



School of Computer Science, UPES, Dehradun

Database Management System

B.Tech – 3rd Semester

PROJECT REPORT FILE

CAMPUS LOST AND FOUND MANAGEMENT

Submitted to:

Prof. Kalluri Shareef Babu

Submitted by:

Shekhar Singh Panwar &
Krishna Baura

Batch: 29

SAP ID: 590016074 &
590015488

The screenshot shows the homepage of a Campus Lost & Found website. At the top, there is a dark blue header with the text "CAMPUS" and "Lost & Found". On the left is a logo consisting of three colorful, curved lines. On the right is a user icon. Below the header, there is a row of five small images showing lost items: a pen, a black leather wallet, a racket, a red pen, and a spiral notebook. The main content area is titled "Lost items" and contains four cards, each representing a lost item:

- Pen**: A red pen. Description: "Stationary • Lost at 1st floor Block 1". Buttons: "View Details" (blue) and "Claim this item" (dark grey).
- Racket**: A tennis racket. Description: "Sports • Lost at Block 1". Buttons: "View Details" (blue) and "Claim this item" (dark grey).
- Notebook**: A spiral-bound notebook. Description: "Stationary • Lost at Block 3". Buttons: "View Details" (blue) and "Claim this item" (dark grey).
- Watch**: A wristwatch. Description: "Accessories • Lost at Volley ball court". Buttons: "View Details" (blue) and "Claim this item" (dark grey).

- **This is the home page of our website.**
- **Here we can View Details of a particular item.**
- **Request to claim an item.**
- **And at top right, we have an admin login button, through which admins can login and manage the items.**

[← Home](#)



Racket

Sports • Lost at Block 1

No description provided.

Date lost: 2025-12-04 • Status: lost

[Claim this item](#)

- This is the item details page.
- It simply shows the details of item provided when adding.
- Here we can request to claim an item.

[← Home](#)



Racket
Filing a claim will store your details for in-person verification.

Your full name *

SAP ID *

Course (optional)

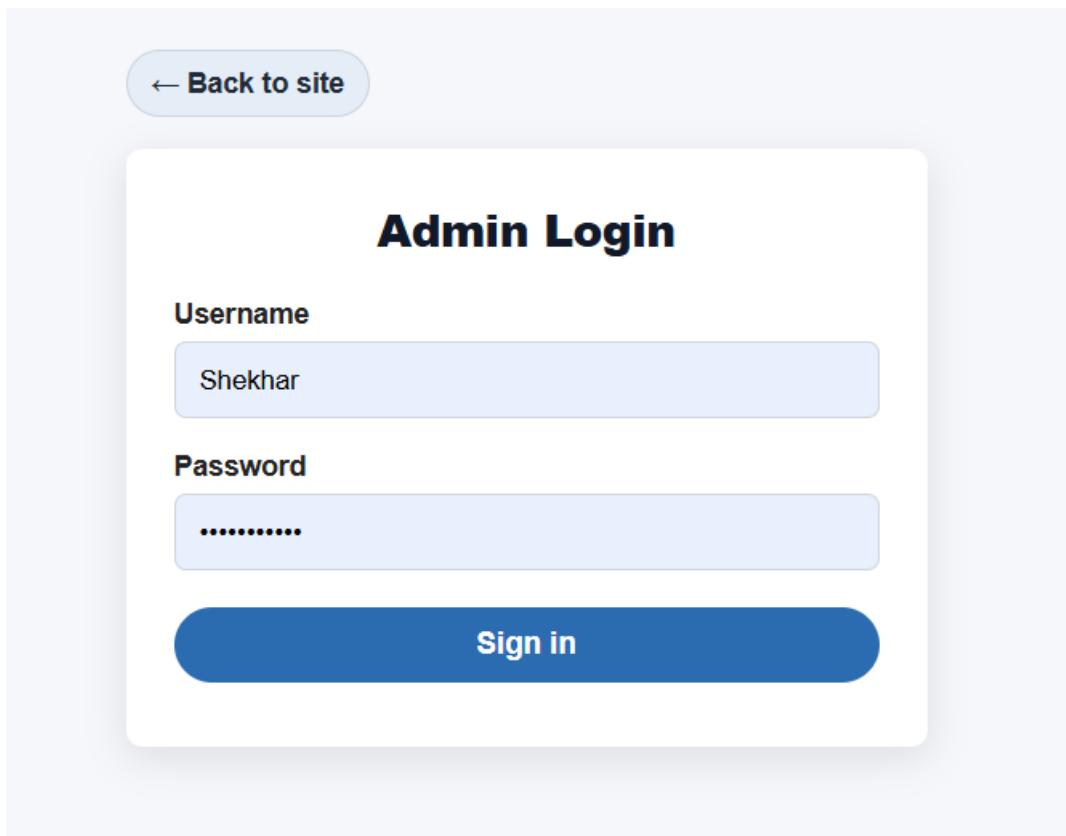
Short message / proof (optional)

Upload proof (JPG/PNG/PDF, max 5MB) (optional)

Choose File No file chosen

Submit Claim

- This is how the claim form looks like.
- We need to fill our details if we want to request to claim a particular item.
- We can also provide any proof here for convenience.



The screenshot shows a login interface titled "Admin Login". At the top left is a "Back to site" button. The main title "Admin Login" is centered above two input fields: "Username" and "Password". The "Username" field contains the text "Shekhar". The "Password" field contains several dots ("....."). A large blue "Sign in" button is at the bottom.

- This is admin login page.
- We can login here using our Username and Password as admin.

The screenshot shows the Admin Dashboard interface. At the top left is a 'Back to portal' button. The top right features four buttons: 'Manage Items' (blue), 'Manage Claims' (dark blue), 'Audit Logs' (light blue), and 'Logout' (red). The dashboard has two main sections: 'Total items' (8) and 'Pending claims' (3). Below these are two tables: one for 'Recently added items' and another for 'Recently added claims'. The 'Recently added items' table lists six items with their IDs, names, and addition dates:

ID	Item	Added
12	Pen	2025-12-07 16:21:41
11	Racket	2025-12-06 17:00:58
10	Notebook	2025-12-04 13:18:32
8	Watch	2025-12-03 13:22:56
7	Watch	2025-11-27 12:44:47
2	Samsung S24 Phone	2025-11-26 19:13:01

- This is the Admin Dashboard page.
- Here admin can view recently added items.
- He can also navigate to Manage Items, Manage Claims, Audit logs page from here.
- When the work is finished, admin can simply click on Logout.

[← Back to Dashboard](#)

Manage Items

Admin: Shekhar

+ Add Item

ID	Image	Item	Category	Location	Date Lost	Status	Added by	Actions
12		Pen	Stationary	1st floor Block 1	2025-12-03	lost	Shekhar	Edit Delete Mark claimed
11		Racket	Sports	Block 1	2025-12-04	lost	Shekhar	Edit Delete Mark claimed
10		Notebook	Stationary	Block 3	2025-11-29	lost	Shekhar	Edit Delete Mark claimed
8		Watch	Accessories	Volley ball court	2025-11-27	lost	Shekhar	Edit Delete Mark claimed
7		Watch	Accessories	Volley ball court	2025-11-27	claimed	Kalluri Shareef Babu	Edit Delete
		Black Leather	Accessories	Library - 2nd floor	2025-11-	...	Shekhar	Edit Delete Mark claimed

- This is the Manage Items page.
- Here admin can Edit, Delete, or Mark an item as claimed.
- He can also add items from this page.

Add New Item

Fill in the details for the lost item. You can upload a photo if available.

Item name *

Description

Category

Location lost

Date lost

 dd-mm-yyyy

Image (JPG/PNG, max 3MB) (optional)

[Choose File](#) No file chosen

[Add item](#) [Cancel](#)

- This is the form to add new item into the database.

[← Back to Dashboard](#)

Manage Claims

Admin: Shekhar

Claim ID	Item	Claimer	SAP ID	Course	Submitted	Status	Actions
6	Samsung S24 Phone (id: 2)	Sam	54964598	BSC	2025-12-06 00:50:25	pending	View Approve Reject
5	Watch (id: 8)	Someone	554899849	B.tech	2025-12-06 00:45:53	pending	View Approve Reject
4	Black Leather Wallet (id: 1)	Marut	590016548	B.tech	2025-12-05 09:48:26	pending	View Approve Reject
2	Watch (id: 7)	Aadi	590016077	B.tech	2025-11-27 12:46:22	approved	View

- This is the manage claims page.
- Here admin can View, Approve, or Reject any claims requests from students.

[← Back to dashboard](#)

Audit Logs

Showing latest 500 audit entries

Filter by action:

Time	Action	Description	Admin	ID
2025-12-07 16:21:41	item_added	Item added: Pen (id=12) by admin_id=1	Shekhar	34
2025-12-07 16:21:41	item_added	Added item: Pen (id=12)	Shekhar	35
2025-12-06 17:00:58	item_added	Item added: Racket (id=11) by admin_id=1	Shekhar	32
2025-12-06 17:00:58	item_added	Added item: Racket (id=11)	Shekhar	33
2025-12-06 15:02:39	item_deleted	Deleted item id=9	Shekhar	31
2025-12-06 00:50:25	claim_submitted	Claim submitted: claim_id=6 for item_id=2 by Sam	-	30
2025-12-06 00:45:53	claim_submitted	Claim submitted: claim_id=5 for item_id=8 by Someone	-	29
2025-12-05 09:48:26	claim_submitted	Claim submitted: claim_id=4 for item_id=1 by Marut	-	28
2025-12-04 13:18:32	item_added	Item added: Notebook (id=10) by admin_id=1	Shekhar	26
2025-12-04 13:18:32	item_added	Added item: Notebook (id=10)	Shekhar	27
2025-12-04 13:16:26	claim_status_changed	Claim status changed: claim_id=3 item_id=9 from pending to approved	Shekhar	22

- This is the audit logs page.
- Here admin can view the logs of activities done in the website.

Report File

Title of the Project

Campus Lost & Found Management System with Claim Workflow Using MySQL

Abstract

Lost items are a very common issue inside large college campuses, where hundreds of students move between classes, labs, libraries, and cafeterias. Traditional methods of tracking lost items — such as physical registers or notice boards — are slow, inefficient, and prone to errors.

This project aims to build a web-based Lost & Found Management System that allows:

- **Students to view recently found items**
- **Users to submit claim requests for items**
- **Admins to add, edit, or delete items**
- **Admins to approve or reject claim requests**
- **Automatic audit logs using MySQL triggers**
- **All data stored securely in a MySQL database**

The system is built using:

- **PHP (backend)**
- **HTML/CSS/JavaScript (frontend)**
- **MySQL (database)**
- **XAMPP (local server)**

This project demonstrates practical implementation of:

- **Database design**
- **SQL queries (including joins, filters, and conditions)**
- **Prepared statements**
- **MySQL triggers**
- **File handling**
- **Structured workflow management**

Introduction

A Lost & Found system helps students locate items misplaced across the campus and prevents permanent loss. Traditionally, managing such items requires manual registers and physical verification. This is not scalable when the number of items increases.

Our proposed system provides:

For General Users (No Login Required)

- View all recently found items
- View detailed information of each item
- Submit a claim request with proof

For Admins (Login Required)

- Add new found items
- Edit/update item details
- Approve or reject claims
- Mark items as claimed
- Check audit logs
- Manage multiple admin accounts

The system is simple, efficient, user-friendly, and demonstrates a real-world application of DBMS concepts.

Objectives of the Project

1. Create a centralized digital system to manage lost and found items.
2. Implement structured storage of item data in MySQL.
3. Allow users to claim items through a guided workflow.
4. Develop secure admin functionality with authentication.
5. Use prepared statements to prevent SQL injection.
6. Implement MySQL triggers to maintain automatic audit logs.
7. Demonstrate end-to-end DBMS usage (DDL, DML, queries, constraints, triggers).

System Requirements

Hardware Requirements

- Minimum 4 GB RAM

- Dual-core processor
- At least 500 MB free disk space

Software Requirements

- Windows / macOS / Linux
- XAMPP (Apache + MySQL + PHP)
- Web Browser (Chrome recommended)
- VS Code / Sublime / Notepad++ for editing code

Existing System

In many institutions, found items are handled like this:

- A guard or staff member maintains a notebook.
- Students manually check the register.
- Matching an item with the owner is time-consuming.
- No image records or digital proof are maintained.
- No tracking of claims or admin responsibility.

This manual process is outdated, slow, and unreliable.

Proposed System

The proposed Lost & Found Management System improves the process by:

✓ Digitizing all found items

Images, location, category, description, and date lost.

✓ Allowing online claim requests

Students can submit:

- Name
- SAP ID
- Course
- Optional proof

✓ Admin management

Multiple admins can:

- Add items
- Edit items
- Delete items
- Approve or reject claims
- Mark items as claimed

✓ Audit Logging

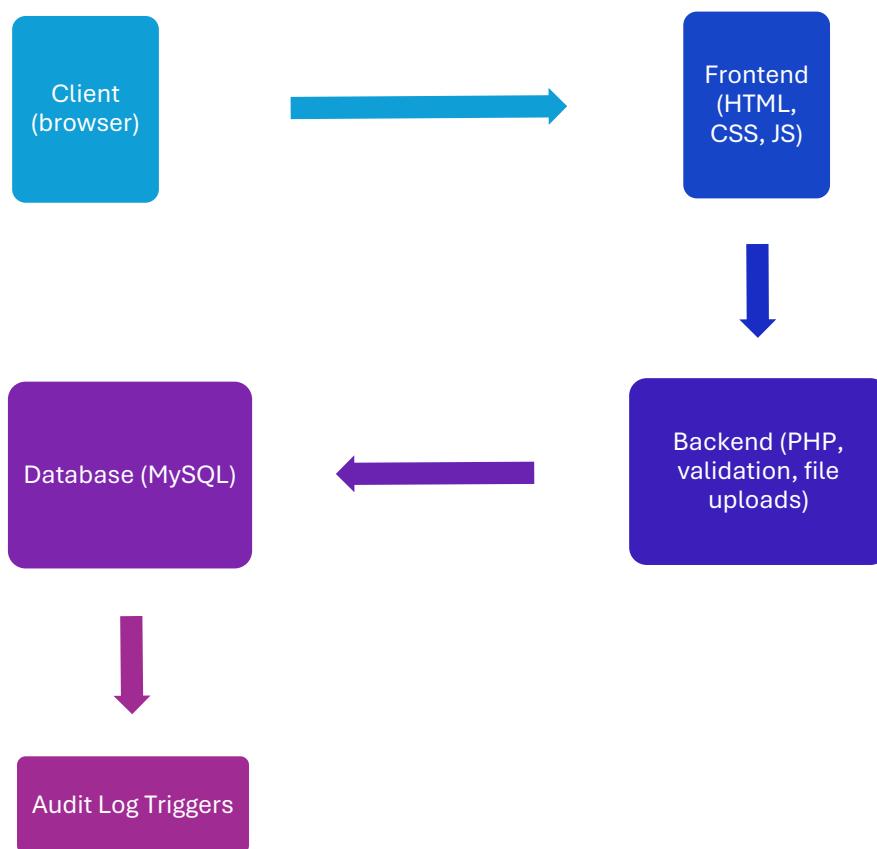
All changes are logged automatically using:

- Prepared statements
- MySQL triggers (for inserts, updates, deletes)

✓ Clean and modern UI

Each page has its own CSS file for customization.

System Architecture



Architecture follows a simple MVC-style flow where each page handles a single purpose.

Database Design

The database name used: campus_lost_found

Tables Used:

1. admin – stores admin accounts
2. items – stores found items
3. claims – stores user claim requests
4. audit_log – stores admin actions
5. MySQL triggers – log item operations automatically

Database Schema

Admin Table

Column	Type	Description
admin_id	INT (PK, AUTO_INCREMENT)	Unique ID
username	VARCHAR(100) (UNIQUE)	Login username
admin_name	VARCHAR(150)	Admin's full name
password	VARCHAR(255)	Hashed password

Items Table

Column	Type	Description
item_id	INT (PK, AUTO_INCREMENT)	Item identifier
item_name	VARCHAR(200)	Name of item
description	TEXT	Details
category	VARCHAR(100)	Bag / Electronics / Card etc.
location_lost	VARCHAR(200)	Where it was found
date_lost	DATE	Date found
image_path	VARCHAR(300)	Stored image filename
status	ENUM('lost','claimed')	Workflow status

added_by_admin	INT (FK admin_id)	Who added it
-----------------------	--------------------------	---------------------

Claims Table

Column	Type	Description
claim_id	INT (PK, AUTO_INCREMENT)	Claim identifier
item_id	INT (FK items.item_id)	Which item
claimant_name	VARCHAR(200)	User name
sap_id	VARCHAR(50)	SAP ID
course	VARCHAR(200)	Course
proof_path	VARCHAR(300)	Optional proof image
message	TEXT	Reason for claiming
status	ENUM('pending','approved','rejected')	Admin decision
created_at	DATETIME	Timestamp
updated_at	DATETIME	Timestamp

audit_log Table

Column	Type	Description
log_id	INT (PK, AUTO_INCREMENT)	Log entry
action_type	VARCHAR(100)	Type of operation
action_description	TEXT	Details
admin_id	INT	Which admin performed the action
timestamp	DATETIME	When

Schema (Description)

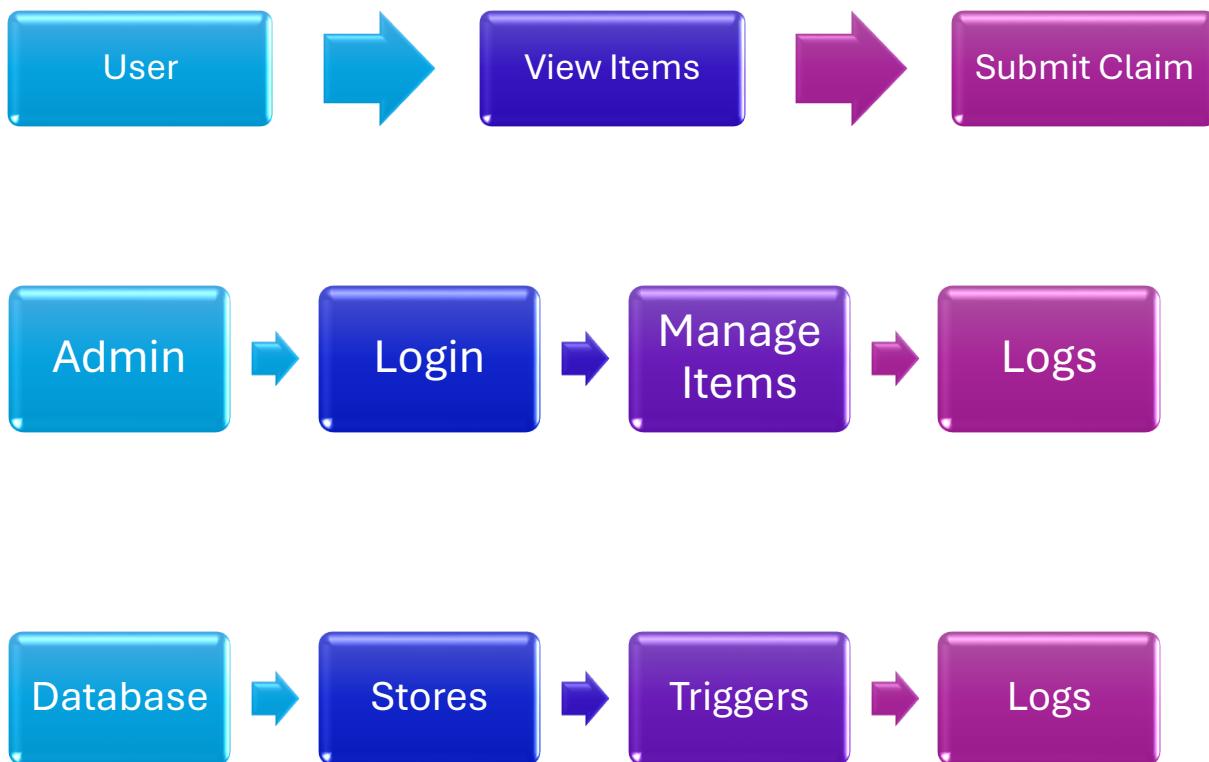
Entities:

- **Admin**
- **Item**
- **Claim**
- **Audit Log**

Relationships:

- One Admin can add many Items
- One Admin can manage many Claims
- One Item can have many Claims
- All operations performed by Admin insert into Audit Log

12. Data Flow Diagram (DFD)



13. Implementation

The system is implemented as a local web application using XAMPP, consisting of:

13.1 Frontend Implementation

The frontend is created using:

- HTML5 for structure
- CSS3 (separate CSS per page for clean styling)

- **JavaScript for banner animation and UI enhancements**
- **Responsive layout for desktop & mobile**

Important frontend features:

✓ Homepage

- **Fixed header banner**
- **Auto-scrolling image carousel showing latest lost items**
- **Grid layout of all items**
- **Buttons for “View Details” and “Claim Item”**

✓ Item Details Page

- **Large preview image**
- **Clear title, category, date lost, location lost**
- **Button for “Claim This Item”**
- **Organized layout using a card-based design**

✓ Claim Request Page

- **Clean form with fields for:**
 - **Name**
 - **SAP ID**
 - **Course**
 - **Proof image**
 - **Message / Explanation**
- **Validation on required fields**
- **Friendly success message page**

13.2 Backend Implementation

The backend is written entirely in PHP, using:

- **Prepared statements to prevent SQL injection**
- **File upload handling (item images + proof images)**
- **Sessions for admin login**
- **Modular folder structure**

- Error handling and form validation

13.3 File/Folder Structure

`campus_lost_found/`

```
|  
|   └── public/  
|       ├── index.php  
|       ├── item_details.php  
|       ├── claim_form.php  
|       ├── claim_success.php  
|       └── assets/  
|           ├── css/  
|           ├── js/  
|           └── images/  
|  
└── admin/  
    ├── login.php  
    ├── dashboard.php  
    ├── manage_items.php  
    ├── add_item.php  
    ├── edit_item.php  
    ├── manage_claims.php  
    ├── view_claim.php  
    └── logs.php  
        └── backend/  
            ├── add_item.php  
            ├── update_item.php  
            ├── delete_item.php  
            └── claim_actions.php
```

```
└── config/
    └── db_connect.php
    └──
    └── sql/
        └── database.sql
```

14. SQL Queries Used

This project demonstrates multiple SQL concepts.

14.1 Table Creation Queries (DDL)

Admin Table

```
CREATE TABLE admin (
    admin_id INT AUTO_INCREMENT PRIMARY KEY,
    admin_name VARCHAR(150),
    username VARCHAR(100) UNIQUE,
    password VARCHAR(255)
);
```

Items Table

```
CREATE TABLE items (
    item_id INT AUTO_INCREMENT PRIMARY KEY,
    item_name VARCHAR(200),
    description TEXT,
    category VARCHAR(100),
    location_lost VARCHAR(200),
    date_lost DATE,
    image_path VARCHAR(300),
    status ENUM('lost','claimed') DEFAULT 'lost',
    added_by_admin INT
);
```

Claims Table

```
CREATE TABLE claims (
    claim_id INT AUTO_INCREMENT PRIMARY KEY,
    item_id INT,
    claimant_name VARCHAR(200),
    sap_id VARCHAR(50),
    course VARCHAR(200),
    proof_path VARCHAR(300),
    message TEXT,
    status ENUM('pending','approved','rejected') DEFAULT 'pending',
    created_at DATETIME,
    updated_at DATETIME
);
```

Audit Log Table

```
CREATE TABLE audit_log (
    log_id INT AUTO_INCREMENT PRIMARY KEY,
    action_type VARCHAR(100),
    action_description TEXT,
    admin_id INT,
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

14.2 Select Queries

View all active lost items

```
SELECT * FROM items WHERE status='lost' ORDER BY date_lost DESC;
```

Fetch item details

```
SELECT * FROM items WHERE item_id = ?;
```

Fetch claims for a specific item

```
SELECT * FROM claims WHERE item_id = ?;
```

14.3 Insert Queries

Add new item

```
INSERT INTO items (item_name, description, category, location_lost, date_lost, image_path,  
added_by_admin)  
VALUES (?, ?, ?, ?, ?, ?, ?);
```

Add claim request

```
INSERT INTO claims (item_id, claimant_name, sap_id, course, proof_path, message,  
created_at)  
VALUES (?, ?, ?, ?, ?, ?, NOW());
```

14.4 Update Queries

Approve a claim

```
UPDATE claims SET status='approved', updated_at=NOW() WHERE claim_id=?;
```

Mark item as claimed

```
UPDATE items SET status='claimed' WHERE item_id=?;
```

Reject claim

```
UPDATE claims SET status='rejected', updated_at=NOW() WHERE claim_id=?;
```

14.5 Delete Query

Remove item

```
DELETE FROM items WHERE item_id = ?;
```

14.6 Complex Queries Demonstrated

Items with their latest claim status

```
SELECT i.item_name, c.status, c.claimant_name  
FROM items i  
LEFT JOIN claims c ON i.item_id = c.item_id  
ORDER BY c.updated_at DESC;
```

Count items by category

```
SELECT category, COUNT(*) AS total  
FROM items
```

```
GROUP BY category;
```

Count pending claims per item

```
SELECT item_id, COUNT(*) AS pending_claims  
FROM claims  
WHERE status='pending'  
GROUP BY item_id;
```

15. MySQL Triggers Used

Trigger for logging item additions:

```
CREATE TRIGGER trg_item_added  
AFTER INSERT ON items  
FOR EACH ROW
```

```
INSERT INTO audit_log(action_type, action_description, admin_id)
```

```
VALUES('item_added', CONCAT('Item added: ', NEW.item_name), NEW.added_by_admin);
```

Trigger for logging claim status updates:

```
CREATE TRIGGER trg_claim_update  
AFTER UPDATE ON claims  
FOR EACH ROW  
INSERT INTO audit_log(action_type, action_description, admin_id)  
VALUES('claim_status_changed',  
CONCAT('Claim ', NEW.claim_id, ' changed from ', OLD.status, ' to ', NEW.status),  
NEW.updated_by);
```

16. Testing & Validation

The system was tested on:

✓ Form validation

- Missing fields
- Incorrect input formats
- Missing image uploads

✓ Database operations

- Insert, Update, Delete
- Relations between the tables
- Claim workflow end-to-end

✓ Error handling

- Non-existing item
- Deleting items with pending claims
- Broken image paths → fallback placeholder

✓ Admin authentication

- Wrong password
- Non-existing user
- Multiple admin accounts

All features worked successfully after debugging and fixes.

17. Output

1. Homepage — fixed banner with scrolling images and item cards grid.
2. Add Item Page — admin entering details and uploading an image.
3. Manage Items Page — table listing items with edit/delete options.
4. Item Details Page — large preview image with claim button.
5. Claim Form Page — form with proof upload.
6. Claims Management Page — admin sees claim list with approve/reject.
7. Audit Logs Page — table showing each admin action.

18. Result

The Campus Lost & Found System was successfully implemented with:

- Full CRUD operations

- Admin authentication
- Item management
- Claim workflow
- Audit logs
- MySQL triggers
- Clean and responsive UI
- Proper DBMS usage and SQL integration

The system solves the practical problem of lost items by providing a streamlined digital alternative.

19. Conclusion

The developed system demonstrates how database concepts can be integrated into a real-world workflow. By using MySQL's features (constraints, relations, triggers), the project ensures data consistency, proper logging, and secure operations.

The implementation showcases complete usage of:

- SQL
- PHP
- Web technologies
- Database normalization
- User authentication
- Real workflow automation

The project successfully meets all DBMS mini-project requirements.

20. Future Scope

1. Student Login System
Allow students to track their claim history.
2. Email / SMS notifications
Notify users when claim gets approved.
3. QR code tagging
Each item can be tagged with a QR to verify ownership.

4. Mobile App Version

Using React Native or Flutter.

5. Admin role levels

Super Admin → Add Admins

Sub Admin → Manage items only

6. AI-based image similarity

To automatically match found items with lost reports.

21. References

- MySQL Documentation
- PHP Manual
- W3Schools (CSS, HTML)
- StackOverflow discussions
- XAMPP Apache + MySQL Guide

THANK YOU