

A

Project Report

on

AI ASSISTED SEARCH FOR IDENTIFYING MISSING PERSONS

Submitted in partial fulfillment of the requirement

For the award of the degree of

BACHELOR OF TECHNOLOGY

IN

CSE (DATA SCIENCE)

Submitted by

P.S.R.K.Sri Ramya

20A31A4415

K. Akankshya

20A31A4410

G. Jaya Avinash

21A35A4403

P. Kavya Sri

20A31A4416

A. L. Thanmai

20A31A4402

Under the esteemed guidance of

Mrs. Ch.Veera Gayathri M.Tech.,

Assistant Professor of CSE (DS)



DEPARTMENT OF CSE (DATA SCIENCE)

PRAGATI ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE, Permanently Affiliated to JNTUK, KAKINADA, Accredited by NBA)

ADB Road, Surampalem, Near Peddapuram, Kakinada District, AP- 533437

2020-2024

A

Project Report

on

AI ASSISTED SEARCH FOR IDENTIFYING MISSING PERSONS

Submitted in partial fulfillment of the requirement

For the award of the degree of

BACHELOR OF TECHNOLOGY

IN

CSE (DATA SCIENCE)

Submitted by

P.S.R.K.Sri Ramya

20A31A4415

K. Akankshya

20A31A4410

G. Jaya Avinash

21A35A4403

P. Kavya Sri

20A31A4416

A. L. Thanmai

20A31A4402

Under the esteemed guidance of

Mrs. Ch.Veera Gayathri M.Tech.,

Assistant Professor of CSE (DS)



DEPARTMENT OF CSE (DATA SCIENCE)

PRAGATI ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE, Permanently Affiliated to JNTUK, KAKINADA, Accredited by NBA)

ADB Road, Surampalem, Near Peddapuram, Kakinada District, AP- 533437

2020-2024

PRAGATI ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, Accredited by NBA)

ADB Road, Surampalem, Near Peddapuram, Kakinada District, AP-533437

CERTIFICATE

DEPARTMENT OF CSE (DATA SCIENCE)



This is to certify that the project report entitled “AI ASSISTED SEARCH FOR IDENTIFYING MISSING PERSONS” is being submitted by **P.S.R.K.Sri Ramya** (20A31A4415), **K. Akankshya** (20A31A4410), **G. J. Avinash** (21A35A4403), **P. Kavya Sri** (20A31A4416), **A .L. Thanmai** (20A31A4402) in partial fulfilment for the award of the Degree of **Bachelor of Technology**, during the year **2020-2024** in CSE (Data Science) of Pragati Engineering College, for the record of a bonafide work carried out by them.

Project Supervisor:

Mrs. Ch. Veera Gayathri M.Tech.,

Assistant Professor

Dept. of CSE (Data Science)

Pragati Engineering College (A)

Head of the Department:

Mr.M.V.Rajesh M.Tech., (Ph.D)

Associate Professor & HOD

Dept. of CSE (Data Science)

Pragati Engineering College (A)

ACKNOWLEDGEMENT

We express our thanks to project supervisor **Mrs.Ch.Veera Gayathri**, Assistant Professor of **CSE (Data Science)**, who deserves a special note of thanks and gratitude, for having extended their fullest co-operation and guidance, without this, project would never have materialized.

We express our deep sense of gratitude to **Mr.M.V.Rajesh**, Associate Professor and HOD of **CSE (Data Science)**, for having shown keen interest at every stage of development of our project and for guiding us in every aspect.

We wish to express our special thanks to our beloved **Dr. K. SATYANARAYANA, Professor & Principal** for giving guidelines and encouragement.

We wish to express our special thanks to our beloved **Dr. G.NARESH, Professor & Vice Principal** for giving guidelines and encouragement.

We wish to express sincere gratitude to our beloved and respected **Dr. P. KRISHNA RAO, Chairman** and **Sri. M. V. HARANATHA BABU, Director (Management)** and **Sri. M. SATISH, Vice-President** for their encouragement and blessings.

We are thankful to all our faculty members of the Department for their valuable suggestions. Our sincere thanks are also extended to all the teaching and non-teaching staff of Pragati Engineering College.

P.S.R.K.Sri Ramya	20A31A4415
K. Akankshya	20A31A4410
G. J. Avinash	21A35A4403
P. Kavya Sri	20A31A4416
A. L. Thanmai	20A31A4402

ABSTRACT

The fact that half of the 174 children who go missing each day in India are still unaccounted for is incredibly troubling. According to a National Crime Records Bureau (NCRB) data, more than one lakh children (1, 11,569 in all) were reported missing up until 2016, and 55,625 of them were still unaccounted for at the end of the year that was cited by the Ministry of Home Affairs (MHA) in Parliament (LS Q no. 3928, 20-03-2018). In order to tackle and help out Police, AI-Assisted Search for Missing Children a GUI application was created in Python that the police may use to open a new case and find missing children. The missing person's image that has been supplied is processed on the backend, and important information are recorded. This is kept in a database together with more details like name, parent's name, age, place, etc. An application using Android is also developed which can be made accessible to the general public. Using this application, a person's photo can be uploaded if found suspicious by the user. There is an option of submitting the photo obliquely or anonymously. Along with the location, this image is saved in the database. The GUI application applies a machine learning algorithm to compare user-submitted photos with those uploaded by the police. Any matches along with the place where the missing individual was last seen if they are discovered can displayed.

TABLE OF CONTENTS

ABSTRACT	1
LIST OF FIGURES	3
Chapter 1. Introduction	5
Chapter 2. Literature Survey	7
Chapter 3. System Analysis	12
3.1 Existing System	13
3.2 Proposed System	13
Chapter 4. System Design	15
4.1 System Architecture	16
4.2 UML Representation	16
Chapter 5. System Implementation	26
5.1 Modules	27
5.2 System Requirements	28
Chapter 6. System Test	30
6.1 Types of Testing	31
6.2 Test Cases	32
Chapter 7. Screen Shots	34
Chapter 8. Conclusion and Future Work	45
References	46
APPENDIX-A Source Code	47
PAPER PUBLICATION	56-66

LIST OF FIGURES

Figure No.	Name of the Figure	Page. No.
CHAPTER 3		
SYSTEM ANALYSIS		
3.2.1	KNN Classifier	14
CHAPTER 4		
SYSTEM ARCHITECTURE		
4.1.1	System Architecture	16
4.2.1	Use Case Diagram of Police	17
4.2.2	Use Case Diagram of User	18
4.2.3	Sequence Diagram of Police	19
4.2.4	Sequence Diagram of User	20
4.2.5	Deployment Diagram	21
4.2.6	Activity Diagram of Police	22
4.2.7	Activity Diagram of User	23
4.2.8	Data flow Diagram	24
CHAPTER 7		
SCREENSHOTS		
7.1	Home Page	34
7.2	About Us	34
7.3	User Login	35
7.4	User Sign-Up	35
7.5	Police Log-in	36

7.6	Login Successful Pop-Up	36
7.7	User Dashboard	37
7.8	Pending Cases	37
7.9	My Cases	38
8.0	User Profile	38
8.1	User Feedback	39
8.2	Register Case	39
8.3	Login Successful Pop-Up	40
8.4	Police Dashboard	40
8.5	Add New Case	41
8.6	Manage Cases	41
8.7	Find Match	42
8.8	User Submitted Cases	42
8.9	Case Analysis Graph	43
9.0	Sentimental Analysis	43

CHAPTER – 1

INTRODUCTION

1.INTRODUCTION

- Every year, countless numbers of missing children are reported in India. It is really distressing to see that children and other people going missing is the norm in times like these, when crime rates are at record highs.
- A society has to be created in such a way that, it has to be healthy and secure for the children. A system has to be developed to keep track and locate the missing children which will be helpful for anti-crime authorities.
- In order to locate missing children and victims of human trafficking, face recognition technology can be employed. Given the concerning figures, it's critical to have an accessible and easy-to-use method of finding the missing kids.
- With image, it will be easier to manage records of missing people and children on a centralized database. It is simple to update or delete the records. It can be used to add, update, maintain, and remove records more quickly. This software will assist the police and the guardians swiftly in locating the missing children or person at anytime and anywhere.

CHAPTER – 2

LITERATURE SURVEY

2. LITERATURE SURVEY

- Face recognition is the task of identifying an individual from an image of their face and a database of known faces. Omkar M. Parkhi et al.[1] stated that despite being a relatively easy task for most humans, “unconstrained” face recognition by machines, specifically in settings such as malls, casinos and transport terminals, remains an open and active area of research. However, in recent years, a large number of photos have been crawled by search engines, and uploaded to social networks, which include a variety of unconstrained material, such as objects, faces and scenes. This large volume of data and the increase in computational resources have enabled the use of more powerful statistical models for general challenge of image classification. This research project evaluates the use of big data dependent machine learning approaches such as deep convolutional neural networks for image classification for the problem of unconstrained facial recognition.
- Rohit Sathe et al.[2], In India a countless number of children are reported missing every year. Among the missing child cases a large percentage of children remain untraced. This paper presents a novel use of deep learning methodology for identifying the reported missing child from the photos of multitude of children available, with the help of face recognition. The public can upload photographs of suspicious child into a common portal with landmarks and remarks. The photo will be automatically compared with the registered photos of the missing child from the repository. Classification of the input child image is performed and photo with best match will be selected from the database of missing children. For this, a deep learning model is trained to correctly identify the missing child from the missing child image database provided, using the facial image uploaded by the public. The Convolutional Neural Network (CNN), a highly effective deep learning technique for image based applications is adopted here for face recognition. Face descriptors are extracted from the images using a pre-trained CNN model VGG-Face deep architecture. Compared with normal deep learning applications, our algorithm uses convolution network only as a high level feature extractor and the child recognition is done by the trained SVM classifier. Choosing the best performing CNN model for face recognition, VGG-Face and proper training of it results in a deep learning model invariant to noise, illumination, contrast, occlusion, image pose and age of the child and it

outperforms earlier methods in face recognition based missing child identification. The classification performance achieved for child identification system is 99.41%. It was evaluated on 43 Child cases.

- Sumeet Pate et al.[3],The main aim of this paper is to identify and locate missing persons, child's and most wanted criminals anywhere any time any place. This paper is to detect this person form any remote CCTV cameras and MOBILE camera as soon as our software detect this person. This application will automatically sends SMS alert to the nearby area police station of that area and also where the main details are registered by the police station. To perform face recognition in different light, and expression, this application extracts various patterns from image. Specifically, this System work on LEM algorithm in this we detect the point and calculate the line segment hausdorff (LSD) distance and then computes the feature. It is a novel face image descriptor inspired by the unique Edge structure of human faces. It is efficient and only doubles the cost of computing patterns, yet is extremely accurate to pose and expression variations. This new system will allow security to the data, by mean of authorizing users. Only those users who have a valid user-id and password can access the system.
- Bharath Darshan Balar et al.[4],In the world, a countless number of people are missing every day which includes kids, teens, mentally challenged, old-aged people with Alzheimer's, etc. Most of them remain untraced. When a person goes missing, the people related to that person or the police can upload the picture of the person which will get stored in the database. The face recognition model in our system will try to find a match in the database with the help of face encodings. It is performed by comparing the face encodings of the uploaded image to the face encodings of the images in the database. If a match is found, it will be notified to the police and the people related to that person along with the location of where the person is found. The face recognition model that we have used maintains an accuracy of 99.38% on the Labelled Faces in the Wild Benchmark which comprises of 13,000 images
- Saurabh P. Bahurupi et al.[5], We have proposed a patch-based principal component analysis (PCA) method to deal with face recognition. Many PCA-based methods for face recognition utilize the correlation between pixels, columns, or rows. But the local spatial information is not utilized or not fully utilized in these methods. We believe that patches are more meaningful basic units for face recognition than pixels, columns, or rows, since faces are discerned by patches containing eyes and noses. To calculate the correlation between patches, face images are divided into patches and then these patches are converted to

column vectors which would be combined into a new "image matrix." By replacing the images with the new "image matrix" in the two-dimensional PCA framework, we directly calculate the correlation of the divided patches by computing the total scatter. Extensive experiments on the ORL and FERET face database are reported to illustrate the performance of the patch-based PCA. Our method promotes the accuracy compared to one-dimensional PCA, two-dimensional PCA, and two-directional two-dimensional PCA.

- Hsin-rung Chou et al.[6], Many face recognition systems boost the performance using deep learning models, but only a few researches go into the mechanisms for dealing with online registration. Although we can obtain discriminative facial features through the state-of-the-art deep model training, how to decide the best threshold for practical use remains a challenge. We develop a technique of adaptive threshold mechanism to improve the recognition accuracy. We also design a face recognition system along with the registering procedure to handle online registration. Furthermore, we introduce a new evaluation protocol to better evaluate the performance of an algorithm for real-world scenarios. Under our proposed protocol, our method can achieve a 22% accuracy improvement on the LFW dataset.

- Shefali Patil et al.[7], Identifying missing people and bringing them back to their families has become a universal issue. Various research publications are examined in this paper. Each of the existing mechanisms has merits and demerits. But the issues related to bringing back missing people have not been perfected 100%. Computing technology has evolved in recent years to include a wide range of flavors that can be used in practically every sector. Information plays a critical function in the computing system, notwithstanding the rapid-fire rise of technology. Every day, a considerable number of people, including children, teenagers, mentally challenged people, elderly people with Alzheimer's disease, and others, go missing around the world. In India, more than 500 missing person concerns are estimated to go unaddressed every day. A facial recognition system is a computer application that can recognize or verify a person by analyzing a digital image or a video frame from a video source. Facial feature detection and recognition are extensively used in current world scenarios and technologies. Artificial intelligence, on the other hand, has given solutions to the issues of the ultramodern world. Artificial intelligence (AI) has been developed to help humans and machines communicate more effectively. The proposed mechanism has been successfully implemented to accurately identify a face with a precision of 90% when compared to 59% using KNN and 43% using SVM with PCA.

-
- Birari Hetal et al.[8],Now-a- days finding the missing person is very difficult task to find out by people or police department, lots of documentation and hard work is there also it takes the lot of time duration as well as there is no guaranty of appropriate result. This application contains functionality to add complaint as well as view all complaints. By using these complaints, Trust members will try to find lost person in various areas. This application will upload complaint on web server which can be accessed by any of the trust member having this application. This project Finding Missing Person using Face Detection on Android Application presents the solution for this problem. We are using four modules User, Police, Compliant holder, Admin for getting appropriate result. Admin continuously Update database and Delete unnecessary data
 - Weiyang Liu et al.[9],This paper addresses deep face recognition (FR) problem under open-set protocol, where ideal face features are expected to have smaller maximal intra-class distance than minimal inter-class distance under a suitably chosen metric space. However, few existing algorithms can effectively achieve this criterion. To this end, we propose the angular softmax (A-Softmax) loss that enables convolutional neural networks (CNNs) to learn angularly discriminative features. Geometrically, A-Softmax loss can be viewed as imposing discriminative constraints on a hypersphere manifold, which intrinsically matches the prior that faces also lie on a manifold. Moreover, the size of angular margin can be quantitatively adjusted by a parameter m . We further derive specific m to approximate the ideal feature criterion. Extensive analysis and experiments on Labeled Face in the Wild (LFW), Youtube Faces (YTF) and MegaFace Challenge show the superiority of A-Softmax loss in FR tasks. The code has also been made publicly available.
 - Paul Upchurch et al.[10],We propose Deep Feature Interpolation (DFI), a new data-driven baseline for automatic high-resolution image transformation. As the name suggests, it relies only on simple linear interpolation of deep convolutional features from pre-trained convnets. We show that despite its simplicity, DFI can perform high-level semantic transformations surprisingly well - sometimes even matching or outperforming the state-of-the-art. This is particularly unexpected as DFI requires no specialized network architecture or even any deep network to be trained for these tasks. DFI therefore can be used as a new baseline to evaluate more complex algorithms and provides a practical answer to the question of which image transformation tasks are still challenging in the rise of deep learning.

CHAPTER - 3

SYSTEM ANALYSIS

3.SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system, titled "AI-Assisted Search for Missing Children," addresses the alarming issue of unaccounted-for missing children in India. The system consists of a GUI application developed in Python for use by law enforcement and an Android application accessible to the general public. The GUI application allows police to initiate new cases, processing supplied images of missing individuals on the backend while recording essential details such as name, parent's name, age, and location in a comprehensive database. Simultaneously, with options for anonymous submissions. Uploaded images, along with location data, are stored in the database. The system employs an ML algorithm to compare user-submitted photos with those provided by the police. In the event of a match, including the last-known location of the missing individual. By combining technology, public participation, and law enforcement efforts, the system aims to enhance the search for missing children and contribute to their swift and safe recovery.

Disadvantages of Existing System:

1. Dependency on image quality.
2. Limited data accessibility.
3. Privacy concerns.

3.2 PROPOSED SYSTEM

A. Classification Algorithm

An example of supervised learning is the classification algorithm which classifies fresh findings derived from training data. A program that does classification divides fresh observations into various classes or groups after learning from the provided dataset or observations. Targets, labels, or categories can be used to describe classes in all cases. In contrast to regression, classification's output variable is a category rather than a value. The Classification algorithm uses labeled input data because it is a supervised learning technique, therefore it comprises input and output information.

B. KNN Algorithm

K-Nearest Neighbor is the most basic supervised learning based machine learning algorithms. This algorithm places the new case in the category that matches the existing categories the most by assuming a correlation between the new instance and the data and previous cases. In order to classify a new data point based on similarity, it stores all of the existing data. This shows that the KNN method can quickly and accurately classify new data. Although it can be used for both classification and regression, classification issues receive the majority of its employment.

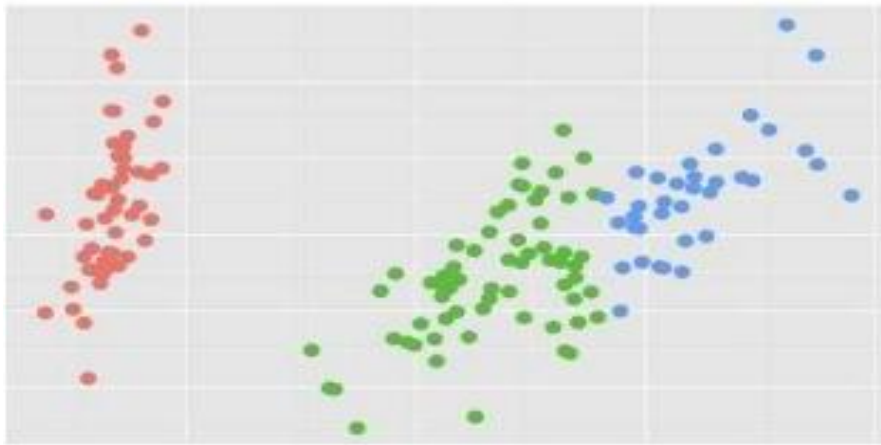


Figure 3.2.1 KNN Classifier

C. Pyqt5 and PostgreSQL

A toolkit for GUI widgets is PyQt. It is a Python interface for the robust and well-liked cross-platform GUI library Qt. The PyQt API is made up of various modules that contain a wide range of classes and functions. The free freeware PyQt is a plug-in for Python that implements the cross-platform GUI toolkit Qt. Both SQL (relational) and JSON (non-relational) querying are supported by PostgreSQL, a powerful, enterprise-class open source relational database. The database system is very reliable. For many web, mobile

Advantages of Proposed System:

1. Enhanced speed and efficiency.
2. Increased public involvement.

CHAPTER - 4

SYSTEM DESIGN

4.SYSTEM ARCHITECTURE

4.1 SYSTEM ARCHITECTURE

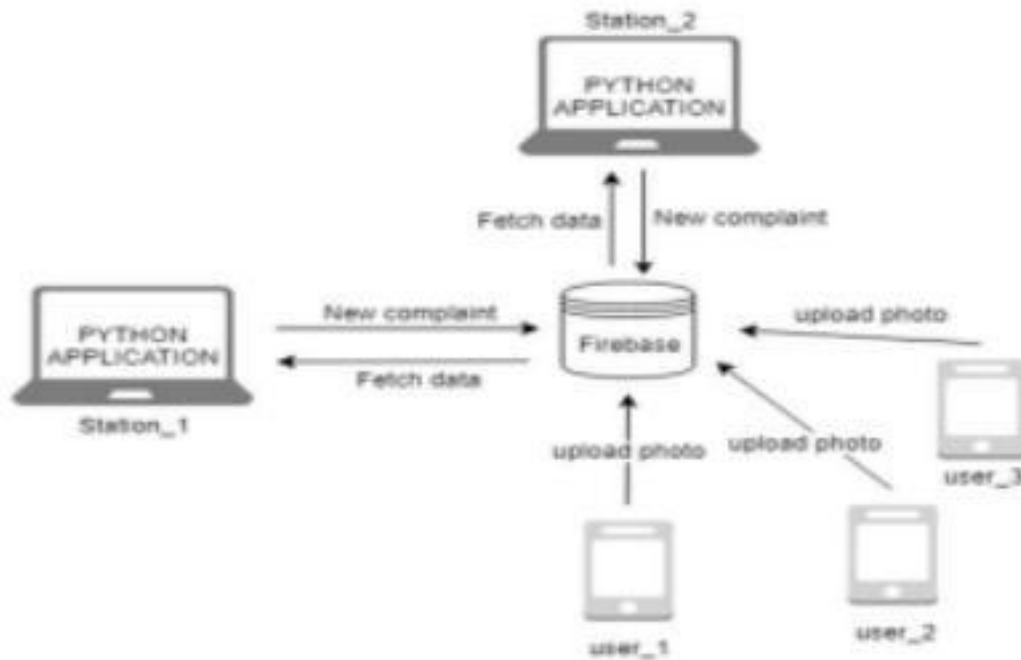


Figure 4.1.1 System Architecture

4.2 UML REPRESENTATION

4.2.1 Use case Diagram:

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

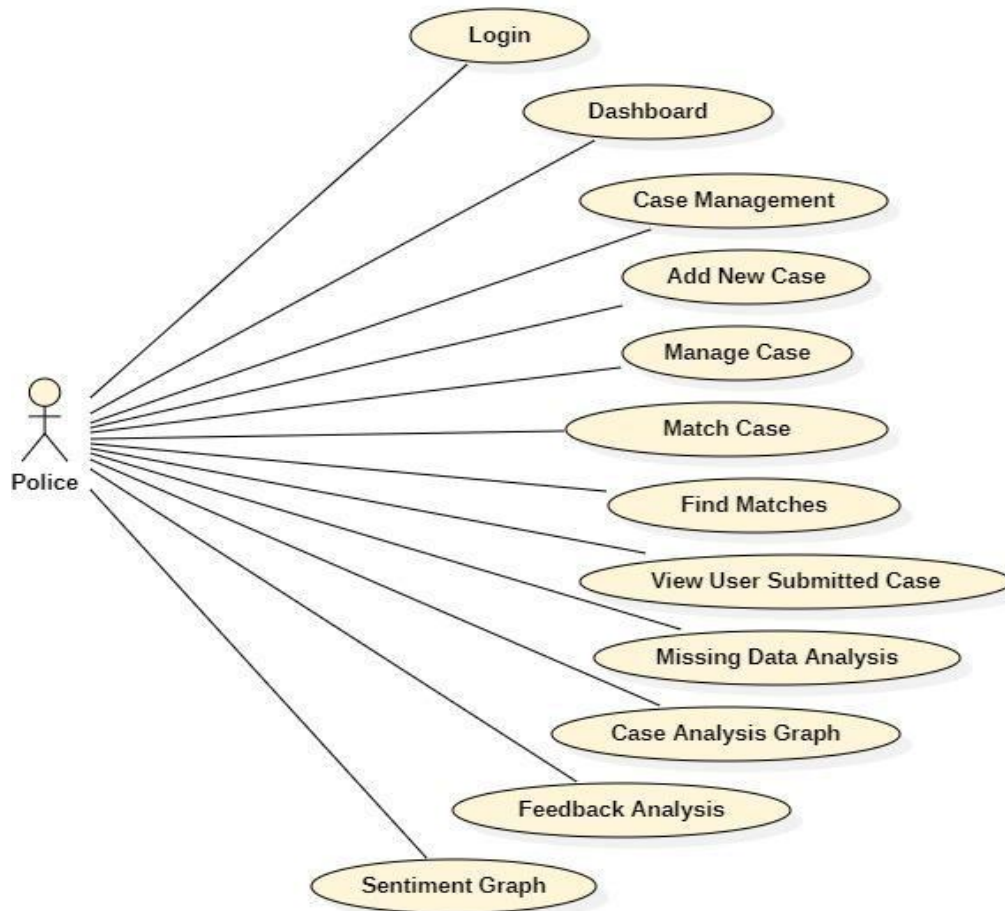


Figure 4.2.1 Use Case Diagram of Police

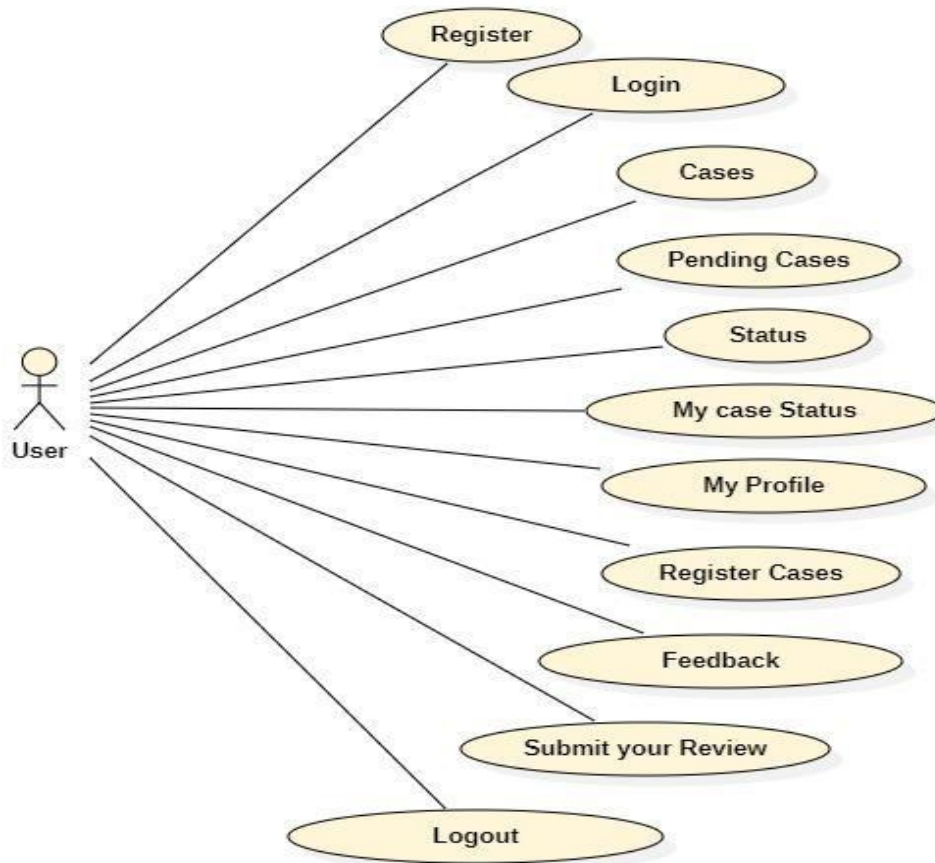


Figure 4.2.2 Use Case Diagram of User

4.2.2 Sequence diagram;

A **sequence diagram** is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams**, **event scenarios**, and timing diagrams.

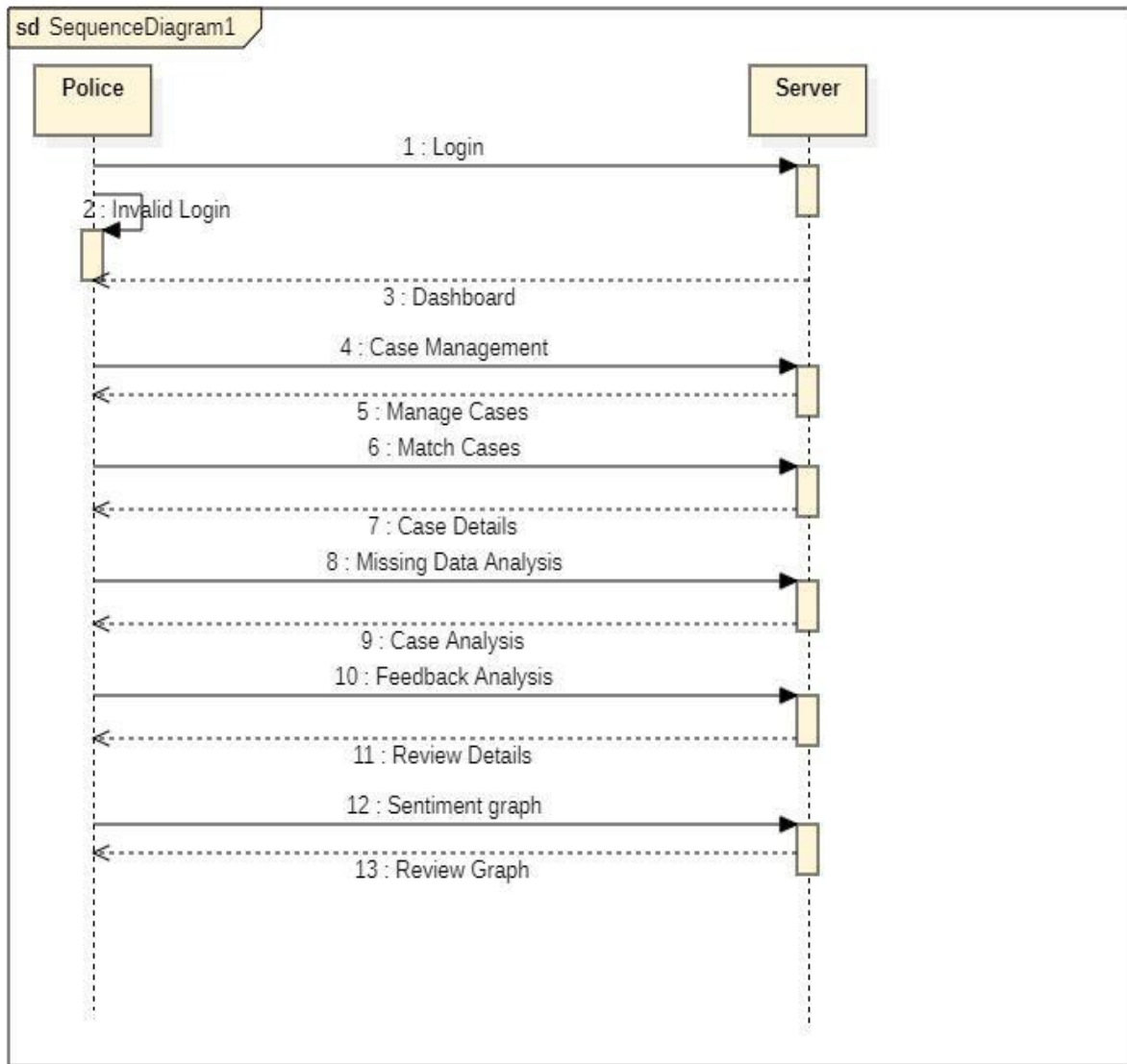


Figure 4.2.3 Sequence Diagram of Police

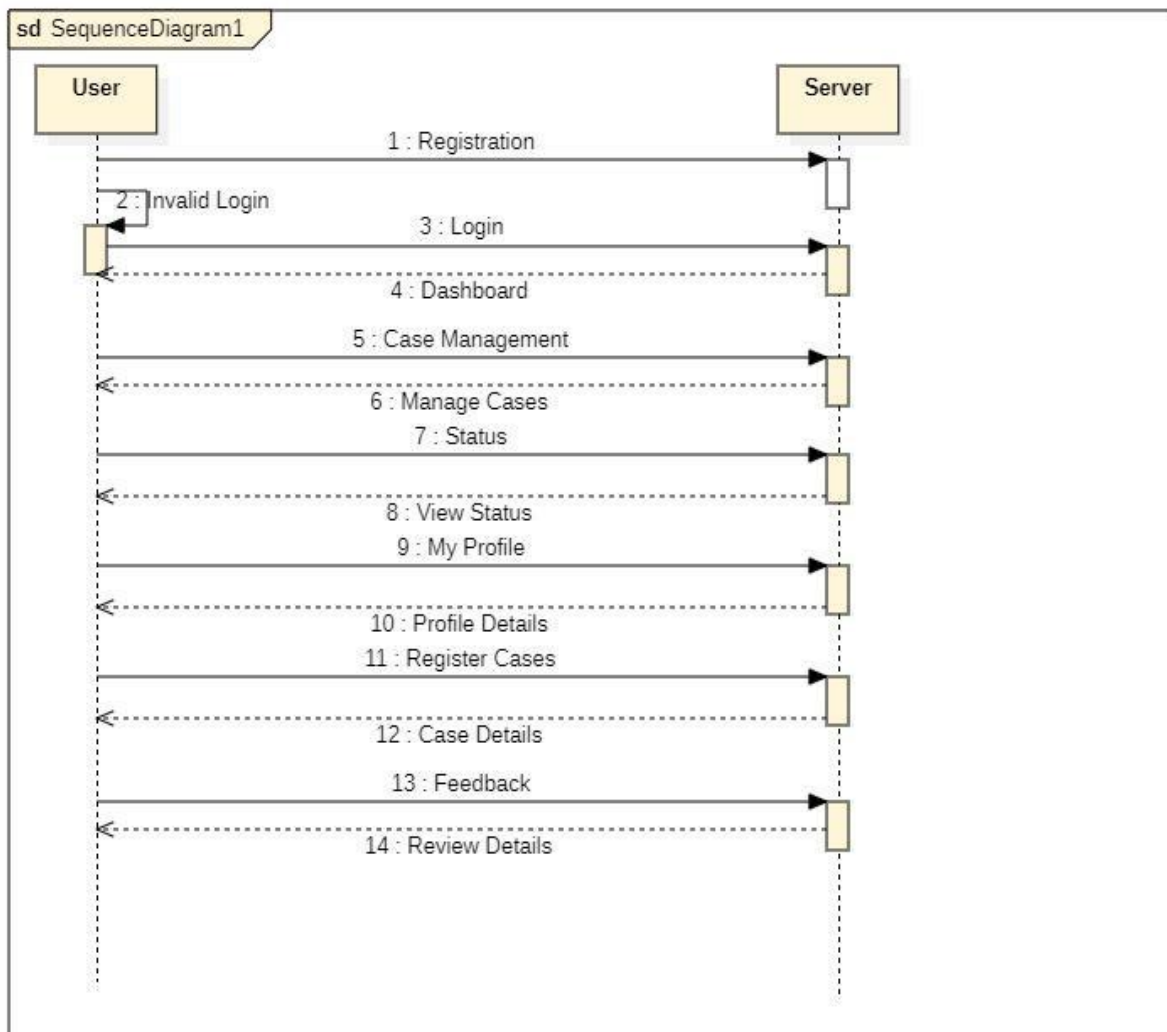


Figure 4.2.4 Sequence Diagram of User

4.2.3 Deployment Diagram:

A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

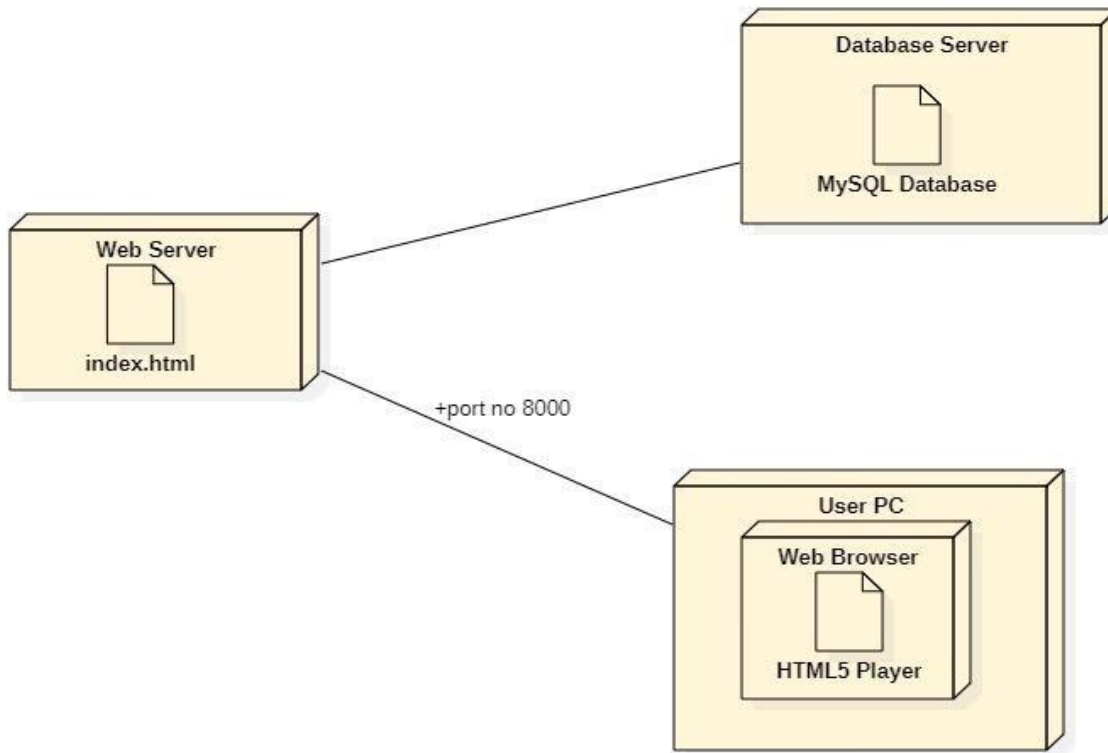


Figure 4.2.5 Deployment Diagram

4.2.4 Activity Diagram:

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent

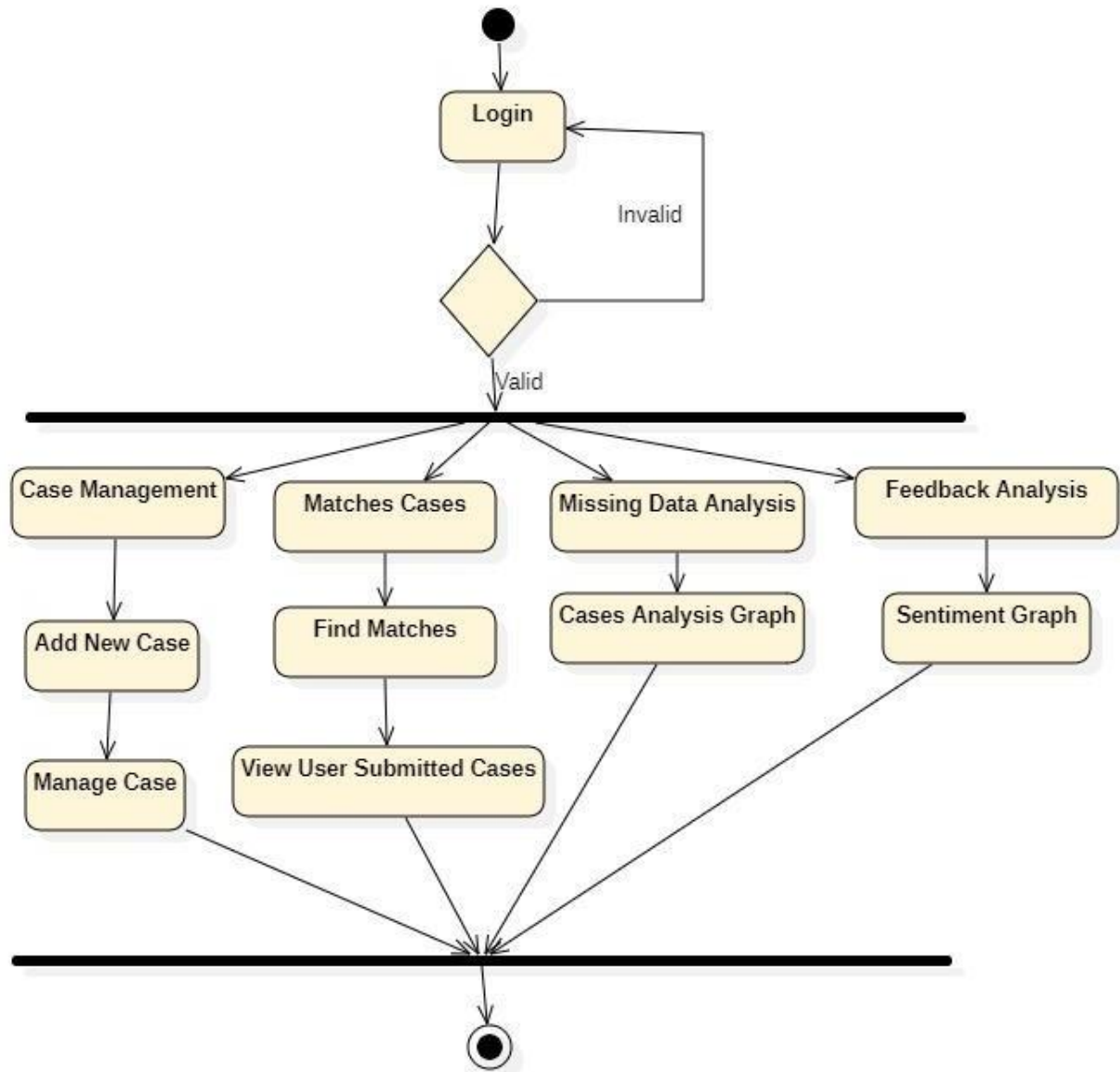


Figure 4.2.6 Activity Diagram of Police

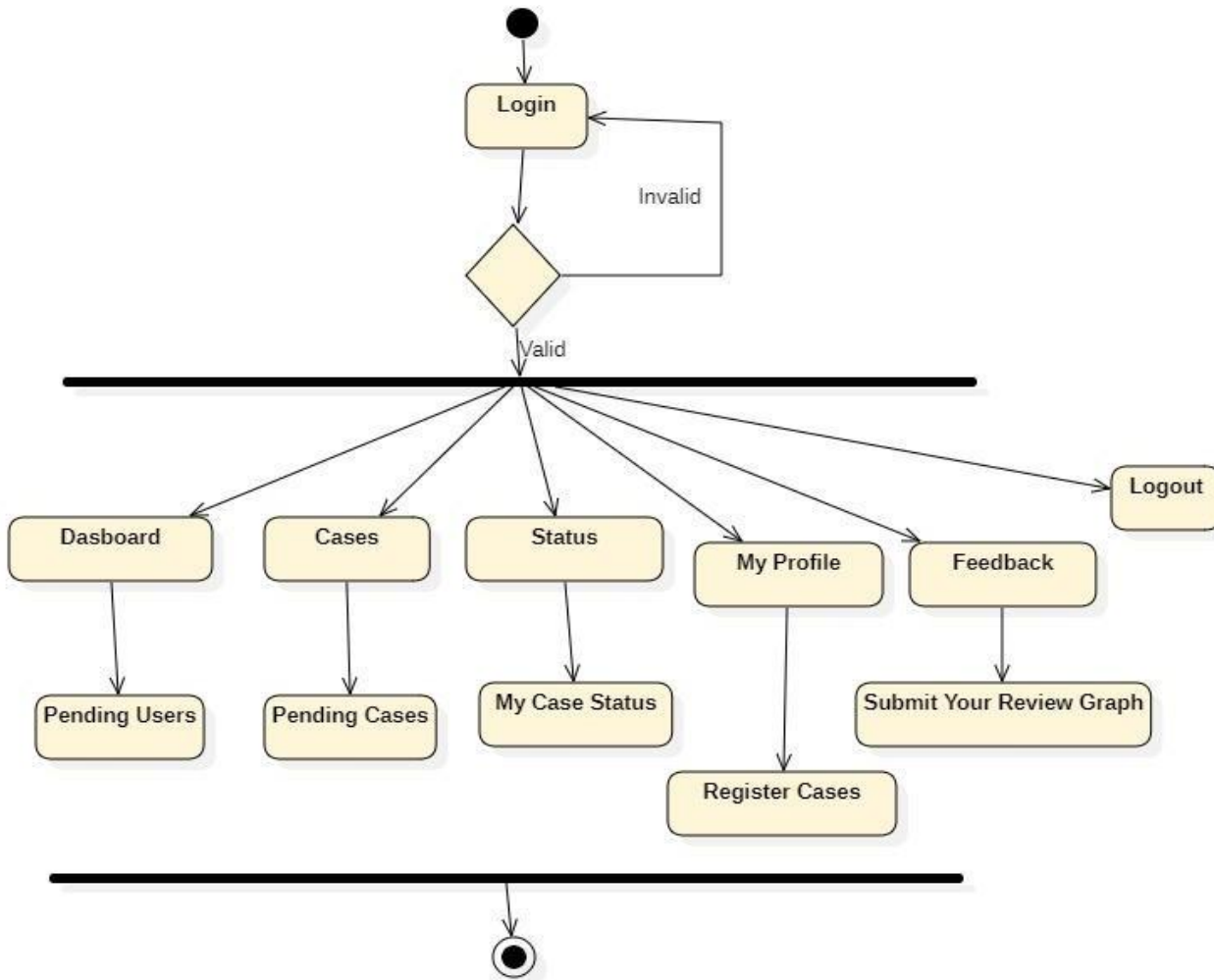


Figure 4.2.7 Activity Diagram of User

4.2.5 Data Flow Diagram:

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated

processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.



Figure 4.2.8 Data Flow Diagram

CHAPTER -5

SYSTEM IMPLEMENTATION

5.SYSTEM IMPLEMENTATION

5.1 MODULES

The "AI-Assisted Search for Identifying Missing Persons" system consists of several modules that work together to achieve its objectives. Here are five key modules:

➤ **User Interface Module:**

This module encompasses both the GUI application for law enforcement in Python and the Android application for public use. It provides an intuitive interface for users to interact with the system, enabling police to input case details and facilitating public users in submitting photos of suspicious individuals. The user interface module is crucial for seamless communication between the system and its users.

➤ **Image Processing Module:**

The image processing module is responsible for handling and enhancing the quality of the images submitted by both law enforcement and the public. It employs advanced algorithms to preprocess images, ensuring that the machine learning algorithm receives standardized inputs for accurate comparison and matching.

➤ **Database Management Module:**

This module manages the storage and retrieval of data related to missing children, including details provided by law enforcement, images, and information submitted by the public. It ensures the systematic organization of data for efficient querying and analysis, supporting the search for missing children.

➤ **Machine Learning Algorithm Module:**

The heart of the system, this module incorporates machine learning algorithms to compare and match user-submitted photos with those uploaded by law enforcement. The algorithm learns from the data it processes, improving its accuracy over time. This module plays a crucial role in identifying potential matches and assisting in locating missing children.

➤ **Privacy and Security Module:**

Given the sensitive nature of the information involved, the privacy and security module ensures the protection of user data. It incorporates encryption mechanisms, access controls, and anonymization features to safeguard the privacy of individuals submitting information. This module addresses ethical considerations and legal requirements associated with handling sensitive information.

- These modules work collaboratively to create a robust and integrated system for the AI-assisted search for missing children, combining user interaction, data processing, machine learning, and privacy considerations.

5.2. SYSTEM REQUIREMENTS

5.2.1 Hardware Requirements

MINIMUM (Required for Execution)		MY SYSTEM (Development)
System	Pentium IV 2.2 GHz	i3 Processor 5 th Gen
Hard Disk	20 Gb	500 Gb
Ram	1 Gb	4 Gb

5.2.2 Software Requirements

Operating System	Windows 10/11
Development Software	Python 3.10
Programming Language	Python
Domain	Machine Learning
Integrated Development Environment (IDE)	Visual Studio Code

Front End Technologies	HTML5, CSS3, Java Script
Back End Technologies or Framework	Django
Database Language	SQL
Database (RDBMS)	MySQL
Database Software	WAMP or XAMPP Server
Web Server or Deployment Server	Django Application Development Server
Design/Modelling	Rational Rose

CHAPTER - 6

SYSTEM TEST

6.SYSTEM TEST

TESTING

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

6.1 TYPES OF TESTING

6.1.1 System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

6.1.2 Module Testing

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

6.1.3 Integration Testing

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

6.1.4 Acceptance Testing

When that user find no major problems with its accuracy, the system passes through a final acceptance test. This test confirms that the system meets the original goals, objectives and requirements established during analysis without actual execution which eliminates wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

6.2 TEST CASES -

Test Case ID	Prerequisites	Test Steps	Input	Expected Output	Status (Pass/Fail)
TC_01	-	Admin Login with Correct Credentials	Admin credentials	Successful login	Pass
TC_02	-	Admin Login with Incorrect Credentials	Incorrect admin credentials	Sorry, invalid username and password message	Pass

TC_03	TC_01	Admin Enters Missing Person Details	Name,age, photo, address , etc.	Details successfully entered	Pass
TC_04	-	User Account Creation	User credentials	Account created successfully	Pass
TC_05	-	User Login with Correct Credentials	User credentials	Successful login	Pass
TC_06	-	User Login with Incorrect Credentials	Incorrect user credentials	Sorry, invalid username and password message	Pass
TC_08	TC_05	User uploads missing person details	Photo, description, etc.	Details successfully uploaded	Pass
TC_10	TC_03, TC_08	Admin Clicks on "Find Match"	User-uploaded image, Admin-uploaded image(same individual)	Match found	Pass
TC_11	TC_03, TC_08	Admin Clicks on "Find Match"	User-uploaded image, Admin-uploaded image(different individual)	Match not found	Pass
TC_12	-	User Provides Feedback	Feedback text	Feedback submitted successfully	Pass
TC_13	TC_12	Sentiment Analysis (Admin View)	-	Sentiment analysis of user feedback displayed	Pass
TC_14	TC_03, TC_08, TC_10, TC_11	Case Analysis Graph(Admin View)	-	Statistics displayed in bar chart (matches found, not found, cases added)	Pass

CHAPTER - 7

SCREENSHOTS

7.SCREENSHOTS

Click join us to use our website:

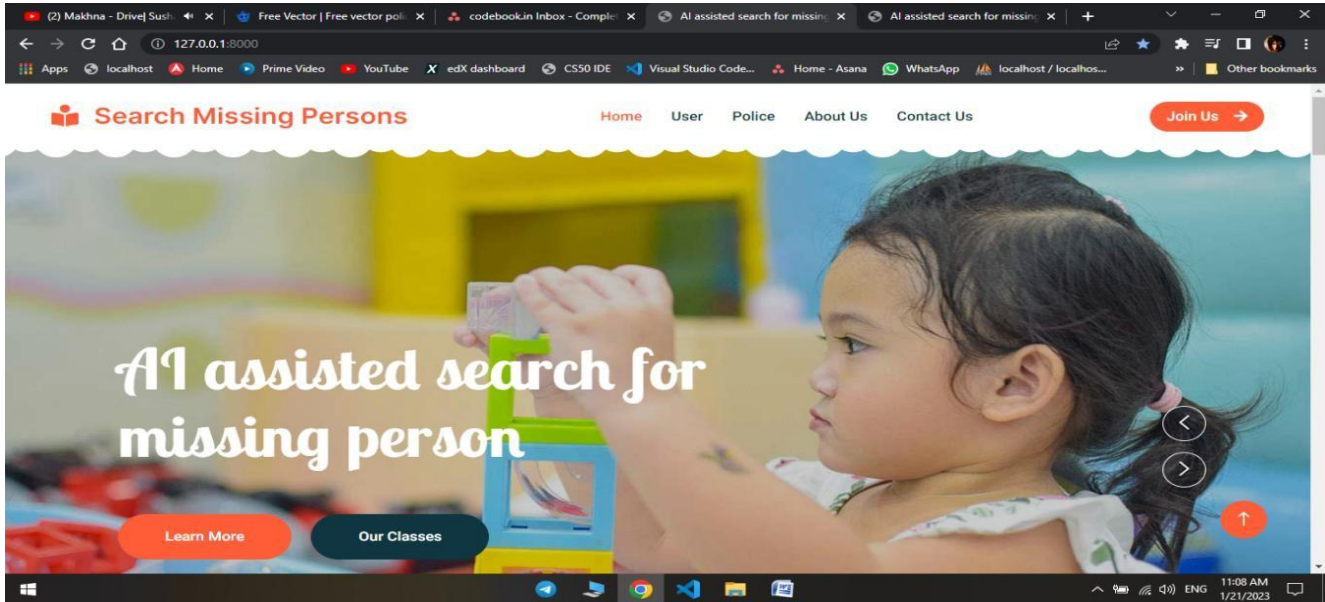


Figure 7.1 Home Page

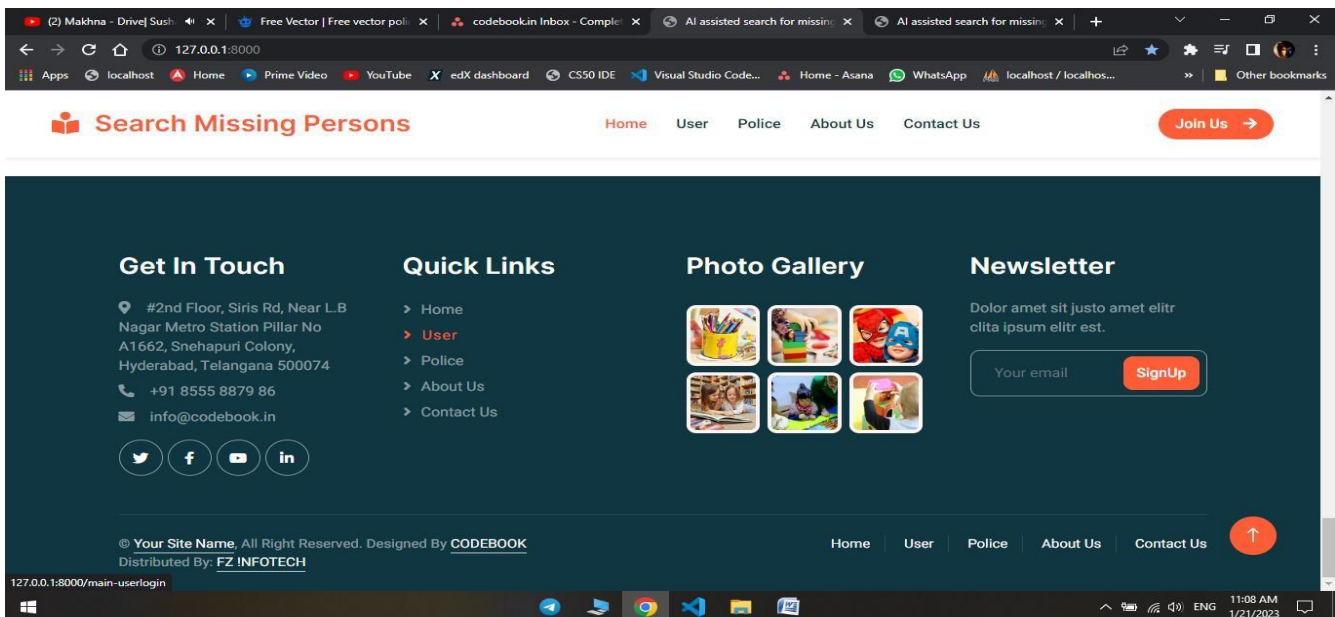


Figure 7.2 About Us

Click "about us" in the above screen and you'll get the details of our website as shown in the above image.

If you're already a user, please log in to our website using your username and password.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/main-userlogin`. The page has a header with the logo "Search Missing Persons" and navigation links: Home, User (active), Police, About Us, and Contact Us. A "Join Us" button is in the top right. The main heading is "USER LOGIN". Below it, there are two input fields: "Your Email" and "Your Password". A red "login" button is centered below the fields. At the bottom, it says "Not Registered ? Sign Up" with a red arrow pointing up. The browser's taskbar at the bottom shows the time as 11:08 AM on 1/21/2023.

Figure 7.3 User Login

If you are new to our website, we invite you to click on the 'Sign Up' button and provide the necessary information to create your account.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/main-userregistration`. The page has the same header as Figure 7.3. The main heading is "USER REGISTRATION". The form contains several fields: "Name" and "Father name" (top row); "Email Address" and "Phone no." (second row); "Photo" (with a "Choose File" button and "No file chosen" text) and "Date Of Birth" (with a date picker showing "mm/dd/yyyy") (third row); "City" and "Password" (bottom row). At the bottom, there is a "Gender" section with radio buttons for "Female" (selected), "Male", and "Other". A red arrow pointing up is in the bottom right corner. The browser's taskbar at the bottom shows the time as 11:09 AM on 1/21/2023.

Figure 7.4 User SignUp

To access the system as a police officer, simply click on the 'Police' option in Fig 7.1, which will prompt the appearance of the screen below.

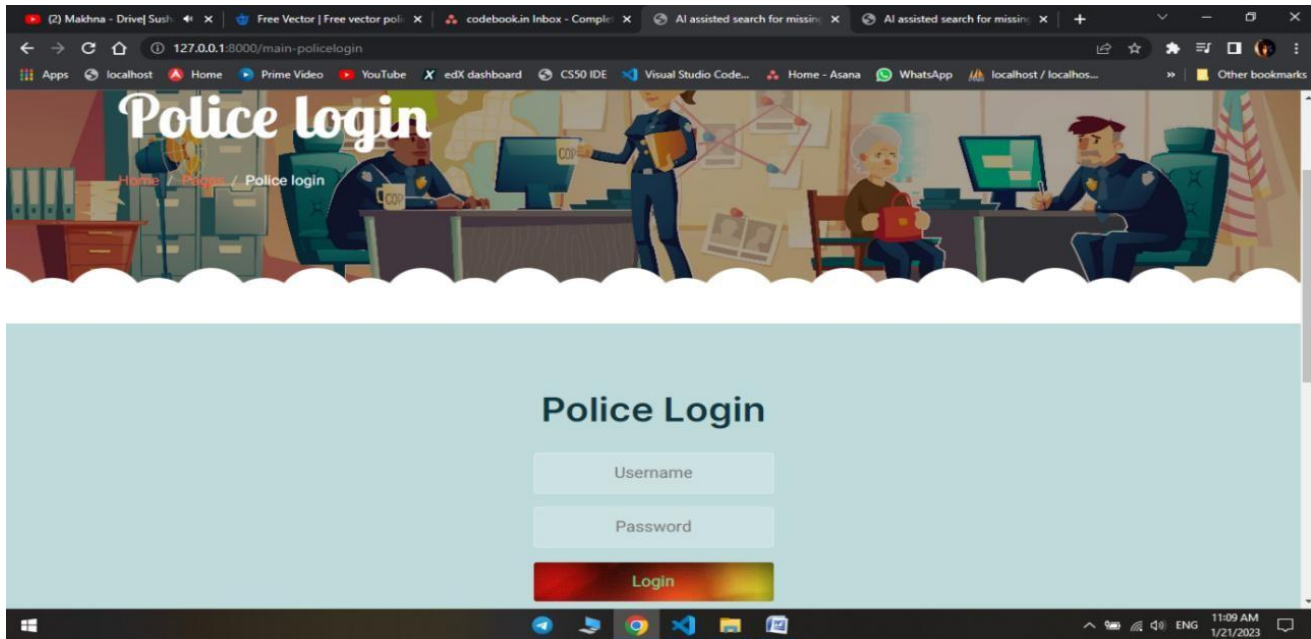


Figure 7.5 Police/Admin Login

After successfully logging in to our website, users will receive the message displayed on the following page.

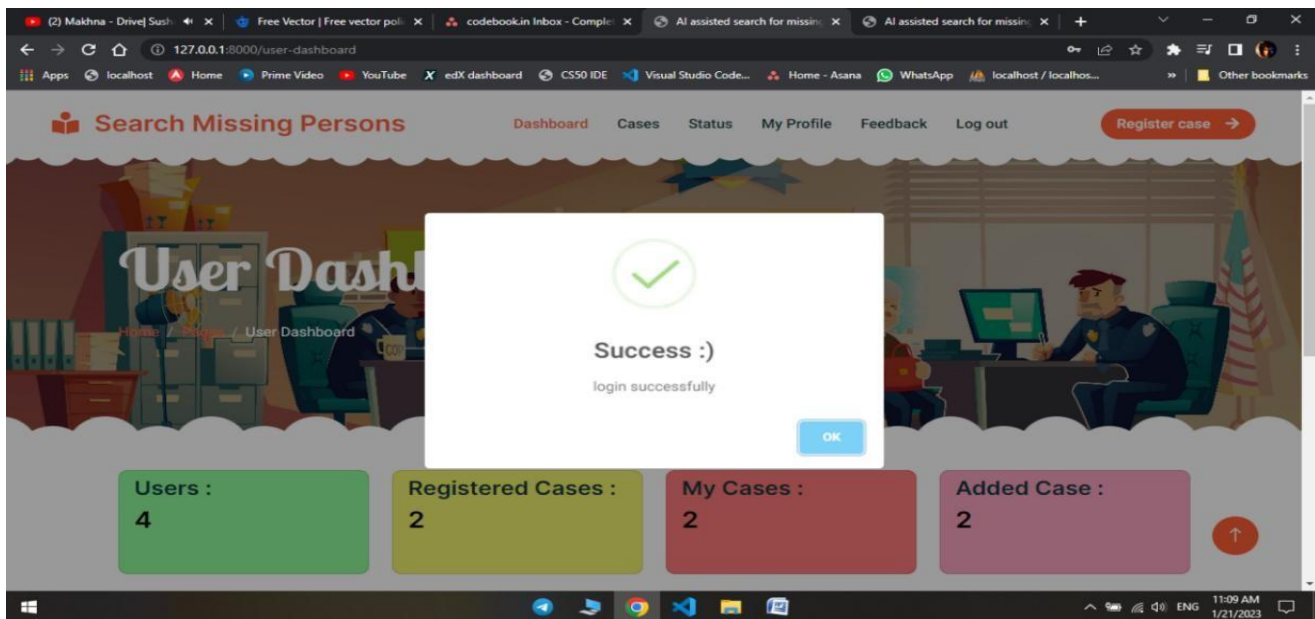


Figure 7.6 "Login Successful" Popup

The page below displays the user dashboard, providing statistical details such as registered cases, personal cases, added cases, and additional options.



Figure 7.7 User Dashboard

After clicking on the 'Cases' section on the previous page, users can view pending missing cases.

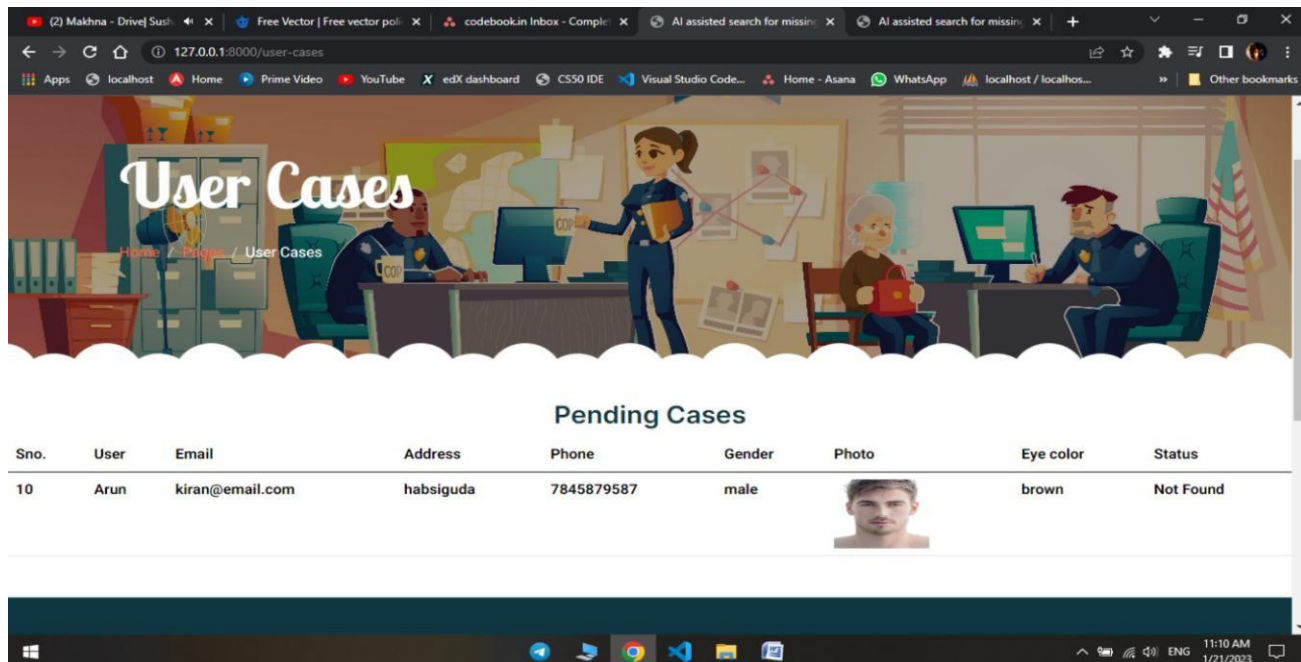


Figure 7.8 Pending Cases

To check the status of the cases which they've uploaded, users should select the 'Status' option in Fig. They'll see the following page which provides information about whether the missing persons have been located or not.

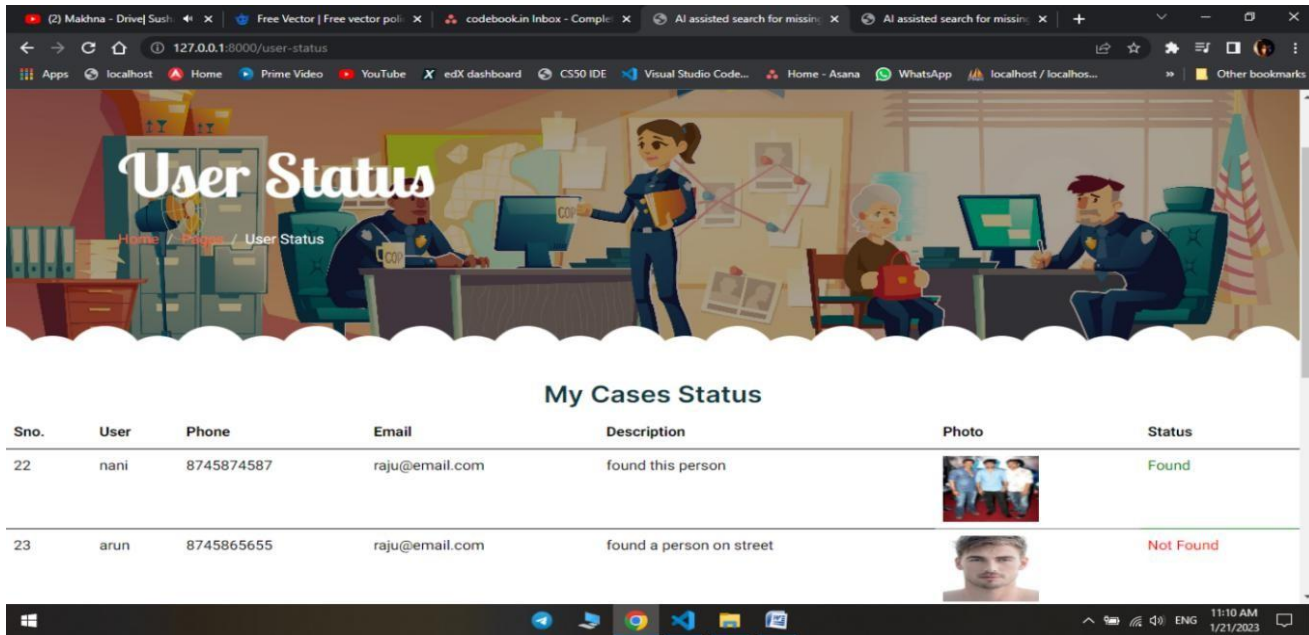


Figure 7.9 My Cases

To add or upload their profile details, users should click on the 'My Profile' option in Figure and proceed to enter their information.

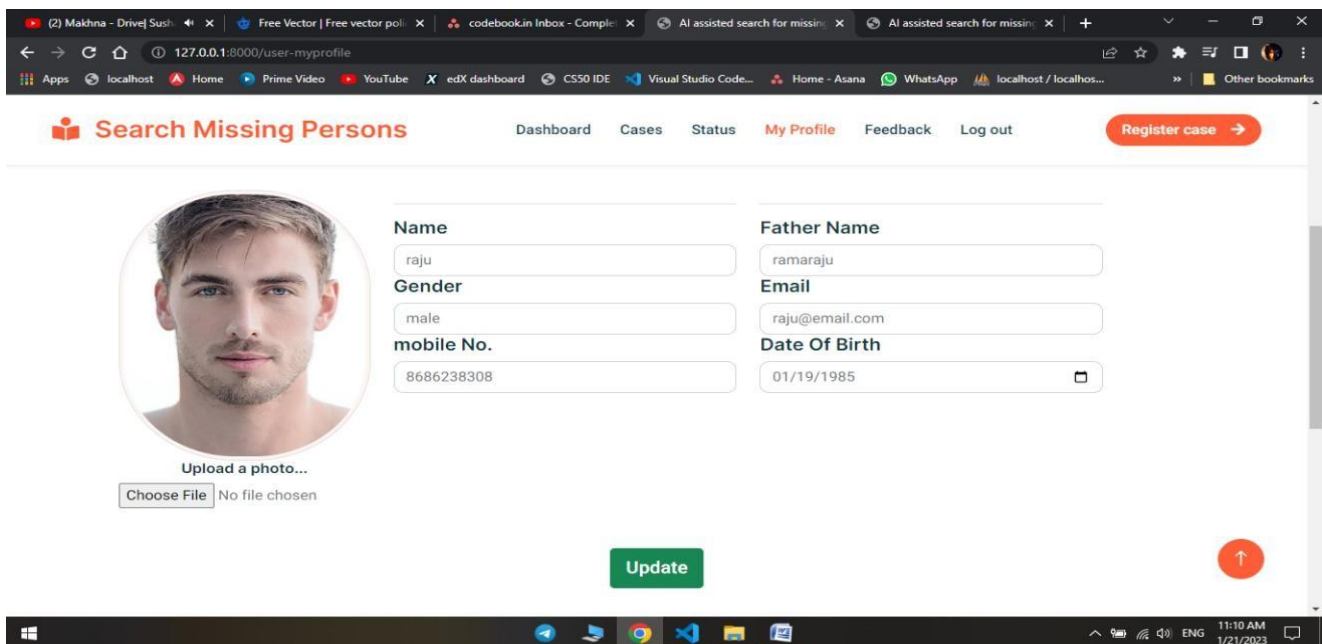


Figure 8.0 User Profile

Users can provide feedback about our website by clicking on 'Feedback' in Figure. The following page will appear, allowing them to rate their experience and provide comments based on their interaction with our website.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/user-feedback`. The page title is "Search Missing Persons". The navigation bar includes "Dashboard", "Cases", "Status", "My Profile", "Feedback", and "Log out". A "Register case" button is in the top right. The main content area is titled "Please rate our service experience for the following parameters". It contains two rating sections: "1. Your overall experience While Respond at time" and "2. Friendliness and courtesy shown to you while Case Registered to Us". Each section has five stars for rating. Below these is a section for "4. Any Other suggestions:" with a text input field. A "Submit your review" button is at the bottom. The Windows taskbar at the bottom shows the time as 11:11 AM on 1/21/2023.

Figure 8.1 User Feedback

"The 'Register Case' option will navigate users to the following page, where they can upload details about the missing person. In the description section, users have the option to provide information about the person or details about where they found that person.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/user-registercase`. The page title is "Search Missing Persons". The navigation bar is the same as in Figure 8.1. The main content area is titled "MISSING PERSON REGISTRATION". It contains a form with the following fields: "Full name", "Your Mobile no.", "Email address" (with placeholder text "enter your email"), "Description of Person" (with placeholder text "Message sent to the police"), and "Upload Image" (with a "Choose File" button and "No file chosen" text). A "Register case" button is in the top right. The Windows taskbar at the bottom shows the time as 11:11 AM on 1/21/2023.

Figure 8.2 Register a Case(User)

After successfully logging in to our website, police will receive the message displayed on the following page.

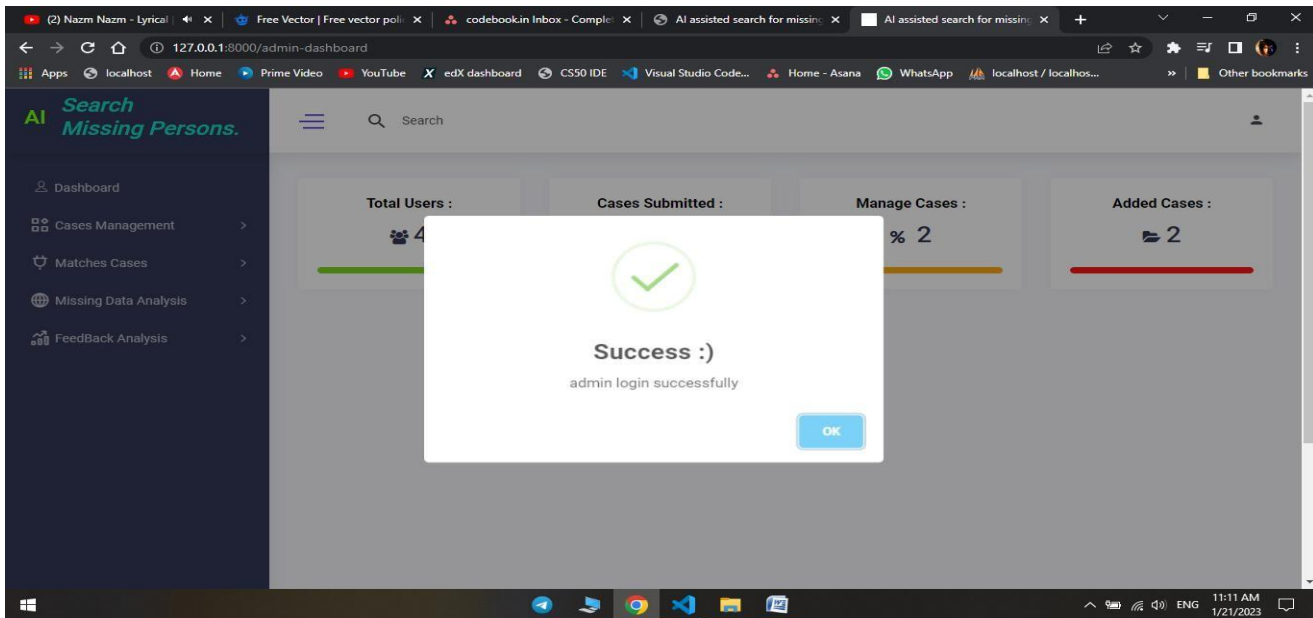


Figure 8.3 "Login Successful" Popup

The page below displays the police dashboard, providing statistical details such as total users, cases submitted, manage cases, and added cases.

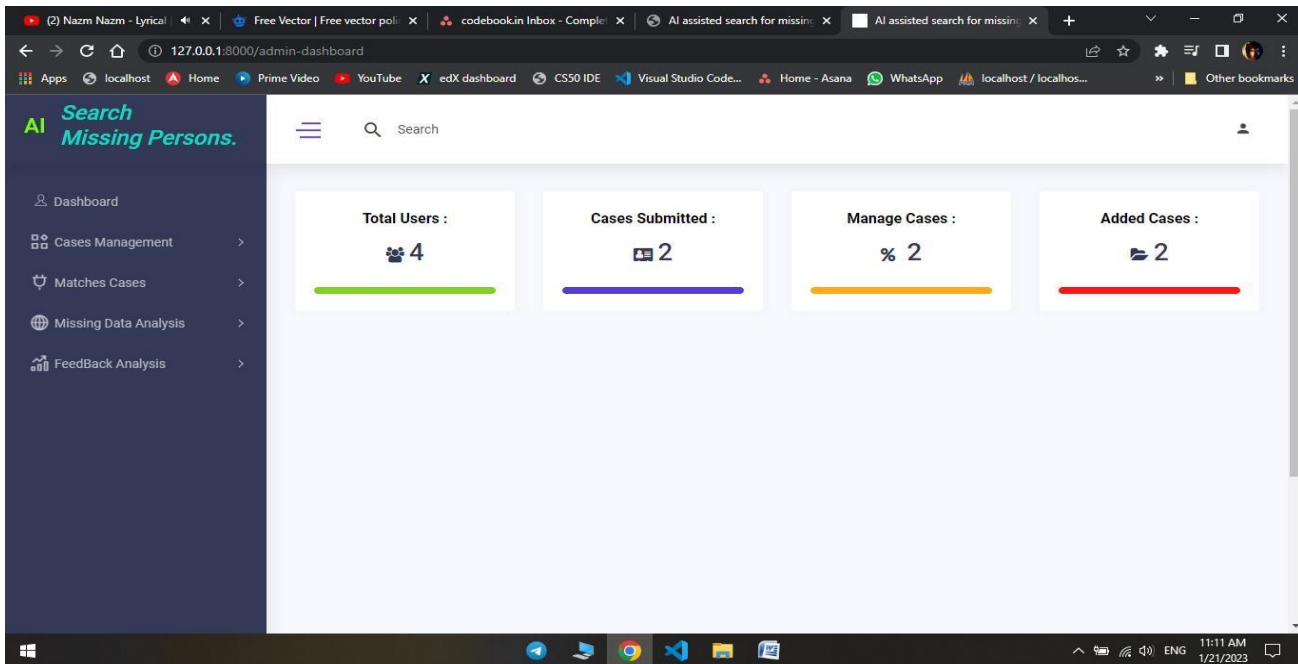


Figure 8.4 Police/Admin Dashboard

The following page displays the 'Add New Case' form, where the police should input details of the missing person exactly as received in the complaint. To upload the missing person's image, click on 'Choose File'.

ADD NEW CASE

Person Name:

Father Name:

Address:

Email Of Parent:

Phone No.:

Photo Of Person: No file chosen

Eye Colour:

Hair Colour:

Gender: ☒ Female ☐ Male ☐ Other

Date of Birth:

Figure 8.5 Add New Case(Police)

By clicking on 'Manage Cases,' the police can access all missing cases along with their details. The 'Edit' option allows them to update case details upon receiving new information.

Manage Cases



CHILD NAME	FATHER NAME	ADDRESS	EMAIL OF PARENT	PHONE NO.	PHOTO OF CHILD	EYE COLOUR	HAIR COLOUR	GENDER	DATE OF BIRTH	ACTION
nani	krishna	hyderabad	krishna@email.com	8143238308		Blue	Black	male	1985-01-19	<input type="button" value="Edit"/>
Arun	kiran	habsiguda	kiran@email.com	7845879587		brown	brown	male	1992-01-19	<input type="button" value="Edit"/>

Figure 8.6 Manage Cases

The following page displays the 'Find Match' option, where the police can check for any matches of the missing persons.

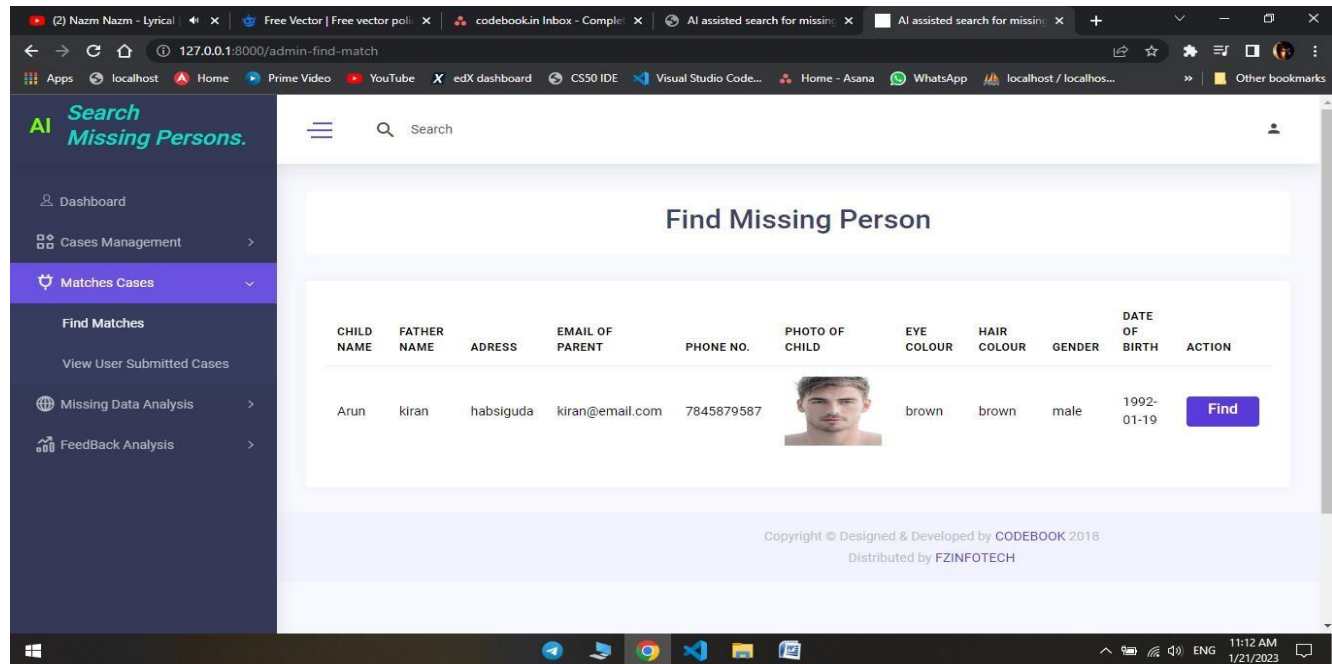


Figure 8.7 Find Match

By clicking on the 'User Submitted Cases' option, users are directed to the following page, displaying details of the person they have found along with the status of the match (found/not found).

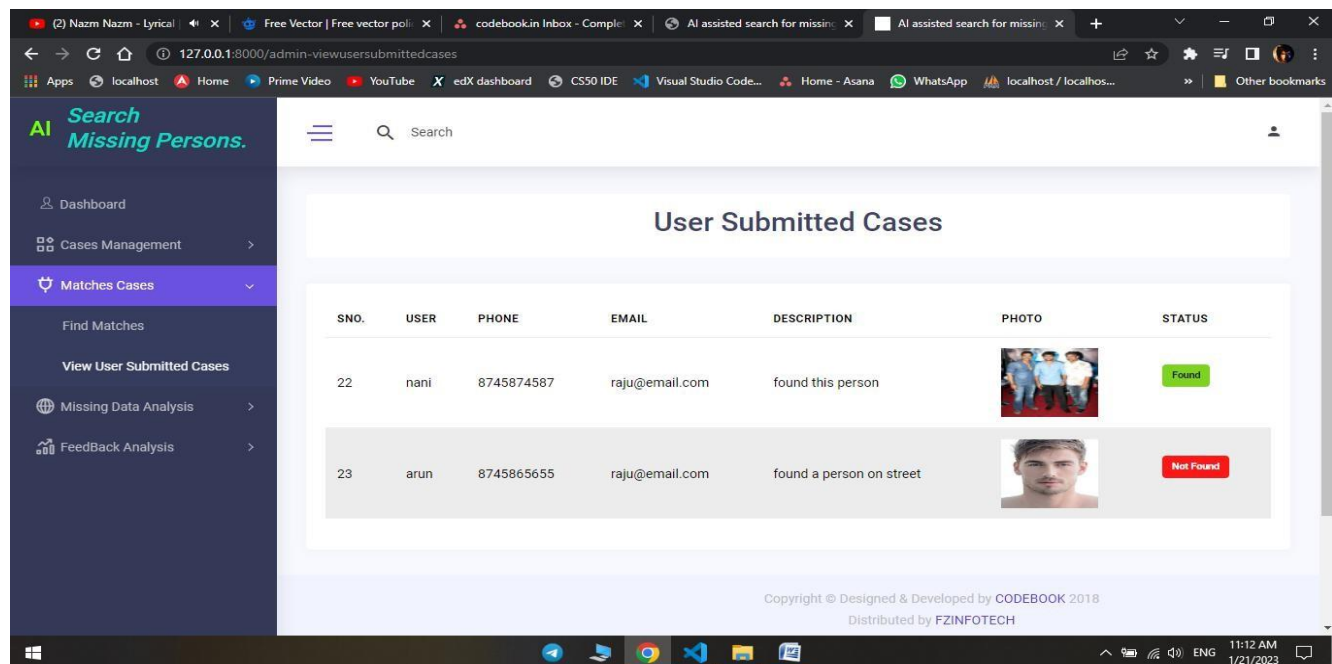


Figure 8.8 User Submitted Cases

The below image depicts the 'Case Analysis Graph,' where the police can view a graphical representation of the cases. The bar graph illustrates added cases, cases submitted, persons found, and those not found.

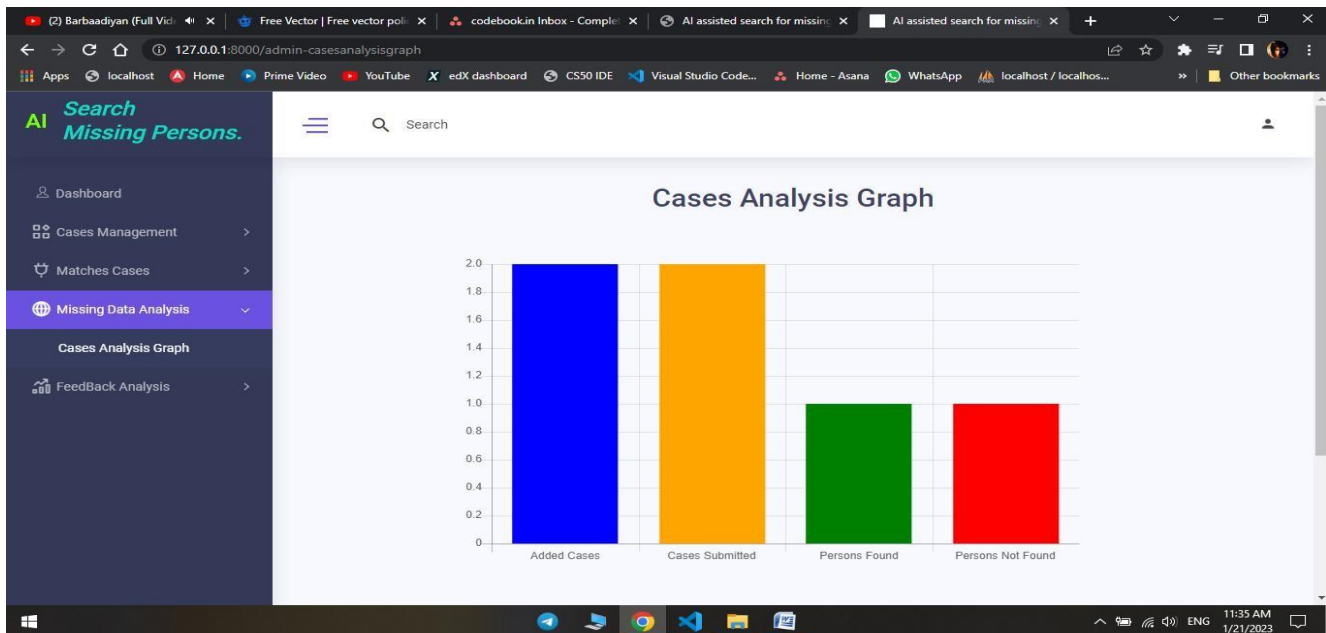


Figure 8.9 Case Analysis Graph

The following page displays the sentiment analysis, showing the sentiment of the users as neutral, positive, or negative, along with an emoji based on the users' responses.

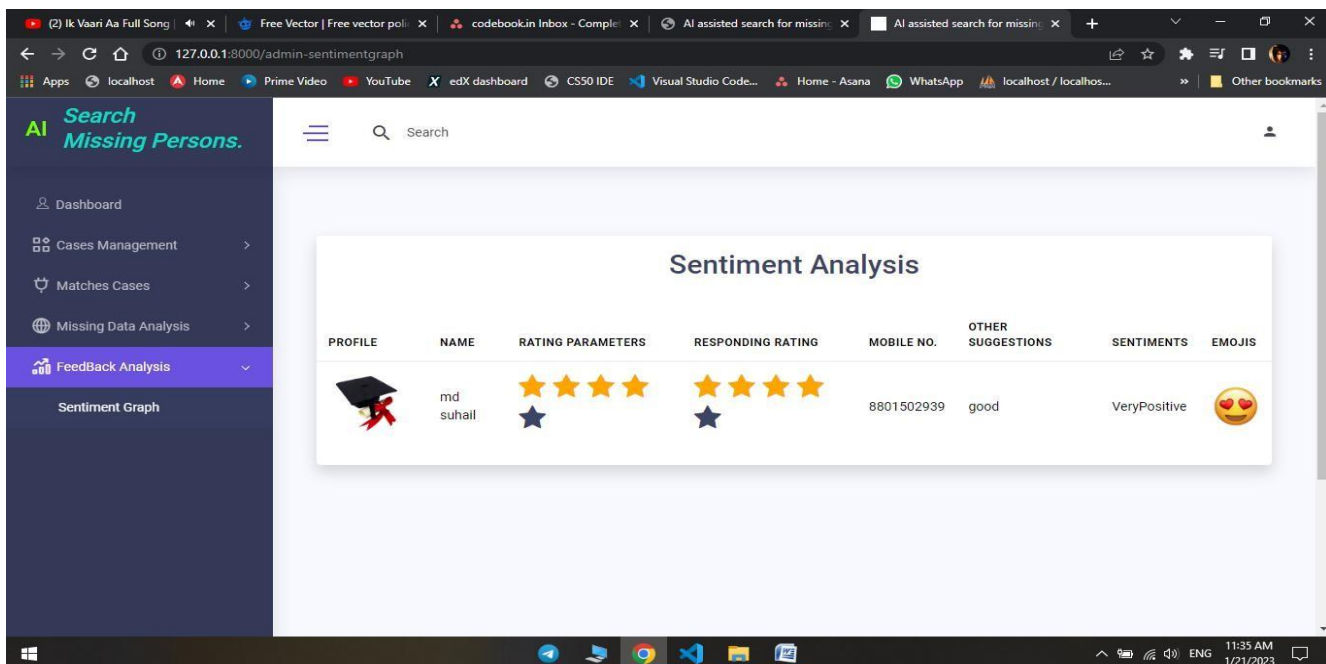


Figure 9.0 Sentiment Analysis

CHAPTER - 8

CONCLUSION AND FUTURE WORK

8. CONCLUSION AND FUTURE WORK

This system is a functioning illustration of an AI-Assisted Search for Missing Children intended to locate missing children. It has several facets and has a variety of useful capabilities, as it's explored in this paper. The primary goal was to make it easier to find and report missing children, which is successful.

When used wisely, this technology can be quite beneficial. Even in hotels, hospitals, and other public places, it may be utilized to quickly locate offenders. This application can be greatly enhanced by utilizing Flask to develop APIs. A fully functional web application can also be created that uses Tensorflow.

REFERENCES-

- [1]. Omkar M parkhi, andrea vedaldi, andrew zisserman, et al, "Deep Face Recognition," in BMVC, volume 1, page 6, 2015.
- [2]. Rohit Satle , Vishnuprasad Poojary , John Abraham , Mrs. Shilpa wakode, "Missing Child Identification Using Face Recognition System" vol.3, issue.1, July – August 2016.
- [3]. Sumeet Pate, "Robust face recognition system for e-crime alert", in International Journal for Research in Engineering Application and Management, Issue 1, MAR, 2016.
- [4]. Bharath Darshan Balar, D S Kavya, Chandana M, Anush E, Vishwanath R Hulipalled, "Efficient Face Recognition System for Identifying Lost People", International Journal of Engineering and Advanced Technology (IJEAT), volume-8, issue-5s, may 2019.
- [5]. Saurabh p.Bahurupi, D.S.Chaudhari,"Principal Component Analysis for Face Recognition," International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, volume-1, issue-5, June 2012.
- [6]. Hsin-rung Chou, Jia-hong Lee, Yi-ming Chan, And Chu-song Chen, "Data-specific Adaptive Threshold For Face Recognition And Authentication", arxiv.Org, 26 Oct 2018.
- [7]. Shefali Patil, Pratiksha Gaikar, Divya Kare, Sanjay Pawar, " Finding Missing Person Using AI", International Journal Of Progressive Research In Science And Engineering, Vol.2, NO.6, Jun 2021.
- [8]. Birari Hetal, Sanyashiv Rakesh, Porje Rohan, Salve Harish, "Android Based Application - Missing Person Finder", JUN 2018 | IRE Journals | Volume 1 Issue 12 | ISSN: 2456-8880.
- [9]. Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, Le Song, "SphereFace: Deep Hypersphere Embedding for Face Recognition",arXiv:1704.08063, volume 4, Jan 2018.
- [10]. Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snaveley, Kavita Bala, Kilian Weinberger, "Deep Feature Interpolation for Image Content Changes",arXiv:1611.05507, vol 2, Jun 2017.

APPENDIX-A

SOURCE CODE

View.py:

```
# Importing necessary modules and models

from django.shortcuts import render, redirect

from userapp.models import *

from django.contrib import messages

from adminapp.models import *

from textblob import TextBlob


# Main user registration view

def main_userregistration(request):

    if request.method == "POST" and request.FILES['photo']:

        # Extracting data from the request

        name = request.POST.get("name")

        father = request.POST.get("father")

        email = request.POST.get("email")

        phone = request.POST.get("phone")

        gender = request.POST.get("gender")
```

```
dob = request.POST.get("dob")

city = request.POST.get("city")

password = request.POST.get("password")

photo = request.FILES['photo']

try:

    # Checking if the user with the provided email already exists

    usermodel.objects.get(email=email)

    messages.info(request, "email already existed")

    return redirect('main_userregistration')

except:

    # Creating a new user

    usermodel.objects.create(

        name=name,

        father=father,

        email=email,

        phone=phone,

        gender=gender,

        dob=dob,

        city=city,

        password=password,
```

```
        photo=photo,

    )

    messages.success(request, "You Have Registered Sucessfully")

    return redirect('main_userlogin')

    return render(request, "main/main-userregistration.html")
```

Main user login view

```
def main_userlogin(request):

    if request.method == "POST":

        email = request.POST.get("email")

        password = request.POST.get("password")

        try:

            # Attempting to authenticate user

            user_registrations = usermodel.objects.get(email=email, password=password)

            request.session['user_id'] = user_registrations.user_id

            messages.success(request, "login successfully")

            return redirect("user_dashboard")

        except:

            messages.error(request, "invalid username and passsword")

            return redirect("main_userlogin")
```

```
return render(request, "main/main-userlogin.html")
```

```
# User dashboard view
```

```
def user_dashboard(request):
```

```
    user1 = usermodel.objects.all().count()
```

```
    user2 = registercasemodel.objects.all().count()
```

```
    user3 = addcasemodel.objects.all().count()
```

```
    case_id = request.session['user_id']
```

```
    user = usermodel.objects.get(user_id=case_id)
```

```
    cases = registercasemodel.objects.filter(user_case=user.user_id).count()
```

```
    return render(request, "user/userdashboard.html", {"user1": user1, "user2": user2, "user3": user3, "cases":  
cases})
```

```
# User register case view
```

```
def user_registercase(request):
```

```
    case_id = request.session['user_id']
```

```
    user = usermodel.objects.get(user_id=case_id)
```

```
    if request.method == "POST":
```

```
name = request.POST.get("name")

mobile = request.POST.get("moblie")

email = request.POST.get("email")

description = request.POST.get("description")

photo = request.FILES["photo"]


registercasemodel.objects.create( name=

    name,

    mobile=mobile,

    email=email,

    description=description,

    photo=photo,

    user_case=user

)

print(name, "tiger")

messages.success(request, "Case has been register Successfully")

return render(request, "user/user-registercase.html")
```

User cases view

```
def user_cases(request):
```

```
cases = addcasemodel.objects.filter(status='Not Found')

return render(request, "user/user-cases.html", {'user': cases})


# User feedback view

def user_feedback(request):

    user_id = request.session['user_id']

    user = usermodel.objects.get(pk=user_id)

    if request.method == "POST":

        overall = request.POST.get('rating1')

        caseregister = request.POST.get('rating2')

        suggestion = request.POST.get('commentText')

        analysis = TextBlob(suggestion)

        if not overall:

            messages.info(request, "Please give the overall ratings")

            return redirect(user_feedback)

        if not caseregister:

            messages.info(request, "Please give the caseregister ratings")

            return redirect(user_feedback)

    sentiment = "
```

```
if analysis.polarity >= 0.5:

    sentiment = 'VeryPositive'

elif 0 < analysis.polarity < 0.5:

    sentiment = 'Positive'

elif -0.5 <= analysis.polarity < 0:

    sentiment = 'Negative'

elif analysis.polarity <= -0.5:

    sentiment = 'VeryNegative'

else:

    sentiment = 'Neutral'

try:

    userfeedback.objects.get(feedback2_id=user)

    messages.info(request, "Feedback has already sended")

    return redirect('user_feedback')

except:

    userfeedback.objects.create(overall=overall, caseregister=caseregister, suggestion=suggestion,

                                feedback2=user, sentiment=sentiment)

    messages.success(request, "Feedback has been send successfully")

return render(request, "user/user-feedback.html", {'np': user})
```

User profile view

```
def user_myprofile(request):

    id = request.session['user_id']

    user = usermodel.objects.get(user_id=id)

    if request.method == "POST":

        name = request.POST.get('name')

        father = request.POST.get('father')

        gender = request.POST.get('gender')

        email = request.POST.get('email')

        mobile = request.POST.get('mobile')

        dob = request.POST.get('dob')


    if not request.FILES.get('photo', False):

        user.name = name

        user.father = father

        user.email = email

        user.gender = gender

        user.phone = mobile

        user.dob = dob
```

```
if request.FILES.get('photo', False):

    image = request.FILES['photo']

    user.name = name

    user.father = father

    user.email = email

    user.gender = gender

    user.phone = mobile

    user.dob = dob

    user.photo = image

user.save()

messages.success(request, "profile updated sucessfully")

return redirect('user_myprofile')

return render(request, "user/user-myprofile.html", {"user": user})
```

```
# User status view
```

```
def user_status(request):

    case_id = request.session['user_id']

    user = usermodel.objects.get(user_id=case_id)

    cases = registercasemodel.objects.filter(user_case=user.user_id).order_by('status')

    return render(request, "user/user-status.html", {"user": cases})
```