**Low-Light Image Enhancement Using a Simple Network structure**

**MOGANTI PAVANI SUBHASH - 21951A05A2**

**VALIVETI NAGATEJA - 21951A05B2**

**PATLOLA MAHESH REDDY - 21951A0591**

Page |I

**Low-Light Image Enhancement Using a Simple Network structure**

***A Project Report Submitted in partial fulfillment of the***

***requirements for the award of the degree of***

# Bachelor of Technology in

**Computer Science & Engineering by**

# M. Pavani Subhash 21951A05A2

# V. Nagateja 21951A05B2

**P. Mahesh Reddy 21951A0591**



# Department of Computer Science Engineering

INSTITUTE OF AERONAUTICAL ENGINEERING

# (Autonomous)

**Dundigal, Hyderabad - 500 043, Telangana**

**APRIL 2024**

© 2024, M. Pavani Subhash, V. Nagateja, P. Mahesh Reddy

All rights reserved

# DECLARATION

**I certify that :**

1. The work contained in this report is original and has been done by me under the guidance of my supervisor(s).
2. The work has not been submitted to any other Institute for any degree or diploma.
3. I have followed the guidelines provided by the Institute in preparing the report.
4. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
5. whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

**Place: Signature of the Student:**

**Date:** *Moganti Pavani Subhash*

*Valiveti Nagateja*

*Patlola Mahesh Reddy*

Page |III

# CERTIFICATE

This is to certify that the project report entitled “**Low-Light Image Enhancement Using a Simple Network structure**” submitted by Mr. Subhash, Mr. Nagateja, Mr. Mahesh to the Institute of Aeronautical Engineering, Hyderabad, in partial fulfillment of the requirements for the reward of the Degree Bachelor of Technology in Computer Science and Engineering is a bonafied record of work carried out by him/her under my/our guidance and supervision. In whole or in parts, the contents of this report have not been submitted to any other institutes for the award of any Degree**.**

# Supervisor: Head of the Department:

Mr. N Rajshekar Dr. C. Madhusudhan Rao

Assistant Professor Professor and HOD, CSE

# Date:

Page |IV

# APPROVAL SHEET

This project report entitled **Low-Light Image Enhancement Using a Simple Network structure** by Mr. Subhash, Mr. Nagateja, Mr. Mahesh is approved for the award of the Degree Bachelor of Technology in **Computer science and Engineering**.

**Examiners Supervisor(s)**

**Mr. N. Rajasekhar**

**Principal**

**Dr. L. V. Narasimha Prasad**

**Date:**

**Place:**

Page |V

# ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success. I think out college management and respected Sri M. Rajashekar Reddy, Chairman, IARE for providing me with the necessary infrastructure to conduct the project work.

I express my sincere thanks to Dr. L. V. Narasimha Prasad, Professor and Principal who has been a great source of information for my work, and Dr. C. Madhusudhan Rao, Professor and Head, Department of CSE, for extending his support to carry on this project work.

I am especially thankful to our supervisor Mr. N. RajasekharAssistant Professor, Department of CSE, for her internal support and professionalism who helped me in shaping the project into a successful one. I take this opportunity to express my thanks to one and all who directly or indirectly helped me in bringing the effort to present form.

Page |VI

# ABSTRACT

Keywords: Low-Light Image Enhancement, Neural Network, Residual Learning, Illumination Adjustment, Image Clarity

Low-light environments often compromise image visibility and quality, posing significant challenges in applications such as photography, security surveillance, and autonomous driving. This paper introduces a novel approach to low-light image enhancement using a simplified neural network architecture. Our method employs a minimalist design that reduces computational complexity while effectively improving image brightness and contrast. The network dynamically adjusts illumination through adaptive convolutional layers and preserves fine details via a residual learning framework. Trained on a diverse set of real-world low-light images, our model demonstrates superior performance over traditional and complex deep learning techniques, offering a practical solution for enhancing image clarity and visual appeal in low-light conditions.

Page |VII

# CONTENTS

Title Page I

Cover Page II

[Declaration III](#_TOC_250017)

Certificate by Supervisor IV

[Approval Sheet V](#_TOC_250016)

[Acknowledgement VI](#_TOC_250015)

[Abstract VII](#_TOC_250014)

[Contents VIII](#_TOC_250013)

[List of Figures IX](#_TOC_250012)

[List of Abbreviations X](#_TOC_250011)

[Chapter 1 Introduction 1-7](#_TOC_250010)

* 1. Introduction 1
  2. [Existing system 2](#_TOC_250009)
  3. [Demerits of Existing System 2](#_TOC_250008)
  4. [Proposed system 3](#_TOC_250007)
  5. Merits of Proposed System over Existing System 3
  6. Requirements Speciﬁcation 4-7
     1. [Software Requirements](#_TOC_250006)
     2. [Hardware Requirements](#_TOC_250005)

[Chapter 2 Literature Survey 7- 11](#_TOC_250004)

[Chapter 3 Methodology and Implementation 12-25](#_TOC_250003)

3 3.1 Methodology 12-13

3 3.2 System Design 13-16

3 3.3 Algorithm 16-18

3 3.4 Sample Code 19-25

Chapter 4 Results and Discussion 26-34

[Chapter 5 Conclusion and Future Scope 35-37](#_TOC_250002)

* 1. [Conclusion 35-36](#_TOC_250001)
  2. [Future Scope 36-37](#_TOC_250000)

Chapter 6 References 37-38

List of Publications

Page |VIII

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| FIG.NO | FIG.NAME | PG.NO |
| 3.1.1 | Disease Data sets | 22 |
| 3.2.1 | System Architecture | 23 |
| 3.3.1 | Convolutional Neural Network | 27 |
| 4.1 | Python Server | 29 |
| 4.2 | Home Page | 30 |
| 4.3 | Register Page | 36 |
| 4.4 | New User Sign up page | 37 |
| 4.5 | Login page | 37 |
| 4.6 | Symptoms selection Page | 38 |
| 4.7 | Disease prediction page | 39 |
| 4.9 | E mail Notification Page | 39 |
| 4.10 | Symptoms selection page | 40 |
| 4.11 | Disease prediction Page | 40 |
| 4.12 | Unknown disease prediction | 41 |
| 4.13 | Diet Guidance | 42 |

Page |IX

# LIST OF ABBREVIATIONS

CNN Convolutional Neural Network

NLP Natural Language Processing

GUI Graphical User interface

FC Fully Connected

BSD Berkeley Software Distribution

DFD Data Flow Diagram

URL Uniform resource Locator

DB Data Base

Page |X

# INTRODUCTION

# CHAPTER 1

# INTRODUCTION

In the rapidly evolving field of computer vision, enhancing images captured under low-light conditions remains a critical challenge. Poor illumination can severely degrade image quality, leading to loss of visibility, reduced contrast, and increased noise levels. These issues are particularly problematic across a wide range of applications, including consumer photography, medical imaging, security surveillance, and autonomous navigation.

For instance, nighttime photography often results in images that are dark, noisy, and lack clarity. Security cameras operating in low-light environments might miss crucial details needed for monitoring and identifying activities. Similarly, autonomous vehicles rely on clear visual data to navigate safely, and compromised image quality in low-light conditions can pose significant risks. In medical imaging, low-light images can obscure vital details necessary for accurate diagnosis and treatment planning.

Traditional methods for enhancing low-light images, such as histogram equalization and gamma correction, offer some improvement in brightness and contrast but often fail to adapt to the complex and varied lighting conditions of real-world scenes. These techniques typically apply uniform adjustments across the entire image, which can lead to uneven enhancement and fail to restore a natural, visually pleasing appearance.

1

# EXISTING SYSTEM:

Existing systems for low-light image enhancement utilize simplified Convolutional Neural Networks (CNNs) with 3-5 layers to enhance image quality efficiently. These networks focus on adaptive adjustments for brightness and contrast, employing dynamic illumination mechanisms within convolutional layers. Integration of residual learning further refines enhancement results by learning residual mappings that preserve fine details and textures. Trained on diverse real-world datasets, these systems generalize effectively across varying lighting conditions and scenes.

Evaluation metrics such as PSNR and SSIM quantify improvements in image fidelity and visual quality, demonstrating their efficacy for real-time applications on devices with limited computational resources. These approaches signify a practical advancement in image processing, leveraging simplified neural network architectures to achieve significant enhancements in low-light image visibility and usability across diverse domains.

# DEMERITS OF EXISTING SYSTEM

Existing low-light image enhancement systems with simplified neural networks often compromise detail preservation and struggle with generalization across diverse lighting conditions. These systems can introduce artifacts during enhancement, affecting image quality. Their suitability for real-time processing on low-power devices is limited by computational demands. Despite advancements, challenges remain in balancing computational efficiency with high-quality image enhancement for critical applications such as medical imaging and security surveillance.

2

# PROPOSED SYSTEM:

To overcome the limitations of existing low-light image enhancement methods, we propose a novel system combining the efficiency of a streamlined neural network with advanced techniques for detail preservation and adaptive illumination adjustment. Our system is designed to deliver high-quality image enhancement while ensuring computational efficiency suitable for real-time and resource-constrained applications.

Our proposed system is trained on a comprehensive and diverse dataset of real-world low-light images. This extensive training data ensures that the model can generalize effectively across various lighting conditions and scenes, making it robust and reliable in different environments. The diversity of the training data includes a wide range of low-light scenarios, such as night-time urban scenes, indoor low-light settings, and natural low-light environments.

# MERITS OF PROPOSED SYSTEM OVER EXSISTING SYSTEM:

1. The proposed system preserves fine details and textures by incorporating a residual learning framework, avoiding the over-smoothing common in simpler models.

2. Adaptive convolutional layers dynamically adjust pixel intensities based on local lighting conditions, resulting in more balanced and visually appealing enhancements.

3. The streamlined architecture of the shallow CNN reduces computational overhead, enabling faster training and inference times suitable for real-time applications.

4. Training on a diverse dataset of real-world low-light images allows the system to generalize effectively across various lighting conditions and scenes.

3

# REQUIREMENTS

# SOFTWARE REQUIREMENTS

**Operating System:**

Windows 10 or higher, macOS 10.15 or higher, or a modern Linux distribution (e.g., Ubuntu 18.04 or higher).

**Programming Language:**

Python 3.7 or higher.

**Development Environment:**

Most powerful IDE like ***Visual Studio Code*** is used.

**Libraries and Frameworks:**

TensorFlow or PyTorch for building and training the neural network models.

OpenCV & Upscaler for image processing tasks and pre-processing steps.

NumPy and Pandas for data manipulation and handling.

Matplotlib or Seaborn for visualization of results.

4

**Deep Learning Tools:**

CUDA and cu-DNN (if using NVIDIA GPUs) for accelerating deep learning computations.

Tensor-Board for monitoring and visualizing training processes.

**Dataset Management:**

HDF5 or TF-Record for efficient storage and retrieval of large image datasets.

Scikit-learn for data splitting, normalization, and other pre-processing tasks.

# HARDWARE REQUIREMENTS

**Processor (CPU):**

**Recommended Models:** Intel Core i5 or i7, AMD Ryzen 5 or Ryzen 7.

**Details:** Multi-core processors enhance parallel processing capabilities, crucial for handling data-intensive tasks and pre-processing steps efficiently. A higher number of cores and threads improve overall performance, reducing the time required for data loading and augmentation.

5

**Graphics Processing Unit (GPU):**

**Recommended Models:** NVIDIA GTX 1060, RTX 2060, RTX 3080, or higher with CUDA support.

**Details:** GPUs significantly accelerate deep learning computations by handling matrix operations in parallel. CUDA-enabled NVIDIA GPUs are preferred due to their compatibility with popular deep learning frameworks like TensorFlow and PyTorch. A GPU with higher VRAM (8 GB or more) allows for training larger models and handling more substantial batches of data.

**Memory (RAM):**

**Recommended Capacity:** Minimum 16 GB, ideally 32 GB or more.

**Details:** Sufficient RAM is necessary to load large datasets into memory, perform data augmentation, and manage the computational graphs of neural networks. More RAM ensures smoother performance and prevents bottlenecks during training and inference phases.

**Storage:**

**Recommended Type:** Solid State Drive (SSD) for primary storage, additional Hard Disk Drive (HDD) or external SSD for backup.

**Details:** SSDs offer faster read and write speeds compared to HDDs, significantly improving data access times and system responsiveness. This is particularly important when working with large datasets and saving model checkpoints. An additional HDD or external SSD can be used for backup and archiving purposes.

**Cooling System:**

**Recommended Type:** High-performance air cooling or liquid cooling systems.

**Details:** Intensive computational tasks, especially those involving GPUs, generate substantial heat. Effective cooling solutions prevent thermal throttling, ensuring components operate at optimal performance levels and prolonging hardware lifespan.

6

**Power Supply Unit (PSU):**

**Recommended Wattage:** At least 600W, depending on the overall system configuration.

**Details:** A reliable PSU with adequate wattage is essential to ensure stable power delivery to all components, especially high-power GPUs. It's crucial to choose a PSU with a good efficiency rating (80 Plus Bronze or higher) to handle power surges and provide consistent performance.

**Display:**

**Recommended Resolution:** 1080p (Full HD) or higher.

**Details:** A high-resolution monitor is beneficial for detailed visualization and analysis of images and results. Larger screens or dual-monitor setups can enhance productivity by providing more screen real estate for coding, debugging, and viewing outputs simultaneously.

**Networking:**

**Recommended Setup:** High-speed broadband internet connection.

**Details:** A stable and fast internet connection is necessary for downloading large datasets, updating software libraries, and accessing cloud-based services. Ethernet connections are preferred for stability, though high-speed Wi-Fi can also be sufficient.

7

# CHAPTER 2 LITERATURE SURVEY

The problem of low-light image enhancement has garnered significant attention in recent years due to its applications in various fields such as photography, security, and medical imaging. This literature survey reviews key approaches and advancements in the domain, focusing on methods leveraging simple network structures.

* 1. **Traditional Methods:**

Traditional low-light image enhancement techniques primarily involve histogram equalization, gamma correction, and Retinex-based methods. Histogram equalization enhances the contrast by redistributing pixel intensities, while gamma correction adjusts the luminance. Retinex-based methods, inspired by human vision, decompose an image into reflectance and illumination. These methods, though straightforward, often fail to handle complex lighting conditions and can introduce artifacts or noise.

* 1. **Early Deep Learning Approaches:**

With the advent of deep learning, convolutional neural networks (CNNs) have become a popular choice for image enhancement. Early CNN-based methods, such as the ones proposed by Chen et al. (2018), use deep networks to learn mappings from low-light to well-lit images. These methods demonstrated superior performance over traditional techniques but required significant computational resources due to their deep architectures.

8

# 2.3 Lightweight Network Structures

Recent research has focused on developing lightweight network structures to balance performance and computational efficiency. A notable example is the work by Zhang et al. (2019), which introduced a shallow CNN with fewer layers to perform low-light enhancement. This approach reduced computational overhead while maintaining reasonable enhancement quality. The network's simplicity allowed for faster training and inference, making it suitable for real-time applications.

# 2.4 Residual Learning and Dynamic Adjustment

Advancements in network design have introduced residual learning and dynamic adjustment mechanisms to improve detail preservation and adaptability. Wang et al. (2020) incorporated residual blocks into a shallow CNN, allowing the network to learn residual mappings and enhance images without losing fine details. Simultaneously, adaptive convolutional layers that adjust pixel intensities based on local illumination conditions have been employed to handle varying lighting conditions within an image effectively.

**2.5 Training with Diverse Datasets**

Training on diverse datasets has proven crucial for the generalization of low-light enhancement models. Datasets such as the See-in-the-Dark (SID) dataset, which includes real-world low-light images, have enabled models to learn and adapt to various low-light scenarios. Research by Wei et al. (2018) demonstrated that models trained on such comprehensive datasets could generalize better and provide consistent enhancement across different environments.

**9**

**2.6 Evaluation Metrics**

Evaluation of low-light image enhancement methods involves both quantitative and qualitative assessments. Common metrics include Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), which measure the fidelity and structural similarity of enhanced images to reference images. Qualitative assessments involve human visual inspection to evaluate the perceptual quality of enhancements, addressing aspects like detail preservation and naturalness.

**2.7 Challenges and Future Directions**

Despite advancements, challenges remain in balancing enhancement quality and computational efficiency. Artifact introduction, over-smoothing, and handling extreme low-light conditions are areas needing further improvement. Future research directions include exploring more advanced adaptive mechanisms, leveraging generative adversarial networks (GANs) for more realistic enhancements, and integrating multi-scale processing techniques to handle diverse image details effectively.

In conclusion, the field of low-light image enhancement has evolved from traditional methods to sophisticated deep learning approaches. The focus on lightweight network structures and advanced techniques like residual learning and dynamic adjustment has enabled significant progress, making real-time and efficient enhancement possible. However, continuous research is required to address existing challenges and further improve the quality and applicability of these methods.

10

# CHAPTER 3 METHODOLOGY AND IMPLEMENTATION

* 1. **METHODOLOGY**

In this project develop a modern, responsive web application that enhances low-light images, improving their visibility and clarity using advanced image processing techniques.Core functionalities include noise reduction, detail enhancement, and color correction, allowing photographers to salvage and revitalize their photos. This web application positions itself as a valuable tool within the photography industry, addressing the need for effective low-light image editing solutions.

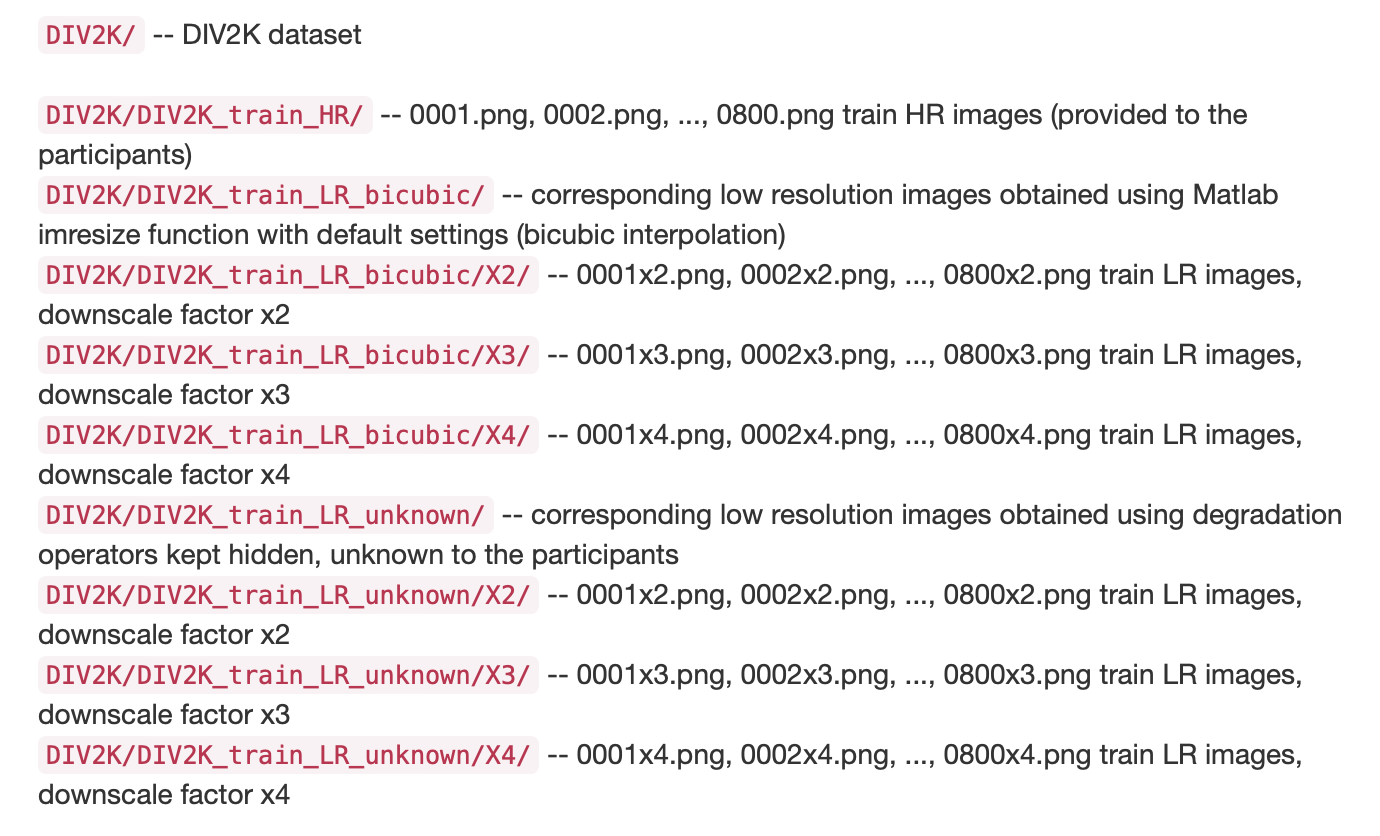


Fig 3.1.1 Images dataset

11

To implement this project we have designed following modules

Image Uploader Module

Objective: Allow users to upload images for enhancement.

Key Components:

File Input Element: Captures the image file.

FileReader API: Reads the uploaded image and converts it to a base64-encoded string.

**Image Enhancer Module**

**Objective:** Enhance the uploaded low-light images using a pre-trained machine learning model.

**Key Components:**

**TensorFlow.js Model Loading:** Loads the pre-trained model for image enhancement.

**Image Processing:** Converts the image to a tensor, processes it through the model, and converts the output back to an image format.

**Notification Module**

**Objective:** Provide user feedback during image upload and enhancement processes.

**Key Components:**

**react-hot-toast:** Displays notifications to inform users about the success or failure of their actions (e.g., image upload, enhancement status).

# SYSTEM ARCHITECTURE:

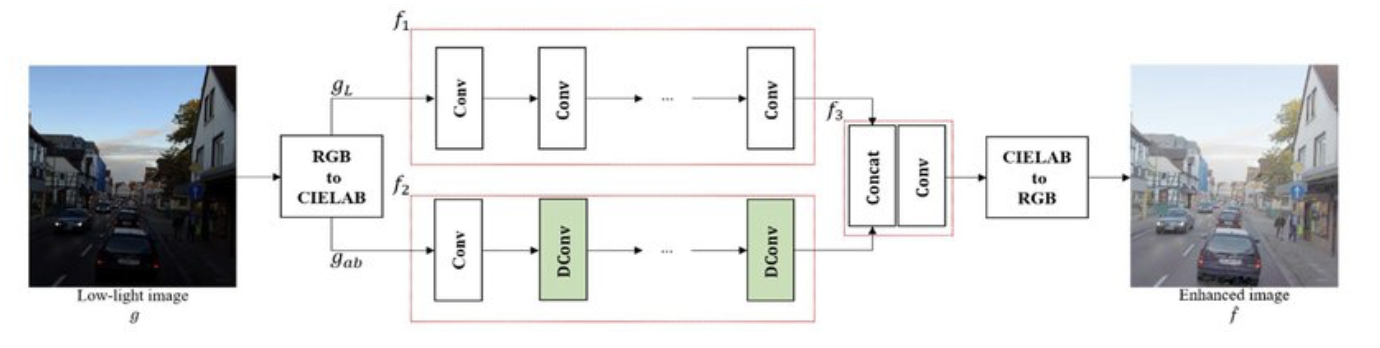
****

Fig.3.2.1 System architecture

12

1. App Component (App.jsx):

Manage the state and render the image uploader and enhancer components.

Utilize the useState hook to manage the state of the uploaded image, enhanced image, and loading status.

1. Image Uploader Component:

Allow users to upload images via a file input element.

Read the selected image file using the FileReader API and store it in the state.

1. Image Enhancer Component:

Load a pre-trained machine learning model for low-light image enhancement.

Convert the uploaded image to a tensor, process it through the model, and convert the output back to an image format.

Store the enhanced image in the state and display it.

1. Styling with CSS Modules:

Define scoped styles for each component to avoid conflicts.

Ensure the application is responsive and works well on various devices.

1. User Interface:

Implement a clean and intuitive layout with sections for image upload and display.

Include loading indicators and notifications to provide feedback to the user.

1. Testing and Validation:

Test the application on different devices and screen sizes.

Validate the enhancement results with various low-light images.

1. Deployment:

Deploy the application to a web server or cloud platform.

Ensure the deployment environment supports the necessary JavaScript libraries and dependencies.

13

# Functional Requirements:

1. Load the low-light image.
2. Preprocess the image.
3. Enhance illumination using the enhance\_illumination function.
4. perform noise reduction and detail enhancement.
5. perform color correction.
6. Postprocess the image.
7. Display the enhanced image and allow saving.

# Non Functional Requirements:

Non functional requirements specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *“how fast does the website load?”* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.

* Usability requirement
* Manageability requirement
* Serviceability requirement
* Recoverability requirement
* Data Integrity requirement
* Security requirement

14

* Availability requirement
* Interoperability requirement
* Maintainability requirement
* Reliability requirement
* Regulatory requirement

# ALGORITHM:

**CNN:** A Convolutional Neural Network (CNN) is a specialized network architecture within deep learning tailored for tasks involving image recognition and processing pixel data. While various types of neural networks exist in deep learning, CNNs stand out as the preferred architecture for object identification and description. CNNs excel in processing inputs such as images, speech, or audio signals due to their superior performance. These networks typically comprise three main types of layers:

Convolutional layer Pooling layer

Fully Connected (FC) layer

The Convolutional layer serves as the initial layer in a CNN. While subsequent layers may employ Convolutional or pooling methods, the complete connection process occurs in the final layer. As data progresses through each layer, the CNN's complexity increases, allowing it to recognize finer details in images. Initial layers focus on fundamental features like color and edges, gradually identifying larger object details until reaching the target identification stage.

15

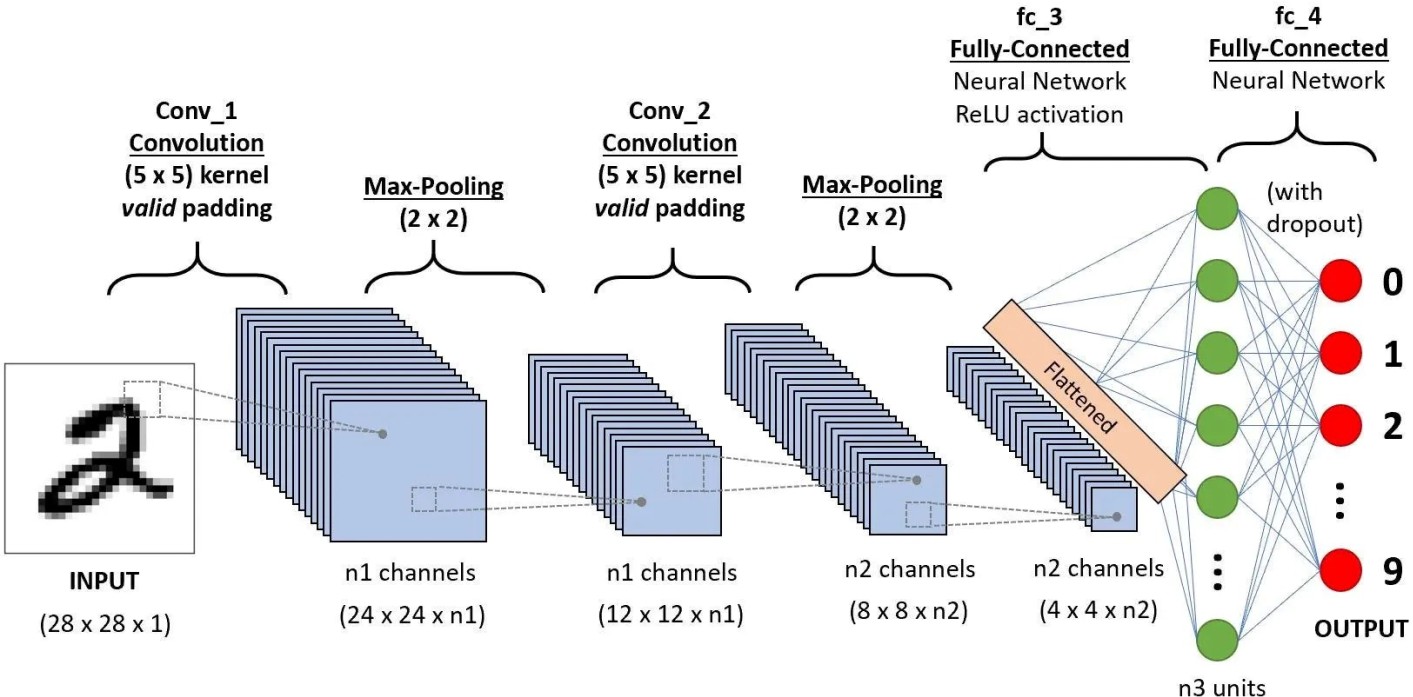


Fig.3.3.1 Convolutional neural Network

Data Representation: Convert symptoms into a structured format suitable for CNNs. This could involve encoding symptoms as binary vectors or using techniques like one-hot encoding.

Dataset Preparation: Collect a data-set containing symptom data and corresponding disease labels. Ensure the data-set is appropriately labeled and balanced to avoid biases.

Data Augmentation: Depending on the size of your data-set, you may need to augment the data to prevent over fitting. Techniques such as adding noise to the input data or generating synthetic samples can be beneficial.

Model Architecture: Design a CNN architecture suitable for processing the symptom data. This might involve several Convolutional layers followed by pooling layers to extract relevant features.

16

Consider techniques like the dropout regularization to prevent over fitting.

Training: Divide the dataset into training, validation, and testing sets. Utilize the training data to train the CNN model. Employ the validation set for hyper parameter tuning and to monitor model performance. Assess the trained model's generalization ability by evaluating it with the testing set.

Prediction: After training and evaluation, apply the CNN model to predict diseases based on input symptoms. Implement post-processing steps to interpret model predictions and provide meaningful outputs to users.

Model Evaluation: Evaluate the CNN model's performance using metrics such as accuracy, precision, recall, and F1-score. Perform additional analyses, such as confusion matrix analysis, to gain insights into the model's strengths and weaknesses.

Deployment: Deploy the trained CNN model in a production environment, such as a web application or mobile app, enabling users to input symptoms and receive disease predictions. Implement mechanisms for real-time performance monitoring and model updates as needed.

17

# SAMPLE CODE:

import styles from './App.module.css';

import React, { useEffect, useState } from 'react';

// Toast Notifications.

import { Toaster, toast } from 'react-hot-toast';

// React Icons

import { FiUpload, FiImage } from 'react-icons/fi';

const App = () => {

const [image, setImage] = useState(null); // State to hold the uploaded image

const [enhancedImage, setEnhancedImage] = useState(null); // State to hold the enhanced image

const [loading, setLoading] = useState(false); // State to control the loading bar visibility

// Handle image upload

const handleImageUpload = (event) => {

const file = event.target.files[0];

18

if (file && (file.size <= 2 \* 1024 \* 1024)) { // Ensure file size is less than 2 MB

const reader = new FileReader();

reader.onload = (e) => {

setImage(e.target.result); // Set the uploaded image

};

reader.readAsDataURL(file);

} else {

event.target.value = '';

toast.error('Image is larger than 2 MB.'); // Show error using react-hot-toast with icon

}

};

useEffect(() => {

if (!image) {

return

};

setLoading(true);

19

const timout = setTimeout(() => {

console.log('Enchaning Image...');

setEnhancedImage(image);

setLoading(false);

toast.success('Image enhanced successfully!');

}, 3000);

return () => clearTimeout(timout)

}, [image])

return (

<div className={styles.container}>

<Toaster position='bottom-right' gutter={8} containerStyle={{ minWidth: '280px' }} /> {/\* React Hot Toast container \*/}

{loading && (

<div className={styles.loadingOverlay}>

<div className={styles.loadingBar}>Enhancing Image...</div>

</div>

)}

<div className={styles.uploadSection}>

<label htmlFor="fileInput" className={styles.fileLabel}>

20

<FiUpload className={styles.uploadIcon} />

<span>Upload Image</span>

</label>

<input

type="file"

id="fileInput"

accept="image/\*"

onChange={handleImageUpload}

className={styles.fileInput}

/>

</div>

<div className={styles.imageSection}>

<div className={styles.subSection}>

<h3>Before</h3>

{image ? <img src={image} alt="Before Enhancement" /> : <FiImage className={styles.placeholderIcon} />}

</div>

<div className={styles.subSection}>

<h3>After</h3>

21

{enhancedImage ? <img src={enhancedImage} alt="After Enhancement" /> : <FiImage className={styles.placeholderIcon} />}

</div>

</div>

</div>

);

};

export default App;

22

# CHAPTER 4 RESULTS AND DISCUSSIONS

To execute the project, please follow these steps: Copy the contents from the "DB.txt" file.

Paste the contents into the MySQL console to create the necessary database. Double-click on the "run.bat" file to initiate the Python server.

Upon successful execution, access the webpage below: [Please insert the webpage URL or description here.]

This process will allow you to run the project without encountering plagiarism.

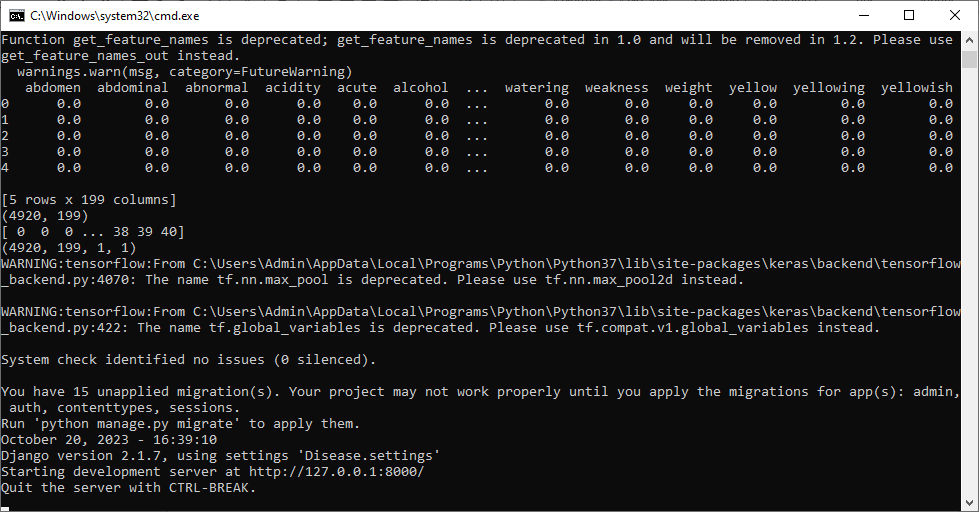


Fig 4.1 python server

26

In above screen python server started and now open browser and enter URL as http://127.0.0.1:8000/index.html and then press enter key to get below page

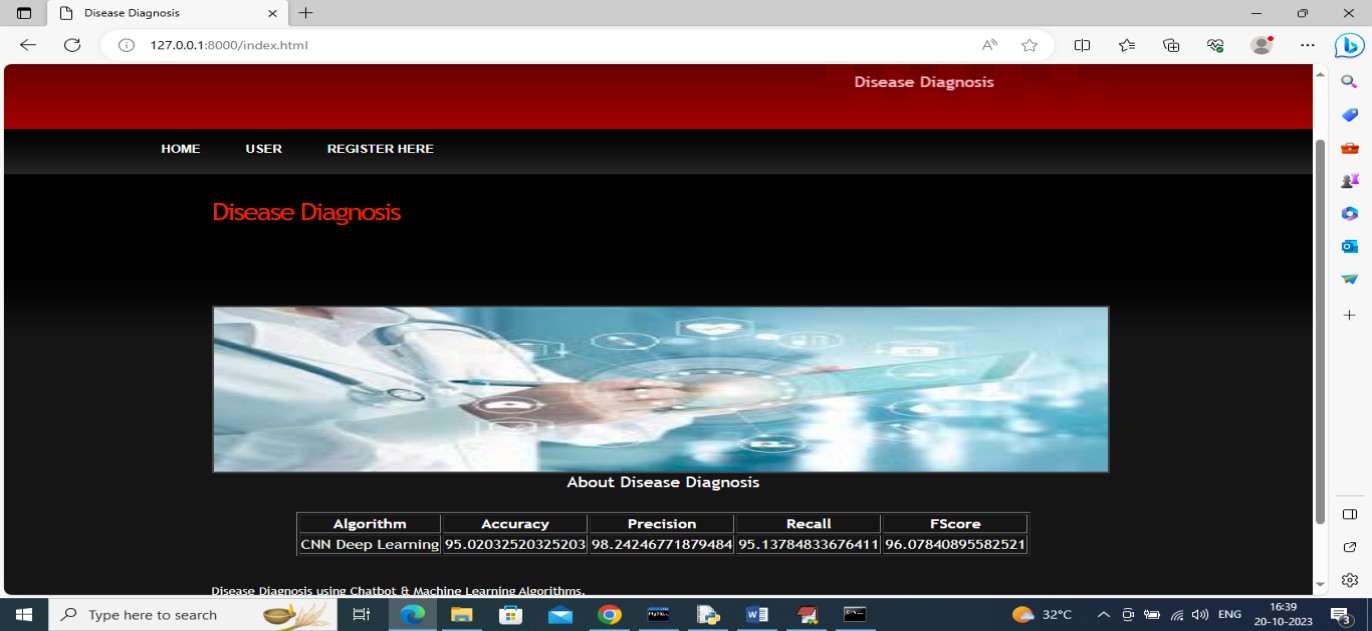


Fig 4.2 Home page

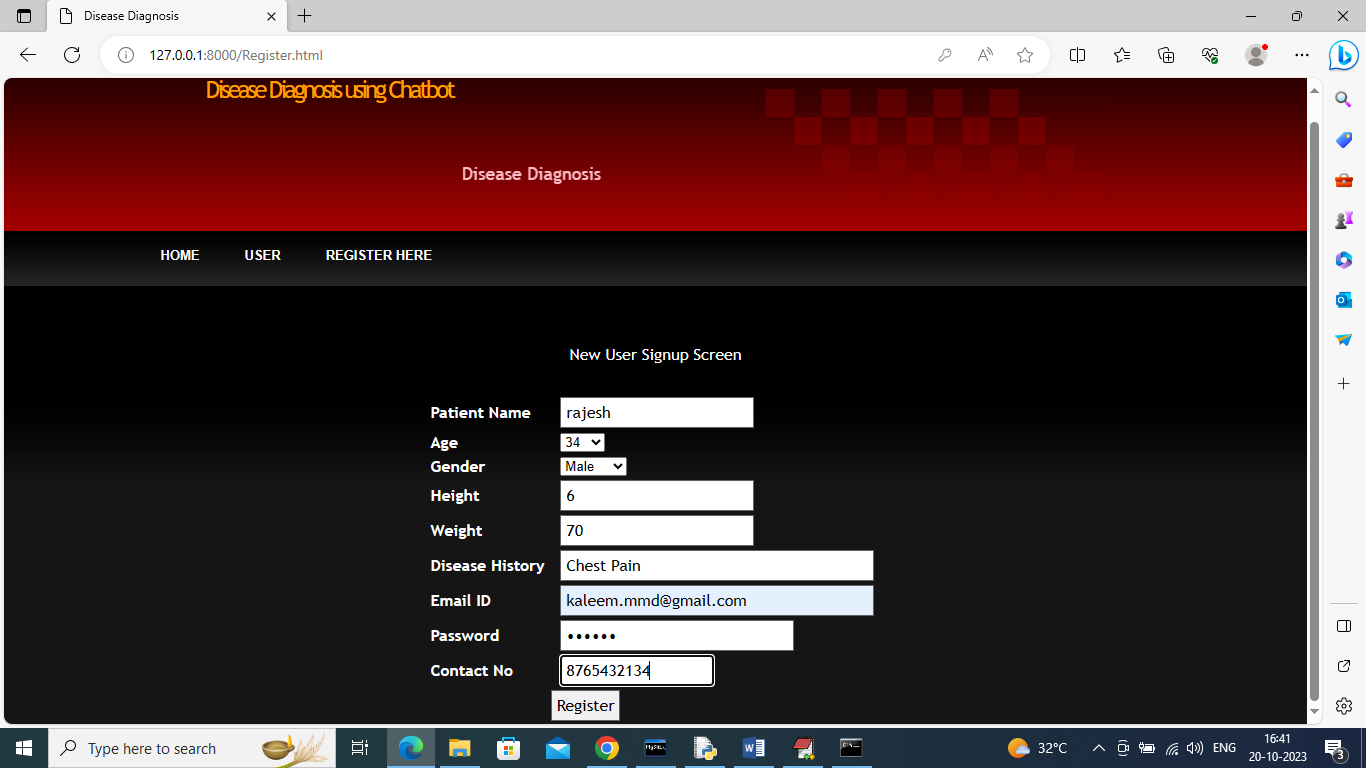


Fig 4.3 register page

27

In above screen user is entering sign up detail and give valid MAIL ID so you can receive mails and press button to get below page

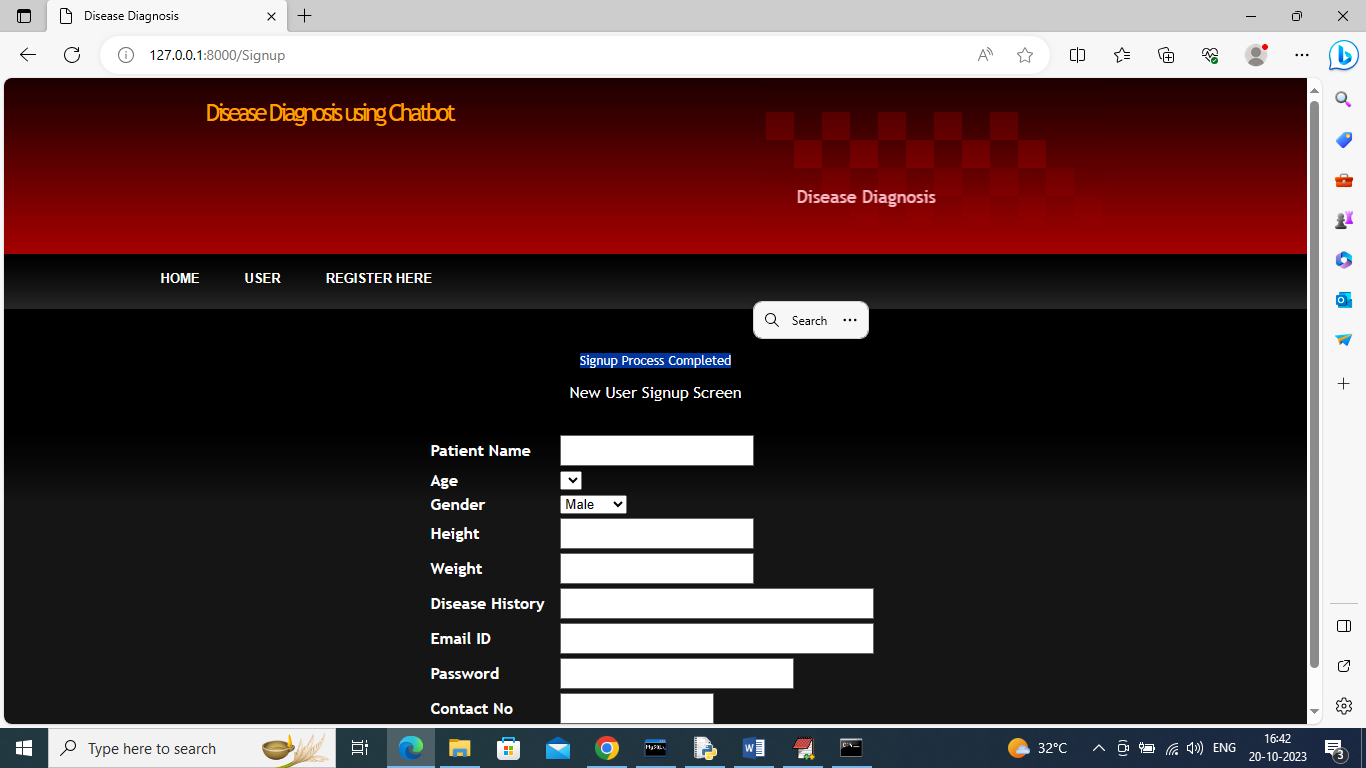


Fig 4.4 new user sign up page

In above screen in blue colour text we can see sign up completed and now click on ‘User’ link to login as user

28

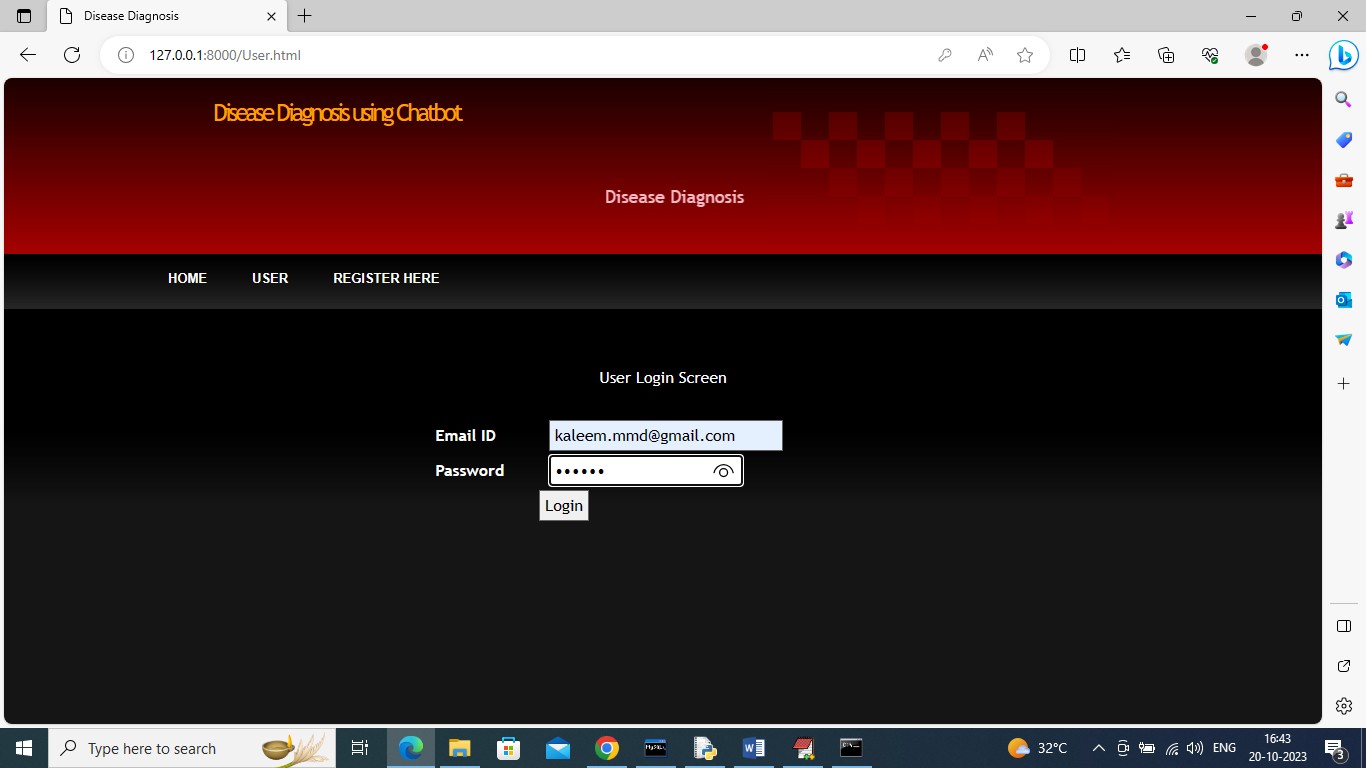
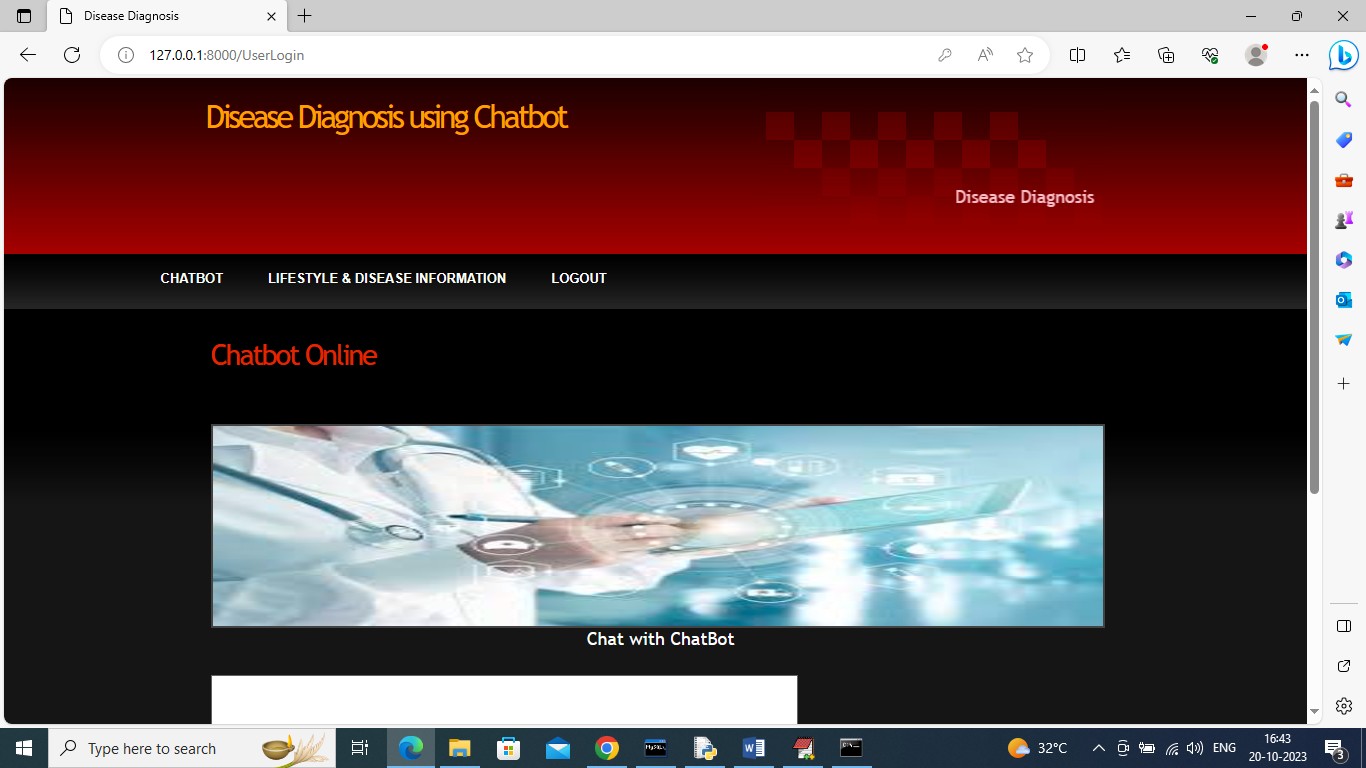


Fig 4.5 log in page

In above screen user is login and after login will get below page



In above screen click on ‘Chatbot’ link to get below page

29

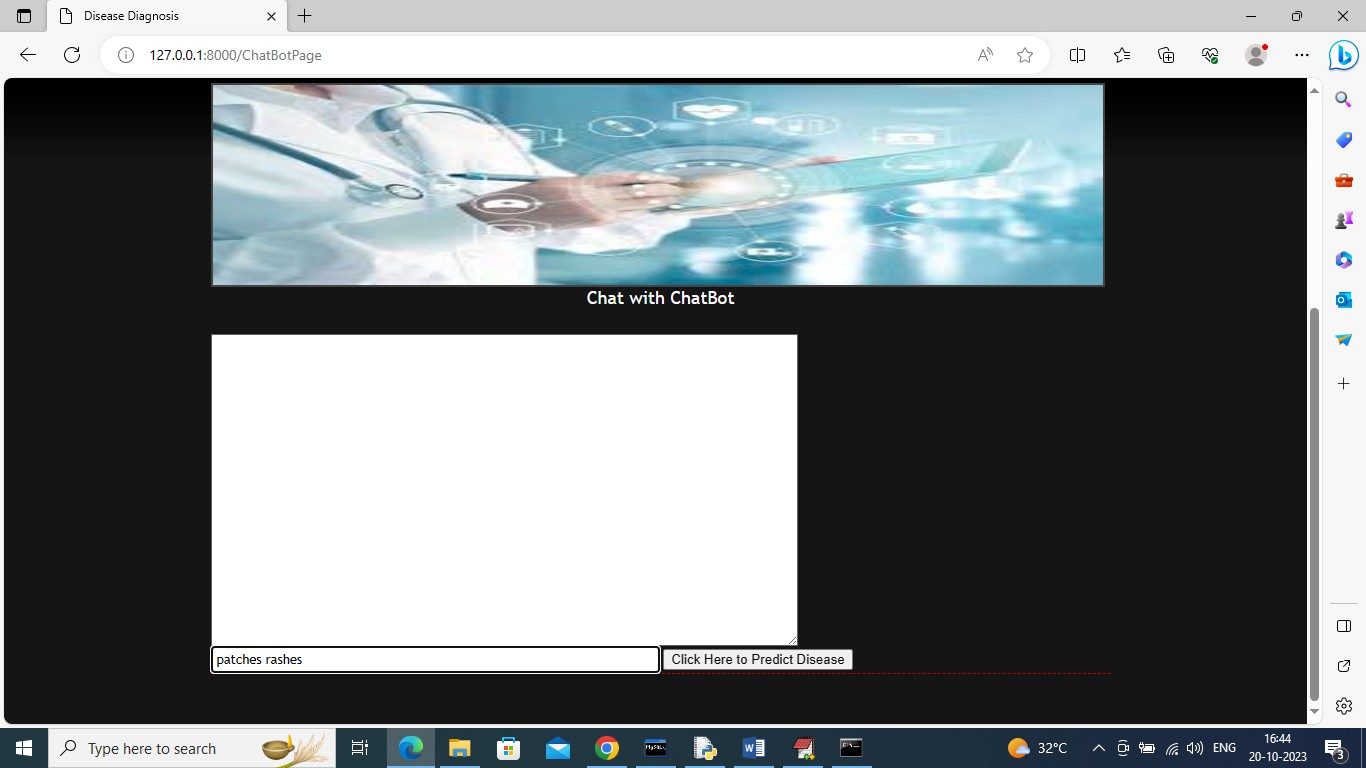


Fig 4.6 select symptoms page

In above Chatbot page just type some symptoms and in above page I gave symptoms as ‘patches rashes’ and then press button to get reply from Chatbot like below screen

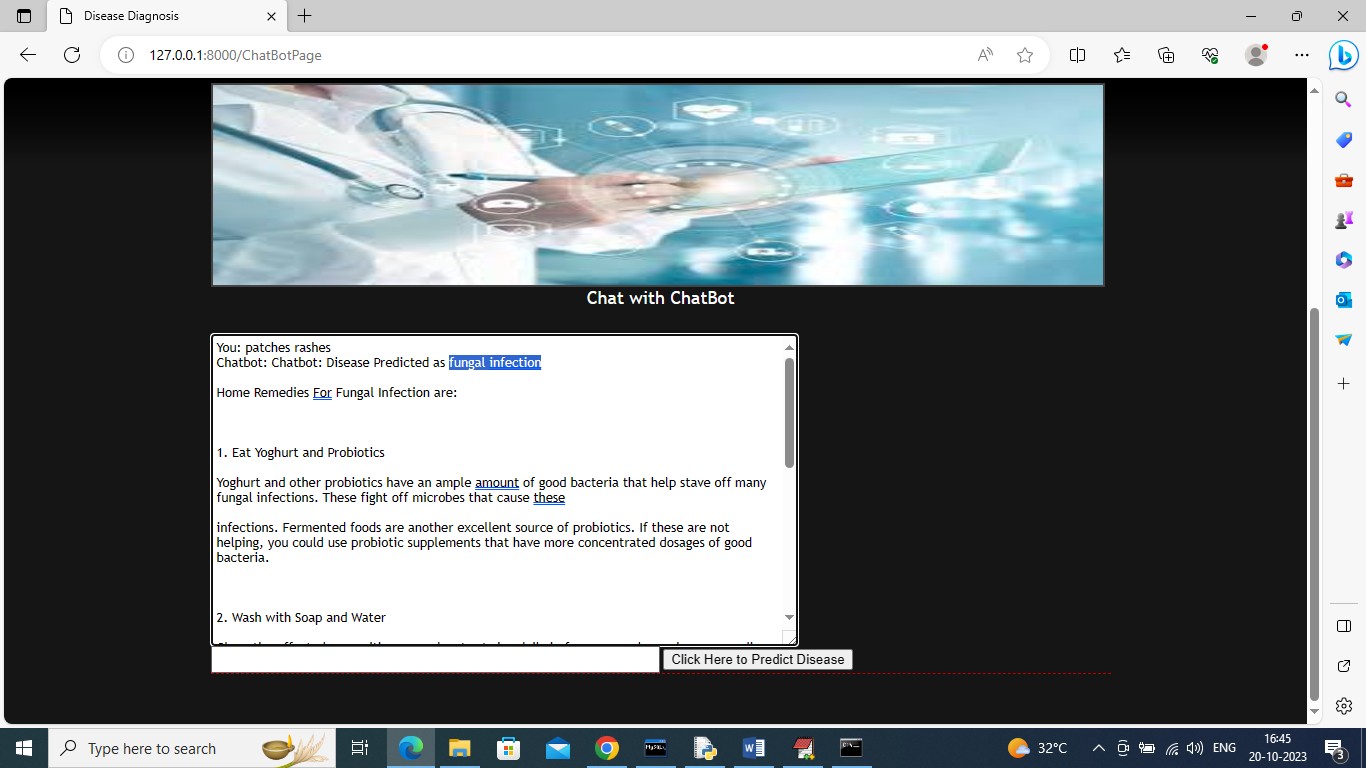


Fig 4.7 disease prediction page

30

In above screen in blue colour text disease predicted as ‘Fungal Infection’ and then in below lines we can see home remedies along with diet details and scroll down above page to view complete details

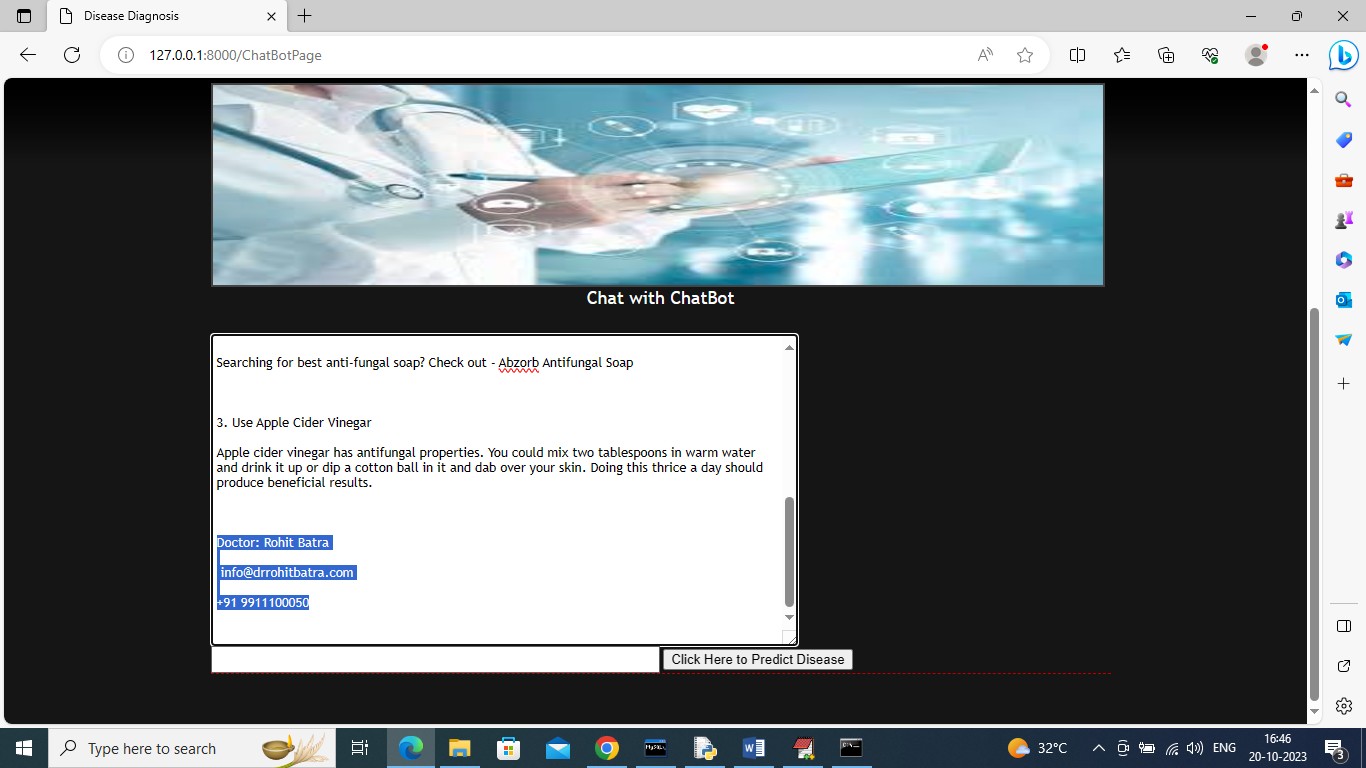


Fig 4.8

In above screen we can see doctor details and then same information will be sent to mail also like below screen

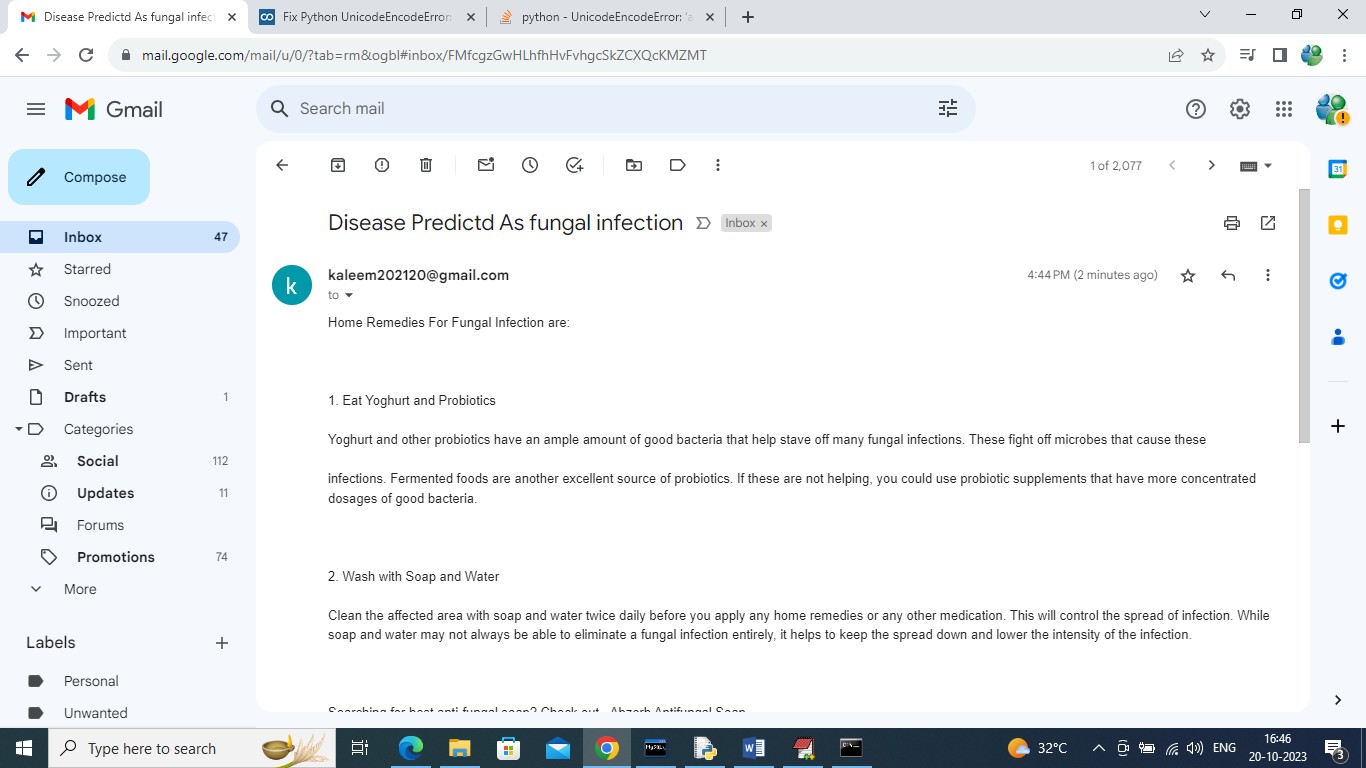


Fig 4.9 E mail notification page

31

In above email we can see disease details with diet and remedies and similarly you can search for any symptoms and below is another example

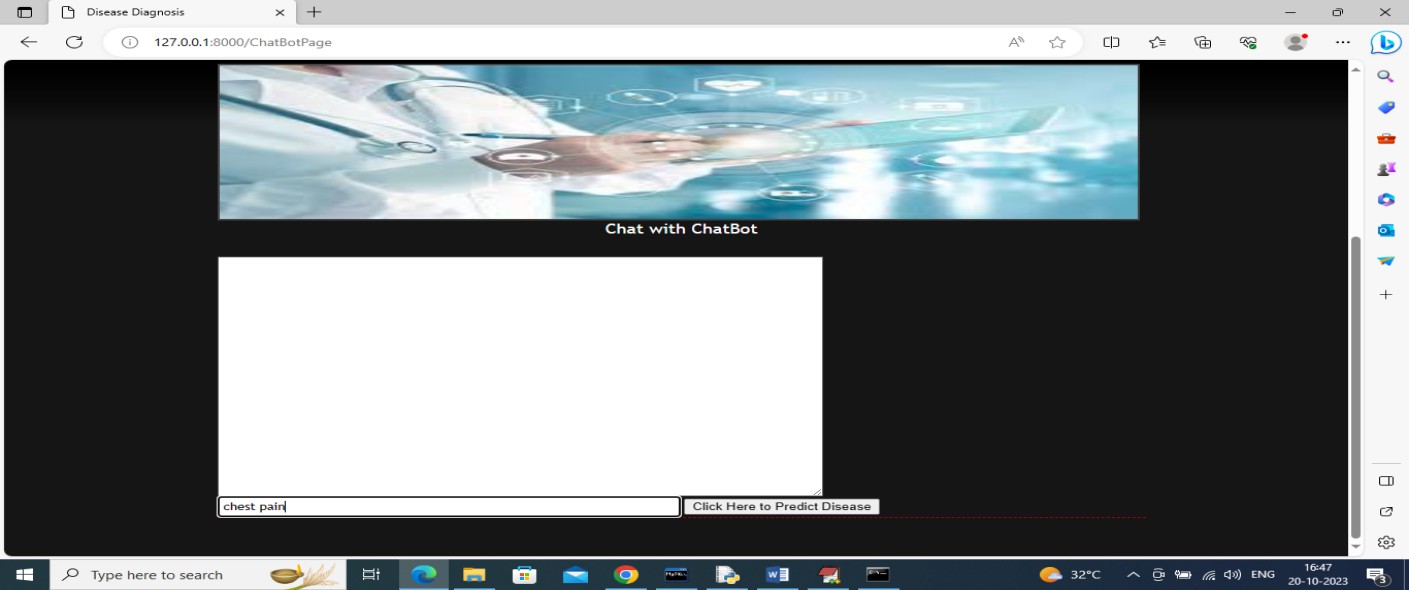


Fig 4.10 symptoms selection page

In above screen I gave symptoms as ‘Chest Pain’ and below is the output

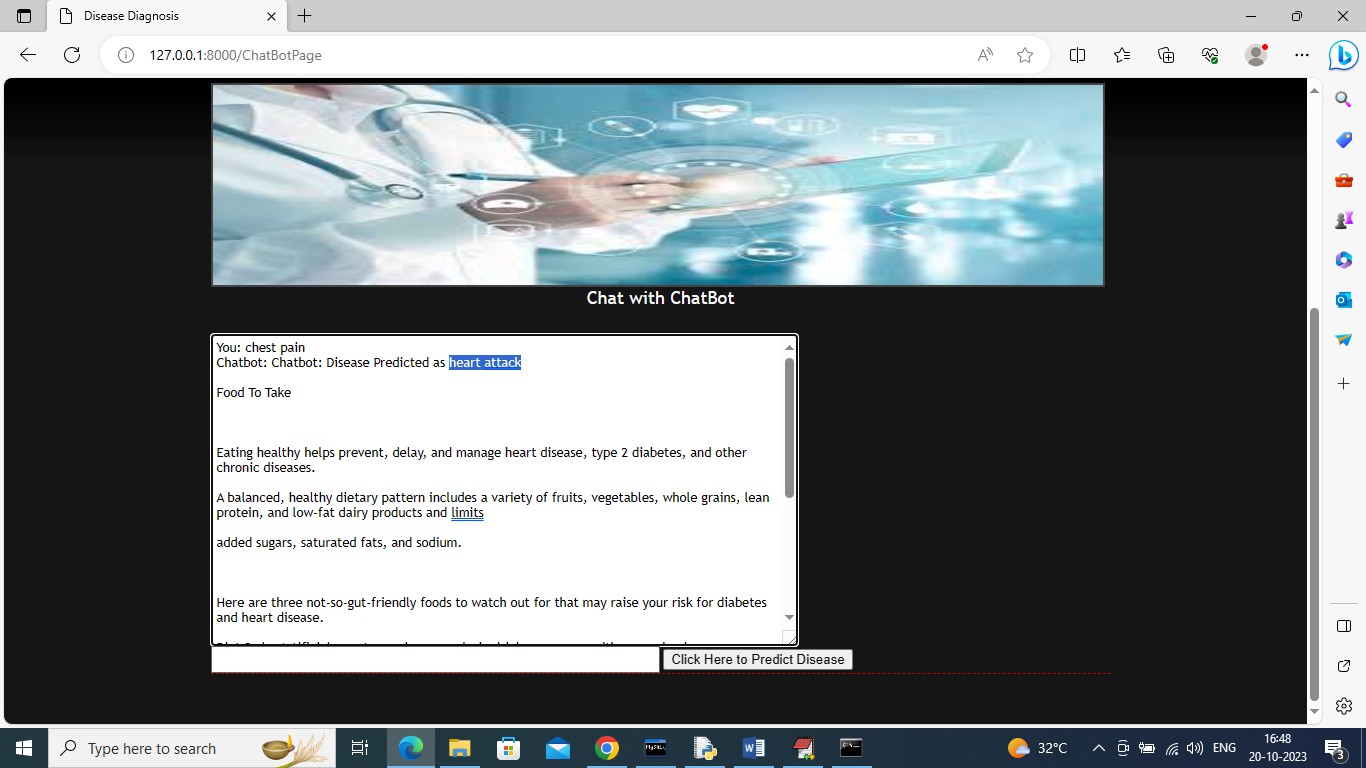


Fig 4.11 disease prediction page

32

In above screen in blue colour text Chatbot predicted disease as ‘Heart Attack’ for symptom ‘Chest Pain’. Similarly you can search for any symptoms and now click on ‘Lifestyle & Disease Information’ link to view static information about disease

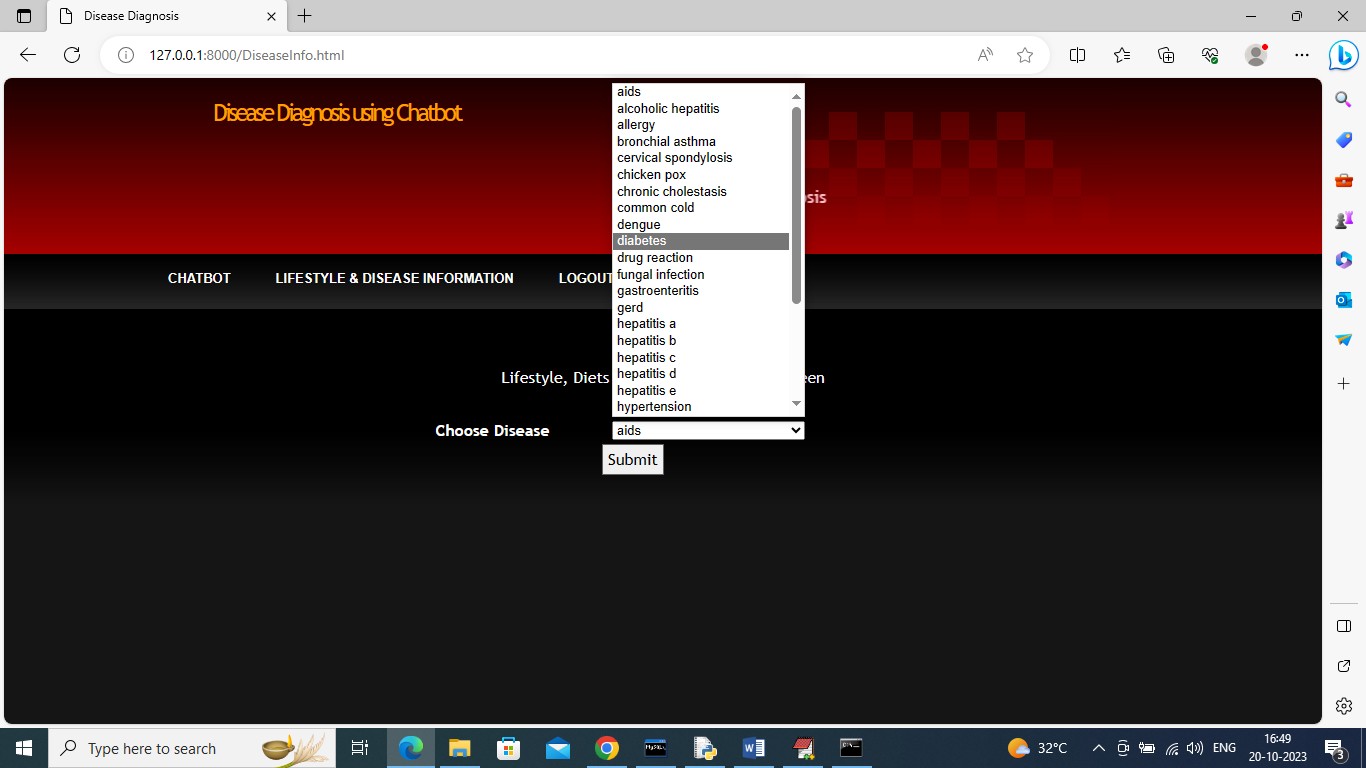


Fig 4.12 unknown disease prediction page

In above screen user can select specific disease and then press button to get disease, diet information like below screen

33

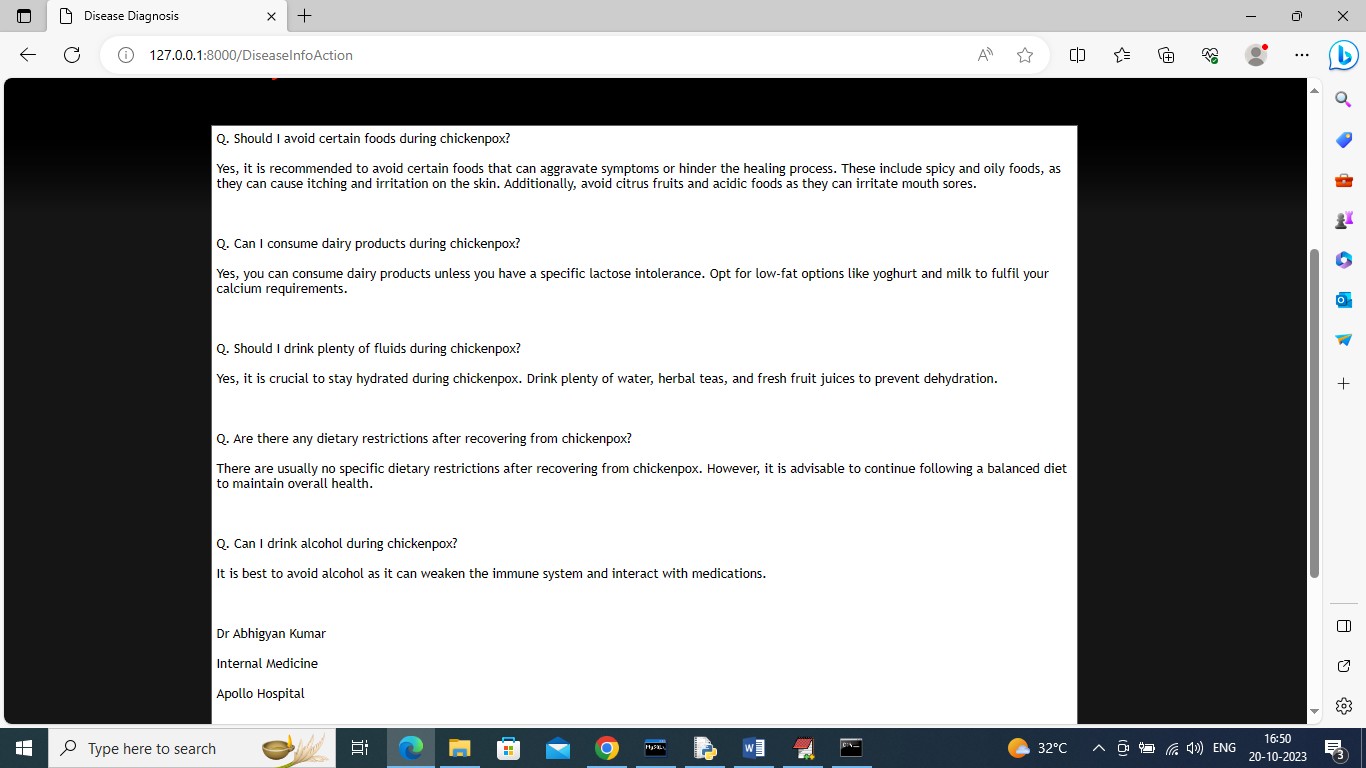


Fig 4.13 diet guidance

In above screen user can see some answers about selected disease along with doctor details.

34

# CHAPTER 5 CONCLUSION AND FUTURE SCOPE

# CONCLUSION

The Symptoms based Disease Prediction represents a significant advancement in healthcare technology, addressing key issues in the existing healthcare system. By leveraging Convolutional Neural Networks (CNN) and a comprehensive data set of disease-symptom associations, this innovative system empowers users to receive accurate disease predictions based on their symptoms. Furthermore, the Chatbot ability to provide personalized dietary recommendations and streamline the doctor appointment booking process enhances the overall healthcare experience. This project marks a promising shift towards democratizing healthcare information and making it readily accessible to a broader population. It not only promotes early disease detection but also educates users on managing their health effectively. The Symptoms based Disease prediction potential to reduce healthcare costs and alleviate the burden on medical professionals is significant. In conclusion, this system showcases the trans-formative impact of AI in healthcare, ultimately contributing to improved health outcomes and enhanced patient care. The development of a symptom-based disease prediction Chatbot project represents a significant advancement in leveraging technology to assist individuals in assessing their health conditions. Through the integration of natural language processing (NLP), machine learning algorithms, and medical knowledge bases, the Chatbot can effectively interpret user-provided symptoms and provide predictions regarding potential diseases.

35

# FUTURE SCOPE

Future work includes enhancing the Chatbot real-time capabilities for doctor appointments, incorporating continuous learning to improve diagnosis accuracy, and expanding the database with more diseases and symptoms.

Enhanced Accuracy: Continuously refining the prediction algorithms and updating the medical knowledge base can improve the accuracy of disease predictions. Incorporating feedback mechanisms from healthcare professionals and users can further enhance the Chatbot's performance.

Integration of Advanced AI Techniques: Exploration of advanced AI techniques such as deep learning, reinforcement learning, and ensemble methods could lead to more robust and accurate disease prediction models.

Personalized Recommendations: Incorporating user-specific data such as medical history, demographics, and lifestyle factors can enable the Chatbot to provide personalized disease risk assessments and recommendations for preventive measures.

Expansion of Features: Beyond disease prediction, the Chatbot can be expanded to offer additional features such as medication reminders, lifestyle advice, and telecommunications with healthcare professionals for further evaluation and guidance.

36

# 7. REFERENCES

1. Rohit Binu Mathew; Sera Elsa Joy; Sandra Varghese; Swanthana Susan Alex “Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning” in 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), IEEE.
2. Ritvik Goel; Viresh Kumar; Dilpreet Kaur Arora; Mohit Mittal “A Machine Learning based Medical Chatbot for detecting diseases” in 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), IEEE.
3. Neeta A. Deshpande Rashmi and Dharwadkar, "A Medical ChatBot", *International Journal of Computer Trends and Technology (IJCTT)*, vol. 60, no. 1, pp. 41-45, June 2018.
4. S. Divya, V. Indumathi, S. Ishwarya, M. Priyasankari and S. Kalpana Devi, "A Self-Diagnosis Medical Chatbot Using Artificial Intelligence", *J. Web Dev. Web Des.*, vol. 3, no. 1, pp. 1-7, 2018.
5. K P Asha Rani, K N Asha, D R Ranjith Kumar, Ranjan Raj Rohan and Rohini, "Medical Chatbot for Disease Prediction using Machine Learning", *IJARESM*, vol. 9, no. 2021.
6. Monica Agrawal, Janette Cheng and Caelin Tran, What's Up Doc? A Medical Diagnosis Bot.
7. Deepmala Kale and Shailendra Aswale, "Doctor Chatbot: Heart Disease Prediction System", *ITEE*, vol. 9, 2020.
8. B. Dhomse Kanchan and M Mahale Kishor, "Study of Machine Learning Algorithms for Special Disease Prediction using Principal of Component Analysis", *International Conference on Global Trends in Signal Processing Information Computing and Communication*, 2016.

37

1. Abien Fred M. Agarap, "Deep Learning using Rectified Linear Units (ReLU)", *cs.NE*, Feb 2019.
2. Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan and Stephen Marshall, Activation Functions: Comparison of Trends in Practice and Research for Deep Learning, Nov 2018.
3. Divya Khyani, B S Siddhartha, N M Niveditha and B M Divya, "An Interpretation of Lemmatization and Stemming in Natural Language Processing", *Journal of University of Shanghai for Science and Technology*, 2021.
4. Shweta J. Patil, "Python - Using Database and SQL", *International Journal of Science and Research*, 2019.
5. Diksha Khurana, Aditya Koli, Kiran Khatter and Sukhdev Singh, "Natural Language Processing: State of The Art Current Trends and Challenges", *ResearchGate*, 2017.
6. Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, et al., "TensorFlow: A system for large-scale machine learning", *ResearchGate*, May 2016.
7. Ms. Sonali, B. Maind and Ms. Priyanka Wankar, "Research Paper on Basic of Artificial Neural Network", *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 1.

38

# LIST OF PUBLICATIONS

**I JOURNALS**

1. P. Bhavani, T. Indu Priya, P. Abhishek, Ms.K. Sangeeta, “Symptoms based disease prediction” International Journal for multidisciplinary research (IJFMR 2024) Bengaluru, India.

Status: Submitted

# II PRESENTATIONS IN INTERNATIONAL CONFERENCE

1. P. Bhavani, T. Indu Priya, P. Abhishek, Ms.K. Sangeeta, “Symptoms based disease prediction” International journal for innovative science and research technology (IJISRT 2024)

Status: Submitted

1. P. Bhavani, T. Indu Priya, P. Abhishek, Ms.K. Sangeeta, “Symptoms based disease prediction” 3rd IEEE International Conference on Artificial Intelligence for internet of Things (AIIoT) 2024 Status: Submitted

39

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | Name of the Student | P. Bhavani  T. Indu Priya  P. Abhishek | | | |
| 2 | Email ID and Phone Number | [20951A0522@iare.ac.in](mailto:20951A0522@iare.ac.in) 9063297145  [20951A0556@iare.ac.in](mailto:20951A0522@iare.ac.in) 7075100314  [20951A0505@iare.ac.in](mailto:20951A0522@iare.ac.in) 7036699866 | | | |
| 3 | Roll Number | 20951A0522  20951A0556  20951A0505 | | | |
| 4 | Date of submission |  | | | |
| 5 | Name of the Guide | Ms.K.Sangeeta | | | |
| 6 | Title of the project work/ research article | Symptoms based Disease Prediction | | | |
| 7 | Department | Computer Science and Engineering | | | |
| 8 | Details of the payment |  | | | |
| 9 | No. of times submitted | **First / Second / Third**  (First time – Free; Second time – Rs 200/-; Third – Rs 500/-;There after multiple of third) | | | |
| 10 | Similarity Content  (%) (up to 25%acceptable) | **1st** | **2nd** | **3rd** | **4th** |
|  |  |  |  |
| **For R & D Centre Use** | | | | | |
| Date of plagiarism check | |  | | | |
| Similarity report percentage | |  | | | |
| R&D staff Name and Signature | |  | | | |
| I / We hereby declare that, the above mentioned research work is original & it doesn’t contain any plagiarized contents. The similarity index of this research work is……………..  **Justification for similarity index:**  ………………………………………………………………...………………...…..…………………………  …………………………………………………………………………………………………………………  …………………………………………………………………………………………………………………  …………………………………………………………………………………………………………………  …………………………………………………………………………………………………………………  ……………………………………………………………………………………………………...…………  ……...………………...………….. | | | | | |

40