

# **Case Study On**

# **Reclaim - Bringing Things Together**

# Introduction

- **Reclaim** is a Lost and Found Management System developed to digitally manage lost and found items.
- The system replaces the traditional manual process with a centralized software solution.
- It allows users to report lost and found items and track their status efficiently.
- Administrators can monitor, verify, and manage all item records through the system.
- The application improves efficiency, reduces errors, and simplifies item recovery.

# Abstract

- Reclaim is designed to manage lost and found activities in a systematic and organized manner.
- The system helps users register lost and found items through a digital platform.
- Administrators can verify item details and manage records efficiently.
- The application ensures better tracking and transparency in item recovery.
- The system is implemented using Spring Boot, MySQL, and web technologies.

# Client Requirements

- The system should allow users to register and log in securely.
- Users should be able to report lost and found items with relevant details.
- The admin should be able to view, update, and manage all item records.
- The system should support searching and filtering of items.
- Item status should be updated and tracked efficiently.

# Technical Terms

- The application uses **role-based login** for users and administrators.
- Backend logic is handled using **Spring Boot controllers and services**.
- **REST endpoints** are used for communication between frontend and backend.
- **Input validation and exception handling** are implemented to avoid errors.
- **Maven** is used for dependency management and project build.

# Technologies Used

- **Frontend:** HTML, CSS, JavaScript
- **Backend:** Java Spring Boot Framework
- **Build Tool:** Maven
- **Database:** MySQL
- **IDE & Tools:** Eclipse / STS, Browser (Chrome)

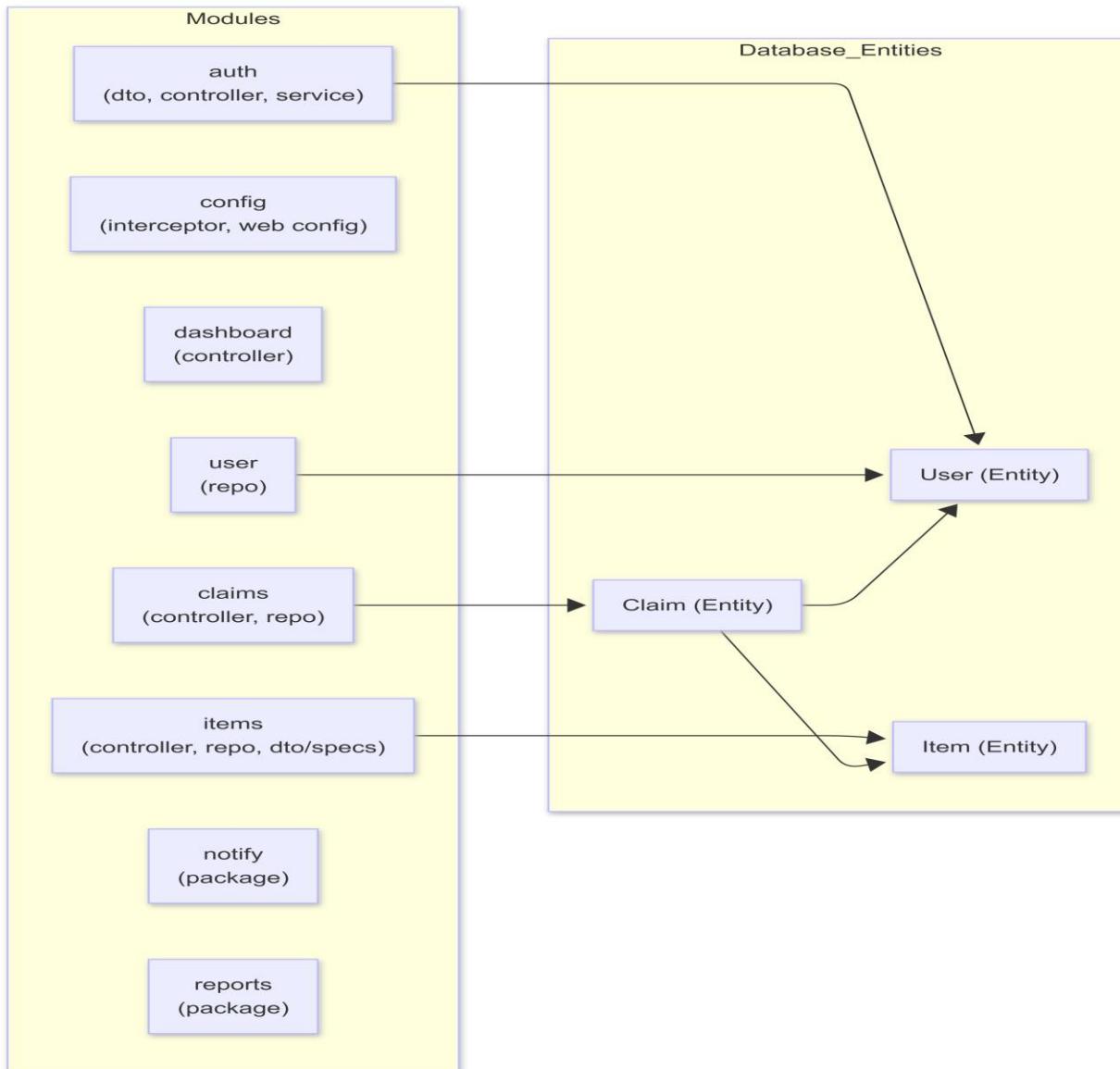
# System Requirements

- **Operating System:** Windows / Linux
- **Backend Environment:** Java 8+ with Spring Boot
- **Database Server:** MySQL
- **Web Server:** Apache Tomcat (embedded)
- **Browser:** Google Chrome

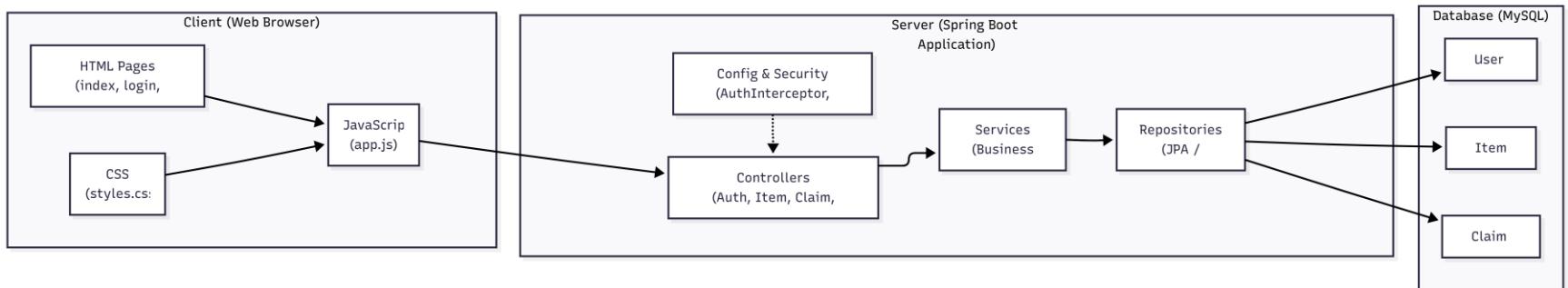
# Project Modules

- **Authentication Module** - handles user login and registration
- **Admin Module** - handles overall controls of the app.
- **User Module** - manages user details and user-related operations
- **Items Module** - manages lost and found item details and status
- **Claims Module** - handles claiming and verification of items
- **Dashboard Module** - displays summary and overview information

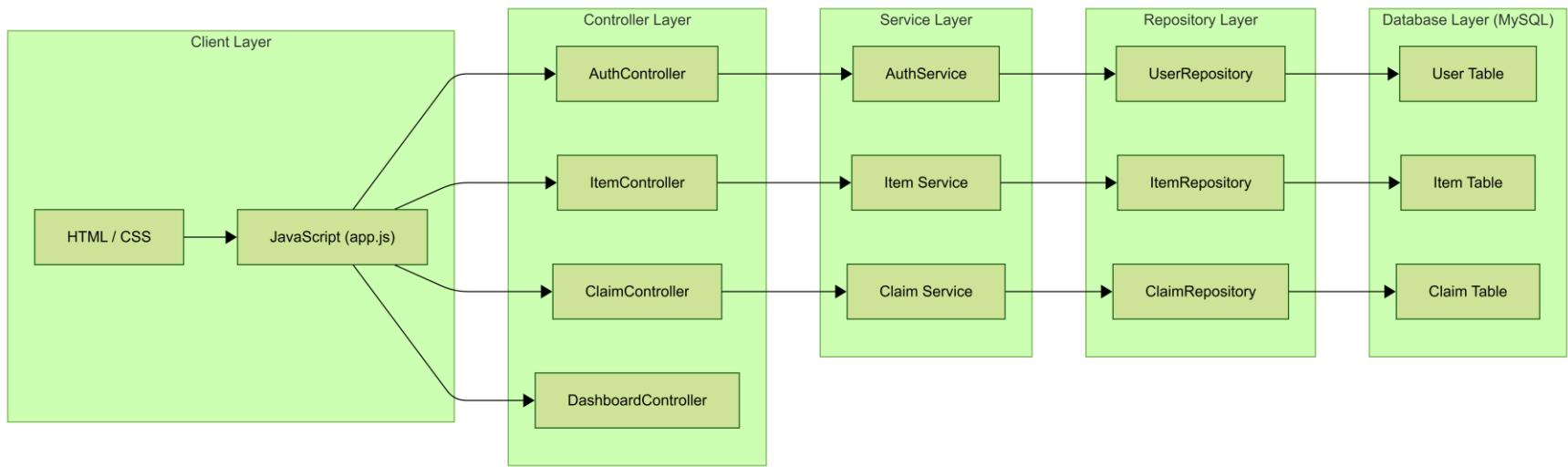
# ER Diagram



# Client-Server Architecture



# Application Layers



# Admin Module

- The **Admin Module** provides centralized control over the entire application.
- Admin can **view, resolve, approve, or reject** item claims.
- Admin has the authority to **delete inappropriate items or claims**.
- Admin can **block or manage users** to prevent misuse of the system.
- This module ensures **security, moderation, and smooth functioning** of the application.

# User Module

- Manages **user profile details** and basic user operations.
- Allows users to **view and manage their information** after login.
- Connects users with the items they create or claim.
- Works with the authentication module for secure access.
- Stores and retrieves user data using **JPA repositories**.

# Item Module

- Manages **lost and found item details** in the system.
- Allows users to **add, update, and delete item information**.
- Supports classification of items based on **category and type**.
- Tracks the **status of items** such as lost, found, or returned.
- Stores item data securely in the database using JPA.

# Dashboard Module

- Provides a **summary view** of lost, found, and claimed items.
- Displays **counts and statistics** for quick understanding.
- Helps administrators **monitor system activity** efficiently.
- Presents consolidated information in a **user-friendly format**.
- Improves decision-making by giving a clear overview of data.

# Claim Module

- Handles the process of **claiming lost or found items**.
- Allows users to submit a **claim request** for a specific item.
- Associates each claim with a **user and an item**.
- Enables tracking of claim **status** such as pending, approved, or rejected.
- Stores and manages claim information using JPA repositories.

# Item Status Flow

Represents the **lifecycle of an item** in the system:

- **Lost** - Item is reported as missing by a user
- **Found** - Item is found and added to the system
- **Verified** - Claim details are checked and approved
- **Returned** - Item is successfully returned to the owner

**Flow:**

Lost → Found → Verified → Returned

# Validation Handling

- Input validation is performed at the **controller level**
- Ensures required fields are **not null or empty**
- Prevents invalid or incomplete data from being processed
- Sends **proper error responses** for invalid requests
- Improves system reliability and data consistency

# Security Handling

- User authentication handled through **login and registration**
- Passwords stored in **encrypted/hashed format**
- Restricted access to protected pages like **dashboard**
- Session-based validation to prevent unauthorized access
- Ensures data safety for users and items

# Data Dictionary

- Defines the **structure of database tables** used in the system
- Describes important fields for **User, Item, and Claim** entities
- Helps understand how data is **stored and related**
- Improves clarity for developers and evaluators
- Acts as a reference for database design

# User Table

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- reclaim\_db
  - Tables
    - claims
    - items
    - users
      - Columns
      - Indexes
      - Foreign Keys
      - Triggers
  - Views
  - Stored Procedures
  - Functions

sys

Administration Schemas

Information

Schema: reclaim\_db

Object Info Session

SQL File 3\*

```
CREATE DATABASE IF NOT EXISTS reclaim_db;
SELECT * FROM CLAIMS;
SELECT * FROM ITEMS;
SELECT * FROM USERS;
```

Result Grid

| ID | Email            | Name     | Password Hash                                  |
|----|------------------|----------|--|
| 1  | abc@abc.com      | abc      | \$2a\$10\$1zpZXsvkFpfadgzRE6a/5Oiof5XWLmk...   |
| 2  | hi@hi.com        | hi       | \$2a\$10\$yTOKShNs5ndk1mzzkAWehNk2A0SNu...     |
| 3  | hi@n.com         | hi       | \$2a\$10\$Sw6TPwemxKTf1b/Sy8iWe87uK/qfn6T...   |
| 4  | shreya@gmail.com | shreya   | \$2a\$10\$nrWjGEXyAZGfaltLE/ZOzW.mtKQJzBL9...  |
| 5  | test@a.com       | test     | \$2a\$10\$pALKiF8e3HYUB5mRTvb94e03j.KoSgYG...  |
| 6  | test@gmail.com   | test     | \$2a\$10\$pJKC1QFL4n7b1rt4kpgVxONBvUjzMSizy... |
| 7  | dudu@bubu.com    | dudububu | \$2a\$10\$S4H2ZjotGNWI2xhRcPEYIuPuZHq7gN...    |

Result Grid Form Editor

Context Help Snippets

Output

| # | Time     | Action                             | Message           | Duration / Fetch      |
|---|----------|------------------------------------|-------------------|-----------------------|
| 1 | 21:38:16 | SELECT * FROM CLAIMS LIMIT 0, 1000 | 2 row(s) returned | 0.015 sec / 0.000 sec |
| 2 | 21:38:48 | SELECT * FROM ITEMS LIMIT 0, 1000  | 4 row(s) returned | 0.031 sec / 0.000 sec |
| 3 | 21:39:17 | SELECT * FROM USERS LIMIT 0, 1000  | 8 row(s) returned | 0.016 sec / 0.000 sec |

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

# Item Table

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

reclaim\_db

- Tables
  - claims
  - items
- Views
- Stored Procedures
- Functions

sys

SQL File 3\*

CREATE DATABASE IF NOT EXISTS reclaim\_db;

SELECT \* FROM CLAIMS;

SELECT \* FROM ITEMS;

Result Grid

| ID | Category | Contact Method | Contact Value  | Created At                 | Description                          | Event Date | Image |
|----|----------|----------------|----------------|----------------------------|--------------------------------------|------------|-------|
| 1  | Document | EMAIL          | nana@gmail.com | 2025-12-16 12:02:19.838506 | Found ID card named Sara             | 2025-12-16 | /upl/ |
| 2  | Phone    | PHONE          | 9876543210     | 2025-12-16 20:29:13.600911 | Brand new orange iphone 17 pro found | 2025-12-15 | /upl/ |
| 3  | Keys     | EMAIL          | test@gmail.com | 2025-12-16 20:43:22.520319 | Bunch of keys found in floor         | 2025-12-16 | NULL  |
| 4  | Toy      | PHONE          | 9876543210     | 2025-12-16 20:56:13.501392 | Found a bright pink flaming toy      | 2025-12-16 | /upl/ |
| *  | NULL     | NULL           | NULL           | NULL                       | NULL                                 | NULL       | NULL  |

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

ITEMS 2

Column: id

Definition:

id bigint AI PK

Output

| # | Time     | Action                             | Message           | Duration / Fetch      |
|---|----------|------------------------------------|-------------------|-----------------------|
| 1 | 21:38:16 | SELECT * FROM CLAIMS LIMIT 0, 1000 | 2 row(s) returned | 0.015 sec / 0.000 sec |
| 2 | 21:38:48 | SELECT * FROM ITEMS LIMIT 0, 1000  | 4 row(s) returned | 0.031 sec / 0.000 sec |

Object Info Session

# Claim Table

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: SQL File 3\*

SCHEMAS

reclaim\_db

- Tables
- claims
- Columns
- id
- created\_at
- status
- claimant\_id
- item\_id

Indexes

Foreign Keys

Triggers

Items

Users

Views

Stored Procedures

Functions

Administration Schemas

Information

Column: id

Definition: id bigint AI PK

Result Grid

|   | id   | created_at                 | status  | claimant_id | item_id |
|---|------|----------------------------|---------|-------------|---------|
| ▶ | 1    | 2025-12-16 20:30:50.874073 | PENDING | 5           | 1       |
| * | 2    | 2025-12-16 20:31:31.505266 | PENDING | 5           | 1       |
| * | NULL | NULL                       | NULL    | NULL        | NULL    |

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

CLAIMS 1

Action Output

| # | Time     | Action                             | Message           | Duration / Fetch      |
|---|----------|------------------------------------|-------------------|-----------------------|
| 1 | 21:38:16 | SELECT * FROM CLAIMS LIMIT 0, 1000 | 2 row(s) returned | 0.015 sec / 0.000 sec |

Object Info Session

# User Interface - Home

The screenshot shows the homepage of the Reclaim platform. At the top left is the brand name "Reclaim.". On the top right are two buttons: "Login" and a red rounded rectangle button labeled "Get Started". The central feature is a large, bold text message: "Lost something? Let's **Reclaim** it together." Below this is a descriptive subtitle: "The community-driven platform to report lost items and find what matters most to you. Simple, fast, and secure." At the bottom center are two call-to-action buttons: a red one on the left labeled "I Lost Something" and a grey one on the right labeled "I Found Something".

Reclaim.

Login

Get Started

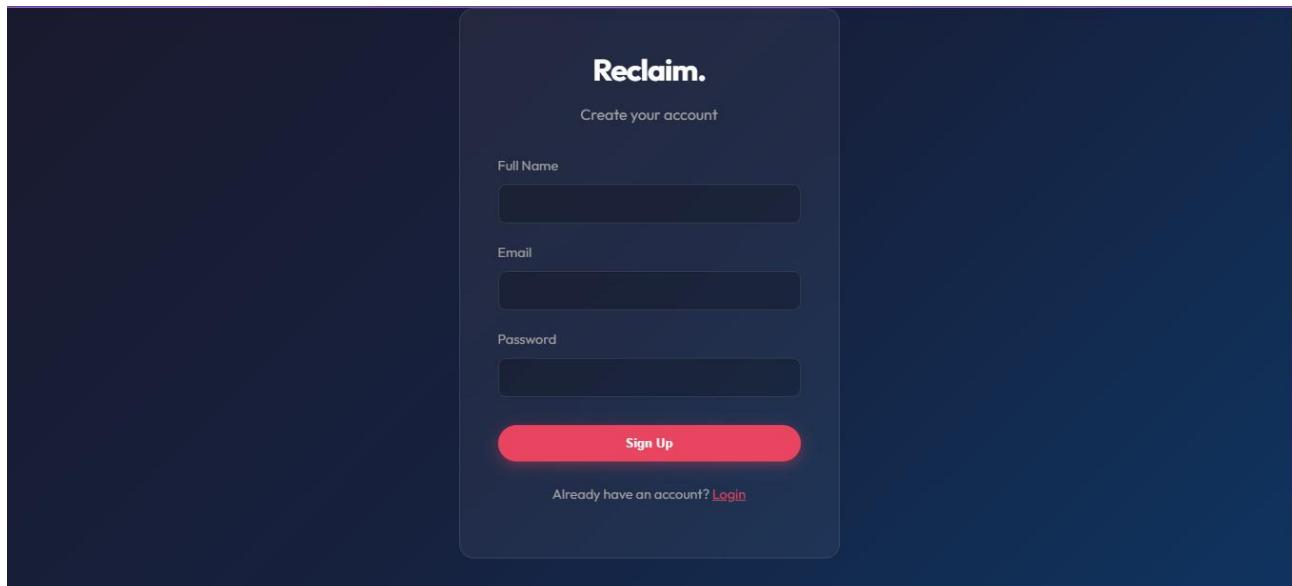
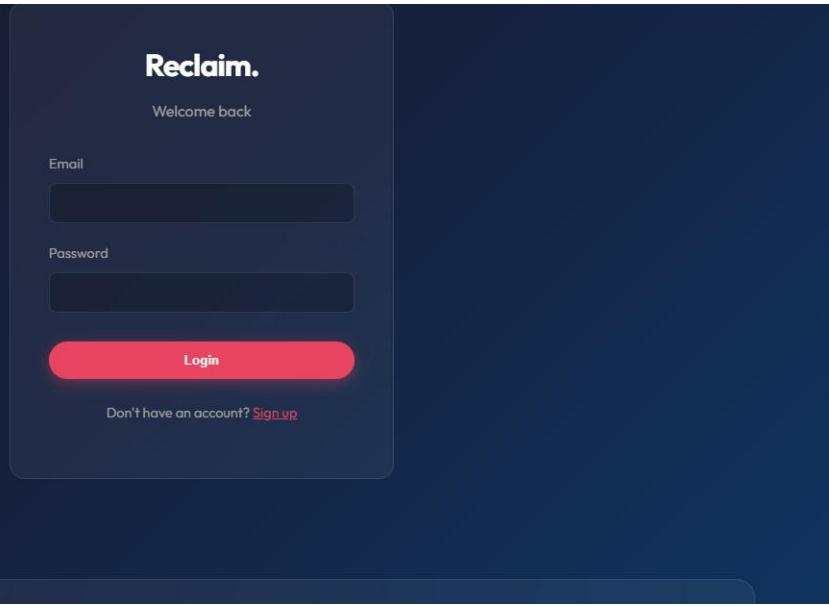
**Lost something?  
Let's **Reclaim** it together.**

The community-driven platform to report lost items and find what  
matters most to you. Simple, fast, and secure.

I Lost Something

I Found Something

# User Interface – Login/Register



# User Interface - Dashboard

Reclaim.

Home   Dashboard   Browse   + Post New   Logout

## My Dashboard

Welcome back, test

Active Lost

1

Active Found

1

My Posts

2

### My Recent Posts

Flamingo Toy  
Toy · Near billing

Bunch of keys found  
Keys · Gate

# User Interface - Admin Controls

Reclaim.

Admin Dash Browse App Logout

## System Overview

Manage community reports and user access

| TOTAL ITEMS | ACTIVE | TOTAL USERS |
|-------------|--------|-------------|
| 5           | 3      | 10          |

**Reported Items**    [User Management](#)

| ITEM                                      | TYPE  | POSTED BY | STATUS   | ACTIONS  |
|---|-------|-----------|----------|--|
| Black Louis Vuitton Wallet<br>Near office | LOST  | test      | ACTIVE   | <a href="#">Resolve</a> <a href="#">Delete</a> |
| Flamingo Toy<br>Near billing              | FOUND | test      | RESOLVED | <a href="#">Resolve</a> <a href="#">Delete</a> |
| Bunch of keys found<br>Gate               | FOUND | test      | ACTIVE   | <a href="#">Resolve</a> <a href="#">Delete</a> |
| iPhone 17 Pro<br>Near gate                | LOST  | test      | RESOLVED | <a href="#">Resolve</a> <a href="#">Delete</a> |
| ID card<br>Near office                    | LOST  | hi        | ACTIVE   | <a href="#">Resolve</a> <a href="#">Delete</a> |

# User Interface - Browse Items

Reclaim.

Home Dashboard Browse + Post New Logout

Keyword Type Category Location

Keys All Electronics, Pets... City or Area Search

**Bunch of keys found**  
Keys - Gate  
Bunch of keys found in floor...  
View Details

FOUND

The image shows a screenshot of a web application interface. The header 'Reclaim.' is on the left, followed by navigation links: Home, Dashboard, Browse, a red 'Post New' button, and Logout. Below the header is a search bar with four input fields: 'Keyword' (containing 'Keys'), 'Type' (set to 'All'), 'Category' (listing 'Electronics, Pets...'), and 'Location' (empty). To the right of the search bar is a red 'Search' button. The main content area features a large, abstract graphic of a key in shades of blue and green. Below the graphic is a card with the title 'Bunch of keys found', subtext 'Keys - Gate', and a truncated message 'Bunch of keys found in floor...'. A 'View Details' button is at the bottom of the card. In the top right corner of the card, the word 'FOUND' is written in a small green box.

# User Interface - Post Lost/Found

## Post an Item

I...

Category

Lost something

Lost something

Found something

Black Louis Vuitton Wallet

Description

Wallet with cash and cards

Location

Date

Near office

17-12-2025

Contact Method

Contact Details

Phone

9876543210

Upload Image (Optional)

Choose File No file chosen

Submit Post

# User Interface – User Controls View

The screenshot displays a user interface for managing lost items. At the top left is the brand logo "Reclaim.". To the right are navigation links: "Home", "Dashboard", and "Browse". The main content area shows a detailed view of a lost item.

**Item Details:**

- Name:** Black Louis Vuitton
- Type:** Wallet
- Status:** LOST (ACTIVE)
- Location:** Near office
- Date:** 2025-12-17

**Description:** Wallet with cash and cards

**Contact Info:**

- PHONE: 9876543210

**Actions:**

- Update Image
- Mark as Resolved
- Delete

**Claims:**

No claims yet.

# User Interface – Non-login View

The screenshot displays a dark-themed user interface for a lost item tracking platform. At the top left is the brand logo "Reclaim.". To the right are three navigation links: "Home", "Dashboard", and "Browse". The main content area features a large, rounded rectangular card for a lost item. On the left side of this card is a placeholder image area with the text "No Image". To the right of the image area, the item details are listed:

- Black Louis Vuitton** (in large bold letters)
- Wallet** (in bold letters)
- Wallet · Near office · 2025-12-17
- Description**: Wallet with cash and cards
- Contact Info**: PHONE: 9876543210

At the bottom of the card is a red button labeled "Claim This Item". Above the card, there are two small circular status indicators: one red circle with the word "LOST" and one green circle with the word "ACTIVE".

# User Interface - Footer

My Recent Posts



Flamingo Toy  
Toy • Near billing  
RESOLVED



Bunch of keys found  
Keys • Gate  
ACTIVE

**Reclaim.**

The trusted community-driven platform to report lost items and find what matters most to you. Simple, fast, and secure.

**Quick Links**

- Home
- Browse Items
- Dashboard
- Post Item

**Contact & Legal**

- [support@reclaim.app](mailto:support@reclaim.app)
- +1(555) 123-4567
- [Privacy](#) [Terms](#)

© 2025 Reclaim App. All rights reserved.

# Testing

- Manual testing performed for all UI pages
- API endpoints tested using browser
- Database operations verified using MySQL queries
- End-to-end flow tested for item posting and claiming
- Basic validation testing handled at controller level

# Security

- Authentication implemented using **login and registration**
- User credentials are stored securely using **hashed passwords**
- Access to protected pages is restricted after login
- Basic authorization ensures only valid users perform actions
- Security logic handled at **controller and interceptor level**

# Error Handling

- Centralized exception handling using Spring Boot
- Validation errors handled at controller level
- Meaningful error messages returned to the client
- Prevents application crashes due to invalid inputs

# Limitations

- Manual verification required for claim approval
- No automated image or AI-based matching
- Web-based application only (no mobile app)
- Depends on user-provided item details
- Limited notification mechanisms

# Future Enhancements

- Mobile application support (Android / iOS)
- Image-based item matching using AI
- Automated claim verification system
- Real-time notifications (email / SMS)
- Admin analytics dashboard

# Conclusion

- Reclaim provides an efficient solution for lost and found management
- Simplifies item reporting and claim verification
- Ensures secure data handling using authentication
- Demonstrates full-stack development using Java Spring Boot
- Can be extended further with advanced features

# Links

GitHub Link:

<https://github.com/Shreyasri30/Placement>

Demo Video Link:

<https://drive.google.com/file/d/182vKVCugOwUtNgbv2bWj3YMa01wwwD5G/view?usp=sharing>

Thank You