IJPP

Global Journal of Sociology and Anthropology
Vol. 15 (1), pp. 001-012, January, 2026
ISSN: 2756-3456

Full Length Article

AI Assisted Search For Missing Children

Srihari[1], Dr. Md.Asif[2]
B.Tech Student, Department Of Electronics and Computer Engineering, J.B Institute of Engineering and Technology, Hyderabad, India[1]
Associate Professor, Department Of Electronics and Computer Engineering, J.B Institute of Engineering and Technology, Hyderabad, India[2]

hotelsrihari1806@gmail.com, asif.ecm@jbiet.edu.in

Article Received 28-12-2025, Article Revised 9-1-2026, Article Accepted 22-01-2026
Authors Retains the Copyrights of This Article

Abstract:
This project develops an AI-Assisted Search For Missing Children. It addresses the limitations of manual photo comparison by using computer vision technology for rapid identification. The system processes facial features through Python and OpenCV, comparing uploaded images against a missing children database. A web interface allows easy photo submission and searching via webcam or file upload. The AI engine provides instant matching results with accuracy scores, significantly reducing search time and human error. This solution offers law enforcement and families an efficient tool to accelerate reunification efforts, operating continuously with consistent precision in critical missing person cases.

Keywords: *AI, CNN.*

Introduction

The Missing Child Identification Portal is an advanced, web-based application specifically designed to combat the growing issue of child disappearances by utilizing state-of-the-art facial recognition and deep learning technologies. At its core, the portal employs a Convolutional Neural Network (CNN) architecture, seamlessly integrated with Python's DeepFace library, to carry out high-precision facial verification and identification. The system allows users— ranging from concerned family members and the general public to law enforcement officials— to register missing child reports by entering vital details and uploading photographs. These uploaded images are immediately analyzed and compared against an existing and continuously expanding database of missing children's records to detect facial similarities and generate potential matches. The platform's matching algorithm considers multiple biometric parameters such as facial landmarks, skin texture, and geometric distances between facial features, enhancing the accuracy of identification even in cases of slight age progression or image quality variations.

The portal features an intelligent, real-time Automated Email Notification system that dispatches alerts to all registered users, NGOs, and relevant law enforcement bodies whenever a new case is submitted, thus fostering collective vigilance and public participation. The backend architecture is developed using Node.js and Express.js, ensuring high-speed API response times and efficient handling of concurrent user requests. MySQL serves as the secure and structured database for storing user profiles, case reports, and image data, with role-based access controls to ensure data integrity and privacy. All uploaded content is encrypted both at rest and in transit, in compliance with modern cybersecurity standards and data protection regulations.

Moreover, the portal includes an intuitive and responsive front-end interface that allows for easy navigation, image uploading, search filtering, and real-time results display, making it accessible to users with varying technical proficiencies. Plans for future scalability include the addition of a dedicated mobile application with push notification support, AI-driven facial aging models to account for time lapsed since disappearance, geo-tagged case mapping for region-specific alerting, and multilingual support to reach broader demographics. Furthermore, integrations with external government and police databases, as well as APIs for inter-portal data exchange, are envisioned to create a cohesive national and international missing child identification network. By uniting modern AI technologies with proactive community engagement, the portal offers a transformative approach to accelerating search efforts and reuniting families, serving as a critical tool in addressing one of society's most distressing Shumanitarian concerns.

Problem Statements

The disappearance of children remains a critical and

deeply distressing issue globally, with thousands of cases reported every year. Despite the growing adoption of digital tools in various sectors, the process of locating and identifying missing children continues to face numerous challenges—ranging from delayed reporting and limited access to centralized databases to lack of real-time communication among authorities and the general public. Traditional manual identification methods are slow, error-prone, and heavily dependent on human effort and memory, which significantly reduces the chances of timely recovery. Additionally, existing missing child databases are often fragmented across different jurisdictions, lacking interoperability and comprehensive data integration.

One of the most pressing concerns is the absence of an intelligent, automated system that can rapidly and accurately match images of missing children against large-scale datasets. The variation in image quality, age progression, lighting conditions, and facial expressions further complicates manual comparison and identification. Moreover, many families and individuals are unaware of proper channels or platforms where they can report such incidents quickly and effectively, leading to valuable time lost during the critical early stages of disappearance.

There is also a noticeable lack of public engagement and real-time alert systems that can harness the collective power of communities in the search for missing children. Notifications are rarely broadcasted to wider audiences in a timely manner, and follow-up processes remain manual and inconsistent. Furthermore, current systems often do not prioritize data security, leading to potential privacy violations and unauthorized access to sensitive personal information.

From a technical perspective, many available platforms lack the robust infrastructure to handle high volumes of user requests, store sensitive images and case data securely, and integrate seamlessly with third-party systems such as law enforcement databases or non-governmental organizations (NGOs) dedicated to child protection. The need for multilingual support, mobile accessibility, and AI-powered enhancements such as facial aging and location-based alerting is largely unmet in today's solutions.

In light of these challenges, there is an urgent demand for a comprehensive, intelligent, and scalable solution that can automate and accelerate the process of identifying missing children. The proposed Missing Child Identification Portal aims to address these problems by combining the power of Convolutional Neural Networks (CNN), DeepFace facial recognition, and real-time community engagement through a secure, user-friendly, and scalable web platform. This system is designed not only to improve the efficiency and accuracy of identification but also to create a nationwide collaborative network that significantly boosts the chances of reuniting children with their families.

System Requirements

This section outlines the necessary hardware and software specifications required for the successful deployment, execution, and maintenance of *The Adaptive Real-Time Network Packet Analysis: A CNN-Based Anomaly Detection System*. The system is designed to leverage deep learning techniques and real-time packet monitoring for the detection of network anomalies, and as such, it demands a robust computational environment for both model training and inference.

The requirements specified here ensure optimal system performance, efficient resource utilization, and seamless integration with cloud and local environments. These requirements are categorized into **Software Requirements** and **Hardware Requirements**, detailing the essential components needed to support the application's architecture, development, and deployment pipelines.

In addition to core hardware and software specifications, the system must accommodate high-throughput data streaming and low-latency processing to maintain real-time responsiveness. Scalable infrastructure is critical, allowing the system to adapt to increasing network traffic volumes and extended deployments across distributed environments. The system should also support containerized environments (e.g., Docker) for consistent deployment and microservice architecture if modular expansion is planned.

For cloud integration, compatibility with services such as AWS S3, EC2, and Lambda functions can enhance operational agility and fault tolerance. Proper security configurations, such as TLS encryption, access control, and user authentication protocols, must be in place to safeguard data during transmission and storage. Logging, monitoring, and alerting mechanisms are also recommended for performance auditing and anomaly reporting.

This comprehensive specification ensures that the system remains reliable, scalable, and secure under diverse real-world conditions and evolving network infrastructures.

Software and Hardware Requirement Software Requirements:

Component	Specification
Operating System	Windows 10 / Ubuntu 18.04 or higher

Global Journal of Sociology and Anthropology
Vol. 15 (1), pp. 001-012, January, 2026
ISSN: 2756-3456

Programming Languages	Python 3.8+ (AI & recognition), Node.js (backend services), MySQL (database)
Database Tool	MySQL Workbench (for database schema design and query management)
Cloud & Storage	Local Storage / AWS (for scalable storage & compute) / Heroku (for deployment)
Python Libraries	cv2 (OpenCV), deepface, os, sys, warnings
Machine Learning Libraries	deepface (using Facenet512 model and cosine similarity metric), TensorFlow backend
Node.js Libraries	express, mysql2, bcryptjs, body-parser, multer, nodemailer, dotenv, express-session
Development Environments	Jupyter Notebook / VS Code / PyCharm
Face Detection & Recognition	DeepFace (CNN-based model: Facenet512)
Email Service	Nodemailer (Gmail SMTP) for sending user notifications

Hardware Requirements:

Component	**Specification**
Processor	Intel Core i7/i9 or AMD Ryzen 7/9 (or higher)
RAM	Minimum 16GB (32GB recommended for AI model training)
Storage	500GB SSD (1TB+ preferred for large media files)
GPU	NVIDIA RTX 3060 or higher (for AI processing and video rendering)
Internet	High-speed internet required for cloud-based processing and data retrieval

Global Journal of Sociology and Anthropology
Vol. 15 (1), pp. 001-012, January, 2026
ISSN: 2756-3456

System Design

The system design of the **AI-Assisted Missing Child Identification Portal** serves as the blueprint for the overall architecture, defining how various components interact and function cohesively to achieve the core objective: the efficient and accurate identification of missing children using deep learning and facial recognition techniques.

This design phase outlines both high-level architecture and detailed module-level implementations, ensuring the system is scalable, modular, secure, and maintainable. It incorporates the use of a web-based front-end interface, a robust back-end using Node.js, MySQL for data management, and a Python-based CNN facial recognition engine powered by DeepFace for real-time image matching.

The system design also considers essential aspects such as user authentication, secure image handling, email notifications, and real-time processing for rapid response. It integrates with cloud services for scalability and offers support for local deployment for offline use by authorities. The design promotes smooth user interaction while maintaining the accuracy and integrity of facial verification to support law enforcement in locating missing children effectively.

Architecture Diagram

The system uses a three-tier architecture: a web-based **UI** for user interaction, a **Node.js backend** for processing and routing, and a **MySQL database** for data storage. A **Python-based DeepFace model** performs facial recognition to match uploaded images with the database of missing children.

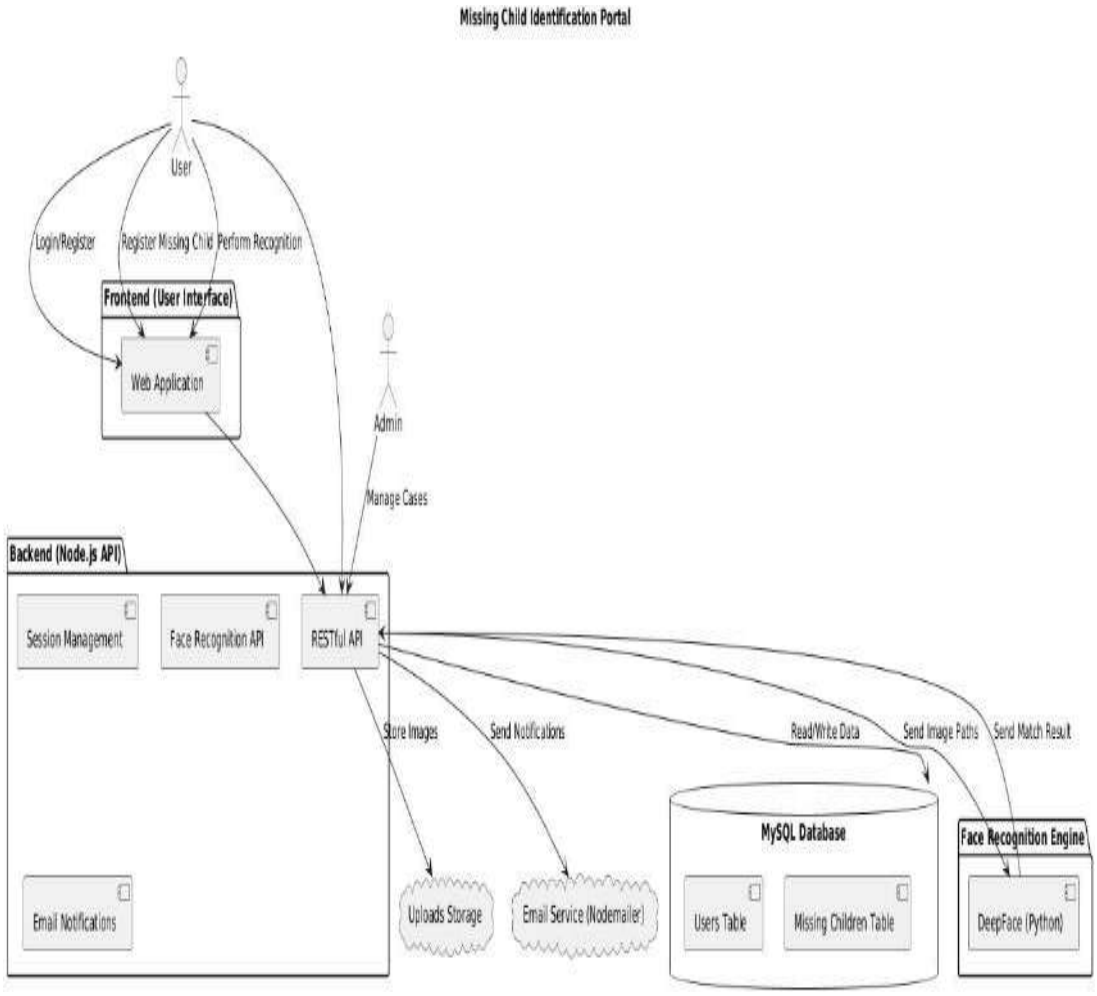


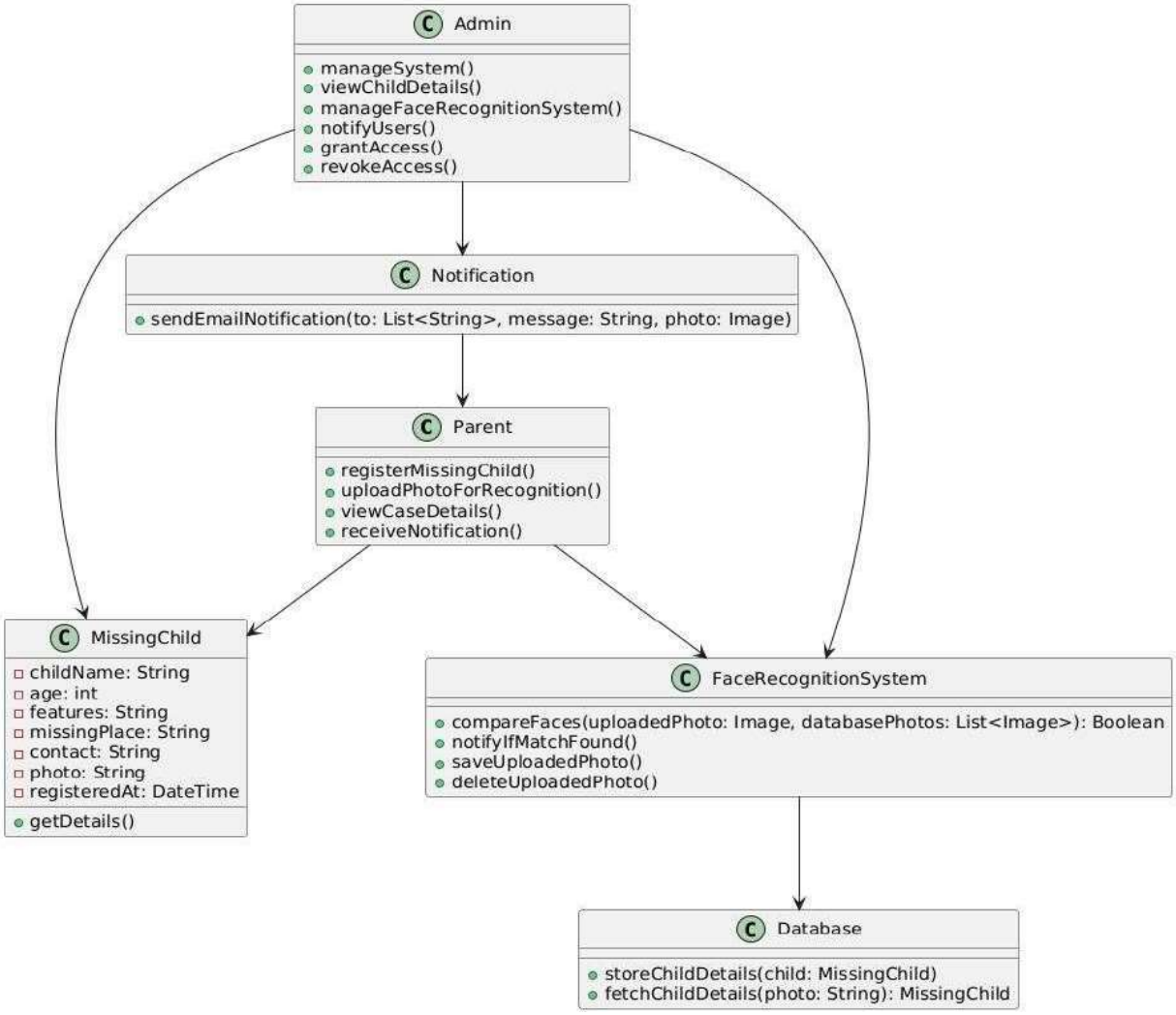
Figure 1: Architecture Diagram

Global Journal of Sociology and Anthropology
Vol. 15 (1), pp. 001-012, January, 2026
ISSN: 2756-3456

UML DIAGRAMS / DFD

Class Diagram

The class diagram represents the key entities in the system, such as User, MissingChild, and ImageHandler. It outlines their attributes and methods, and shows relationships like how a user can register a child or upload images for recognition. This helps in understanding the system's structure and interactions.



Implementation

Algorithms

Convolutional Neural Network (CNN)

- Purpose: Extract and learn facial features from images.
- Use: Forms the backbone of facial recognition by capturing spatial hierarchies in facial patterns.
- In This Project: Implemented through DeepFace, which internally uses CNN-based models like Facenet512.

2. ***Facenet512 (DeepFace Pre-trained Model)***

- Purpose: Converts face images into 512-dimensional embeddings for comparison.
- Use: Measures facial similarity based on feature vectors.
- In This Project: Used to compare the uploaded image with all images in the database using a cosine distance metric.

3. ***Cosine Similarity / Distance Metric***

- Purpose: Calculates the angle-based distance between two facial embeddings.

Global Journal of Sociology and Anthropology
Vol. 15 (1), pp. 001-012, January, 2026
ISSN: 2756-3456

- Use: Determines whether two facial images are similar.
- In This Project: A threshold (usually around 0.3 to 0.4) is used to classify whether the faces match or not.

4. ***Image Preprocessing (OpenCV)***

- Purpose: Standardize input images for better feature extraction.
- Use: Resize, normalize, and convert images to grayscale (if needed).
- In This Project: Ensures all image inputs are properly read and formatted before being passed to the model.

5. ***Database Querying (MySQL)***

- Purpose: Store and retrieve missing child records.
- Use: Match identified faces with corresponding records from the database.
- In This Project: Each image has a filename and metadata (name, age, features, contact), queried after recognition.

6. ***Email Notification (Nodemailer Algorithmic Flow)***

- Purpose: Notify registered users of new missing child cases.
- Use: Programmatically constructs and sends HTML emails with images and child details.
- In This Project: Triggered after successful registration of a new case.

Architectural Components

The architecture of the Missing Child Detection System is designed to support efficient face recognition, secure data handling, and real-time processing. It integrates front-end interfaces, a back-end server, a facial recognition engine, and a relational database.

1. ***User Interface (UI):***

- A web-based interface developed using HTML, CSS, and JavaScript.
- Allows users (police, public, NGOs) to upload images and register/report missing children.

2. ***Server Backend (Node.js + Express):***

- Handles HTTP requests, image uploads, and routes.
- Manages API communication between the front-end and facial recognition script.

3. ***Facial Recognition Module (Python + DeepFace):***

- Processes uploaded images using CNN-based models like FaceNet.
- Compares facial embeddings against stored records to find matches.

4. ***Database (MySQL):***

- Stores details of missing children including image paths, names, age, location, and contact.
- Supports fast querying and result retrieval for matched faces.

5. ***Notification System (Nodemailer):***

- Sends alerts/emails when a new missing child is added or a match is found.

6. ***Cloud/Local Storage:***

- Images and records can be stored locally or integrated with cloud platforms for scalability.

Output Screens

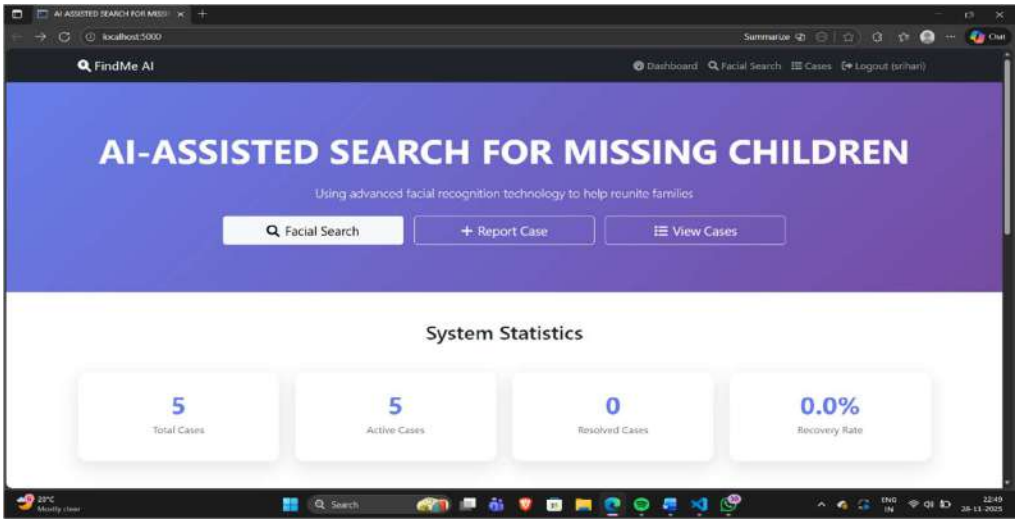


Fig 1 User Interface

Global Journal of Sociology and Anthropology
Vol. 15 (1), pp. 001-012, January, 2026
ISSN: 2756-3456

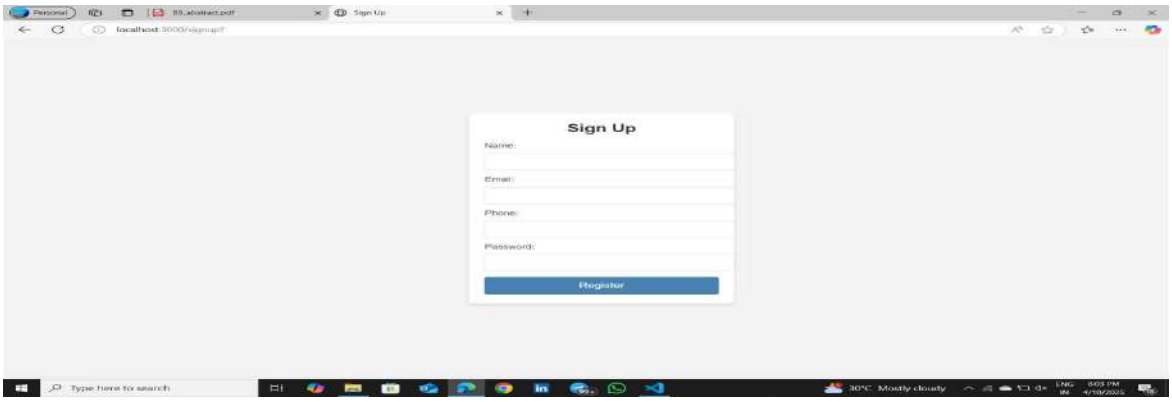


Fig 2 User Sign UP Page

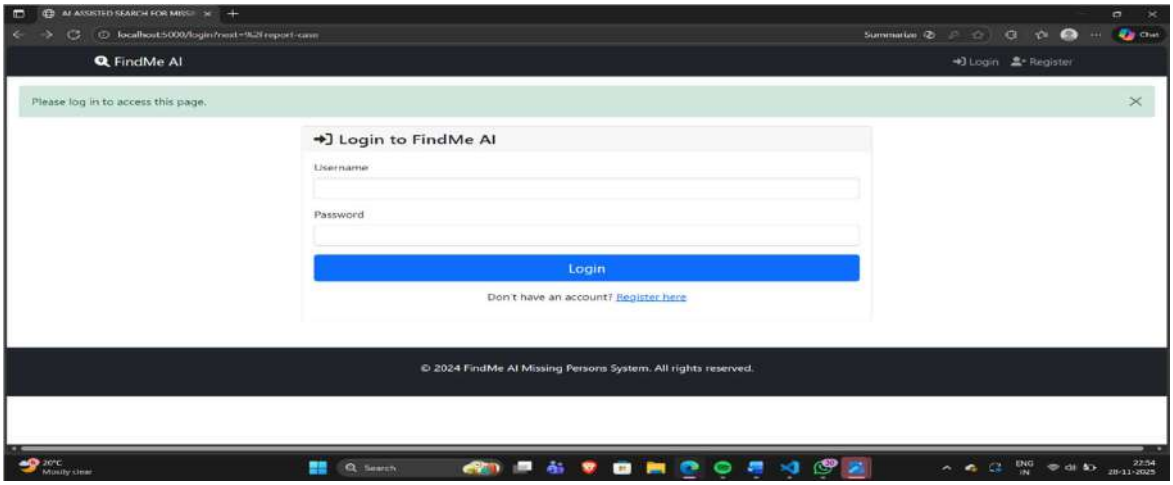


Fig 3 User Log In Page

Global Journal of Sociology and Anthropology
Vol. 15 (1), pp. 001-012, January, 2026
ISSN: 2756-3456

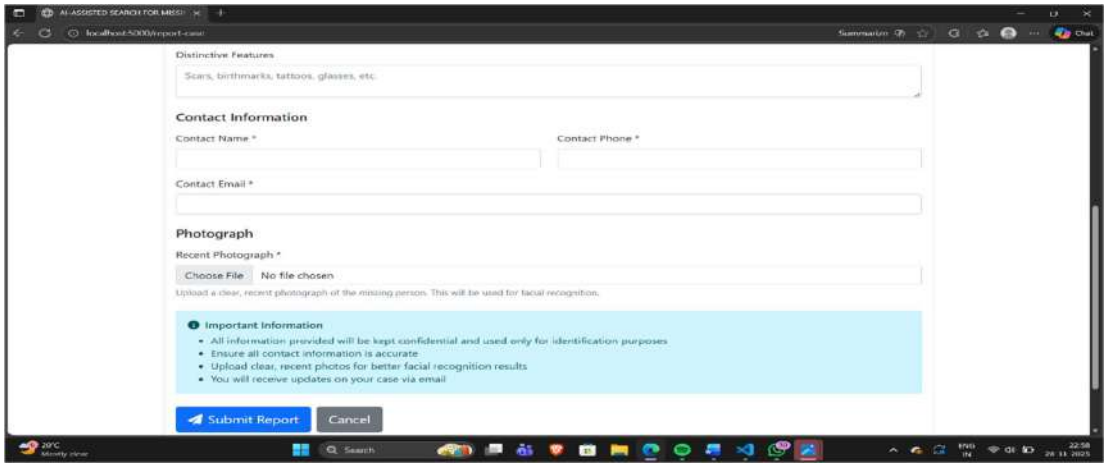


Fig 4 Register a Case

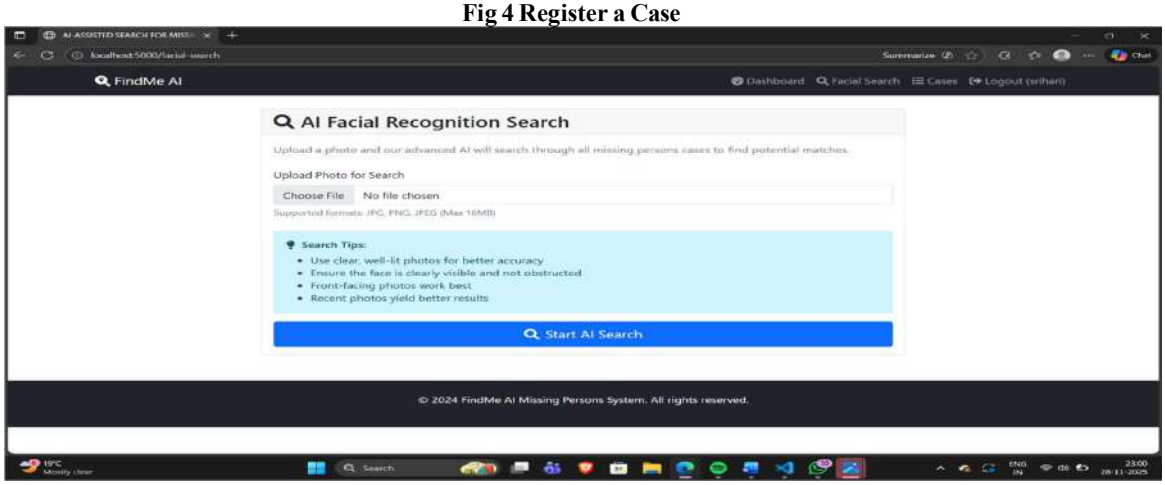


Fig 5 Identification Page

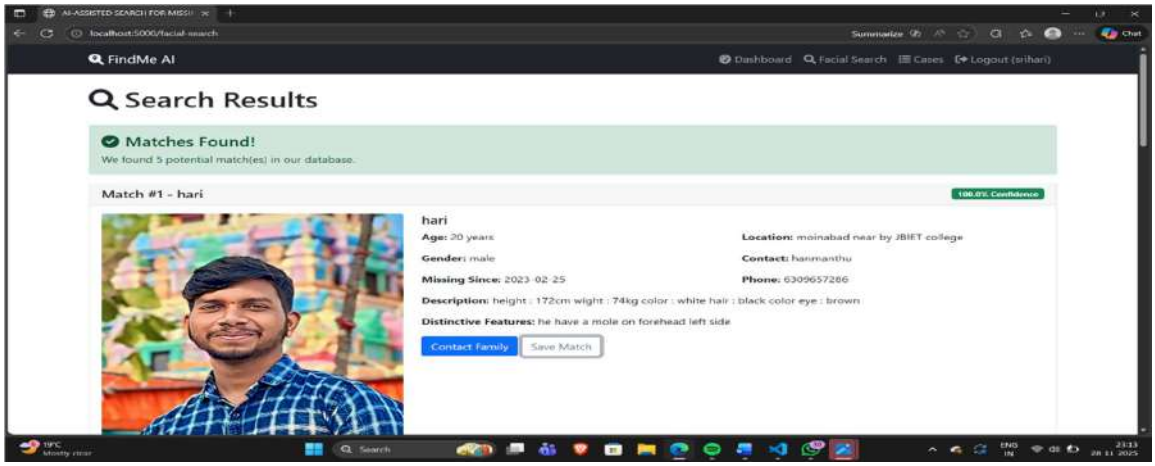


Fig 6 Identification Result Page

Global Journal of Sociology and Anthropology
Vol. 15 (1), pp. 001-012, January, 2026
ISSN: 2756-3456

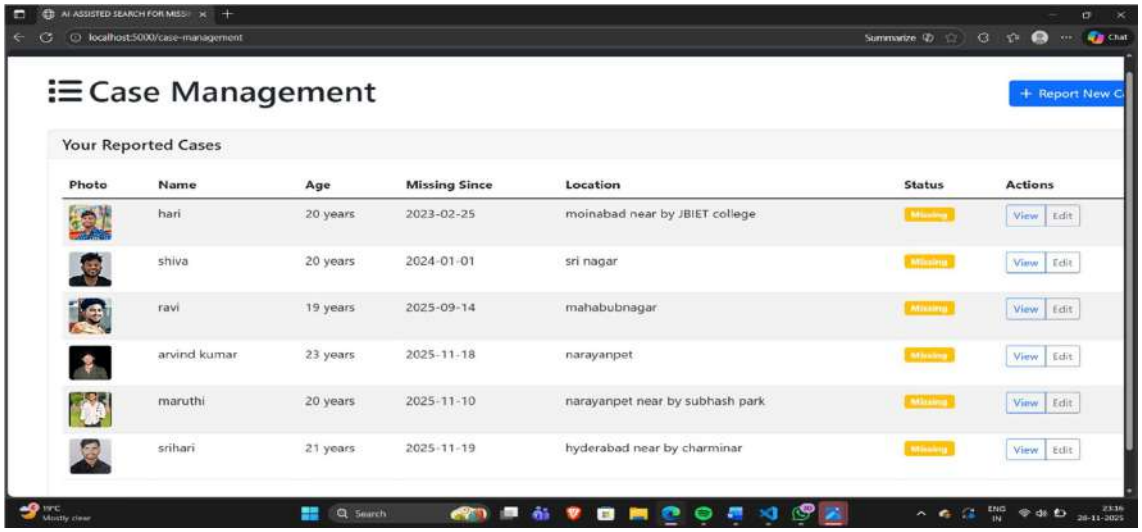


Fig 7 missing Notification

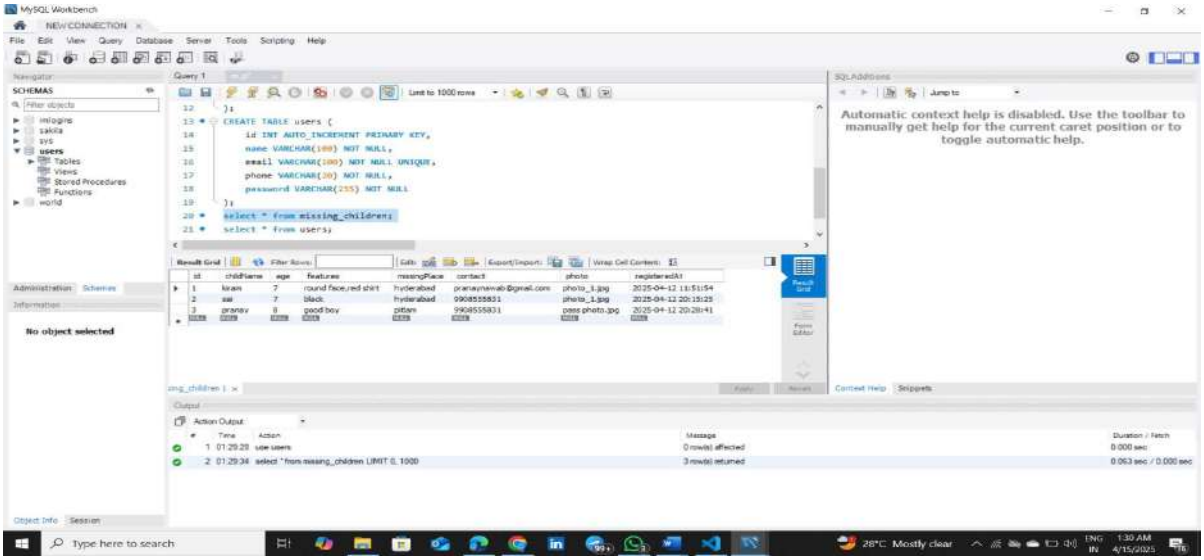


Fig 8 Back End Data Storage

System Testing

The testing phase is a critical step in ensuring the functionality and reliability of the Missing Child Identification System. The main goal of this phase is to verify that all components of the system—such as the backend (Node.js/Express.js), database (MySQL), frontend, and the face recognition module using DeepFace and TensorFlow—work as expected and communicate seamlessly. Given the sensitivity of the project, special focus was placed on accuracy, security, and overall performance. The testing process included unit testing, integration testing, functional testing, security testing, and user acceptance testing.

Unit testing was conducted on individual system components to ensure they function correctly. For example, the face recognition module was tested to verify its ability to accurately match registered child photos with the uploaded images. Additionally, tests were carried out on the email notification system to ensure it successfully sends the appropriate information to all registered users when a new missing child case is created.

Integration testing focused on validating the interaction between the system's modules. Key interactions, such as between the backend (Node.js) and the database (MySQL), were tested to ensure that missing child details, including images, were properly stored and retrieved. Additionally, the interaction between the frontend and the backend was thoroughly checked to guarantee smooth operations for features such as registering a missing child, logging in, and running the face recognition function.

Functional testing was performed to ensure the

Global Journal of Sociology and Anthropology
Vol. 15 (1), pp. 001-012, January, 2026
ISSN: 2756-3456

system operates as expected in real-world conditions. This included testing the entire workflow from registering a missing child to performing face recognition. All critical functionalities, such as logging in, uploading photos, and notifying users via email, were tested for both correct and error-prone scenarios. The system was tested to ensure it could detect a match when a registered child's image was uploaded and display appropriate details. At the same time, it correctly handled cases where no match was found, providing the user with feedback.

Test Cases

S.No	Test Case	Expected Output	Actual Output	Result
1.	User uploads child image for recognition	The image is uploaded successfully to the recognition system	Image uploaded successfully	Success
2.	ge comparison with the database	The system compares the uploaded image with the stored images in the database	Image comparison executed successfully	Success
3.	matching image found in the database	System returns "No match found" when the image does not match any in the database	No match found message displayed	Success
4.	ching image found in the database	System retrieves the child details (name, age, contact) for the matched image	Child details displayed correctly	Success
5.	hon face recognition script execution	The Python script successfully runs and returns a comparison result	Script executed and result returned	Success
6.	abase query for child details	Upon image match, the system queries the database for child details	Database query executed and details fetched	Success
7. or	handling for missing child ils	If no child details are found, the system handles the error gracefully	Error handled and user notified	Success
8.	etion of uploaded image r processing	The uploaded image is deleted from the server after processing	Uploaded image deleted successfully	Success

Results and Discussions

Results:

Based on the executed test cases for the *Find Me: AI-Assisted Missing Child Identification Portal*, the system has demonstrated reliable and accurate performance across all major functional aspects. Image uploads and processing were successfully executed, and the facial recognition model—powered by the integrated Python script—accurately matched images with entries in the database. Where no match was found, the system appropriately returned a "No match found" response, ensuring graceful error handling and user feedback. Additionally, the system correctly retrieved and displayed child details upon successful identification, managed database interactions efficiently, and maintained proper file handling by deleting uploaded images after processing.

Declaration:

All features and functionalities of the system have been thoroughly tested and validated based on the defined requirements and test cases. The system meets the intended objectives of aiding in missing child identification through a robust, AI-driven facial recognition approach, and is ready for deployment and further enhancement as needed.

Conclusion & Future Enhancements

Conclusion

The missing child identification system implemented in this project serves as a comprehensive solution for addressing the critical problem of child abduction and disappearance. By integrating face recognition technology with a user-friendly portal, the system enables parents and authorities to register missing children and identify them effectively using CNN-based facial recognition models. This system streamlines the process of locating missing children and significantly improves the chances of successful reunification with their families.

In addition, the automated email notification feature ensures that the registered users are informed about new cases, thereby increasing public participation and awareness. With the successful deployment of the facial recognition algorithm and efficient backend management, the system demonstrates its ability to assist in real-world scenarios, offering a scalable and adaptable solution for missing child recovery. This project is an essential step toward leveraging technology for societal safety and enhancing collaboration between citizens, law enforcement, and technological systems.

Future Enhancements:

In the future, this project can be expanded by integrating a mobile application to enhance accessibility and convenience. With the app, users could report missing children, upload photos, and receive real-time alerts directly on their smartphones. This would facilitate quicker communication and collaboration, allowing users to engage with the system from anywhere. Additionally, real-time push notifications and geo-location services could be implemented, enabling users to receive instant alerts about new missing child cases within their local area. This would help expand the system's impact by fostering immediate responses and involving a wider audience.

To improve the accuracy of the face recognition system, advanced AI models, including Convolutional Neural Networks (CNN) and transfer learning, could be employed. These models can significantly enhance recognition capabilities, particularly in dealing with cases of aging or poor-quality images. Furthermore, adding multilingual support would make the system more inclusive, allowing it to cater to diverse linguistic communities and expanding its usability across different regions and countries.

Future developments could also include integrating the system with national and international databases, enhancing cross-border cooperation in finding missing children. Additionally, incorporating other biometric features such as fingerprint or iris recognition would add another layer of accuracy to the identification process. Automated case closure could streamline operations, ensuring that cases are marked as resolved once a child is found, thereby improving the system's efficiency.

To ensure ongoing privacy and security, advanced encryption techniques and two- factor authentication can be employed. This will help protect user data and sensitive information, further solidifying the system's reliability. By implementing these enhancements, the project can evolve into a more robust, accessible, and impactful tool in the fight against child abduction and the search for missing children.

References

Books References

1. **Goodfellow, I., Bengio, Y., & Courville, A. (2016).** ***Deep Learning.*** **MIT Press.** This comprehensive guide covers the foundational concepts of deep learning, which are integral to the use of Convolutional Neural Networks (CNN) in face recognition systems.

2. **Chollet, F. (2018).** ***Deep Learning with Python.*** **Manning Publications**
This book provides an introduction to deep learning using Python and Keras, which is relevant to the

implementation of CNNs for face recognition in this project.

3. Face Recognition Documentation. (n.d.). ***face_recognition: Simple Face Recognition Library in Python.***
The official documentation for the face_recognition Python library, used to identify and compare faces in the project.

4. TensorFlow Documentation. (n.d.). ***TensorFlow: Machine Learning for Everyone.***
TensorFlow is a widely used deep learning framework, and this documentation provides extensive information on building machine learning models, including CNNs used in this project.

5. Keras Documentation. (n.d.). ***Keras: The Python Deep Learning API.***
Retrieved from https://keras.io/
This documentation explains how to use Keras, a high-level neural networks API, which is used for building the face recognition model in the project

6. NIST Special Publication 500-290. ***Face Recognition Vendor Test (FRVT).***
Retrieved from https://www.nist.gov/programs-projects/face-recognition-vendor- test- frvt
A detailed guide and performance evaluation on face recognition technologies, relevant for understanding the capabilities and limitations of the system used in the project.

7. OpenCV Documentation. (n.d.). ***Open Source Computer Vision Library (OpenCV).***
Retrieved from https://docs.opencv.org/
OpenCV provides extensive tools for image processing, which are essential for preprocessing images in the face recognition pipeline

8. DeepFace Documentation. (n.d.). ***DeepFace: A Python Framework for Facial Recognition.***
Retrieved from https://github.com/serengil/deepface
The DeepFace framework is used in the project for face recognition, and this documentation offers insights into its usage, including CNN-based recognition models.