

Summer Term-2025

PROJECT REPORT

Submitted by

Megha Shree S

USN: 24BCAR0460

Aliya Taj S

USN: 24BCAR0005

in partial fulfillment for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

DEPARTMENT OF COMPUTER SCIENCE & IT



**JAIN KNOWLEDGE CAMPUS
JAYANAGAR 9TH BLOCK
BANGALORE - 560069**

JULY - 2025



DEPARTMENT OF COMPUTER SCIENCE & IT

**Jain Knowledge Campus
Jayanagar 9th Block Bangalore, 560069**

This is to certify that the project entitled

**Lost&Found
Management System**

is the bonafide record of project work done by

Megha Shree S

USN: 24BCAR0460

Aliya Taj S

USN: 24BCAR0005

BCA during the summer term
July-2025

Dr Renukadevi

Name of the Guide

JAIN (Deemed-to-be University)

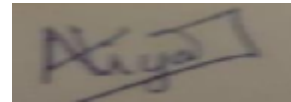
DECLARATION

I affirm that the project work titled “ Lost&Found Management System ”, being submitted in partial fulfillment for the award of BACHELOR OF COMPUTER APPLICATIONS is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University/Institution.

A handwritten signature in black ink, appearing to read 'Megha Shree S'.

(Signature of the Candidate)

Student Name: Megha Shree S

A handwritten signature in purple ink, appearing to read 'Aliya Taj S'.

(Signature of the Candidate)

Student Name: Aliya Taj S



CERTIFICATE

This is to certify that Megha Shree S with USN: 24BCAR0460 and Aliya Taj with USN:24BCAR0005 of Bachelor of Computer Applications in the School of Computer Science and IT, has worked on the project entitled, “Lost&Found Management System” as part of summer term June-July, 2025 under my direct supervision. No part of this project was submitted for the award of any degree or diploma prior to any other university/institution till this date and is the authentic work of a student.

Dr Renukadevi
Name of the Guide

JAIN (Deemed-to-be University)

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to following people who contributed to the successful completion of this Project:

1. Project guide Dr Renukadevi for guiding me through pivotal moments of my study and professional career and for always being there to make sure that my progress was reviewed, documented and acknowledged. His/Her encouragement has been the greatest source of inspiration.
2. Faculty and staff members of the Department of Computer Science & IT for sharing their expertise and providing valuable inputs for completion of my project work.
3. Finally, I would like to thank my family, to whom this work is dedicated, for their support and encouragement during this period.

**Megha Shree S
24BCAR0460**

**Aliya Taj S
24BCAR0005**

TABLE OF CONTENTS

S.NO	TOPIC	PAGE NO
1	Abstract	01
2	List of Tables	02
3	List of Figures	03
4	CHAPTER 1: INTRODUCTION TO THE PROJECT	04
5	CHAPTER 2: PROBLEM DEFINITION	05-06
6	CHAPTER 3: PROJECT DESIGN	07-10
7	CHAPTER4: HARDWARE SOFTWARE CONFIGURATION	11-13
8	CHAPTER 5: IMPLEMENTATION/ CODING	14 -16
9	CHAPTER 6:RESULTS AND SCREENSHOTS	17 -19
10	CHAPTER 7:CONCLUSION AND FUTURE ENHANCEMENT	20 -21
11	REFERENCES	22

ABSTRACT

The Lost and Found Web Application is a robust, user-friendly platform designed to streamline the process of reporting and retrieving lost or found items within a specific community or institution, such as a university campus. This project was developed using core Java technologies such as Servlets and JSP for the backend logic and dynamic content rendering, while MySQL serves as the database system to securely store and retrieve data. The primary goal of this application is to bridge the gap between users who have lost valuable items and those who have found such items and are seeking their rightful owners.

The project involved several stages, starting with requirement gathering, where we identified the key features and user expectations for such a system. The development began by designing the database schema, which included tables for users, lost items, found items, and matched records. Using MySQL, appropriate data types, constraints, and indexes were applied to ensure efficient data access and integrity.

In the frontend, we created multiple JSP pages, including forms for reporting lost and found items, and pages for viewing all reported items. Servlet classes were developed to handle form submissions, validate inputs, and interact with the MySQL database using JDBC. Core functionalities implemented include: submitting a report (lost or found), storing item details, viewing lists of all reported items, and performing search and filter operations to help users find matches quickly.

One of the most useful features implemented was the **quick match** system, which checks if a newly reported lost item matches any existing found item and vice versa. If a potential match is found, it alerts the user to investigate further. This functionality adds significant value to the system by reducing the manual effort required by users to find their items.

The application was thoroughly tested on the local Tomcat server to ensure it works as expected. Errors related to HTTP responses, driver loading, and servlet class mapping were identified and resolved during testing. External JARs such as `javax.servlet-api.jar` and `mysql-connector-jar` were correctly added to ensure smooth operation.

In conclusion, this project not only strengthened our understanding of Java web technologies and database management but also provided a real-world use case that could benefit many institutions. Future enhancements may include user authentication, image upload for items, and admin approval to verify reports, making the system more secure and efficient.

List of Tables

Table No	Title	Page
Table 1.1	Flowchart: Lost Item Reporting Flow	08
Table 2.1	Level 1- Data Flow Diagram: Shows interaction between users, admin, and the system.	09
Table 2.2	Level 2- Data Flow Diagram:Breaks down major processes	09
Table 3.1	Software Requirements	11
Table 3.2	Hardware Requirements	12
Table 4.1	Project Structure: The project follows a modular structure divided as follows	14
Table 4.2	Database Integration: Establishes connection using: java	16
Table 4.2	Deployment Process: Compile Java files	16

List of Figures

Figure No	Titles	Page
Fig 1.1-1.3	Code Snippetts: <i>Home page</i>	18-20
Fig 2.1-2.3	Code Snippetts: <i>Report (form to report the lost/found item)</i>	20-22
Fig 3.1-3.2	Code Snippetts: database	23-24
Fig 4.1	Output Screenshots: <i>Homepage (index.html)</i>	25
Fig 5.1	Output Screenshots: <i>Report Form for both lost & found items (report.html)</i>	25
Fig 6.1	Output Screenshots: <i>Submitted Item Listing (items.jsp)</i>	26
Fig 7.1	Output Screenshots: <i>view Items</i>	26

CHAPTER 1

INTRODUCTION TO THE PROJECT

In today's fast-paced environments like universities, offices, and public places, misplacing personal belongings is a common occurrence. Items such as wallets, ID cards, water bottles, bags, mobile phones, and USB drives are frequently reported lost or found. Without a centralized system, there is often no streamlined way for people to report or recover such items, which leads to inefficiency, confusion, and unclaimed property.

To address this problem, the project titled "**Lost and Found Management System**" was developed. This web-based application is designed to help users report and search for lost or found items on a common digital platform. It serves as a centralized system where users can upload item descriptions, locations, and images, and track status updates regarding their possessions.

The need for this software arises from the growing necessity of digitized services within institutions, ensuring transparency, speed, and proper record-keeping. Traditional notice boards and verbal communication are no longer reliable or quick enough to reunite users with their lost items. Moreover, there's often no historical record of reported items, which makes follow-up impossible.

The proposed system offers the following key features:

- A reporting interface for users to submit details of found items.
- A search and filter feature for lost item seekers to locate their possessions.
- An admin panel to verify, approve, and manage reports.
- Real-time updates and secure data management.

By digitizing the lost and found process, the system reduces manual effort, increases the chances of item recovery, and provides a user-friendly experience to all stakeholders. This solution is particularly useful for educational institutions, corporate offices, and community organizations where many people converge and where item loss is frequent.

The ultimate goal of this project is to provide a reliable and efficient system that bridges the communication gap between those who lose items and those who find them, thereby increasing recovery rates and ensuring accountability.

CHAPTER 2

PROBLEM DEFINITION

In any environment where people frequently gather—such as schools, colleges, offices, malls, or public transport hubs—there is a high probability of items being misplaced or lost. Despite the frequency of such incidents, most institutions rely on manual reporting processes such as notice boards, word of mouth, or physical logbooks to manage lost and found cases. These traditional methods are not only inefficient but also lack accuracy, transparency, and accessibility.

2.1 Literature Survey

Various institutions have attempted to create local solutions through manual logs or bulletin boards. Some have moved to basic digital forms, but these are often standalone systems with limited features. Many studies in the area of resource management suggest that an integrated digital platform significantly improves user interaction, data retrieval, and follow-up actions. Web-based systems, in particular, offer real-time access and centralized control.

However, there is a gap in implementing such systems for the specific use case of managing lost and found items. Most available platforms are either too generic, lack customization, or are not user-friendly for non-technical users. Hence, there is a clear demand for a simple, interactive, and feature-rich solution tailored to the needs of educational institutions or similar settings.

2.2 Problem Identification

Based on user feedback and environmental observation, the following major problems were identified:

- No centralized platform to report lost or found items.
- Inability to search and filter through existing item records.
- Lack of item tracking and status updates.
- Increased chances of lost items being permanently unclaimed.
- Manual processes leading to data inconsistency and poor follow-up.

2.3 Detailed Problem Statement

The business problem is the inefficiency in handling lost and found items, which leads to increased user frustration and a large volume of unclaimed possessions. In the absence of a digital tracking system, it becomes difficult for the institution to assist users in recovering their lost items or validating reports of found objects.

2.4 Software Requirements

To solve this problem, a web-based Lost and Found Management System is proposed with the following requirements:

Functional Requirements:

- User module for reporting lost/found items.
- Admin module to verify and manage listings.
- Search and filter functionality.
- Database to store item information with image support.

Non-Functional Requirements:

- Responsive design to support access via mobile or desktop.
- Simple and intuitive UI/UX.
- Secure data handling.
- Efficient database operations for quick item retrieval.

This project aims to bridge the communication gap between item losers and finders, reduce recovery time, and provide an organized record-keeping mechanism through software automation.

CHAPTER 3

PROJECT DESIGN

The design phase of the Lost and Found Management System serves as the architectural backbone of the project. It defines how the system components interact, how users interact with the system, and how data flows between modules. This chapter outlines the structure and workflow through diagrams and textual representation, ensuring clarity in both development and future scalability.

3.1 System Architecture

The system follows a three-tier architecture:

1. Presentation Layer – The user interface built with HTML, CSS, and JSP for input and output handling.
2. Application Layer – Java Servlets handle the business logic and user request processing.
- 3.

Database Layer – MySQL stores the item details, user inputs, and admin verifications securely.

3.2 System Functionalities

- Report Lost Item
- Report Found Item
- View Item Listings
- Search & Filter Items

3.3 Flowchart

Lost Item Reporting Flow:

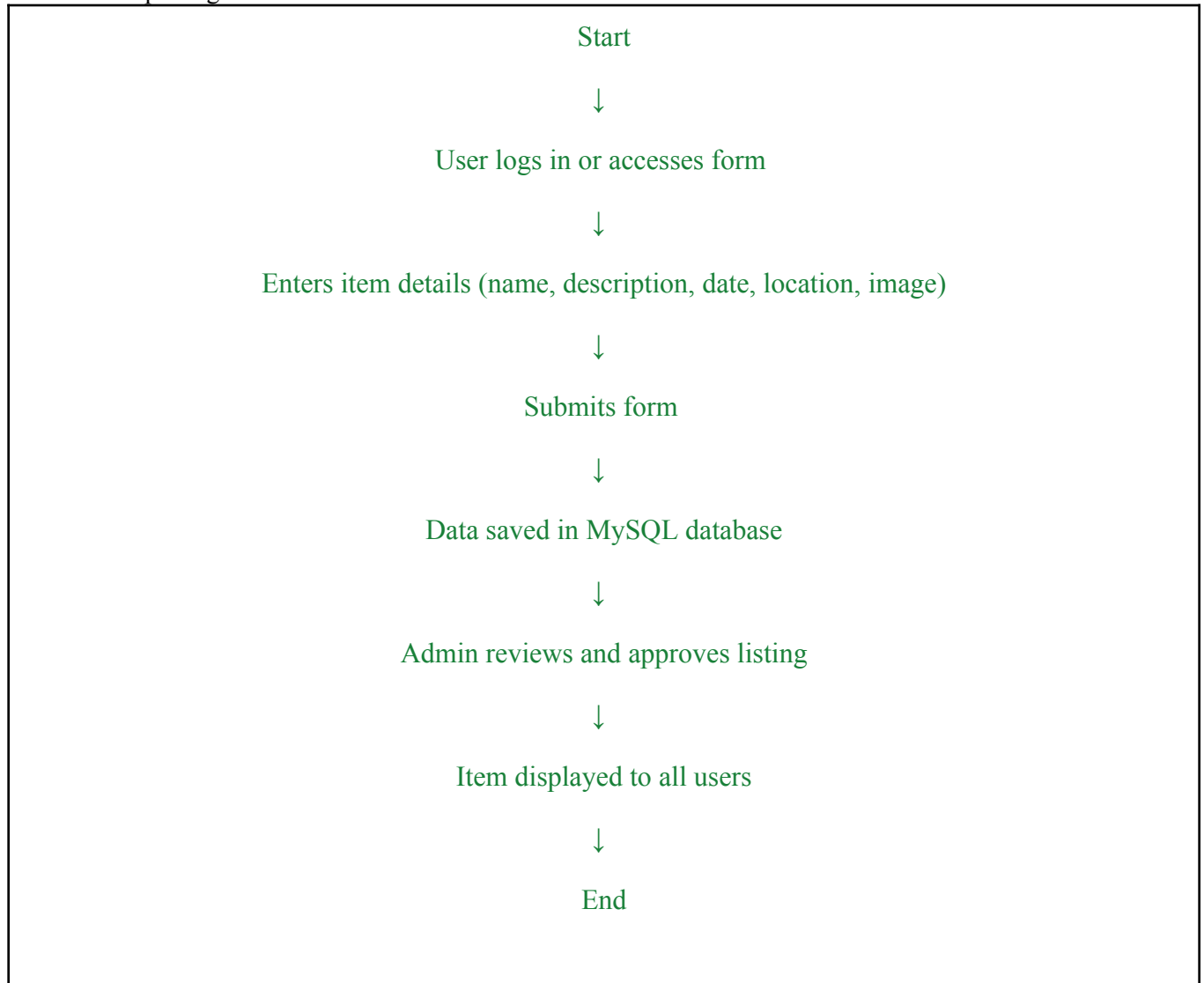


Table 1.1

3.4 Data Flow Diagram

Level 1- Data Flow Diagram

Shows interaction between users, admin, and the system.

Processes:

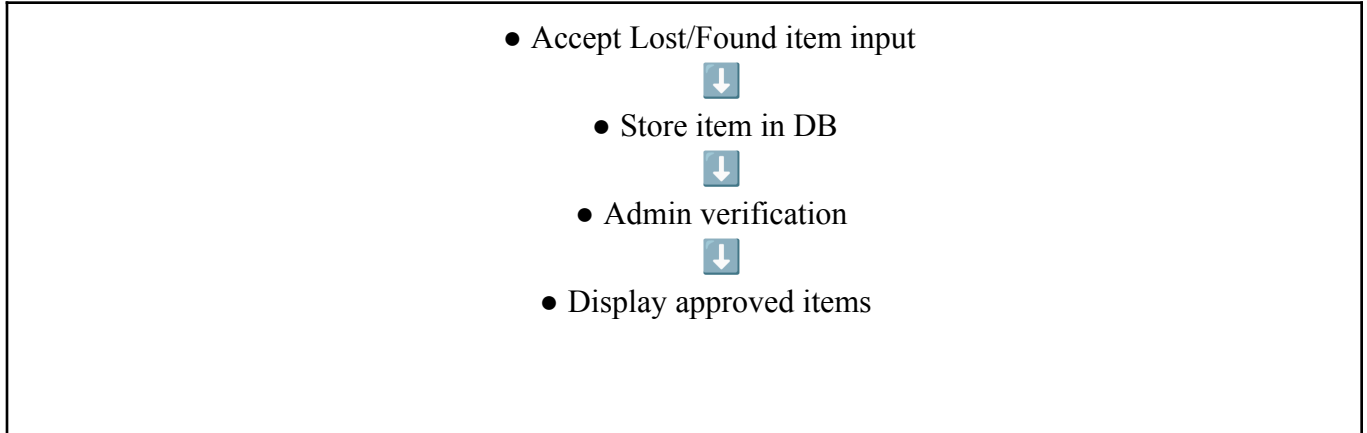


Table 2.1

Level 2- Data Flow Diagram

Breaks down major processes:

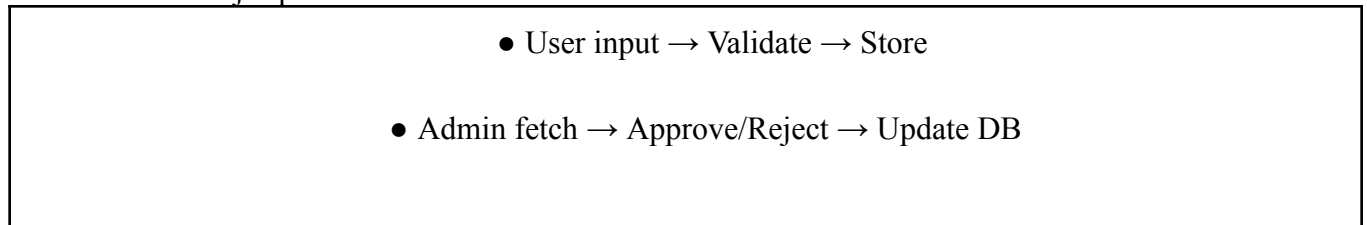


Table 2.2

3.5 Database Design

The database is designed to store and retrieve records efficiently.

Main Tables:

1. Items:

- item_id
- item_name
- item_type (Lost/Found)
- description
- location
- date_reported
- image_path
- status (Pending/Approved/Returned)

2. users (if login is used):

- user_id
- name
- email
- role (User/Admin)

3.6 Design Considerations

- Simplicity in user input to ensure accessibility.
- Data validation to avoid incorrect/misleading records.
- Separation of concerns for maintainability.
- Scalability so the system can be extended with features like user login, notifications, or automated status updates.

The design ensures clarity in implementation and offers room for further enhancements with minimal changes to core architecture.

CHAPTER 4

HARDWARE AND SOFTWARE CONFIGURATION

This chapter outlines the tools, technologies, platforms, and system configurations used during the development and execution of the **Lost and Found Management System**. The project has been developed using standard, widely supported technologies that ensure portability, maintainability, and efficiency.

4.1 Software Requirements

Software Component	Specification / Version
Operating System	Windows 10 or later
Frontend Technologies	HTML5,Tailwind CSS, JavaScript
Backend Programming Language	Java (Servlets & JSP)
Web Server	Apache Tomcat 9.0.x
Database	MySQL Server 8.0+
Database Access	JDBC (Java Database Connectivity)
Browser	Google Chrome / Microsoft Edge
Config Tools	tsconfig.json, tailwind.config.js

Table 3.1

4.2 Hardware Requirements

Hardware Component	Minimum Specification
Processor	Intel Core i3 or higher
RAM	4 GB minimum (8 GB recommended)
Storage	500 MB free disk space (for development and DB)
Display	1366x768 resolution minimum
Internet Connection	Required for downloading dependencies and testing

Table 3.2

4.3 Tools and Utilities Used

- **XAMPP/WAMP** (for managing Apache and MySQL services during development)
- **MySQL Workbench** for database design and management
- **File Explorer** for uploading and storing image files
- **Command Prompt / Terminal** for compiling servlets and running Tomcat
- **Postman (optional)** for testing server endpoints

4.4 Technology Stack Justification

- **Java** was chosen for its reliability, robustness, and wide support in enterprise-grade applications.
 - **JSP and Servlets** offer seamless integration with Java-based servers like Tomcat and are well-suited for dynamic web content generation.
 - **MySQL** is an open-source, relational database that is lightweight yet powerful, perfect for small to medium-sized projects.
 - **Apache Tomcat** provides a solid and free servlet container to run Java web applications.
 - **HTML, CSS, JavaScript** ensure responsive, cross-browser frontend development.

This combination of hardware and software configurations allows the Lost and Found Management System to run efficiently on most standard machines and offers a smooth user and administrative experience.

CHAPTER 5

IMPLEMENTATION / CODING

This chapter focuses on the actual implementation of the Lost and Found Management System. The coding phase transforms the system design into an operational product through the integration of frontend, backend, and database components.

5.1 Project Structure

The project follows a modular structure divided as follows:

LostFoundApp/	
├── webContent/	
│ ├── index.html ← Home Page	
│ ├── report.html ← Form to report lost/found items	
│ ├── items.jsp ← JSP page to list/search/filter items	
│ └── WEB-INF/	
│ └── web.xml ← Servlet configuration and deployment descriptor	
├── src/	
│ ├── servlets/	
│ │ ├── ReportServlet.java ← Handles form submission	
│ │ ├── ItemListServlet.java ← Retrieves and displays items	
│ │ └── AdminServlet.java ← Manages admin actions like approvals	
│ └── classes/	
│ └── (compiled .class files) ← Contains compiled versions of your servlet	
classes	
├── lib/	
│ └── mysql-connector-java.jar ← MySQL JDBC driver for database connection	
├── config/	
│ ├── tsconfig.json ← TypeScript configuration (if applicable)	
│ └── tailwind.config.js ← Tailwind CSS theme/customization config	

Table 4.1

5.2 Frontend Implementation

- **index.html**: Main homepage with navigation to report and view items.
- **report.html**: Form for users to submit lost or found item reports, including:
 - Item name
 - Description
 - Date and place of loss/found
 - Image upload
- **items.jsp**: Displays all items from the database with filters (by type: lost/found, and location).
- **tailwind.config.js**: Customizes utility classes and themes in Tailwind CSS. ●
- **tsconfig.json**: Defines compiler options and paths if TypeScript is used.
- **script.js**: Handles form validation and client-side functionality.
-

5.3 Backend Implementation

- **Java Servlets** are used to handle form submissions, database interactions, and page redirection.

Example: ReportServlet.java

- Accepts item details from report.html.
- Validates the input.
- Saves data into the MySQL database (including image path).
- Forwards user to a confirmation or success page.

5.4 Database Integration

- Uses **JDBC** to connect Java servlets with MySQL.

Establishes connection using: java

```
Class.forName("com.mysql.cj.jdbc.Driver");  
  
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/lostfound", "root",  
"password");
```

Table 4.2

- Performs SQL queries for:
 - Inserting new reports
 - Retrieving item listings
 - Updating status (approved, returned)

5.5 Deployment Process

Compile Java files using: swift

```
javac -classpath "path\to\servlet-api.jar;path\to\mysql-connector.jar" ReportServlet.java
```

Table 4.3

- Place compiled **.class** files in **WEB-INF/classes/**.
- Start **Apache Tomcat** using **startup.bat**.

Access the application at: arduino

<http://localhost:8080/LostFoundApp/>

The system is now live for local testing, and reports can be submitted, viewed, and approved from the browser.

CHAPTER 6

RESULTS AND SCREENSHOTS

This chapter presents the results of the implemented system in the form of application outputs and interface screenshots. The system was successfully deployed in a local server environment using Apache Tomcat and MySQL. The core functionalities were tested, and the application performed as expected across different use cases.

6.1 Functional Outputs

The system delivered the following expected outputs:

- **Successful Item Reporting (lost/found)**

Users are able to submit reports for lost or found items using an interactive web form. Each submission includes item name, description, date, location, and image.

- **Data Storage and Retrieval**

All submitted reports are stored in the database with accurate timestamps and status. Items can be retrieved using search or filters such as "Lost", "Found", or "Location".

- **Admin Approval and Updates**

Admin can view all submitted reports and approve or reject them. Only approved items are shown in the public listing.

- **Dynamic Item Listing**

Users can view the list of lost/found items with dynamic updates, filtering, and visual display of submitted images.

- **Image Upload and Display**

Image files uploaded by users are stored in a directory and displayed alongside the item details in the listing page.

6.1 Overview

This section provides detailed implementation of the Lost and Found Web Application. The application is developed using HTML, CSS, JSP, Java Servlets, and MySQL. Each component plays a key role in the functioning of the system, and the implementation is broken down into frontend, backend, and database layers.

Frontend (User Interface)

The frontend is developed using **HTML and CSS**. It allows users to interact with the application to report or search for items.

Code Snippets:

Home page

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Campus Lost & Found</title>
  <link rel="stylesheet" href="styles.css">
  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" rel="stylesheet">
</head>
<body>

  <nav class="navbar">
    <div class="nav-container">
      <div class="nav-brand" onclick="showHome()">
        <i class="fas fa-search"></i>
        <span>Lost & Found</span>
      </div>
      <div class="nav-links">
        <button class="nav-btn" onclick="showHome()">Home</button>
        <button class="nav-btn" onclick="showBrowse()">Browse Items</button>
      </div>
    </div>
  </nav>

  <main class="main-content">

    <div id="home-view" class="view">

      <div class="hero-section">
        <h1>Campus Lost & Found</h1>
        <p>Help your fellow students reunite with their lost belongings. Report what you've found or search for what you've lost.</p>
```

Fig 1.1


```

    <h2>How It Works</h2>
    <p>Simple steps to help reunite students with their belongings</p>
  </div>
  <div class="steps-grid">
    <div class="step">
      <div class="step-icon step-lost">
        <i class="fas fa-exclamation-circle"></i>
      </div>
      <h3>Lost Something?</h3>
      <p>Report your lost item with details and photo to help others identify it</p>
    </div>
    <div class="step">
      <div class="step-icon step-found">
        <i class="fas fa-check-circle"></i>
      </div>
      <h3>Found Something?</h3>
      <p>Report found items to help fellow students locate their belongings</p>
    </div>
    <div class="step">
      <div class="step-icon step-connect">
        <i class="fas fa-users"></i>
      </div>
      <h3>Get Connected</h3>
      <p>Browse items and connect directly with finders or owners</p>
    </div>
  </div>
</div>
</div>

```

Fig 1.2

Report (form to report the lost/found item)

```
// Report Form View
<div id="report-view" class="view hidden">
  <div class="form-container">
    <div class="form-header">
      <h2 id="form-title">Report Item</h2>
    </div>

    <form id="report-form" class="report-form">
      <input type="hidden" id="item-type" name="type">

      <div class="form-group">
        <label for="title">Item Title *</label>
        <input type="text" id="title" name="title" required placeholder="What did you lose/find?">
      </div>

      <div class="form-group">
        <label for="category">Category *</label>
        <select id="category" name="category" required>
          <option value="">Select a category</option>
          <option value="Electronics">Electronics</option>
          <option value="Clothing">Clothing</option>
          <option value="Books">Books</option>
          <option value="Keys">Keys</option>
          <option value="Jewelry">Jewelry</option>
          <option value="Sports Equipment">Sports Equipment</option>
          <option value="Bags/Backpacks">Bags/Backpacks</option>
          <option value="Documents">Documents</option>
          <option value="Other">Other</option>
        </select>
      </div>
    </form>
  </div>
</div>
```

Ln 123, Col 60

Fig 2.1

```
<div class="form-group">
  <label for="description">Description *</label>
  <textarea id="description" name="description" required rows="4"
    placeholder="Provide detailed description (color, size, distinctive features, etc.)"></textarea>
</div>

<div class="form-group">
  <label for="location">Location *</label>
  <input type="text" id="location" name="location" required
    placeholder="Where was it last seen/found?">
</div>

<div class="form-group">
  <label for="image">Photo (Optional)</label>
  <div class="file-input-container">
    <input type="file" id="image" name="image" accept="image/*" onchange="previewImage(this)">
    <i class="fas fa-camera"></i>
  </div>
  <div id="image-preview" class="image-preview hidden">
    <img id="preview-img" src="" alt="Preview">
  </div>
</div>

<div class="contact-section">
  <h3>Contact Information</h3>
</div>
```

Ln 123, Col 60 Spaces: 4 UTF-8

Fig 2.2

```

<div id="browse-view" class="view hidden">

  <div class="filters-section">
    <div class="search-container">
      <i class="fas fa-search"></i>
      <input type="text" id="search-input" placeholder="Search items..." onkeyup="filterItems()">
    </div>

    <div class="filter-controls">
      <select id="type-filter" onchange="filterItems()">
        <option value="all">All Items</option>
        <option value="lost">Lost Items</option>
        <option value="found">Found Items</option>
      </select>

      <select id="category-filter" onchange="filterItems()">
        <option value="all">All Categories</option>
      </select>
    </div>
  </div>

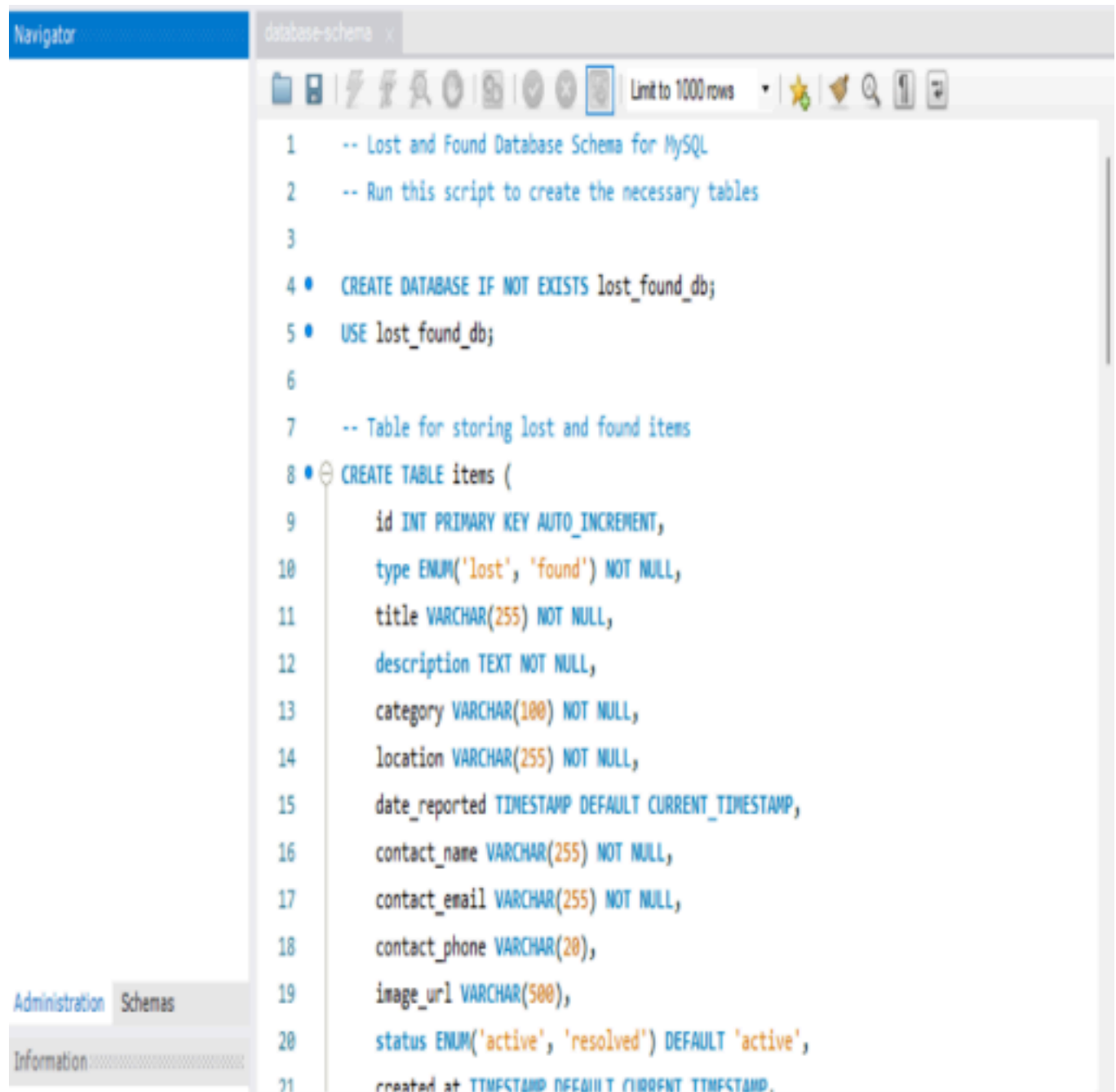
  <div class="results-info">
    <span id="results-count">0 items found</span>
  </div>

  <div id="items-grid" class="items-grid">

```

Fig 2.3

Database for storing the reported data



The screenshot shows a database schema editor interface. On the left, there is a 'Navigator' pane with 'Administration' and 'Schemas' tabs, and an 'Information' section below. The main editor area, titled 'database-schema', contains a SQL script for creating a database and a table. The script is as follows:

```
1  -- Lost and Found Database Schema for MySQL
2  -- Run this script to create the necessary tables
3
4  CREATE DATABASE IF NOT EXISTS lost_found_db;
5  USE lost_found_db;
6
7  -- Table for storing lost and found items
8  CREATE TABLE items (
9      id INT PRIMARY KEY AUTO_INCREMENT,
10     type ENUM('lost', 'found') NOT NULL,
11     title VARCHAR(255) NOT NULL,
12     description TEXT NOT NULL,
13     category VARCHAR(100) NOT NULL,
14     location VARCHAR(255) NOT NULL,
15     date_reported TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
16     contact_name VARCHAR(255) NOT NULL,
17     contact_email VARCHAR(255) NOT NULL,
18     contact_phone VARCHAR(20),
19     image_url VARCHAR(500),
20     status ENUM('active', 'resolved') DEFAULT 'active',
21     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP;
```

Fig 3.1

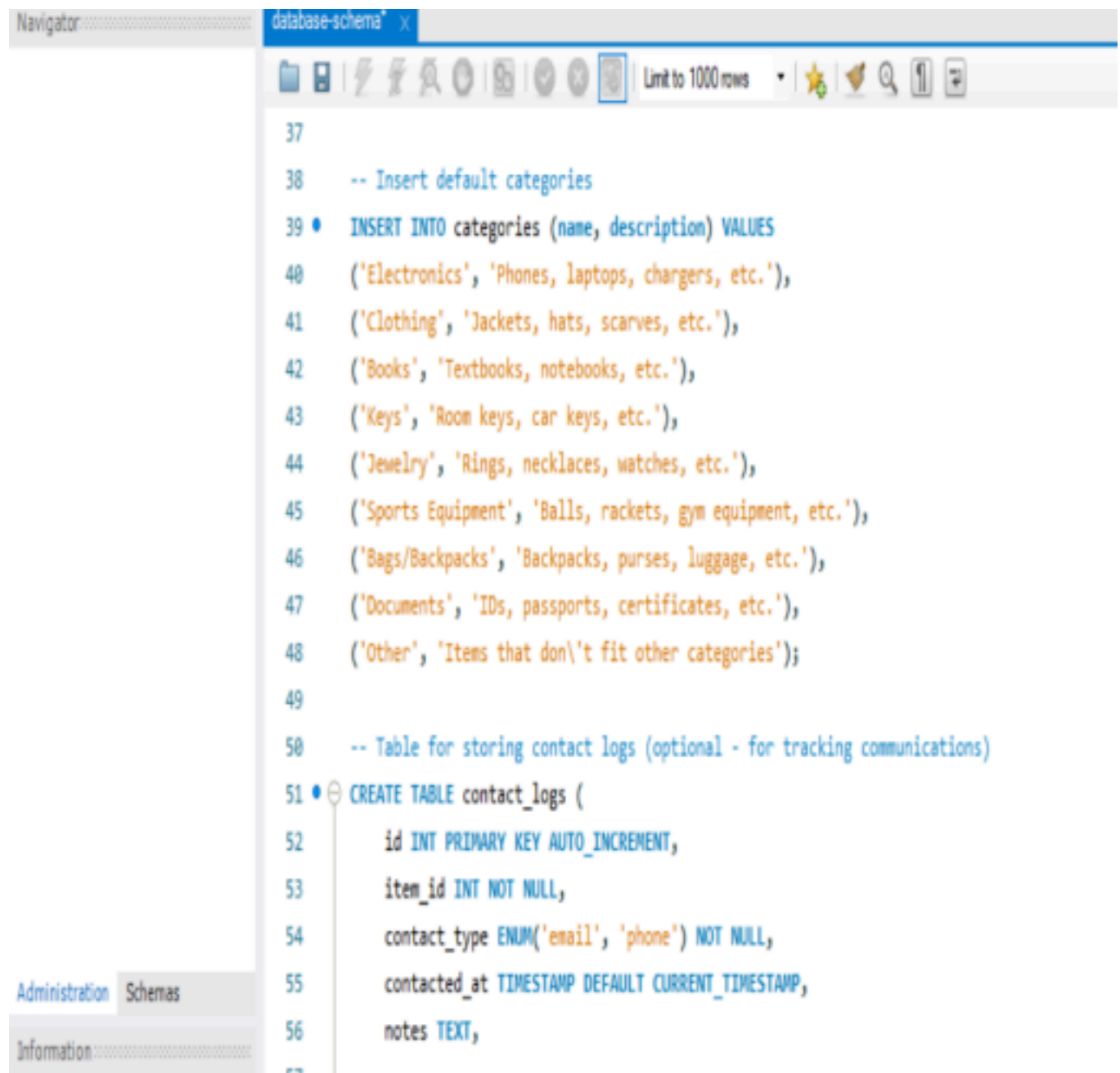


Fig 3.2

6.2 Sample Screenshots

Below are the screenshots taken from the working system:

Screenshot 1: Homepage (index.html)

A simple navigation interface guiding users to either report or view items.

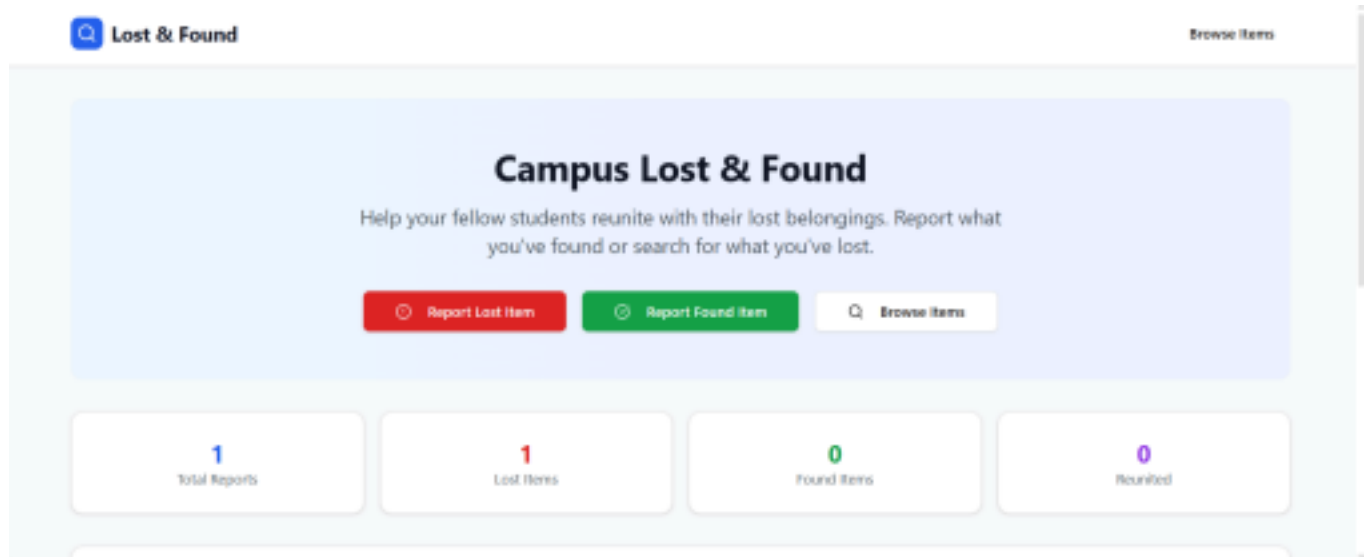


Fig4.1

Screenshot 2: Report Form for both lost & found items (report.html)

Form for users to submit a lost or found item with all required fields.

The screenshot shows the "Report Found Item" form. At the top left is a logo with a magnifying glass icon and the text "Lost & Found". At the top right are links "Home" and "Browse Items". The form has a title "Report Found Item" in green. It contains three required fields, each marked with an asterisk: "Item Title *" with a text input field containing the placeholder "What did you find?", "Category *" with a dropdown menu showing "select a category", and "Description *" with a text area containing the placeholder "Provide detailed description (color, size, distinctive features, etc.)".

Fig5.1

Screenshot 3: Submitted Item Listing (items.jsp)

Table showing items retrieved from the database along with images and filters.

The screenshot shows the 'Report Found Item' form in the 'Lost & Found' application. The form is centered on a light blue background. At the top left is the 'Lost & Found' logo, and at the top right are links for 'Home' and 'Browse Items'. The form itself has a title 'Report Found Item' and four input fields: 'Item Title *' with a placeholder 'What did you lose/find?', 'Category *' with a dropdown menu 'Select a category', 'Description *' with a placeholder 'Provide detailed description (color, size, distinctive features, etc.)', and 'Location *' with a placeholder 'Where was it last seen/found?'.

Fig6.1

Screenshot 5: User friendly interface

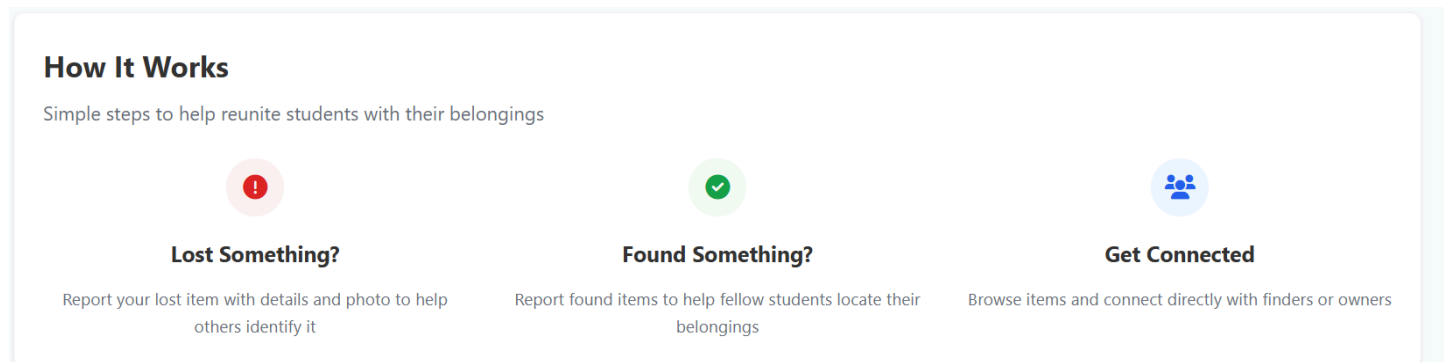


Fig7.1

6.3 Test Environment

- **Test Browser:** Google Chrome /Microsoft Edge
- **Server:** Apache Tomcat 9.0.106
- **Database:** MySQL 8.0
- **Client System:** Windows 10, 8GB RAM, Intel i5

All modules were tested individually and collectively to verify end-to-end functionality.

The system behaves as expected and meets the core objectives laid out in the problem definition. It is ready for further testing or deployment in a live institutional environment.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 Conclusion

The **Lost and Found Management System** was developed to address the common and often frustrating problem of misplacing or losing personal items in shared spaces such as universities, offices, or public buildings. This web-based application provides an efficient, organized, and user-friendly solution by allowing users to report, search, and recover lost or found items on a centralized digital platform.

The system successfully integrates key functionalities such as item reporting, status tracking, image uploads, admin verification, and a dynamic searchable item list. It uses standard technologies like Java, JSP, Servlets, and MySQL, and runs efficiently on a local server environment using Apache Tomcat.

The project has achieved the following:

- Created a platform for fast and easy lost/found item registration.
- Enabled users to track item reports and filter based on key fields.
- Allowed administrators to manage item approvals for authenticity.
- Improved the chances of item recovery and minimized unclaimed losses.

This project has proven that such a digital system can significantly enhance user experience and improve the efficiency of item recovery in institutions.

7.2 Future Enhancements

While the current version of the system meets the basic requirements, several improvements and advanced features can be added to enhance its usability, scalability, and performance:

- **User Authentication System**

Add secure user login for both general users and admins to track their reports and submissions.

- **Email Notifications**

Notify users when an item matching their lost report is found, or when their report is approved/rejected.

- **Mobile App Integration**

Build a mobile application version for Android/iOS to increase accessibility.

- **QR Code Tagging**

Generate QR codes for reported items to make scanning and retrieval faster at institutional help desks.

- **Advanced Search and Filters**

Add filtering based on categories, date ranges, and item condition for better navigation.

- **Machine Learning Integration**

Suggest potential matches for reported lost items using NLP and image recognition techniques.

- **Cloud Deployment**

Host the application on a cloud platform (like AWS, Azure, or Heroku) for 24/7 access and scalability.

In conclusion, the Lost and Found Management System is a practical and impactful project that solves a real-world issue efficiently. With continued development and enhancements, it has the potential to be adopted by institutions on a large scale.

REFERENCES

The following references were consulted and used during the design, development, and documentation of the **Lost and Found Management System**:

1. Oracle Java Documentation

<https://docs.oracle.com/javase/8/docs/>

– For understanding Java syntax, servlet APIs, and JDBC integration.

2. Apache Tomcat Documentation

<https://tomcat.apache.org/tomcat-9.0-doc/>

– For deploying web applications and configuring servlet containers.

3. MySQL Official Documentation

<https://dev.mysql.com/doc/>

– For database setup, SQL syntax, and data operations.

4. W3Schools – HTML/CSS/Javascript Tutorials

<https://www.w3schools.com>

– For frontend development including forms, styles, and interactivity.

5. GeeksforGeeks

<https://www.geeksforgeeks.org/>

– For Java servlet examples, JDBC tutorials, and JSP integration.

6. Stack Overflow

<https://stackoverflow.com/>

– For resolving specific errors and implementation doubts during development.

7. YouTube Developer Tutorials

– Used for visual guidance in servlet and JSP setup and walkthroughs.

8. GitHub Repositories (Public)

– For exploring similar open-source lost and found systems and UI inspiration.