MCA202- Software Engineering Lab

Trigger Points

MCA202 Software Engineering Project Report

Mid-Semester Evaluation

Submitted by:

Ashish Bansal (2024010021)

Ashish (2024010022)

Bikkramjit Singh (2024010028)

MCA First Year

Submitted to: DR. Ashima Singh

Department of Computer Science and Engineering

Thapar Institute of Engineering and Technology, Patiala

May 2025

TABLE OF CONTENTS

Diagrams Index

1. Planning phase

1.1 Project Write up

Need of This Project and Executive summary:

Trigger Points is an innovative project aimed at providing a scientifically validated, drug-free alternative for stress relief and pain management. It leverages AI-driven image recognition to identify and visualize key acupressure points on the hand. The system serves as an accessible tool for individuals seeking natural healing solutions by accurately detecting and suggesting pressure points to alleviate stress, headaches, and muscle tension. By integrating advanced computer vision techniques, this project offers an intelligent, user- friendly, and interactive approach to acupressure therapy. It empowers users with self-care techniques and fosters holistic well-being by making traditional healing methods more accessible and precise.

Solution:

The Acupressure Hand Diagnostic System offers a real-time interactive platform where users can:

- Detect and visualize their hand using live camera input, leveraging **MediaPipe** and **OpenCV** for accurate hand tracking and landmark recognition.
- Select from a list of predefined health conditions, each linked to a specific **acupressure point**, which is shown clearly on the user's hand in real time.
- Interact through an intuitive frontend built with **HTML**, **CSS**, and **JavaScript**, allowing smooth navigation and responsive user experience.
- Authenticate and manage user access using **Flask** and **CouchDB**, where **user data** is stored securely for login and registration purposes.

Functional Requirements of the Project

a) Core Features:

- **Hand Recognition:**
  The system uses OpenCV and MediaPipe to detect and track the user's hand in real-time via the device's ca4mera.

- **Point Identification:**
  Based on the selected health condition, the system maps and visualizes corresponding acupressure points on the user's hand using landmark detection.
- **User Guidance:**
  Visual cues and on-screen indicators help guide the user to correctly position their hand and identify the exact acupressure points.

b) User Management:

- **User Registration:**
  New users can create accounts to personalize their experience and save preferences.
- **Login and Logout Functionality:**
  Secure authentication allows users to log in and out, maintaining their session and data.

Quality Attributes

a) Performance:

- The detection and point mapping process must be highly responsive. Real-time hand recognition and acupressure point display should complete within a few seconds, ensuring a seamless experience.
- To optimize performance, the system is designed to detect and process only one hand at a time. This limitation ensures faster frame processing, reduces computational load, and supports real-time responsiveness, especially on mobile and low-power devices.

b) Usability:

- The interface should be clean, intuitive, and easy to use for all age groups. It must be optimized for both desktop and mobile browsers to support accessibility across devices.

c) Reusability:

- The OpenCV and MediaPipe based hand detection and visualization module should be modular and reusable. It should be adaptable for integration into other health and wellness applications in the future, promoting scalability and maintainability.

2. Requirement analysis Phase

2.1 Use Case Template

Use-Case Name:

- Identify and Display Acupressure Points

Actors:

- Primary User:End-user scanning hand
- System: Hand recognition with mediapipe and opencv

Preconditions:

- User has access to a camera-enabled device.
- The application is installed and running properly.

Basic Flow:

1 Users can scan hand Using device camera.
2. System identifiesHand landmarks and maps acupressure points.
3. Relevant points are display and with names, locations.
4. User selects points to view more detailed information

Postconditions:

- User gains knowledge of acupressure points and their uses.
- Points are displayed in real time.

2.2 Use Case Diagram

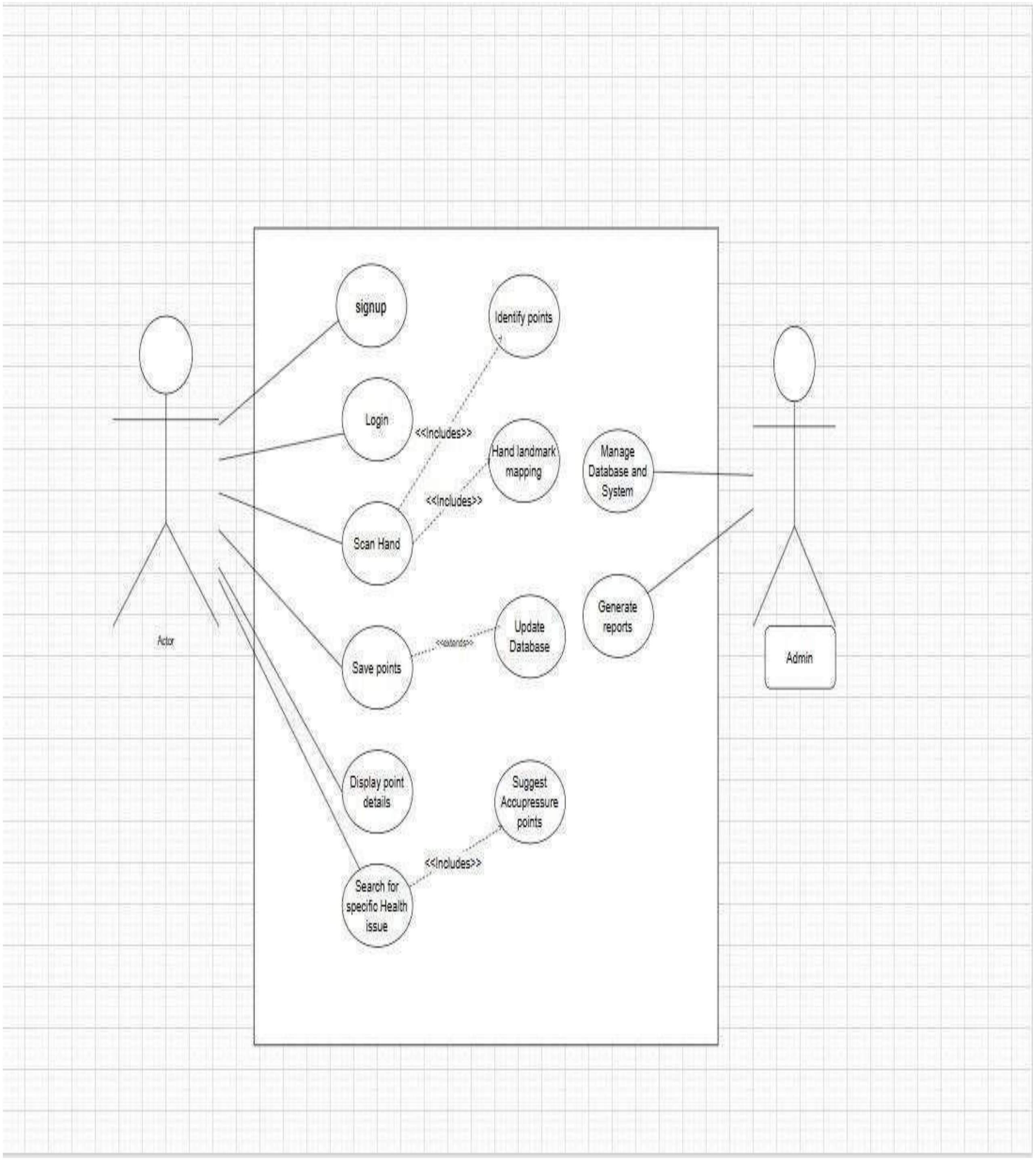


Fig.1

2.3 Data Flow Diagram Level 0 DFD

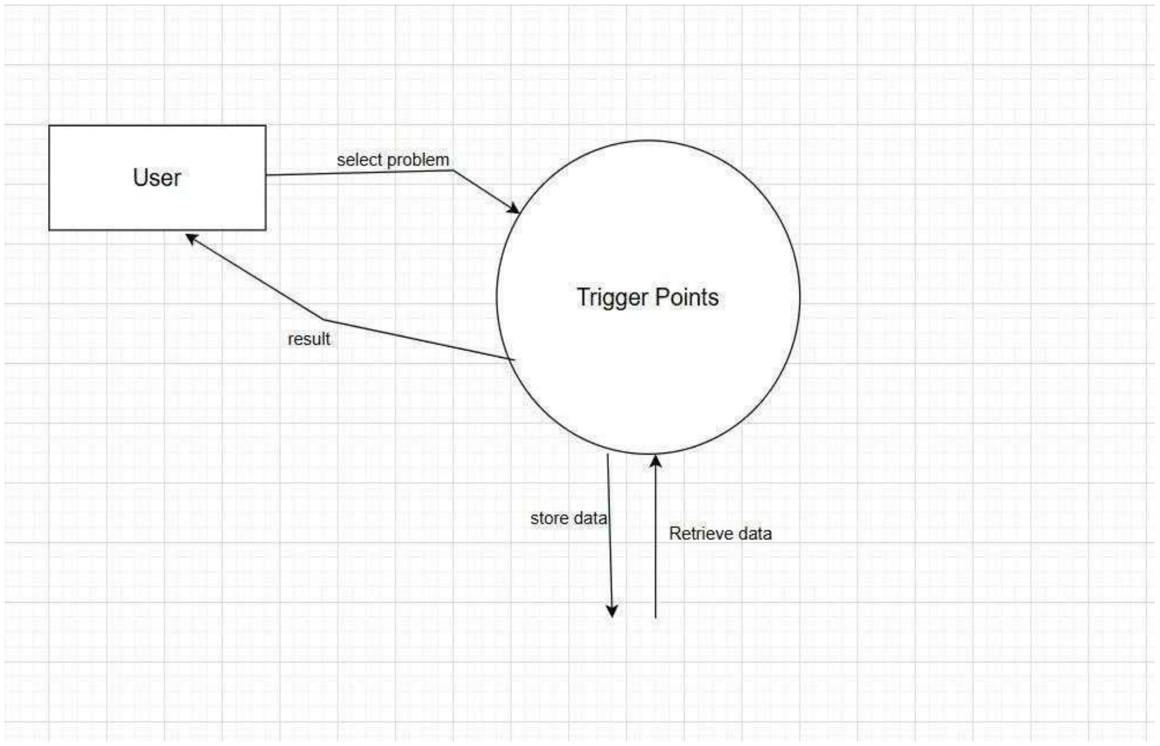


Fig.2

Level 1 DFD:

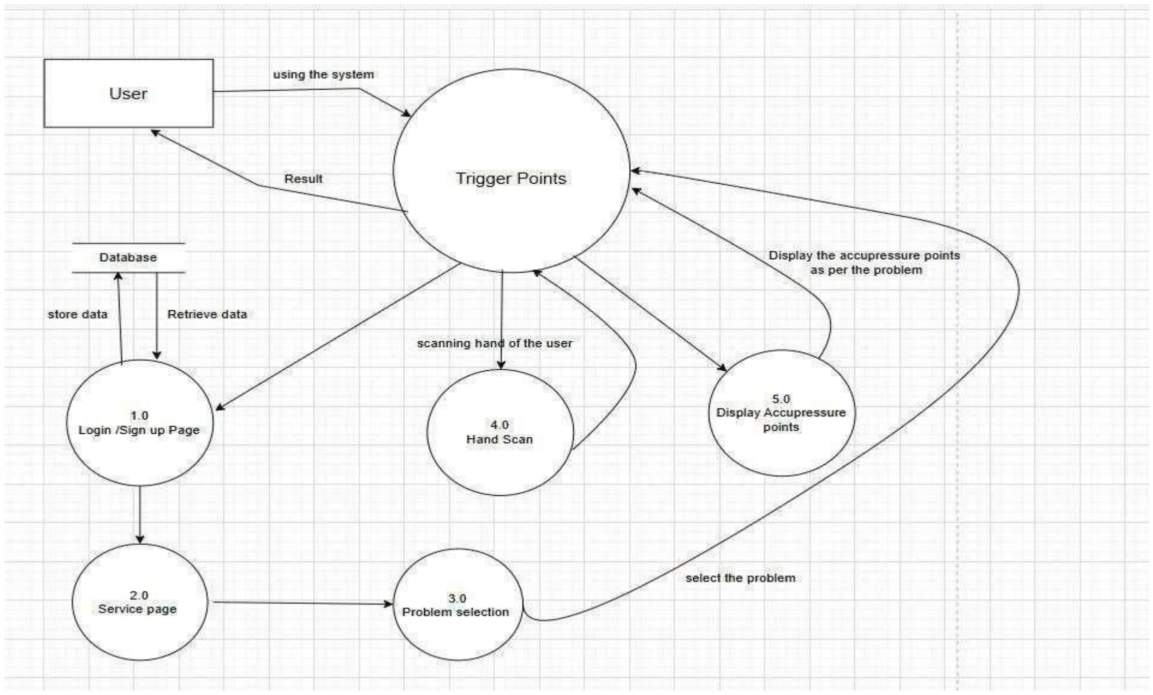


Fig.3

Software Requirements Specification (SRS)

Version 1.0

Trigger points

TABLE OF CONTENTS

1. Introduction

1.1 Purpose

The purpose of this document is to provide a comprehensive description of the **Trigger Points** system, a real-time acupressure identification and visualization platform. This system utilizes **MediaPipe** and **OpenCV** to detect the user's hand and display corresponding acupressure points based on selected health conditions. The document details the core functionality of the system, supported with diagrams and procedural explanations. It also outlines key assumptions, technical requirements, and system constraints. The primary audience for this document includes developers, researchers, and healthcare professionals involved in the development, evaluation, and future enhancement of the system.

1.2 Scope of the Development Project

The **Trigger Points** platform is designed to assist users in identifying key acupressure points on the hand using **real-time image processing with MediaPipe and OpenCV**. The system aims to improve health and wellness by providing drug-free solutions for pain relief and stress management. The project focuses on efficient data processing, accurate point detection, and user engagement, while ensuring the security and privacy of user data.

Key functionalities of the Trigger Points system include the following:

- **Identity and Authentication:**
  The system will securely authenticate users and manage sessions to ensure authorized access.
- **Hand Scanning & Acupressure Identification:**
  Users will scan their hands, and the system will detect key acupressure points using real-time image processing with **MediaPipe and OpenCV**, providing relevant health benefits.
- **Personalized Recommendations:**
  Based on the selected health condition, the system will suggest targeted pressure techniques for pain relief and relaxation.
- **Real-Time Data Processing:**
  The system will process camera input in real time and provide instant visual feedback to users.

- **Health Issue-Based Search:**
  Users will be able to search for specific health ailments, and the system will recommend corresponding acupressure points on the hand.

1.3 Definitions, Abbreviations, and Acronyms Definitions

Table 1: Definitions for Most Commonly Used Terms

Term	Definition
Acupressure Points	Specific points on the body that, when pressed, can relieve pain and promote wellness.
User Interface	The visual part of the system that allows users to interact with the software, including buttons, menus, and screens.
Security Functionality	Practices and techniques used to protect user data and the system from threats and unauthorized access.
Hand Recognition	A system that uses image processing techniques to identify and map acupressure points on a user's hand.
User Profile	A digital identity in the system containing preferences, scanned history, and wellness goals.

Table 1

Table 2: Full Forms for Commonly Used Abbreviations

S. No.	Abbreviation	Full Form
1.	SRS	Software Requirements Specification
2.	UI	User Interface
3.	DBMS	Database Management System
4.	API	Application Programming Interface
5.	AI	Artificial Intelligence

Table 2

1.4 References

[1] Acupressure Points:
https://www.medicalnewstoday.com/articles/what-is-acupressure

[2] User Interface Definition.:
https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI

[3] AI-Based Hand Recognition. Available:
https://www.sciencedirect.com/topics/computer-science/hand-gesture recognition#:~:text=Hand%20gesture%20recognition%20refers%20to,and%20Computer% 2DIntegrated%20Manufacturing%2C%202022

[4] User Authentication. Available:
https://www.techtarget.com/searchsecurity/definition/user-authentication

[5] Cloud Storage Security. Available:
https://www.sciencedirect.com/science/article/pii/S1877705817302862

[6] AI Image Processing. Available:
https://www.tandfonline.com/doi/abs/10.2147/BCTT.S175311

[7] Encryption Techniques. Available:
https://ieeexplore.ieee.org/abstract/document/7204010/

[8] React Router. Available:
https://link.springer.com/chapter/10.1007/978-1-4842-4391-6_9

[9] Tailwind CSS. Available:
https://v3.tailwindcss.com/

1.5 Overview

Trigger Points is a wellness platform that enables users to identify acupressure points on their hands for pain relief and stress management. The platform uses real-time hand tracking through MediaPipe and OpenCV to detect specific points on the hand associated with various health conditions.

Users interact through a secure and user-friendly interface to scan their hand, receive condition-based acupressure point visualizations, and explore natural wellness techniques. The system supports basic user management and provides a focused, real-time experience by displaying one point at a time, allowing users to switch between health issues as needed.

Key Features:

- Secure login and user authentication.

- Real-time hand scanning using MediaPipe and OpenCV for acupressure point identification.
- Condition-based point display and switching.
- Simple user profile management (email/password only).
- Modular structure for future integration with healthcare professionals.

This document outlines the system’s scope, functionality, user requirements, and design specifications to support the effective development and deployment of the platform.

2. Overall Description

2.1 Product Perspective

Trigger Points is a real-time acupressure and wellness platform designed to simplify hand-based acupressure therapy using scientifically structured visualizations. The system supports both **web and mobile browsers**, ensuring wide accessibility and ease of use. Instead of using AI, it leverages **MediaPipe and OpenCV** for image-based hand recognition to accurately identify acupressure points, helping users minimize trial-and-error and improve effectiveness.

2.2 Product Functions

- **User Account Management:**
  Users can register, log in, and recover passwords via a secure authentication system built with Flask and CouchDB.
- **Hand Recognition & Point Visualization:**
  Real-time image processing using MediaPipe and OpenCV detects hand landmarks and displays one acupressure point at a time based on the selected health condition.
- **Personalized Guidance:**
  Condition-based suggestions guide users in applying pressure for relief and relaxation.

- **Search & Filtering:**
  Users can search by health issues and select from a list of predefined conditions.
- **Privacy & Security:**
  Basic user data is securely stored in CouchDB, sessions are managed securely in Flask.
- **Real-Time Feedback:**
  Immediate on-screen feedback helps guide correct hand positioning and pressure application.
- **Progress Tracking (Planned Feature):**
  Future updates may include a history of user interactions and scanned sessions.
- **Educational Resources:**
  Basic tutorials and informational content are available for users to understand techniques.

2.3 User Characteristics

- **General Users:**
  Individuals seeking natural pain relief and wellness through self-guided acupressure therapy.
- **Practitioners (Optional):**
  Health professionals who may validate point placements or guide users offline.
- **Administrators:**
  Responsible for maintaining the system, managing user activity, and ensuring system integrity.

2.4 Constraints, Assumptions and Dependencies

- Requires stable internet connectivity for real-time video processing and user interactions.
- Optimized for modern mobile and desktop browsers (responsive design).
- Single-hand detection only (one hand at a time).
- Secure login is mandatory for personalized access.
- The initial release focuses only on hand acupressure (not full body).

- Does not provide medical treatment; follows non-medical, ethical wellness guidelines.
- Requires periodic maintenance and updates.
- Includes basic tutorials but not professional medical supervision.
- No AI/ML model is used; all processing is based on rule-based logic using MediaPipe landmark data.
- System does not store scan history or pressure feedback; real-time use only, with no long-term data logging.

3. Specific Requirements

3.1 Interface Requirements

- **Intuitive UI:**
  The platform should offer a clean, simple, and user-friendly interface accessible to all age groups.
- **Cross-platform Compatibility:**
  The system must work efficiently on major desktop and mobile browsers (e.g., Chrome, Firefox, Safari).
- **Secure Authentication:**
  A login system with encrypted password handling and session management using CouchDB for secure user access.

3.2 Performance Requirements

- **Purpose:** Manage user accounts and facilitate secure login/logout processes.
- **Inputs:** Email address, password, and optionally username or profile data.
- **Processing:** Verifies credentials against stored data, manages sessions, and handles registration.
- **Outputs:** Authentication status, login success/failure messages, and session tokens.

Hand Recognition Module

- **Purpose:** Detect and identify specific acupressure points on the user's hand for wellness evaluation.

- **Inputs:** Live camera feed or uploaded hand images.
- **Processing:** Detects the hand's position and extracts reference points for mapping.
- **Outputs:** Visual or textual representation of identified hand acupressure points.

Security Module

- **Purpose:** Ensure secure user access and safeguard data against unauthorized use.
- **Inputs:** Login credentials, session tokens, and access requests.
- **Processing:** Encrypts sensitive data, validates user identity, and manages permission controls.
- **Outputs:** Secure access confirmation or rejection based on verification.

Data Management Module

- **Purpose:** Store, manage, and retrieve user interaction history for personalized insights.
- **Inputs:** User-submitted data, scan records, and manually entered notes.
- **Processing:** Saves data in a structured format and allows efficient querying.
- **Outputs:** Historical reports, personalized feedback, and condition tracking over time.

3.3. Quality Attributes

Performance:

The system shall deliver low latency operations and provide real-time feedback for hand recognition and data display, ensuring a responsive and fluid user experience.

- **Usability:**

  The user interface must be intuitive and accessible across various devices (desktop, tablet, mobile). Design should follow consistent navigation patterns and support users with minimal technical expertise.
- **Security:**

  The application must implement strict access controls to prevent unauthorized use. User data should be securely stored and protected through authentication mechanisms and encrypted communication.
- **Reusability:**

  The codebase should follow a modular architecture, allowing components (e.g., user management, hand recognition, data storage) to be reused or extended independently for future updates or additional features.

3.4 Logical Database Requirements

- **Scalability:**
  The system must be able to support a growing number of concurrent users without degradation in performance. It should efficiently handle increased load through horizontal or vertical scaling.
- **Response Time:**
  All user interactions, including login, data retrieval, and hand recognition processing, must complete within 2 seconds under normal operating conditions.
- **Data Consistency:**
  The system shall ensure real-time consistency for all critical data, including user inputs, scan results, and health history. Any changes must be immediately visible across the system.
- **Availability:**
  The system must maintain 99.9% uptime annually, ensuring uninterrupted access except during scheduled maintenance windows.
- **Security:**
  All data transmissions must use standard encryption protocols to protect user credentials and sensitive health data. Access to secure features must be restricted via authenticated and authorized sessions.

3.5 Other Requirements:

- **Maintainability:**
  The system should be built with clean, well-documented code and use industry-standard practices to simplify debugging, enhancements, and long-term maintenance.
- **Scalability**
  The architecture should support scalability in terms of both performance and user base, allowing for easy integration of future modules (e.g., additional hand zones, health features, or analytics).

- **Modularity**:
  The platform must use a modular design approach, ensuring components such as authentication, hand recognition, data storage, and UI rendering can be developed, tested, and upgraded independently.
- **Localization and Accessibility:**
  The interface should support basic accessibility guidelines (e.g., readable fonts, high-contrast UI), and should allow for future localization (support for multiple languages and regional formats).

4.Change History

version	Date	Changes
1.0	**25-03-2025**	**Initial Release**

5. Document Approvers

The Trigger points project has been approved by:

Name: Dr. Ashima Singh
Designation: Assistant Professor
Date: 20-05-25

3. Design phase

3.1 Class Diagram

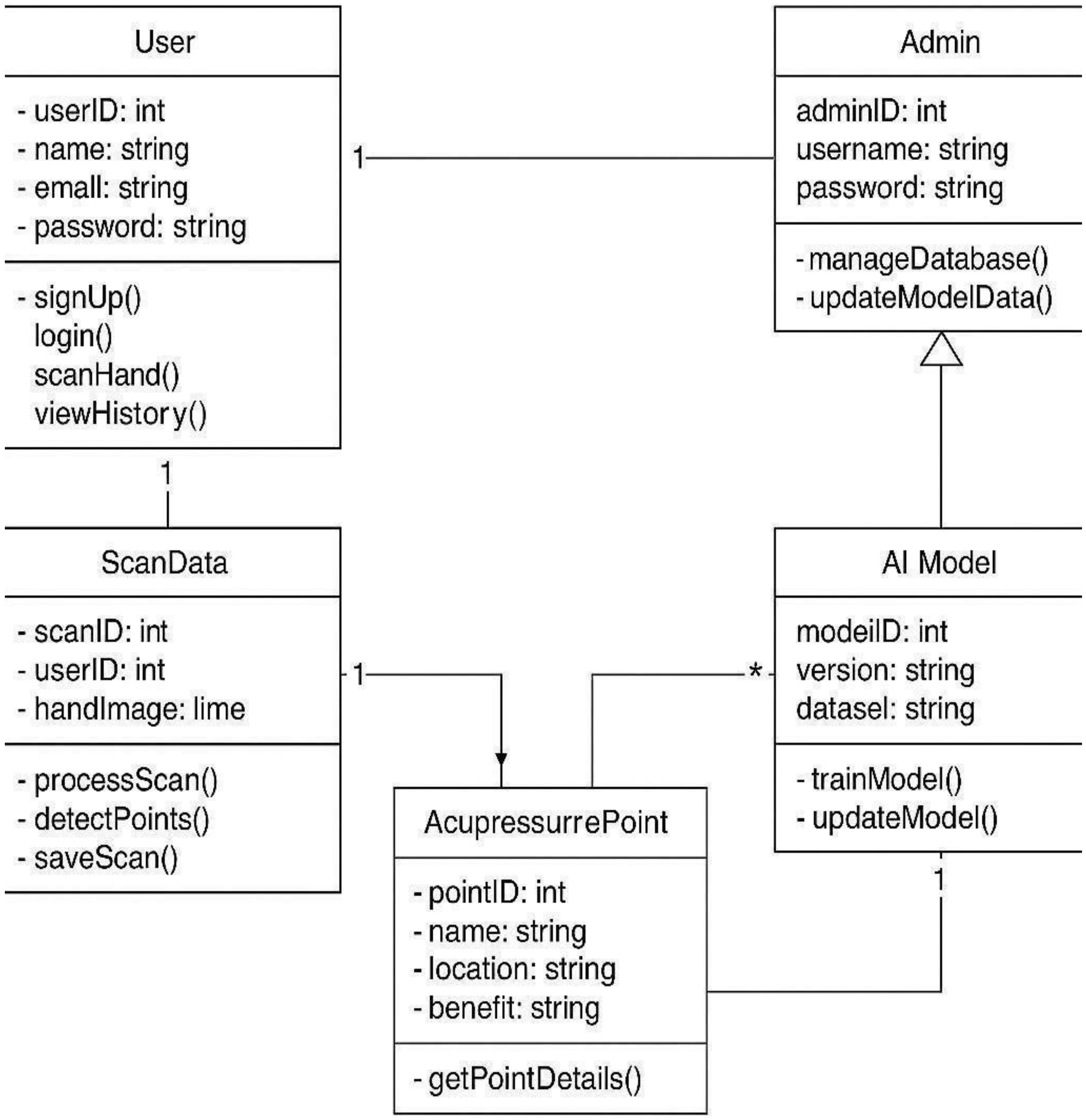


Fig.4

3.2 Module Structure Chart

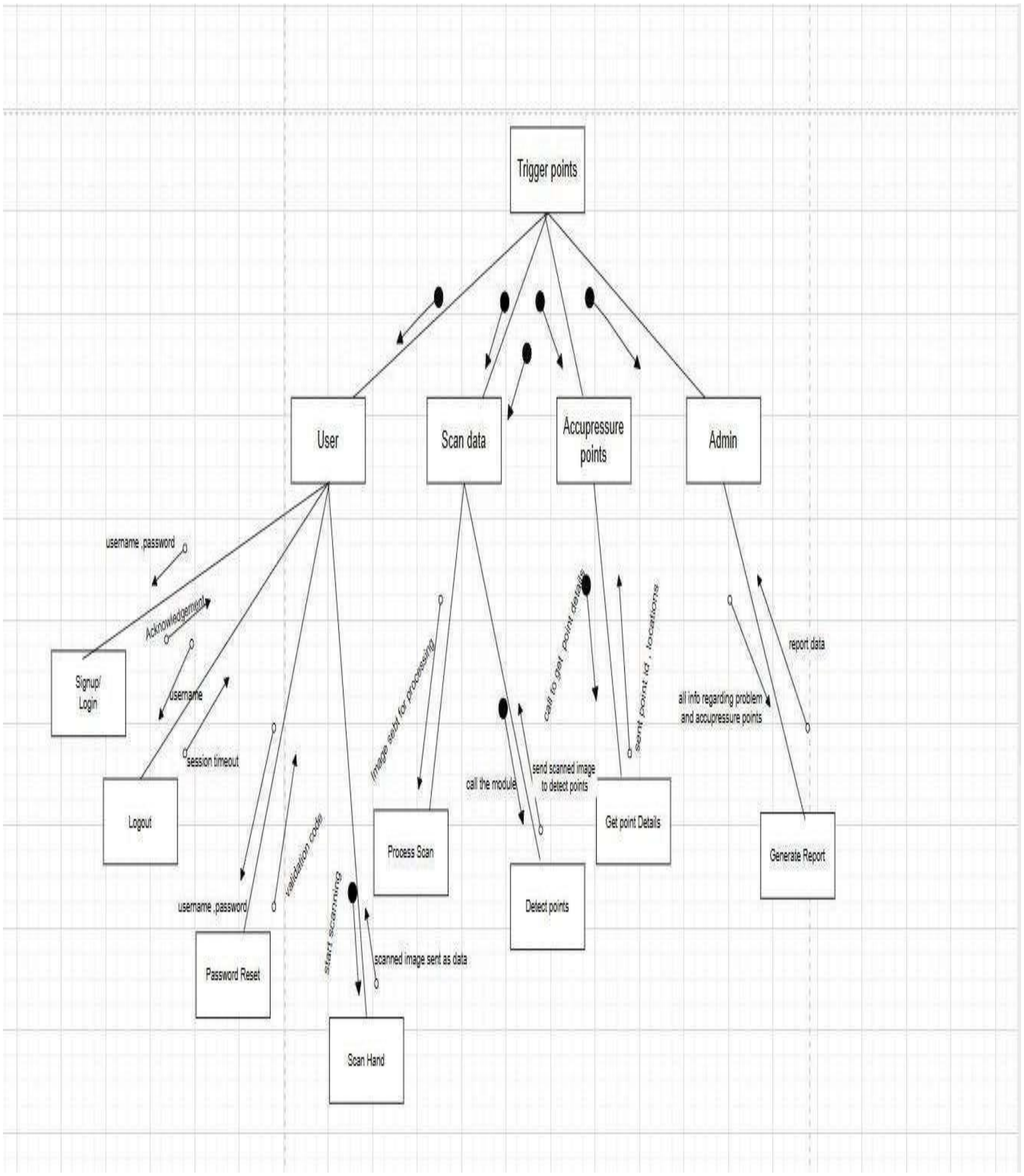


Fig.5

3.3 Swimlane Diagram

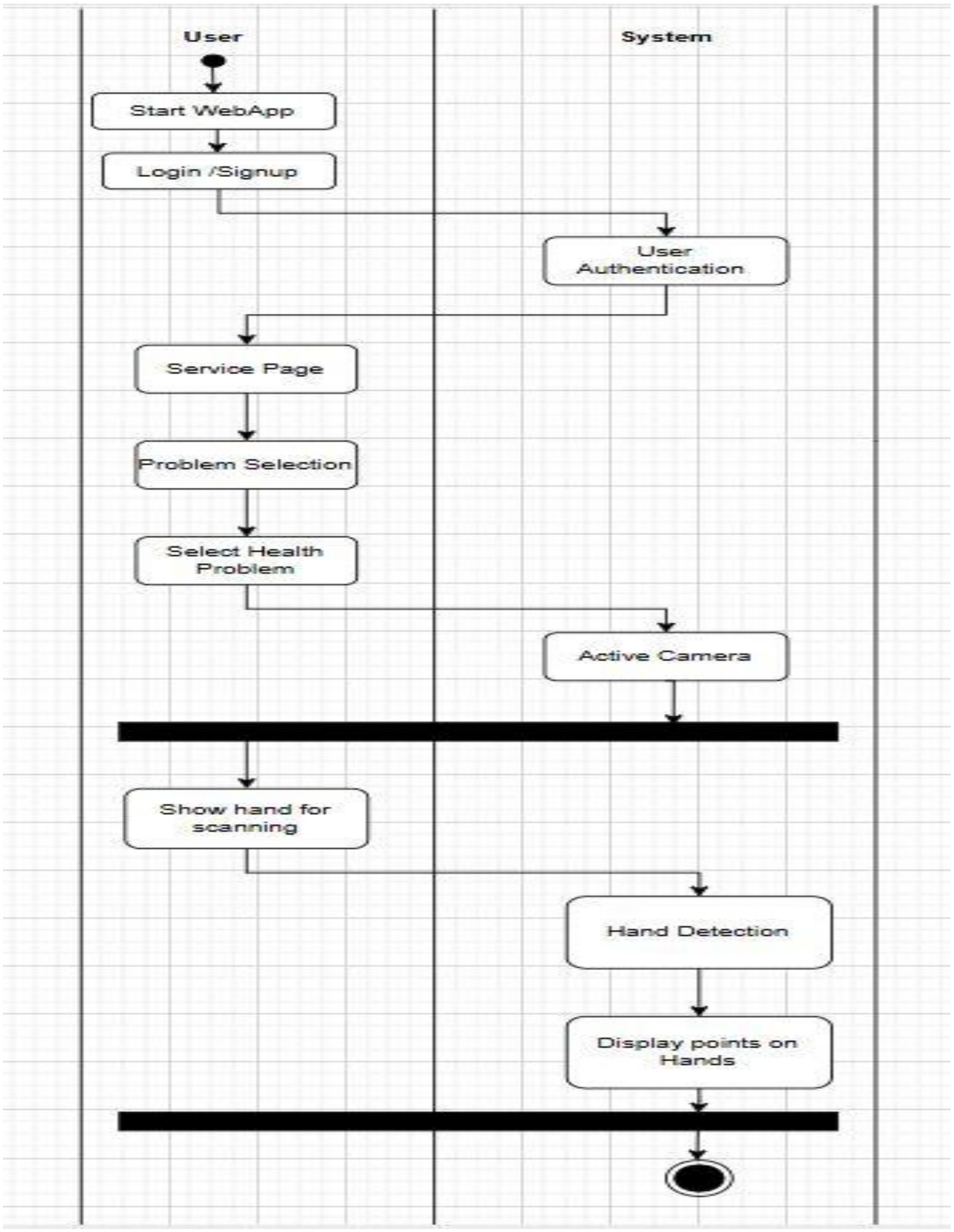


Fig.6

3.4 Sequence Diagram

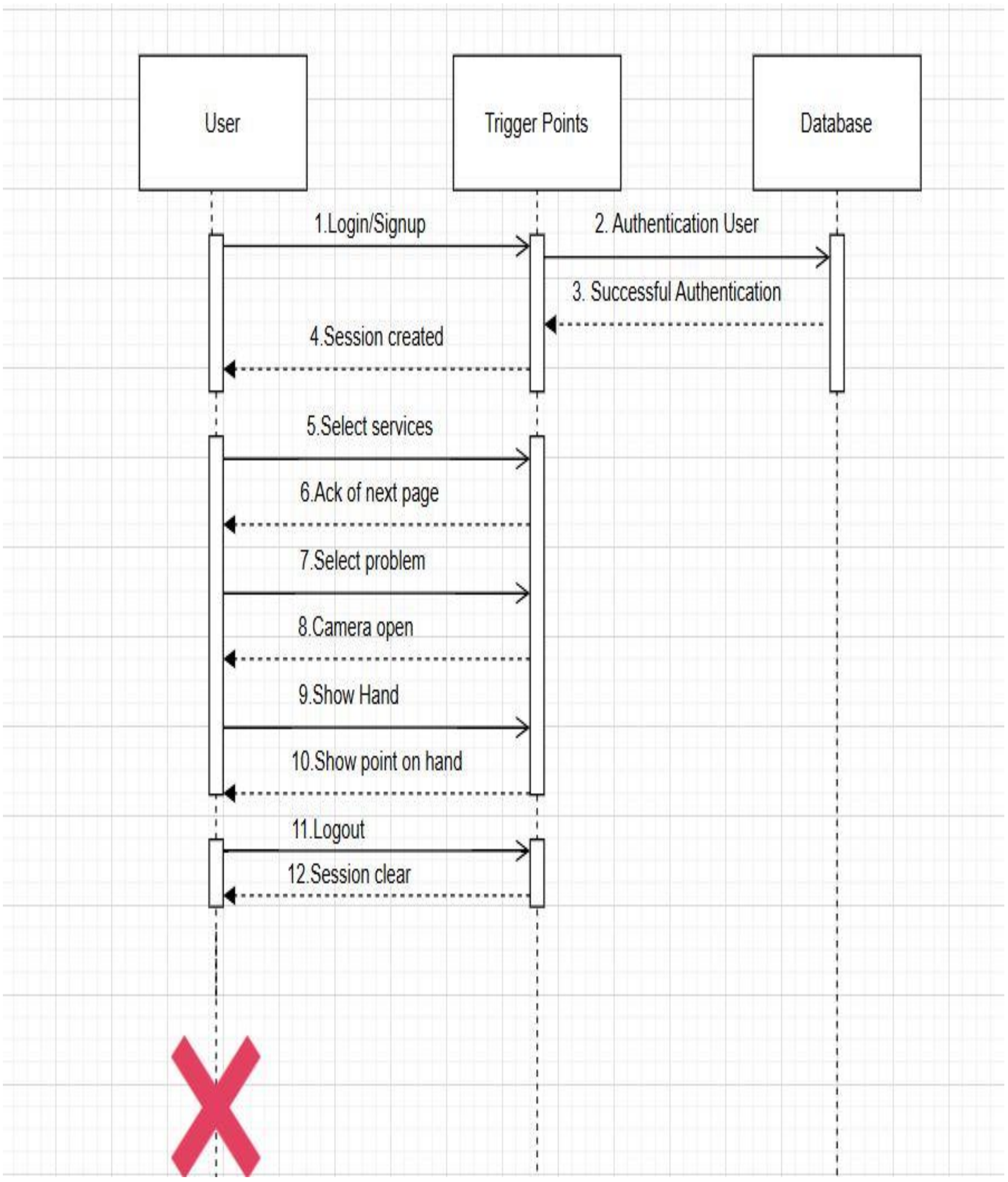


Fig.7

4. Coding and Testing

4.1 Screenshots of Working GUIs of the Project on the basis of Use case Scenario

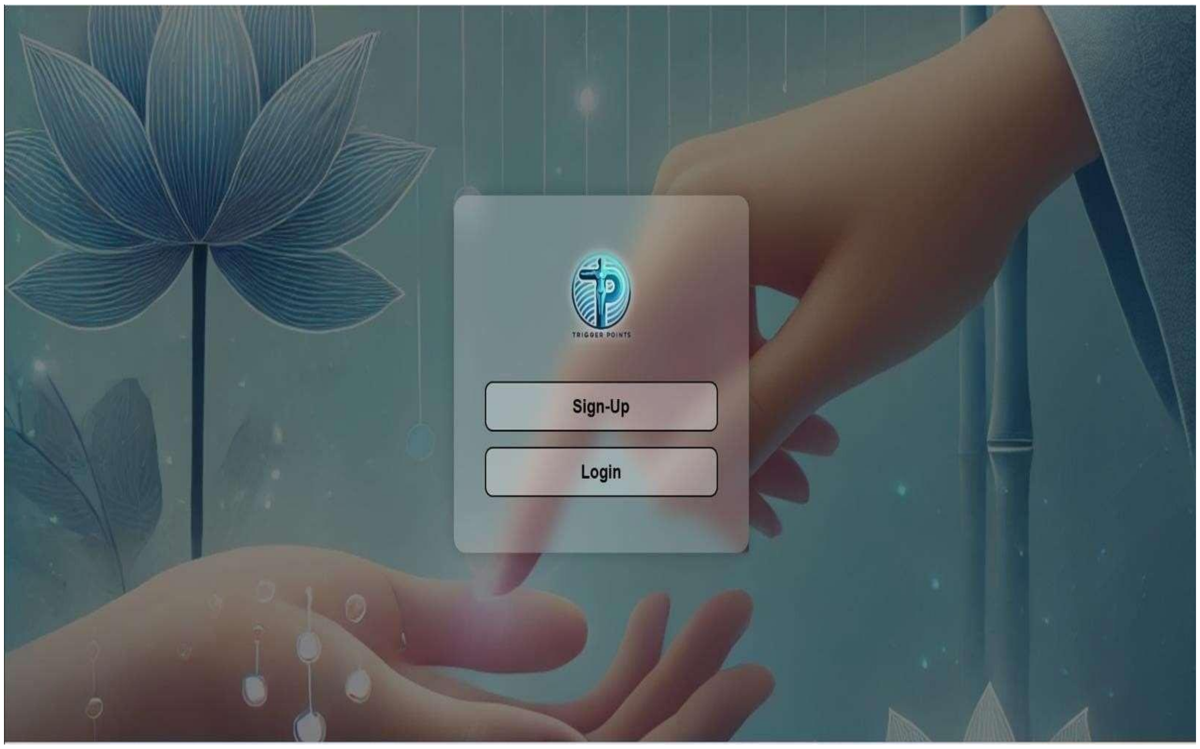


Fig.8 Login/Signup Page

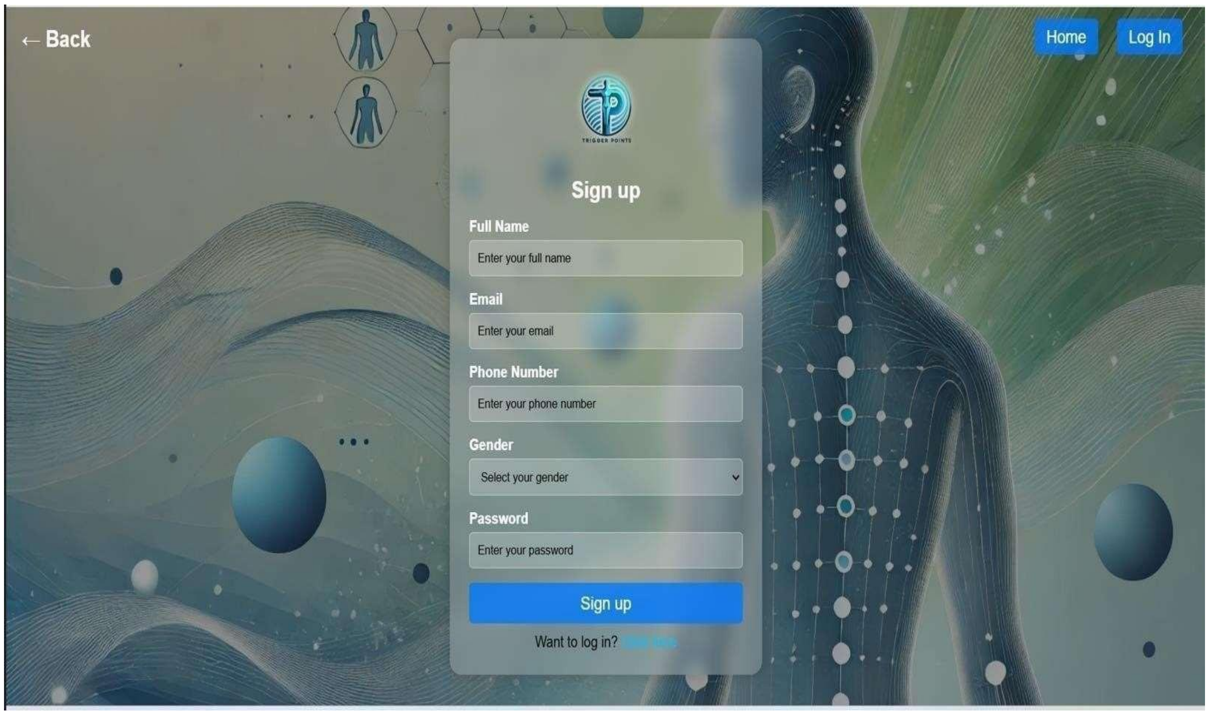


Fig:9 Sign-up Page

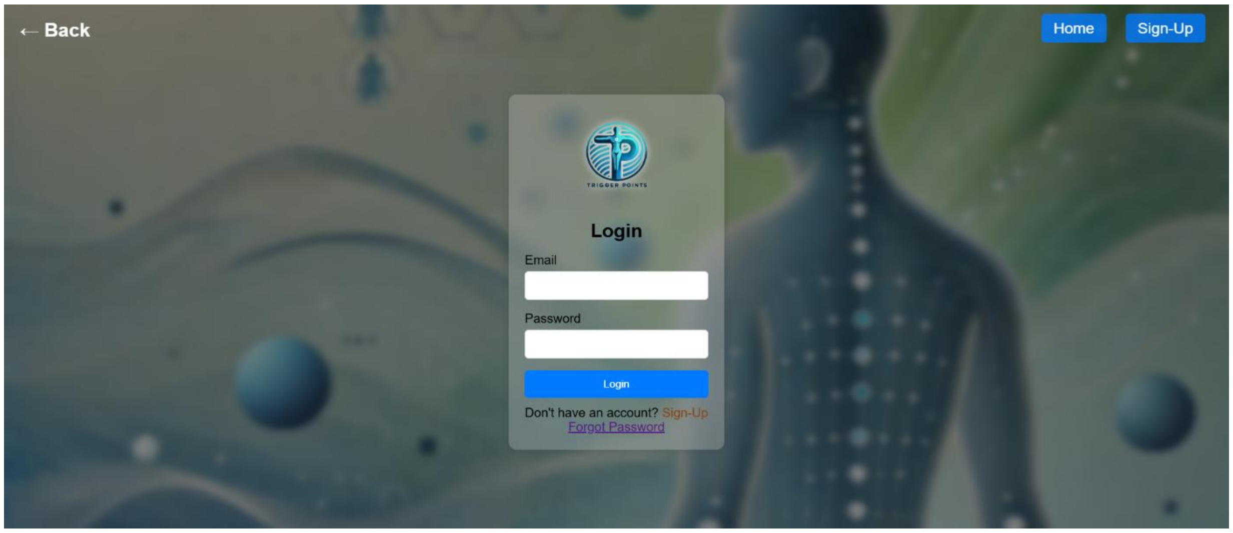
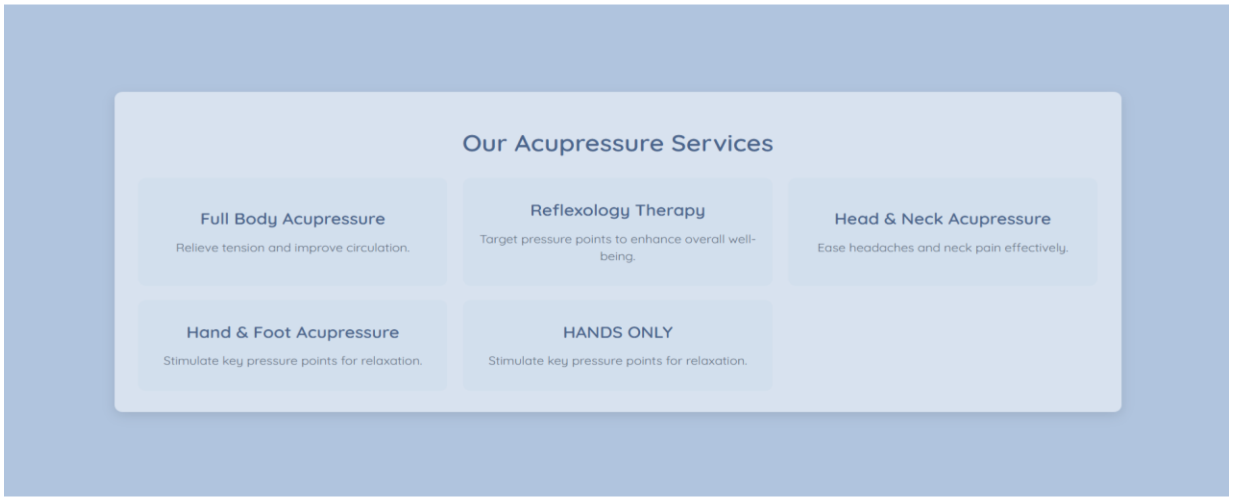


Fig:10 Login page


Our Acupressure Services

Full Body Acupressure
Relieve tension and improve circulation.

Reflexology Therapy
Target pressure points to enhance overall well-being.

Head & Neck Acupressure
Ease headaches and neck pain effectively.

Hand & Foot Acupressure
Stimulate key pressure points for relaxation.

HANDS ONLY
Stimulate key pressure points for relaxation.

Fig:11 Service page

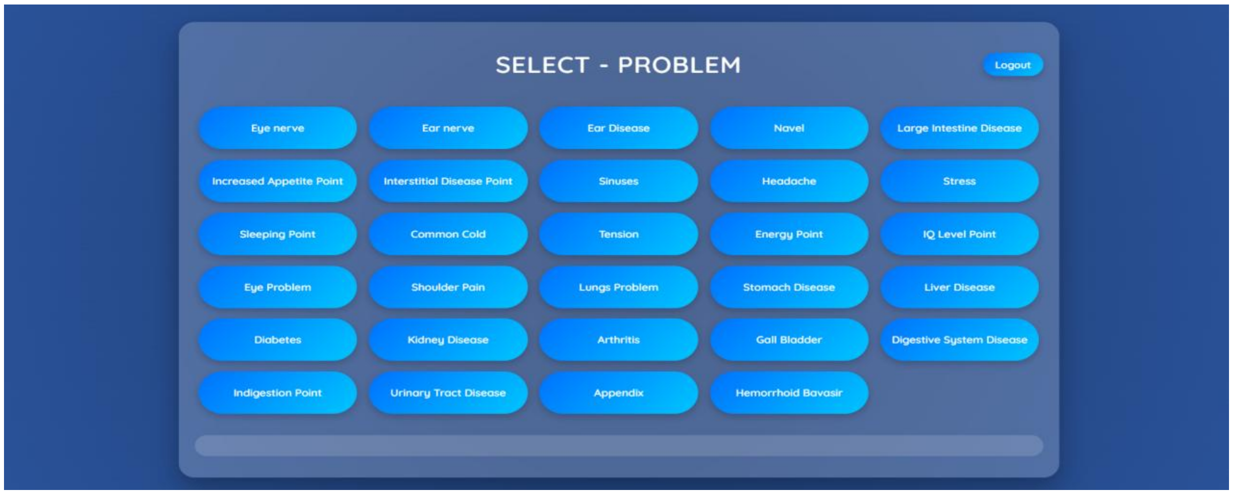
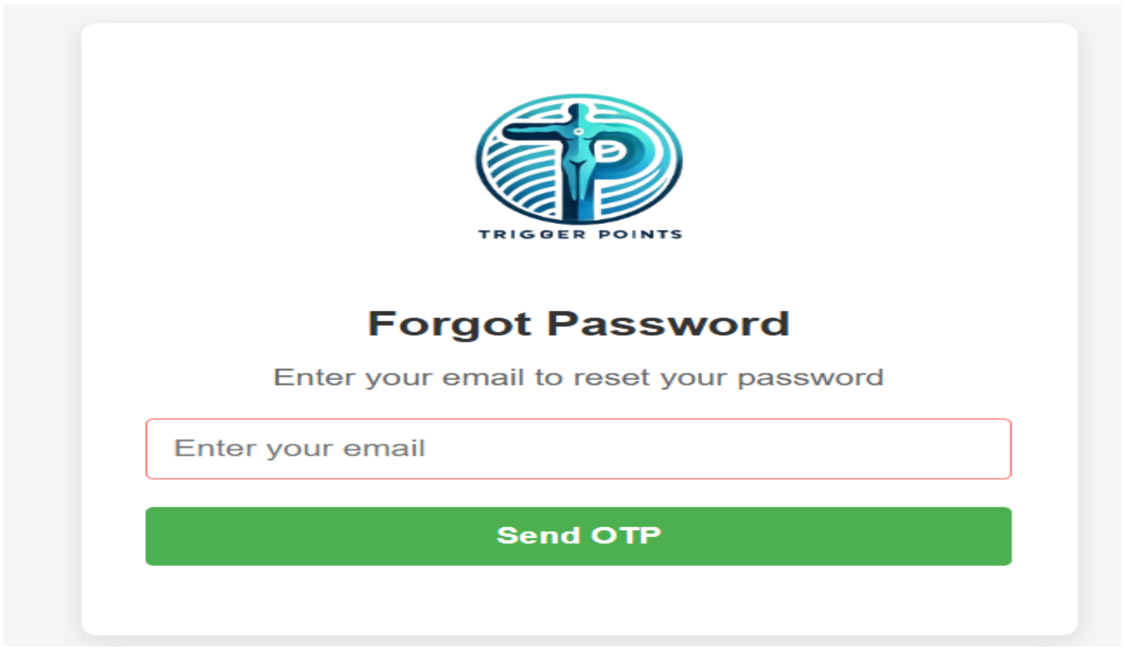


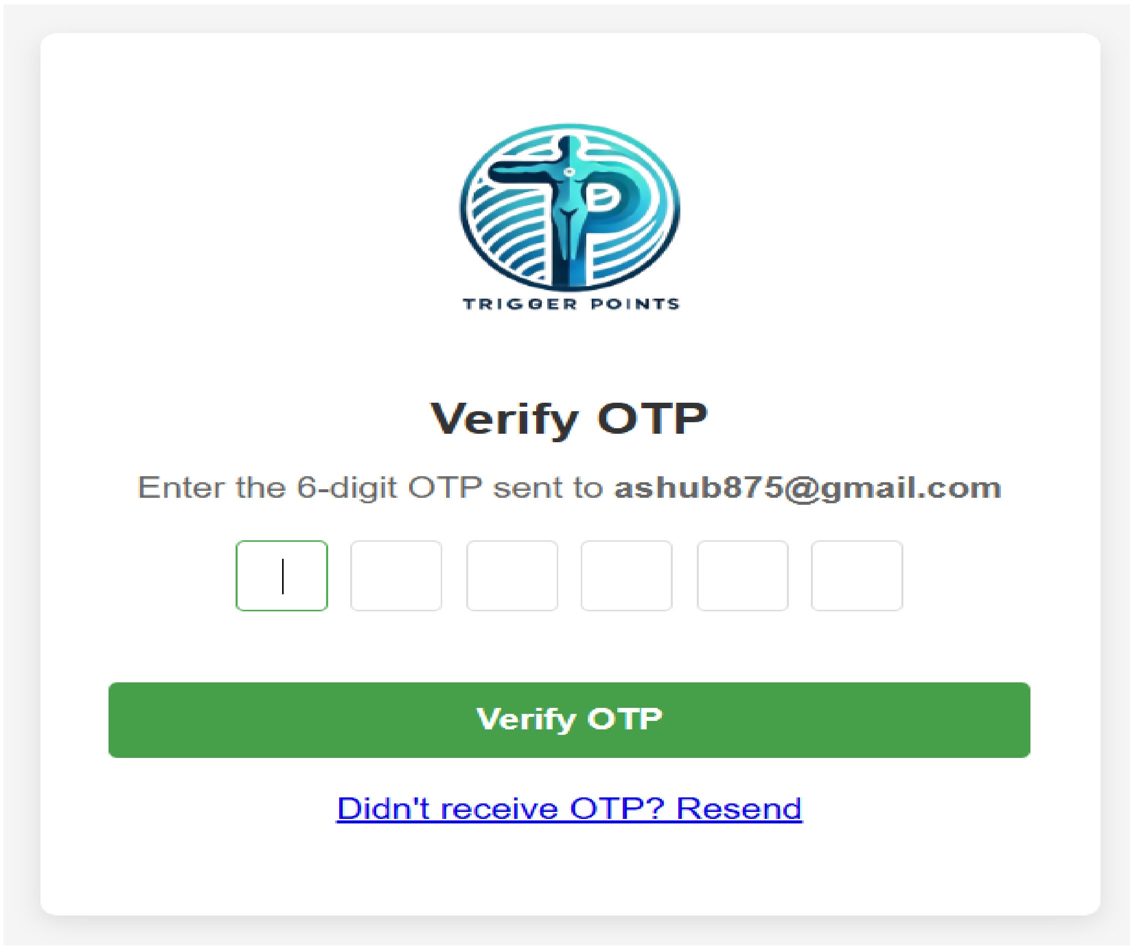
Fig:12 Problem page


TRIGGER POINTS

Forgot Password

Enter your email to reset your password

Fig:13 Forget password

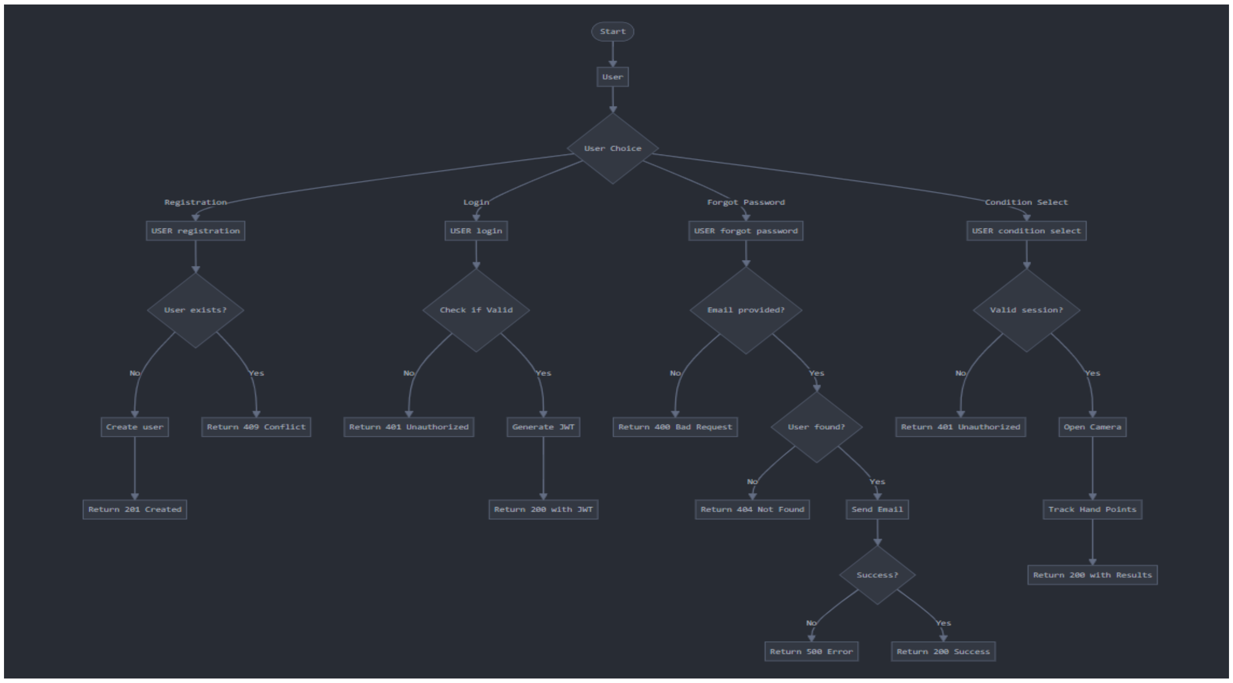

TRIGGER POINTS

Verify OTP

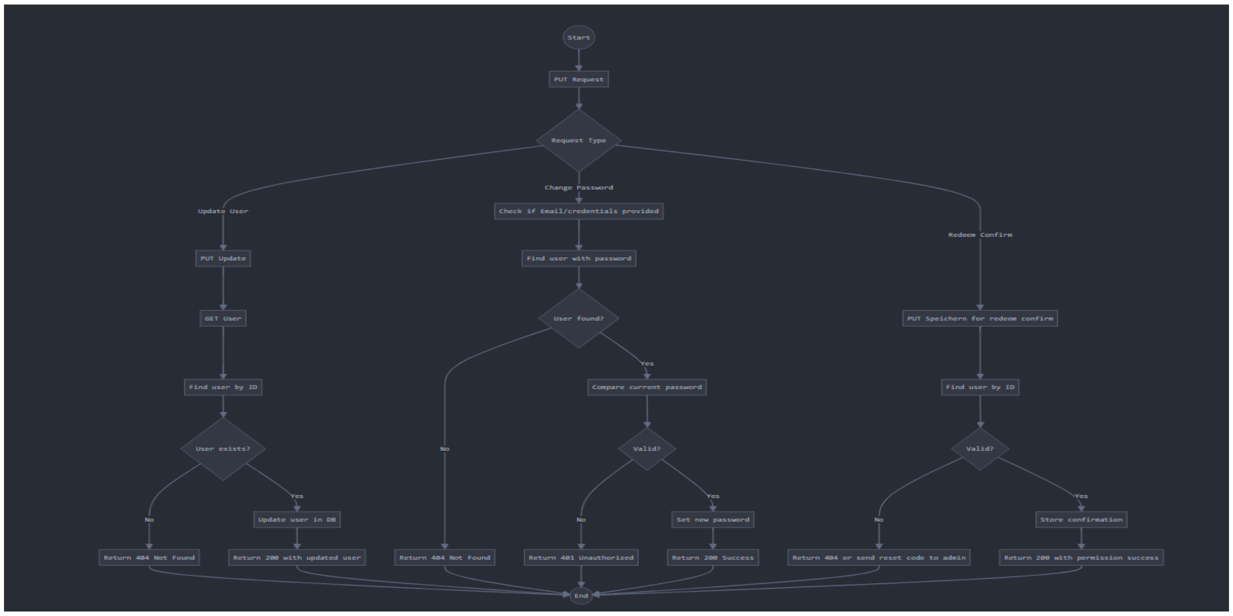
Enter the 6-digit OTP sent to **ashub875@gmail.com**

Fig:14Verify OTP

4.2 Cyclomatic complexity calculation for all code snippets



M=E-N+2P
Nodes=26, Edges=28
Connected components: 1
Cyclomatic Complexity=28-26+2(1) =4



M=E-N+2P
Nodes=23, Edges=25
Connected components: 1
Cyclomatic Complexity=25-23+2(1) =4

4.3 Test Case Report (by the Testing Team)

Test Case 1

Test Case 1:	**Test Case Name:** Forget password OTP Generation	**Page:** 1
System: Trigger points	**Subsystem:** Authentication	
Designed by:	**Design Date:** 30/04/25	
Executed by: Akhil Garg	**Execution Date:**	
Short Description: Test user forgot password flow with OTP generation and verification		

Pre-conditions

- User is registered with email in the system
- The system is running and /forgot-password endpoint is available
- Email service is configured and operational
- OTP generation service is functioning

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Navigate to login page and click "Forgot Password"	System displays forgot password form requesting email	Pass	
2	Enter registered email user@gmail.com	System validates email format	Pass	
3	Click "Send OTP"	System generates 6-digit OTP and sends to registered email. Displays OTP entry screen with confirmation message: "OTP sent to user@gmail.com	Pass	
4	Enter incorrect OTP	System shows error message: "Invalid OTP. Please try again."	Pass	
5	Enter correct OTP	System validates OTP and displays password reset form	Pass	
6	Enter new password that doesn't meet requirements (e.g. password length is less than 6)	Password does not meet all requirements	Pass	
7	Enter valid new password "NewPass@123" and confirm password "NewPass@123"	System accepts password	Pass	
8	Click "Reset Password"	System updates password and shows success message: "Password reset successful"	Pass	
9	Try login with old password	"Invalid password"	Pass	
10	Try login with new password "NewPass@123"	System authenticates user and system navigate to service page	Pass	

Post-conditions: -

- User password is updated in the database
- Old password no longer works for authentication
- OTP is invalidated after successful password reset

The OTP enter time is just 1 minute which is very less it can be increased.

Test Case 2

Test Case 2: **System:Trigger points** **Designed by:** **Executed by:**Bhuvesh Gupta **Short Description**: Test Hand scanning and acupressure point identification	**Test Case Name:** Hand recognition and Acupressure point detection **Subsystem:** MediaPipe Hand Landmark Detection **Design Date:** 30/04/25	**Page:1**

Pre-conditions: -
- **User is logged in to the application**
- **Device has functioning camera access**
- **User is on the hand scanning page**

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Navigate to "Problem Statement" section	System displays categorized list of health conditions/problems	Pass	
2	Select specific problem (e.g., "Headache")	System acknowledges selection and System activates the camera	Pass	
3	Position hand within the frame	System uses MediaPipe to detect hand and displays real-time landmark tracking	Pass	
4	Position hand is not in the frame	Hand is not in range or not straight	Pass	
6	Click on "camera screen and then q" to quit camera	System properly closes camera stream and returns to Problem page	Pass	

Post-conditions
- Problem-specific acupressure points are accurately identified and highlighted

Test Case 3

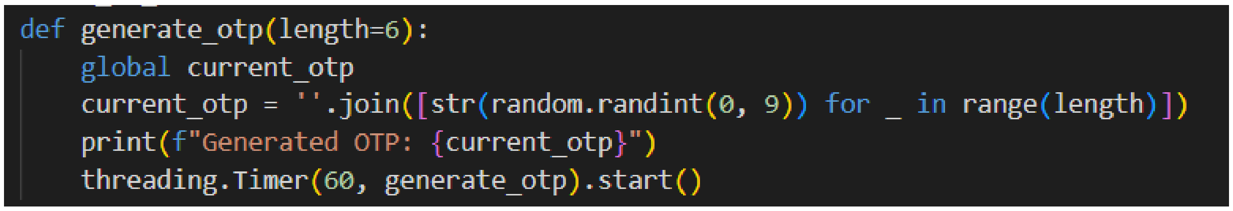
Test Case 2: **System:Trigger points** **Designed by:** **Executed by:** Anchal Jain & Mayank **Short Description: Validates system's ability to detect if a hand is within the specified range (0.3-0.6),**	**Test Case Name: Hand in Range** **Subsystem:** Range and Orientation Detection **Design Date:** 30/04/25	**Page:1**

Pre-conditions: -
- **Applicable on All the problems**
- **Camera / Sensor working properly**
- **Display Screen is working properly**

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Place hand within range (0.4 meters from sensor) with palm straight	System detects hand within range and displays problem points on the hand	Pass	
2	Place hand within range (0.5 meters from sensor) with palm not straight	System detects hand within range but shows message "Hand is not in range or not straight"	Pass	
3	Place hand outside of range (0.2 meters from sensor) with palm straight	Place hand outside of range (0.7 meters from sensor) with palm straight	Pass	
4	Place hand outside of range (0.7 meters from sensor) with palm straight	System displays message "Hand is not in range or not straight"	Pass	
5	Slowly move hand from out of range (0.7m) to within range (0.5m) with palm straight	System transitions from showing "Hand is not in range or not straight" to displaying problem points once hand enters valid range	Pass	

Post-conditions
- All detection mechanisms function as expected
- No error logs generated due to normal test usage

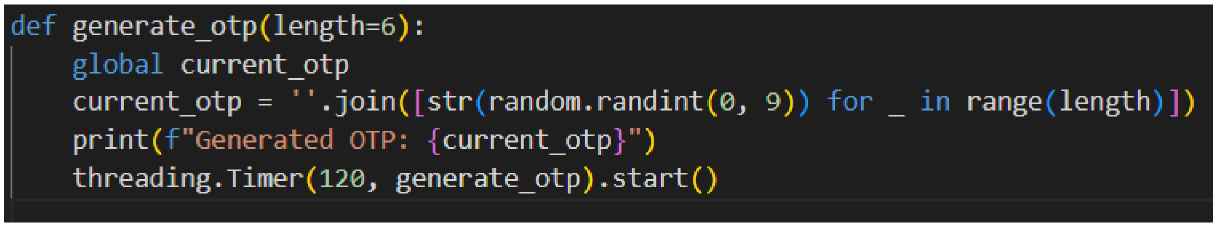
4.4 Screenshots of working GUIs of project - Labelling of Errors (by the Testing Team)



```
def generate_otp(length=6):
    global current_otp
    current_otp = ''.join([str(random.randint(0, 9)) for _ in range(length)])
    print(f"Generated OTP: {current_otp}")
    threading.Timer(60, generate_otp).start()
```

The time of enter OTP is 60 sec is very less.

4.5 Screenshots of the improved version of the Working Modules (by the Developer's Team)



```
def generate_otp(length=6):
    global current_otp
    current_otp = ''.join([str(random.randint(0, 9)) for _ in range(length)])
    print(f"Generated OTP: {current_otp}")
    threading.Timer(120, generate_otp).start()
```

The time of enter OTP is increase to 120 sec

4.6 Test Case Report with improvements (highlighted) (by the Developer's Team)

Overview of Fixes:

#	Test Scenario	Issue (Before)	Improvement(now)
1	OTP	OTP enter time is very less	Increase the time of OTP enter

Detailed Test Cases:

TC ID	Test Case Description	Steps	Expected Result	Status
TC-01	Enter OTP120 sec	Input OTP on OTP page	"Shift to the Enter new password page"	pass
TC-01	Enter OTP after 120 sec	Input OTP on OTP page	"Show the Invalid OTP"	pass