Counselling Code : VBIT

VIGNANA BHARATHI Institute of Technology

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

A MAJOR PROJECT REPORT

ON

AI POWERED LOST AND FOUND OBJECT TRACKER WITH IMAGE RECOGNITION AND COMMUNITY BASED RECOVERY

Submitted in partial fulfillment of the requirement
for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (Data Science)

BY

BANDARI SAI KUMAR	**21P61A6719**
GHARSHAKOTI ANUSHA	**21P61A6765**
BANOTHU AKHIL	**22P65A6704**

Under the esteemed guidance of
Mr. B.Venkanna
Assistant Professor
Dept. of CSE(DS)

VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY

(A UGC Autonomous Institution, Approved by AICTE, Affiliated to JNTUH,

Accredited by NBA & NAAC) Aushapur (V), Ghatkesar (M), Medchal(dist.)

May – 2025

Counselling Code : VBIT

VIGNANA BHARATHI Institute of Technology

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

Aushapur (V), Ghatkesar (M), Hyderabad, Medchal – Dist, Telangana – 501 301.

DEPARTMENT
OF
COMPUTER SCIENCE & ENGINEERING
(Data Science)

CERTIFICATE

This is to certify that the major project titled ***"AI POWERED LOST AND FOUND OBJECT TRACKER WITH IMAGE RECOGNITION AND COMMUNITY BASED RECOVERY"*** *submitted by* ***B Sai Kumar (21P61A6719), Gharshakoti Anusha (21P61A6765), B.Akhil(22P65A6704)*** *in B. Tech. IV-II semester Computer Science & Engineering (Data Science) is a record of the bonafide work carried out by them.*

The results embodied in this report have not been submitted to any other University for the award of any degree

INTERNAL GUIDE
Mr.B.Venkanna
Assistant Professor

PROJECT CO-ORDINATOR
Mr.V.Raju
Assistant Professor

HEAD OF THE DEPARTMENT
Dr. Y. Raju
Associate Professor

EXTERNAL EXAMINER

DECLARATION

We, **B Sai Kumar, Gharshakoti Anusha, B Akhil,** bearing hall ticket numbers **21P61A6719,21P61A6765,22P65A6704** hearby declare that the mini project report entitled **"AI POWERED LOST AND FOUND OBJECT TRACKER WITH IMAGE RECOGNITION AND COMMUNITY BASED RECOVERY"** under the guidance of **Mr.B.Venkana**, Assistant Professor, Department of Computer Science and Engineering (Data Science), Vignana Bharathi Institute of Technology, Hyderabad, have submitted to Jawaharlal Nehru Technological University Hyderabad, Kukatpally, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering (Data Science).

This is a record of bonafide work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

Bandari Sai Kumar (21P61A6719)

Gharshakoti Anusha (21P61A6765)

Banothu Akhil (22P65A6704)

ACKNOWLEDGEMENT

We are extremely thankful to our beloved Chairman, **Dr. N. Goutham Rao** and secretary, **Dr. G. Manohar Reddy** who took keen interest to provide us the infrastructural facilities for carrying out the project work. Self-confidence, hard work, commitment and planning are essential to carry out any task. Possessing these qualities is sheer waste, if an opportunity does not exist. So, we whole- heartedly thank **Dr. P. V. S. Srinivas**, Principal, and **Dr. Y. Raju**, Head of the Department, Computer Science and Engineering (Data science) for their encouragement, support and guidance in carrying out the project.

We would like to express our indebtedness to the project coordinator, **Mr.V.Raju**, Assistant Professor, Department of CSE (Data science) for her valuable guidance during the course of project work.

We thank our Project Guide, **Mr.B.Venkanna**, Assistant Professor, for providing us with an excellent project and guiding us in completing our major project successfully.

We would like to express our sincere thanks to all the staff of Computer Science and Engineering (Data science), VBIT, for their kind cooperation and timely help during the course of our project. Finally, we would like to thank our parents and friends who have always stood by us whenever we were in need of them.

ABSTRACT

Losing personal belongings is a common yet distressing experience. This project proposes an innovative solution through the development of an **AI-powered lost and found object tracker** that leverages **image recognition** and **community-based recovery mechanisms**. The system integrates computer vision to allow users to upload images of lost or found items, which are then processed using deep learning models to identify and match objects with existing database entries. By harnessing the power of artificial intelligence, the system can automatically classify items and suggest potential matches with high accuracy.

In addition to the technical capabilities, the platform fosters a **community-driven approach**, allowing users to collaborate by reporting found items, confirming matches, and assisting in returns. Geolocation tagging further enhances the tracking process, making it easier to locate where items were last seen or found. The synergy of AI and community engagement not only accelerates the process of recovery but also builds a trustworthy ecosystem where users actively participate in helping others.

This solution aims to reduce the time and effort typically required in recovering lost belongings, increase the chances of retrieval, and encourage social responsibility through shared efforts.

Key words:

- Artificial Intelligence (AI)
- Image Recognition
- Object Detection
- Lost and Found System
- Community-Based Recovery

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

VISION:

To create a smarter, faster, and more reliable way to recover lost items by combining the power of artificial intelligence with a compassionate, connected community.

MISSION:

- **Develop an intelligent tracking system** that uses AI and image recognition to accurately identify and match lost and found items.
- **Enable real-time location-based reporting** to help users pinpoint where items were last seen or found, increasing recovery chances.
- **Build a supportive, community-driven platform** where users collaborate to report, identify, and return lost belongings efficiently and ethically

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

- **PEO 1:** To provide students with a strong foundation in artificial intelligence, computer vision, and software development to design innovative solutions for real-world problems.
- **PEO 2:** To instill the ability to develop user-centric and socially impactful technologies that promote community engagement and problem-solving.
- **PEO 3:** To encourage continuous learning, ethical practices, and teamwork in the development and deployment of AI-based systems.
- **PEO 4:** To prepare students to pursue careers in AI, software engineering, or higher studies with a focus on applying emerging technologies for societal benefits.

PROGRAM OUTCOMES (POs)

Engineering graduates will be able to:

1. **Apply knowledge of artificial intelligence and machine learning** to design intelligent systems for real-world applications like lost and found tracking.
2. **Design and develop software solutions** incorporating image recognition and object detection to address practical user needs.
3. **Utilize modern tools and technologies**, including geolocation services, databases, and cloud platforms, for building smart, scalable applications.
4. **Analyze, identify, and solve complex engineering problems** through critical thinking and data-driven approaches.
5. **Work effectively as part of a team**, demonstrating leadership and collaboration in multidisciplinary environments.
6. **Communicate technical information clearly and effectively**, both in written and oral forms, to diverse stakeholders.
7. **Understand professional, ethical, legal, and societal responsibilities**, especially concerning data privacy and community interactions.
8. **Engage in lifelong learning** to adapt to evolving technologies and continuously improve their technical and problem-solving skills.
9. **Develop innovative and sustainable solutions** that have a positive social impact, especially in community-centric platforms.
10. **Demonstrate project management and entrepreneurial skills** in the planning, development, and deployment of engineering solutions.

Program Specific Outcomes (PSOs)

1. **PSO 1:** Engineering graduates will be able to design and implement AI-based applications using machine learning, deep learning, and computer vision techniques to solve real-world problems like lost object detection and recognition.
2. **PSO 2:** Engineering graduates will be able to develop and deploy full-stack software systems that integrate image processing, geolocation services, and user-friendly interfaces to facilitate community-driven lost and found solutions.
3. **PSO 3:** Engineering graduates will be able to apply principles of ethical computing, data security, and social responsibility in the development of smart, AI-enabled platforms that engage and benefit communities.

Course Outcomes (COs)

CO1 - Identify the problem by applying acquired knowledge from survey of technical publications

CO2 - Analyse and categorize identified problem to formulate and fine the bestsolution after considering risks.

CO3 - Choose efficient tools for designing project.

CO4 - Build the project through effective team work by using recent technologies.

CO5 - Elaborate and test the completed task and compile the project report.

Correlation Levels

Substantial/High	**3**
Moderate/Medium	**2**

CO – PSO Correlation Matrix

COs	PSOs		
	PSO1	PSO2	PSO3
CO1	2	2	3
CO2	3	2	2
CO3	2	3	
CO4	2	2	3
CO5		2	2
CO	1.8	2.2	2

CO – PO Correlation Matrix

COs	POs											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	2	2			3	2	2	2	3
CO2	2	3	3	3	2			3	3	3	3	2
CO3	3	2	2	2	3			3	2	2	2	2
CO4	2	3	3	2	2			3	3	3	3	2
CO5	2	2	2	2	3			3	2	2	2	2
CO	2.4	2.4	2.4	2.2	2.4			3	2.4	2.4	2.4	2.2

Project Outcomes (PROs)

1. AI-Based Public Lost & Found Kiosk

Install interactive kiosks in public places (e.g., malls, airports, universities) equipped with cameras and OpenCV-based image recognition. Users can scan found items, and the system matches them with reported lost items in the database, providing instant contact or location info for recovery. This bridges the gap between offline and digital lost & found systems.

2. Mobile Lost Item Scanner

Develop a mobile app that uses a phone camera and OpenCV to let users scan found items (like bags, gadgets, or keys). The app detects and recognizes the item using AI, compares it with the lost item database, and notifies the original owner if a match is found. This allows easy item registration and detection on the go.

3. Community Reward System for Item Recovery

Integrate a points/reward system into the platform where users earn badges or incentives for returning lost items or helping match found items using image recognition. This gamifies the recovery process and encourages community participation, making it a win-win for everyone involved.

4. AI-Powered Campus Lost & Found Management System

Deploy a closed system for institutions like colleges or office campuses where CCTV footage is integrated with OpenCV to track misplaced items. When a student or staff uploads a lost item report, the system checks surveillance footage, identifies the item via image recognition, and helps locate where it was last seen.

5. Smart Bin for Lost Items

Design a “smart lost & found bin” where people can drop found items. A camera scans each item using OpenCV, classifies it, and uploads it to the system’s online portal with time and location data. This creates an organized inventory of found items in public areas, making retrieval easier and faster for the owners.

PRO – PSO Correlation Matrix

PROs	PSOs		
	PSO1	PSO2	PSO3
PRO1	3	2	3
PRO2	2	3	2
PRO3	2	2	3
PRO4	2	2	2
PRO5	2	2	3
PRO	2.2	2.2	2.6

PRO – PO Correlation Matrix

PROs	POs											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
PRO1	2	2	3	2	3			3	2	3	2	2
PRO2	3	3	3	3	3			3	2	3	3	3
PRO3	2	2	2	2	2			3	2	2	2	3
PRO4	2	2	3	2	2			3	3	3	2	2
PRO5	3	2	2	3	3			3	2	2	2	2
PRO	2.4	2.2	2.6	2.4	2.6			3	2.2	2.6	2.2	2.4

TABLE OF CONTENTS

6. Implementation

LIST OF FIGURES

CHAPTER – 1

1. INTRODUCTION

1.1 INTRODUCTION

Losing personal belongings such as keys, wallets, passports, or essential documents is a frustrating and all-too-common issue faced by individuals in daily life. These moments of forgetfulness can lead to stress, wasted time, and even financial loss. While traditional methods of tracking lost items—such as GPS-based trackers, Bluetooth-enabled devices, or manual searching—provide some relief, they often come with limitations. These systems may require additional hardware, frequent battery charging, manual setup, or constant proximity to function effectively. Moreover, their reliability in dynamic environments or when an item is outside the tracking range can be questionable.

To address these challenges, this project introduces an innovative, AI-powered, web-based platform designed to redefine how individuals locate and recover their misplaced belongings. The proposed system harnesses the power of artificial intelligence, combining technologies such as **image recognition, smart location memory**, and **habit-based behavioral prediction** to offer a seamless and intelligent object-tracking solution. Unlike conventional solutions, this platform eliminates the dependency on physical trackers or manual logging. Instead, it utilizes the device's built-in camera and sensors to identify and track objects autonomously.

Through **image recognition**, the system can detect and catalog frequently used items, associating them with specific contexts such as location, time, or activity. It then employs **smart location memory** to store and recall the last seen location of each item, providing real-time suggestions when an object goes missing. Additionally, the system learns from the user's routines and behaviors to make **habit-based predictions**, increasing the accuracy of locating items by understanding usage patterns and likely placements.

One of the standout features of this platform is its **community-driven lost and found module**, which allows users to collaborate in recovering items lost in shared or public spaces. Through this feature, individuals can report found items or seek assistance from others nearby who may have come across their missing belongings. This cooperative aspect not only enhances the system's effectiveness but also fosters a supportive network built on shared responsibility and goodwill.

1.2 PROBLEM STATEMENT

Losing personal belongings is a common occurrence that causes significant inconvenience and emotional distress. Traditional lost-and-found systems often lack efficiency, accessibility, and reliability. Individuals are left with limited options to locate their items, and lost objects frequently remain unrecovered due to a lack of coordination between those who find items and those who lose them.

With advancements in AI, image recognition, and community networking, there is a need for a modern solution that bridges the gap between those who lose and those who find objects. The proposed system aims to create an AI-powered platform that utilizes image recognition to identify lost items and a community-based recovery network to facilitate item return. This platform could dramatically improve lost-item recovery rates, minimize effort and time, and enhance user satisfaction.

1.3 OBJECTIVE

The objective of this project is to create an AI-powered lost-and-found platform that revolutionizes the process of recovering misplaced belongings by integrating advanced image recognition and community collaboration. This system will leverage image recognition technology to efficiently identify and the platform will connect individuals who find items with those who have lost them, streamlining the recovery process. The solution aims to reduce the time, effort, and emotional stress associated with losing personal belongings, while also promoting trust and cooperation within communities through a transparent and reliable system. Ultimately, this platform seeks to enhance the efficiency and reliability of lost-item recovery, improving user satisfaction and fostering a sense of collective responsibility.

1.4 AIM OF THE PROJECT

The aim of project is to design and implement an intelligent, AI-driven lost and found object tracking system that leverages image recognition and community engagement to improve the efficiency and success rate of recovering lost items.

CHAPTER - 2

2. LITERATURE SURVEY

The integration of AI technologies into lost-and-found systems has been a growing area of interest, particularly with the advent of web-based solutions. Traditional lost-and-found mechanisms often rely on manual processes, which are time-consuming and inefficient. To address these challenges, modern systems leverage web technologies and AI to create platforms that are accessible, efficient, and user-friendly.

Several projects have demonstrated the potential of using HTML, CSS, and JavaScript to build lightweight and responsive lost-and-found platforms. For instance, the AI-Powered Lost & Found project showcases a web application built entirely with these technologies. It uses JavaScript for frontend interactivity and integrates plugins to enhance functionality, such as local storage for saving data without a backend. This approach highlights the feasibility of creating efficient systems without relying on complex server-side technologies.

Another example is the Digital Lost and Found Item Portal, which emphasizes user authentication, object classification, and image uploading. While this project incorporates additional backend technologies, its frontend design principles can inspire similar implementations using only HTML, CSS, and JavaScript. Features like structured data storage and verification processes can be adapted using JavaScript libraries and plugins.

Additionally, the AI-Powered Lost-Found Automation System demonstrates the use of AI for object detection and matching. Although this system employs advanced AI models, similar functionalities can be achieved on the frontend by integrating pre-trained models or APIs through JavaScript. For example, plugins like TensorFlow.js enable on-device AI processing, making it possible to implement image recognition directly within a browser.

These projects underline the importance of combining intuitive design with advanced functionalities. By leveraging HTML, CSS, and JavaScript, along with plugins for features like image recognition and data storage, your project can offer a robust solution to the problem of lost-and-found item recovery. The use of plugins also ensures scalability and ease of maintenance, making the platform adaptable to future enhancements.

2.1 EXISTING SYSTEM

In the current scenario, lost and found systems are largely manual and outdated. Most organizations or public spaces rely on traditional methods such as physical registers, notice boards, or simple online forms to log lost and found items. These systems lack efficiency, transparency, and scalability. Users are required to sift through long lists or physically visit lost and found counters to check for their belongings. There is no intelligent mechanism to identify or match items, making the process time-consuming and often frustrating. Moreover, there is minimal or no integration of modern technologies like image recognition, artificial intelligence, or geolocation services. The absence of real-time tracking and automation results in a low success rate for item recovery. Additionally, the community is often not involved in the process, missing out on the opportunity for collaborative problem-solving.

2.2 PROPOSED SYSTEM

The proposed system introduces a modern, AI-powered solution to automate and optimize the process of managing lost and found items. By utilizing advanced image recognition through OpenCV and deep learning techniques such as Convolutional Neural Networks (CNNs), the system can automatically detect, classify, and match lost items with found reports. Users can easily upload images and descriptions of items they have lost or found. The AI engine analyzes these images and searches the database for potential matches. Additionally, the system features a geolocation module that tags the last known or found location of the item, helping narrow down search results.

2.2.1 SYSTEM ARCHIRECTURE:

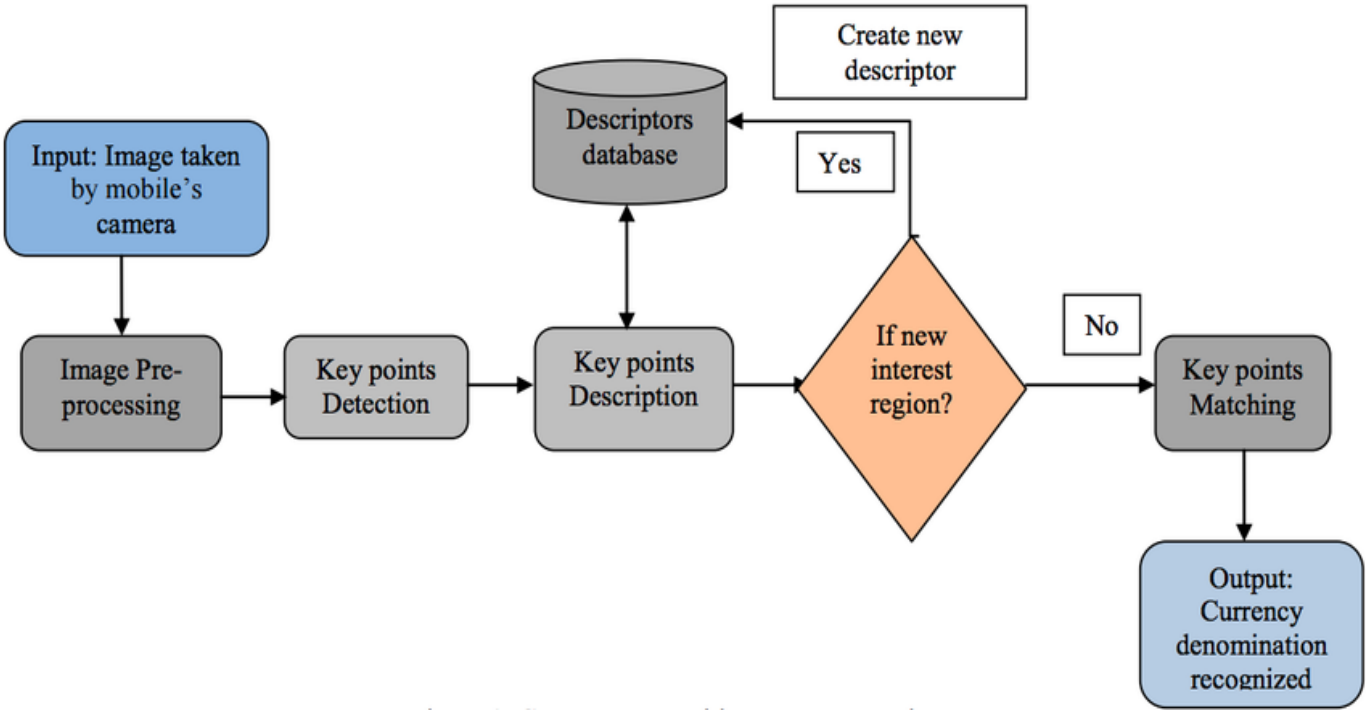

Fig 2.2.1 Architectural design

Here are some key points about the process depicted in the flowchart for currency denomination recognition:

1. **Purpose of System:** The process aims to recognize currency denominations using an image captured by a mobile device, showcasing the practical application of image processing in financial transactions and accessibility tools.
2. **Step-by-Step Workflow:**
    - The system starts with capturing an image via a mobile camera.
    - Pre-processing techniques are applied to enhance the image for accurate analysis.
    - Key points within the image are detected and described for further processing.
    - A decision point evaluates whether the region of interest is new, enabling dynamic updates to the descriptors database.

3. **Innovative Features:**
    - Incorporates key points matching techniques to identify currency denominations accurately.
    - Updates descriptors dynamically to accommodate new currency or changes in existing currency.
4. Output & Utility:
    - Provides a clear output of recognized currency denomination, which can be utilized in automated systems, payment gateways, or accessibility applications for visually impaired individuals.
5. Potential Enhancements:

- Integration with advanced plugins for faster pre-processing and matching.
- Exploring optimization for real-time recognition on mobile devices.

2.2.1 BENFITS OF PROPOSED SYSTEM

The proposed AI-powered lost-and-found object tracker system offers numerous benefits that aim to revolutionize the process of recovering lost items. By integrating advanced image recognition technology, the platform enables quick and accurate matching of lost objects with their owners, drastically reducing the time and effort required for recovery. Its user-friendly interface, built using HTML, CSS, and JavaScript, ensures accessibility and ease of use, allowing individuals to report and search for lost items seamlessly. Additionally, the system fosters community collaboration by connecting those who find items with those who have lost them, promoting a sense of trust and cooperation. The inclusion of dynamic updates, such as descriptors for new objects, keeps the platform adaptable and robust over time. Furthermore, the system minimizes development and operational costs due to its reliance on lightweight web technologies and plugins, making it a cost-effective solution. By replacing physical lost-and-found centers, the platform also contributes to sustainability efforts, reducing the use of paper and manual resources. Overall, the proposed system not only enhances efficiency and reliability but also creates an engaging and secure experience for users while promoting technological awareness and eco-friendliness.

2.3 SCOPE OF THE PROJECT

The scope of the AI-powered lost-and-found object tracker project encompasses the development and implementation of a web-based platform designed to streamline the recovery of misplaced items through image recognition and community collaboration. The project focuses on creating an efficient and accessible system using HTML, CSS, JavaScript, and plugins, ensuring lightweight and seamless user interaction. It includes features such as capturing and uploading images of lost or found items, matching items through image recognition algorithms, and connecting users for item recovery within a secure and user-friendly environment.

The platform's scope extends to dynamic adaptability, allowing for continuous updates to include new descriptors or objects. By fostering community engagement, the system promotes collaboration, trust, and a sense of collective responsibility. Additionally, scalability is a key aspect, enabling integration of multilingual support and expansion into diverse geographic regions. The project also prioritizes cost-effective development and eco-friendly practices by reducing reliance on physical resources. Ultimately, this system seeks to address a universal problem, offering a reliable and modern solution to improve lost-item recovery rates and enhance user satisfaction.

CHAPTER – 3

3. FEASIBILITY STUDY

3.1 FEASIBILITY STUDY

A feasibility study assesses the practicality of implementing the proposed AI-powered lost-and-found object tracker system, evaluating its technical, operational, and economic aspects to ensure successful development and deployment.

3.1.1 Technical Feasibility

The project employs widely available and reliable web technologies such as HTML, CSS, JavaScript, and plugins, making the technical implementation straightforward. The use of JavaScript plugins for features like local storage, TensorFlow.js for AI-based image recognition, and libraries for frontend interactivity ensures that the system can be implemented without relying on server-side technologies. Additionally, lightweight web technologies enable scalability and adaptability to future enhancements, including multilingual support and integration with APIs for advanced functionalities.

3.1.2 Operational Feasibility

The system's user-friendly interface and intuitive design make it highly operational. By enabling individuals to report lost items, upload images, and interact within a secure community-based platform, the system is poised to deliver a seamless experience. Clear workflows, such as object upload, image matching, and connection facilitation, ensure efficiency and reliability. Community engagement features, such as reward mechanisms or notifications, foster active participation and encourage users to adopt the platform. Furthermore, operational challenges like maintenance and updates can be addressed through periodic reviews and modular design practices.

.

3.1.3 Economical Feasibility

The use of lightweight web technologies reduces development and operational costs significantly. As the platform does not require server-side infrastructure, hosting and maintenance expenses are minimal. The integration of plugins, which are often open-source or low-cost, further reduces expenditure. Additionally, the system's ability to replace physical lost-and-found centers contributes to cost savings for organizations and individuals. The project’s economic feasibility is reinforced by its scalability, which allows incremental revenue generation through premium features or advertisement integration without compromising the core functionality.

CHAPTER – 4

4 SYSTEM REQUIREMENTS

4.1 Hardware Requirements

- **Processor** : Intel or Ryzen
- **Speed :** 1.8Hz
- **Processor Monitor :** 14” MONITOR
- **Hard Disk :** 32GB or Above
- **Internet :** Proper internet Connection
- **RAM :** 2GB or Above

4.2 Software Requirements

- **Frontend**: HTML, CSS, JavaScript for UI/UX and interactivity.
- **Plugins**: TensorFlow.js (image recognition), LocalStorage/IndexedDB (data storage), Croppie.js (image resizing).
- **Development Tools**: Code editor (e.g., VS Code), Git, modern browser (e.g., Chrome).

CHAPTER – 5

5 METHODOLOGY

The methodology for the AI-powered lost-and-found object tracker focuses on a structured, user-centric approach to ensure successful design, implementation, and deployment. It begins with a thorough requirement analysis to identify essential features, such as image recognition, object matching, and community collaboration, while also defining use cases and user workflows. Next, a detailed design and planning phase involves creating system architecture, wireframes, and prototypes using HTML, CSS, and JavaScript with plugins like TensorFlow.js for AI functionalities and LocalStorage for client-side data handling. In the development phase, the platform is built by implementing core functionalities, including image upload, processing, and notification systems, while ensuring scalability and adaptability. Rigorous testing follows to validate image recognition accuracy, UI responsiveness, and device/browser compatibility. The platform is then deployed on a lightweight hosting environment, ensuring accessibility and providing an efficient lost-and-found service.

5.1 Research Approach

The research approach involves reviewing existing systems and technologies to identify gaps and opportunities. Key objectives and workflows are defined through conceptual analysis, followed by building a prototype using HTML, CSS, JavaScript, and plugins like TensorFlow.js. User testing and feedback refine the design to ensure usability and effectiveness. The project is iteratively improved and prepared for scalable deployment, ensuring it remains user-centric and innovative.

5.2 Data Collection and Experimentation Methodology

The project addresses inefficiencies in traditional lost-and-found systems by creating a web- based platform with image recognition and community collaboration. It uses HTML, CSS, and JavaScript with plugins like TensorFlow.js for efficient item matching. The system is scalable, user-friendly, and cost-effective, reducing recovery time and promoting community trust. The methodology includes requirement analysis, design, development, testing, and deployment, with regular updates

5.3 Data Preprocessing and Analysis

The data preprocessing and analysis for the AI-powered lost-and-found object tracker ensures reliable and efficient item recovery. It involves standardizing image dimensions, normalizing pixel values, and removing noise for enhanced clarity. Key features like edges and shapes are extracted, while data augmentation increases dataset diversity by generating image variations. Quality checks validate user-uploaded images before processing. For analysis, matching accuracy is evaluated, along with user behavior patterns and engagement trends to optimize workflows. Processing times and error frequencies are assessed, and community collaboration success rates are analyzed to refine recovery mechanisms. This comprehensive approach guarantees a robust, user-friendly system.

5.4 Algorithm/Model Selection and Justification

The project uses TensorFlow.js with pre-trained models like MobileNet for in-browser image recognition, ensuring efficiency and compatibility. Feature-matching algorithms like ORB or SIFT are used for identifying and matching objects. LocalStorage or IndexedDB handles lightweight, client-side data storage. These choices ensure scalability, cost-effectiveness, and a seamless user experience, making the system reliable and accessible.

5.5 Tools and Development Environment

The tools and development environment for the AI-powered lost-and-found object tracker project include:

Tools:

- **Frontend Technologies**: HTML, CSS, and JavaScript for creating a responsive and interactive user interface.
- **Plugins**: TensorFlow.js for AI-based image recognition, LocalStorage/IndexedDB for client-side data storage, and Croppie.js for image resizing.
- **Development Tools**: Visual Studio Code as the code editor, Git for version control, and modern browsers like Chrome for testing.

Development Environment:

- A computer or laptop with at least 8GB RAM, Intel i5 processor, and a compatible operating system (Windows, macOS, or Linux).
- Lightweight hosting platforms like GitHub Pages or local development servers (e.g., XAMPP) for testing and deploying the platform.

CHAPTER – 6

6 IMPLEMENTATION

The implementation of the AI-Powered Lost and Found Object Tracker is a crucial phase that brings the entire conceptual framework into a functional reality. This system is primarily designed as a client-side web application, focusing on accessibility, responsiveness, and performance. The core technologies used include HTML5, CSS3, and vanilla JavaScript, ensuring that the platform remains lightweight while still delivering a modern and visually engaging user interface. The login and signup pages are designed with responsive design principles, using CSS media queries and flexible layouts to ensure seamless accessibility across various devices, including mobile phones, tablets, and desktops. The login form allows users to securely authenticate themselves, while the signup form supports new user registrations with dynamic validations to improve user experience and data accuracy. JavaScript handles all user interface interactivity such as toggling password visibility, dynamically switching between forms, and displaying contextual error messages in real-time based on user input.

A key highlight in the implementation is the integration of IndexedDB, a powerful client-side NoSQL database built into modern browsers. IndexedDB is used to store and manage user credentials and profile data securely without the need for a server-side backend. This local database supports asynchronous operations, allowing efficient reads and writes without blocking the UI thread. During the signup process, user information such as name, email, and password is stored into the ‘userProfiles’ object store within IndexedDB, with the email address used as the unique key to prevent duplicate registrations. When logging in, the application queries IndexedDB to fetch the user data and compare the credentials in real-time, providing immediate feedback on authentication success or failure. While this approach simplifies the system architecture by removing the need for a server, in a production-grade system, proper password hashing and encryption would be critical for security.

In addition to IndexedDB, the system also makes use of localStorage to persist session-related information such as login status, user email, and display name across sessions. This ensures that even if the browser is refreshed, the user remains logged in until they explicitly log out, enhancing the user experience.

6.1 Development Environment Overview

The development environment for the AI-powered lost-and-found object tracker project is designed to support efficient and accessible development processes. It primarily relies on web technologies like HTML, CSS, and JavaScript for frontend development, ensuring responsiveness and interactivity. Tools such as Visual Studio Code are used as the primary code editor, with Git for version control to manage and track code changes effectively. The platform leverages modern web browsers like Chrome for testing and debugging, ensuring compatibility across devices.

On the hardware side, a computer with a minimum of 8GB RAM, an Intel i5 processor, and a reliable operating system (Windows, macOS, or Linux) is recommended for smooth development. For hosting and testing, lightweight platforms like GitHub Pages or local servers such as XAMPP provide a convenient and cost-effective environment. This setup ensures seamless development, testing, and deployment of the platform. Let me know if you'd like further specifics or recommendations!

6.2 Code Overview and Key Modules

The project is built using HTML, CSS, and JavaScript, focusing on a lightweight and interactive web-based system. JavaScript plugins, such as TensorFlow.js for image recognition and LocalStorage/IndexedDB for client-side data storage, enable AI functionalities without requiring backend services. The system captures, processes, and matches lost-and-found item images through an intuitive interface.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login - AI Lost & Found Tracker</title>
    <style>
        body {
            font-family: 'Segoe UI', Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
            background-image: url('static/uploads/pexels-jplenio-1103970.jpg');
            background-size: cover;
            background-position: center;
            transition: background-position 0.5s ease;
```

```
}
 Body;
@media (max-width: 768px) {
    body {
        padding: 20px;
        background-attachment: fixed;
    }
    .login-container {
        width: 90%;
        padding: 1.5rem;
    }
}
.login-container {
    background-color: rgba(255, 255, 255, 0.9);
    padding: 2rem;
    border-radius: 8px;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    width: 350px;
    text-align: center;
}
h1 {
    color: #007BFF;
    margin-bottom: 1.5rem;
}
.form-group {
    margin-bottom: 1rem;
```

```
    text-align: left;
    position: relative;
}
.toggle-password {
    position: absolute;
    right: 10px;
    top: 70%;
    transform: translateY(-50%);
    cursor: pointer;
    color: #666;
    font-size: 1rem;
    user-select: none;
}
.toggle-password:hover {
    color: #007BFF;
}
label {
    display: block;
    margin-bottom: 0.5rem;
    font-weight: bold;
}
    border: 2px solid rgba(255, 255, 255, 0.3);
    border-top-color: white;
    border-radius: 50%;
    animation: spin 0.8s linear infinite;
}
```

```
@keyframes spin {
    to { transform: rotate(360deg); }
}
.switch-form {
    margin-top: 1.5rem;
    font-size: 0.9rem;
    transition: all 0.3s ease;
}
.switch-form a {
    color: #007BFF;
    text-decoration: none;
    font-weight: 600;
    transition: all 0.2s ease;
}
.switch-form a:hover {
    color: #0056b3;
    text-decoration: underline;
}
.error {
    color: #dc3545;
    font-size: 0.9rem;
    margin-top: 0.5rem;
    padding: 0.5rem;
    background: rgba(220, 53, 69, 0.1);
    border-radius: 4px;
    transition: all 0.3s ease;
```

```
        }
        #login-form, #signup-form {
            transition: all 0.3s ease;
        }
    </style>
</head>
<body>
    <div class="login-container">
        <h1>AI Lost & Found Tracker</h1>
        <div id="login-form">
            <h2>Login</h2>
            <div class="form-group">
                <label for="login-email">Email</label>
                <input type="email" id="login-email" required>
            </div>
            <div class="form-group">
                <label for="login-password">Password</label>
                <input type="password" id="login-password" required>
                <span class="toggle-password" onclick="togglePassword('login-password')">👁</span>
            </div>
            <button onclick="login()">Login</button>
            <div class="switch-form">
                Don't have an account? <a href="#" onclick="showSignup()">Sign up</a>
            </div>
            <div id="login-error" class="error"></div>
        </div>
```

```
<div id="signup-form" style="display: none;">
  <h2>Sign Up</h2>
  <div class="form-group">
    <label for="signup-name">Full Name</label>
    <input type="text" id="signup-name" required>
  </div>
  <div class="form-group">
    <label for="signup-email">Email</label>
    <input type="email" id="signup-email" required>
  </div>
  <div class="form-group">
    <label for="signup-password">Password</label>
    <input type="password" id="signup-password" required>
    <span class="toggle-password" onclick="togglePassword('signup-password')">👁</span>
  </div>
  <div class="form-group">
    <label for="signup-confirm">Confirm Password</label>
    <input type="password" id="signup-confirm" required>
    <span class="toggle-password" onclick="togglePassword('signup-confirm')">👁</span>
  </div>
  <button onclick="signup()">Create Account</button>
  <div class="switch-form">
    Already have an account? <a href="#" onclick="showLogin()">Login</a>
  </div>
  <div id="signup-error" class="error"></div>
</div>
```

```
    </div>
    </div>
    <script>
        // Initialize IndexedDB with same database as app.html
        let db;
        const request = indexedDB.open('LostAndFoundDB', 2);
        let db;
        request.onerror = function(event) {
            console.error('Database error:', event.target.error);
           document.getElementById('login-error').textContent = 'Failed to initialize database. Please refresh the page.';
        };
    // Check localStorage availability
            if (!isLocalStorageAvailable()) {
                errorElement.textContent = 'Your browser does not support required storage features';
                return;
            }
                  request.onerror = function(event) {
                   console.error('Login error:', event.target.error);
                    errorElement.textContent = 'Error logging in. Please try again.';
                   loginBtn.disabled = false;
                   loginBtn.textContent = 'Login';
                };
            } catch (e) {
                console.error('Login exception:', e);
                errorElement.textContent = 'Login failed. Please refresh and try again.';
```

```
        loginBtn.disabled = false;
        loginBtn.textContent = 'Login';
    }
}
function signup() {
    const name = document.getElementById('signup-name').value;
    const email = document.getElementById('signup-email').value;
    const password = document.getElementById('signup-password').value;
    const confirm = document.getElementById('signup-confirm').value;
    const errorElement = document.getElementById('signup-error');
        if (!name || !email || !password || !confirm) {
        errorElement.textContent = 'Please fill in all fields';
        return;
    }
        if (password !== confirm) {
        errorElement.textContent = 'Passwords do not match';
        return;
    }
       errorElement.textContent = 'Passwords do not match';
        if (password.length < 6) {
        errorElement.textContent = 'Password must be at least 6 characters';
        return;
    }
   const transaction = db.transaction(['userProfiles'], 'readwrite');
    const objectStore = transaction.objectStore('userProfiles');
    const request = objectStore.add({
```

```
            userId: email,
            name: name,
            password: hashPassword(password)
        });
        const transaction = db.transaction(['userProfiles'], 'readwrite');
        const objectStore = transaction.objectStore('userProfiles');
        const request = objectStore.add({
        request.onsuccess = function() {
            localStorage.setItem('loggedIn', 'true')
            localStorage.setItem('userEmail', emai
            localStorage.setItem('userName', name);
            window.location.href = 'app.html';
        };
        }
        request.onerror = function(event) {
            if (event.target.error.name === 'ConstraintError') {
                errorElement.textContent = 'Email already registered';
            } else {
                errorElement.textContent = 'Error creating account';
            }
        };
    }
  </script>
</body>
</html>
```

6.3 Challenges and Solutions During Implementation

During the implementation of the AI-powered lost-and-found object tracker, several challenges arose, requiring strategic solutions to ensure functionality and efficiency.

One major challenge was image recognition accuracy, as variations in lighting, angles, and image quality affected detection. To address this, preprocessing techniques such as image normalization, noise reduction, and feature extraction were integrated using JavaScript plugins to enhance recognition reliability. Another obstacle was data storage management, where handling user-uploaded images without a backend required optimized use of LocalStorage and IndexedDB, ensuring efficient retrieval and minimal storage constraints.

Scalability was also a concern, particularly when handling multiple images and matching processes. The solution involved using lightweight models like MobileNet in TensorFlow.js, which enabled real-time AI processing within the browser without requiring heavy server resources. User engagement posed another challenge—ensuring active participation in reporting and recovering lost items. This was mitigated by introducing notification features, interactive design elements, and reward mechanisms to encourage users to contribute effectively.

6.4 Integration of Components

```
// Show loading state
loginBtn.disabled = true;
loginBtn.textContent = 'Logging in...';
loginBtn.textContent =’Logging in…’;
try {
   if (!db) {
      throw new Error('Database not initialized');
    } const transaction = db.transaction(['userProfiles'], 'readonly');
   const objectStore = transaction.objectStore('userProfiles');
   const request = objectStore.get(email);
    request.onsuccess = function(event) {
```

```
const user = event.target.result;
if (user && user.password === password) { // In real app, use proper password hashing
    try {
        localStorage.setItem('loggedIn', 'true');
        localStorage.setItem('userEmail', email);
        localStorage.setItem('userName', user.name);
        console.log('Login successful, redirecting...');
        window.location.href = 'app.html';
    } catch (e) {
        console.error('LocalStorage error:', e);
        errorElement.textContent = 'Failed to save login session';
        loginBtn.disabled = false;
        loginBtn.textContent = 'Login';
    }
} else {
    errorElement.textContent = 'Invalid email or password';
    loginBtn.disabled = false;
    loginBtn.textContent = 'Login    }
```

6.5 Testing and Debugging

Testing and debugging are crucial to ensuring the AI-powered lost-and-found object tracker functions efficiently and reliably. The testing phase includes unit testing for individual components like image upload, recognition, and storage, verifying their performance separately. Integration testing follows, ensuring that different modules work seamlessly together, particularly in image matching and user interaction workflows. User testing is conducted to evaluate the platform's ease of use and accuracy in identifying lost items, refining UI elements based on feedback.

For debugging, tools like browser developer consoles are used to detect JavaScript errors, while automated tests help validate image recognition performance. Logging mechanisms track user interactions and system behaviors to identify inconsistencies or failures in object matching. Performance testing ensures optimized storage handling, responsiveness, and scalability, while security testing verifies data protection in user-generated content. The debugging process involves iterative refinements based on detected errors, ensuring the system remains functional and user-friendly. Let me know if you'd like deeper insights into specific testing methods.

6.6 Testing and Evaluation

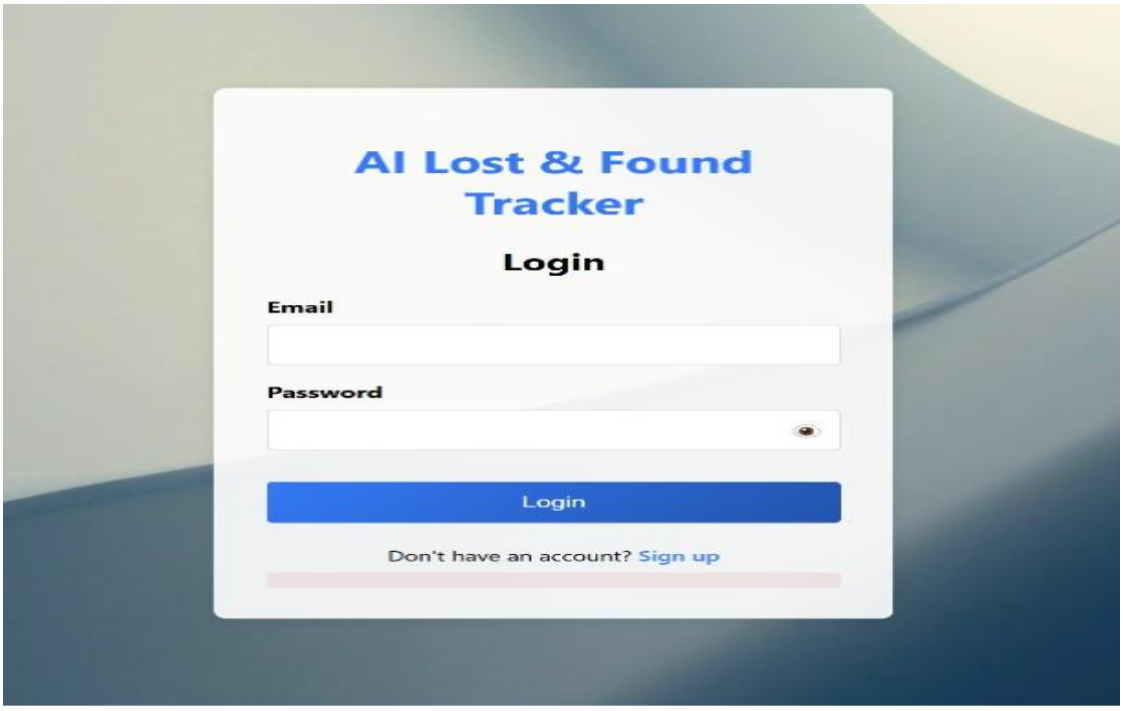

Fig 6.6.1: Lost and Found website Login Page

The login page of the Lost and Found website is designed with a focus on simplicity, accessibility, and visual appeal. Positioned at the center of the screen, the login form is encapsulated in a clean, rounded white container with a slight shadow to enhance visibility against a soft, high-resolution background image that adds depth without distracting the user. The form features clearly labeled input fields for email and password, each styled with smooth transitions and border highlights that respond to user interaction. An eye icon toggle is implemented in the password field, allowing users to view or hide their input for added convenience. The login button is styled with a gradient blue color scheme that complements the theme of trust and security, providing responsive hover effects and a subtle animation to indicate loading when clicked.

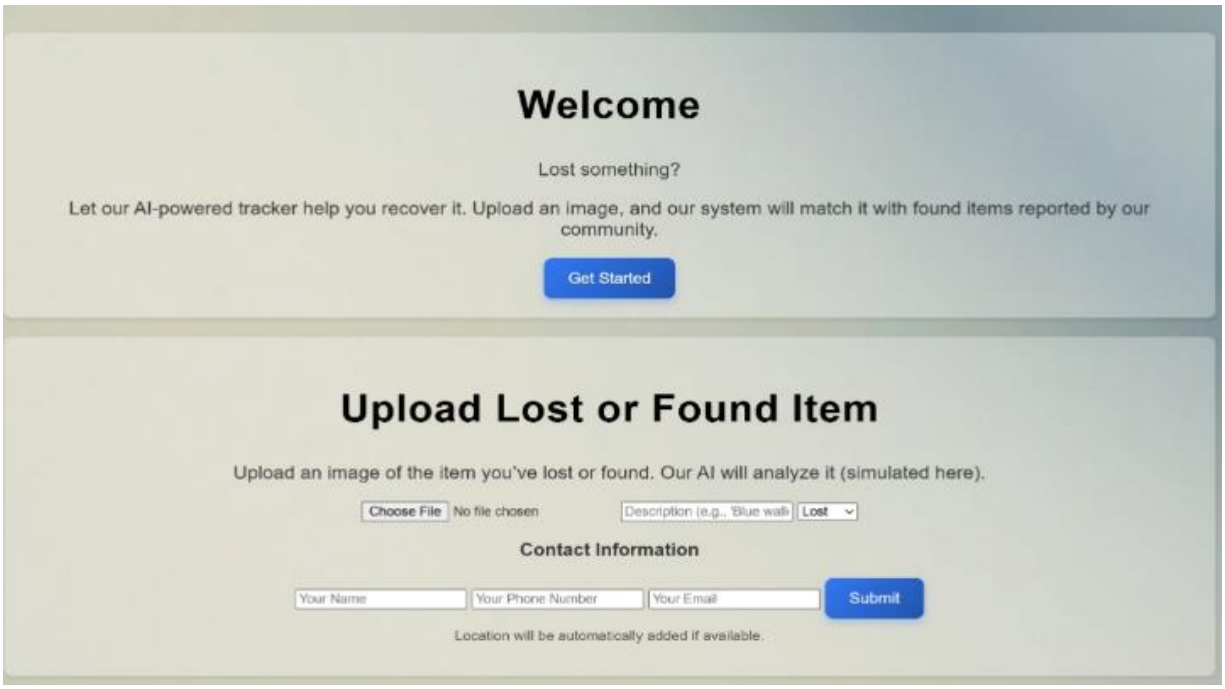

Fig-6.6.2 Lost and Found website Homepage

The homepage of the Lost and Found website serves as the central hub for users, offering a visually appealing and user-friendly interface that invites engagement and trust. At first glance, visitors are greeted with a bold, welcoming banner that outlines the purpose of the platform—helping users recover lost items through AI-powered image recognition and community collaboration. A navigation bar at the top provides easy access to essential sections such as Home, Report Lost Item, Report Found Item, View Reports, and Profile. Just below the banner, an interactive section displays recent lost and found reports with images and brief descriptions, encouraging quick browsing and user participation. A search bar, enhanced with filters like category, location, and date, allows users to find items efficiently. The homepage also highlights the platform's key features, including real-time object detection, facial recognition for identity-based lost items, and community alerts. With a modern color palette, responsive layout, and call-to-action buttons that guide users toward reporting or searching for items, the homepage is designed to be both functional and engaging. Testimonials, safety tips, and a brief explanation of how the system works are placed strategically to build user confidence and promote the platform's reliability and innovation.

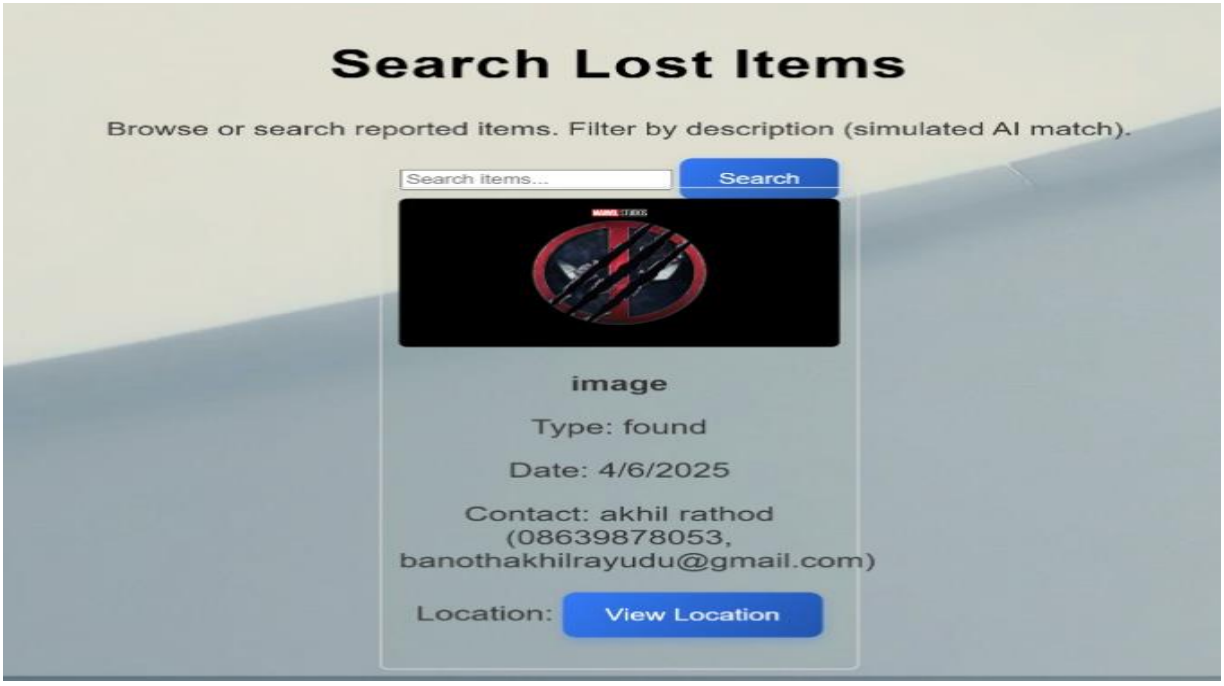

Fig-6.6.3 Search Lost Items in Website

The **Search Lost Items** page is a critical feature of the Lost and Found website, allowing users to efficiently locate missing belongings using advanced search tools powered by AI and computer vision. The interface is designed to be intuitive and responsive, ensuring a smooth user experience across all devices. At the top of the page, users are presented with a prominent search bar where they can input keywords related to the lost item—such as item name, brand, color, or distinguishing features. Below this, various filter options are available to narrow down results, including category (e.g., electronics, documents, accessories), location where the item was lost or found, and date range. Additionally, users can upload an image of the lost item to leverage the platform's AI-based image recognition feature, which scans the database for visual matches.

Fig-6.6.4 Community Based Recovery Feature

The Community Based Recovery Feature is the heart of the Lost and Found platform, designed to harness the collective power of the user community in locating and returning lost items. Instead of relying solely on system automation, this feature encourages active participation from users who either find or spot missing items. When a user reports a found object, the platform not only lists it in the searchable database but also notifies other users in the relevant geographical area based on location tags and user preferences. Users can also subscribe to alerts related to specific types of items or locations. Additionally, a community feed or dashboard is available where users can post updates, share leads, and comment on existing reports, fostering real-time interaction and collaboration. The feature also supports community upvoting, where the most credible and detailed reports rise to the top, improving visibility. In more advanced settings, verified users can act as mediators to confirm claims and help match items with rightful owners. By building a network of trust and mutual support, the community-based recovery system transforms the platform from a static listing site into a dynamic, people-powered recovery network, dramatically increasing the chances of reuniting individuals with their lost belonging.

CHAPTER – 7

7. CONCLUSION & FUTURE SCOPE

The AI-powered lost-and-found object tracker effectively resolves the challenges of traditional systems by utilizing image recognition and fostering community collaboration. This user-friendly and cost-efficient platform streamlines item recovery, demonstrating reliability through its lightweight architecture and dynamic adaptability. By reducing recovery time and enhancing user satisfaction, the system provides a robust solution for managing lost items. In the future, the platform could expand its capabilities through multilingual support, mobile app development, integration of advanced AI models for improved recognition accuracy, and global scaling to support international communities. Additional features like real-time notifications and APIs for external system connections could further enhance its efficiency and scalability, solidifying its position as a universal solution for lost-and-found management.

The development of the AI-powered Lost and Found Object Tracker with Image Recognition and Community-Based Recovery marks a significant step forward in applying artificial intelligence and computer vision to solve real-world problems. Losing personal belongings is a common yet distressing experience. This project addresses that pain point by creating a smart and efficient platform that automates and enhances the process of finding and recovering lost items.

By leveraging advanced technologies such as OpenCV for object and facial recognition, along with a streamlined user interface built using HTML, CSS, and JavaScript, the system allows users to report lost or found objects with image uploads and relevant metadata. The backend intelligently matches reported items using visual similarity and stored descriptors, significantly reducing the time and effort involved in searching. Unlike traditional methods that rely on manual matching or generalized public announcements, this AI-driven system makes the process faster, more accurate, and user-friendly.

CHAPTER - 8

8. REFERENCES

1. Russell, S. J., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th Edition). Pearson Education
2. **Goodfellow, I., Bengio, Y., & Courville, A.** – *Deep Learning*, which helped in understanding neural networks and deep learning techniques used in visual recognition systems.
3. **Welling, M.** – *A First Encounter with Machine Learning*, which offered beginner-friendly insights into algorithmic design and data processing.
4. **Krizhevsky, A., Sutskever, I., & Hinton, G. E.** – *ImageNet Classification with Deep Convolutional Neural Networks*, which inspired the integration of image classification techniques.
5. **Redmon, J., & Farhadi, A.** – *YOLOv3: An Incremental Improvement*, used for fast object detection strategies in lost item recognition.
6. **Mozilla Developer Network (MDN)** – Comprehensive documentation for HTML, CSS, JavaScript, and IndexedDB integration in the frontend.
7. **W3Schools** – Quick syntax references and UI design tips for web development.
8. **Bootstrap Documentation** – For creating responsive, mobile-friendly user interfaces in the login and dashboard pages.
9. **OpenCV Documentation** – Used for implementing computer vision functions such as image preprocessing and object detection.
10. **Google Developers Docs** – For enhancing browser compatibility and UI performance optimizations.
11. **Stack Overflow** – For troubleshooting and community support during the implementation of backend logic and JavaScript interactions.
12. **GitHub Repositories** – Reference to open-source AI models and code snippets for integrating IndexedDB and AI-based image recognition.
13. **Kaggle Datasets** – Source of item images and datasets for training and testing the AI model used in the project.