
- **Input**:
  - caseId (String): Unique identifier for the case.
- **Output**: Case details (JSON)

Key Code:

```
const q = query(collection(db, "cases"), where("caseId", "==", caseId));
 const querySnapshot = await getDocs(q);
```

PROJECT - APIs

6. Temporary Image Deletion API

- **Purpose**: Deletes temporary images for guest users after a certain timeout period.
- **Method**: Firebase Storage deleteObject
- **Input**:
  - imageUrl (String): URL of the uploaded image.
- **Output**: None.

Key Code:

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

PROJECT - APIs

7. Authentication APIs

Sign-Up API

- **Purpose**: Creates a new user account and stores user details (name, role, email) in Firestore.
- **Method**: Firebase Authentication + Firestore setDoc

Key Code:

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

Login API

- **Purpose**: Logs in a user with email and password.
- **Method**: Firebase Authentication signInWithEmailAndPassword

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

PROJECT – APIs

Flask API and CORS

```python
# 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

```javascript
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);
  }
};
```

WIKI PAGE LINK

To see our progress

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

DEMO

THANK YOU!