UNIVERSITY OF CALCUTTA

BSc. (HONOURS) SEMESTER – VI EXAMINATION, 2023
(UNDER CBCS)

PROJECT WORK

DOT 1.0: Directors of Truth

SUBJECT CODE: CMSA

PAPER CODE: CC-6-13-Pr & CC-6-14-Pr

CERTIFICATE

This is to certify that the project entitled ***"DOT 1.0: Directors of Truth"*** has been done and submitted successfully by the undersigned students, as part of their University of Calcutta curriculum for the 3-year undergraduate degree course in **B.Sc. Computer Science (Hons.)**, under Prof. Abhishek Dey, presented for the 6th semester examination of the courses **CMSA-CC-6-13-Pr & CMSA-CC-6-14-Pr**, held on **4th August, 2023**. Furthermore, this is an original piece of work, and meets all the necessary criteria, to be accepted as a project work submitted for a Bachelor's degree programme in Computer Science.

GROUP MEMBERS:

DEBOSMITA BANDYOPADHYAY
OINDRILA PATHAK
TANISHA DEY

ACKNOWLEDGEMENT

We take the opportunity to express profound sense of gratitude, appreciation and respect to all those who helped us throughout the development of our project titled ***'DOT 1.0'***. Your collective efforts have been instrumental in overcoming challenges, solving errors in code, and making our endeavours a success.

First and foremost, we are deeply indebted to our project supervisor, **Professor Abhishek Dey**, whose unwavering encouragement has been a constant source of inspiration. His belief in our vision has spurred us on, motivating us to strive for excellence. Without his assistance and dedicated involvement in every step throughout the process, our project would have never been accomplished. We would like to convey our heartfelt gratitude to our professors of the Department of Computer Science, Bethune College. Their enthusiasm for the project has always made a strong impression on us.

We thank our families for their trust on us through all ups and downs of life. A special mention goes to the vibrant online communities that have embraced us with open arms. Their willingness to share knowledge, assist with troubleshooting, and offer valuable insights have been invaluable assets in tackling the complexities of coding and programming.

Last but not least, we must acknowledge the tireless efforts of our own team and our beloved machines. Together, we form a dynamic synergy that has proven to be a formidable force in addressing challenges with unwavering precision and dedication. Our collective pursuit of excellence, coupled with the computational power of our machines, has led us to achieve feats we once thought were beyond reach.

In conclusion, we wish to extend our sincerest appreciation to all those who have played a role in our success whose generosity, expertise, and unwavering support have made all the difference. We remain committed to pushing the boundaries of innovation and excellence, and we look forward to continuing this incredible journey together.

CONTENTS

1. ABSTRACT

In today's complex and interconnected world, crime has become an unfortunate and persistent reality. From petty theft to organized crimes and violent offenses, the impact of criminal activities extends far and wide, affecting individuals, communities, and nations at large. As societies strive to maintain law and order, it has become increasingly crucial to develop efficient and comprehensive systems to manage and combat crime effectively. Thus, if we can come up with ways to record and predict danger, in detail, before it occurs, or come up with a "model" that can assist police officers, it would lift the burden of police and help in preventing crimes. To achieve this, we suggest including databases to store data and access data easily, machine learning (ML) and computer vision algorithms and techniques to analyse the stored data. Here we try to develop a short-term forecast method using FIS which can be used by both the law enforcing agencies as well as common people alike for immediate decision making. In summary, DBMS, ML and computer vision techniques can bring about an evolution in society at large.

2. INTRODUCTION

D.O.T stands for 'Directors Of Truth'. The necessity for D.O.T stems from the multifaceted challenges posed by criminal activities. Traditional methods of law enforcement, while valuable, often fall short on the face of evolving criminal tactics and technological advancements. To effectively counter and prevent crime, law enforcement agencies, policymakers, and communities need access to reliable and integrated systems that can collect, interpret, analyse, and disseminate information efficiently.

Computer vision is a branch of artificial intelligence that trains the computer to understand and comprehend the visual world, and by doing so, creates a sense of understanding of a machine's surroundings. It mainly analyses data of the surroundings from a camera, and thus its applications are significant. In D.O.T we use it for face recognition. We have used computer vision in two of our modules,

> D.O.T aids in the investigation by checking if the missing person for whom a report is being filed is already in its vast database (has been found and registered under the label of un-detected people). Thus, it helps in finding information about a missing person in minutes.
>
> ID verification helps the common citizens to identify suspects, frauds, thereby increasing the security of commoners. Timely access to accurate and comprehensive data not only ensures that criminals are brought to justice but also serves as a deterrent to potential offenders, ultimately contributing to the overall reduction of crime.

D.O.T implements fuzzy logic which is used for danger prediction in a selected area. The proposed system employs fuzzy logic to handle the vagueness and ambiguity present in data related to man-made incident. Fuzzy sets are utilized to represent the degrees of membership for various related parameters, such as past crime records of that area, rate of reports lodged, rate of missing cases filed and the danger perceived by the residents in the selected area. Fuzzy rules are defined based on analysis of data available, research done in the given topic and logic to establish the relationships between these parameters and the likelihood of criminal incidents. By aggregating and analysing these fuzzy outputs, the system generates danger predictions that can be visualized and interpreted.

D.O.T can play a pivotal role in supporting law enforcement agencies' operational efficiency. Traditional paper-based processes are often cumbersome and time-consuming, making it difficult for officers to access critical information promptly. By implementing an integrated system, law

enforcement personnel can streamline their workflows, share real-time data, and lodge cases effectively in minutes, allowing authority to respond rapidly. We use CSS and Java Script to create a simple and effective front end.

Further, D.O.T gives the common men a deeper and more transparent insight in the workings of crime combatting authorities. They can view the total number of cases being solved as well as those pending in real time. They can also check the current status of the cases they have lodged with minimal hassle, using database management systems thereby saving a lot of time and effort.

3. RELATED WORKS

In the field of face recognition systems, various research works have been conducted. One notable study [6] proposes a robust face recognition algorithm based on deep learning techniques, achieving high accuracy in identifying individuals from facial images. The approach leverages convolutional neural networks (CNNs) for feature extraction and classification, leading to reliable and efficient face recognition systems.

A research paper [11] utilizes Fuzzy Logic and Deep learning based on the crime type, analyze and identify important crime hot-spots and pre-diction and prevention over the protection safety of future crimes.

Another paper is "A REAL-TIME CRIME RECORDS MANAGEMENT SYSTEM FOR NATIONAL SECURITY AGENCIES"[13] which discusses the design and implementation of a computerized real-time Crime Records Management System (CRMS) for the Nigerian Police Force (NPF).

4. METHODOLOGY

4.1.Data Flow Diagram of the System:

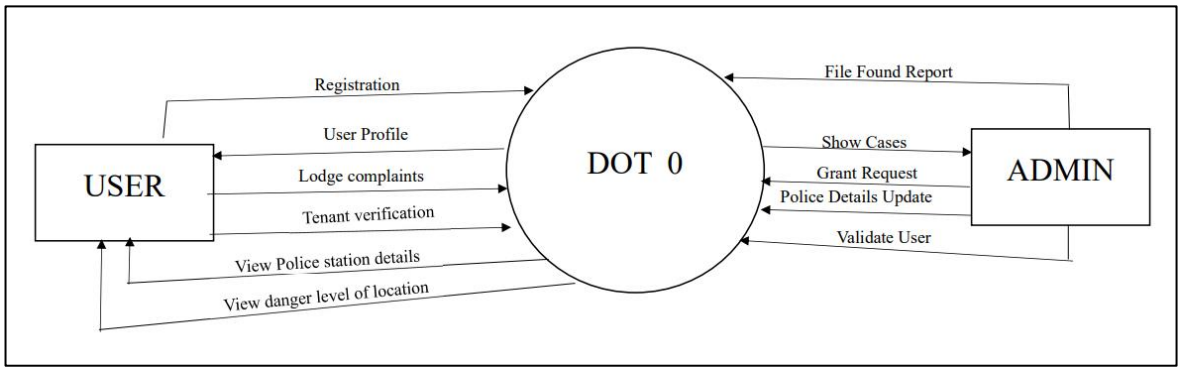

Figure 1.1*:* Level-0 Data Flow Diagram (Context Diagram)

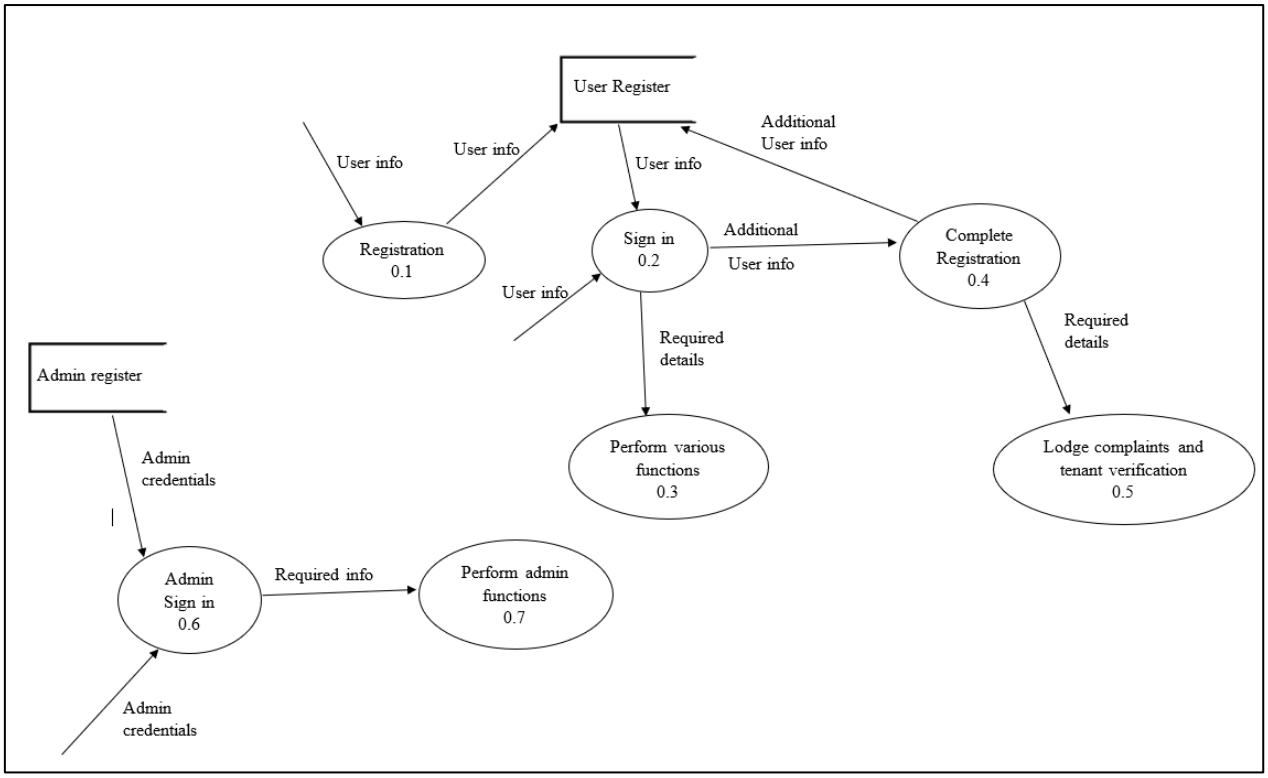

Figure 1.2*:* Level-1 Data Flow Diagram (DFD)

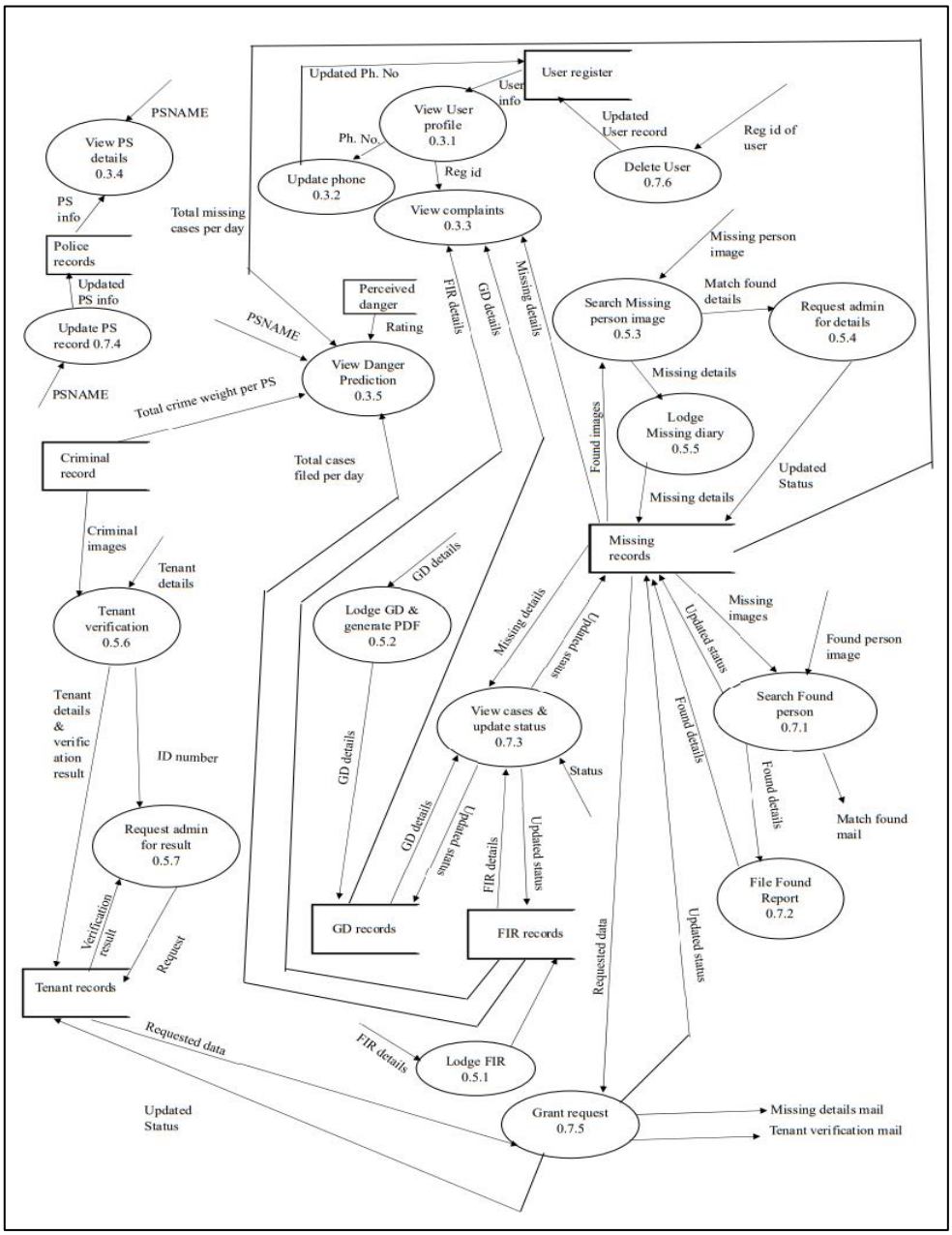

Figure 1.3*:* Level-2 Data Flow Diagram (DFD)

4.2. Website Landing

This is the very first page of the website which provides information about the website, all the services of the website, feedback form for user and user login, sign up and admin login options respectively.

(USER END)

4.3.User Sign In

The website is so designed that a user will be denied access without logging in to the website first. Hence, the Sign in Page acts as the entry point. This page requires the user to give the correct username and password combination in order to log in to the website. If the user cannot remember his/her password, he/she can change it using the 'Forgot password?' option. The process of allowing only registered users is implemented because to generate profiles of registered candidates on the website and to keep a record of the services the user is using related to compliance or identity verification on the website.

4.4.User Registration

Any user who does not have an existing account in the website, must register first in order to continue.

The registration page collects the following information from the user.

1. Name: The user must enter his/her full name.
2. Email Id: The user must enter his/her active email id.
3. User Name: The user must enter a valid name.
4. Password: A strong password must be used and remembered for future logging in.

These four basic pieces of information are collected in order to make the entire registration process hassle-free and fast. Upon submission, the validity for each of the inputs is checked and allowed only after passing the respective checks. In order to verify the e-mail Id, a verification code is sent to the e-mail e-id registered by the user, by entering which the user completes his/her initial registration. These are stored in the database and the values are matched every time the user tries to log in to the website.

4.5. Menu Page

The main page of the website is the Menu Page. It is an essential component of the website since it explains exactly what DOT has to offer. It contains links to all sections, including 'Complete Registration', 'View Profile', 'Complaints', 'Identity Verification', 'Nearest Police Station', 'location Security', as well as the opportunity to log out of the website. There are three sections on the menu page that shows Daily Logins, Total number of lodged Complaints and Total number of Feedbacks respectively. Also, the menu page has a message section from our website to the general public. Along with these, the menu page provides all the helpline numbers of Kolkata Police.

4.6.Location Security

Any user who is interested to know about the danger level of any area covered by Kolkata Municipal Corporation must first select the Police Station name.

On selecting the police station name, the user is shown a gauge which displays the danger percentage in the selected police station. It also displays other security factors like:

1. Past Crime Level
2. Number of Reports Filed on a daily basis
3. Missing Count level
4. Perceived Danger level from the citizens of that area

It also displays a bar graph showing all the police stations in our database so that we can make a comparative study about the danger level of one Police Station with respect to another. the fuzzy code will be executed and we get the danger level based on the current data in the database in the form of a meter gauge.

(ADMIN END)

4.7.Admin Login

The website admin can login through Admin Login page which is only visible to the user but not accessible. This page requires the admin to give the correct username and password combination in order to log in to the admin section.

4.8.Admin Dashboard

Admin Dashboard page is the primary page of admin which contains important links to all sections of admin, including 'Found Form', 'Show Complaint Status' to view all complaints of all

categories and update status if required, 'Grant Request' to accept valid user request, 'Police Details Update' to update police records, if necessary, 'User ID Verification' to check the validity of the user, as well as the opportunity to log out of the website.

4.9.Danger Level Prediction System:

Fuzzy Input Calculation

The four crisp input values are:

1. Previous Crime Record: Here a weight is given to different types of crimes committed in the selected police station during the last 5 years. The aggregate of the weights is calculated to generate a crisp value.
2. Reports filed: Here a weight is given to different types of man-made incidents under the selected police station and its aggregate is calculated with respect to each day and an average of all the values generated for each day is calculated to generate a crisp value.
3. Missing reports filed: Here the number of missing reports filed per day is calculated in the selected police station and its aggregate is calculated with respect to each day and an average of all the values generated for each day is calculated to generate a crisp value.
4. Perceived danger: Here a survey is conducted to find what people think about the danger levels in the area of the police station they reside in and an average of all the feedbacks provided is calculated to generate a crisp value.

Table 1 displays common crimes are divided into 4 categories and the weights given to different crimes are based on the seriousness of each crime.

Crime Level	List of Incidents
Level 1	Accident, Child Labour, Witch Act, Domestic Violence, Dowry Act, Eve Teasing, Gambling, Bullying
Level 2	Swindle, Extortion, Theft, Loot, Fraud
Level 3	Kidnapping, Smuggler, Protection racketeering, Loan sharking, Drug-trafficking
Level 4	Murder, Terrorism, Rape, Organized Riot, Robbery

Table 1: Represents all possible incident which affect security of a given area

Note: Here we have excluded all forms of cybercrimes because it does not affect the physical security of a given area.

Fuzzy Inference System

A fuzzy inference system is a framework that uses fuzzy logic to make decisions or perform computations based on fuzzy rules and fuzzy sets. It involves several steps to process the fuzzy inputs and generate crisp outputs. The predictive power of a fuzzy inference system lies in the representation and manipulation of uncertainty and imprecision. By using fuzzy logic, the system can capture the gradual and non-binary relationships between input variables and generate predictions on the danger level with respect to a given police station which is dynamic as more data is inserted in the system. Here are the detailed steps involved in a fuzzy inference system:

Fuzzification

Here we convert the crisp input values into fuzzy values by mapping them to appropriate fuzzy sets using membership functions. The four crisp input values are: Previous Crime Record, Reports filed, Missing reports filed, Perceived danger.

We determine the degree to which each of the inputs belongs to the appropriate fuzzy set(s) through fuzzy membership functions. We have considered three linguistic variables namely: low, medium and high. **Table 2 and Figure3, Figure4, Figure5, Figure6** shows the range and type of MF used to represent each of the linguistic variables. Depending upon the input value, we will get degree of membership in [0, 1] for the variables with respect to each of the three fuzzy sets representing three linguistic variables.

Inputs	**Linguistic Variables**	**Range**	**MF Type**
Past Crime	Low	[0, 10]	Open Left
	Medium	[8,22]	Trapezoidal
	High	[20,30]	Open Right
Reports Filed	Low	[0,9]	Open Left
	Medium	[6,14]	Trapezoidal
	High	[11,20]	Open Right
Missing Count	Low	[0,9]	Open Left
	Medium	[6,14]	Trapezoidal
	High	[11,20]	Open Right
Perceived Danger	Low	[0,4]	Open Left
	Medium	[3,7]	Trapezoidal
	High	[6,10]	Open Right

Table 2: **The range and type of MF used to represent each of the linguistic variables**

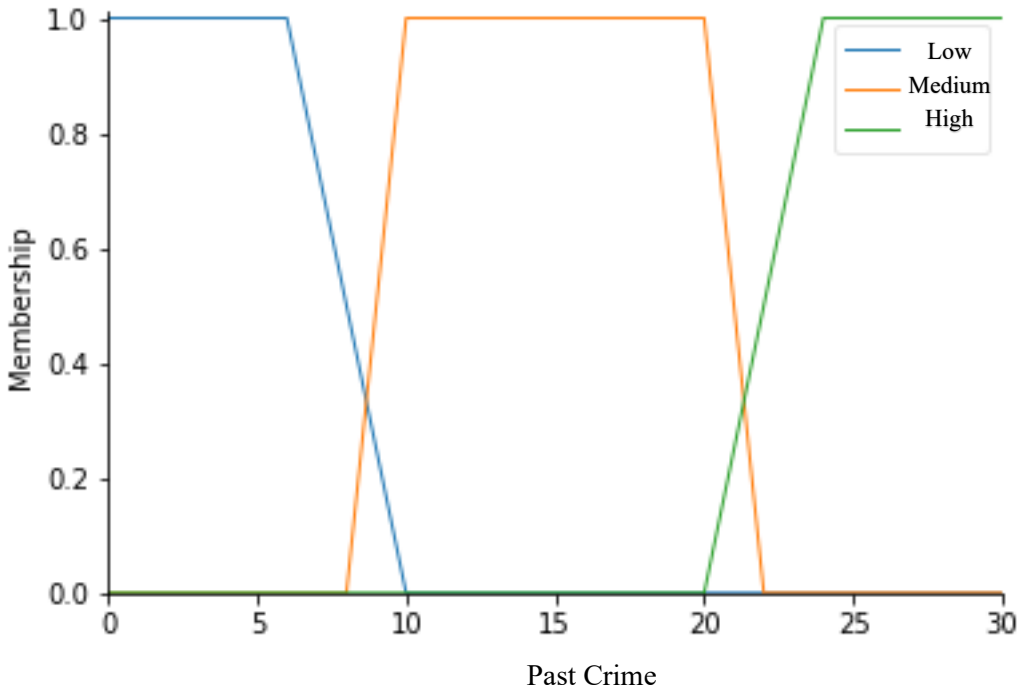

Figure 2: Graph displaying the membership profile of past crime

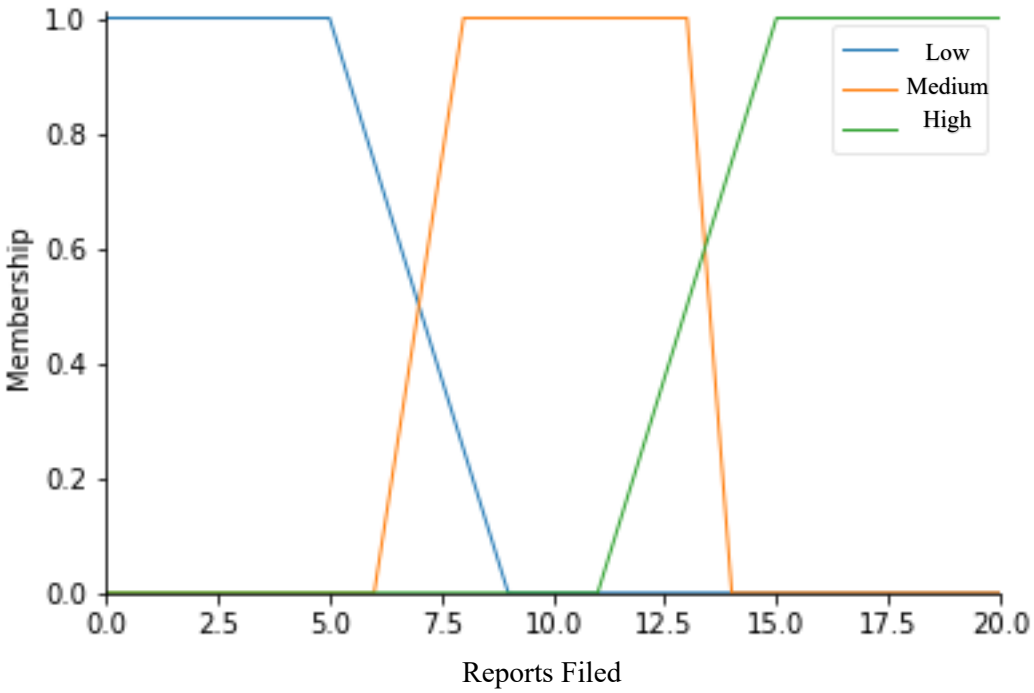

Figure 3: Graph displaying the membership profile of rate of reports (other than missing) filed

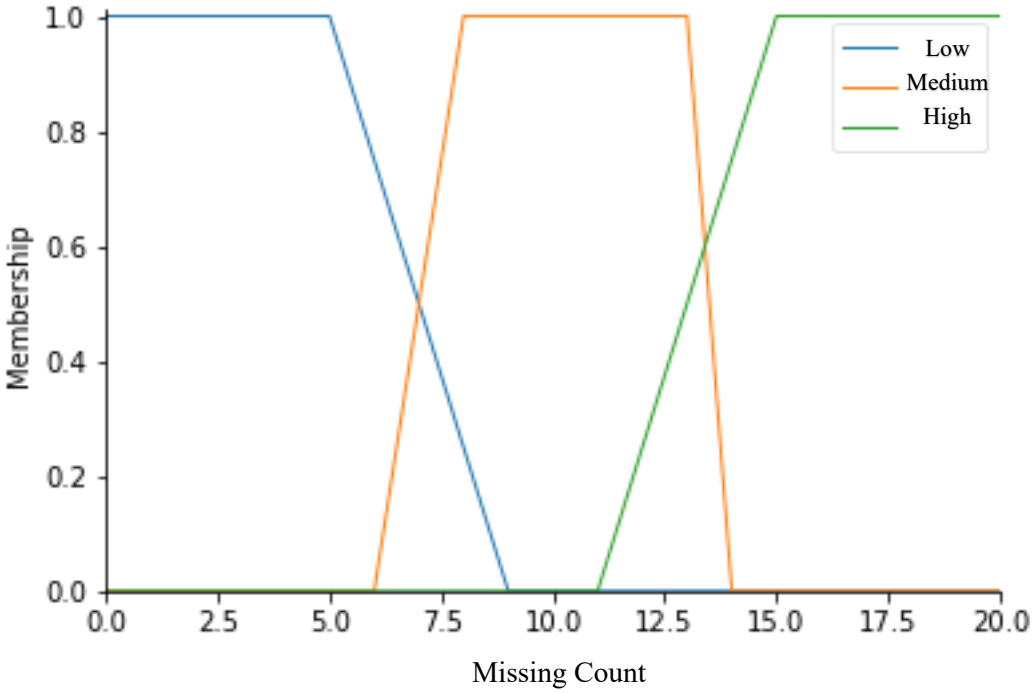

Figure 4: Graph displaying the membership profile of rate of missing reports filed

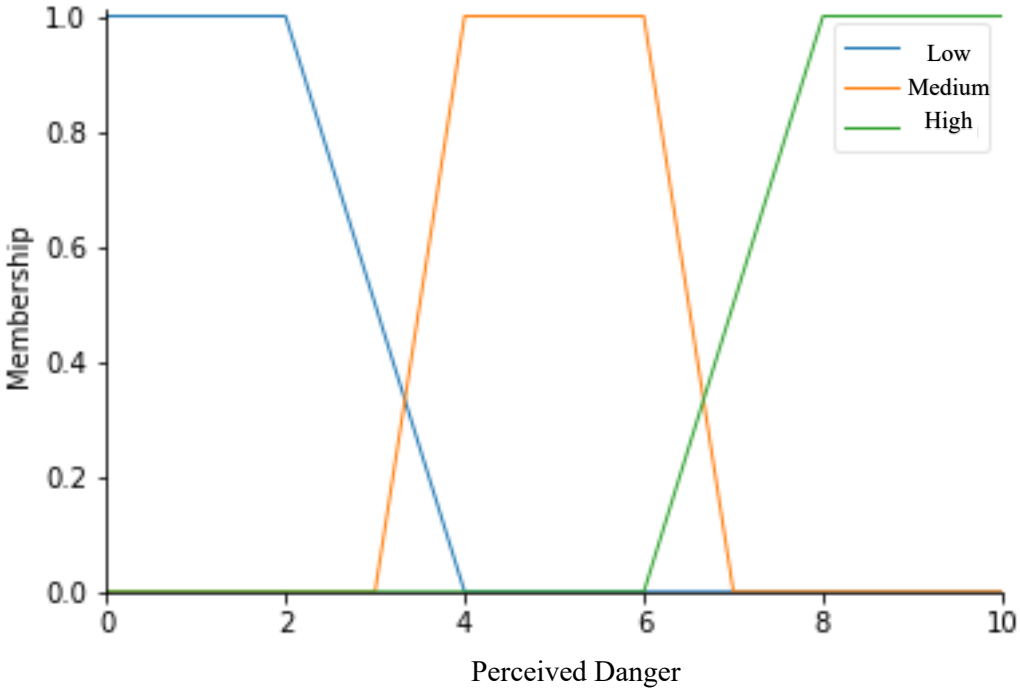

Figure 5: Graph displaying the membership profile of perceived danger rate

It must be noted that the use of overlapping MF represents that a member of a particular fuzzy set can also have partial membership with respect to some other fuzzy set. In other words, it represents that each of the four inputs can have partial membership with respect to more than one fuzzy set. Thus, by using the above mentioned MFs, the crisp values representing Past Crime, Reports Filed, Missing Count and Perceived Danger are fuzzified (i.e. membership values of each of the three crisp inputs with respect to five fuzzy sets are determined through fuzzy MFs). For example, if the crisp input value representing perceived danger rating at 3.2 then our calculation predicts: membership value with respect to fuzzy set Low is 0.4 and with respect to fuzzy set Medium is 0.2, as shown in Figure 6**.**

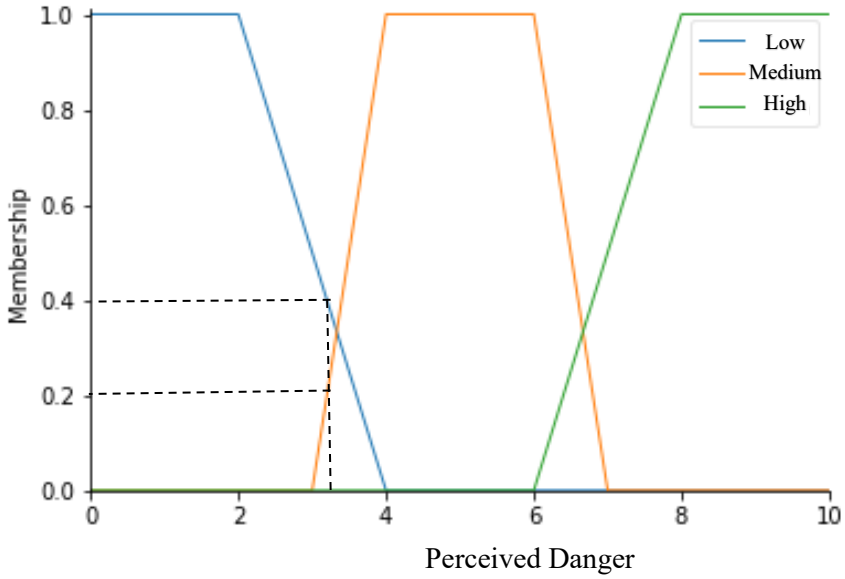

Figure 6: Graph displaying the membership profile of perceived danger rate which shows membership values for a given crisp input

After we calculate the membership degrees of the input values in each fuzzy set we move on to:

Rule Evaluation

Here, we define a set of fuzzy rules that represent the knowledge or expert understanding of the problem domain. Each fuzzy rule consists of an antecedent (input condition) and a consequent (output action).

For the rule base used in this system, we had a set of 3^4 i.e. 81 possible cases fuzzy rules out of which 10 are shown in **Table 4**, which are evaluated by implication process. The priority order of the variables is: Past Crime, Reports Filed, Missing Count and Perceived Danger (Higher to lower). For each rule, the input to the implication process is the single value obtained by evaluating the antecedent through application of fuzzy operators. The output is one of five fuzzy sets representing linguistic variables: Very High, High, Medium, Low, Very Low to describe danger level. This fuzzy output set is obtained by reshaping the fuzzy set corresponding to the consequent part of the rule with the help of the number given by the antecedent. **Table 3 and** Figure**7** shows the range and type of MF used to represent each of the linguistic variables for the output fuzzy sets. The individual fuzzy sets obtained by evaluating the rules is combined by taking algebraic sum of all inputs into a single resultant fuzzy set. This aggregation process takes the truncated membership profiles returned by the implication process as its input, and produces one fuzzy set as the output.

Inputs	**Linguistic Variables**	**Range**	**MF Type**
Danger Level	Very Low	[0, 30]	Open Left
	Low	[10,60]	Trapezoidal
	Medium	[40,90]	Trapezoidal
	High	[70,100]	Open Right
	Very High	[90,100]	Open Right

Table 3: Range and type of MF used to represent each linguistic variable for the output fuzzy sets

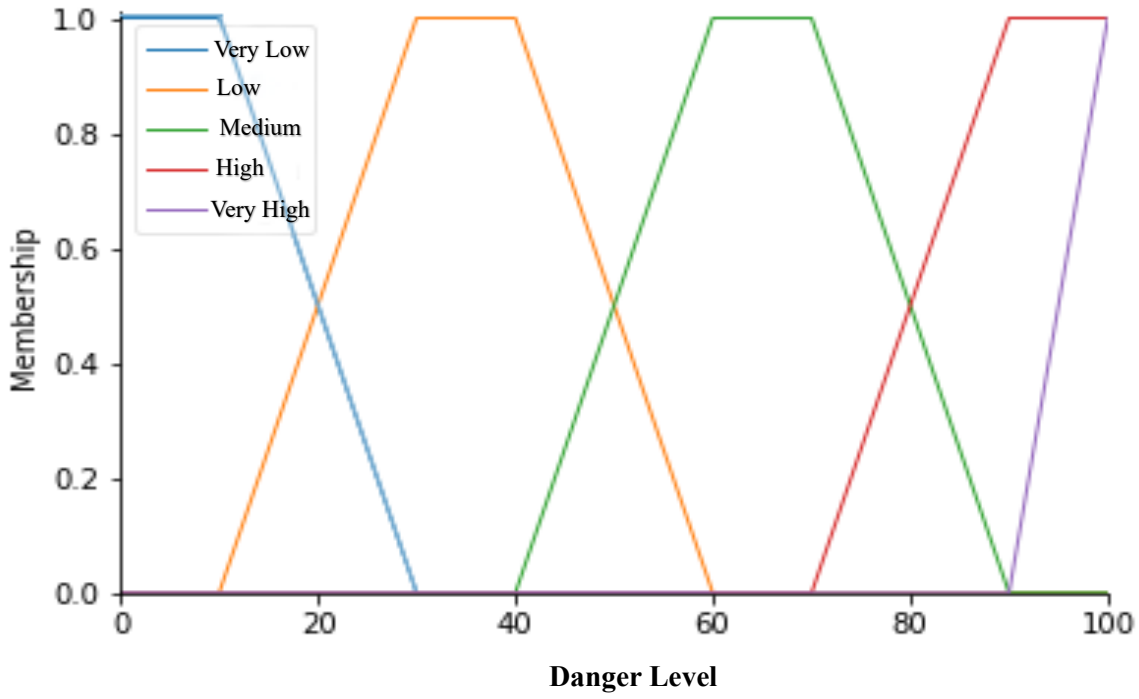

Figure 7**:** Graph displaying the membership Profile for Danger Level

Past Crime		Reports Filed		Missing Count		Perceived Danger	Danger Level
Low	AND	Low	AND	Low	AND	Low	Very Low
Low	AND	Low	AND	Medium	AND	Low	Very Low
Medium	AND	Low	AND	Low	AND	Low	Low
Medium	AND	Medium	AND	Low	AND	Low	Low
Medium	AND	Medium	AND	Medium	AND	Medium	Medium
Medium	AND	Medium	AND	Low	AND	Medium	Medium
High	AND	Medium	AND	High	AND	High	High
High	AND	Medium	AND	Medium	AND	Medium	High
High	AND	High	AND	Medium	AND	Low	Very High
High	AND	High	AND	High	AND	High	Very High

Table 4: Rules used in the system

We thus evaluate the degree of compatibility or matching between the fuzzified input values and the antecedents of each fuzzy rule and calculate the activation strength or firing strength of each rule, which represents the degree of truth or applicability of the rule.

Aggregation

Combine the activated rules to determine the overall contribution of each fuzzy set to the output. Aggregate the membership degrees of the consequent fuzzy sets from the activated rules. This step involves considering the overlapping or conflicting fuzzy sets and deciding how their membership degrees should be combined or merged.

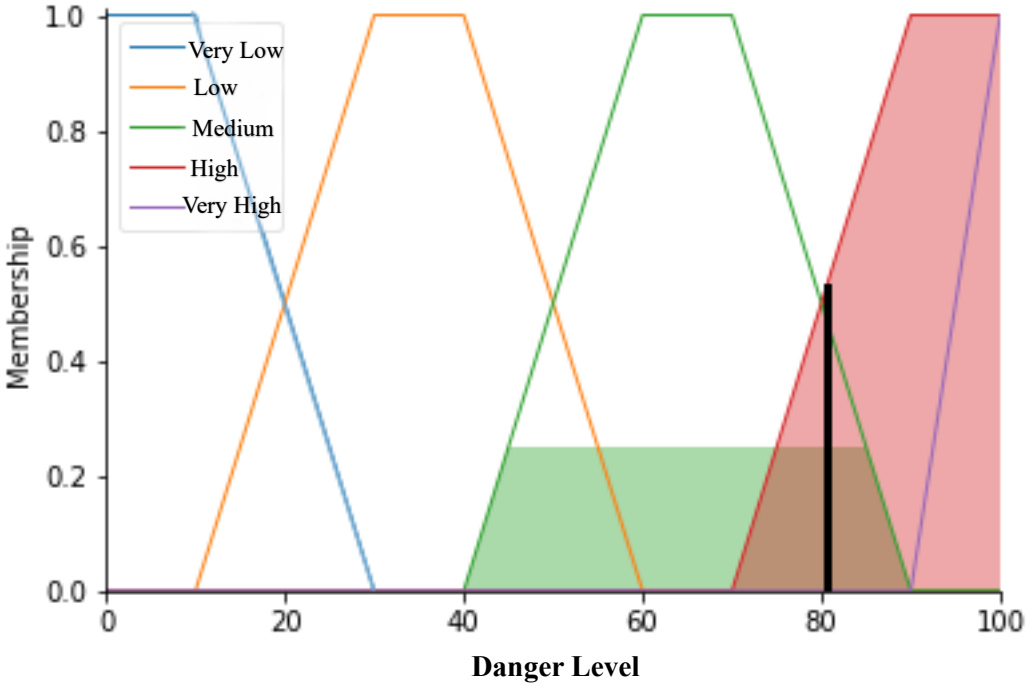

Figure 8: Graph displaying the final output of the FIS

Here the input values are-
Past Crime:28 Reports Filed:8 Missing Count:9 Perceived Danger:9

Defuzzification

Convert the aggregated fuzzy output into a crisp output value that can be used for further processing or decision-making. Here we use the centroid method. We determine the crisp output value based on the distribution of membership degrees in the aggregated fuzzy set. The crisp value obtained in this step gives the final output (% of danger) of the FIS.

Thus, the Danger % of the Figure8 above is calculated to be 80.53%.

4.10. Face Recognition Basic Algorithm

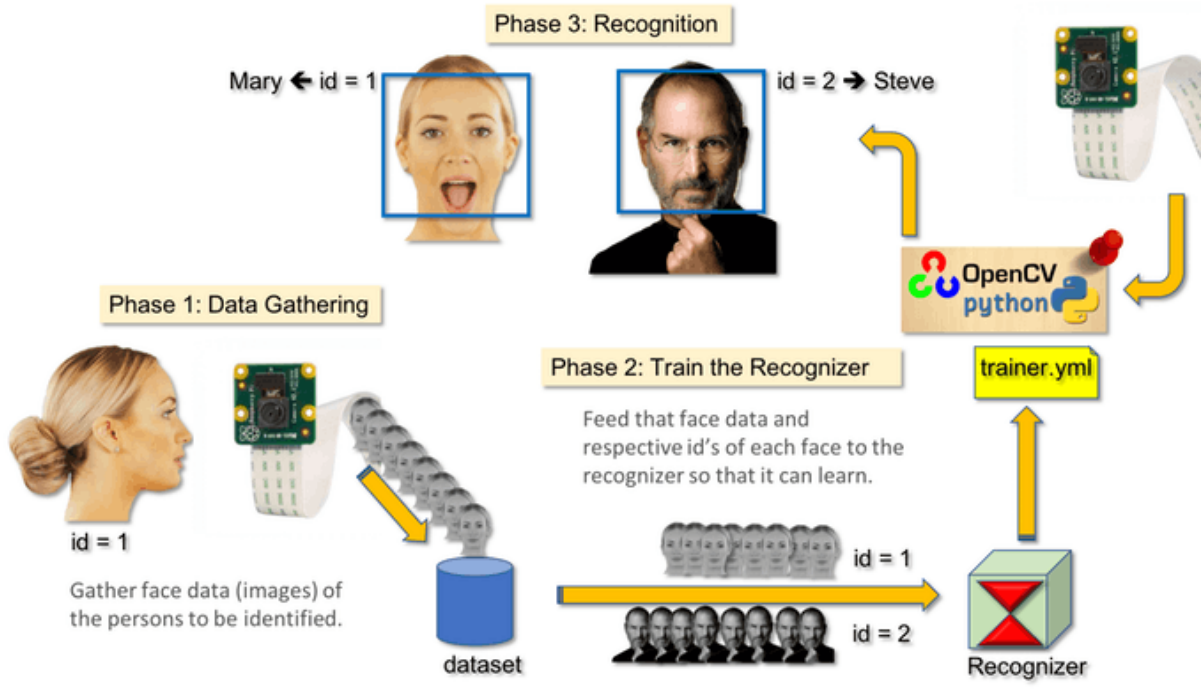

Figure 9: The block diagram that resumes 3 Phases of the Face Recognition model

The 3 Phases of the Face Recognition model are:

Phase 1: Face Detection and Data Gathering

- Loading Images and Converting to RGB

  We load all the images currently in the database and store it as a list. We also load the image with which we are supposed to compare the images in the database. Let's call this image 'test image'. Once imported the image has to be converted to RGB.

- Find Faces Locations

  First, we will find the faces in our images. This is done using HOG (Histogram of Oriented Gradients) at the backend. HOG is a popular computer vision algorithm used for object detection, particularly in images. It analyses the gradients or the intensity variations of an image by dividing it into small regions called cells. For each cell, HOG computes a histogram of gradient orientations, representing the distribution of edge orientations within that region.

  The Histogram of Oriented Gradients (HOG) descriptor is a feature extraction technique used in image processing, particularly for object detection. The process involves computing the image gradient in the x and y directions, dividing the image into 8x8 cell blocks, and creating histograms of gradients for each block. Normalization is applied to make the descriptor invariant to lighting conditions. The histograms are then concatenated to form a final feature vector. This technique reduces the image's complexity and improves the performance of object detection algorithms like Support Vector Machines (SVM) for face detection and recognition tasks. The resulting feature vector is used to train classifiers for face detection purposes.

HOG is effective for face detection because faces have distinct patterns of local gradients due to the presence of facial features like eyes, nose, and mouth. These features contribute to the unique orientation patterns captured by the HOG descriptors. By analysing the gradients and their orientations, HOG can identify potential face regions within an image.

Figure 10: Image generated using HOG algorithm used for detecting the features of the face

- Warping faces to remove unwanted rotations using transformations.

Phase 2: Train the Recognizer

After detecting and aligning the face, the image is typically fed into a pretrained neural network that outputs a set of 128 measurements, known as face embeddings or face encodings. These measurements are unique to each particular face and capture the essential characteristics of the face in a high-dimensional vector representation.

The process of generating face embeddings involves passing the aligned face image through a deep neural network, such as a convolutional neural network (CNN), that has been trained specifically for face recognition tasks. The neural network has learned to extract discriminative features from faces such as such as the shape of the eyes, nose, and mouth, as well as the texture and spatial relationships between these features and map them into a compact representation.

The training process works by looking at 3 face images at a time:

1. Load a training face image of a known person
2. Load another picture of the same known person
3. Load a picture of a totally different person

Then the algorithm looks at the measurements it is currently generating for each of those three images as shown in Figure**11**. It then tweaks the neural network slightly so that it makes sure the measurements it generates for #1 and #2 are slightly closer while making sure the measurements for #2 and #3 are slightly further apart. After repeating this step millions of times for millions of images of thousands of different people, the neural network learns to reliably generate 128 measurements for each person. Any ten(approximate) different pictures of the same person should give roughly the same measurements.

The advantage of using a pretrained neural network for face recognition is that it can leverage the knowledge gained from a large dataset and capture intricate patterns and features that are difficult for traditional algorithms to discern. The 128-dimensional face embeddings produced by the neural network provide a rich and compact representation of the face.

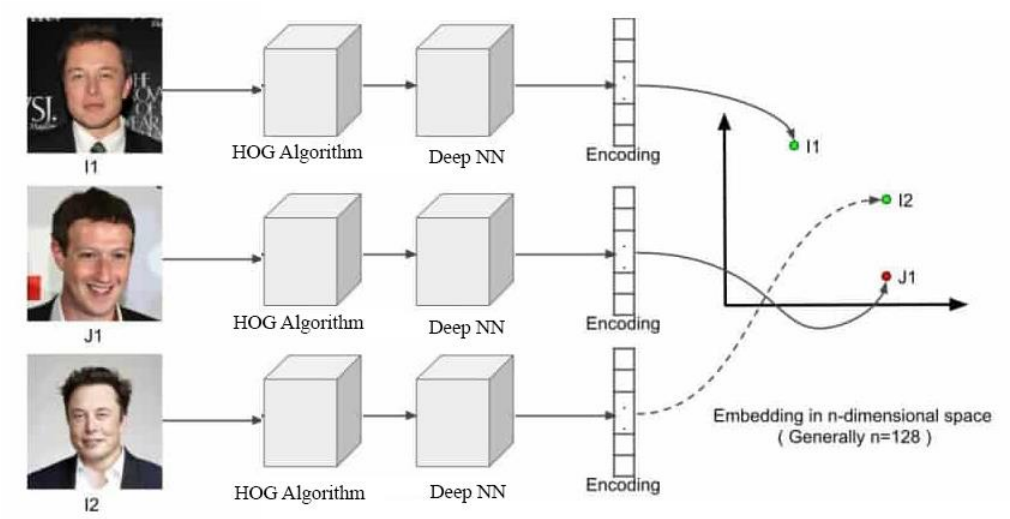

Figure 11**:** Generation of 128-dimensional face encodings

Phase 3: Face Recognition

We use Euclidean distance measures how far away two items are. In order to apply this to a pattern recognition task, we will need the feature vectors (i.e. lists of numerical values) or face encodings. We do a pairwise comparison to work out how distant two faces are. Here for each pair, one of the image is the one being used to check if the given person is present in the system or not and the other is a variable which represents the encodings of faces of all the people present in the database one at a time. We have decided to set a threshold of 0.5, where we treat two images as being the same face when they have a distance which is less than the threshold (typically where the distance between the feature vectors is small).

Once a suitable match has been found with a distance less than the threshold we declare the person as being present in the database. Based on the Euclidean distance value we can determine the name of the person and display appropriate results. If no suitable match has been found, then we declare the person is not present in the database. Control is re-directed to provide an option to register the person in the database in that case.

4.11. Landing Page Sections

Awareness page provides videos on how people deal with danger and protect themselves and others. **About Us section** provides a brief description about 'DOT 1.0', its goals and methods to achieve the goals. **Services section** gives an idea about all the services of the website. **Feedback Form section** provides a form. Users can give any kind of feedback related to this website through this form. The options are as follows: **Name, Email, Phone Number**. Next, a field is provided for the user to type feedback message in not more than 300 words. The form is connected to the backend.

5. IMPLEMENTATION

ER Diagram of The Entire System

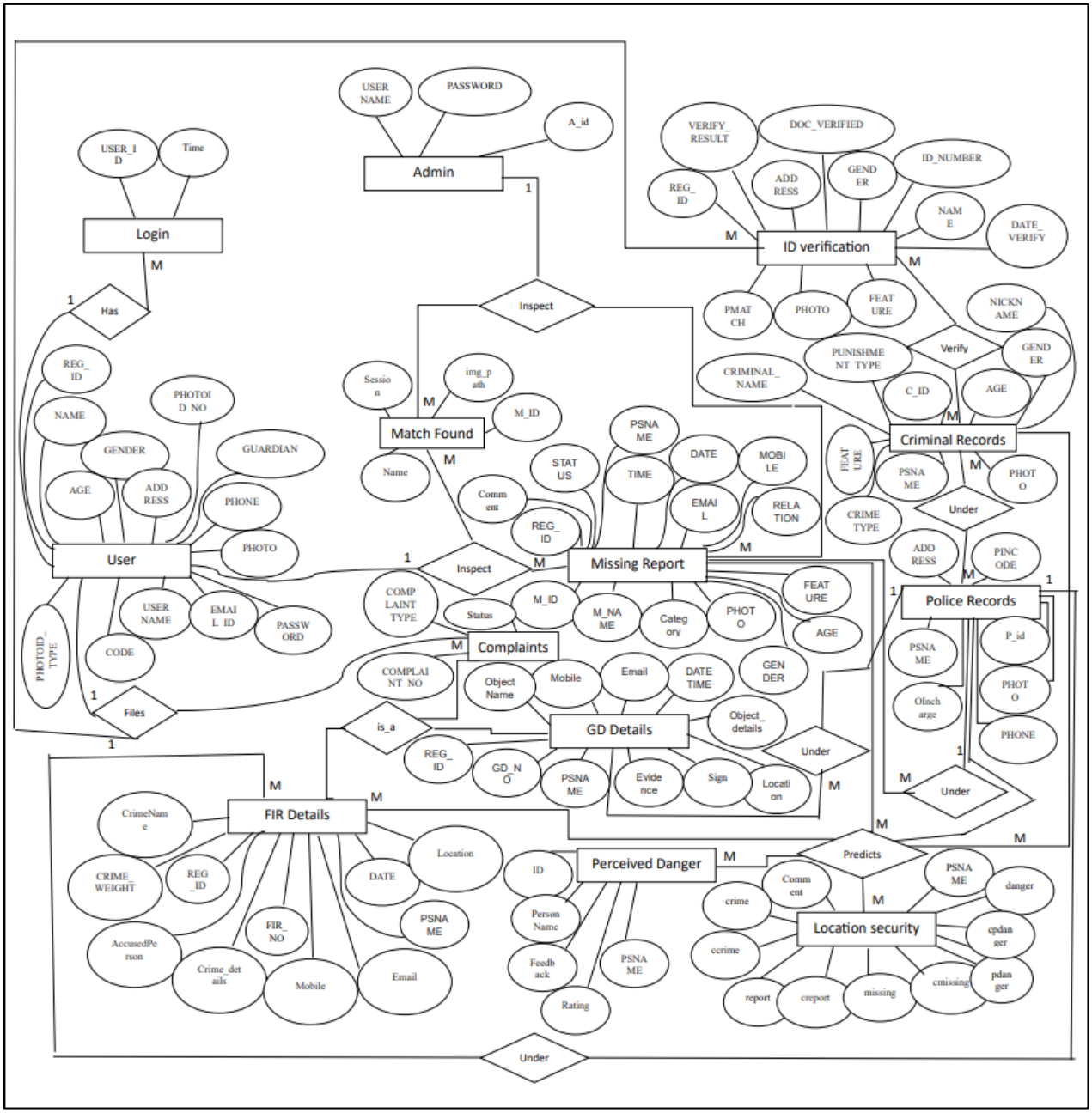

Figure 12**:** ER Diagram of the system

5.1. Creating the database

We have made **15** tables in our database named *'POLICE'*. The database has been created using ***XAMPP* server** and the tables are created using ***phpMyAdmin*.**

Our system performs various functions on the basis of which the database can be classified into 6 subsections –

- **Lodge complaints-**
  This section is further divided into 3 parts: i) File FIR, ii) File GD**,** iii) File Missing report.

In FIR, we file a complaint regarding any crime or any accident. GD report deals with filing reports regarding missing objects, and Missing report deals with filing reports about missing persons. So here in this section of our database there are four tables. These are –

a) fir_details
b) gd_details
c) missing_report
d) complaint_status

The attributes of these tables are –

- fir_details-
  1) REG_ID (Registration id of the user is stored)
  2) FIR_NO (A unique number given to each Fir report is stored)
  3) CrimeName(Name of the crime or incident is stored)
  4) CRIME_WEIGHT(Weight allotted to each type of crime is stored)
  5) AccusedPerson(If FIR is lodged against a person then the name of that person is stored)
  6) Crime_details(Details of the crime or incident is stored)
  7) Mobile(Mobile no of the user is stored)
  8) Email(Email id of the user is stored)
  9) Location(Location of the crime is stored)
  10) DATE(Date of occurence of the crime or incident is stored)
  11) PSNAME(Name of nearest police station is stored)
- gd_details-
  1) REG_ID (Registration id of the user is stored)
  2) GD_NO (A unique number given to each GD report is stored)
  3) ObjectName(Name of the missing or lost object is stored)
  4) Object_details(Details of the missing or lost object is stored)
  5) Mobile(Mobile no of the user is stored)
  6) Email(Email id of the user is stored)
  7) Location(Last location of the missing object is stored)
  8) DATETIME(Date and Time when the object was lost is stored)
  9) PSNAME(Name of nearest police station is stored)
  10) Evidence(Photo of the missing object if present is stored)
  11) Sign(Photo of signature of the user is stored)
- missing_report (this table is used to store both missing person details and found person details)
  1) REG_ID (Registration id of the user is stored)
  2) M_ID (A unique number given to each missing report is stored)
  3) M_NAME(Name of the missing/found person is stored)
  4) Category(Stores whether it is lost person or found person)
  5) PHOTO(Photo of the missing/found person is stored)
  6) GENDER(Gender of the missing/found person is stored)
  7) AGE(Age of the missing/found person is stored)
  8) FEATURE(Feature of the missing/found person is stored)
  9) RELATION(Relationship of the lost person with applicant is stored)
  10) EMAIL(Email id of the user is stored)
  11) MOBILE(Mobile no of the user is stored)

12) DATE (Date when the person went missing is stored)
13) TIME(Time when the person went missing is stored)
14) PSNAME(Name of nearest police station is stored)
15) LOCATION(Last location of the missing person is stored)
16) STATUS(Status of the case is stored)

- complaint_status
  1) COMPLAINT_NO(Stores the FIR No or GD No)
  2) COMPLAINT_TYPE(Stores the name of the table in which the complaint is stored)
  3) STATUS(Status of the case is stored)

Here, on clicking Lodge FIR, the user is redirected to a form which requires the user to fill all the details about the crime or incident that has occurred. On submission of the form the user is redirected to FIR_SUBMITTED-1.php . Through the use of session variables, we have stored the registration id of the user. Now we extract each of the details filled in the form in different variables by using: - $_POST['name']

Here we also, categorised each type of crime and given a particular weight to crime belonging to each category. Then we insert the data into the complaint_status and fir_details table.

The **Lodge GD**, does similar work as the Lodge FIR upto submission of the form. It deals with the missing objects. After submission all the details of the GD report is stored in the gd_details and complaint_status(which shows status of the filed complaint) table and is redirected to a page which shows the GD no and a button to download the GD (as the copy of GD is required by the user for many purposes).

On clicking the Download GD button the user is redirected to a pdf which contains a GD letter having all the details. The details are extracted by using session variables and from the gd_details table. The pdf is generated by using 'mpdf'.

In the **Missing Diary** page , the user is required to fill the form and provide multiple photos of the missing person for better recognition. These photos are stored in the folder – "Missing_images" And these details are stored in the missing_report table.

There is a page found report similar to missing report ,the only difference here is that here the found report is filed by the admin when the police founds a person and stores the image in the 'faces' folder and the details are inserted in missing report table. To differentiate between found report and missing report in the same table the attribute 'Category' is used.

We have also used the tables mentioned here to show the complaints filed by the user and the total no. of reports filed through our system. The admin also can view all the reports filed and can update their status.When admin change the status of a missing report to matchfound then the images stored for the corresponding missing report is moved to another folder named MATCHED_PIC.

- **Face Recognition**–

This is further divided into 3 sections : i) Admin End ii) User End iii) Tenant verification

Admin End -This part of our system aims to recognize a found person's (by police) face by comparing it with the images of person who has been reported by the users and is available

in our database i.e in 'Missing_images' folder. When no match is found the admin is redirected to lodge a Found Report.

User End -This part of our system aims to recognize a missing person's face by comparing it with the images of person who has been found by the police and is available in our database i.e in the 'faces' folder. When a match is found the user needs to click Send Request button , which makes a request which can be viwed by the admin. And as per the request , a mail containing details of the person with whom match is found will be sent. When no match is found the user is redirected to lodge a Missing Report.

Tenant verification- This part of our system aims to verify if a tenant's face matches with any criminal by comparing it with the images of criminals which is available in our database i.e in the 'cfaces' folder. For this the user needs to fill up the form with necessary details including the tenant's photoid no and photo. On submission a request is sent to the admin for reviewing the tenant's details and then admin sends a mail to the user which contains information about the tenant whether he/she has a past criminal record or not. All the mails are sent using phpmailer.
So here in this section of our database there are three tables. These are –

a) missing_details
b) matchfound
c) id_verification

The attributes of these tables are –

- missing_details
  1) NAME (Name of the missing person is stored)
  2) IMAGE(Location of image of missing person is stored)
- matchfound
  1) Name (Name of the missing person is stored)
  2) M_ID (ID of the person with whom missing person is matched is stored)
  3) img_path(The image which is searched is stored)
  4) Session(A unique number assigned to each matchfound entry)
- id_verification
  1) REG_ID (Registration id of the user is stored)
  2) PMATCH (Match % of the tenant with criminal is stored)
  3) DOC_VERIFIED (Document type of the tenant used for verification is stored)
  4) ID_NUMBER (Photo id number of the tenant used for verification is stored)
  5) ADDRESS (Address of the tenant is stored)
  6) GENDER (Gender of the tenant is stored)
  7) NAME (Name of the tenant is stored)
  8) FEATURE (Physical features of the tenant is stored)
  9) PHOTO (Photo of the tenant is stored)
  10) VERIFY_RESULT(Result of verification is stored)
  11) DATE_VERIFY(Date of verification is stored)

- **FUZZY –** This section of system shows the danger level of the areas under each police station. So here in this section of our database there are five tables. These are –
  a) criminal_record

b) fir_details (mentioned earlier)
c) missing_report (mentioned earlier)
d) perceived_danger
e) fuzzy

The attributes of these tables are –

- criminal_record
  1) C_ID (Unique id given to each criminal is stored)
  2) CRIMINAL_NAME(Name of the criminal is stored)
  3) NICKNAME(Nickname of the criminal is stored)
  4) GENDER (Gender of the criminal is stored)
  5) AGE(Age of the criminal is stored)
  6) PSNAME (Police station under which the criminal is convicted is stored)
  7) PHOTO (Photo of the criminal if present is stored)
  8) FEATURE (Physical features of the criminal is stored)
  9) CRIME_TYPE(Type of crime committed by the criminal is stored)
  10) PUNISHMENT_TYPE (Tenure of punishment is stored)
- perceived_danger
  1) ID (Unique id for each person who gave feedback is stored)
  2) PersonName (Name of the person who gave the feedback is stored)
  3) Feedback (Feedback is stored)
  4) Rating (Rating of the area on the basis of danger is stored)
  5) PSNAME (Police station name is stored)
- fuzzy
  1) comment (the fuzzy output set for danger level is stored)
  2) crime(crisp value of crime is stored)
  3) ccrime(the fuzzy set selected with respect to crisp value of crime is stored)
  4) report (crisp value of reports is stored)
  5) creport (the fuzzy set selected with respect to crisp value of reports is stored)
  6) missing (crisp value of missing report is stored)
  7) cmissing (the fuzzy set selected with respect to crisp value of missing reports is stored)
  8) pdanger (crisp value of percieved danger is stored)
  9) cdanger (the fuzzy set selected with respect to crisp value of percieved danger is stored)
  10) danger (percentage of danger of area under each police station is stored)
  11) PSNAME(Police station name is stored)

- **POLICE RECORDS**

This section allows the user to view the details of each police stations within Kolkata. And the admin can update the Officer incharge name, contact no and the photo of the officer incharge for every particular police station in our database.So here in this section of our database there is one table –

police_record

1) P_id(Unique id given to each police station)
2) PSNAME (Police station name is stored)

3) OIncharge (Name of Officer Incharge is stored)
4) PHOTO (Photo of the Officer Incharge is stored if present)
5) PHONE (Contact no of the police station is stored)
6) ADDRESS (Address of the police station is stored)
7) PINCODE (Pincode of the police station is stored)

- **USER REGISTRATION AND LOGIN**

In this section of our database there are three tables-

a) user_registration
b) admin_details
c) user_login

The attributes of these tables are –

- user_registration
  1) REG_ID (Registration id of the user is stored)
  2) NAME (Name of the user is stored)
  3) GUARDIAN (Name of guardian of the user)
  4) GENDER (Gender of the user is stored)
  5) AGE(Age of the user is stored)
  6) ADDRESS (Address of the user is stored)
  7) PHONE (Contact no of the user is stored)
  8) PHOTOID_TYPE (Photoid type of the user is stored)
  9) PHOTO (Photo of the user is stored)
  10) PHOTOID_NO (Photoid no of the user is stored)
  11) EMAIL_ID (Email id of the user is stored)
  12) USERNAME (Username of the user is stored)
  13) PASSWORD (Password of the user is stored)
  14) CODE (OTP of the user is stored)
- admin_details
  1) A_id(Stores id of the admin)
  2) USERNAME (Username of the admin is stored)
  3) PASSWORD (Password of the admin is stored)
- user_login
  1) USER_ID (Stores registration id of the user)
  2) Time (Stores time of login of the user)

The user/admin enters the user/admin dashboard page by logging in with his/her registered username and password through this sql query:

(Admin Section)

"SELECT PASSWORD, A_ID FROM ADMIN_DETAILS WHERE BINARY USERNAME='$USERNAME';"

BINARY is used to differentiate upper- and lower-case characters.

In the user dashboard, we have used the user_login table and shown the total no of daily logins by using the query:

"SELECT COUNT(DISTINCT USER_ID) as Count FROM user_login WHERE Time > NOW() - INTERVAL 1 DAY ;"

User Profile page shows the complete information of the user and further provide two options – **Show Complaints** and **Update phone number**

Update phone number- In this page the user will input his new or changed phone number and after submitting the page the phone number will be updated in the user's profile.

To update phone number, we have used the sql query:

"UPDATE USER_REGISTRATION SET PHONE = '$MOBILE' WHERE USERNAME = ' ". $USERNAME. " ';"

Using the details from , user_registration table ,the admin can view all the users of our website and can delete the profile of suspicious users.

- **FEEDBACK-**We made a feedback table to store the suggestions from the end user so that we can improve the system according to the need of users.

  - **feedback**
    The attributes of this table are –
    1) Name (Name of the person)
    2) Phone (Contact No of the person)
    3) Email (Email of the person)
    4) Message (Feedback given by the person)

  In the user dashboard, we have shown the total no of feedbacks by using the query:

  SELECT COUNT(*) as COUNT FROM feedback.

PHPMailer-

Mail has been used at various places on this website for user registration or to send any information to the user.

Mailing process is done using 'phpmailer'. Here a host's email is given through which mail is sent to the user. The only PHP function that supports this directly is mail(). With IsSMTP(), it uses PHPMailer's accompanying SMTP class to act as an SMTP client here user's, talking to a mail server or host directly.

To check the user's verification code on the verify page, we have used the sql query:

if ($code === $otp)

Where the 'code' is fetched from the user registration page with the session variable and 'otp' is the input of the code given by the user through mysqli_real_escape_string() function.

5.2: Frontend Implementation

5.2.1: Creation of GUI

UI design is required to increase the usability of the features provided by the website that are aesthetically appealing, and to optimize for different screen sizes. The GUI of DOT 1.0 has been designed keeping in mind that the website aims to comfort those who came for help after any kind of danger. Dealing with such users requires empathic, kind, and comfortable communication. Hence, the challenge was to achieve that friendliness through the frontend only, with which the user actually interacts.

Colour Psychology and Choice of Colours

The purpose of our website is to show people the direction of light from darkness, so at the beginning of this website dark red and black colours are used and as people enter the website and face the truth, the colours of the website become lighter yellow and red. And finally, when people get the solution, it is revealed in white.

Font

The fonts were so chosen that do not appear formal, But mysterious. Those which are compatible with the crime, such as 'Baskerville Old Face' (for heading), 'Bell MT', etc. were selected.

Background Images

Here the background images are basically different gradients. The user enters with a lot of doubt and frustration, so the bright light can convey a different idea about our website in the user's mind. The bright colour of the page to solve one doubt gives courage to the user and at the end white or light-yellow background is used to make the user completely at ease mentally to solve any problem.

5.2.2: Frontend Languages

Three fundamental languages, HTML, CSS, and Javascript, were used to create the front-end of the website. HTML is the basic technology to build any webpage.CSS has been thoroughly used in our project to design the elements and create interactive animations. Javascript has been utilized for generating dynamic webpages and creating alert box pop ups.

5.2.3: Neumorphism Implementation

The entire website has been developed purely based on Neumorphism, which is a UI design trend that imitates the look of frosted glass. It is a multi-layered approach with objects floating in space creating a semi-transparent background with a sublime shadow and border.

5.3: Implementation of AI Modules

All the modules related to artificial intelligence have been created using Python programming language.

Face Recognition

We have made the use of the following library functions packages for implementation purpose:

1. **Numpy**- It is used for performing mathematical and logical operations on arrays.
2. **OpenCV-** It is a huge open-source library for computer vision, machine learning, and image processing.
3. **face_recognition-** It is used to recognize and manipulate faces.
4. **OS-** It provides a portable way of using operating system dependent functionality like reading or writing to the file system, running shell commands, manipulating environment variables, and more.
5. **MySQL connector-** It is used to enable Python programs to access MySQL databases

Implementation of Face Recognition Algorithm

Start

Step 1: The code begins with importing necessary libraries. Then it initializes variables such as 'img_path', 'path', 'images', 'classNames', 'myList'.

The 'img_path' and 'path' variables store the path of the image and directory containing images, respectively. 'images' and 'classNames' are empty lists to store the images and their class names. The 'myList' variable stores the list of all the files present in the directory.

Step 2: The code defines several functions including those to find data, find encodings and to modify tables.

Step 2.1: The function to find data establishes a connection to the MySQL database, executes a SQL query to fetch data from table and returns the necessary variables. This function is used to extract information on the path of the image with which we compare all the images in the database.

Step 2.2: The function to find encodings takes images as input, converts them to RGB format using OpenCV, encodes them using the face_recognition library, and returns the encoded list of images.

Step 2.3: The function to modify table establishes a connection to the MySQL database, executes a SQL query to insert, modify or delete the data to display appropriate result.

Step 3: After defining the functions, the code fetches path of the image which is to be compared with all other images in the database from the table and stores it in 'img_path' variables. Then, it calls the 'findEncodings' function to encode all the images present in the 'images' list and store them in the 'encodeListKnown' variable.

Step 4: Read the image from the 'img_path' variable using OpenCV, convert it to RGB format, and detect faces in the image using the face_recognition library. The detected faces are then encoded and stored in the 'encodesCurFrame' variable.

Step 5: The code then uses a loop to compare the face encodings of the detected faces with the encodings of known faces stored in the 'encodeListKnown' variable. Since we have the

encodings for both faces, then we can compare these 128 measurements of these two faces to find similarities. To compare the package uses one of the most common machine Learning methods linear SVM classifier. We can use the compare_faces function to find if the faces match. This function returns True or False. Similarly, we can use the face_distance function to find how likely is the faces match in terms of numbers.

Step 6: To check if a match is found.

Step 6.1: If a match is found then,

It determines the class name with the highest similarity score and the code then calls necessary functions to insert or modify the data in the table of the MySQL database.

The code then prints the output which is a string containing the name of the best match, and the accuracy percentage if necessary.

Step 6.2: Else,

If no match is found, we call necessary functions to insert or modify the data in the table of the MySQL database.

If no suitable match is found, the code prints the string "No suitable match found.".

Stop

In our project we have used the face recognition algorithm in 2 modules.

1. We have used it while filing a missing complaint to check if the person we are filing the complaint for is already present in the system so that necessary actions can be taken. This can be done in both the user end (explained in page 18) as well as the admin end (explained in 19).
2. We have used it for identity verification. Here we check if the given image matches with that of any registered offender depending on which necessary actions can be taken. (Explained in page 19).

Fuzzy Inference System

We have made the use of the following library functions packages for implementation purpose:

1. **Numpy-** For performing mathematical and logical operations on arrays.
2. **Mysql.connector**- To enable Python programs to access MySQL databases
3. **Skfuzzy**-For implementing machine learning and Fuzzy Inference System

Implementation of Fuzzy Inference System

Start

Step 1: Import the necessary libraries: 'numpy', 'Mysql.connector' and 'skfuzzy'.

Step 2: Define the universe variables for each input and output variable. These variables represent the range of values for each variable.

Step 3: Generate fuzzy membership functions for each variable. Membership functions define the degree of membership of a value to a fuzzy set.

Step 4: Call necessary functions to return values to be given as input to the variables: ‘pdanger’, ‘missing’, ‘crime’, and ‘report’.

Step 5: Define the weights for each input variable. Weights determine the importance of each variable in the fuzzy inference process.

Step 6: Calculate the membership values for each input variable using the ‘interp_membership’ function. This function determines the degree of membership of a value to a fuzzy set based on the membership function.

Step 7: Find the set with the highest membership value for each input variable.

Step 8: Update the fuzzy sets with the highest membership values for each input variable, indicating the linguistic terms (e.g., low, medium, high) associated with each variable into the table.

Step 9: Apply fuzzy rules to determine the activation levels of the output variable, ‘danger’.

Step 10: Define the fuzzy rules using the ‘np.fmin’, ‘np.fmin’ and ‘np.logical_not’ functions to combine the membership values of the input variables based on the logical AND, OR and NOT operations respectively.

Step 11: Calculate the activation level of each rule by applying the fuzzy rule to the corresponding input variables.

Step 12: Aggregate the activation levels of each rule to determine the overall activation level of the output variable for each fuzzy set.

Step 13: Defuzzify the output variable, ‘danger’, to obtain a crisp value using the ‘defuzz’ function.

Step 14: Update the defuzzified value of ‘danger’ along with its linguistic term (e.g., very low, low, medium, high, very high) based on the highest activation level in the table with respect to each police station.

Stop

The algorithm follows a typical fuzzy inference process, including fuzzification (calculating membership values), rule evaluation (applying fuzzy rules), and defuzzification (obtaining a crisp output value).

6. RESULTS AND DISCUSSIONS

Figure 13: Screenshot of Landing Page

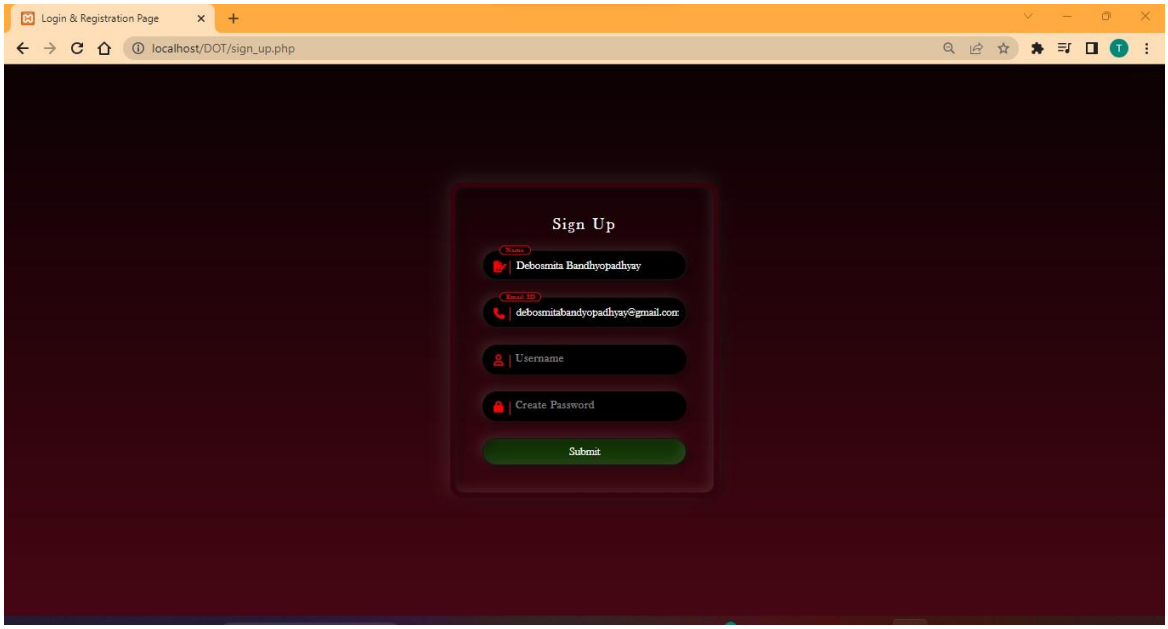

Figure 14: Sign Up Page

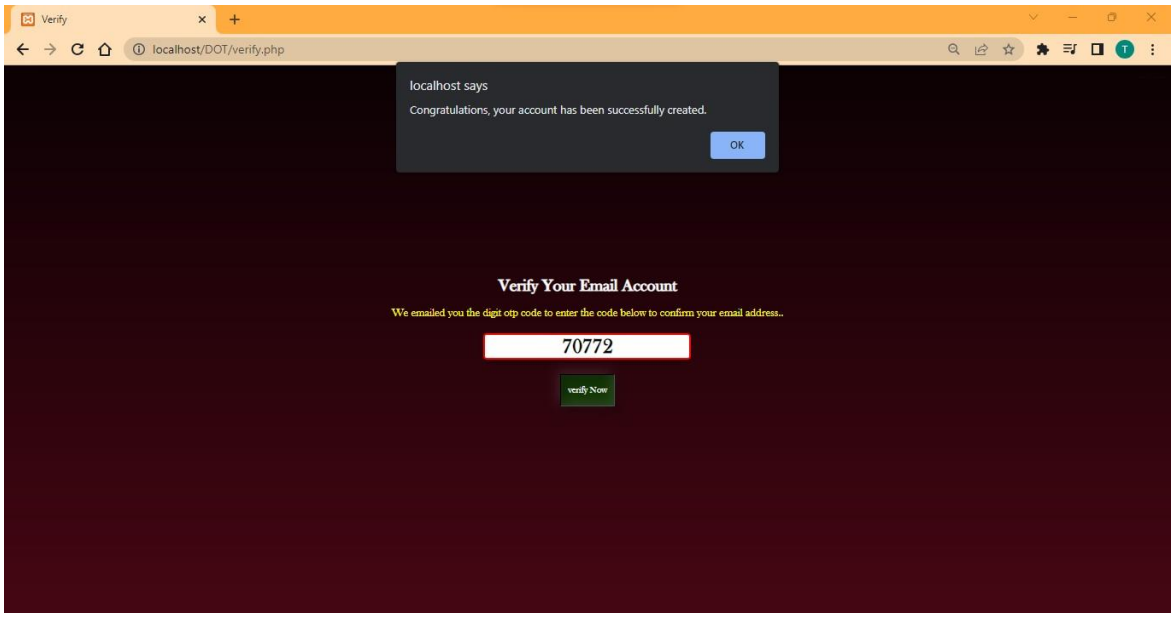

Figure 15: Screenshot of Email Verification Page

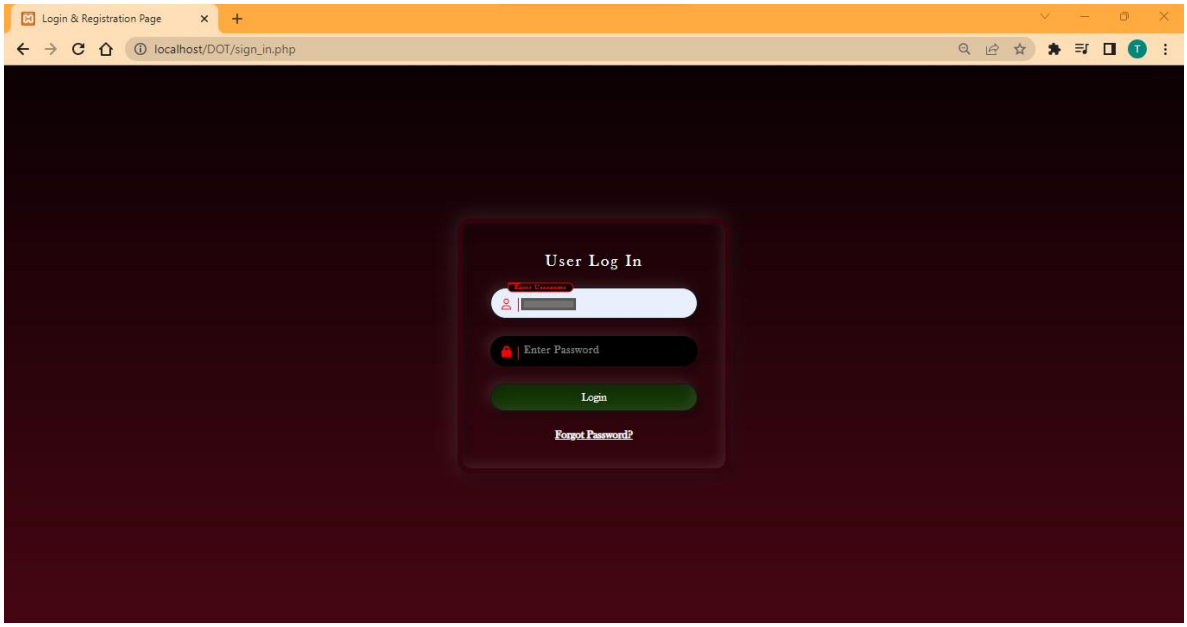

Figure 16: Screenshot of User Login Page

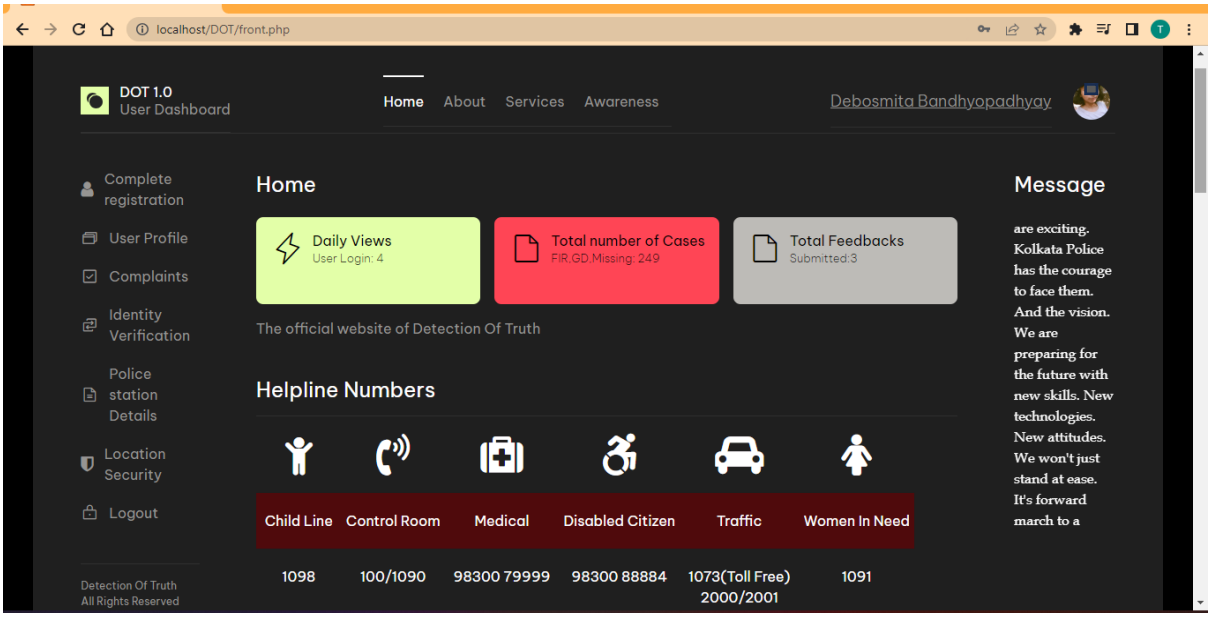

Figure 17: Screenshot of User Dashboard Page

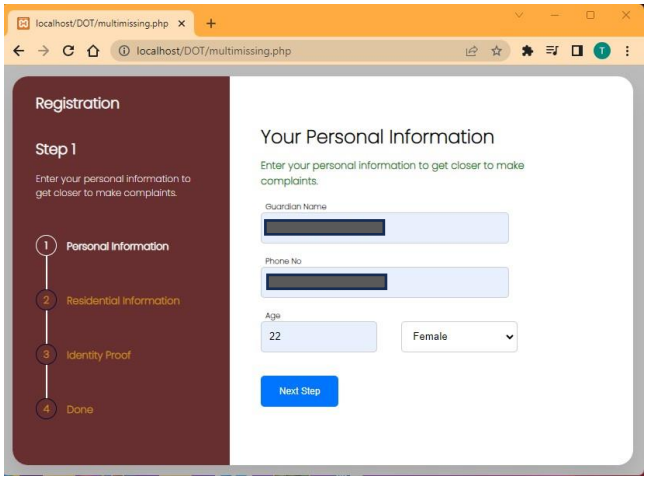

Figure 18: Screenshot of Complete Registration Page (1)

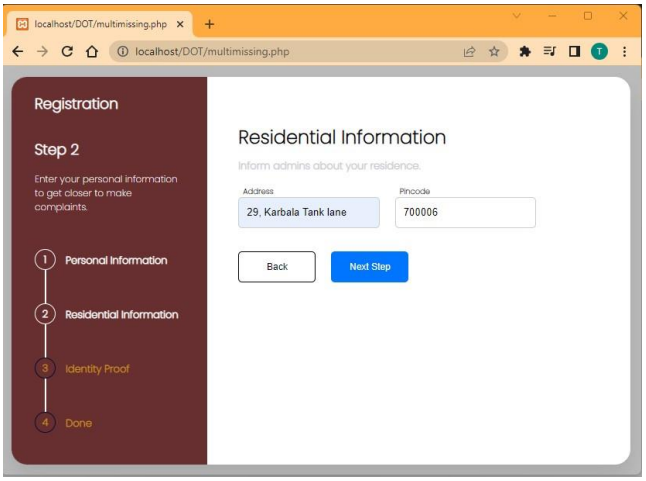

Figure 19: Complete Registration Page (2)

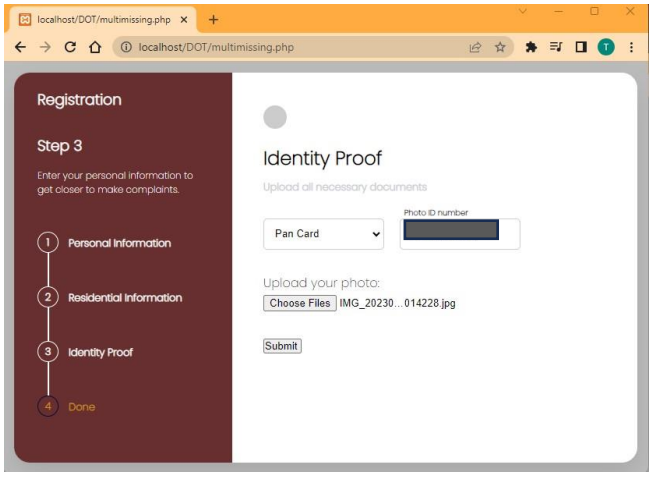

Figure 20: Complete Registration Page (3)

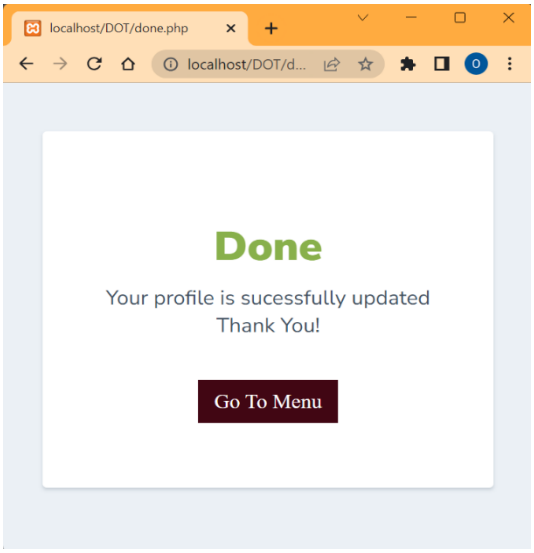

Figure 21: Screenshot of Update Successful Page

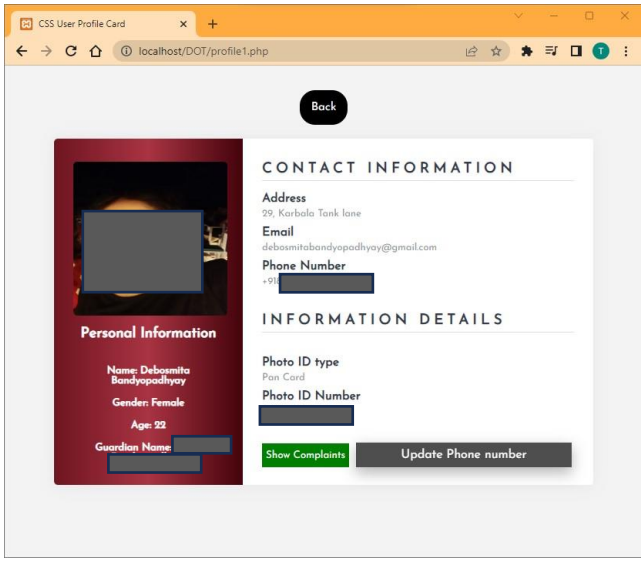

Figure 22: Screenshot of User Profile Page

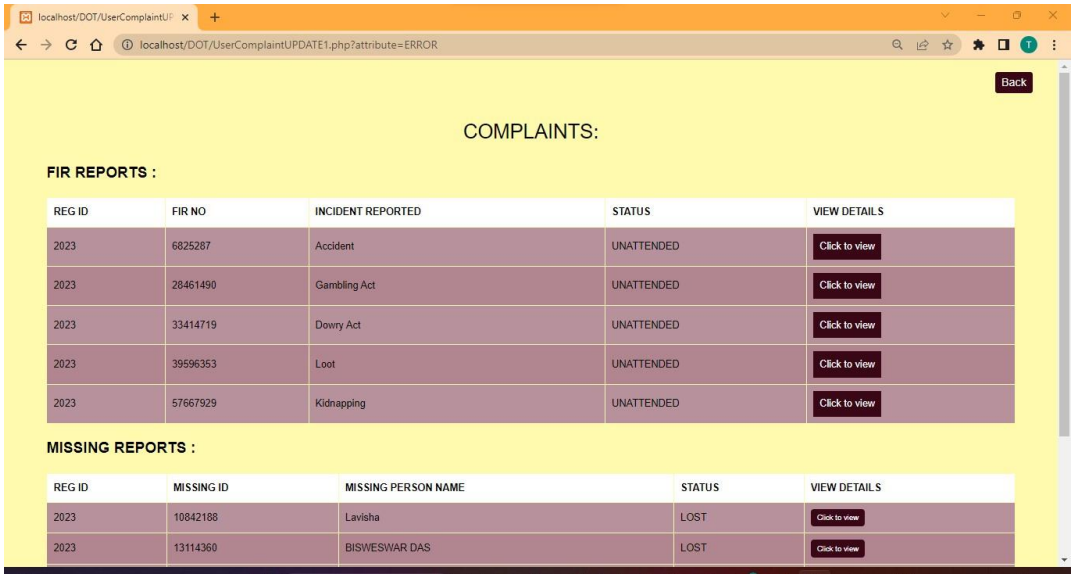

Figure 23: Screenshot of User Complaints Page

Figure 24: Screenshot of View Complaints Page

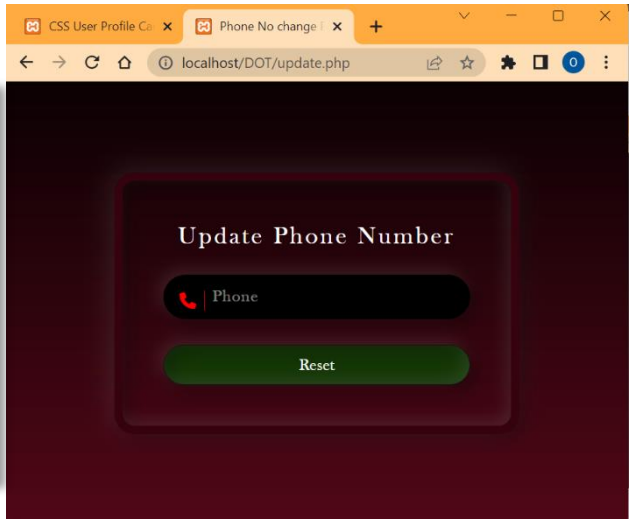

Figure 25: Screenshot of Phone Number Update Page

Figure 26: Screenshot of Complaints (menu) Page

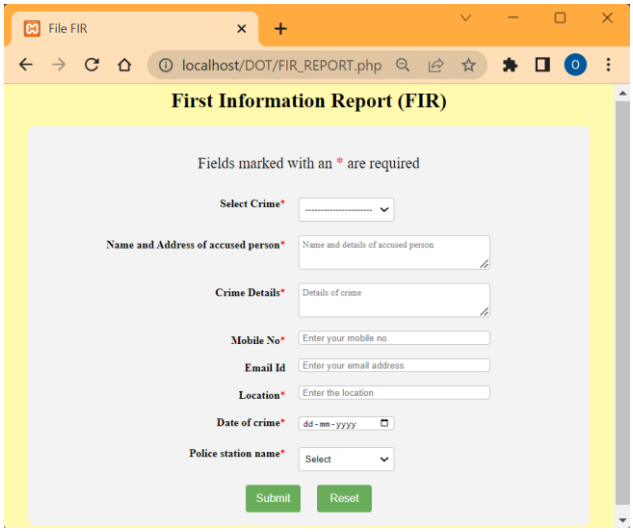

Figure 27: Screenshot of FIR Report Page

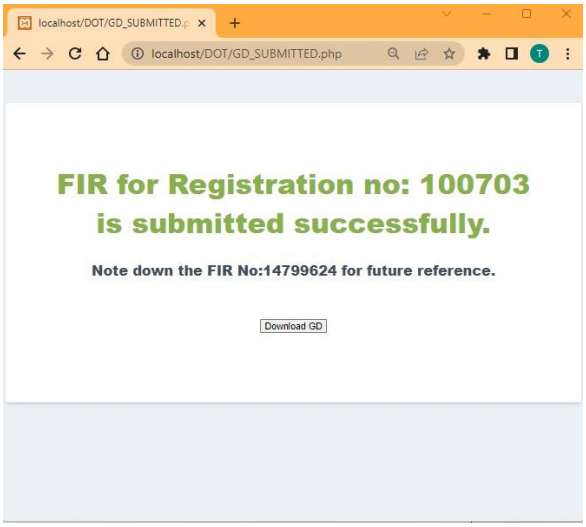

Figure 28: Screenshot of FIR Report Submission Page

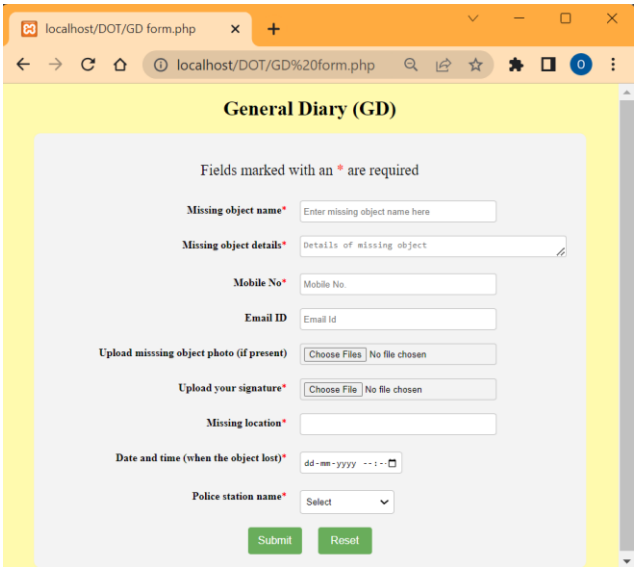

Figure 29: Screenshot of GD Report Page

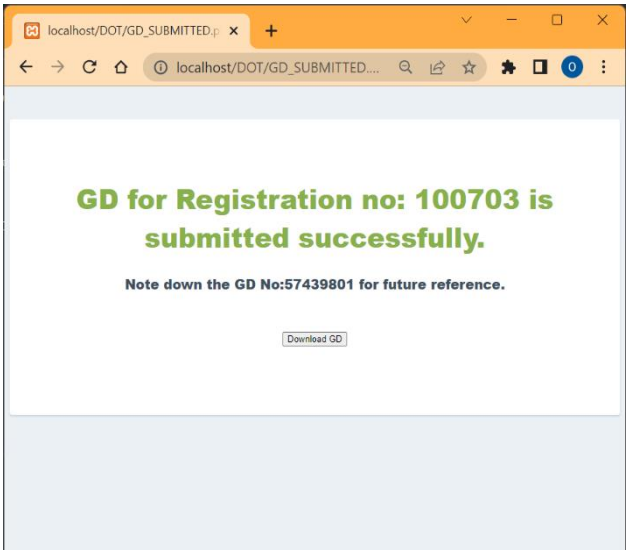

Figure 30: Screenshot of GD Report Submission Page

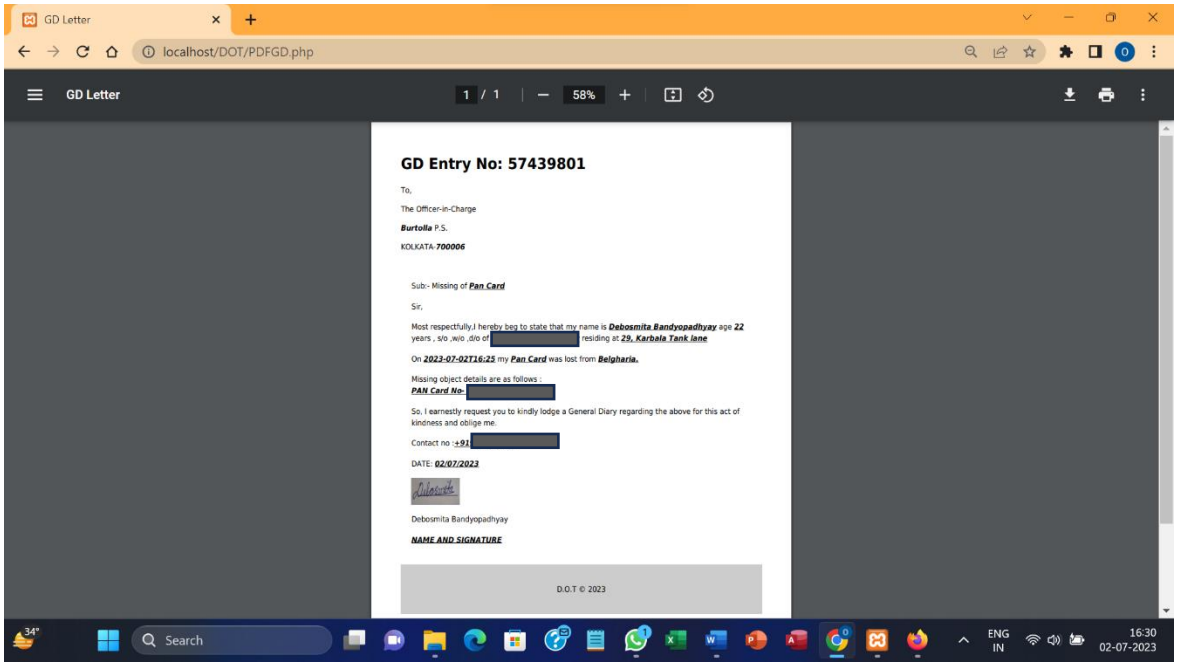

Figure 31: Screenshot of Download GD Report Page

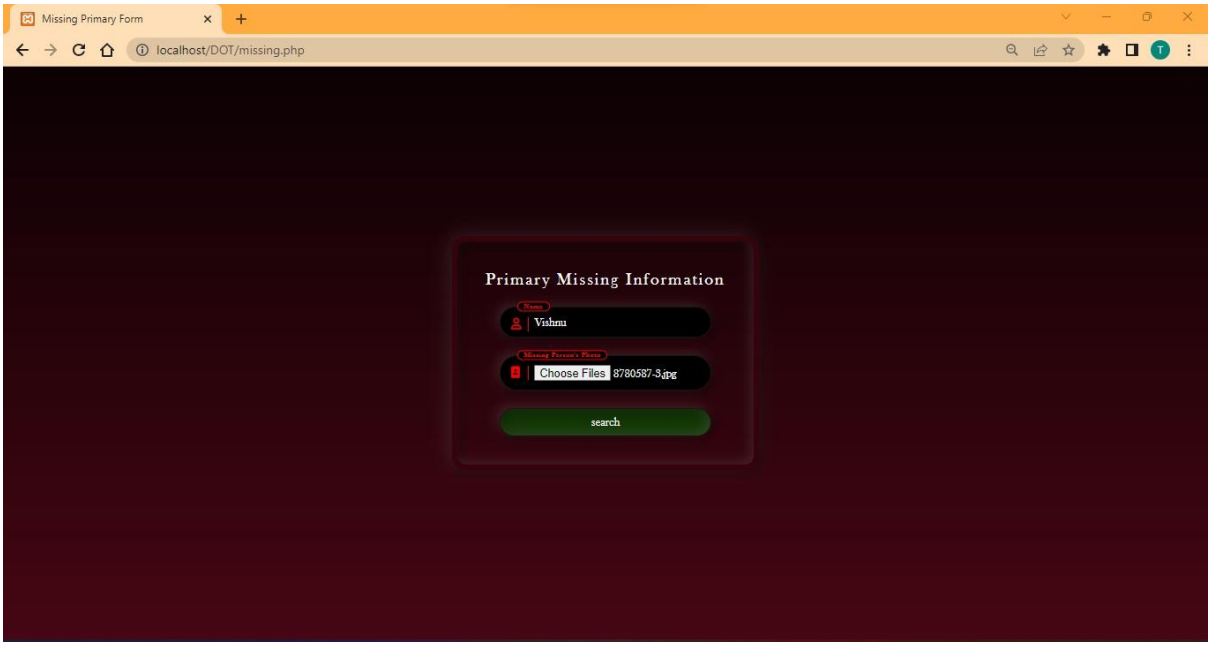

Figure 32: Screenshot of Primary Missing Report Page

Figure 33: Screenshot of Face Recognition output Page (Match Found)

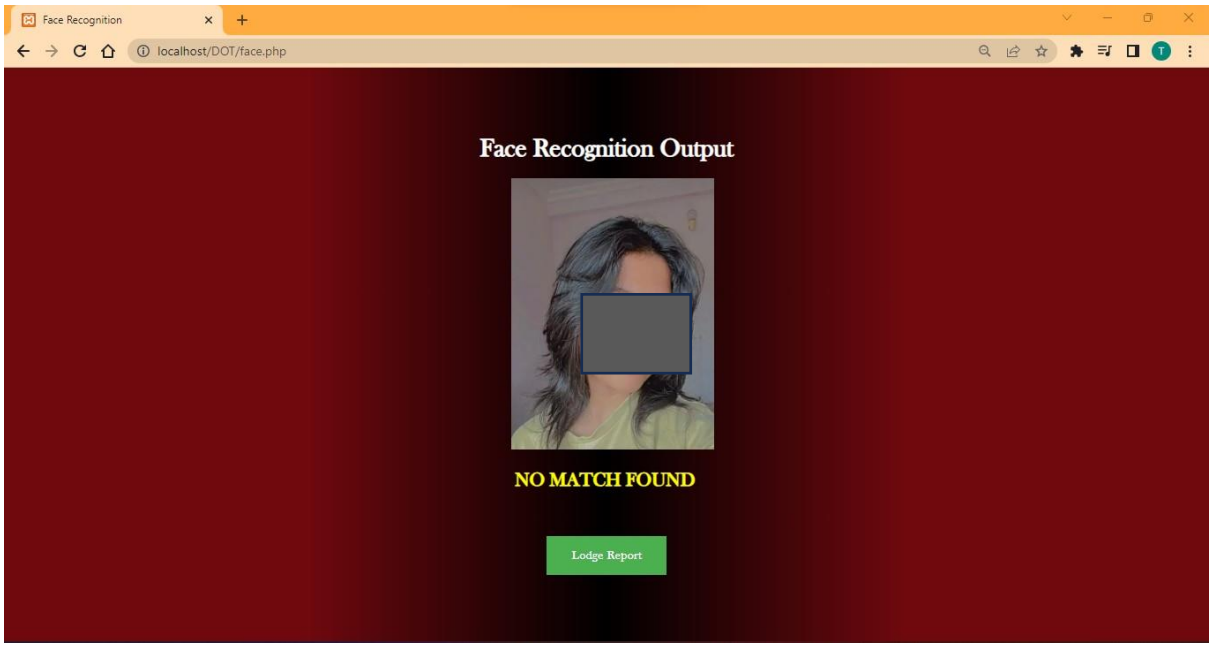

Figure 34: Screenshot of Face Recognition output Page (No Match Found)

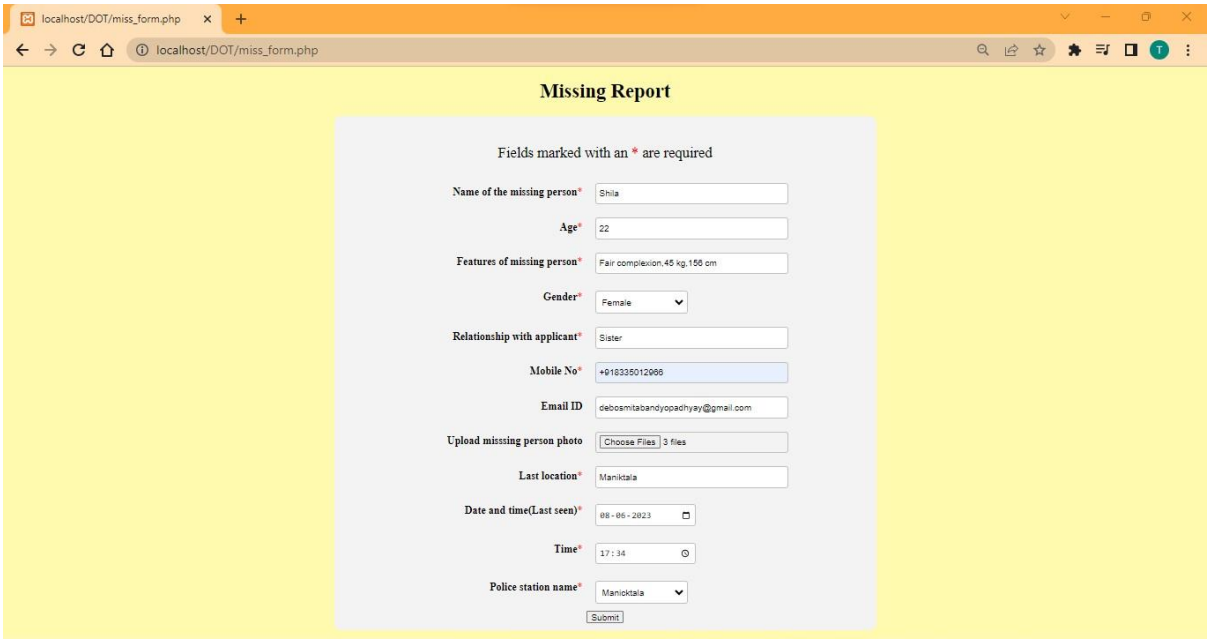

Figure 35: Screenshot of Missing report Page if no match found

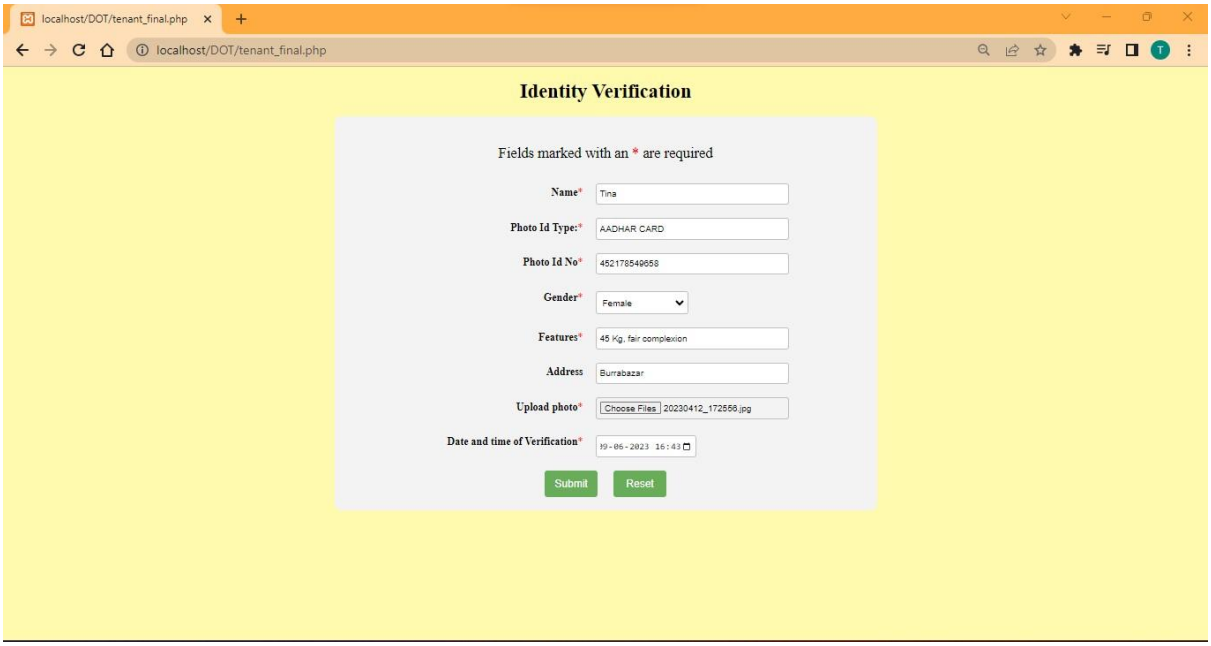

Figure 36: Screenshot of Identity Verification Page

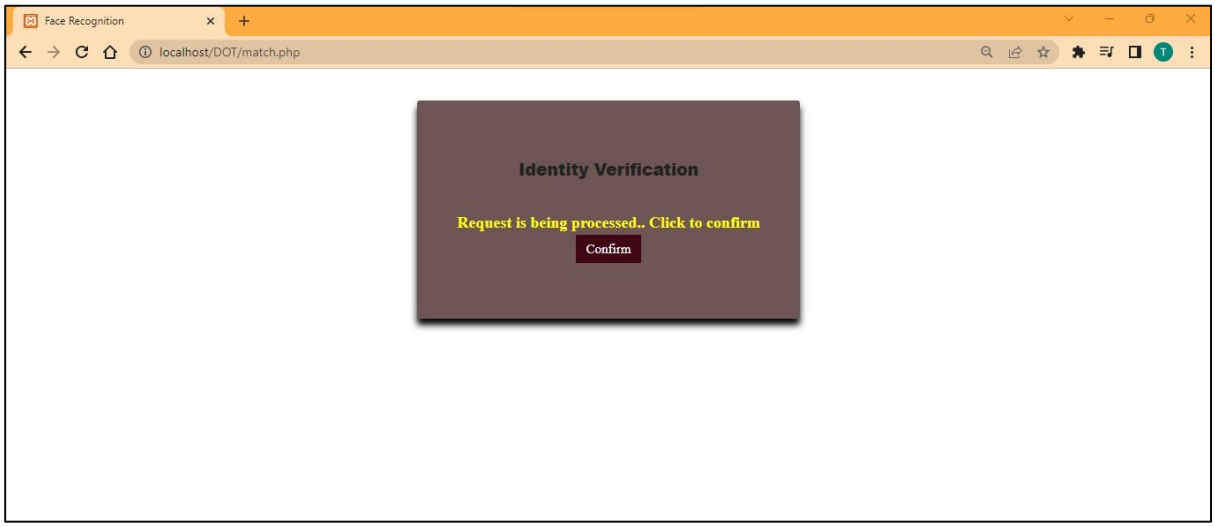

Figure 37: Screenshot of ID verification Confirmation Page

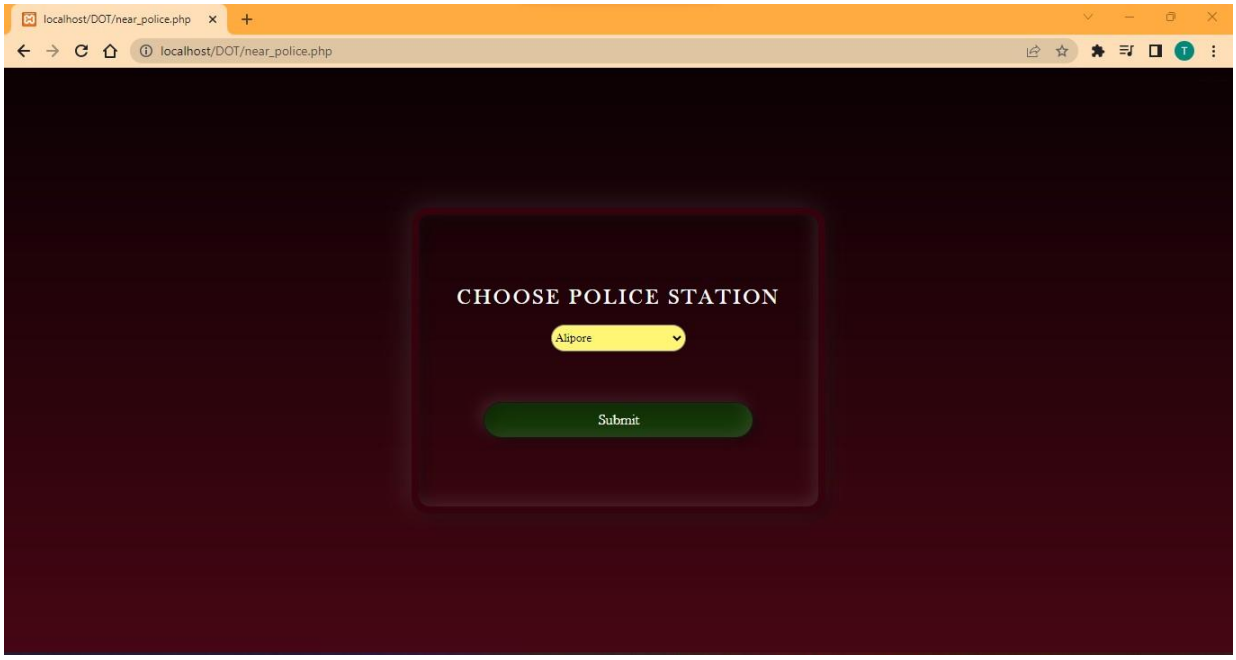

Figure 38: Screenshot of Nearest Police Station Search Page

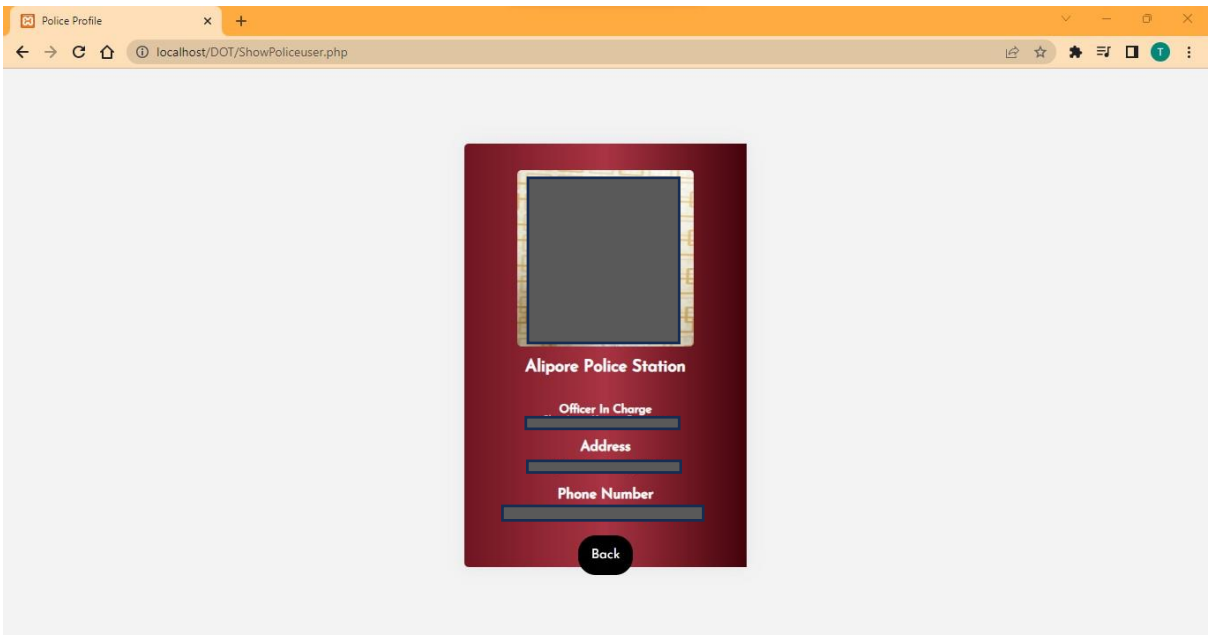

Figure 39: Screenshot of Show Police Station Page

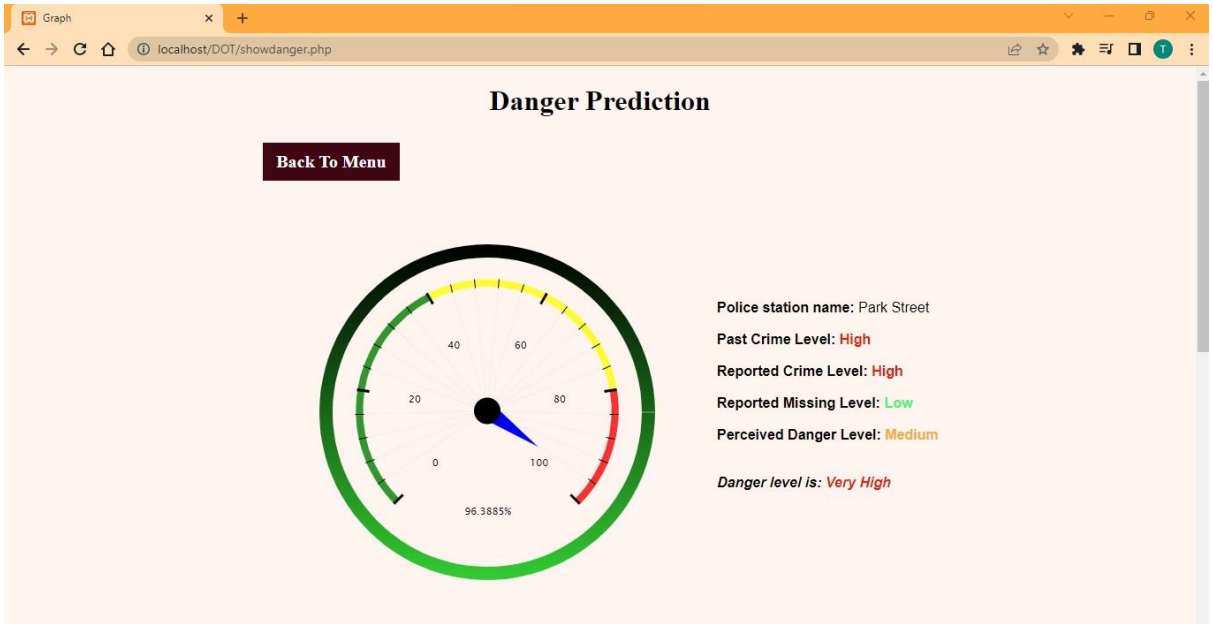

Figure 40: Screenshot of Danger Prediction (specific area) Page (1)

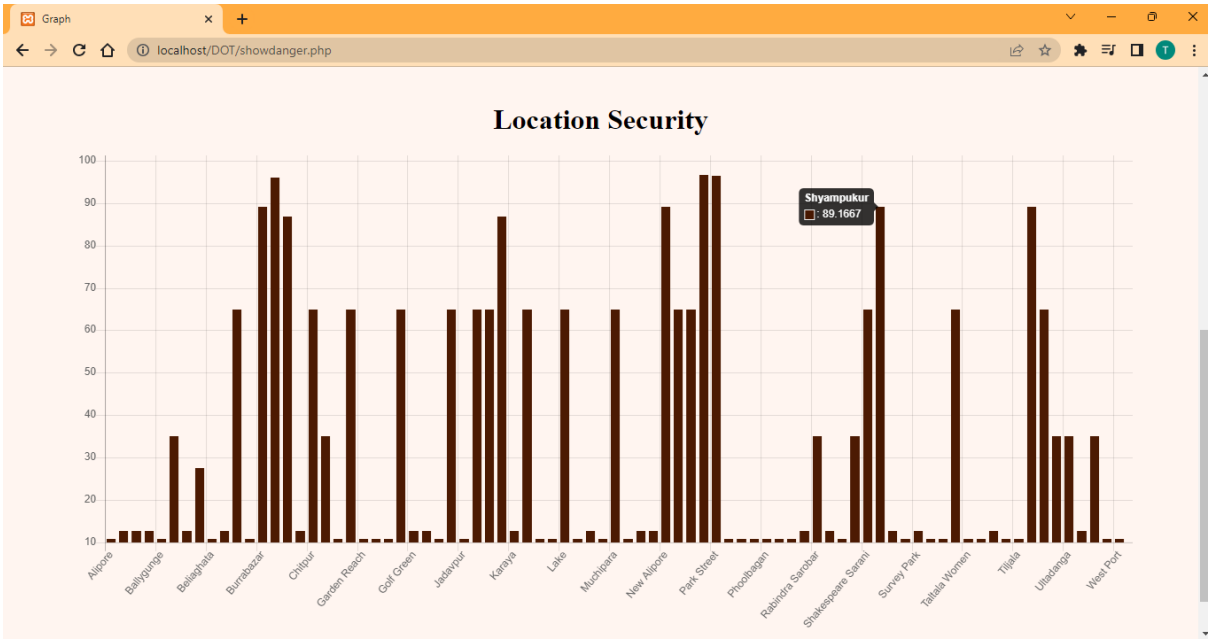

Figure 41: Screenshot of Danger (all) Prediction Page (2)

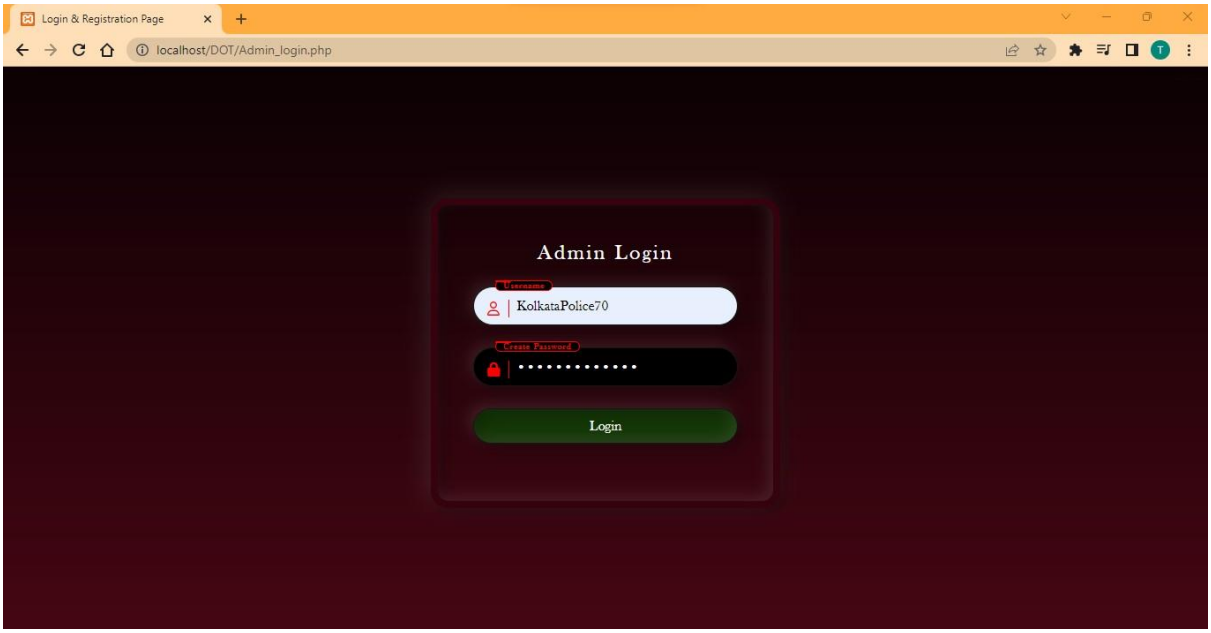

Figure 42: Screenshot of Admin Login Page

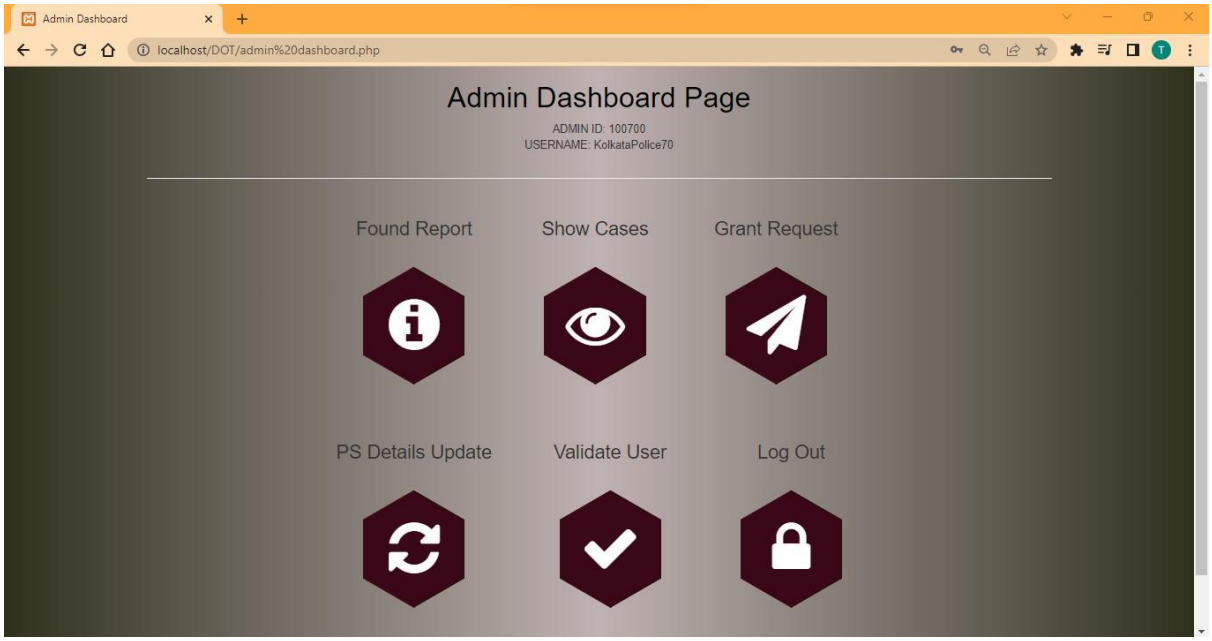

Figure 43: Screenshot of Admin Dashboard Page

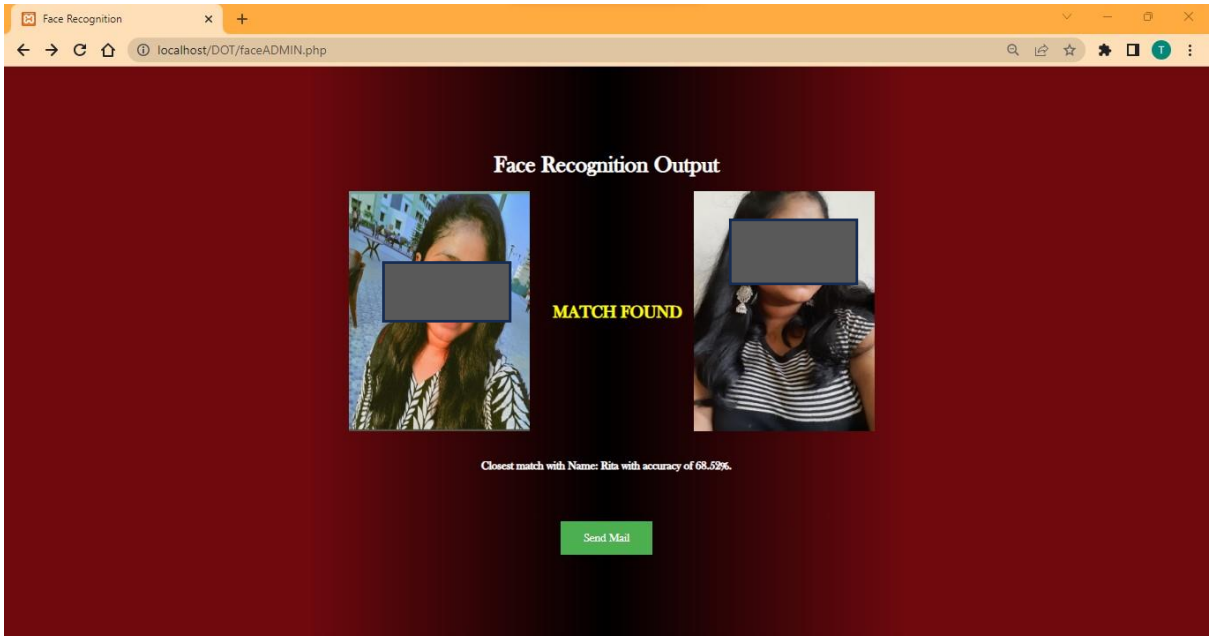

Figure 44: Screenshot of Match found Page for Primary Found report

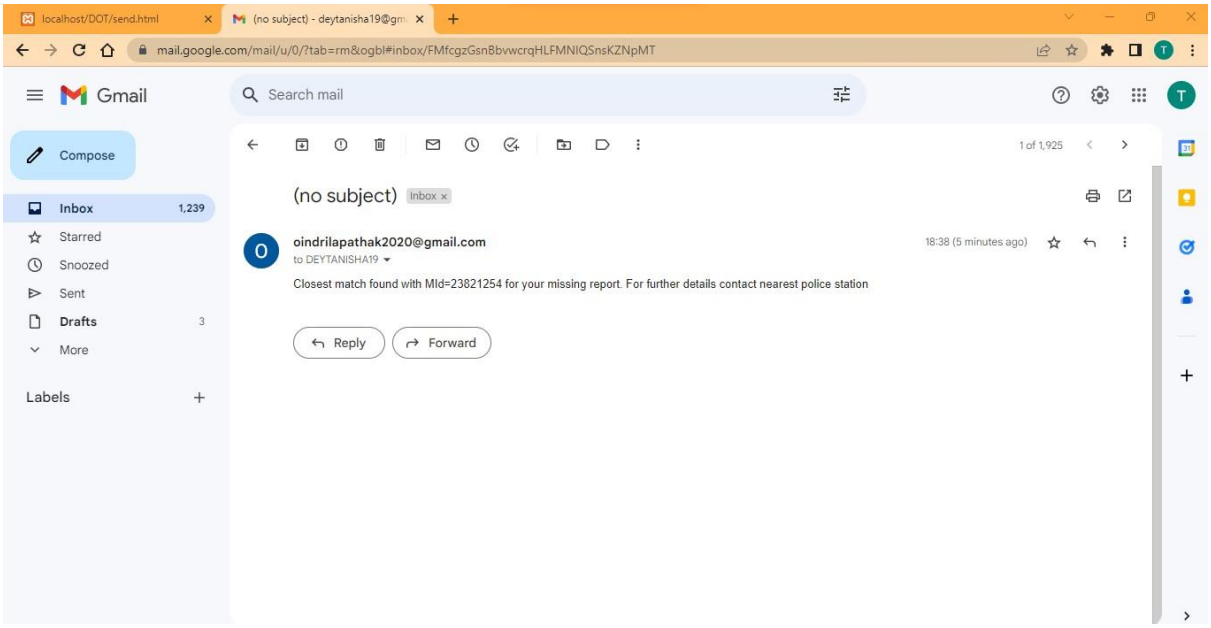

Figure 45: Screenshot of Gmail Page of the user with closest match found info.

Figure 46: Screenshot of Grant request Page

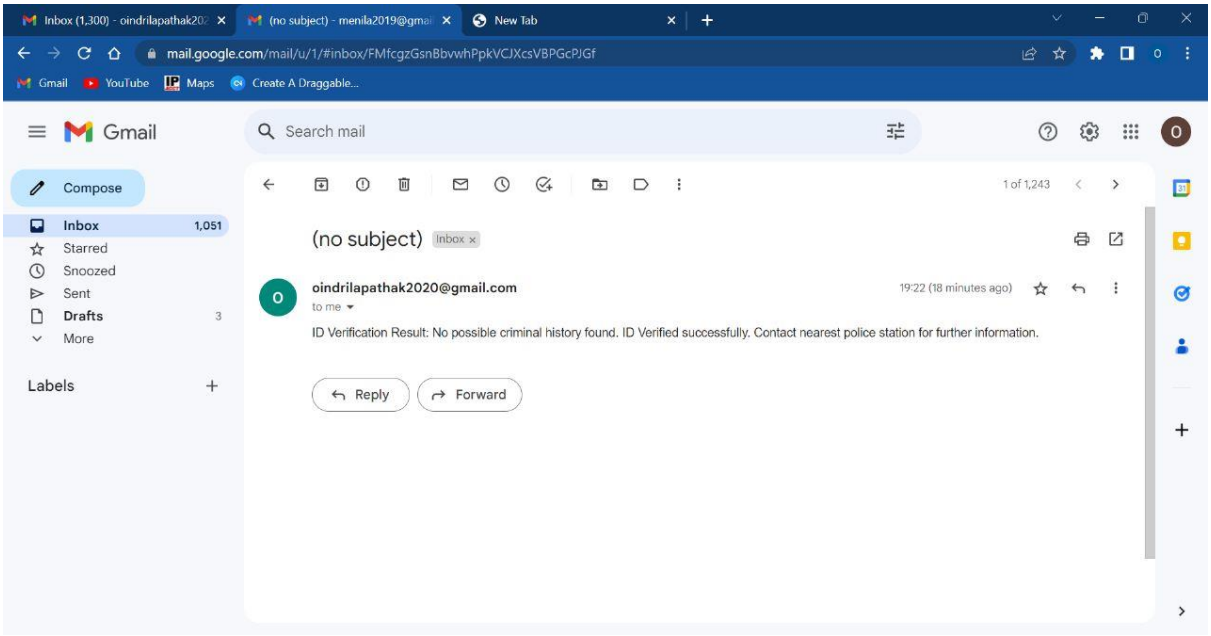

Figure 47: Screenshot of Gmail Page of the user with ID verification Result

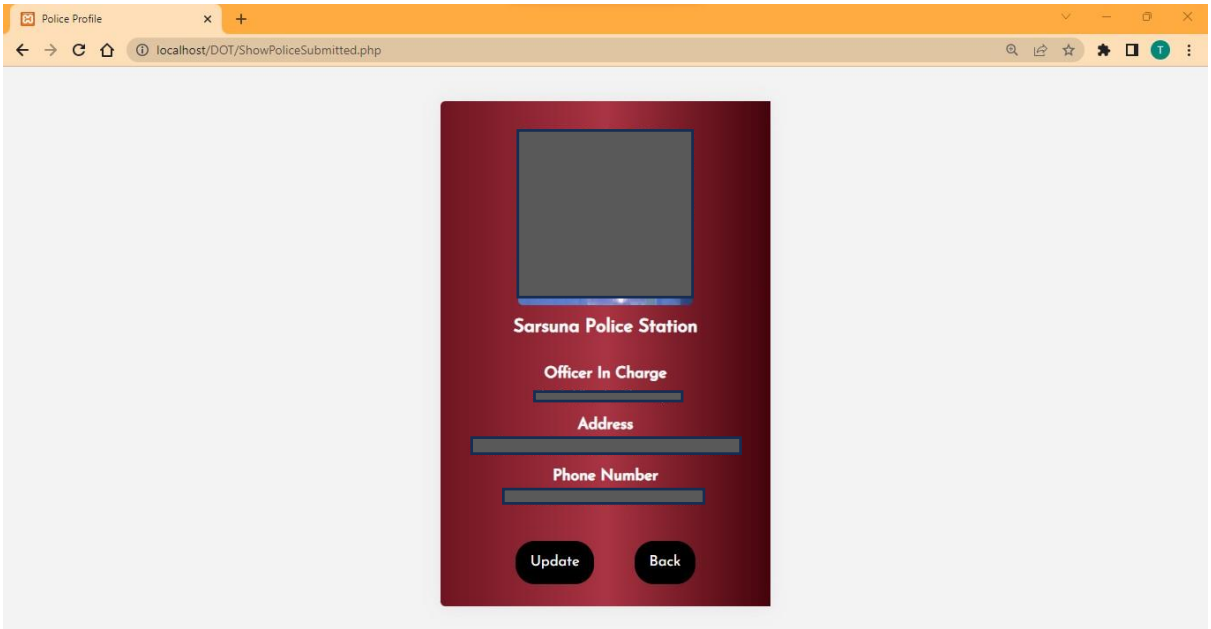

Figure 48: Screenshot of Show Police Station Page (Admin)

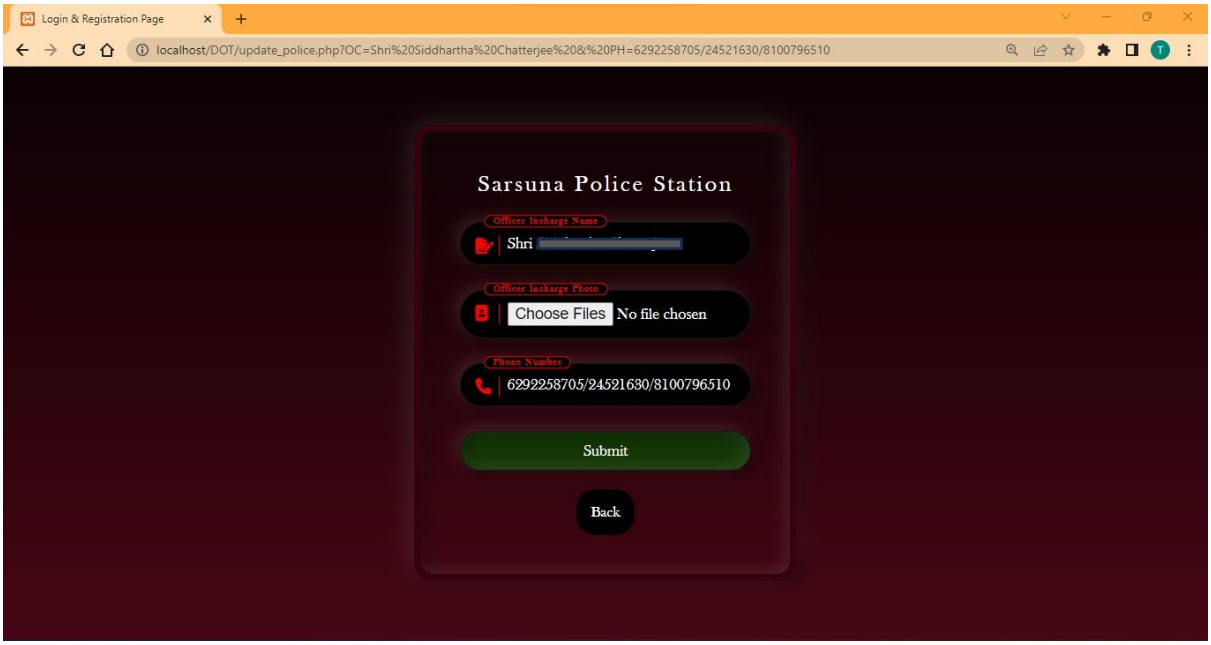

Figure 49: Screenshot of Police Details Update Page

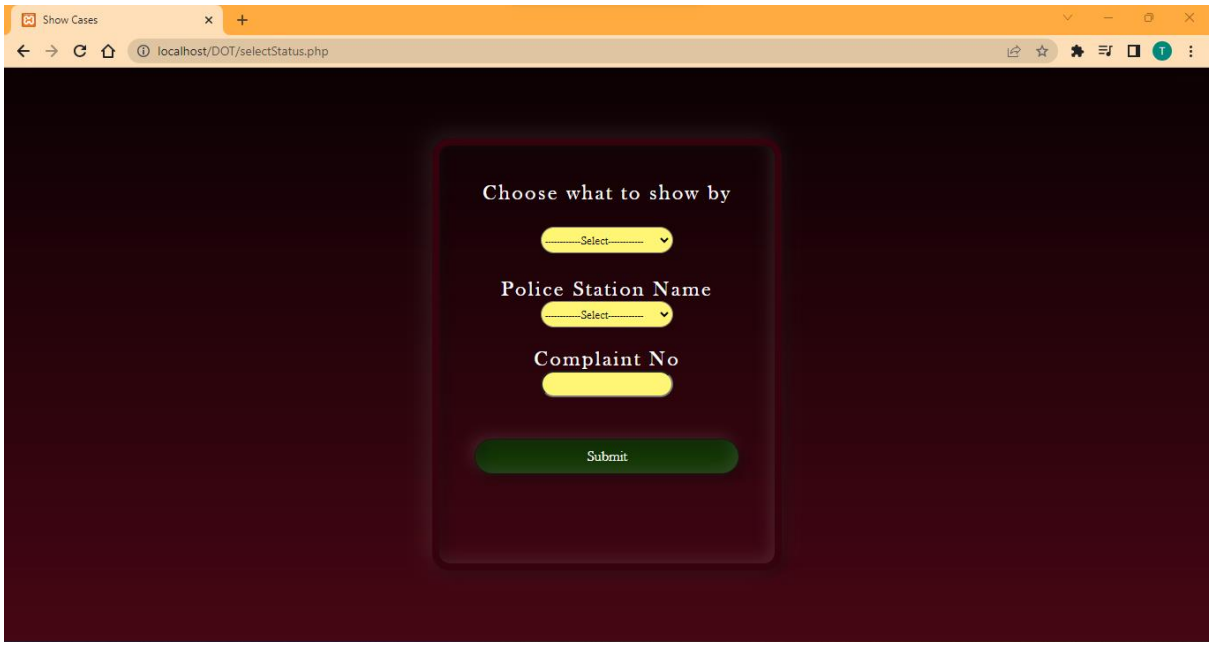

Figure 50: Screenshot of Show Cases Page

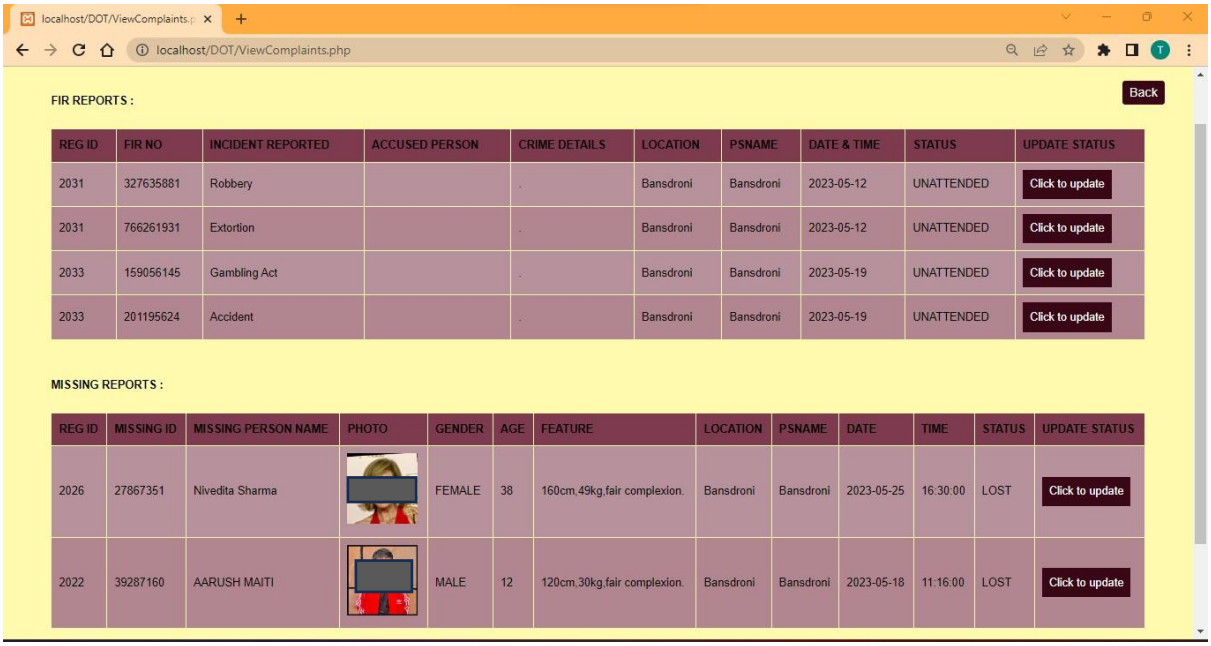

Figure 51: Screenshot of View Complaints Page (Admin)

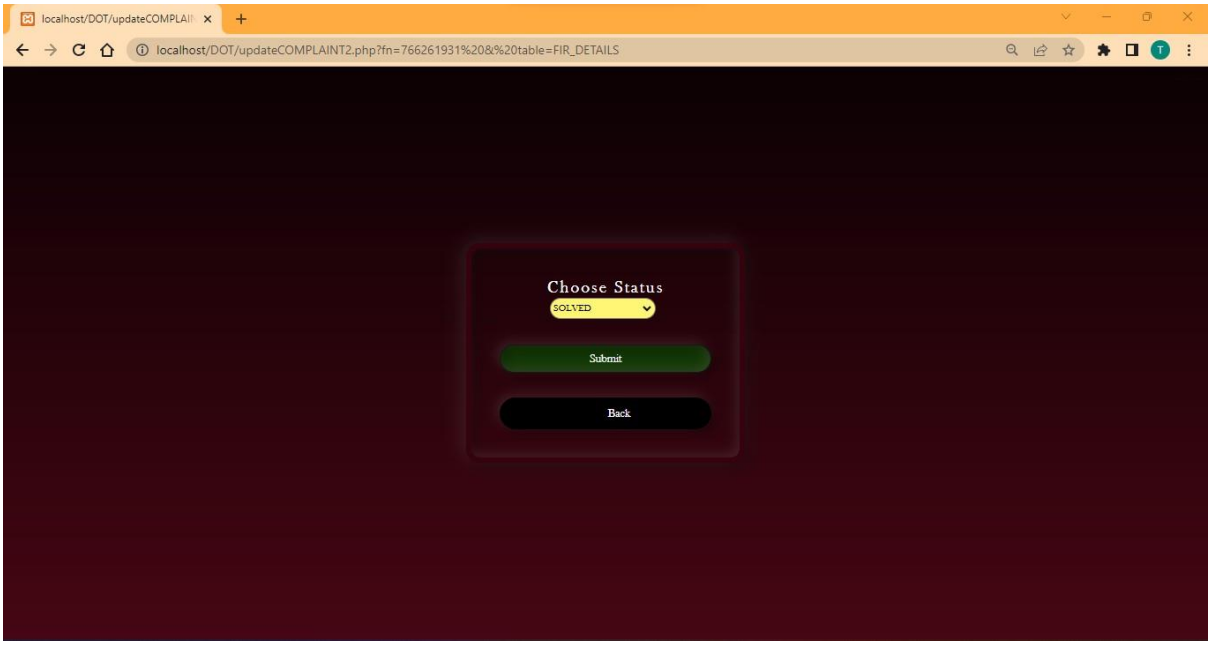

Figure 52: Screenshot of Update Complaints Page

REG ID	NAME	EMAIL ID	PHONE	GUARDIAN NAME	AGE	GENDER	ADDRESS	PHOTO ID TYPE	PHOTO	PHOTO ID NO	DELETE PROFILE
2021	TANISHA DEY	DEYTANISHA19@GMAIL.COM	7076968451	KAMAL DEY	21	FEMALE	NAIHATI	PAN CARD		AHUPD6702B	Delete Profile
2022	SUDIPTA MAITI	MAITISUDIPTA30@GMAIL.COM	8348536903	SATYA RANJAN MAITI	22	FEMALE	HALDIA TOWNSHIP , PURBA MEDINIPUR	PAN CARD		AGUKW8972C	Delete Profile
2023	PARAMITA DAS	DASPARAMITA1421@GMAIL.COM	8574632145	BISWESWAR DAS	24	FEMALE	BALLYGUNGE	AADHAR CARD		452169853254	Delete Profile
2024	ARUN KUMAR	ARUN23@GMAIL.COM	9856742135	TAPAN KUMAR	41	MALE	ALIPORE	PAN CARD		LGUKW8572C	Delete Profile
2025	ANKITA MANDAL	MANDALANKITA18@GMAIL.COM	9865472145	HEMABRATA MANDAL	25	FEMALE	TOLLYGUNGE	AADHAR CARD		652314527458	Delete Profile
2026	RAHUL SHARMA	ROHANANKIT51@GMAIL.COM	8452769421	ANKIT SHARMA	22	MALE	PHULBAGAN	PAN CARD		HGNKW8671K	Delete Profile

Figure 53: Screenshot of User Validation Page

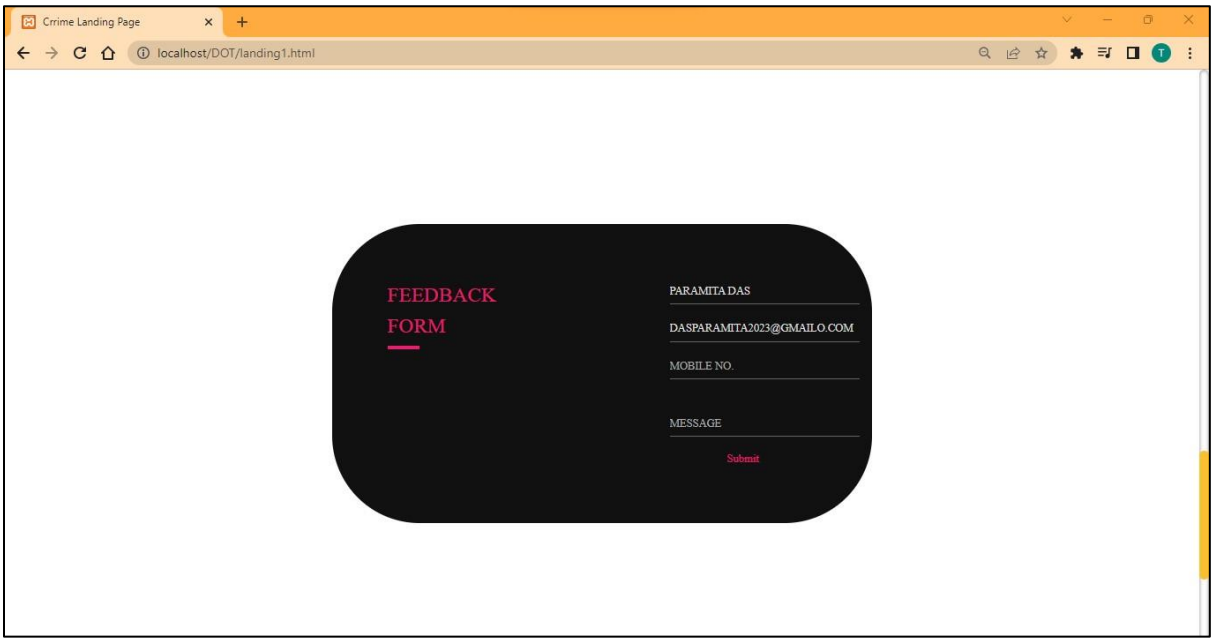

Figure 54: Screenshot of Feedback form section (Landing Page)

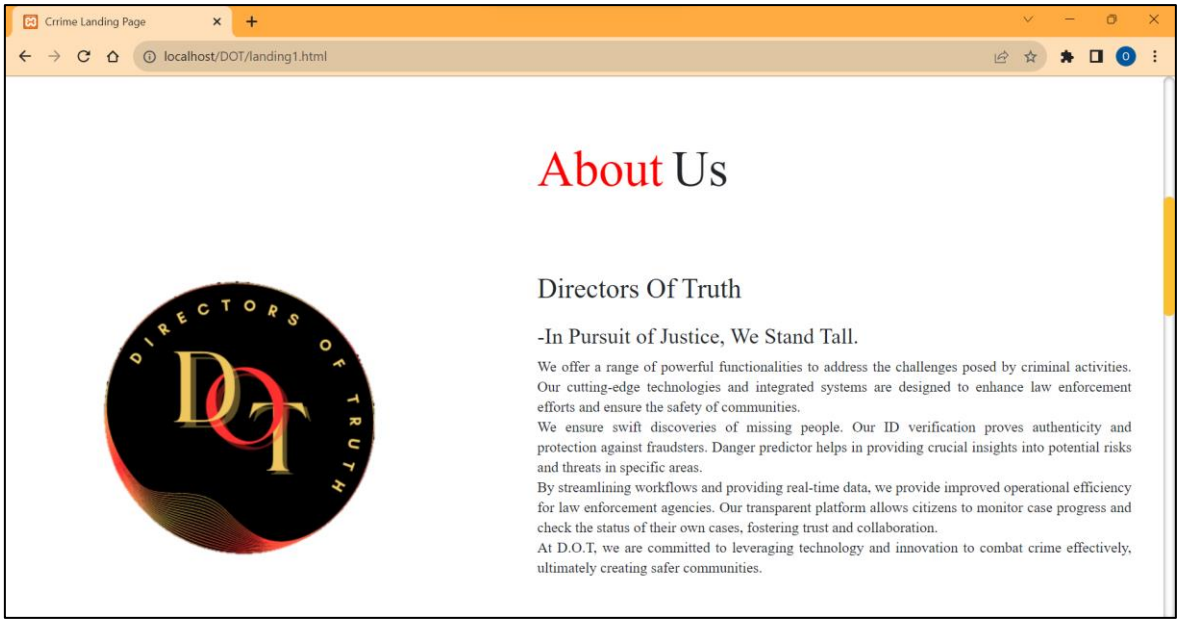

Figure 55: Screenshot of About Us section (Landing Page)

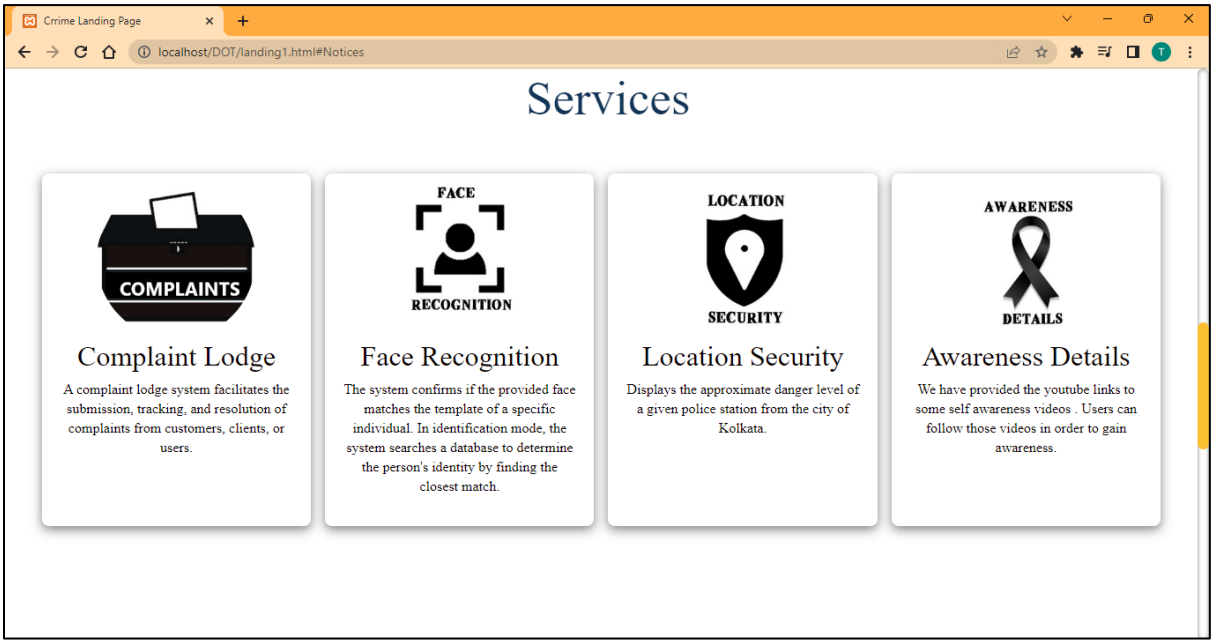

Figure 56: Screenshot of Services section (Landing Page)

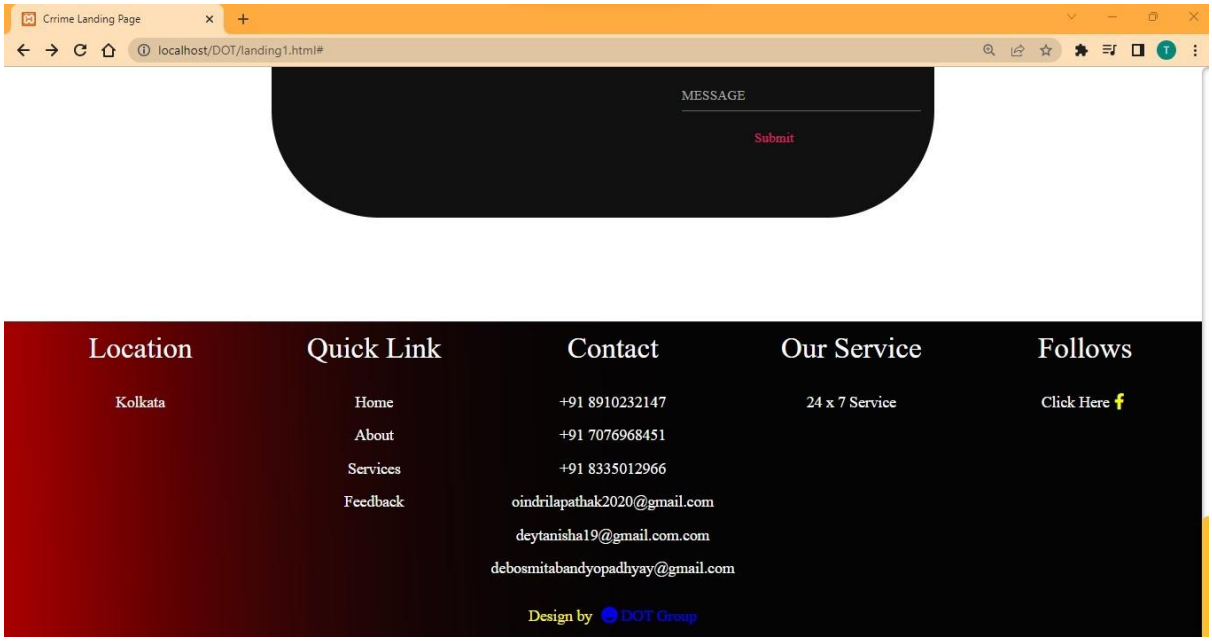

Figure 57: Screenshot of Contact Info. section (Landing Page)

7. FUTURE SCOPE

i. It's important to note that while fuzzy logic can handle uncertainty and imprecision, the accuracy and performance of the prediction models depend on the model's parameter's precision, which we may need to be adjusted and optimized through techniques like cross-validation or adaptive learning to improve prediction accuracy. Also, prediction model may be inaccurate due to insufficient or poor-quality data, overlooked variables and their priorities, or dynamic and unpredictable factors in the underlying system. We could not find a lot of data and most of the data is generated to simulate the real world. Furthermore, the rule base of the inference system needs more refinement through the assistance of experts in the domain of criminology.

ii. There are chances where the face recognition system might fail because of very low-quality image.

iii. Miscreants could use the system by exploiting information they should not be privy to in the first place.

iv. Maintenance of the software is necessary from time to time since there is no scope to upload bulk information about previous crime record or perceived danger levels.

v. The domain of the software could be expanded to contain police stations under West Bengal if not the whole of India at large.

vi. Digital Signature or similar methods could be used while generating the PDFs for FIR and GD to ensure the authenticity of the PDFs (thus, because of safety reasons we are currently generating PDFs only for GDs).

8. CONCLUSION

In conclusion, the necessity for a crime management system like D.O.T has become increasingly apparent in our modern society. By harnessing the power of technology and data analysis, such systems empower law enforcement agencies, communities, and policymakers to combat incident more effectively. From enhancing public safety to improving operational efficiency of incident management bodies and supporting investigations, D.O.T can serve as a critical tool in creating secure and thriving societies.

9. REFERENCES

1. Analysis of the security reasons behind compulsory login of a user: https://www.whitehatsec.com/glossary/content/login-page#:~:text=A%20login%20page%20is%20a,or%20part%20of%20a%20website
2. Significance and description of a homepage: https:// www.computerhope.com /jargon/ h/ homepage.html
3. Introduction to Fuzzy Logic: https://www.geeksforgeeks.org/fuzzy-logic-introduction/
4. "Introduction to Soft Computing-Neuro Fuzzy and Genetic Algorithm" by Samir Roy, Udit Chakraborty
5. Center of Sum Method of Defuzzification: https://codecrucks.com/center-of-sums-cos method-for defuzzification/#:~:text=Center%20of%20Sums%20(CoS)%20method%20is%20most%20co mmonly%20used%20defuzzification,otherwise%20identical%20to%20CoG%20method.
6. Face Recognition Process: https://www.computervision.zone/lessons/face-attendance-video-lesson/
7. Fuzzyalgorithm: https://scikit-fuzzy.readthedocs.io/en/latest/auto_examples/plot_tipping_problem.html
8. Rani A, Rajasree S (2014) Crime trend analysis and prediction using mahanolobis distance and dynamic time warping technique. Int J Comput Sci Inf Technol 5(3):4131–4135
9. Gorr W, Harries R (2003) Introduction to crime forecasting. Int J Forecast 19(4):551–555. https://doi.org/10.1016/S0169-2070(03)00089-X
10. Gorr W, Olligschlaeger AM, Thompson Y (2000) Assessment of crime forecasting accuracy for deployment of police. Int J Forecast 2000:743–754
11. Farjami, Y., Abdi, K. A genetic-fuzzy algorithm for spatio-temporal crime prediction. J Ambient Intell Human Comput (2021). https://doi.org/10.1007/s12652-020-02858-3
12. An Attribute Weighted Fuzzy Clustering Algorithm for Mixed Crime Data Revatthy Krishnamurthy1 and J. Satheesh Kumar2 (2016)
13. A REAL-TIME CRIME RECORDS MANAGEMENT SYSTEM FOR NATIONAL SECURITY AGENCIES Oludele Awodele, Onuiri Ernest E., Olaore Olufunmike A., Sowunmi Oluwawunmi O. Ugo-Ezeaba Anita A. (2015)