

**ITEM RETRIEVAL MANAGEMENT SYSTEM
A MINI-PROJECT REPORT**

Submitted by

AMIRTHA S R 240701036

ANU SHALINI B 240701043

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project "**ITEM RETRIEVAL MANAGEMENT SYSTEM**" is the Bonafide work of "**AMIRTHA S R, ANU SHALINI B**" who carried out the project work under my supervision.

SIGNATURE

Dr. THEEPĀ

ASSISTANT PROFESSOR SG

Dept. of Computer Science and Eng,
Rajalakshmi Engineering College
Chennai

This mini project report is submitted for the viva voce examination to be held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

In our state transportation plays a major role. Even though there are various multinational companies like ola and uber that operate on a larger scale, the local market is bereft of any such hassle-free application system. To incorporate this drawback on the local transportation system, our team develop a database system to help the local market to maintain the data and organize it efficiently. The main objective of this project is to allocate the available taxis according to the customer requirement. This system helps to maintain the availability and details of taxis in a local transportation agency. This will allow the agency to beat the competition from rival agencies by providing efficient service.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr. E.M. MALATHY** and our Deputy Head Of The Department **Dr. J. MANORANJINI** for being ever-supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide **Dr. THEEPA** for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

- 1. AMIRTHA S R**
- 2. ANU SHALINI B**

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	iv
1	INTRODUCTION	1
1.1	INTRODUCTION	8
1.2	SCOPE OF THE WORK	8
1.3	PROBLEM STATEMENT	8
1.4	AIM AND OBJECTIVES OF THE PROJECT	8
2	SYSTEM SPECIFICATIONS	9
2.1	SOFTWARE SPECIFICATIONS	9
3	MODULE DESCRIPTION	10
4	CODING	11
5	SCREENSHOTS	16
6	CONCLUSION AND FUTURE ENHANCEMENT	18
	REFERENCES	19

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
5.1	INDEX PAGE	15
5.2	REGISTRATION PAGE	15
5.3	LOGIN	16
5.4	CHOICE SELECTION	16
5.5	LOST ITEM CREATION	17
5.6	FOUND ITEM CREATION	17

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The project helps people to report their lost item, and allows the people who found items which are not of their own to report, these data are checked for similarity and are restored back to the owner

1.2 SCOPE OF THE WORK

The lost and found management system helps users retrieve their items back among their busy schedule by notifying the user when the item is matched.

1.3 PROBLEM STATEMENT

The need for the project is that many people tend to lose their belongings in different places and the retrieval of the products seem impossible with the fast-moving circumstances. Hence this system serves as an approach to help them find their valuables fast and easier with just a few details.

1.4 AIM AND OBJECTIVES OF THE PROJECT

The aim of the Lost and Found Management System is to efficiently facilitate the recovery and secure return of lost personal property to their owners. The objective is to maintain a transparent, organized, and auditable record of all found and claimed items.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 SOFTWARE SPECIFICATIONS

Operating System	:	WINDOWS 10
Front - End	:	JSP
Back - End	:	Oracle
Language	:	Java ,SQL

CHAPTER 3

MODULE DESCRIPTION

This Item retrieval application consists of a single User Access Module. When the program runs, it will present a Login Window for confirmation. The person interacting with the system can only log in as a Standard User to perform daily operational tasks.

1. User login

When the person tries to log in as a Standard User, they will be prompted to enter their unique credentials (username, email and password) for secure authentication. Upon successful login, they are directed to a central Action Choice Page. This page provides two operational options: "I lost something" (to report lost item) and "I found something" (to record a found item).

CHAPTER 4

SAMPLE CODING

Found item servlet

```
import java.io.File;  
  
import java.io.IOException;  
  
import java.nio.file.Paths;  
  
import java.sql.*;  
  
import jakarta.servlet.ServletException;  
  
import jakarta.servlet.annotation.MultipartConfig;  
  
import jakarta.servlet.annotation.WebServlet;  
  
import jakarta.servlet.http.*;  
  
@WebServlet("/FoundItemServlet")  
  
@MultipartConfig(fileSizeThreshold = 1024 * 1024 * 2,  
    maxFileSize = 1024 * 1024 * 10,  
    maxRequestSize = 1024 * 1024 * 50)  
  
public class FoundItemServlet extends HttpServlet {  
  
    private static final long serialVersionUID = 1L;
```

```
private static final String DB_URL =
"jdbc:oracle:thin:@localhost:1521:xe";

private static final String DB_USER = "240701036";

private static final String DB_PASSWORD = "rec";

private static final String UPLOAD_DIR = "uploads";

protected void doPost(HttpServletRequest request,
HttpServletResponse response)

throws ServletException, IOException {

HttpSession session = request.getSession(false);

if (session == null || session.getAttribute("userId") == null) {

response.sendRedirect("login.jsp");

return;

}

int userId = (int) session.getAttribute("userId");

String itemName = request.getParameter("name");

String description = request.getParameter("description");
```

```
String foundDateStr = request.getParameter("FoundDate");

String city = request.getParameter("city");

String building = request.getParameter("building");

String floor = request.getParameter("floor");

String foundLocation = city + ", " + building + ", Floor " + floor;

Date foundDate = Date.valueOf(foundDateStr);

Part filePart = request.getPart("image");

String fileName = null;

if (filePart != null && filePart.getSize() > 0) {

    fileName =
Paths.get(filePart.getSubmittedFileName()).getFileName().toString();

    String appPath = request.getServletContext().getRealPath("");

    String uploadPath = appPath + File.separator + UPLOAD_DIR;

    File uploadDir = new File(uploadPath);

    if (!uploadDir.exists()) uploadDir.mkdir();
```

```
filePart.write(uploadPath + File.separator + fileName);

}

Connection conn = null;

PreparedStatement ps = null;

try {

    Class.forName("oracle.jdbc.driver.OracleDriver");

    conn = DriverManager.getConnection(DB_URL, DB_USER,
DB_PASSWORD);

    String sql = "INSERT INTO found_items (FOUND_ITEMS_ID, USER_ID,
ITEMS_NAME, DESCRIPTION, FOUND_DATE, FOUND_LOCATION,
IMAGE_PATH, STATUS, CREATED_AT) " +
"VALUES (FOUND_ITEMS_SEQ.nextval, ?, ?, ?, ?, ?, ?, ?, ?)";

    ps = conn.prepareStatement(sql);

    ps.setInt(1, userId);
```

```
ps.setString(2, itemName);

ps.setString(3, description);

ps.setDate(4, foundDate);

ps.setString(5, foundLocation);

ps.setString(6, fileName);

ps.setString(7, "Found");

ps.setTimestamp(8, new Timestamp(System.currentTimeMillis()));

ps.executeUpdate();

response.sendRedirect("found.jsp?success=true");

} catch (Exception e) {

    e.printStackTrace();

    response.sendRedirect("found.jsp?success=false");

} finally {

    try { if (ps != null) ps.close(); } catch (Exception e) {}

    try { if (conn != null) conn.close(); } catch (Exception e) {}

}
```

```
    }  
  
}  
  
}
```

Sample 1

This Found Item Servlet handles the submission of a newly found item by a logged-in user. It first validates the user's session. It then processes item details (name, location, date, description) and handles the upload of an image file to a designated server directory. Finally, it persists all the found item data, including the image file path, into the found_items Oracle database table.

```
package com.lostfound;  
  
import java.io.IOException;  
  
import java.sql.Connection;  
  
import java.sql.DriverManager;  
  
import java.sql.PreparedStatement;  
  
import java.sql.ResultSet;  
  
import java.sql.SQLException;  
  
import jakarta.servlet.ServletException;
```

```
import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import jakarta.servlet.http.HttpSession;

@WebServlet("/LoginServlet")

public class LoginServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    private String jdbcURL = "jdbc:oracle:thin:@localhost:1521:xe";

    private String jdbcUsername = "240701036";

    private String jdbcPassword = "rec";

    protected void doPost(HttpServletRequest request,
    HttpServletResponse response)

        throws ServletException, IOException {

        String email = request.getParameter("email");

        String password = request.getParameter("password");

        if (email == null || password == null || email.isEmpty() ||
password.isEmpty()) {

            response.sendRedirect("login.jsp?error=empty");
        }
    }
}
```

```
    return;

}

Connection conn = null;

PreparedStatement ps = null;

ResultSet rs = null;

int userId = -1; // Initialize userId

try {

    Class.forName("oracle.jdbc.driver.OracleDriver");

    conn = DriverManager.getConnection(jdbcURL, jdbcUsername,
        jdbcPassword);

    String authSql = "SELECT USER_ID, NAME FROM USERS WHERE
EMAIL = ? AND PASSWORD = ?";

    ps = conn.prepareStatement(authSql);

    ps.setString(1, email);

    ps.setString(2, password);

    rs = ps.executeQuery();

    if (rs.next()) {

        userId = rs.getInt("USER_ID");
```

```
String name = rs.getString("NAME");

HttpSession session = request.getSession(true);

session.setAttribute("userId", userId);

session.setAttribute("userName", name);

session.setAttribute("email", email);

System.out.println(" ✅ Login successful for: " + name + ", ID: " +
userId);

rs.close();

ps.close();

boolean matchFound = false;

try {

    String matchSql = "SELECT COUNT(*) FROM MATCHES WHERE
LOST_ITEM_USER_ID = ?";

    ps = conn.prepareStatement(matchSql);

    ps.setInt(1, userId); // Bind the User ID (NUMBER)

    rs = ps.executeQuery();

    if (rs.next() && rs.getInt(1) > 0) {

        matchFound = true;
    }
}
```

```
    System.out.println("⚠ DEBUG: Match found! Count: " +
rs.getInt(1));

} else {

    System.out.println("DEBUG: Match not found. Count: 0");

}

} catch (SQLException matchE) {

    System.err.println("✗ Match Check DB Error: Failed to query
MATCHES table. Defaulting to no match.");

    matchE.printStackTrace();

    matchFound = false;

}

session.setAttribute("matchFound",
String.valueOf(matchFound));

response.sendRedirect("choice.jsp");

return;

} else {

    System.out.println("✗ Invalid login attempt for email: " + email);

    response.sendRedirect("login.jsp?error=invalid");

return;
```

```
    }

} catch (SQLException e) {

    System.err.println("✖ CRITICAL DB/AUTH ERROR (Redirecting to
login):");

    e.printStackTrace();

    response.sendRedirect("login.jsp?error=db_critical_failure");

} catch (ClassNotFoundException e) {

    System.err.println("✖ Oracle JDBC Driver not found:");

    e.printStackTrace();

    response.sendRedirect("login.jsp?error=driver_exception");

} catch (Exception e) {

    System.err.println("✖ General Server Exception:");

    e.printStackTrace();

    response.sendRedirect("login.jsp?error=server_exception");

} finally {

    try {

        if (rs != null) rs.close();

        if (ps != null) ps.close();

    }
```

```
if (conn != null) conn.close();  
  
} catch (SQLException e) {  
  
    e.printStackTrace();  
  
}}}
```

Sample 2

This Login Servlet handles user authentication against an Oracle database. Upon successful login, it creates a session with the user's ID, name, and email. It then checks the MATCHES table to see if any new matches were found for the user's lost items, setting a match Found flag before redirecting to choice.jsp

CHAPTER 5

SCREEN SHOTS

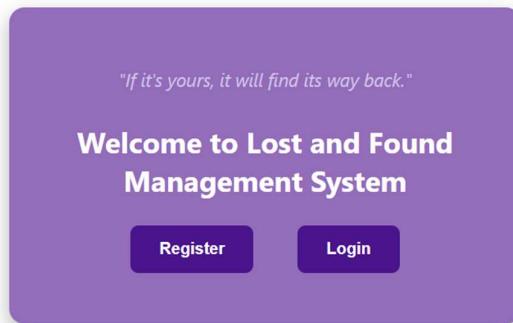


Fig 5.1 Index page

The image shows the registration page of the application. The background is purple. At the top, the word "Register" is centered. Below it are five input fields: a text field containing "murali", an email field containing "murali@gmail.com", a password field ending with "...", a dropdown menu set to "Student", and a date/time field showing "29-10-2025 16:26". To the right of the password field is a green circular icon with a white key symbol. At the bottom is a large blue "REGISTER" button. Below the button, a small note says "Already registered? Login here".

Fig 5.2 Registration Page

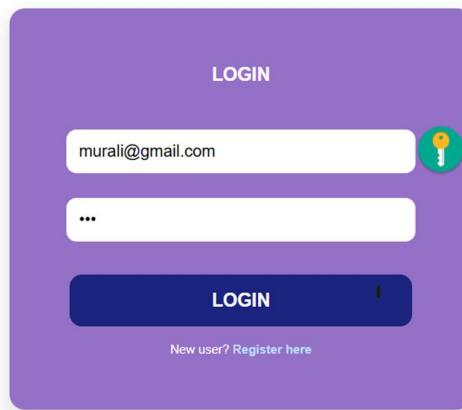


Fig 5.3 Login Page

A screenshot of a choice selection page for a lost and found management system. The top navigation bar includes 'Reviews' (highlighted in pink), 'LOST AND FOUND MANAGEMENT SYSTEM', 'About Us', 'Hi, swetha', and 'Categories'. On the left, there's a sidebar with three reviews: 'I found my phone here!', 'This helped me get my ID back.', and 'Amazing system!'. The main content area shows a white card with the question 'What would you like to do?' and two purple buttons: 'I Lost Something' and 'I Found Something'. A small note at the bottom says 'All that is lost shows there is so much to be found.'

Fig 5.4 choice selection

Report a Lost Item

Item Name
pen

Item Description (max 500 characters)
parker black gold cap

Category of Item
Valuables

Date Lost
28-10-2025

City/District in Tamil Nadu
Chennai

Building Name / Landmark
vm mall

Floor / Room No
1st floor

Date of Creation
2025-10-28

Upload Image (optional)
Choose File | No file chosen

Submit Lost Item

Fig 5.5 Deletion of Booking

Lost & Found - Report Found Item

Report a Found Item

Your Name

Phone Number
e.g., +91 9876543210

Item Description (max 500 characters)

Category of Item
Selected Category

Date Found
dd-mm-yyyy

City/District in Tamil Nadu
Selected District

Building Name / Landmark

Floor / Room No

Date of Creation
2025-10-29

Upload Image (optional)
Choose File | No file chosen

Submit Found Item

Fig 5.6 Database creation

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

The deployment of a digital Lost and Found Management System (LFMS) marks a significant step forward, successfully transitioning the process from an inefficient, manual effort to a streamlined, centralized, and transparent service. The system's success is measured by its ability to increase the recovery rate of items and dramatically improve user satisfaction by providing real-time status tracking and automated notifications. It frees up staff time, allowing administrators to manage the inventory efficiently through a single database.

For future scalability and effectiveness, the system should focus on integrating advanced technologies. The most impactful enhancement is the deployment of AI-powered image matching, which allows the system to automatically compare photos of found items against lost reports using visual recognition. This eliminates manual searches and ensures much quicker, more accurate matches. Other key additions include Location-Based Services (LBS) for pinpointing item locations on a map, and implementing Smart Locker integration to secure found items and automate the retrieval process via secure codes, further enhancing security and convenience.

REFERENCES

1. <https://www.w3schools.com/sql/>
2. <https://www.tutorialspoint.com/oracle/index.htm>
3. <https://www.wikipedia.org/>
4. <https://www.codecademy.com/learn/learn-java>