PROJECT REPORT

AI LOCALIZATION AND CLASSIFICATION OF SKIN DISEASE WITH ERYTHEMA USING IBM WATSON

Team ID: PNT2022TMID27240

Batch: B7-1A3E

TEAM LEADER:

Name : G. CHARAN TEJA

Register Number : 311019104024

TEAM MEMBERS:

Name : DHARANI VISHAL S P

Register Number : 311019104018

Name : JERLIN T D

Register Number : 311019104029

Name : KISHORE R

Register Number: 311019104041

KCG COLLEGE OF TECHNOLOGY- karapakkam.

Chennai-600097.

TABLE OF CONTENTS

1.INTRODUCTION

1.1 PROJECT OVERVIEW

Skin is the largest and most sensitive part of the human body which protects our inner vital parts and organs from the outside environment, hence avoiding contact with bacteria and viruses. Skin also helps in body temperature regulation. The skin consists of cells, pigmentation, blood vessels, and other components. It is comprised of 3 main layers, namely, the epidermis, the dermis, and the hypodermis.

Skin diseases occur commonly among humans. They are usually caused by factors like different organism's cells, a different diet, and internal and external factors, such as the hierarchical genetic group of cells, hormones, and immune system of conditions. These factors may act together or in a sequence of skin disease. There are chronic and incurable diseases, like eczema and psoriasis, and malignant diseases like malignant melanoma.

A patient can recover from skin diseases if it is detected and treated in the early stages and this can achieve cure ratios of over 95%. Hence, it is important to identify these diseases at their initial stage to control them from spreading.

1.2 PURPOSE

Skin Disease Detection at edge predicts the disease of skin from the image of that infected part in less than one second. This system simply takes a skin disease image and gives the disease name with accuracy and time taken for prediction.

Skin diseases are primarily diagnosed visually, beginning with an initial clinical screening and followed potentially by dermoscopic analysis. Such a system is often prone to errors.

The main purpose of this project is to improve the accuracy of diagnostic systems by using Image Processing and classification techniques.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

Skin diseases are primarily diagnosed visually, beginning with an initial clinical screening and followed potentially by dermoscopic analysis. To ascertain what type of skin disease a person has, they must visit a dermatologist. The dermatologist then performs visual analysis using various tests, some of which include:

1. Patch test: Known allergens are applied to a patch of skin and left for some time. The skin is then tested for a reaction.

2. Biopsy: Skin is removed using a scalpel, a blade or a biopsy tool and taken to a laboratory for analysis. Culture: Skin of affected area or hair or nails are cultured to determine which microorganism is causing the infection.

Such a system is often time consuming and requires a number of expert professionals. Since there are people involved in this process, it is prone to human errors. This system is also quite expensive as laboratories charge a lot of fees for the tests.

2.2 REFERENCES

Domain Name: Detection of Skin Diseases

Use case Name: AI- Based Localization and Classification of skin disease with erythema using IBM Watson

Paper 1

Authors: Md. Nazmul Hossen, VijayakumariPanneerselvam

Year: 2022

Title: Detection of Skin Diseases

Proposed solution :An image augmentation strategy was followed to enlarge the dataset and make the model more general.

Pros and Limitation:The CNN-based skin disease classificationmerged with the federated learning approach is a breathtaking concept to classify human skin diseases while ensuring data security.

Problem proposed: It is used to detect the skin cancer diseases.

Paper 2

Authors: Md. Nazmul Hossen, VijayakumariPanneerselvam.

Year: 2022

Title: Melanoma type detection

Proposed solution :The gradient descent Similarity Measure presented CNN approach for Text Processing (SMTP).In this research the result in the development of two different methods for identifying the melanoma cancer stages.

Pros and Limitation:The suggested SMTP loss function causes exceptionally little loss in compared to all of the previous loss functions, and this has shown to be very effective in terms of enhancing sensitivity, specificity, and accuracy.

Problem proposed: This research resulted in the development of two different methods for identifying the stages of melanoma cancer.

Paper 3

Authors: G. Glorindal,S. Arun Mozhiselvi,T. Ananth Kumar,K. Kumaran

Year: 2021

Title: A Simplified Approach for Melanoma Skin Disease

Proposed solution :An image processing approach with an easily driven Application Programmable Interface commonly known as API, has been proposed to diagnose skin diseases at their earlier stages.

Pros and Limitation:The image processing follows preprocessing, segmentation, feature extraction, and classification steps, which apply contrast stretching and median filter, Fuzzy C Means, Grey Level CoOccurrence Matrix(GLCM), and Gabor filter, Support Vector Machine .

Problem proposed: This methodology gives a simple approach to detect a skin disease, especially melanoma.

Paper 4

Authors: Tryan Aditya Putra,SyahidahIzzaRufaida

Year: 2020

Title: Enhanced Skin Condition Prediction

Proposed solution :In this study,we propose a dynamic training and testing augmentation capable of increasing performance significantly.

Pros and Limitation:There is still only limited research focused on dynamic data augmentation, even in the fields of machine learning and computer vision.

Problem proposed: This method delivers a superior result, and this study also shares the searched augmentation policy utilized, which requires extraordinary resources.

Paper 5

Authors: Rashmi Patil, Sreepathi Bellary

Year: 2020

Title: Machine learning approach in melanoma cancer stage detection

Proposed solution :This study utilised two different deep learning algorithms, namely the Lesion Feature Network (LFN) and the Lesion Indexing Network (LIN).

Pros and Limitation:The experiment shows that the suggested (LIN) performs better than existing machine learning algorithms for lesion segmentation and classification.

Problem proposed: The major three aims in the field of skin lesion image processing to identify the melanoma cancer.

Paper 6

Authors: Kassem MA, Hosny KM, Fouad MM.

Year: 2020

Title: Skin Lesions Classification into Eight Classes for ISIC

Proposed solution :This method was used to train the CNN from start to finish.In order to train CNN, a dataset consisting of 129,450 clinical photos is used.

Pros and LimitationIt is necessary to determine which malignancies are the most prevalent, and in the second scenario, it is necessary to determine which skin cancer is the deadliest.

Problem proposed: The results of the author's experiment, CNN performed better than all of the specialists who were tested in both tasks..

Paper 7

Authors :Ichim L, Popescu D

Year: 2020

Title: Melanoma Detection

Proposed solution :The proposed a variety of methodologies that can be used to ensure good training and learning evenwith a limited amount of training data to detect the melanoma disease.

Pros and Limitation:This approach enables the classification network to extract more representative and specific features from segmented findings rather of the complete dermoscopy images, hence reducing the quantity of training data that is required to be collected.

Problem proposed: The results in better identification accuracy of melanoma detection.

Paper 8

Authors: Ling-Fang Li, Xu Wang, Wei-Jian Hu

Year: 2020

Title: Skin Disease Image Recognition

Proposed solution :The skin disease image recognition method based on deep learning is better than those of dermatologists and other computer-aided treatment methods in skin disease diagnosis.

Pros and Limitation:We also analyze the current progress in this field and predict four directions

Problem proposed: The traditional and machine learning-based skin disease diagnosis and treatment methods.

Paper 9

Authors: Aziz A., Hartono R. and Abdilah R.

Year: 2020

Title: Detection of Skin Diseases

Proposed solution :The creation of an expert application of skin disease detection using methods like Naive Bayes, CNN, SVM methods.

Pros and Limitation:The mentioned methods are necessary to help all people who want to know about skin diseases that are being experienced or need information about skin diseases.

Problem proposed: The methods is used in this application is used to detect the skin diseases.

Paper 10

Authors: Konstantin Korotkov, Josep Quintana, Ricard Campos, Am´ericaJes´us-Silva, Pablo Iglesias, Susana Puig

Year: 2019

Title: An Improved Skin Lesion Matching Scheme

Proposed solution :This method makes optimal use of a trained 19-layer deep CNNs and therefore does not rely on prior knowledge of the data.

Pros and Limitation:When using cross entropy as the loss function for picture segmentation, which is a normal process, there is a severe imbalance between both the amount of foreground and background pixels.

Problem proposed: As a result, an original loss function based on Jaccard distance has also been designed to minimise the requirement for sample reweighting.

2.3 PROBLEM STATEMENT DEFINITION

Problem Statement:

Now a day's people are suffering from skin diseases, More than 125 million people suffering from Psoriasis also skin cancer rate is rapidly increasing over the last few decades especially Melanoma is most diversifying skin cancer. If skin diseases are not treated at an earlier stage, then it may lead to complications in the body including spreading of the infection from one individual to the other. The skin diseases can be prevented by investigating the infected region at an early stage. The characteristic of the skin images is diversified so that it is a challenging job to devise an efficient and robust algorithm for automatic detection of skin disease and its severity. Skin tone and skin colour play an important role in skin disease detection. Colour and coarseness of skin are visually different. Automatic processing of such images for skin analysis requires quantitative discriminator to differentiate the diseases.

To overcome the above problem we are building a model which is used for the prevention and early detection of skin cancer, psoriasis. Basically, skin disease diagnosis depends on the different characteristics like colour, shape, texture etc. Here the person can capture the images of skin and then the image will be sent the trained model. The model analyses the image and detect whether the person is having skin disease or not.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Patient	Find the cause for the disease	I can't able to find disease accurately	The model accuracy is low	Frustrated
PS-2	Patient	Figure out if my disease is erythema or not	I can't find any website which gives accurate result	There is no customized websites to detect only erythema and so the accuracy is low	Worried and Irresolute

3. IDEATION & PROPOSED SOLUTION

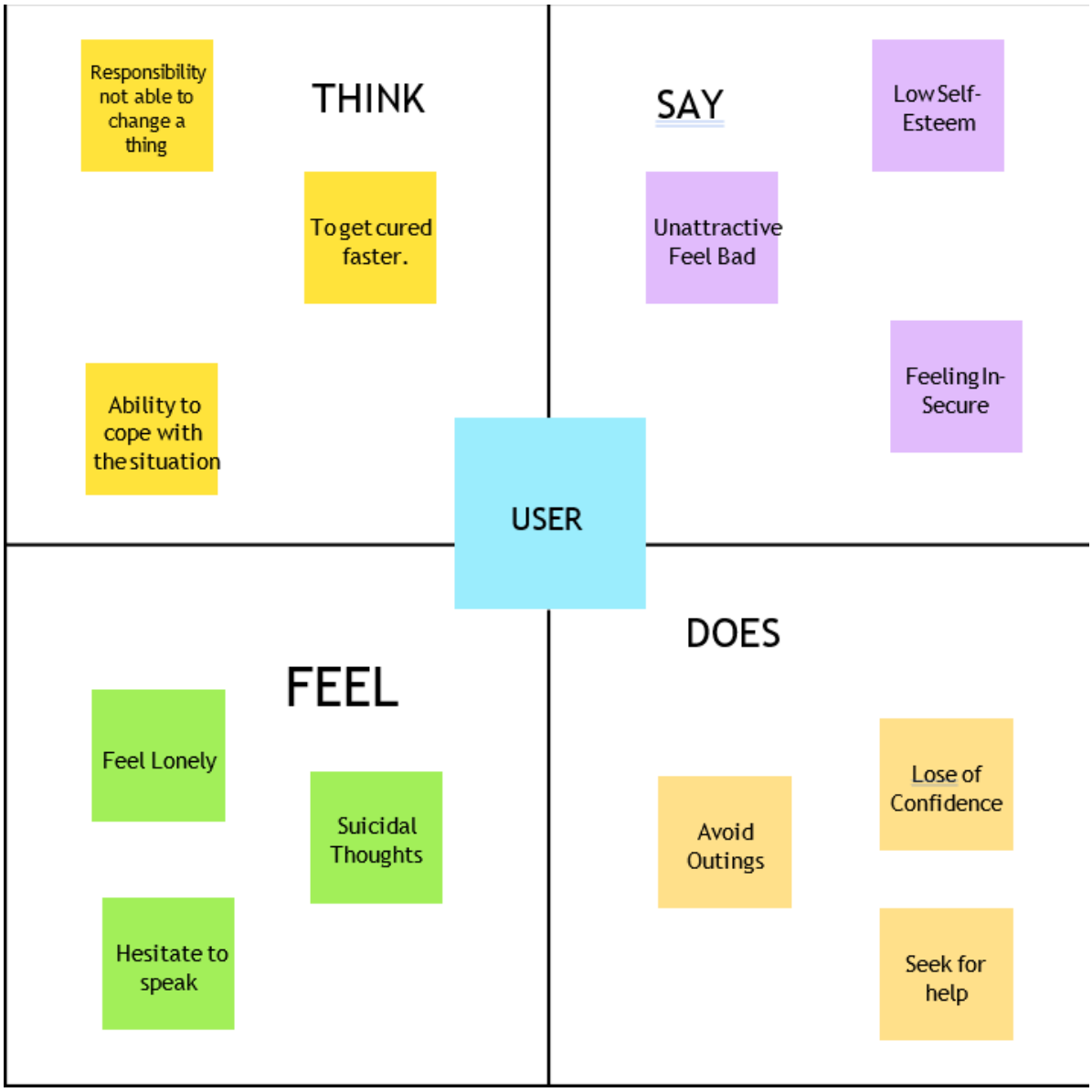
3.1 EMPATHY MAP CANVAS

Empathy Map:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

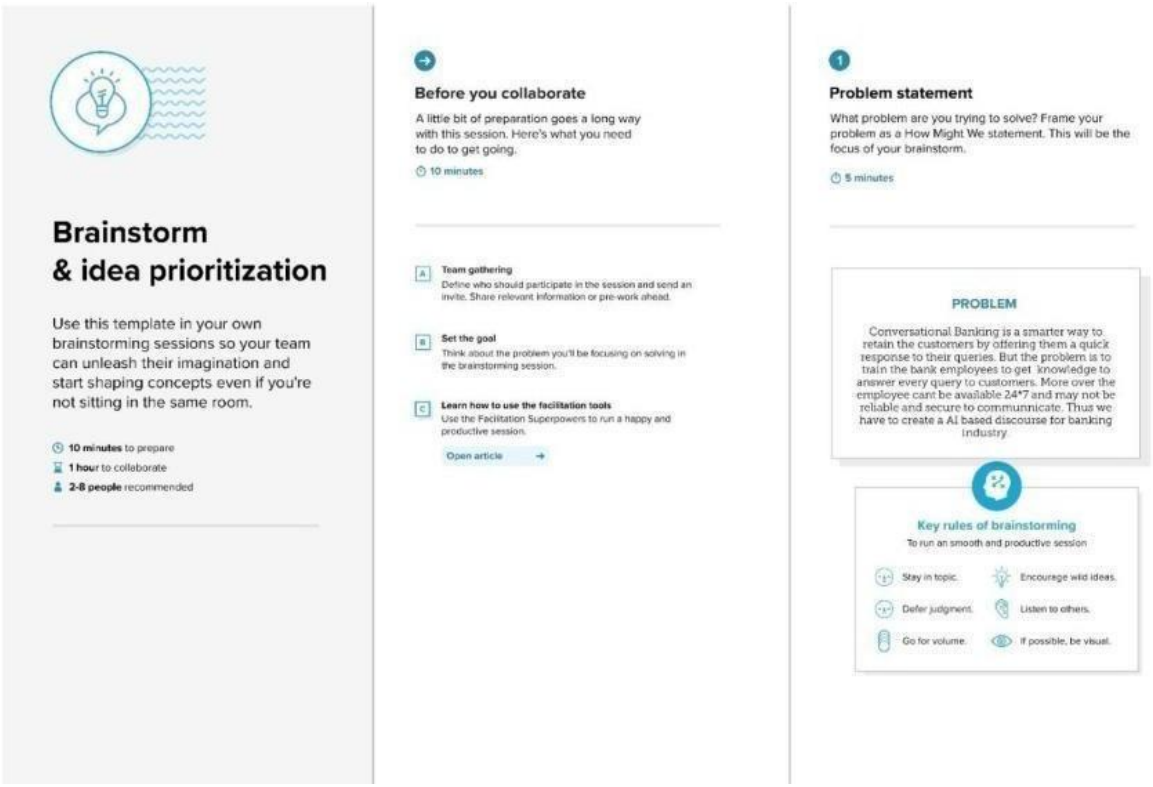
It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 IDEATON & BRAINSTORMING

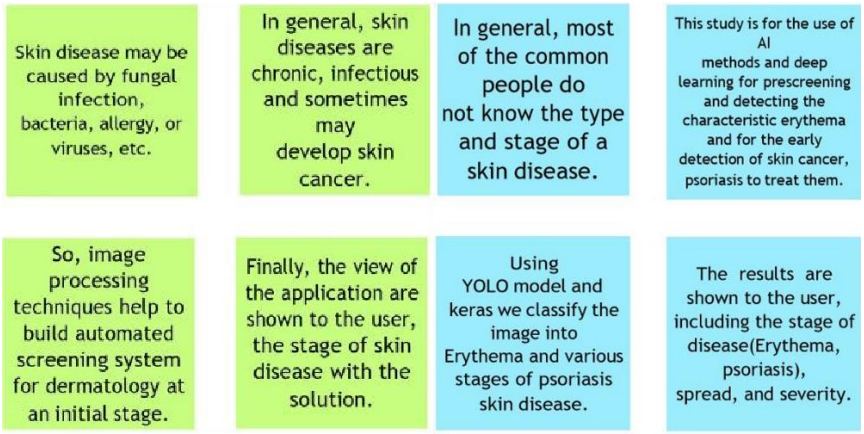
Step-1: Team Gathering, Collaboration and Select the Problem Statement



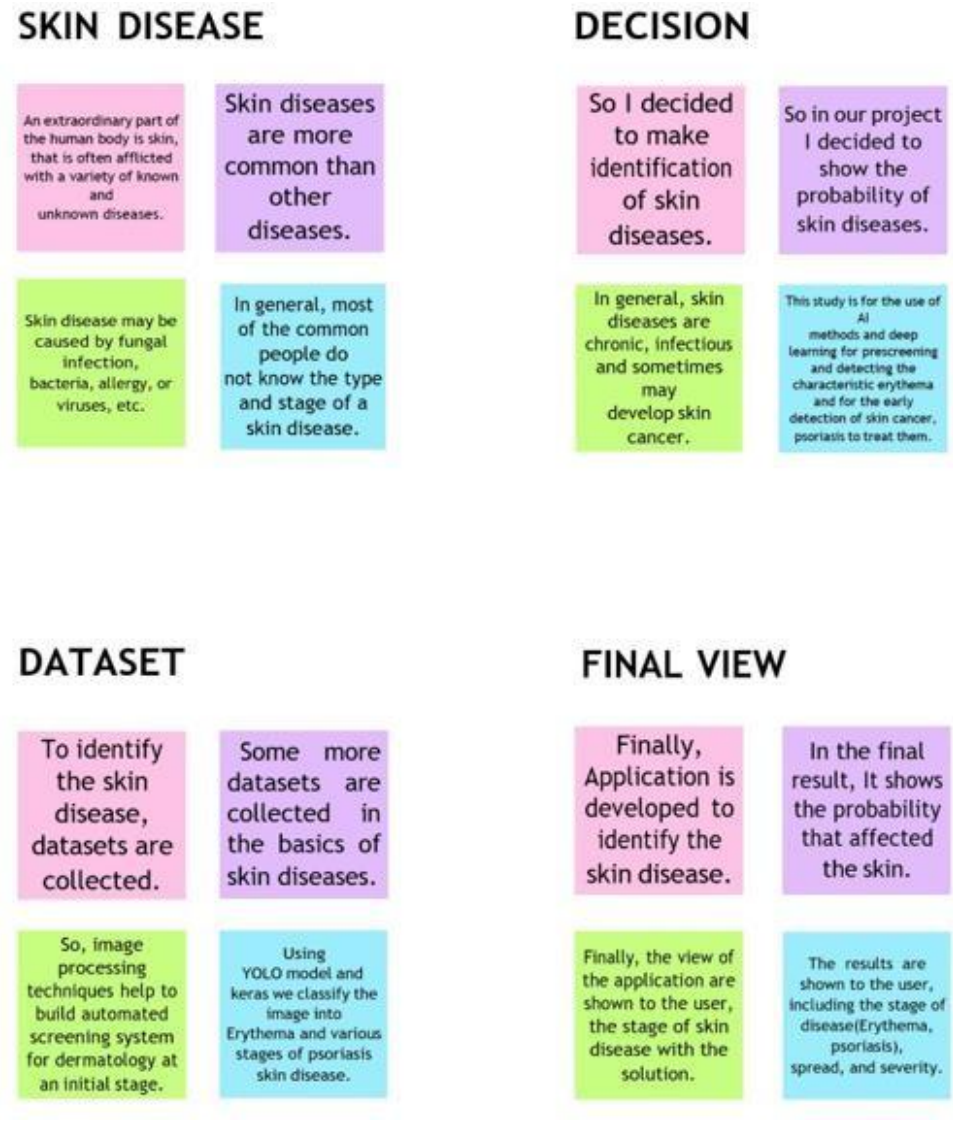
Step-2: Brainstorm, Idea Listing and Grouping

G. Charan Teja **Dharani Vishal S P**

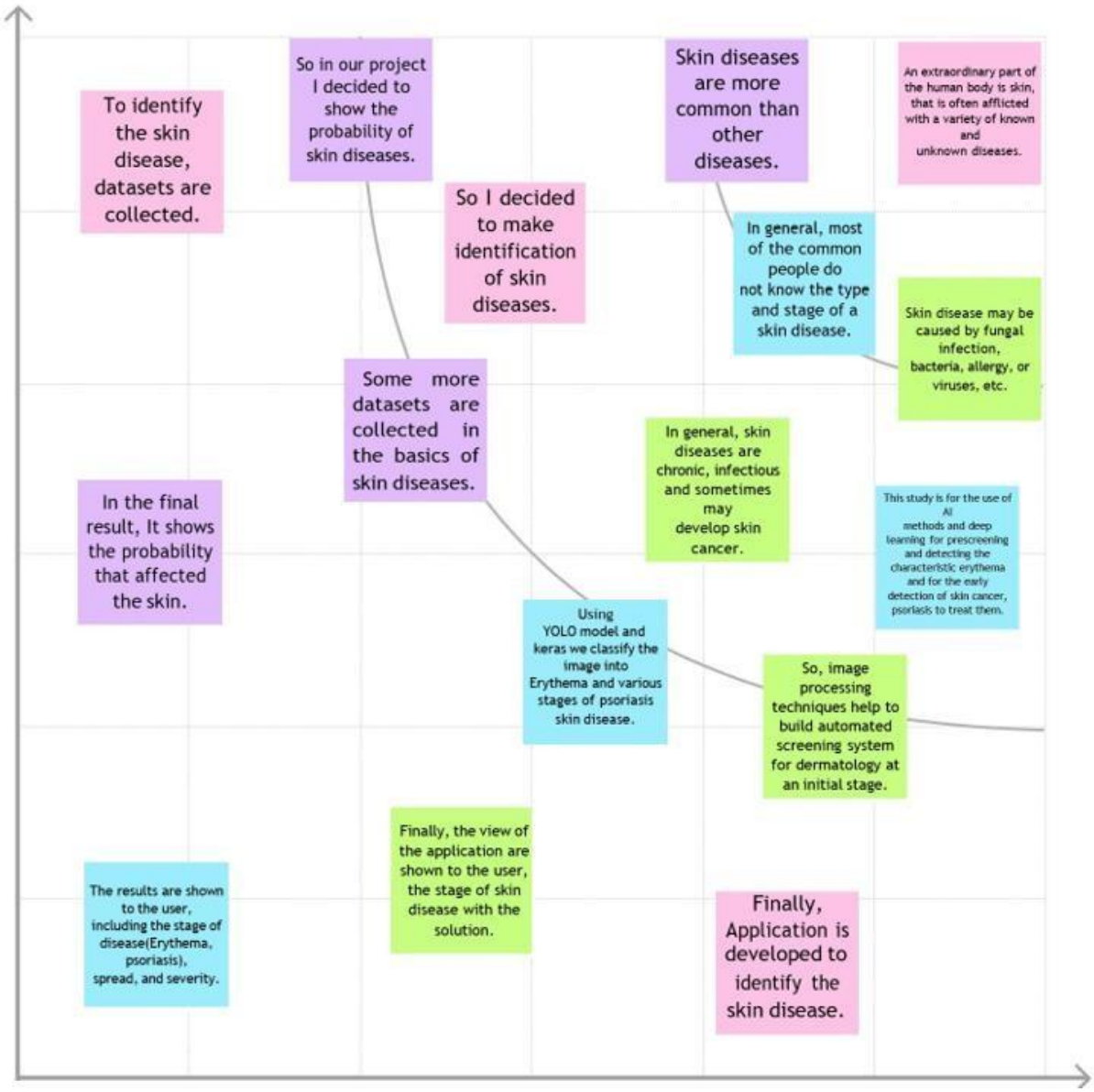
Jerlin T D **Kishore R**



Step-3: Group Ideas



Step-3: Prioritise

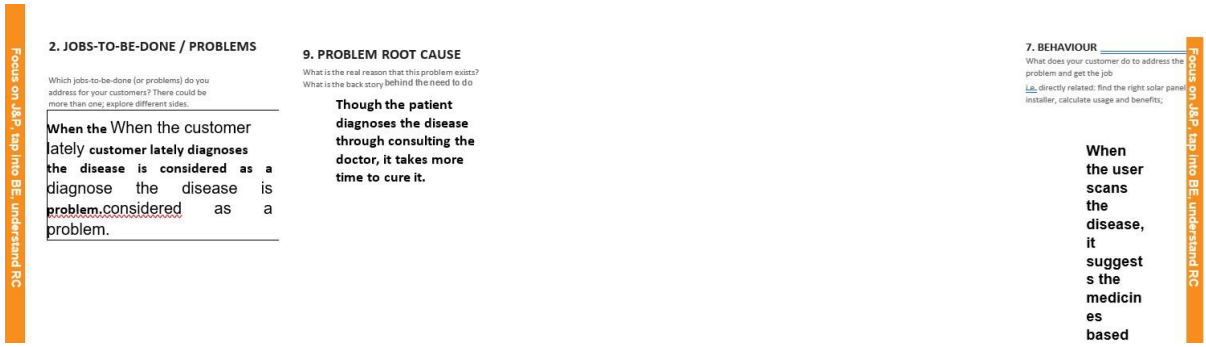


3.3 PROPOSED SOLUTION

Proposed Solution Template:

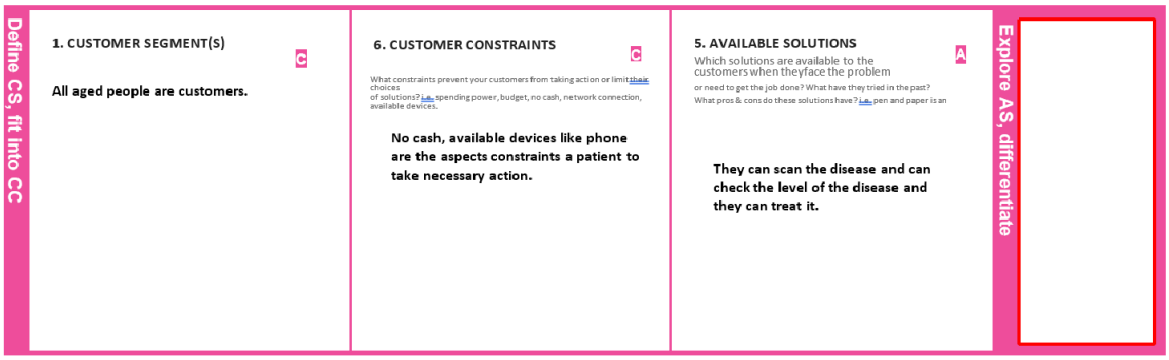
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The user is a busy worker who requires a quick solution with greater precision for his or her skin issue but does not have time to physically visit specialists.
2.	Idea / Solution description	A person has the ability to take pictures of skin, which are then sent to a trained model. The model examines the photograph and determines whether or not the subject has a skin condition.
3.	Novelty / Uniqueness	Additionally, noise-filled images have been captured and improved using powerful algorithms to identify diseases.
4.	Social Impact / Customer Satisfaction	Different skin disorders can be detected by just submitting photographs, and this approach is quite effective at helping people in the community identify ailments earlier.
5.	Business Model (Revenue Model)	Our return on investment will be the creation and distribution of a proprietary product that will be used as a solution.
6.	Scalability of the Solution	This system is more scalable because it accepts any picture type, regardless of resolution, and offers good performance in any situation.

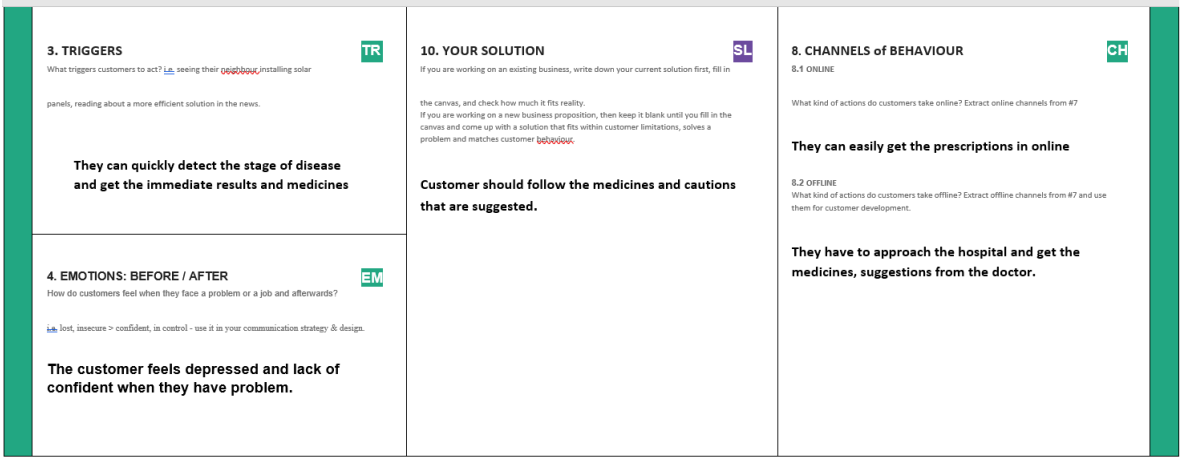
3.4 PROPOSED SOLUTION FIT



Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID28659





4.REQUIREMENT ANALYSIS

4.1 FUNCTONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Registration process**	Registration through Mobile Number Registration through Google Account Registration through Facebook
FR-2	**User Confirmation**	Confirmation via E-mail Confirmation via Call Confirmation via One Time Password
FR-3	**Patient Image Capturing Process**	Provide Access to Capture Image Through Camera Provide Access to Upload Image Through Gallery
FR-4	**Patient Medicine Reminder**	Remind the Patients to take their Medicines/ointments At right time through remaindering alarm.
FR-5	**Suggestion Box**	Patients can take suggestions from the Doctors through Chats.
FR-6	**Flareup Cycles**	Patients can know their medicine level from doctors Through message.

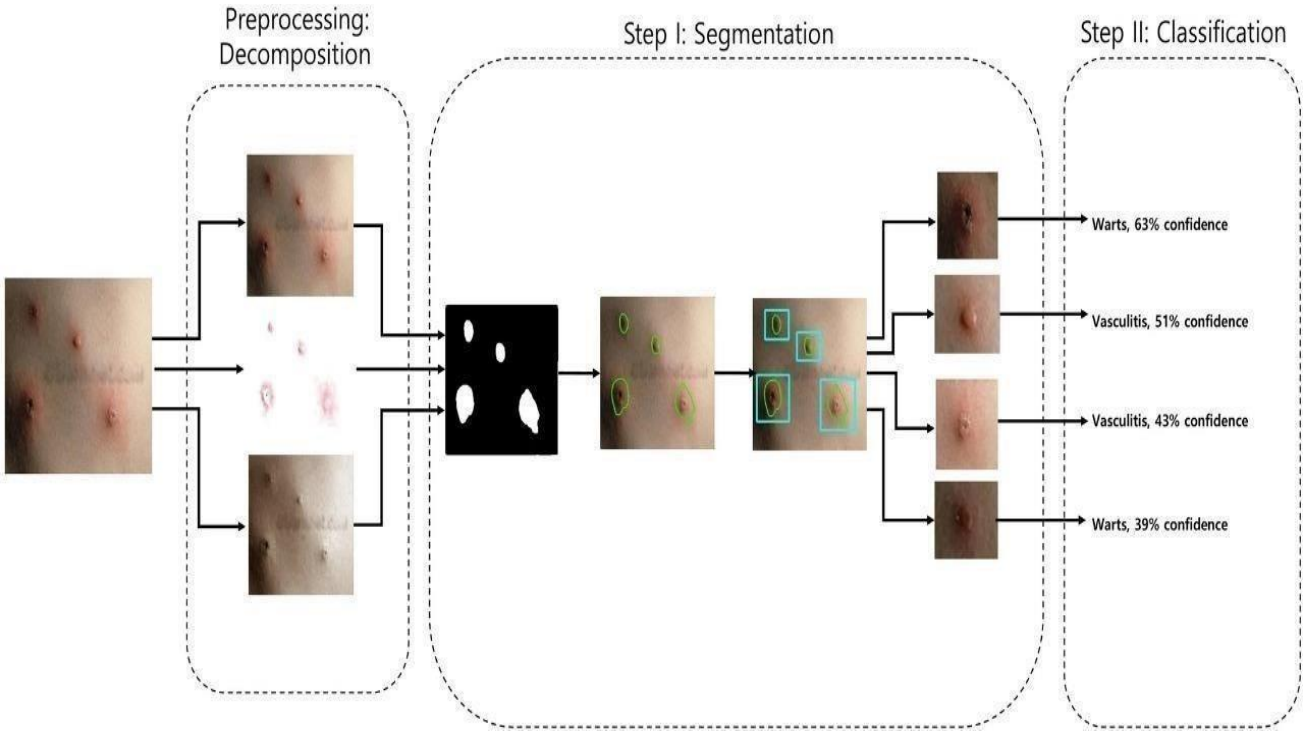
4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It doesn't specify parts of the system functionality, only how that functionality is to be perceived by the user
NFR-2	**Security**	Data privacy and security practices may vary based on users and their age
NFR-3	**Reliability**	Extent to which the software system consistently performs the specified functions without failure.
NFR-4	**Performance**	He website's load time should not be more than one second for users
NFR-5	**Availability**	How likely the system is accessible to a user at a given point in time
NFR-6	**Scalability**	The ability to appropriately handle increasing (and decreasing) workloads.

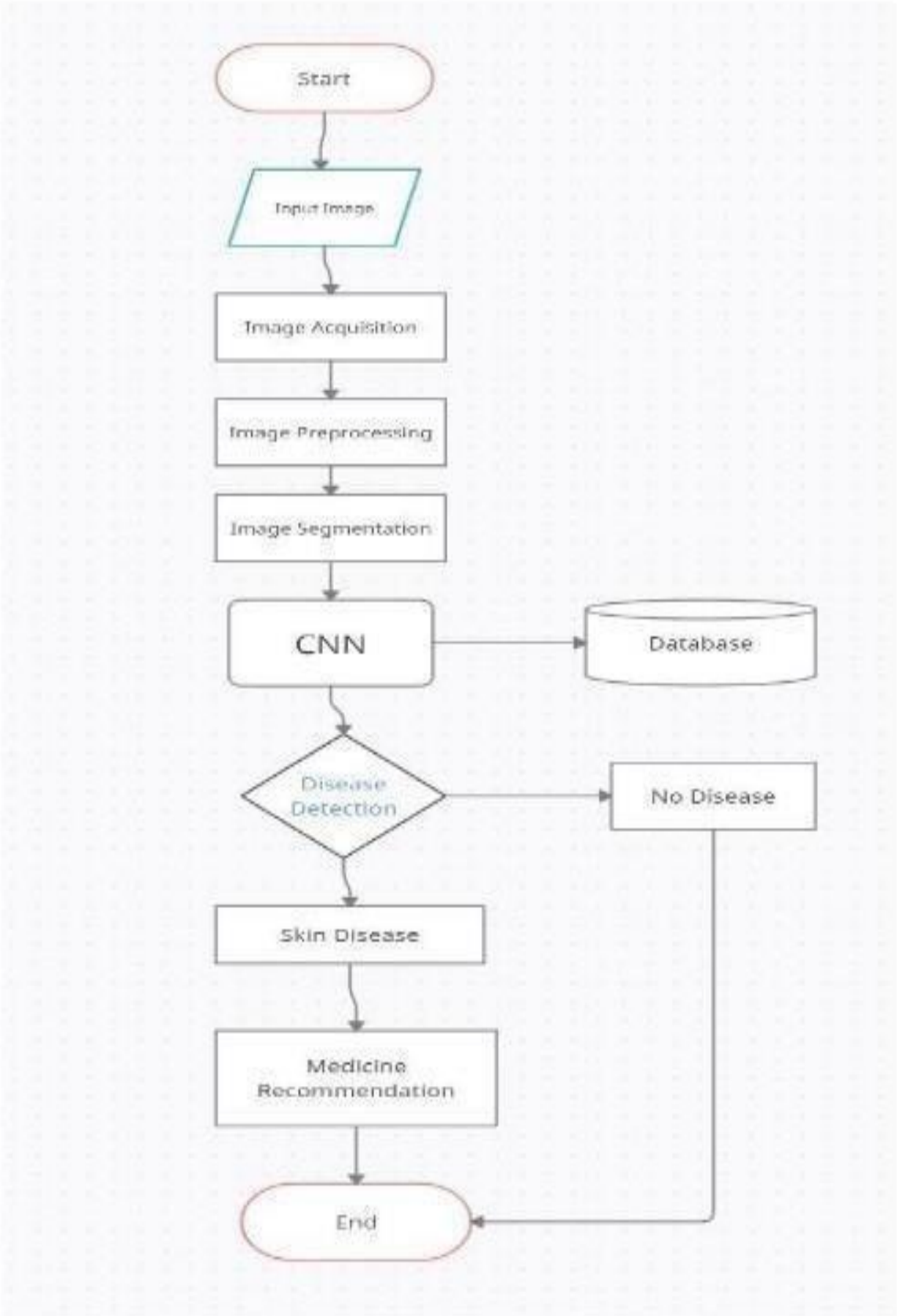
5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



Example: DFD Level 0 (Industry Standard)


Start
Input Image
Image Acquisition
Image Preprocessing
Image Segmentation
CNN
Database
Disease Detection
No Disease
Skin Disease
Medicine Recommendation
End

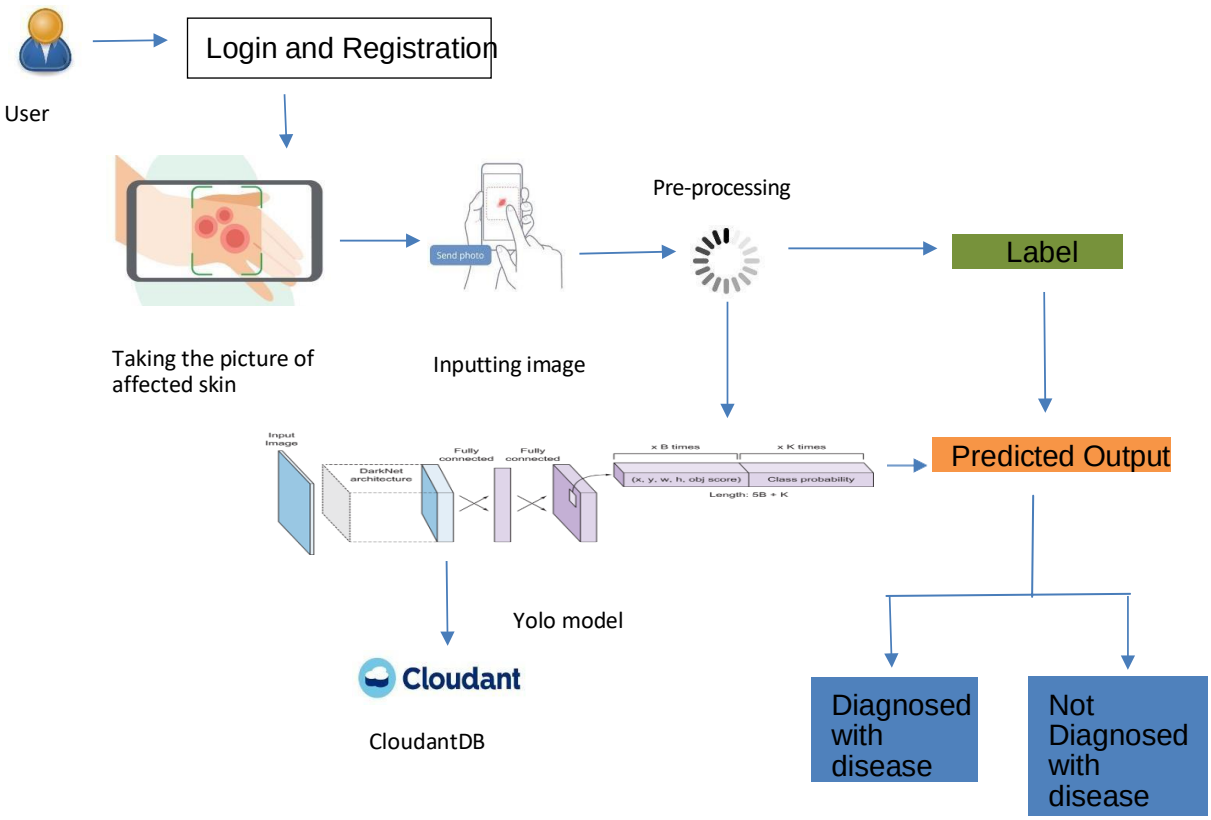
5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Solution Architecture:

The process of bridging the gap between business issues and technological solutions, known as solution architecture, is intricate and has numerous sub-processes. Its aims include:

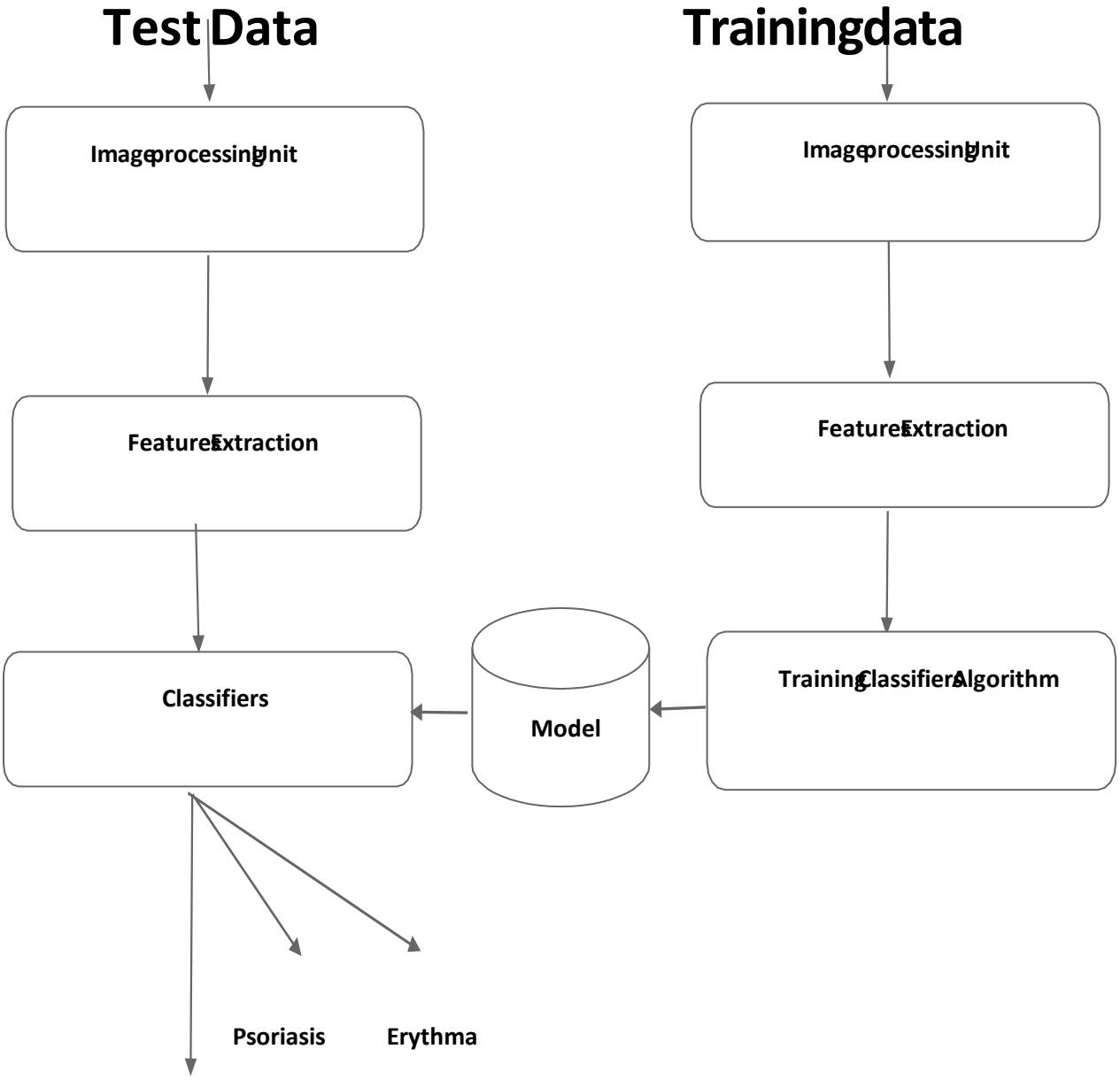
1. Find the greatest technological remedy to address current company issues.
2. Describe to the project's stakeholders the software's structure, traits, behaviour, and other features.
3. Define the solution's requirements, development stages, and features..
4. Specifications on how the solution is defined, maintained, and delivered should be provided.

Solution Architecture Diagram:



Architecture and data flow of the skin disease(erythema) prediction application

Technical Architecture:



Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
	USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
Dashboard	USN-5	As a user, I can Access my Dashboard.		Medium	Sprint-3
Registration process	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	sprint-4
Result	USN-5	Responding to each email you receive can make a lasting impression on customers.	Offer a solution for how your company can improve the customer's experience.	High	Sprint-3
Managing	USN-5	Do-it-yourself service for delivering Everything.	set of predefined requirements that must be met to mark a user story complete.	High	sprint-4

6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	1	High	Heymanth GD Mohanadinesh Theophilus ShyamSewag
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Heymanth GD Mohanadinesh Theophilus ShyamSewag
Sprint-1		USN-3	As a user, I can register for the application through Facebook	2	Low	Heymanth GD Mohanadinesh Theophilus ShyamSewag
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Heymanth GD Mohanadinesh Theophilus ShyamSewag
Sprint-2	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Heymanth GD Mohanadinesh Theophilus ShyamSewag
Sprint-2		USN-6	As a user, I can log into the application by entering username & password	2	Medium	Heymanth GD Mohanadinesh Theophilus ShyamSewag
Sprint-3	Dashboard	USN-7	Home page displays the problem statement	2	Medium	Heymanth GD Mohanadinesh Theophilus ShyamSewag
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**

Sprint-3		USN-8	The home page contains the demo video through which the user can learn how to use the system.	1	High	Heymanth GD Mohanadinesh Theophilus ShyamSewag
Sprint-4	Analysis	USN-9	Image is uploaded from the gallery and it is predicted	1	High	Heymanth GD Mohanadinesh Theophilus ShyamSewag
Sprint-4		USN-10	Through the webcam a image is captured and skin disease is predicted	2	Low	Heymanth GD Mohanadinesh Theophilus ShyamSewag
Sprint-4		USN-11	Diseases will be predicted with its accuracy rate and be displayed on the screen.	1	High	Heymanth GD Mohanadinesh Theophilus ShyamSewag
Sprint-2	Logout	USN-12	Once the disease is detected the user will logout the system	2	Medium	Heymanth GD Mohanadinesh Theophilus ShyamSewag

6.2 SPRINT DELIVERY SCHEDULE

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	27 Oct 2022
Sprint-2	20	6 Days	27 Oct 2022	05 Nov 2022	20	30 Oct 2022
Sprint-3	20	6 Days	29 Nov 2022	12 Nov 2022	20	02 Nov 2022
Sprint-4	20	6 Days	02 Nov 2022	19 Nov 2022	20	05 Nov 2022

7.CODING & SOLUTIONING

7.1 FEATURE 1

1. HTML for web page making
2. CSS, js, vendor for static design content
3. Python for the web server integration
4. Java script for uploading pictures.

7.2 FEATURE 2

1.Creating IBM cloud account

2.Accesing IBM database

3.Storing User credentials

8.TESTING

8.1 TEST CASES

8.1 Test Cases

- ✞ Verify user is able to see home page.
- ✞ Verify user is able to see Dashboard page.
- ✞ Verify user is able to navigate to Report page.
- ✞ Verify filters are working

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

8.2 USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

3. Test Case Analysis

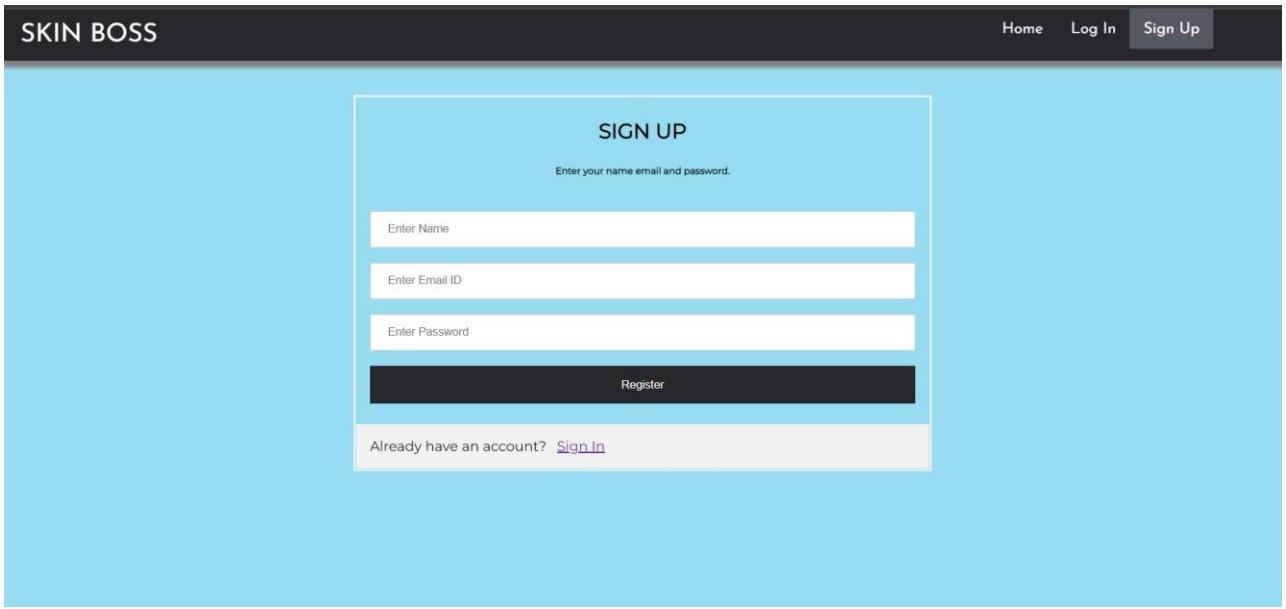
This report shows the number of test cases that have passed, failed and understand

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

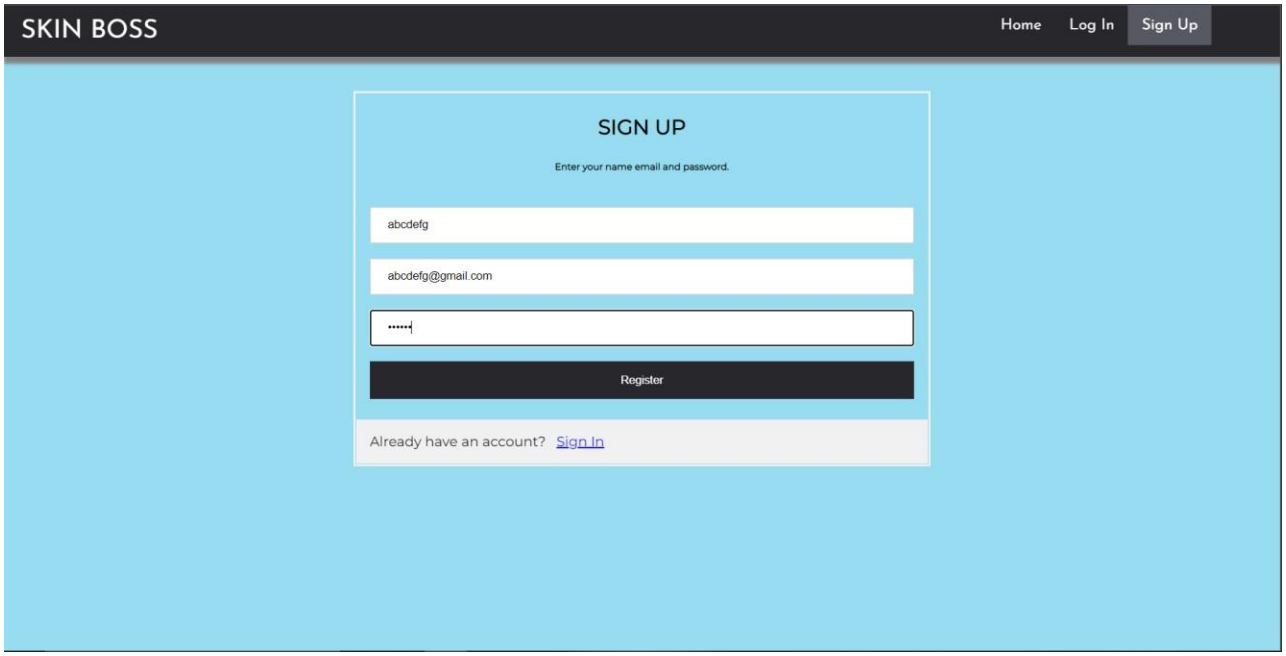
9.RESULTS

9.1 PERFORMANCE METRICS

Registration page:

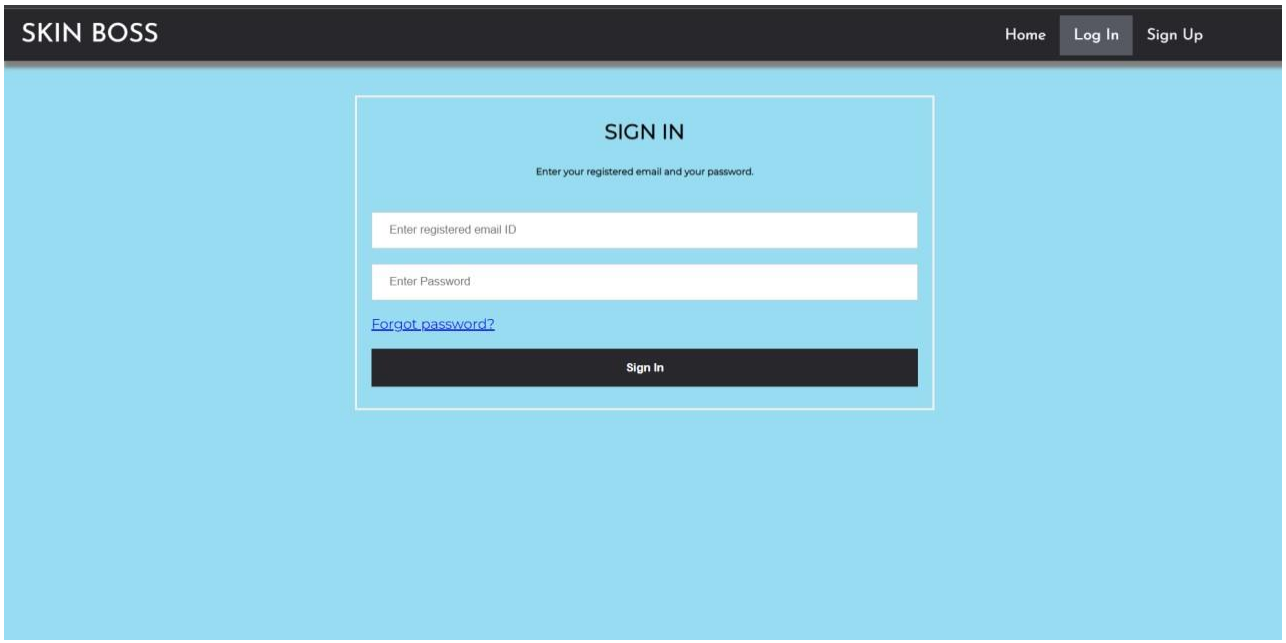


Creating a Account:

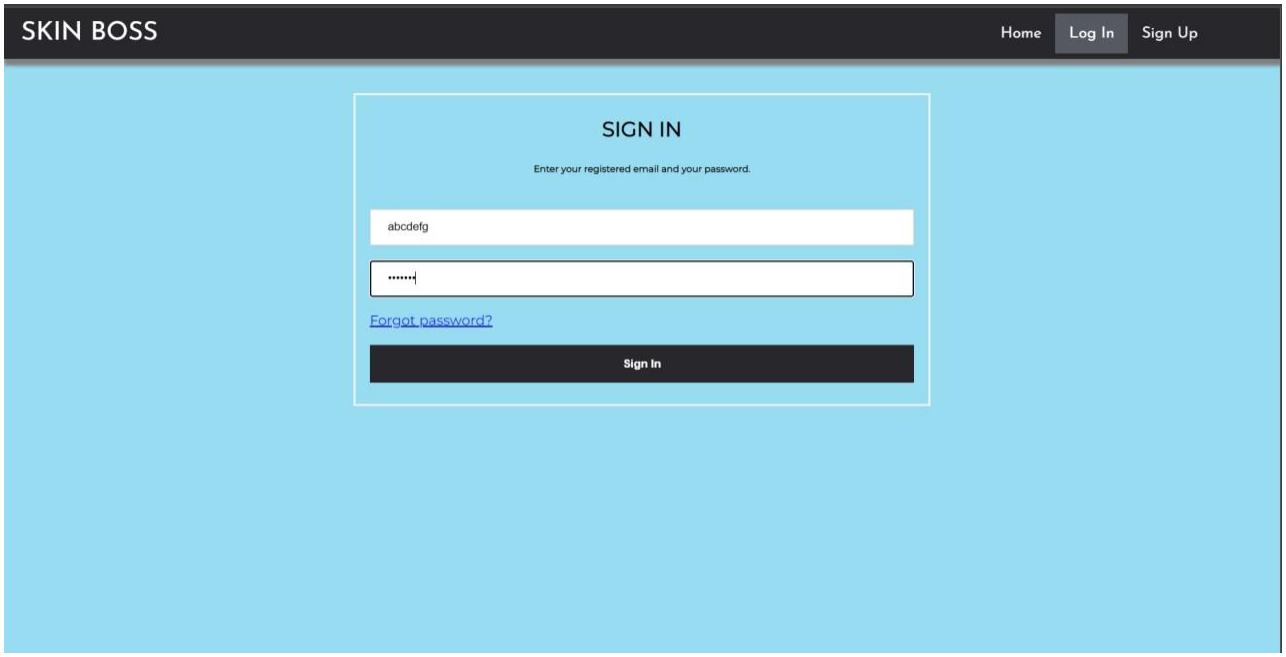


With this the credentials of user will be updated to the IBM cloud database for further use.

LogIn page:

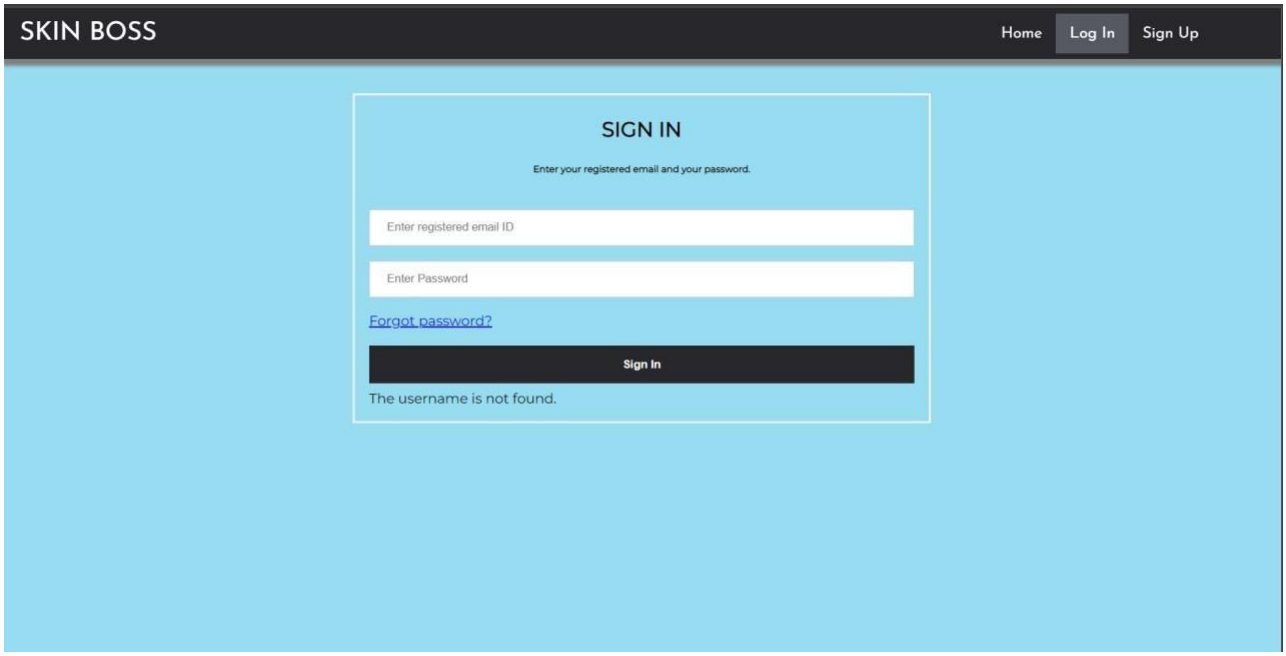


Logging into the Account:

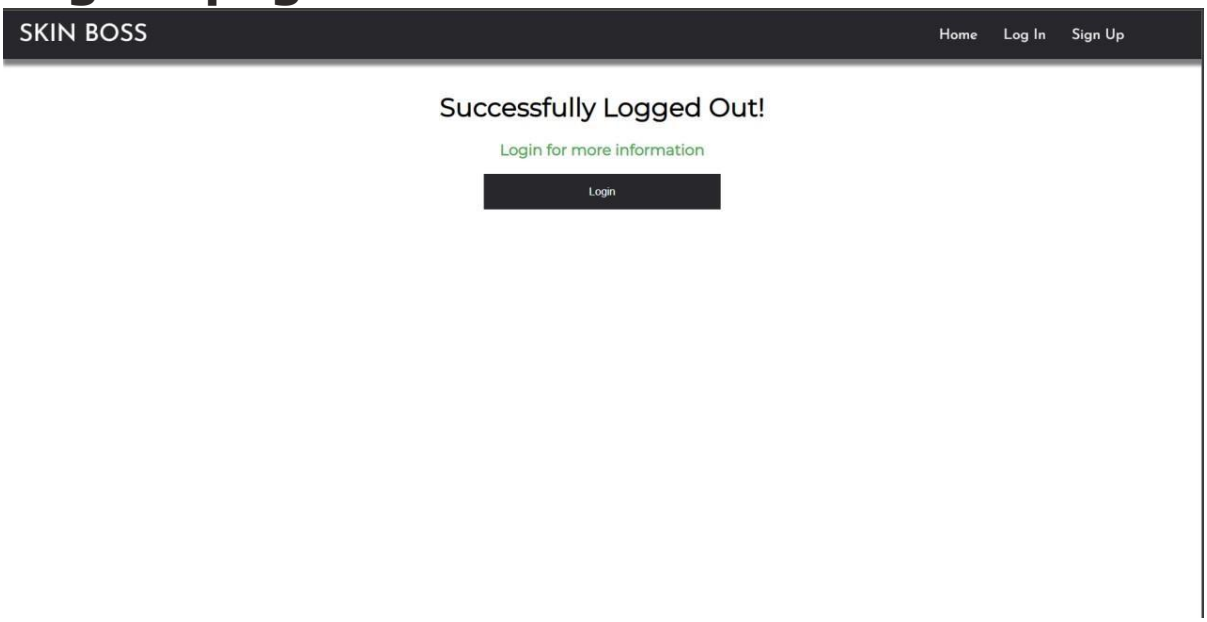


After entered the credentials correctly it will redirect to Prediction page for Analysis but is the credentials
are not correct it will through message as "username not found"

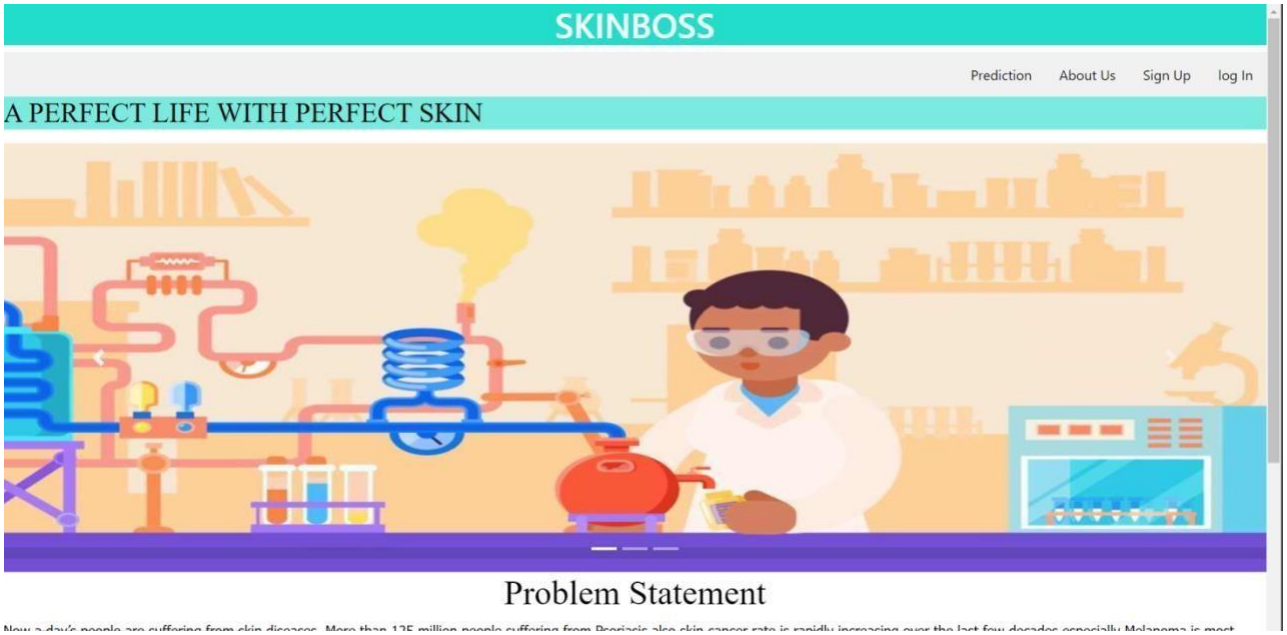
Username not found page:

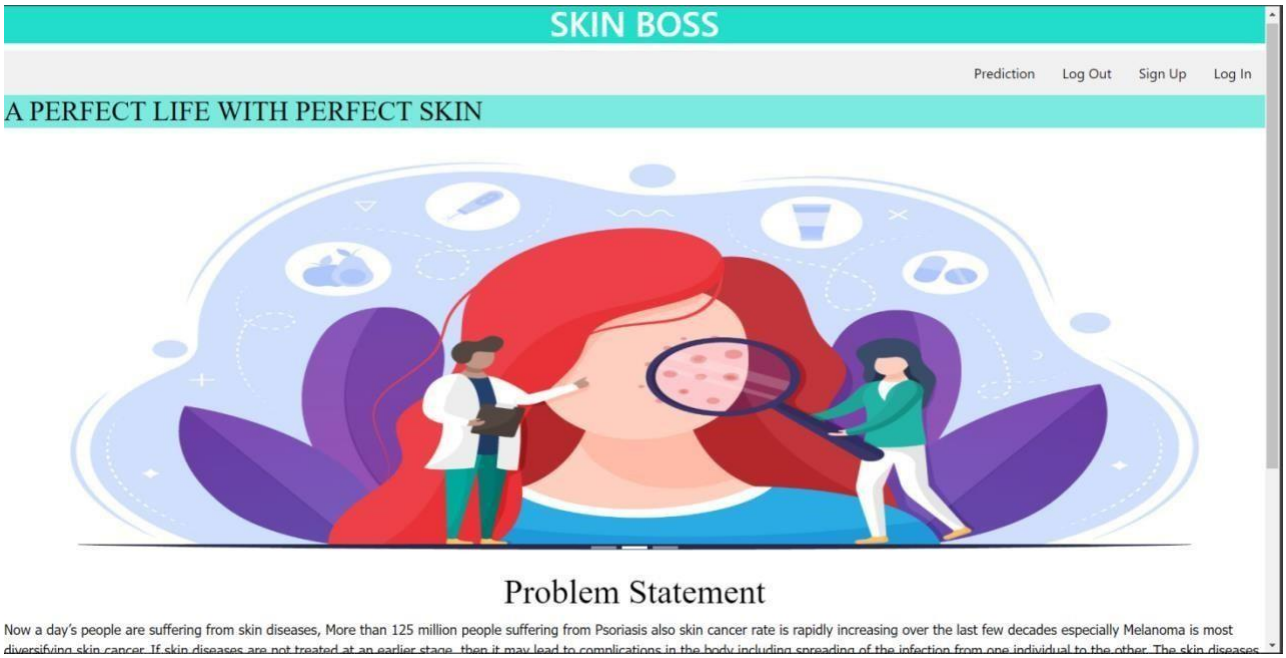


Logout page:



Dash board(Home Page):

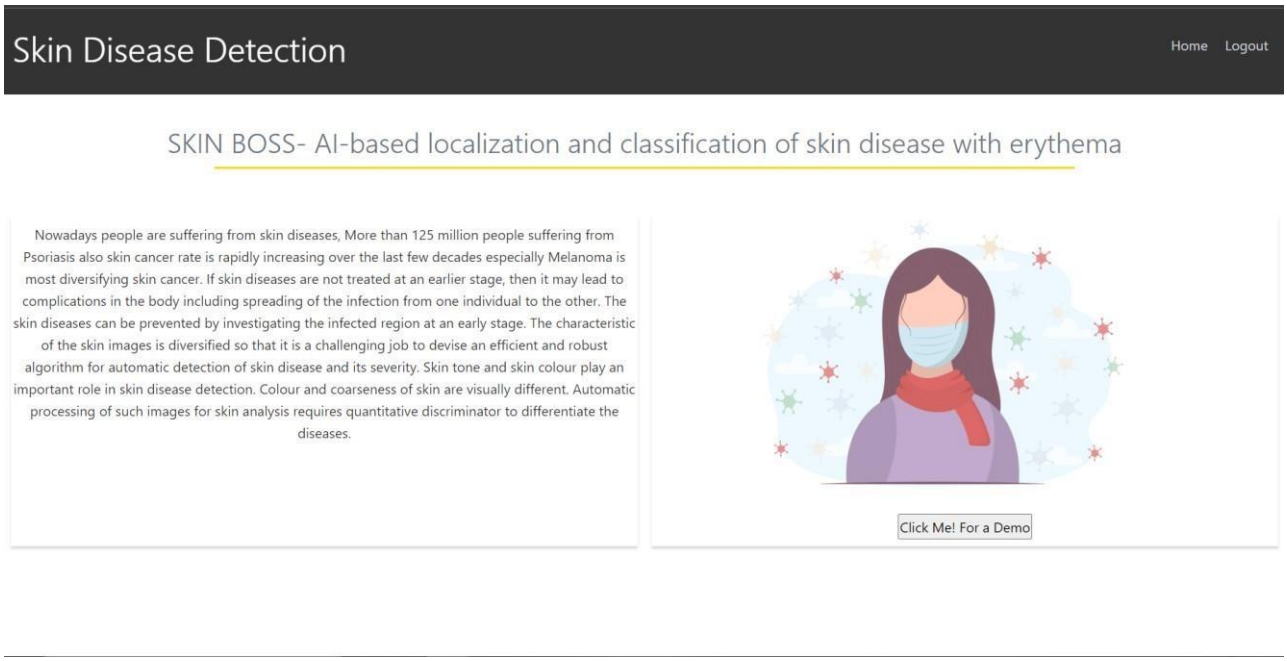




This is the Home page and from this page the user can directed to any page they want with proper credentials.

Prediction Page:

As we all know this is the important page for this website , we have created this page to predict the skin disease that user where affected to and to provide them the results.



In this page we have provided with a "click me! For a Demo" button by clicking on to this button the prediction YOLO structure software will run and gives the results.

10. ADVANTAGES & DISADVANTAGES

Advantages of using chatbots in banking customer service

- It Provides a free basic dermoscopic tutorial for the general public.
- This system can be used by dermatologists to give a better diagnosis and treatment to the patients.
- The system can be used to diagnose skin diseases at a lower cost.
- This system can be used to detect and classify diseases as well as their severity.

Disadvantages of using chatbots in banking customer service

- Internet connectivity is the major issue.
- In case of low Quality images it may show inaccurate information.
- Everyone is not aware of this application

11 CONCLUSION

The system proposed is a Skin Disease Detection System. This system uses test images to detect if it is healthy or not; if not, then classified as Melanoma, Psoriasis or Rosacea.

This system can be used by dermatologists to give a better diagnosis and treatment to the patients. The system can be used to diagnose skin diseases at a lower cost.

In future, this system can be improved to detect and classify more diseases as wellas their severity.

12. FUTURE SCOPE

In future, this machine learning model may bind with various websites which can provide real time data for skin disease prediction. Also, we may add large historical data on skin disease which can help to improve the accuracy of the machine learning model. We can build an android app as a user interface for interacting with the user. For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates, and train it on clusters of data rather than the whole dataset.

13. APPENDIX

SOURCE CODE

PYTHON

The popular Python frameworks used by developers for web development is Flask. In this article, you will get introduced to Python framework. Along with this, we will also see some of the basic implementations along with some HTTP methods.

```
import requests
import numpy as np
import os
from flask import Flask, app, request, render_template
import sys
from flask import Flask, request, render_template, redirect, url_for
import argparse

from keras_yolo3.yolo import YOLO
from keras_yolo3.yolo import detect_video
from tensorflow import keras
from PIL import Image
from timeit import default_timer as timer
import tests
import pandas as pd
import numpy as np
import random


def get_parent_dir(n=1):
    """ returns the n-th parent dicrectory of the current
    working directory """
    current_path = os.path.dirname(os.path.abspath(__file__))
    for k in range(n):
        current_path = os.path.dirname(current_path)
    return current_path


src_path =r'C:\Users\sujat\Desktop\yolo_structure-master\2_Training\src'
print(src_path)
```

```
src_path =r'C:\Users\sujat\Desktop\yolo_structure-master\2_Training\src'
print(src_path)
utils_path =r'C:\Users\sujat\Desktop\yolo_structure-master\utils'
print(utils_path)

sys.path.append(src_path)
sys.path.append(utils_path)

import argparse
from keras_yolo3 import yolo, yolo_video
from PIL import Image
from timeit import default_timer as timer
from Utils.utils import load_extractor_model, load_features, parse_input, detect_object
import tests
import utils
import pandas as pd
import numpy as np
from Get_File_Paths import GetFileList
import random
import yolo

os.environ["TF_CPP_MIN_LOG_LEVEL"] = "3"

# Set up folder names for default values
data_folder = os.path.join(get_parent_dir(n=1), "yolo_structure-master","Data")
image_folder = os.path.join(data_folder,
                            "Source_Images")

image_test_folder = os.path.join(image_folder, "Training_images")
```

```
detection_results_folder = os.path.join(image_folder, "Test_Image_Detection_Results")
detection_results_file = os.path.join(detection_results_folder, "Detection_Results.csv")

model_folder = os.path.join(data_folder, "Model_Weights")

model_weights = os.path.join(model_folder, "trained_weights_final.h5")
model_classes = os.path.join(model_folder, "data_classes.txt")

anchors_path = os.path.join(src_path, "keras_yolo3", "model_data", "yolo_anchors.txt")

FLAGS = None


from cloudant.client import Cloudant

# Authenticate using an IAM API key
client = Cloudant.iam('27d9e818-b901-49b0-a700-83fefb928605-bluemix','np6eucoskfwrKd5RPnidhxGwKAQsI-09-LlBkdYx

# Create a database using an initialized client
my_database = client.create_database('my_database')


app=Flask(__name__)

#default home page or route
@app.route('/')
def index():
    return render_template('home.html')
```

```python
@app.route('/home.html')
def home():
    return render_template("home.html")


#registration page
@app.route('/signup')
def signup():
    return render_template('signup.html')

@app.route('/afterreg', methods=['POST'])
def afterreg():
    x = [x for x in request.form.values()]
    print(x)
    data = {
    '_id': x[1], # Setting _id is optional
    'name': x[0],
    'psw':x[2]
    }
    print(data)

    query = {'_id': {'$eq': data['_id']}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))

    if(len(docs.all())==0):
        url = my_database.create_document(data)
```

```
        #response = requests.get(url)
        return render_template('signup.html', pred="Registration Successful, please login using your details")
    else:
        return render_template('signup.html', pred="You are already a member, please login using your details")

#login page
@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/afterlogin',methods=['POST'])
def afterlogin():
    user = request.form['_id']
    passw = request.form['psw']
    print(user,passw)

    query = {'_id': {'$eq': user}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))


    if(len(docs.all())==0):
        return render_template('login.html', pred="The username is not found.")
    else:
        if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
            return redirect(url_for('prediction'))
        else:
```

```
    input_video_paths = []
    for item in input_paths:
        if item.endswith(img_endings):
            input_image_paths.append(item)
        elif item.endswith(vid_endings):
            input_video_paths.append(item)

    output_path = FLAGS.output
    if not os.path.exists(output_path):
        os.makedirs(output_path)

    # define YOLO detector
    yolo = YOLO(
        **{
            "model_path": FLAGS.model_path,
            "anchors_path": FLAGS.anchors_path,
            "classes_path": FLAGS.classes_path,
            "score": FLAGS.score,
            "gpu_num": FLAGS.gpu_num,
            "model_image_size": (416, 416),
        }
    )

    # Make a dataframe for the prediction outputs
    out_df = pd.DataFrame(
        columns=[
            "image",
            "image_path",
            "xmin",
            "ymin",
```

```
    if input_video_paths:
        print(
            "Found {} input videos: {} ...".format(
                len(input_video_paths),
                [os.path.basename(f) for f in input_video_paths[:5]],
            )
        )
        start = timer()
        for i, vid_path in enumerate(input_video_paths):
            output_path = os.path.join(
                FLAGS.output,
                os.path.basename(vid_path).replace(".", FLAGS.postfix + "."),
            )
            detect_video(yolo, vid_path, output_path=output_path)

        end = timer()
        print(
            "Processed {} videos in {:.1f}sec".format(
                len(input_video_paths), end - start
            )
        )
    # Close the current yolo session
    yolo.close_session()
    return render_template('prediction.html')


""" Running our application """
if __name__ == "__main__":
    app.run(debug=True)
```

WEBSITE BASED CODES

Home page:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">


    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwF
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHf


    <script src="https://kit.fontawesome.com/8b9cdc2059.js" crossorigin="anonymous"></script>
    <link href="https://fonts.googleapis.com/css2?family=Akronim&family=Times New Roman&display=swap" rel="stylesheet">

    <title>SKINBOSS</title>
</head>
<body>
<style>
.icon-bar {
  width: 90px; /* Set a specific width */
  background-color: #555; /* Dark-grey background */
}

.icon-bar a {
  display: block; /* Make the links appear below each other instead of side-by-side */
  text-align: center; /* Center-align text */
  padding: 16px; /* Add some padding */

</style>
    <header id="head" class="header">

   <section id="navbar">
           <h1 class="heading">SKIN BOSS</h1>
       <div class="nav--items">

    <a href="{{url_for('login')}}">Log In</a>
    <a href="{{url_for('signup')}}">Sign Up</a>
              <a href="{{url_for('logout')}}">Log Out</a>
               <a href="{{url_for('prediction')}}">Prediction</a>
       </div>
   </section>

   <div class="top">
    <h2 class="title text-muted">
        <p style="...">A PERFECT LIFE WITH PERFECT SKIN</p>
    </h2>

</div>
   <section id="slider">
    <div id="carouselExampleIndicators" class="carousel" data-ride="carousel">
        <ol class="carousel-indicators ">
            <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active "></li>
            <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
            <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
        </ol>

        <div class="carousel-item active">
```

```
        </ol>

        <div class="carousel-item active">
            <img class="d-block w-100" src="https://regencyhealthcare.in/wp-content/uploads/2021/04/Artboard-5-
        </div>
        <div class="carousel-item">
            <img alt="Second slide" class="d-block w-100" src="https://img.graphicsurf.com/2020/10/skin-disease
        </div>

        <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">
            <span class="carousel-control-prev-icon" aria-hidden="true"></span>
            <span class="sr-only">Previous</span>
        </a>
        <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">
            <span class="carousel-control-next-icon" aria-hidden="true"></span>
            <span class="sr-only">Next</span>
        </a>
    </div>
       <div class="Problem">
           <h1 class="head">Problem Statement</h1>
           <p style="...">
               Now a day's people are suffering from skin diseases, More than 125 million people suffering fro

               To overcome the above problem we are building a model which is used for the prevention and early
           <h1 class="head">Proposed Solution</h1>
           <p style="...">
               Different skin disorders can be detected by just
               submitting photographs, and this approach is
               quite effective at helping people in the
```

Login Page:

```
<!DOCTYPE html>
<html >

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>SKIN BOSS</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>

<style>
.header {
           top:0;
           margin:0px;
           left: 0px;
           right: 0px;
           position: fixed;
           background-color: #28272c;
           color: white;
           box-shadow: 0px 8px 4px grey;
           overflow: hidden;
           padding-left:20px;
           font-family: 'Josefin Sans';
```

```
<body style="...">

<div class="header">
 <div style="...">SKIN BOSS</div>
  <div class="topnav-right" >

    <a  href="{{url_for('home')}}">Home</a>
    <a href="{{url_for('login')}}">Log In</a>
    <a class="active"  href="{{url_for('signup')}}">Sign Up</a>

  </div>
</div>
<div id="SignIn" class="SignIn">


    <form action="{{url_for('afterreg')}}" method="post">
        <div class="imgcontainer">
      <h2>SIGN UP</h2>
      <h6 class="information-text">Enter your name email and password.</h6>
        </div>

        <div class="container">
            <input type="text" placeholder="Enter Name" name="name" required><br>
            <input type="email" placeholder="Enter Email ID" name="_id" required><br>
            <input type="password" placeholder="Enter Password" name="psw" required>

            <button type="submit">Register</button><br>

        </div>
```

The other documents and html soruce codes are available in githup.

The attachments that is supported for the execution of the of this html and python file is upload in git repo the link is provided below.

GITHUB & PROJECT DEMO LINK:

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-21950-1659798894/tree/main/Final%20Deliverables/Project

PROJECT DEMO LINK:

https://drive.google.com/drive/folders/1dyN63we05PsWuIr4laXvymE_km0aKDNt