

SKIN CARE ROUTINE



AUTONOMOUS

A MINI PROJECT REPORT

Submitted by

ARINI GK (710721205005)

MULLAIVENDHAN A (710721205030)

SREE VISHNU M (710721205051)

in partial fulfilment of the award of the degree

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Dr. N.G.P. INSTITUTE OF TECHNOLOGY, COIMBATORE -641048

(AN AUTONOMOUS INSTITUTION)

ANNA UNIVERSITY:: CHENNAI 600 025

APRIL 2023

BONAFIDE CERTIFICATE

Certified that this project report " SKIN CARE ROUTINE " is the Bonafide work of "ARINI GK(710721205005), MULLAIVENDHAN A (710721205030), SREE VISHNU M (710721205051), "who carried out the project work under my supervision.

SIGNATURE

Dr. M. KRISHNAMOORTHY M.E., Ph.D.,
HEAD OF THE DEPARTMENT

Department of Information Technology,
Dr. N. G. P Institute of Technology,
Kalapatti Road,
Coimbatore-641 048.

Submitted for the End Semester Project Viva-Voce held on

SIGNATURE

Ms.R.RATHIYA AP/IT,
SUPERVISOR

Assistant Professor (SG),
Department of Information Technology,
Dr. N.G.P Institute of Technology,
Kalapatti Road,
Coimbatore-641 048.

INTERNAL EXAMINER EXTERNAL EXAMINER ACKNOWLEDGEMENT

we extend our heartiest thanks to Dr. NALLA G. PALANISWAMI, Chairman, KMCH & Dr. N.G.P. Educational Institutions for providing us the necessary infrastructure to do our project work.

we express our gratitude to Dr. THAVAMANI D. PALANISWAMI, Secretary, Dr. N.G.P. Institute of Technology, for providing us the facilities to do our project work.

We would like to express our hearty thanks and gratitude to Dr. S.U. PRABHA, M.E., Ph.D., Principal, Dr. N.G.P. Institute of Technology, for her earnest encouragement.

we extend our deep sense of gratitude to Dr. M. KRISHNAMOORTHY, M.E., Ph.D., Head of the Department, Department of Information Technology, for his valuable guidance and constructive suggestion at all stages of our project from inception to completion.

We express our hearty thanks to our project guide Ms.R.RATHIYA,, Assistant Professor (SG), Department of Information Technology, for his/her valuable guidance and timely help for completing our project.

We express our sincere thanks to our mini project coordinator Ms. BIJI ROSE, M.E., Assistant Professor (SG), Department of Information Technology, for her support in developing our project.

We would also like to express our gratitude to the faculty members of Department Information Technology and also to our family for their kind patronage.

ABSTRACT

The "Skin care Routine" project is a web-based solution designed to provide users with valuable insights into their skin condition and tailored skincare advice. The application leverages the Flask framework for web development and integrates OpenCV for image processing, enabling users to upload an image of their skin and select their skin type.

Upon image submission, the system employs a series of image processing techniques to analyze the image. It counts the number of pimples and identifies their areas of concern. Subsequently, the application generates personalized skincare recommendations based on both the pimple count and the user's selected skin type.

Key features of this project include a user-friendly interface, a dynamic recommendation system, and an easy-to-follow image upload process. Users can access valuable guidance on skincare routines, products, and regimens, making it a valuable tool for individuals seeking to enhance their skincare practices.

Future development plans encompass enhancing user engagement with social features, data tracking, and AI-driven image analysis. With an emphasis on privacy, security, and regulatory compliance, the project aims to provide a holistic and reliable solution for skin health assessment and skincare recommendations. As it evolves, it seeks to cater to a diverse audience, ensuring the availability of multilingual support and extensive documentation to aid users in their skincare journey.

KEYWORDS: Skin care Website, Web Application, image analysis, image upload process.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	iii
	LIST OF TABLES	
	LIST OF FIGURES	vii
	LIST OF ABBREVIATION	viii
1	INTRODUCTION 1	
	1.1 OBJECTIVE	1
2	LITERATURE SURVEY	2
	2.1 INTRODUCTION	2
	2.2 LITERATURE REVIEW	2
	2.2.1 To develop an information system Using the waterfall development Method	2
	2.2.2 Challenges faced by stakeholders	2
	2.2.3 To make an easy way to find the Required property	3
	2.2.4 To develop a typical web-based Online platform for both tenants and house owners	
	2.2.5 To manage tasks and issue easily in an exceedingly more convenient	4
	2.2.6 Solve all the issues experienced with the present manual system	5
3	SYSTEM ANALYSIS	
	3.1 EXISTINS SYSTEM	5

	3.1.1 Drawbacks of Existing System	5
	3.2 PROPOSED SYSTEM	5
	3.2.1 Block Diagram	6
4	SYSTEM SPECIFICATION	7
	4.1 HARDWARE SPECIFICATION	7
	4.2 SOFTWARE SPECIFICATIONS	7
	4.2.1. FRAMEWORK USED	7
5	SYSTEM OVERVIEW	
	5.1 MODULE DESCRIPTION	8
6	SYSTEM TESTING & IMPLEMENTATION	12
7	RESULT AND OUTPUT	14
	7.1 EXPERIMENTAL RESULT	14
	7.2 OUTPUT SCREEN	15
8	CONCLUSION AND FUTURE	
	ENHANCEMENT	18
	8.1 CONCLUSION	18
	8.2 FUTURE WORK	19
	APPENDIX	
	A READ ME FILE	20
	A. 1. ABOUT THE APPLICATION	20
	A.2 DEVELOPERS	20
	B. SOURCE CODE	21
	B.1 CODING	21

LIST OF ABBREVIATION

HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
PYTHON	high-level, interpreted, and general-purpose programming language
FLASK	popular web framework for Python

CHAPTER - 1

INTRODUCTION

In the realm where technology meets skincare, our project, "Skin Care Routine: Empowering Beauty through Artificial Intelligence," marks a pioneering stride into the future of personal care. Harnessing the prowess of Artificial Intelligence, we delve deep into the world of skincare, revolutionizing the way we understand and nurture our skin. In an era where precision and personalization are paramount, AI emerges as the guiding light. Our project explores innovative algorithms and data-driven insights that decode the intricacies of individual skin types and their unique needs. From tailored product recommendations to predictive analysis of skin health, we unlock the potential of AI to transform skincare routines into intelligent, adaptive regimens. Join us on this transformative journey as we unveil the synergy between human beauty aspirations and the cutting-edge capabilities of Artificial Intelligence, redefining the very essence of skincare. Welcome to a future where science, technology, and self-care converge seamlessly – "Skin Care Routine: Empowering Beauty through Artificial Intelligence."

1.1 OBJECTIVE

The objective for this project is to create a web application that offers skin analysis and skincare recommendations based on user-uploaded images and their specified skin types. The project aims to provide a user-friendly and informative platform for users to assess their skin condition and receive personalized advice for skincare.

Key project objectives include:

1. **Skin Analysis:** Enable users to upload images of their skin, which will be processed to count pimples and assess skin conditions.
2. **Personalized Recommendations:** Generate skincare recommendations tailored to the user's specific skin condition, taking into account their pimple count and skin type (oily or dry).
3. **User-Friendly Interface:** Create an intuitive and user-friendly web interface that allows for easy image uploads and selection of skin type.

4. **Informative Feedback:** Provide clear and informative feedback to users, including their pimple count and actionable skincare recommendations.
5. **Web Accessibility:** Ensure that the web application is accessible to a wide range of users and devices, promoting inclusivity and usability.
6. **Scalability:** Design the application with the potential for future growth, such as accommodating a larger user base or additional features.
7. **Quality and Accuracy:** Strive for accuracy in pimple detection and the generation of meaningful skincare advice.
8. **User Engagement:** Encourage user engagement by creating a platform where users can track their skin condition over time and share their results with others.
9. **Privacy and Security:** Prioritize the privacy and security of user data, ensuring that sensitive information is handled responsibly.
10. **Potential for Monetization:** Explore monetization strategies if the project is intended to generate revenue, such as offering premium features or partnering with skincare product providers.
11. **Feedback and Improvement:** Gather user feedback and continuously improve the application to enhance the user experience and the quality of skincare recommendations.

The project's overarching goal is to provide users with valuable insights into their skin health and empower them to make informed decisions about their skincare routines. It should offer a valuable resource for individuals seeking to improve their skin's condition and maintain optimal skin health.

CHAPTER – 2

LITERATURE SURVEY

2.1. INTRODUCTION

"Skin Care Routine Optimization through AI: A Comprehensive Exploration," embarks on a profound journey akin to the pioneering strides made in cancer cell detection. Just as early diagnosis transforms cancer outcomes, our study mirrors this paradigm shift by integrating cutting-edge Artificial Intelligence models into the skincare landscape. This literature survey navigates the historical precedents and contemporary advancements in AI-driven skincare algorithms. From established models to emerging technologies, we scrutinize the intricacies of existing frameworks, evaluating their efficacy in revolutionizing personal skincare routines. By exploring the nuances of these models, we aim to unveil the transformative potential of AI, emphasizing its pivotal role in redefining the future of skincare. This survey not only illuminates the path taken by predecessors but also charts the course for future innovations, highlighting the enduring importance of AI in enhancing skincare practices and fostering a new era in self-care.

2.2 LITERATURE REVIEW

2.2.1 Promoting and Maintaining Skin Integrity in End-of-Life Care: A Systematic Review

A structured, individualised skin care regimen, consisting of gentle skin cleansing and moisturising is recommended (Lichterfeld et al., 2015;LeBlanc et al., 2018a;LeBlanc et al., 2018b;Wounds UK, 2018;,. Traditional washing with water and alkaline soap should be avoided as it compromises skin barrier integrity and increases skin pH (Lichterfeld et al., 2015;Moncrieff et al., 2015;Wounds UK, 2018). The use of no-rinse cleansers or soap-free liquid wash products, reflecting the pH-range of the acid mantle of healthy skin (pH 4.5-6.5)

2.2. 2 Skin tears: Epidemiology, Classification and Measurement

The importance of skin integrity is generally accepted for the fields of skin care because it is a quality indicator of patient care (Lichterfeld et al., 2015), and maintaining skin integrity with daily routine skin care, such as a bed bath, indicates higher cost-effectiveness than wound treatment (Flanagan et al., 2014)

2.2.3 Role of nurses in promoting the skin health of older people in the community

Maintaining and improving skin health is internationally recognised as one of the salient quality indicators across the entire continuum of health care settings. Evidence-informed practices to promote skin integrity and prevent breakdown are strongly recommended to safeguard patient safety, minimise risks, benchmark performance, and deliver care that is more cost-effective compared to wound treatment [1][2][3]. With an ageing population, age-related skin conditions are increasingly prevalent, demanding a shift in knowledge, attitude, and practice to address skin health.

2.2.4 Dry skin in home care: A representative prevalence study

To measure the prevalence and severity of dry skin in the home care nursing setting in Germany and to determine demographic and health characteristics associated with skin dryness. Background Advanced age and skin care dependency are risk factors for the development of skin dryness. Dry skin has a negative impact on the quality of life and increases the risk for secondary cutaneous infections and other adverse skin conditions. The prevalence of dry skin in home care is unknown. Design A representative multicenter prevalence study was conducted in home care services in Germany during July 2015.

2.2.5 Skin Care Routine Enhancement Through AI: A Review of Hybridized Neural Networks in Deep Learning Frameworks

In recent research, our exploration focuses on augmenting skincare routines using hybridized neural networks within deep learning frameworks. This study delves into the integration of advanced algorithms, aiming to elevate the accuracy and personalization of skincare regimens. By harnessing the power of hybridized neural networks, our project seeks to redefine skincare routines, offering promising advancements in personalized self-care strategies. Through this research, we aim to contribute to the evolution of skincare practices, paving the way for more precise and tailored skincare solutions.

.

CHAPTER-3

SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

DermExpert is an AI-powered skincare platform developed by a team of dermatologists and AI experts. This web app utilizes sophisticated image recognition algorithms and machine learning to assess users' skin conditions. Users can upload pictures of their skin concerns, and the system provides instant analysis, identifying issues like acne, eczema, or allergies. DermExpert offers personalized skincare routines, product recommendations, and expert advice based on the analysis. Its user-friendly interface and accurate assessments have made it a popular choice among individuals seeking personalized skincare solutions.

3.1.1. DRAWBACKS OF EXISTING SYSTEM

- **Beauty Product Recommendation Websites:** Various websites suggest beauty products based on user preferences and reviews. While helpful, these platforms lack in-depth analysis of individual skin types and conditions. Users might receive recommendations based on general trends rather than personalized assessments, limiting the efficacy of product suggestions.
- **Virtual Skin Analysis Tools:** Some virtual skin analysis tools exist, allowing users to upload images for basic skin assessments. However, these tools often lack the depth of analysis required for comprehensive skincare routines. They might provide surface-level suggestions but lack the integration of advanced algorithms for precise and tailored recommendations.

3.2. PROPOSED SYSTEM

To enhance the project and propose a more robust system, you can consider the following features and components for a comprehensive skincare analysis and recommendation system:

1. User Registration and Profiles:

- Implement user registration and user profiles to store user preferences, history, and past analysis results.

2. User Authentication and Security:

- Add user authentication and authorization to ensure that users' data is secure and to manage access to features.

3. User Dashboard:

- Create a user dashboard where users can upload images, view their analysis history, and receive personalized recommendations.

4. Image Upload and Processing:

- Improve the image processing pipeline to enhance the accuracy of pimple detection. Explore advanced computer vision techniques and machine learning models for more precise skin analysis.

5. Advanced Skincare Recommendations:

- Enhance the recommendation engine by taking into account additional factors, such as age, gender, and specific skincare concerns (e.g., acne, wrinkles, or dark spots).

6. Data Visualization:

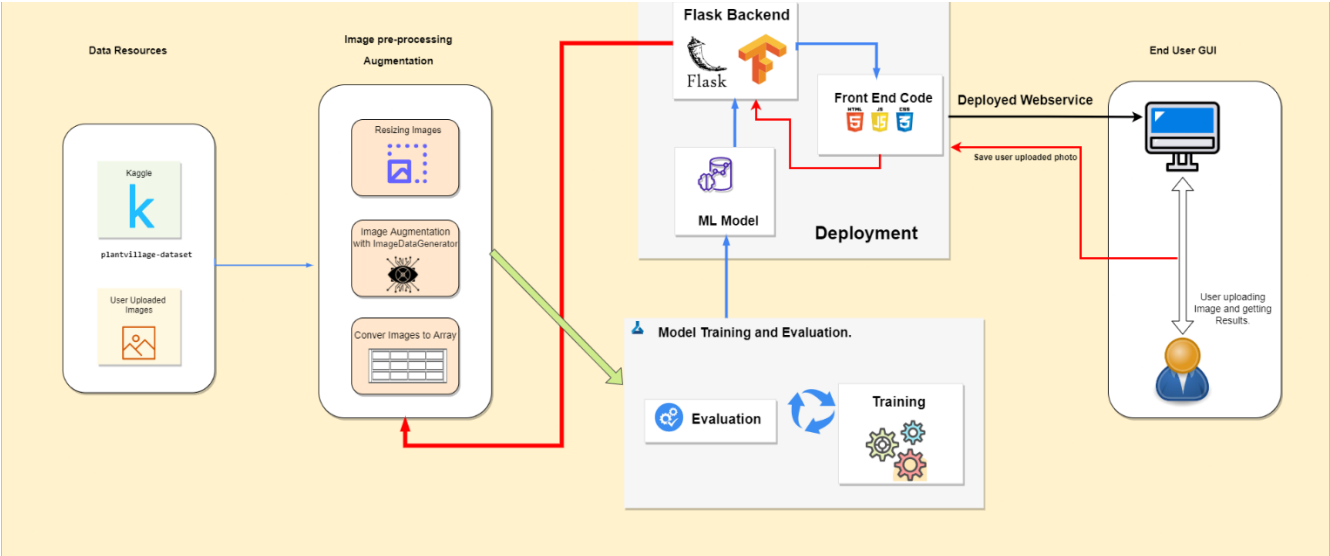
- Implement data visualization tools to help users track changes in their skin condition over time, providing visual representations of progress.

7. Community and Social Features:

- Incorporate social features that allow users to share their results, interact with others, and discuss skincare tips and product recommendations.

3.2.1. BLOCK DIAGRAM

Fig.I. BLOCK DIAGRAM



CHAPTER-4

SYSTEM SPECIFICATION

4.1. HARDWARE SPECIFICATION

This section gives the details and specification of the hardware on which the system is expected to work.

RAM	. 16GB
CPU	.. GHz
Architecture	.. 6 bit
Minimum Core	i5 processor
Program Storage Maximum	5 Omb

4.2. SOFTWARE SPECIFICATIONS

This section gives the details of the software that are used for the development.

- Language: Python
- Python version : 3.10.1
- Markup Language : HTML

4.2.1. FRAMEWORK USED

- Flask
- OpenCV (cv2)
- NumPy
- Jinja2 (render_template)

CHAPTER-5

SYSTEM OVERVIEW

5.1. MODULE DESCRIPTION

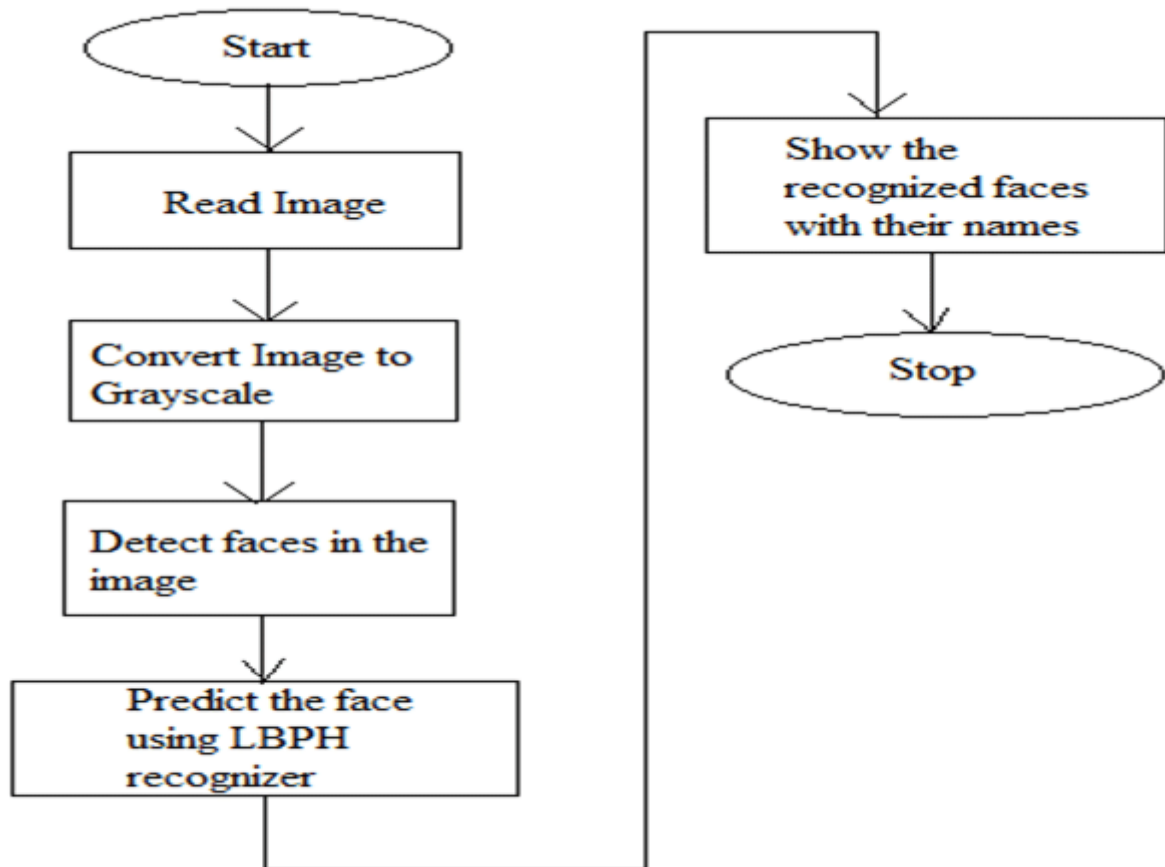
There is a total of two major parts of the project plan, which are frontend and backend parts. In this scientific research project HTML, CSS are used for the frontend design of the system.

HTML:HTML stands for Hypertext Markup Language which allows the user to make and structure sections, paragraphs, headings, titles, line breaks, add media, links, blockquotes, etc. for websites and applications.

CSS: Stands for Cascading Style Sheets which is a simple design language intended to simplify the method of constructing this website presentable. It's designed to enable the separation of presentation and content, including layout, colors, spacing, padding, fonts, and so on.

PYTHON: Python is a high-level, interpreted, and general-purpose programming language known for its simplicity and readability. Python has gained immense popularity due to its clean and easy-to-understand syntax, making it a great choice for both beginners and experienced programmers.

FLASK: Flask is a popular web framework for Python that allows you to build web applications quickly and with minimal code. It is known for its simplicity and flexibility, making it a great choice for developing web applications, RESTful APIs, and other web-related projects.



Data Base

Fig.7. System architecture

CHAPTER-6

SYSTEM TESTING AND IMPLEMENTATION

6.1. SYSTEM IMPLEMENTATION

For skin care of an ordinary person, implementation, and testing of the proposed system, a web application interface has been developed using HTML, CSS, that allows us to make web pages interactive and responsive. The code you've provided implements a basic web application using the Flask framework. This web application allows users to upload an image and select their skin type. The provided code serves as the foundation for a user-friendly web application built with Flask, a Python web framework. The primary objective of this application is to empower users to upload an image of their skin condition, along with specifying their skin type. Upon image submission, the application initiates an image processing pipeline using the OpenCV library. This pipeline begins by reading the uploaded image, converting it to grayscale, applying a Gaussian blur, and then thresholding to create a binary representation of the image. Contours in this binary image are then identified and evaluated to determine the number of skin blemishes.

The application goes a step further by offering personalized skincare advice. Depending on the pimple count and the user's skin type (either "oily" or "dry"), it generates tailored skincare recommendations. These recommendations guide users on the appropriate products and treatments for their specific skin condition.

With further development, this foundation can be expanded to include additional features, such as user authentication, a more sophisticated user interface, and the capability to store and track skincare advice over time.

The process consists of three steps

1. Flask Web Application Setup
2. Image Processing Functions
3. Flask Routes
4. Uploading and Saving the Image
5. Displaying Results
6. Debug Mode

6.1.1. Flask Web Application Setup

- The code initializes a Flask web application using `app = Flask(__name__)`. This sets up the web server and application context.

6.1.2. Image Processing Functions

- The code defines two functions for image processing and recommendation generation:
 - `process_image(image_path, skin_type)`: This function takes the path to an uploaded image file and the selected skin type as input. It uses the OpenCV library (`cv2`) to process the image and count pimples based on predefined criteria. The function returns the pimple count and a list of skincare recommendations.
 - `recommend_medicine(pimple_count, skin_type)`: This function takes the pimple count and skin type as input and generates skincare recommendations based on these parameters. The recommendations vary based on the pimple count and whether the skin type is "oily" or "dry."

6.1.3. Flask Routes

- The application defines two routes:
 - `'/'` (root route) is associated with the `index` function. This route renders an HTML template named `'index.html'`, which likely contains a form for uploading an image and selecting the skin type.
 - `'/upload'` is associated with the `upload` function. This route handles HTTP POST requests for uploading an image and processing it. It expects an image file in the request and the selected skin type. If a valid image is uploaded, it is processed using the `process_image` function, and the results are displayed on the `'result.html'` template.

6.1.4. Uploading and Saving the Image

- In the `upload` function, the code checks if a file has been uploaded and saves it to the server. The file path is used as input to the `process_image` function.

6.1.5. Displaying Results

- If a file is successfully uploaded and processed, the code renders the 'result.html' template, passing the pimple count and skincare recommendations as variables to be displayed on the web page.

6.1.6. Debug Mode

- When you run this script with `if __name__ == '__main__':`
`app.run(debug=True)`, it starts the Flask development server in debug mode. This mode is convenient during development, as it automatically reloads the server when you make code changes and provides detailed error messages in the browser

CHAPTER-7

RESULT AND OUTPUT


7.1. EXPERIMENTAL RESULT

The code you've provided is a Flask application for processing skin images and providing skincare recommendations. To see the output and results of this application, you need to run it and interact with it in a web browser. Make sure you have Flask and the required libraries (OpenCV, NumPy) installed in your Python environment. Save the code to a Python file, for example, `skincare_app.py`. Open a terminal or command prompt, navigate to the directory where `skincare_app.py` is located. Run the application using the following command: `python skincare_app.py`. This will start the Flask development server, and you should see output indicating that the server is running. Open a web browser and go to `http://localhost:5000/` or the URL specified in your terminal where the application is running. This will take you to the application's main page. On the main page, you should see a form that allows you to upload an image and select a skin type (e.g., "oily" or "dry"). Upload an image of a skin condition and select a skin type. Click the "Upload" or "Submit" button. The application will process the uploaded image, count pimples, and generate skincare recommendations based on the image analysis and the skin type you selected. You will be redirected to a result page where you can view the pimple count and the skincare recommendations. The recommendations will be displayed as output on the result page, and you can follow the skincare advice provided. You can repeat the process with different images and skin types to see how the application's analysis and recommendations change based on the input.

7.2 OUTPUT SCREEN

Skin Care

[Home](#)[About](#)[Skin Type And Disease](#)[Check My Skin Type](#)[Our Team](#)[Contact](#)



Your Skin Is Your Best Accessory

Protect Yourself From Skin Disease

Learn More


Skin Care

[Home](#)[About](#)[Skin Type And Disease](#)[Check My Skin Type](#)[Our Team](#)[Contact](#)

What Are Skin Diseases?

Skin Diseases Are Conditions That Affect Your Skin. These Diseases May Cause Rashes, Inflammation, Itchiness Or Other Skin Changes. Some Skin Conditions May Be Genetic, While Lifestyle Factors May Cause Others. Skin Disease Treatment May Include Medications, Creams Or Ointments, Or Lifestyle Changes.

Learn More



Skin Type And Skin Diseases

15

Skin Type And Skin Diseases

Cysts

Cysts Are Larger, Pus-Filled Pimples That Are Deep Beneath The Skin's Surface. They Can Be Painful, And If Not Treated, They May Lead To Scarring.

Nodules

Nodules Are Large, Painful, Solid Lumps Beneath The Surface Of The Skin

Pustules

Pustules Are Similar To Papules But Have A Visible White Or Yellowish Center Filled With Pus. They Are Often Referred To As "Zits" And Are Usually Red And Swollen.

Whiteheads

Whiteheads Are Small, Raised Bumps On The Skin With A White Or Yellowish Center. They Occur When Hair Follicles Become Clogged With Oil, Dead Skin Cells, And Bacteria

Blackheads (Open Comedones)

Blackheads Are Similar To Whiteheads But Have An Open Pore At The Surface

Papules

Papules Are Small, Red, Inflamed Bumps On The Skin Without A Visible Center. They Are Typically Tender To The Touch And Occur When The Walls Surrounding The Hair Follicles Break Down Due To Inflammation.

Whiteheads (Closed Comedones)

Whiteheads Are Small, Raised Bumps On The Skin With A White Or Yellowish Center. They Occur When Hair Follicles Become Clogged With Oil, Dead Skin Cells, And Bacteria

Check Your Skin Type

Choose A File

Select Your Skin Type: Normal ▼

Submit

Contact Us

Skin Care

We Detect The Number Of Pimple In Your Face
And Recommend The Type Of Vitamin Taken

Quick Links

[Home](#)
[About](#)
[Skin Type And Disease](#)
[Check My Skin Type](#)
[Our Team](#)
[Contact Us](#)

Connect With Us

[Facebook](#)
[Instagram](#)
[Linkedin](#)
[Twitter](#)

Image Analysis Result

Number of potential pimples: **11**

Recommended OTC products for pimples:

You may want to consult a dermatologist for more severe acne cases.

[Upload Another Image](#)

CHAPTER-8

CONCLUSION AND FUTURE WORK

8.1. CONCLUSION

The provided project demonstrates a web application for skin analysis and personalized skincare recommendations. It leverages Flask, OpenCV, and HTML templates to create a user-friendly platform for users to upload images of their skin and receive guidance tailored to their specific skin type and pimple count. **User-Centric Approach** This project prioritizes user experience by providing a simple and accessible interface for uploading images and receiving skincare recommendations.

The integration of OpenCV allows for basic image processing, enabling the automatic counting of skin blemishes. However, there is room for further refinement and sophistication in image analysis.

The application generates skincare recommendations based on the user's pimple count and skin type, acknowledging the importance of tailored advice for effective skincare.

To expand and enhance this project, consider incorporating user profiles, machine learning for more accurate analysis, and additional features like user interaction and mobile applications.

When working with user data, it's crucial to prioritize data security and comply with privacy regulations.

Solicit user feedback and adapt the application based on user experiences and evolving skincare best practices

8.2. FUTURE WORK

- Implement user authentication to allow users to create accounts and have personalized skincare profiles.
- Store user data and analysis results in a database associated with each user.
- Enhance the user interface with modern design principles, making it more appealing and intuitive.
- Provide clear instructions and feedback during the image upload and analysis process.

- Allow users to track their skincare progress over time, saving previous analysis results and recommendations.
- Implement data visualization to display trends and improvements.
- Create an admin panel to manage user accounts, moderate content, and monitor the system's performance and usage.
- Develop a mobile app version of the system for on-the-go image analysis and recommendations.
- Ensure that the app is compatible with various mobile devices and operating systems.

APPENDIX

A. READ ME FILE

A.1. ABOUT THE APPLICATION

The application is created using Flask, a lightweight web framework for Python. Flask is used to handle HTTP requests, route them to the appropriate functions, and render HTML templates. The application includes two main functions for image processing and skincare recommendations `process_image(image_path, skin_type)`. This function processes the uploaded image by converting it to grayscale, applying a Gaussian blur, thresholding the image, and identifying and counting skin blemishes (pimples) based on contour analysis. `recommend_medicine(pimple_count, skin_type)` This function generates skincare recommendations based on the pimple count and the selected skin type. The recommendations are customized for different scenarios, such as the severity of acne and skin type (oily or dry). `./upload` route: This route is associated with the upload function and is used to handle HTTP POST requests for image uploads and processing. If a valid image is uploaded, it is processed, and the results are presented to the user.

- The application allows users to upload an image through a web form. The uploaded image is saved to the server, and the file path is passed to the image processing function.
- After image processing is complete, the application renders an HTML template named 'result.html' to display the pimple count and skincare recommendations.
- The application runs in debug mode (app.run(debug=True)) during development. Debug mode automatically reloads the server when code changes are made and provides detailed error messages in the browser.

A.2. DEVELOPERS

ARINI GK	sriariniguru@gmail.com
MULLAIVENDHAN A	mullaivendhanarumugam@gmail.com
SREE VISHNU M	sreevishnumanian27@gmail.com

B. SOURCE CODE

B.1. CODING

INDEX.HTML

```
<!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">
  <title>Document</title>
  <!-- bootstrap cdn link -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">

  <!-- font awesome cdn link -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
```

```

<!-- custom css file link -->
<link rel="stylesheet" type="text/css" href="{ { url_for('static', filename='style.css') } } ">

</head>
<body>
    <!-- header section starts -->

<header class="fixed-top py-3">

    <div class="container d-flex align-items-center justify-content-between">

        <a href="#" class="logo">Skin Care<span class="fas fa-bacteria"></span></a>

        <div id="menu-bar" class="fas fa-bars d-inline-block d-md-none"></div>

        <nav class="nav">
            <a href="#home">home</a>
            <a href="#about">about</a>
            <a href="#prevent">skin type and disease</a>
            <a href="#check">check my skin type</a>
            <a href="#experts">Our Team</a>
            <a href="#contact">contact</a>
        </nav>

    </div>

</header>

<!-- header section end-->
<!-- home section starts -->

<section class="home" id="home">

    <div class="container">

        <div class="row min-vh-100 align-items-center pt-5">

            <div class="col-md-6">
                
            </div>

            <div class="col-md-6 content text-center text-md-start pl-md-5">
                <span>your skin is your best accessory </span>
                <h3>protect yourself from skin disease</h3>
            </div>
        </div>
    </div>

```

```

        <a href="#" class="link-btn">learn more</a>
    </div>

</div>

</div>

</section>

<!-- home section ends -->
<!-- home section starts -->
<section class="about" id="about">

    <div class="container">

        <div class="row align-items-center flex-wrap-reverse">

            <div class="col-md-6 content">
                <h3>What are Skin Diseases?</h3>
                <p>Skin diseases are conditions that affect your skin. These diseases may cause
rashes, inflammation, itchiness or other skin changes. Some skin conditions may be genetic,
while lifestyle factors may cause others. Skin disease treatment may include medications,
creams or ointments, or lifestyle changes.</p>

                <a href="#" class="link-btn">learn more</a>
            </div>

            <div class="col-md-6">
                
            </div>

        </div>

    </div>

</div>

</section>

<!-- about section ends -->
<!-- prevent section starts -->

<section class="prevent" id="prevent">

    <h1 class="heading"> <span>Skin Type and Skin Diseases</span> </h1>

```

```
<div class="container">
```

```
<div class="d-flex flex-wrap justify-content-center">
```

```
<div class="box p-4 m-2">
```

```

```

```
<h3>Cysts</h3>
```

```
<p>Cysts are larger, pus-filled pimples that are deep beneath the skin's surface. They can be painful, and if not treated, they may lead to scarring.</p>
```

```
</div>
```

```
<div class="box p-4 m-2">
```

```

```

```
<h3> Nodules</h3>
```

```
<p> Nodules are large, painful, solid lumps beneath the surface of the skin</p>
```

```
</div>
```

```
<div class="box p-4 m-2">
```

```

```

```
<h3>Pustules</h3>
```

```
<p>Pustules are similar to papules but have a visible white or yellowish center filled with pus. They are often referred to as "zits" and are usually red and swollen.
```

```
</p>
```

```
</div>
```

```
<div class="box p-4 m-2">
```

```

```

```
<h3>Whiteheads</h3>
```

```
<p>Whiteheads are small, raised bumps on the skin with a white or yellowish center. They occur when hair follicles become clogged with oil, dead skin cells, and bacteria</p>
```

```
</div>
```

```
<div class="box p-4 m-2">
```

```

```

```
<h3>Blackheads (Open Comedones)</h3>
```

```
<p>Blackheads are similar to whiteheads but have an open pore at the surface</p>
```

```
</div>
```

```
<div class="box p-4 m-2">
```

```

```

```
<h3>Papules</h3>
```

```
<p>Papules are small, red, inflamed bumps on the skin without a visible center. They are typically tender to the touch and occur when the walls surrounding the hair follicles break down due to inflammation.</p>
```

```
</div>
```

```
<div class="box p-4 m-2">
```

```

```

```
<h3>Whiteheads (Closed Comedones)</h3>
```

<p>Whiteheads are small, raised bumps on the skin with a white or yellowish center. They occur when hair follicles become clogged with oil, dead skin cells, and bacteria</p>

</div>

</div>

</div>

</section>

<!-- prevent section ends -->

<!-- check -->

<section class="check" id="check">

<h1 class="heading">

Check Your Skin Type

</h1>

<input type="file" id="real-file" hidden="hidden" />

<center><button type="button" id="custom-button">Choose A File</button></center>

<center>

<label for="skin-type">Select Your Skin Type:</label>

<select id="real_file" name="skin-type">

<option value="normal">Normal</option>

<option value="oily">Oily</option>

<option value="dry">Dry</option>

<option value="combination">Combination</option>

<option value="sensitive">Sensitive</option>

</select>

</center>

<center><button type="button" id="submit-button">Submit</button></center>

</section>

<div class="m">

<form method="POST" action="/upload" enctype="multipart/form-data" id="upload-form">

<input type="file" name="image" id="file-input" accept="image/*" required>

<label for="skin_type"></label>

<!-- <select name="skin_type" id="skin-type">

<option value="oily"></option>

<option value="dry"></option>

</select> -->

```

<br>
<input type="submit" value="Upload and Analyze" id="submit-input" style="display:
none;">
</form>
</div>

```

```

<script>
// Add an event listener to the custom button for file selection
document.getElementById("custom-button").addEventListener("click", function () {
    document.getElementById("file-input").click();
});

// Add an event listener to the real file input for tracking the selected file
document.getElementById("file-input").addEventListener("change", function () {
    var selectedFile = this.files[0];
    if (selectedFile) {
        document.getElementById("real-file").value = selectedFile.name;
    }
});

// Add an event listener to the submit button for form submission
document.getElementById("submit-button").addEventListener("click", function () {
    document.getElementById("submit-input").click();
});
</script>
<!-- check ends -->
<!-- experts section starts -->

```

```

<section class="experts" id="experts">

```

```

    <h1 class="heading"> meet our <span>Team</span> </h1>

```

```

    <div class="container">

```

```

        <div class="d-flex justify-content-center flex-wrap">

```

```

            <div class="box">

```

```

```

```

                <h3>Arini G K</h3>

```

```

                <span>3 rd year IT</span>

```

```

                <div class="share">

```

```

                    <a href="#" class="fab fa-instagram"></a>

```

```

                    <a href="#" class="fab fa-linkedin"></a>

```



```

        </div>
    </div>

    <div class="box">
        
        <h3>Mullaivendhan A</h3>
        <span>3 rd year IT</span>
        <div class="share">

            <a href="#" class="fab fa-instagram"></a>
            <a href="#" class="fab fa-linkedin"></a>
        </div>
    </div>

    <div class="box">
        
        <h3>Sree Vishnu M</h3>
        <span>3 rd year IT</span>
        <div class="share">
            <a href="#" class="fab fa-instagram"></a>
            <a href="#" class="fab fa-linkedin"></a>
        </div>
    </div>
</div>

</section>

<!-- experts section ends -->
<!-- contact section starts -->

<section class="contact" id="contact">

    <h1 class="heading"> <span>contact</span> Us </h1>

    <div class="container">

        <div class="row flex-wrap-reverse">

            <div class="col-md-7 p-2">
                <form action="">
                    <input type="text" placeholder="name" class="box">
                    <input type="email" placeholder="email" class="box">

```

```

        <input type="number" placeholder="number" class="box">
        <textarea      name=""      placeholder="message"      id=""      cols="30"
rows="10"></textarea>
        <input type="submit" class="link-btn" value="send message" name=""
id="">
    </form>
</div>

```

```

</div>

```

```

</div>

```

```

</section>

```

```

<!-- contact section ends -->

```

```

<!-- footer section starts -->

```

```

<section class="footer">

```

```

    <div class="container">

```

```

        <div class="d-flex flex-wrap justify-content-center text-center text-sm-start">

```

```

            <div class="box p-3 m-2">
                <h3>Skin Care</h3>
                <p>we detect the number of pimple in your face and recommend the type of
vitamin taken</p>
            </div>

```

```

            <div class="box p-3 m-2">
                <h3>quick links</h3>
                <a href="#">home</a>
                <a href="#">about</a>
                <a href="#">skin type and disease</a>
                <a href="#">check my skin type</a>
                <a href="#">our team</a>
                <a href="#">contact us</a>

```

```

            </div>

```

```

        <div class="box p-3 m-2">

```

```

    <h3>Connect with us</h3>
    <a href="#">facebook</a>
    <a href="#">instagram</a>
    <a href="#">linkedin</a>
    <a href="#">twitter</a>
  </div>

```

```

</div>

```

```

    <div class="credit"> created by <span>N.G.P ITECH</span> | all rights reserved
  </div>

```

```

</div>

```

```

</section>

```

```

<!-- footer section ends -->

```

```

<!-- js file link -->
<script      src="C:\Users\sreev\OneDrive\Desktop\mini\Skin-Disease-Detection-master
(1)\Skin-Disease-Detection-master\Front-end\main.js"></script>
</body>
</html>

```

RESULT.HTML

```

<!DOCTYPE html>
<html>
<head>
  <title>Image Analysis Result</title>

  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='styles.css')
  }}">
</head>
<body>
  <h1>Image Analysis Result</h1>
  <p>Number of potential pimples: {{ pimple_count }}</p>
  <p>Recommended OTC products for pimples:</p>
  <ul>
    {% for recommendation in recommendations %}
      <li>{{ recommendation }}</li>
    {% endfor %}
  </ul>

```

```

</ul>
<a href="/">Upload Another Image</a>
                                <script                                type="text/javascript"
src="C:\Users\sreev\OneDrive\Desktop\aaa\js\upload.js"></script>
</body>
</html>

```

APP.PY

```

from flask import Flask, request, render_template
import os
import cv2
import numpy as np

app = Flask(__name__)

def process_image(image_path, skin_type):
    image = cv2.imread(image_path)
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)
    _, thresh = cv2.threshold(blurred, 100, 255, cv2.THRESH_BINARY)
    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    min_pimple_area = 50
    pimple_count = 0
    for contour in contours:
        if cv2.contourArea(contour) > min_pimple_area:
            pimple_count += 1

    recommendations = recommend_medicine(pimple_count, skin_type)
    return pimple_count, recommendations

def recommend_medicine(pimple_count, skin_type):
    recommendations = []
    if pimple_count <= 5:
        recommendations.append("Consider using a gentle cleanser and a mild exfoliant.")
    else:
        recommendations.append("You may want to consult a dermatologist for more severe
acne cases.")

    if skin_type == "oily":
        recommendations.append("Look for products with salicylic acid or benzoyl peroxide.")
    elif skin_type == "dry":
        recommendations.append("Opt for non-comedogenic, hydrating products.")

    return recommendations

```

```

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/upload', methods=['POST'])
def upload():
    uploaded_file = request.files['image']
    skin_type = request.form.get('skin_type')
    if uploaded_file.filename != "":
        file_path = os.path.join(uploaded_file.filename)
        uploaded_file.save(file_path)
        pimple_count, recommendations = process_image(file_path, skin_type)
        return render_template('result.html', pimple_count=pimple_count,
                                recommendations=recommendations)
    return 'No file selected for upload.'

if __name__ == '__main__':
    app.run(debug=True)

```

STYLE.CSS(for index)

```

/* styles.css */
:root{
    --main-color:#926ad4;
    --black:#244361;
    --box-shadow:0 5px 10px rgba(0,0,0,.1);
}

*{
    margin:0; padding:0;
    box-sizing: border-box;
    outline: none; border:none;
    text-decoration: none !important;
    text-transform: capitalize;
    font-family: Verdana, Geneva, Tahoma, sans-serif;
}

html{
    scroll-padding-top: 60px;
}

section{
    padding:20px 0;
}

```

```

}

section:nth-child(odd){
  background:white;
}

.heading{
  text-align: center;
  padding:0 15px;
  padding-bottom: 5px;
  color:var(--black);
  font-size: 30px;
}

.heading span{
  color:var(--main-color);
}

.link-btn{
  display: inline-block;
  background:var(--main-color);
  color:#fff;
  border-radius: 50px;
  padding:10px 25px;
  font-size: 17px;
  cursor: pointer;
  box-shadow: var(--box-shadow);
}

.link-btn:hover{
  color:#fff;
  background:magenta;
}

header{
  box-shadow: var(--box-shadow);
  background: #fff;
}

header .logo{
  font-size: 20px;
  color:var(--black);
}

```

```

header .logo span{
  color:var(--main-color);
}

header .nav a{
  margin-left: 3px;
  border-radius: 50px;
  color:var(--black);
  padding:5px 15px;
}

header .nav a:hover{
  background: var(--main-color);
  box-shadow: var(--box-shadow);
  color:#fff;
}

#menu-bar{
  color:var(--black);
  cursor: pointer;
  font-size: 25px;
  transition: .2s linear;
}

.home{
  min-height: 100vh;
  background: url(/Images/home-bg.png) no-repeat;
  background-position: center;
  background-size: cover;
}

.home .content span{
  color:var(--main-color);
  font-size: 20px;
}

.home .content h3{
  color:var(--black);
  font-size: 50px;
  font-weight: bolder;
}

.about .content h3{
  color:var(--black);
  font-size: 33px;
}

```

```

.about .content p{
  color:#777;
  font-size: 14px;
}
.prevent .box{
  flex:1 1 300px;
  background:#f8f8fe;
  border-radius: 5px;
  text-align: center;
}

.prevent .box img{
  margin-bottom: 10px;
  height: 70px;
}

.prevent .box h3{
  color:var(--black);
  font-size: 20px;
}

.prevent .box p{
  color:var(--black);
  font-size: 14px;
}

#custom-button {
  padding: 10px;
  color: white;
  background-color: var(--main-color);
  border: 1px solid;
  border-radius: 5px;
  width: 100%;
  min-width: 50px;
  max-width: 300px;
  font-size:0.875em;
  display:block;
  left:-60px;
  margin-top:35px;
  cursor: pointer;
}

#custom-button:hover {
  background-color: magenta;
}

```



```

}

#custom-text {
  margin-left: 10px;
  font-family: sans-serif;
  color: black;
}
#submit-button {
  padding: 10px;
  color: white;
  background-color: var(--main-color);
  border: 1px solid;
  border-radius: 5px;
  width: 100%;
  min-width: 70px;
  max-width: 300px;
  font-size: 1em;
  display: block;
  left: -60px;
  margin-top: 35px;
  cursor: pointer;
}
#submit-button:hover {
  background-color: magenta;
}

.experts .box {
  width: 260px;
  background: #fff;
  border-radius: 5px;
  box-shadow: var(--box-shadow);
  margin: 5px;
  padding: 15px;
  text-align: center;
  position: relative;
  overflow: hidden;
}

.experts .box img {
  background: #f8f8fe;
  border-radius: 5px;
  width: 100%;
  margin-bottom: 8px;
}

```

```

.experts .box h3{
  font-size: 22px;
  margin:2px 0;
  color:var(--black);
}

.experts .box span{
  font-size: 15px;
  color:var(--main-color);
}

.experts .box .share{
  position: absolute;
  top:5px; right: -50px;
  transition: .2s;
}

.experts .box:hover .share{
  right:15px;
}

.experts .box .share a{
  border-radius: 5px;
  background:var(--main-color);
  color:#fff;
  display: block;
  height: 40px;
  width: 40px;
  line-height: 40px;
  margin-top: 5px;
}

.experts .box .share a:hover{
  background:var(--black);
}

.contact form{
  padding:15px;
  border-radius: 5px;
  box-shadow: var(--box-shadow);
}

.contact form .box,
.contact form textarea{
  background:#f8f8fe;
  border-radius: 5px;
  font-size: 17px;
}

```

```
padding:10px;
margin:7px 0;
width: 100%;
text-transform: none;
color:var(--black);
}
```

```
.contact form textarea{
  resize: none;
  height: 200px;
}
```

```
.contact .map{
  border-radius: 5px;
  height: 100%;
  width:100%;
}
```

```
.footer .box{
  flex:1 1 250px;
}
```

```
.footer .box h3{
  font-size: 20px;
  color:var(--black);
}
```

```
.footer .box p{
  color:#777;
}
```

```
.footer .box a{
  display: block;
  font-size: 14px;
  color:#777;
  padding:5px 0;
}
```

```
.footer .box a:hover{
  color:var(--main-color);
}
```

```
.footer .credit{
  text-align: center;
  color:var(--black);
  border-top: 1px solid rgba(0,0,0,.1);
}
```

```

    margin-top: 10px;
    padding-top: 20px;
}

.footer .credit span{
    color:var(--main-color);
}

.skin-type:hover {
    background-color: #f0f0f0; /* Change this to the desired color */
}

/* media queries */

@media (max-width:991px){

    .home .content h3{
        font-size: 30px;
    }

}

@media (max-width:768px){

    header .nav{
        position: absolute;
        top:100%; left: 0; right: 0;
        background: #fff;
        border-top: 1px solid rgba(0,0,0,.1);
        border-bottom: 1px solid rgba(0,0,0,.1);
        transition: .2s linear;
        clip-path: polygon(0 0, 100% 0, 100% 0, 0 0);
    }

    header .nav.active{
        clip-path: polygon(0 0, 100% 0, 100% 100%, 0% 100%);
    }

    header .nav a{
        display: block;
        width: 100%;
        margin:10px;
    }
}

```

```

    .fa-times {
        transform: rotate(180deg);
    }
}
/* Hide the "Choose A File" button */
.m {
    display: none;
}

```

STYLES.CSS(for result)

```

/* styles.css */
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-image: url('1-04244.jpg'); /* Replace 'background.jpg' with your
background image file */
    background-size: cover;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

.container {
    max-width: 600px;
    padding: 40px;
    background-color: rgba(255, 255, 255, 0.8); /* Semi-transparent white background
*/
    border-radius: 10px;
    box-shadow: 0 0 20px rgba(0, 0, 0, 0.2);
    text-align: center;
}

h1 {
    font-size: 36px;
    color: #333;
}

.result-container {
    margin-top: 30px;
}

```

```

.result-text {
    font-size: 18px;
    margin: 10px 0;
}

.result-number {
    font-weight: bold;
    color: #ff5733; /* Orange color for result number */
}

.recommendation-list {
    list-style: none;
    padding: 0;
}

.recommendation-list li {
    font-size: 16px;
    margin: 5px 0;
    color: #4caf50; /* Green color for recommendations */
}

.upload-link {
    display: block;
    margin-top: 30px;
    font-size: 18px;
    color: #007bff; /* Blue color for upload link */
    text-decoration: none;
}

```

REFERENCES

1. Wagner SK, Fu DJ, Faes L, Liu X, Huemer J, Khalid H, D, Korot E, Kelly C, Balaskas K, Denniston AK, Keane PA. Insights into Systemic Disease through Retinal Imaging-Based Oculomics. Transl Vis Sci Technol. 2020 Feb 12;9(2):6. doi: 10.1167/tvst.9.2.6. Erratum in: Transl Vis Sci Technol. 2021 Jul 1;10(8):13. PMID: 32704412; PMCID: PMC7343674.
2. Masih, N. , Naz, H. & Ahuja, S. Multilayer perceptron based deep neural network for early detection of coronary heart disease. Health Technol. I 1, 127—138 (2021). <https://doi.org/10.1007/s12553-020-00509-3>.
3. Jeena, R.S., Sukesh Kumar, A. , Mahadevan, K. (2019). A Novel Method for Stroke

Prediction from Retinal Images Using HOG Approach. In: Thampi, S. , Marques, O., Krishnan, S. , Li, KC., Ciuonzo, D., Kolekar, M. (eds) *Advances in Signal Processing and Intelligent Recognition Systems. SIRS 2018. Communications in Computer and Information Science*, vol 968. Springer, Singapore. https://doi.org/10.1007/978-981-13-5758-9_12.

4. Khan NC, Perera C, Dow ER, Chen KM, Mahajan VB, Mruthyunjaya P, Do DV, Leng T, Myung D. Predicting Systemic Health Features from Retinal Fundus Images Using Transfer-Learning-Based Artificial Intelligence Models. *Diagnostics (Basel)*. 2022 Jul 14;12(7):1714. doi: 10.3390/diagnostics12071714. PMID: 35885619; PMCID: PMC9322827.
5. Q. Wu and A. Cheddad, "Segmentation-based Deep Learning Fundus Image Analysis," 2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA), Istanbul, Turkey, 2019, pp. 1-5, doi: 10.1109/IPTA.2019.8936078.
6. S. , Deepa, et al. "Vascular Disease Prediction Using Retinal Image Acquisition Algorithm." *Real-Time Applications of Machine Learning in Cyber-Physical Systems*, edited by Balamurugan Easwaran, et al., IGI Global, 2022, pp. 185-199. <https://doi.org/10.4018/978-1-7998-9308-0.ch012>.
7. Sajeev, S. et al. (2019). Deep Learning to Improve Heart Disease Risk Prediction. In: et al. *Machine Learning and Medical Engineering for Cardiovascular Health and Intravascular Imaging and Computer Assisted Stenting. MLMECH CVII-STENT 2019* 2019. *Lecture Notes in Computer Science ()*, vol 11794. Springer, Cham. https://doi.org/10.1007/978-3-030-33327-0_12.
8. Larroza, A.; L6pez-Lereu, M.P.; Monmeneu, J.V.; Gavara, J.; Chorro, F.J.; Bodi, V.; Moratal, D, "Texture analysis of cardiac cine magnetic resonance imaging to detect nonviable segments in patients with chronic myocardial infarction", *Med. Phys.*, vol 45, pp.1471 1480, 2018.
9. Vidya, K.s.●, Ng, Acharya, U.R.●, Chou, s.M.●, San Tan, Ghista, D.N., "Computer-aided diagnosis of myocardial infarction using ultrasound images with DWT, GLCM and HOS methods: Acomparative study", *Comput. BioMed.*, vol 62, PP. 86-93. 2015.
10. Sudarshan, v.K.●, Acharya, U.R.●, Ng, San Tan, Chou, s.M.●, Ghista, D.N. "An integrated index for automated detection of infarcted myocardium from crosssectional echocardiograms using textonbased features (Part 1) ", *Comput.Biol. Me*, vol 71, pp.231-240,2016.

11. Madani, A.; Ong, J.R.; Tibrewal, A.; Mofrad, M.R., "Deep echocardiography: Dataefficient supervised and semi-supervised deep learning towards automated diagnosis of cardiac disease", NPJ Digit. Med., vol 1, pp. I—I 1, 2018.
12. xu, xu, Gao, Zhao, Zhang, Zhang, Du, Zhao, Ghista, D.; Liu, H, "Direct delineation of myocardial infarction without contrast agents using a joint motion feature learning architecture", Med. Image Anal. , vol 50, pp.82-94, 2018.
13. Gbantous, C.M.; Kamareddine, L.; Farhat, R.; Zouein, F.A.; Mondello, S.; Kobeissy, F.; Zeidan, A, "Advances in cardiovascular biomarker discovery", Biomedicines, vol 8, pp.552, 2020.
14. A Umamageswari, J Shiny Duella, K Raja, "Identifying Diabetics Retinopathy using Deep Learning based Classification" ,2021 22nd International Arab Conference on Information Technology (ACIT), 1-6, 2021.