## Test Plan: AI Derma Checker Web Application

Document Version: 1.0

Date: June 10, 2025

Prepared By: Sadia Yaqoob and Noor-ul-Huda

## Introduction:

1.1 Purpose:

The purpose of this test plan is to outline the strategy and activities for testing the "Skin Disease Detection Web Application." This document specifies the testing scope, objectives, types of testing, responsibilities, environment, and criteria for successful completion of the testing phase.

1.2 System Overview:

This web application allows users to upload an image of their skin for automated disease detection using a custom-trained Deep Learning (DL) model. After image upload, users make a payment to view the results. Upon successful payment, they can see the predicted disease, confidence score and access a downloadable PDF report containing detailed information, symptoms and care tips.

1.3 Scope of Testing:

This test plan covers the functional, performance, security, usability, and AI/ML-specific aspects of the web application, including the image upload, payment processing, results display, and PDF generation functionalities.

**2. Test Objectives**

The primary objectives of this testing effort are to:

* **Verify** that the application functions according to specified requirements (image upload, payment, results display, PDF generation).
* **Assess** the accuracy, robustness, and fairness (bias) of the integrated DL model.
* **Ensure** the payment gateway integration is secure and reliable.
* **Evaluate** the application's performance under expected user loads.
* **Identify** and document any defects or deviations from expected behavior.
* **Ensure** the PDF report is accurate, well-formatted, and downloadable

**3. Test Approach (In/Out):**

* 1. In Scope:
* User Interface (UI) interactions (navigation, form submissions).
* Image upload functionality (various formats, sizes, and valid/invalid inputs).
* Payment gateway integration (successful/failed transactions, error handling).
* DL model inference and accuracy on various skin conditions (and healthy skin).
* Display of disease results (name, confidence score).
* PDF report generation and download (content, formatting, layout, watermark, logo, date).
* Error handling and informative error messages.
* Security aspects (data privacy, input validation, payment security).
* Application performance (response times, load handling).
* Cross-browser compatibility.

## **3.2 Out of Scope:**

## Testing of third-party services beyond their integration points (e.g., internal workings of the payment gateway provider).

## Load testing beyond specified concurrency limits (e.g., extreme stress testing).

## Deep-dive infrastructure testing (unless directly impacting functionality).

## Accessibility testing for compliance standards (unless specified as a requirement).

## Testing of fairness (bias) of the integrated DL model beyond specified limit.

# 4. Test Strategy:

# 4.1 Overall Approach:

# Testing will follow a black-box approach, focusing on the functionality of the application from a user's perspective without knowledge of internal code structure.

# It will also incorporate specific white-box elements for DL model evaluation and coding as well (e.g., analyzing model outputs). End-to-end testing will be performed for critical user flows.

# 4.2 Testing Phases:

# Unit Testing: (Performed by developers) Individual components (e.g., image processing module, payment API connector).

# Integration Testing: Verifying communication between modules (e.g., upload -> model -> results).

# System Testing: Comprehensive testing of the entire integrated system against requirements. This phase includes:

# DL Model Testing: Specific evaluation of AI performance.

# Payment Testing: Transaction flows.

# PDF Generation Testing: Report accuracy and integrity.

# User Acceptance Testing (UAT): (Performed by business users/stakeholders) Validating the application meets business needs and user expectations.

# 4.3 Test Environment:

# Development Environment: For unit and initial integration testing.

# Staging/QA Environment: Dedicated environment mirroring production for system testing.

# OS: [e.g., Ubuntu 22.04 LTS]

# Web Server: [e.g., Uvicorn]

# Database: [e.g., PostgreSQL, MongoDB]

# DL Framework: [e.g., PyTorch/TensorFlow, YOLOv8]

# Python Version: [e.g., 3.9+]

# Payment Gateway Sandbox/Test Credentials.

# Client Devices: Various desktop browsers (Chrome, Firefox, Edge, Safari), mobile browsers (Android, iOS).

# 4.4 Test Data Management

# Image Data: A diverse set of skin images, including:

# Images for each of the 10 defined disease classes.

# Images with varying skin tones (critical for bias detection).

# Images with different lighting conditions and different body parts

# 5. Test Deliverables

# Test Plan Document (this document)

# Test Cases (detailed steps, expected results)

# Bug Reports/Defect Logs

# Summary Test Report

# 6. Entry Criteria

# All major development features for a release cycle are code-complete.

# All critical unit tests have passed.

# The build is stable and deployed to the QA environment.

# Required test data is prepared and accessible.

# Test environment is configured and validated.

# 7. Exit Criteria

# All critical and high-priority defects are resolved and retested.

# At least 95% of planned test cases have been executed.

# Test execution for critical paths shows 100% pass rate.

# Payment gateway integration is confirmed fully functional.

# All PDF report content and formatting are validated.

# UAT sign-off is obtained.

# 8. Suspension Criteria and Resumption Requirements

# Suspension: Testing will be suspended if a blocking defect (P1/Critical) is found that prevents further significant testing. example(DL Model give No-Response or Garbage Output)

# Resumption: Testing will resume once the blocking defect is fixed and a new stable build is deployed.

# 9. Approval

# This Test Plan has been reviewed and approved by the following stakeholders:

# [Test Lead Name]: Burhan Ahmed Date:  10/6/2025