IIITA Lost and Found Help Desk Project

This project aims to develop a comprehensive Lost and Found Help Desk application to reduce frequent email inquiries. The system enables users to post and claim lost or found items through a centralized dashboard, streamlining the process for campus or community environments. It supports two main user roles: those who find items and those who lose them, facilitating quick and organized item recovery.

With database tables to accurately capture item and user information, and key features like email-verified registration and admin approvals, this app improves accountability and efficiency in managing lost belongings.

BY- Group 25

Vinay Kumar, Nikhil Chauhan, Aniket Kumar, Aastha, Abhinav

Tech Stack : React, Node.js, Express, MySQL, Tailwind CSS

React.js

For the front-end, React.js offers a component-based architecture, enhancing UI development and maintainability. React ensures a responsive and interactive user experience.

Node.js

The back-end utilizes Node.js, a JavaScript runtime environment, providing scalability and non-blocking I/O operations. It's efficient for handling numerous requests.

My**SQL**

MySQL, a relational database management system, is used to store and manage all the system's data, including users, complaints, and personnel details.

Application Front-End Architecture

1

Login/Signup Page

Entry point with secure authentication and email verification.

2

Dashboard

Main interface showing lost and found posts with search and filter options.
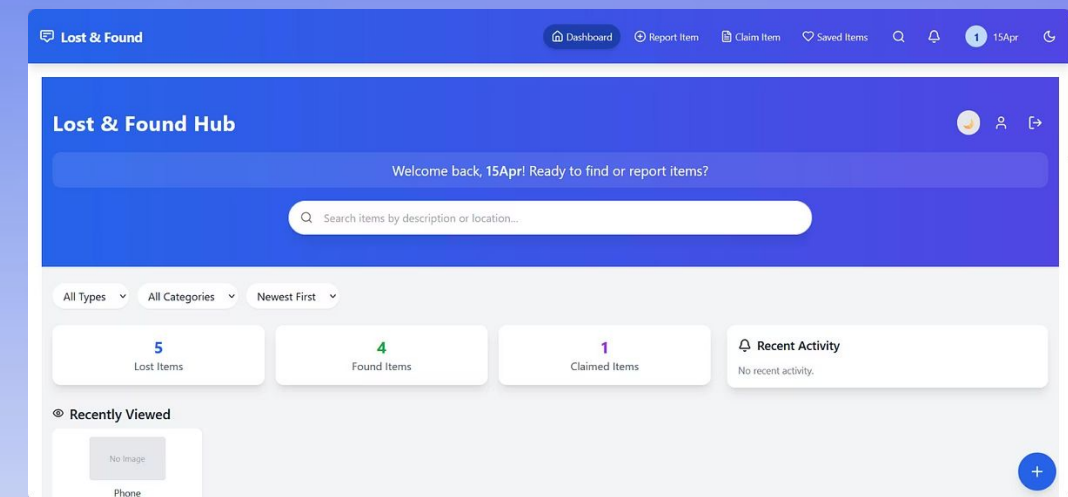
3

Report Item Page

Forms for users to report lost or found items with validation for data accuracy.

4

Claim Items Page

Users submit claims on found items, triggering status updates.

User Authentication: Login and Signup Modules

1 **Registration Module**

Allows students to create accounts with email verification for security. This ensures only genuine users access the platform.

2 **Login Module**

Provides secure, role-based access to users and admin with validation to protect user data and system integrity.

3 **Security Features**

Includes password encryption, email verification, and session management to safeguard user accounts.

Welcome Back!

Log in to your account below

Email

Enter your email

Password

Enter your password

Login

Forgot Password?

Or login with

Google   Facebook

Don't have an account? Sign up

Lost & Found Hub

🌙 👤 ⏻

Welcome back, **15Apr**! Ready to find or report items?

🔍 Search items by description or location...

All Types ⌄	All Categories ⌄	Newest First ⌄

5 Lost Items	**4** Found Items	**1** Claimed Items	🔔 **Recent Activity** No recent activity.

👁 **Recently Viewed**

No Image

Phone

➕

Dashboard Overview

Lost and Found Posts

The dashboard displays real-time posts of lost and found items. Users can browse and search unclaimed items efficiently.

Search Functionality

A robust search bar enables filtering by item description, category, or location, helping users quickly locate relevant posts.

Reporting Lost and Found Items

Found Item Reporting

Users enter details about items they have found, including description and collection location, generating a dashboard post.

Lost Item Reporting

Users report lost belongings with identifying info and personal contact details, creating an alert for potential matches.

Data Accuracy

Forms validate inputs to ensure consistency and completeness of the reported item data for reliable tracking.

Report an Item

Fill in the details below to report a lost or found item.

Description *

Describe the item (e.g., black wallet, red umbrella)

Location *

Where was it lost or found?

Item Type

Lost

Category *

other

Upload Image (Optional)

Click to upload an image (JPEG/PNG, max 5MB)

Claiming Lost Items and Status Tracking

✓ Item Claim Process

Users claim lost items by submitting personal details linked to the posted found item. The claim triggers status updates.

⟳ Status Updates

Once claimed, item status automatically changes to "Found", reflected on the dashboard to prevent duplicate claims.

⛁ Record Maintenance

All claim interactions are logged in the database for audit trails, supporting transparency and accountability.

Claim an Item

Search, review, and submit proof to claim a found item.

Search Items (by description, location, or ID)

🔍 Search found items...

Item ID *

Enter or select the Item ID

Example: valid ID is listed in the item search above.

Proof of Ownership *

Describe details that prove the item is yours (unique marks, receipts, etc.)

[Claim Item]   [Back]

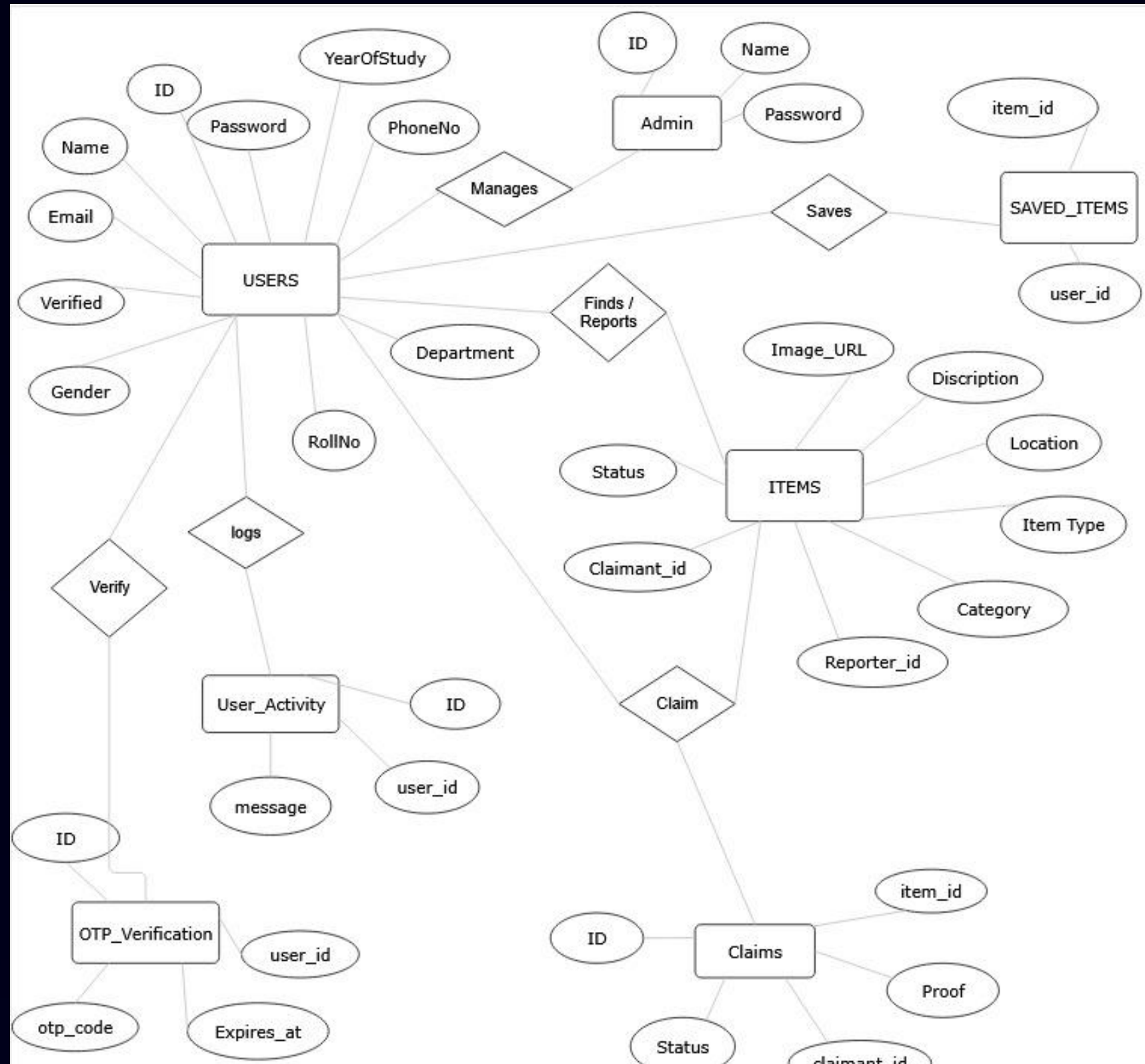# Admin Dashboard and Approvals

Admin Access

Restricted to authorized personnel who oversee claims and item statuses to ensure legitimate recoveries.

Claims Approval

Admins review and approve claims, updating database records to reflect accurate ownership transfers.

Analytics and Monitoring

Dashboards provide status views on lost items, claims, and user activity for operational insights.
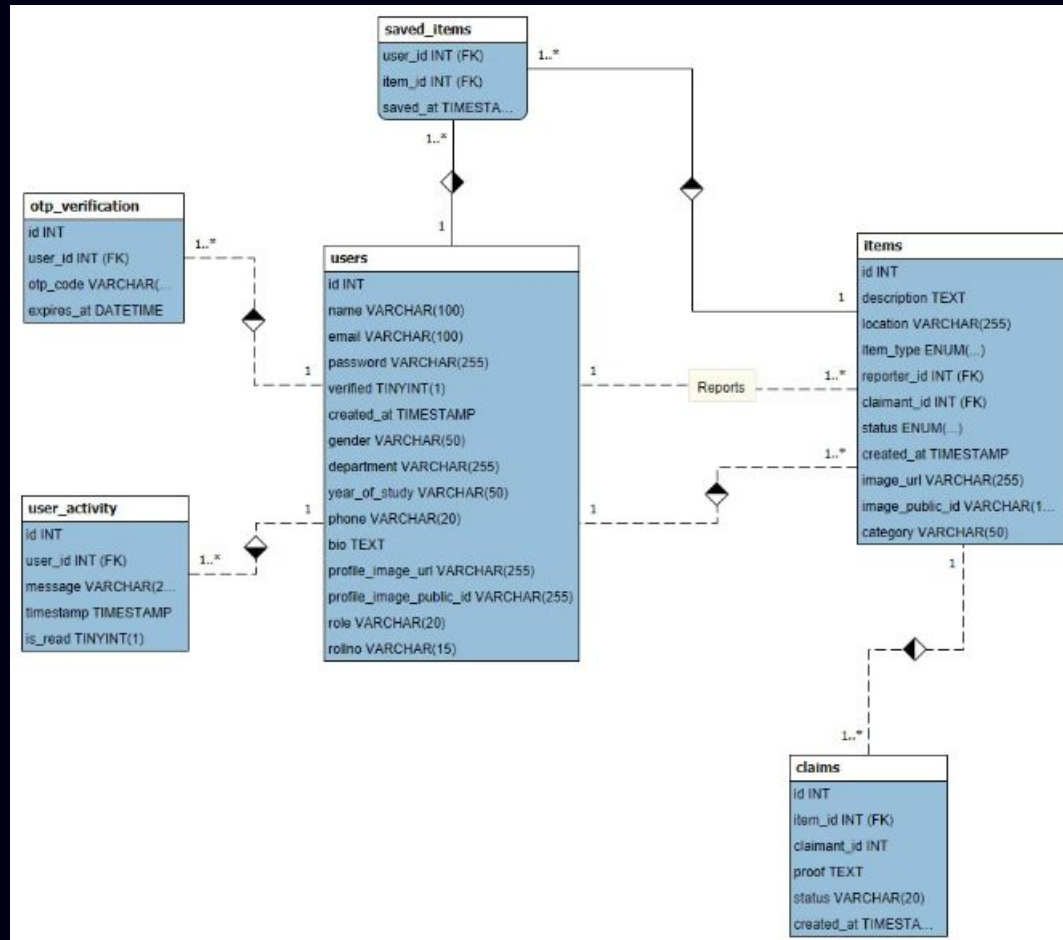
Database Overview

Content:

- This system helps users **report, claim, and save lost or found items**.

Main entities:

- **Users**: Register, report items, claim items, receive OTPs & notifications.
- **Items**: Lost/found items with details like location, type, and status.
- **Claims**: Users can claim items with proof and status tracking.
- **Saved Items**: Users can bookmark items to view later.
- **OTP Verification**: Ensures secure login with OTP.
- **User Activity**: Stores messages and alerts sent to users.

Entity Relationships, Cardinality & Types

Content:

- 🔁 **Relationships:**
- **Users - Items**
  - 1 User reports *many* items (1:N) → via `reporter_id`
  - 1 Item claimed by *one* user → `claimant_id` (optional FK)
- **Users - Claims - Items**
  - 1 Item   *many* Claims
  - 1 User can make *many* claims
- **Users - Saved_Items - Items**
  - Many-to-Many (M:N)
  - One user can save multiple items and vice versa
- **Users - OTP_Verification**
  - 1:N → One user can have many OTP entries
- **Users - User_Activity**
  - 1:N → One user can receive many messages

Entities, Attributes, and Keys in Lost & Found DB

✅ **Strong Entities:**

- **Users**
- Independent entity with `id` as primary key
- Stores user details like name, email, phone, role, etc.
- Connected to items, claims, saved items, OTPs, and activities

- ◆ **Items**
- Independent entity with `id` as primary key
- Describes reported items (description, location, type, status)
- Linked to the user who reported or claimed the item

✅ **Weak Entities:**

- **Claims** (dependent on Items and Users)
  - **Primary Key**: `id`
  - **Foreign Keys**: `item_id`, `claimant_id`
  - **Attributes**: proof, status, created_at
- **OTP_Verification**
  - **Primary Key**: `id`
  - **Foreign Key**: `user_id`
  - **Attributes**: otp_code, expires_at
- **User_Activity**
  - **Primary Key**: `id`
  - **Foreign Key**: `user_id`
  - **Attributes**: message, timestamp, is_read
- **Saved_Items** (association table)
  - **Composite Primary Key**: `user_id + item_id`
  - **Attributes**: saved_at

Summary and Next Steps
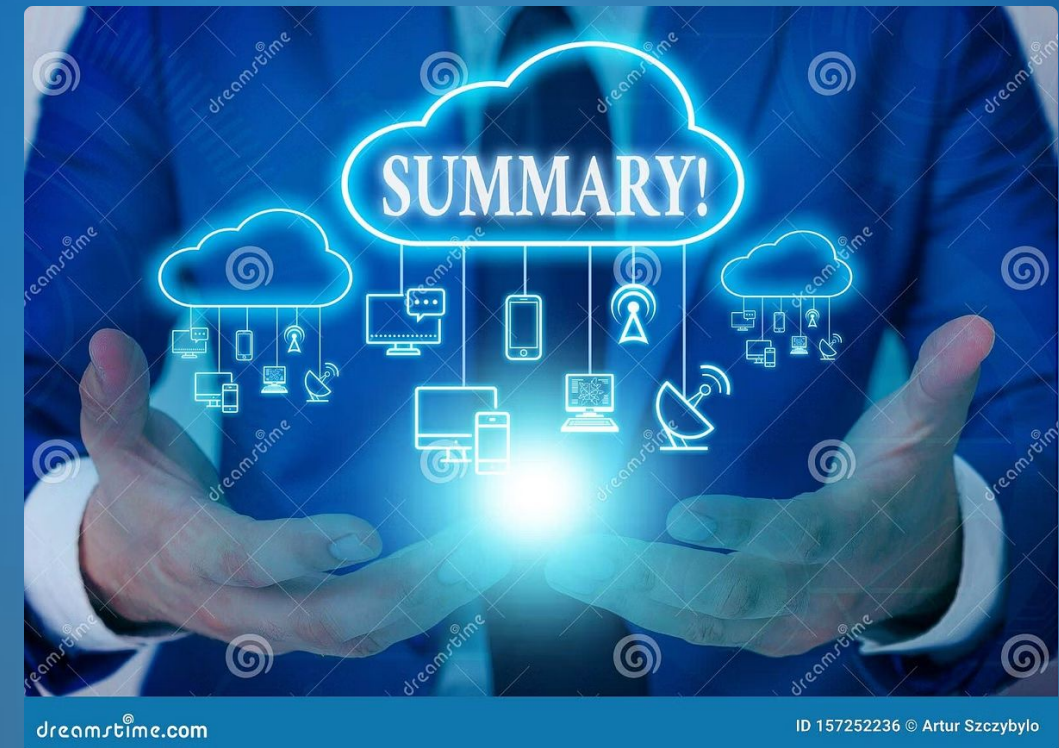
1 **Project Recap**

Developed a robust lost and found system with user authentication, reporting, claim management, and admin control.

2 **Technical Completion**

Database normalization and ER modeling ensure reliable data handling, complemented by a responsive front-end.

3 **Future Enhancements**

Plan to integrate notification systems, mobile app support, and enhanced security protocols for scalability.

SUMMARY!

dreamstime.com
ID 157252236 © Artur Szczybylo

Thank you

By- Group 25

- Vinay Kumar Shakya - IIT2023227

- Nikhil Chauhan - IIT2023217

- Aniket Kumar - IIT2023230

- Abhinav Kumar- IIT2023218

- Aastha Mangesh Borkar- IIB2023031