INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD

DEPARTMENT OF COMPUTER SCIENCE AND IT

SEMESTER PROJECT

COURSE: INTRODUCTION TO DATABASE SYSTEM

COURSE CODE: CS 242

PROFESSOR: MR IMRAN SAEED

SECTION: C

SUBMITTED BY:

- ***WAHAB EJAZ (5033-FOC/BSCS/F23)***
- ***ABUZAR AMIR (5028-FOC/BSCS/F23)***
- ***ZARAR MOAVIYA (5022-FOC/BSCS/F23)***

Table of Contents

Chapter 1: Introduction

1.1 Introduction

In any educational institution, it is common for students and staff to misplace or lose their personal belongings. Items such as books, identity cards, water bottles, wallets, electronic devices, and clothing are often reported lost or found across various departments and locations. Without a proper system to manage these cases, the recovery process is inefficient, prone to delays, and frequently fails to return items to their rightful owners.

This project introduces a "Lost and Found Management System for IIUI (International Islamic University Islamabad)," a relational database system built to efficiently manage the processes of reporting, tracking, and claiming lost and found items. The system is developed using Microsoft Access, employing relational tables, forms for easy data entry, queries for data retrieval, and reports for administrative analysis.

The system is specifically designed for the IIUI environment, where thousands of students and staff members interact across a large campus, making the management of lost and found items a critical need. This project demonstrates core database design principles such as normalization, entity-relationship modeling, form creation, query development, and reporting.

1.1.1 Purpose

The primary purpose of this document is to outline the requirements, functionality, and scope of the Lost and Found Management System (LFMS) for IIUI. It serves as a comprehensive Software Requirements Specification (SRS) that guides both development and future enhancements. The LFMS aims to:

- Streamline the process of recording lost and found items.
- Reduce manual errors and redundancy in information handling.
- Provide a structured and searchable database for quick access.
- Improve transparency, accountability, and item recovery efficiency.

1.1.2 Scope

The Lost and Found Management System is designed to:

- Register lost and found item reports submitted by students or staff.
- Store detailed information about items, their description, category, and color.
- Maintain records of the location where items were lost or found.
- Allow administrators to verify and approve item claims.
- Enable authorized users to manage data through user-friendly forms.
- Provide printed reports for summary and administrative use.

This system does not include automated matching or artificial intelligence for item comparison; all claims are verified manually by administrators.

1.1.3 Intended Audience

The following are the primary stakeholders and audience for this system:

- Project Supervisor: Reviews system design, functionality, and documentation.
- Database Developers: Responsible for implementing tables, relationships, and queries.
- University Admin Staff: End users who will operate and manage the system.

1.1.4 Definitions, Acronyms, and Abbreviations

Term	Description
LFMS	Lost and Found Management System
IIUI	International Islamic University Islamabad
DBMS	Database Management System
ERD	Entity Relationship Diagram
SRS	Software Requirements Specification
CRUD	Create, Read, Update, Delete

2. Overall Description

2.1 Product Perspective

The LFMS is a standalone desktop database solution created in Microsoft Access. It features normalized relational tables that are connected through appropriate relationships. The project uses built-in Access capabilities to develop interactive forms, dynamic queries for data search and filtering, and printable reports.

2.2 Product Functions

The system offers the following functionalities:

- Record reports of lost and found items.
- Store and categorize item details like name, color, and description.
- Maintain students and admin details for tracking and verification.
- Record the location where the item was found or lost.
- Allow admins to view, verify, and approve claims.
- Generate reports listing lost items, found items, claims, and statistics.

2.3 User Classes and Characteristics

User Role	Description
Admin	Can add, edit, and delete records, verify claims, and generate reports.

User Role	Description
Student	Can report a lost item or claim a found item by filling out a form.
Viewer	Can only view item listings and basic info (optional).

3. System Objectives

The main goals of this project are to:

1. Digitize the Lost and Found Process: Eliminate manual record-keeping and provide a centralized database.
2. Improve Traceability: Track when and where an item was found/lost, who found it, and who claimed it.
3. Enhance Efficiency: Allow authorized personnel to quickly search for and filter items using queries.
4. Ensure Data Accuracy: Reduce the risk of human error through validation and structured data input forms.
5. Support Future Growth: Provide a scalable structure to add new features such as login systems, notifications, and automatic matching algorithms in later versions.

4. Benefits of the System

- Centralization: All data is stored in one place, improving accessibility.
- Accountability: Each claim is linked to a student and verified by an admin.
- Search Capability: Dynamic queries help search lost/found items by name, category, color, or location.
- User-Friendly: Easy-to-navigate forms and interface make the system accessible even to non-technical staff.
- Reporting: Generate printable reports summarizing monthly or yearly statistics.

5. Limitations

- Only accessible on computers with Microsoft Access installed.
- No multi-user support in real-time (single-user desktop model).
- No automatic image attachment for item verification.
- Claim verification is manual; not based on AI or pattern recognition.

Chapter 2 – Database Design

2.1 Entity Classes & Attributes

Here we describe each entity class involved in the system. These entities represent the real-world objects and participants in the **Lost and Found management process at International Islamic University Islamabad (IIUI).** Every entity is defined with at least five attributes, including static and dynamic attributes as required.

2.1.1. Admin

Purpose:
The Admin is the central authority responsible for managing records in the Lost and Found system. Admins verify claims, enter records for found items, and manage student requests.

Attributes:

- **AdminID** *(Primary Key)*: Unique identifier for each admin.
- **Username**: Login name for the admin.
- **Password**: String used for system authentication.
- **PhoneNo**: Admin's contact number for internal communication.
- **Email**: Official email address of the admin

2.1.2. Student

Purpose:
Students are the users of the system who can report lost items, register found items and make claims. Each student has a unique registration number used for identification.

Attributes:

- **RegNo** *(Primary Key)*: Unique registration number for each student.
- **StudentName**: Full name of the student.
- **Email**: Student's email address for communication and system access.
- **ContactNo**: Mobile number for contacting the student.
- **Department**: The department to which the student belongs (e.g., CS, BBA, IR).

2.1.3. Claim

Purpose:
This entity tracks the claiming process of lost or found items. It links students, lost items, found items, and admins who approve or reject claims.

Attributes:

- **ClaimID** *(Primary Key)*: Unique identifier for each claim.
- **ClaimDate**: The date when the claim was submitted.
- **RegNo** *(Foreign Key)*: The student who made the claim.
- **LostID** *(Foreign Key)*: The ID of the lost item being claimed.
- **FoundID** *(Foreign Key)*: The ID of the found item being claimed.
- **AdminID** *(Foreign Key)*: The admin who processes the claim.

2.1.4. Item

Purpose:
This is a parent entity that contains common details for all items, whether lost or found. It helps reduce redundancy in data storage.

Attributes:

- **ItemID** *(Primary Key)*: Unique identifier for each item.
- **ItemName**: General name of the item (e.g., "Wallet", "Laptop").
- **Category**: Type of item (e.g., Electronics, Documents, Accessories).
- **Description**: Additional details about the item (brand, marks, etc.).
- **Color**: Main color of the item for easier identification.

2.1.5. FoundItem

Purpose:
This entity stores data about items that are found by students or staff and reported to the Lost and Found system.

Attributes:

- **FoundID** *(Primary Key)*: Unique identifier for each found item.
- **ItemID** *(Foreign Key)*: Links to the Item entity.
- **FounderContact**: Contact number of the person who found the item.
- **FoundDate**: Date when the item was found.
- **LocationID** *(Foreign Key)*: Location where the item was found.

2.1.6. LostItem

Purpose:
This entity records the details of items that students have reported as lost.

Attributes:

- **LostID** *(Primary Key)*: Unique identifier for each lost item.
- **ItemID** *(Foreign Key)*: Links to the Item entity.
- **RegNo** *(Foreign Key)*: Student who reported the item lost.
- **LostTime**: Date and time when the item was lost.
- **LocationID** *(Foreign Key)*: Where the item was last seen.

2.1.7. Location

Purpose:
Location stores information about various places on campus where items can be lost or found. It is used to help locate or trace missing items.

Attributes:

- **LocationID** *(Primary Key)*: Unique ID for each location.
- **PlaceName**: Name of the place (e.g., "Library", "Café").
- **Floor**: The floor on which the location exists (e.g., Ground, First).
- **Description**: Additional notes about the location (e.g., near stairwell).

These entities form the backbone of our database system. Each plays a vital role in maintaining a structured record of interactions, reducing confusion, and providing clear, trackable data to users and administrators.

2.2 Relationships Between Entities

2.2.1. Admin – Claim

Relationship: One-to-Many
Explanation: One Admin can process multiple Claims, but each Claim is handled by only one Admin.

- **AdminID** in Claim is a foreign key referring to Admin.

2.2.2. Student – LostItem

Relationship: One-to-Many
Explanation: A single student can report multiple lost items. Each LostItem is reported by one student.

- **RegNo** in LostItem refers to Student.

2.2.3. Student – Claim

Relationship: One-to-Many
Explanation: A student may claim multiple items over time, but each claim belongs to a single student.

- **RegNo** in Claim refers to Student.

2.2.4. Item – LostItem

Relationship: One-to-One (per lost entry)
Explanation: Each lost item refers to a single unique item entry from the Item table.

- **ItemID** in LostItem is a foreign key from Item.

2.2.5. Item – FoundItem

Relationship: One-to-One (per found entry)
Explanation: Each found item also corresponds to a single item from the Item table.

- **ItemID** in FoundItem is a foreign key from Item.

2.2.6. Location – LostItem

Relationship: One-to-Many
Explanation: Multiple items can be reported lost at the same location. Each LostItem has one location.

- **LocationID** in LostItem refers to Location.

2.2.7. Location – FoundItem

Relationship: One-to-Many
Explanation: Many found items can be reported at a single location.

- **LocationID** in FoundItem refers to Location.

2.2.8. LostItem – Claim

Relationship: One-to-One (optional)
Explanation: A LostItem may be claimed (or not), and each claim can optionally reference one LostItem.

- **LostID** in Claim is a foreign key from LostItem.

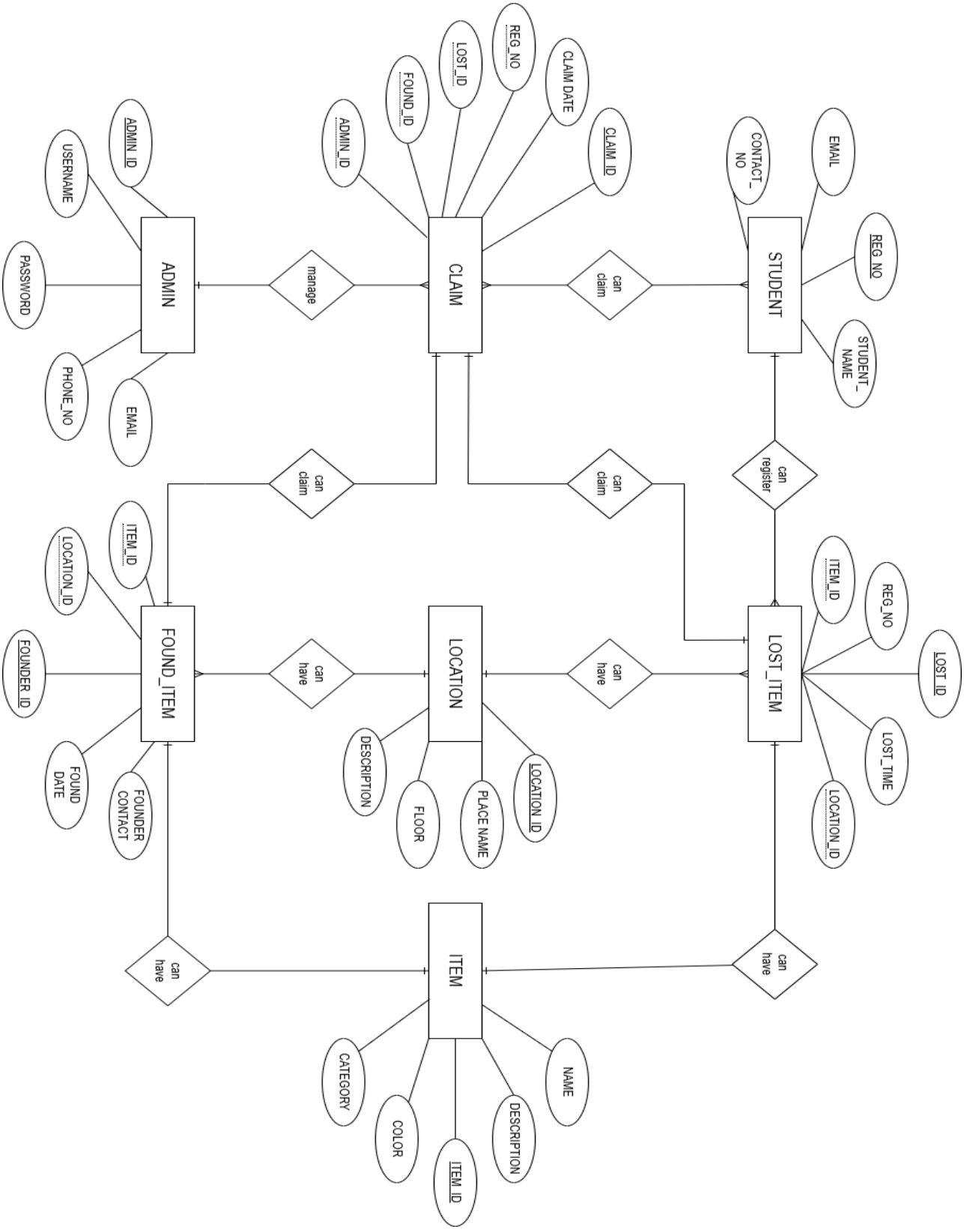
2.2.9. FoundItem – Claim

Relationship: One-to-One (optional)
Explanation: A FoundItem may be linked to a Claim when someone tries to match and recover it.

- **FoundID** in Claim is a foreign key from FoundItem.

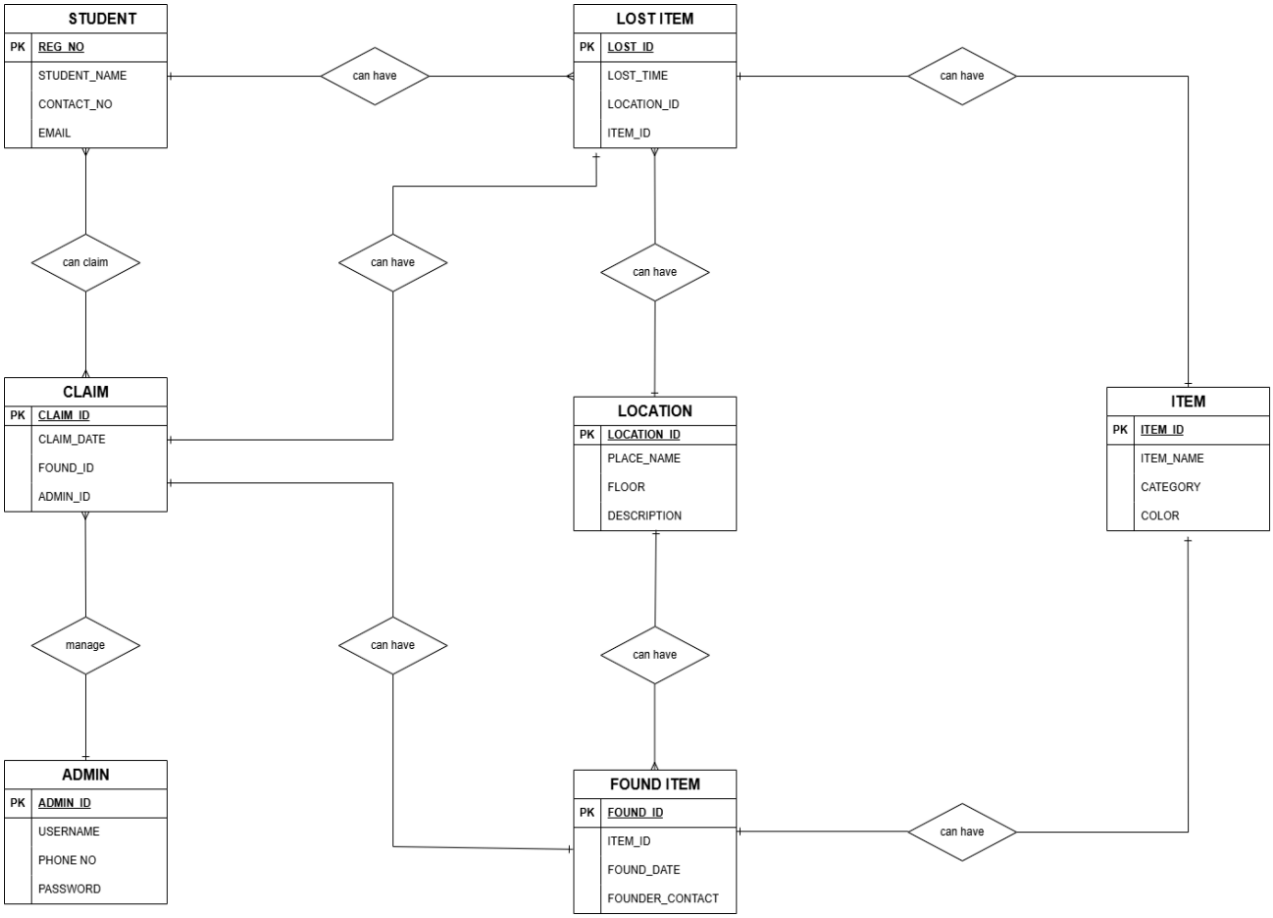
2.3 ERD (Entity-Relationship Diagram)

We used **dbdiagram.io** to draw the ERD. It visually represents all entity classes, their attributes, primary keys, and the relationships between them.



2.4 Relational Model Diagram

The following relational model diagram visually represents how the database tables (entities) in the Lost and Found Management System are structured and interconnected. It highlights the relationships among entities such as Students, Items, Claims, and Locations, ensuring a normalized and efficient database design.



2.5 NORMALIZATION OF TABLES

🔷 ADMIN (3NF ✅)

AdminID (PK)	Username	Password	PhoneNo	Email
A01	wahab	pass123	03001234567	wahab@iiui.edu.pk
A02	abuzar	pass456	03007654321	abuzar@iiui.edu.pk
A03	zarar	pass789	03111223344	zarar@iiui.edu.pk
A04	abdullah	pass321	03451239876	abdullah@iiui.edu.pk
A05	umer	pass654	03219876543	umer@iiui.edu.pk

🔍 3NF Explanation:

- 1NF: Atomic values, no repeating groups.
- 2NF: No partial dependency (AdminID is the only key).
- 3NF: No transitive dependencies. Email, phone, etc., depend only on AdminID.

🔷 STUDENT (3NF ✅)

RegNo (PK)	StudentName	Email	ContactNo	Department
S001	anees	anees@iiui.edu.pk	03123456789	CS
S002	wasay	wasay@iiui.edu.pk	03001239876	BBA
S003	ehsan	ehsan@iiui.edu.pk	03111222333	EE
S004	hamza	hamza@iiui.edu.pk	03334567890	Islamic St.
S005	ali	ali@iiui.edu.pk	03007894561	SE

🔍 3NF Explanation:

- Each attribute is atomic and fully functionally dependent on the RegNo.
- No transitive dependencies like Department Name → Department Code.

🔷 ITEM (3NF ✅)

ItemID (PK)	ItemName	Category	Description	Color
I01	Wallet	Personal	Black leather wallet	Black
I02	Notebook	Stationery	Red lined notebook	Red
I03	Glasses	Personal	Reading glasses	Black
I04	Water Bottle	Accessory	Steel thermos	Silver
I05	USB Drive	Electronics	32GB Kingston	Blue

🔍 3NF Explanation:

- No repeating groups or partial dependencies.
- No non-key attribute depends on another non-key (e.g., Description doesn’t determine Category).

🔷 LOCATION (3NF ✅)

LocationID (PK)	PlaceName	Floor	Description
L01	Main Library	2nd	Near reading area
L02	Cafeteria	Ground	Near vending machines
L03	Admin Block	1st	Front desk area
L04	Masjid	Ground	Near shoe racks
L05	CS Dept	Basement	Near entrance

🔍 3NF Explanation:

- No transitive dependencies (e.g., Floor doesn't imply Description).
- Each attribute depends solely on LocationID.

🔷 FOUND ITEM (3NF ✅)

FoundID (PK)	ItemID (FK)	FounderContact	FoundDate	LocationID (FK)
F001	I01	03111222333	2025-05-20	L01
F002	I02	03009877654	2025-05-21	L02
F003	I03	03456712345	2025-05-23	L03
F004	I04	03331234567	2025-05-25	L04
F005	I05	03213456789	2025-05-26	L05

🔍 3NF Explanation:

- All attributes are atomic and fully dependent on FoundID.
- No transitive dependencies (e.g., LocationID doesn’t determine Contact).

🔷 LOST ITEM (3NF ✅)

LostID (PK)	ItemID (FK)	RegNo (FK)	LostTime	LocationID (FK)
LST001	I01	S001	2025-05-18	L01
LST002	I02	S002	2025-05-19	L02

LST003	I03	S003	2025-05-20	L03
LST004	I04	S004	2025-05-22	L04
LST005	I05	S005	2025-05-23	L05

🔍 3NF Explanation:

- Each attribute depends only on LostID.
- No derived or transitive dependency exists (e.g., you can't derive RegNo from ItemID alone).

🔷 CLAIM (3NF ✅)

ClaimID (PK)	ClaimDate	RegNo (FK)	LostID (FK)	FoundID (FK)	AdminID (FK)
C001	2025-05-22	S001	LST001	F001	A01
C002	2025-05-23	S002	LST002	F002	A02
C003	2025-05-24	S003	LST003	F003	A03
C004	2025-05-25	S004	LST004	F004	A04
C005	2025-05-26	S005	LST005	F005	A05

🔍 3NF Explanation:

- Fully normalized: All fields are atomic and depend only on the primary key ClaimID.
- No redundant data or transitive dependencies.

Chapter 3 – Implementation

3.1 DBMS Used

This chapter describes how the database project was created and implemented using **Microsoft Access (MS Access)**. We built the database for a Lost and Found Management System for **International Islamic University Islamabad (IIUI)**. We used a **relational database model**, created **tables for each entity**, and built **forms, queries, and reports** to make the system easy to use.

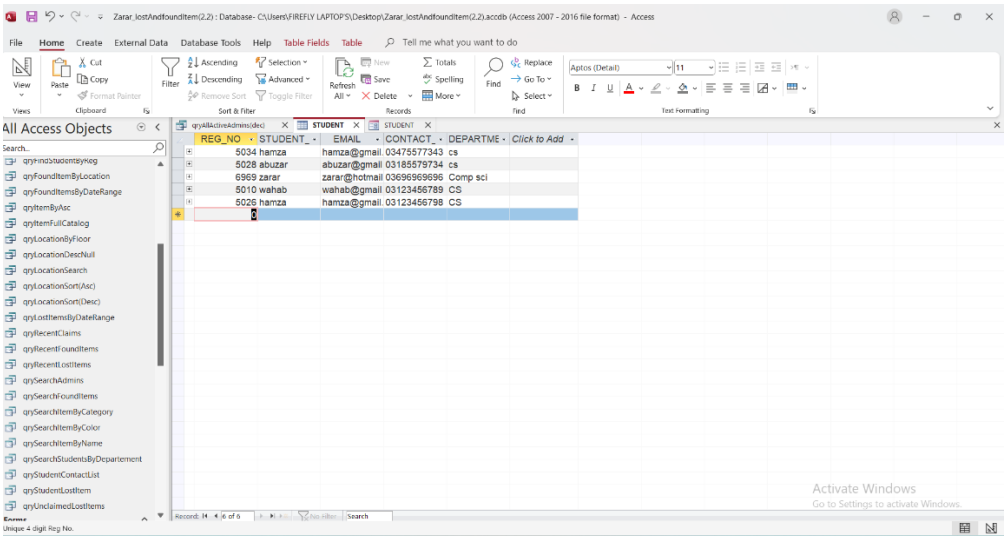
Database Management System Used

We used **Microsoft Access** because it is user-friendly, supports relational databases, and provides a graphical interface for designing tables, queries, forms, and reports — all in one place.

3.2 How Tables Were Created (Example: Student Table)

To create the Student table:

1. We opened MS Access and clicked on **"Create > Table Design"**.
2. In the design view, we added the following fields:
    - RegNo (set as **Primary Key**)
    - StudentName
    - Email
    - ContactNo
    - Department
3. We set appropriate **data types**:
    - Text for names, department, and email.
    - Number for contact no (if needed).
4. We saved the table as **Student**.



We repeated a similar process for all other tables: **Admin**, **Item**, **FoundItem**, **LostItem**, **Claim**, and **Location**. Foreign key fields were added to connect the tables based on relationships we described in the ERD.

3.3 How Queries Were Created

We created both **dynamic** and **static** queries in MS Access.

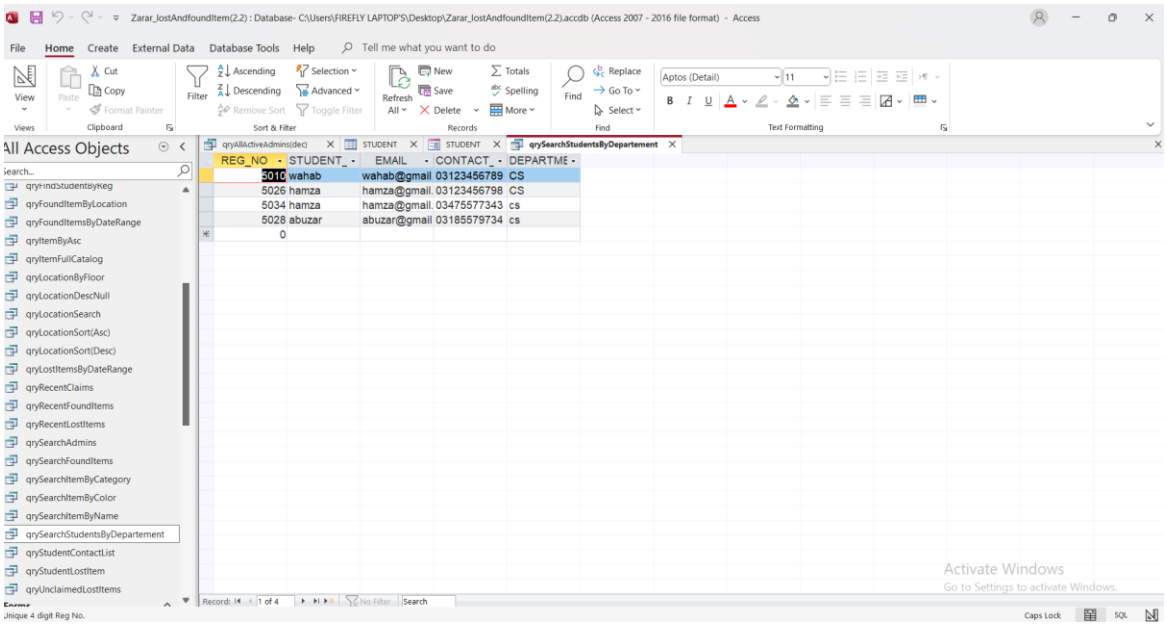
Example of a Static Query:

A static query gives fixed results. For example:

- A query that shows all students who belong to the "CS" department.
- This query always gives the same results unless the data changes.

How we created it:

1. Clicked on **Create > Query Design**.
2. Selected the Student table.
3. Added StudentName, RegNo, and Department.
4. In the criteria row for Department, we wrote "CS" and saved the query.



Example of a Dynamic Query:

A dynamic query asks the user to input a value each time. For example:

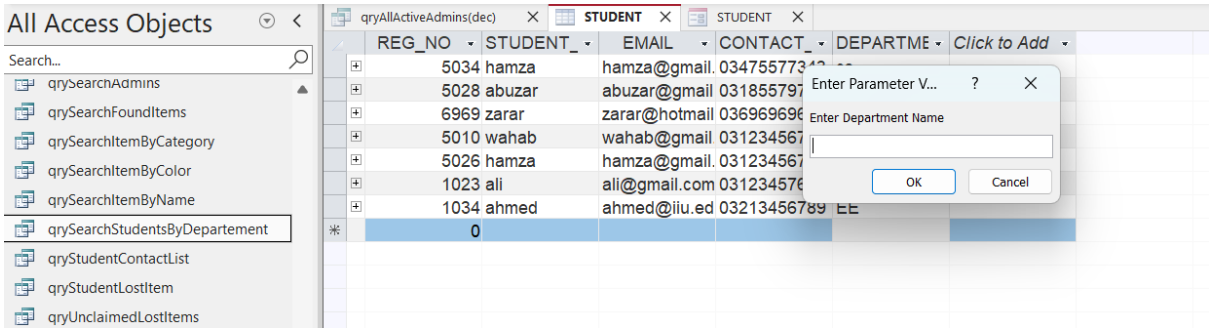
- A query that asks: “Enter department name” — and then shows students only from that department.

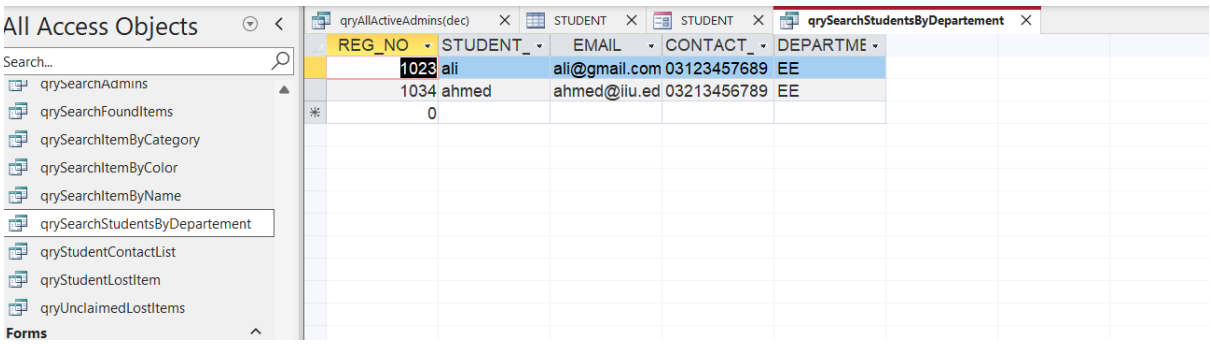
How we created it:

1. Same steps as above, but in the Department criteria field, we wrote:

   [Enter Department Name]

2. This makes the query dynamic — it works differently depending on what the user types.



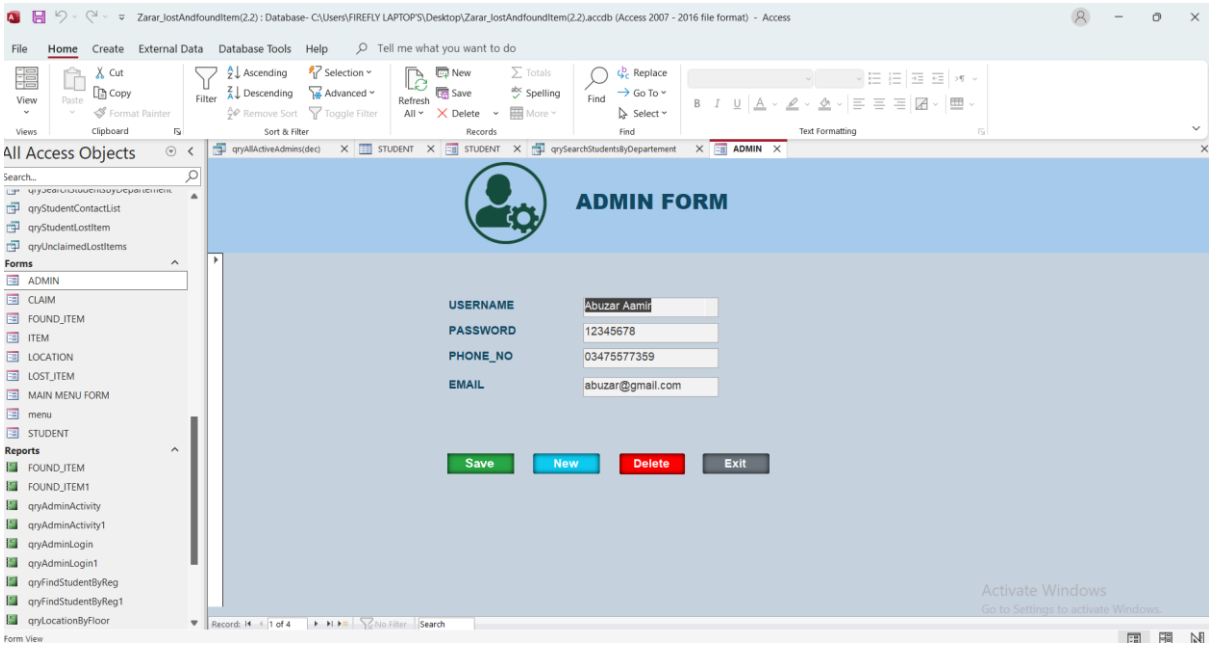


3.4 How Forms Were Created

Forms were made to make it easy for admins and users to enter or view data without directly opening the tables.

Example: Admin Form

1. We clicked on **Create > Form Wizard**.
2. Selected the Admin table.
3. Chose all fields: AdminID, Username, Password, PhoneNo, Email.
4. Finished the wizard and got a basic form.
5. We customized it by:
    - Changing the layout to "Columnar".
    - Adding professional-looking buttons like:
        - **Save/Register**: To add new admin.
        - **Next/Previous**: To move through records.
        - **Close**: To exit the form.



3.5 Reports and Main Menu

We created **reports** to print or display data in a readable format, like summaries of claims, lists of found items, etc.

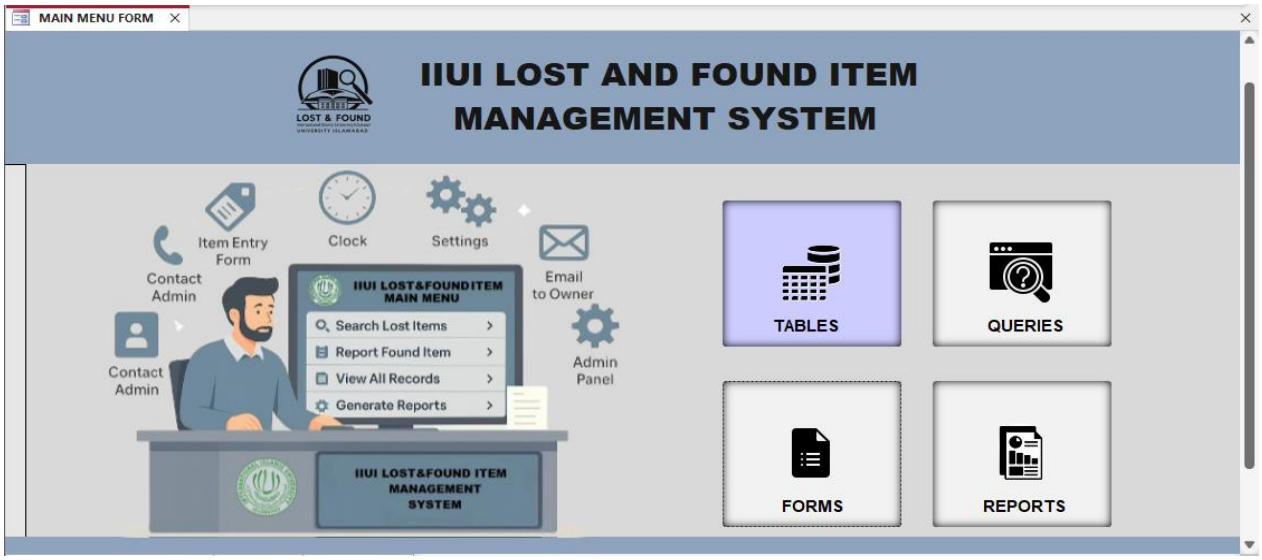
Example:

A report that shows all found items by date.

We also made a **Main Menu Form**:

- This form has buttons to open other forms and reports.

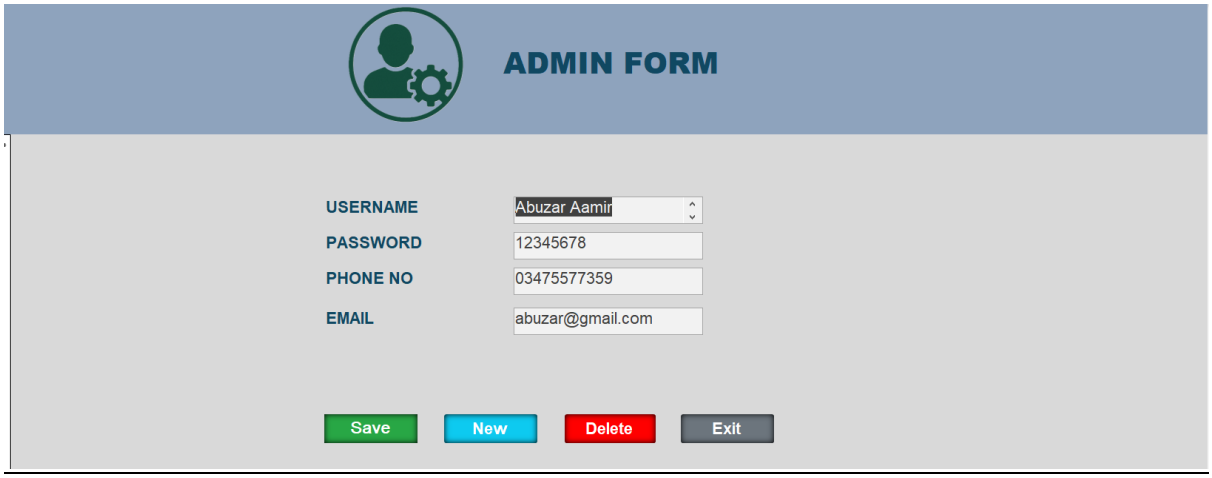
- Example: “Add New Student”, “View Claims”, “Print Report”.

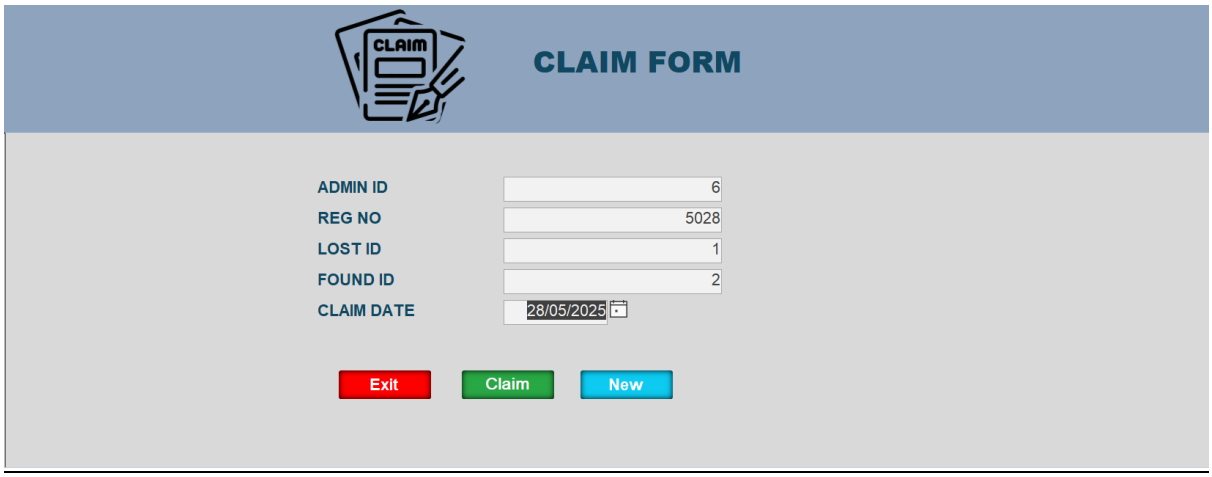


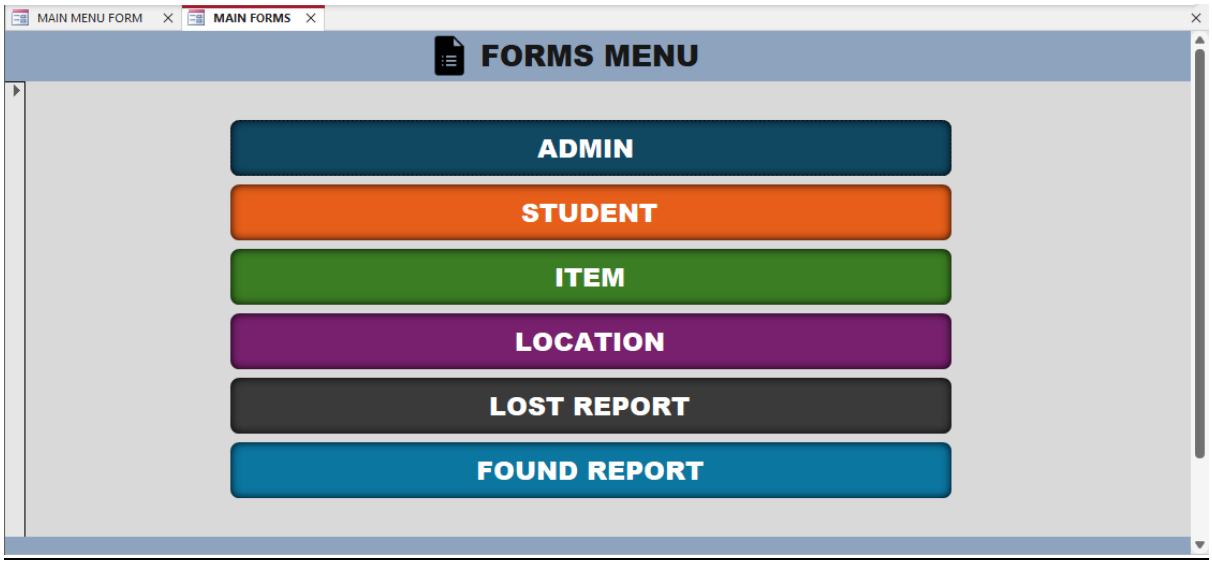
Appendix A – Screenshots

Here

Forms







FOUND ITEM

ITEM NAME	
ITEM ID	1
FOUNDER CONTACT	03475577359
FOUND DATE	27/05/2025
LOCATIONS	
LOCATION ID	1

Save | New | Exit

ITEM FORM

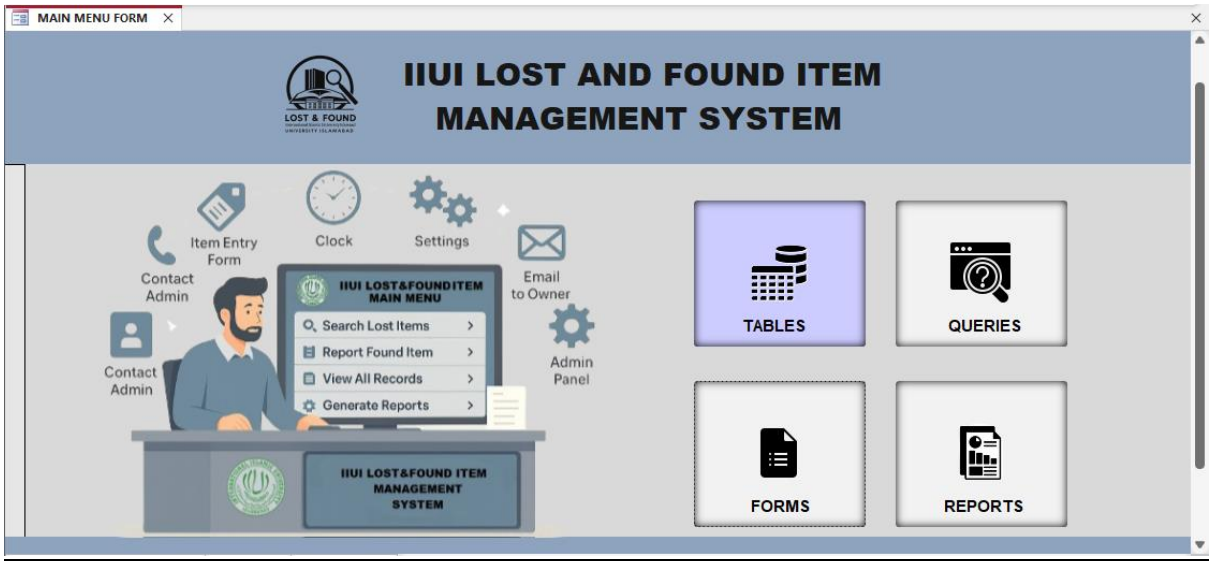
ITEM NAME	charger
CATEGORY	
COLOR	white
DESCRIPTION	white color charger

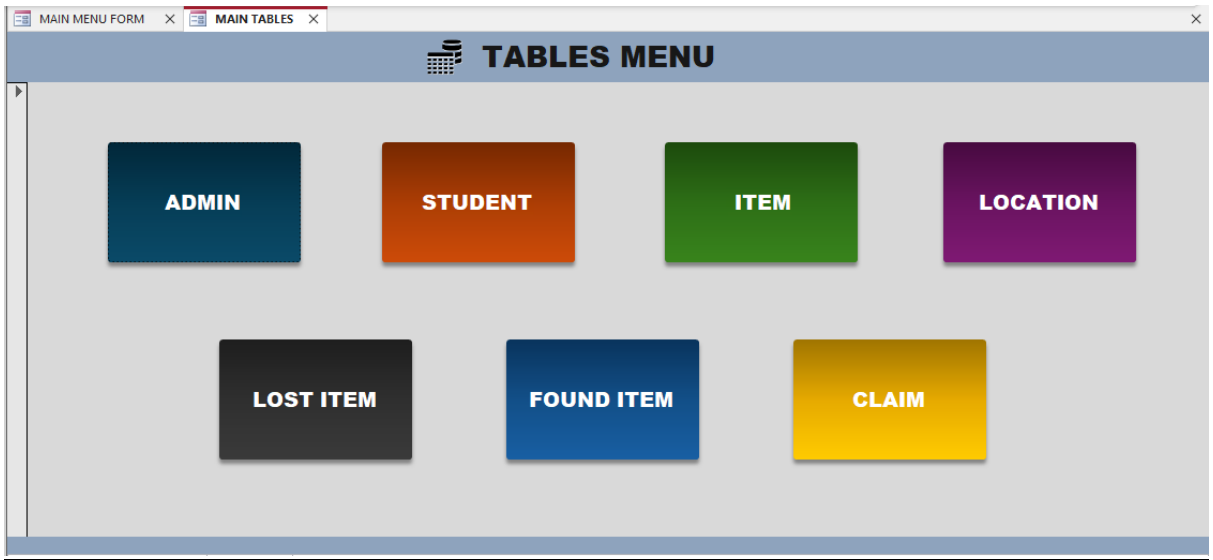
Report | New | Delete | Exit

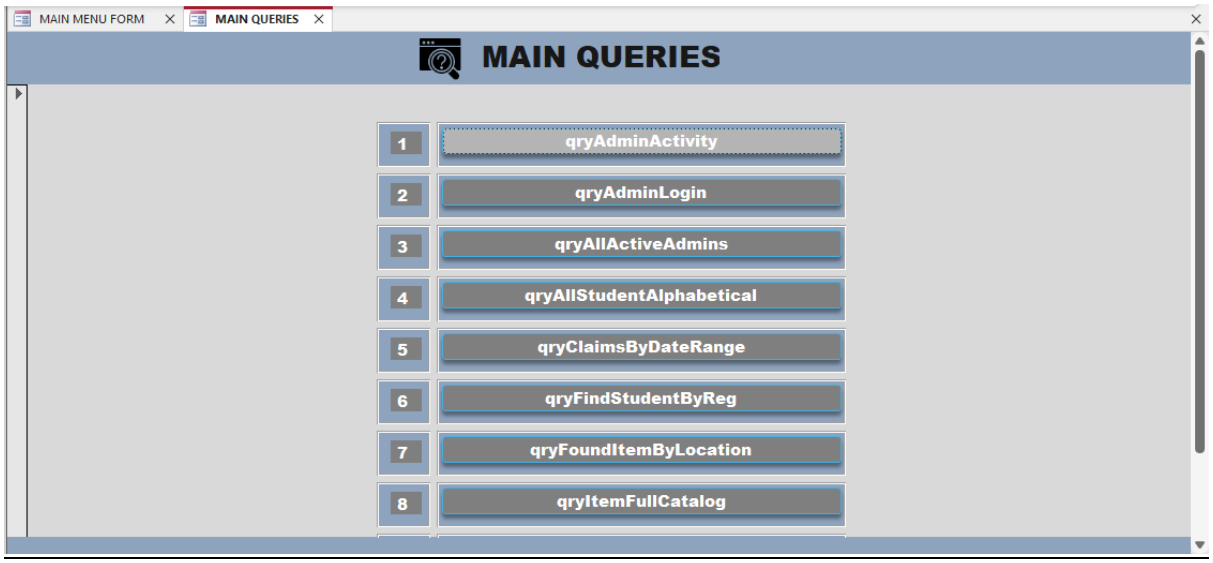
LOCATION

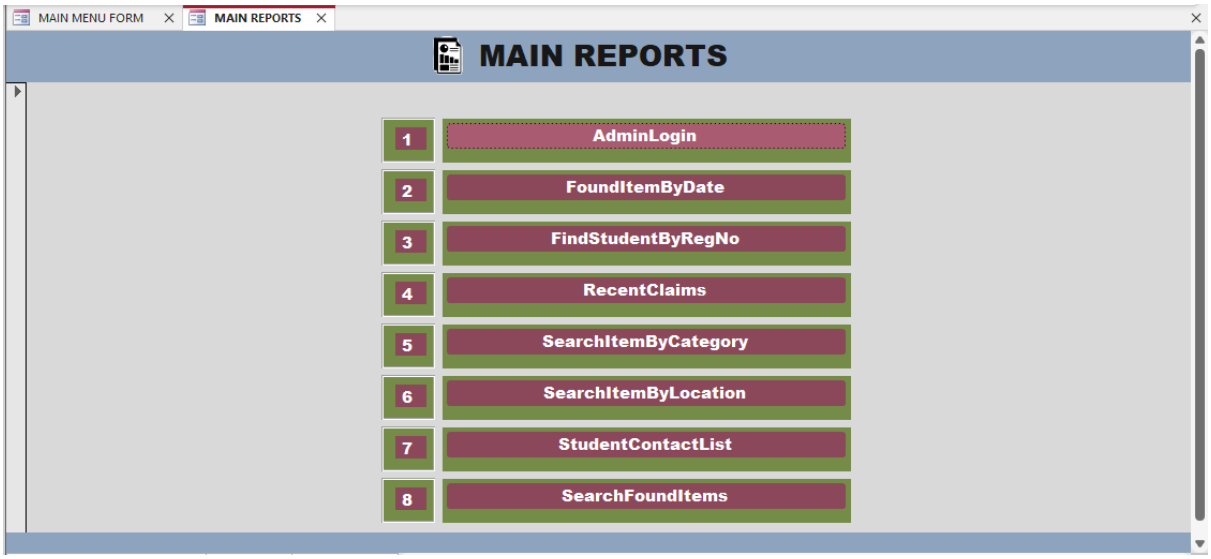
PLACE NAME	C Block
FLOOR	1st
DESCRIPTION	iiui near cs lab 3

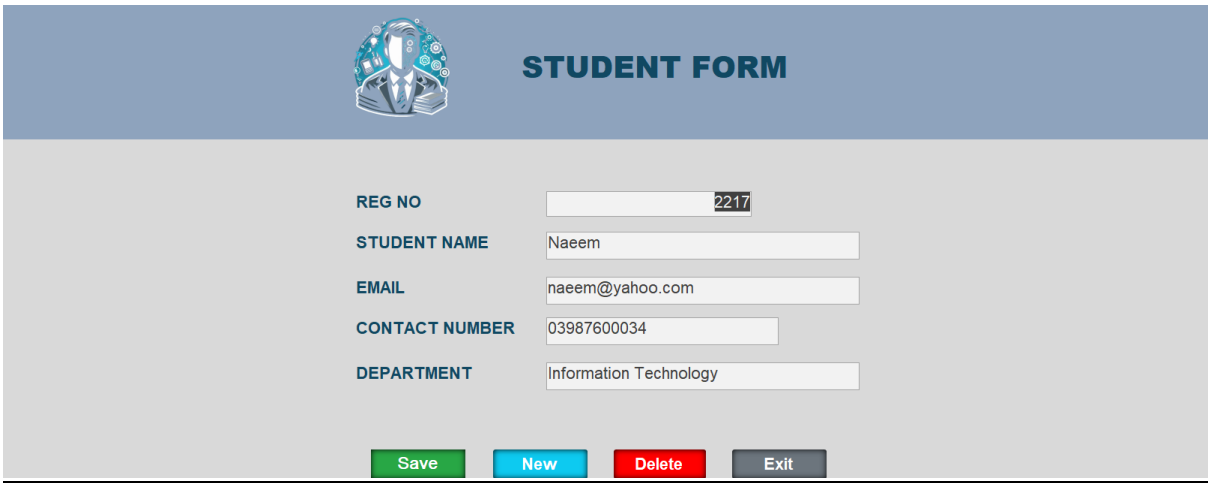
Save | New | Exit

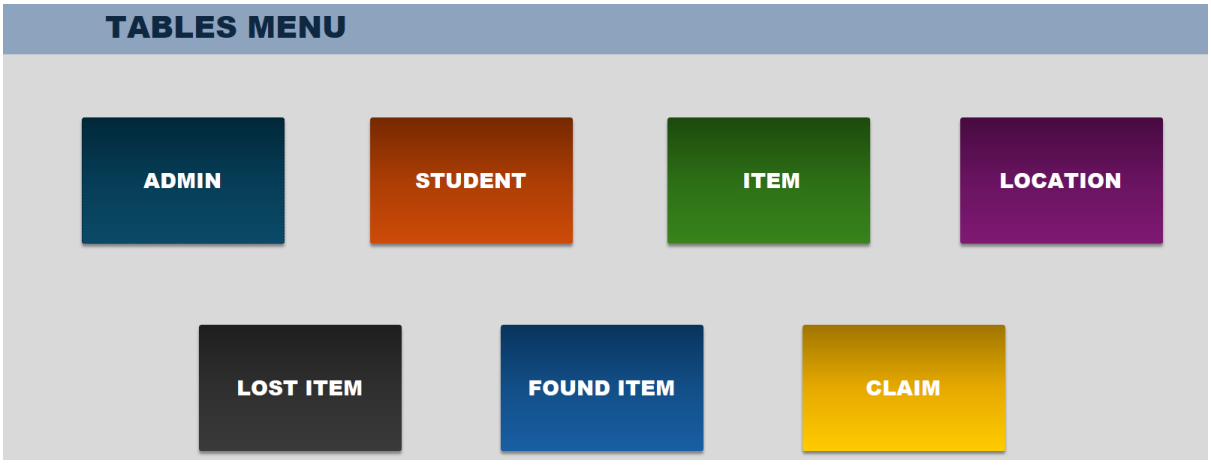

MAIN MENU FORM
IIUI LOST AND FOUND ITEM MANAGEMENT SYSTEM
LOST & FOUND
Item Entry Form
Clock
Settings
Contact Admin
Email to Owner
IIUI LOST&FOUND ITEM MAIN MENU
Search Lost Items
Report Found Item
View All Records
Generate Reports
Contact Admin
Admin Panel
IIUI LOST&FOUND ITEM MANAGEMENT SYSTEM
TABLES
QUERIES
FORMS
REPORTS

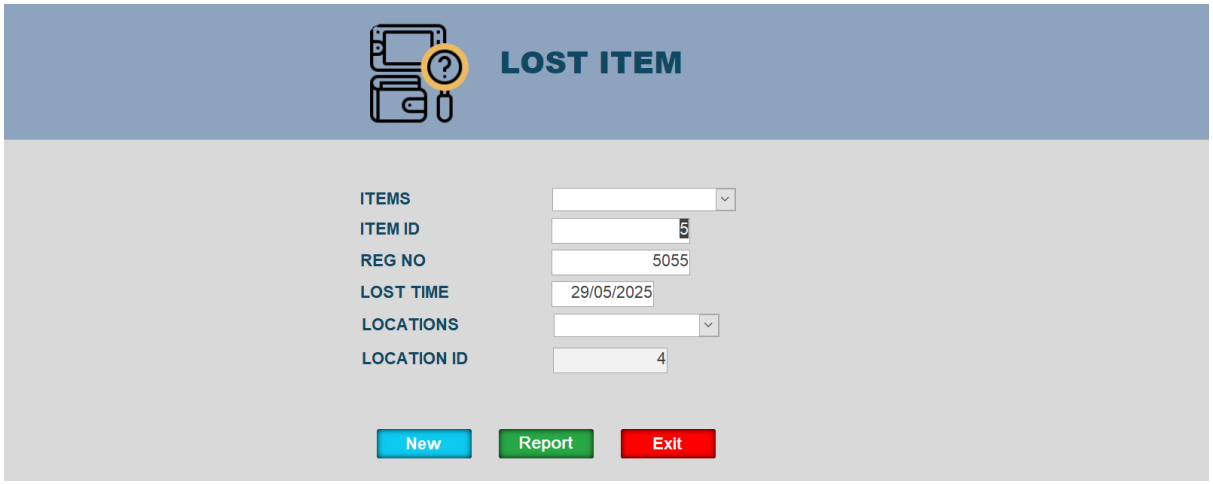

MAIN MENU FORM
MAIN TABLES
TABLES MENU
ADMIN
STUDENT
ITEM
LOCATION
LOST ITEM
FOUND ITEM
CLAIM


MAIN MENU FORM
MAIN QUERIES
MAIN QUERIES
1
qryAdminActivity
2
qryAdminLogin
3
qryAllActiveAdmins
4
qryAllStudentAlphabetical
5
qryClaimsByDateRange
6
qryFindStudentByReg
7
qryFoundItemByLocation
8
qryItemFullCatalog


MAIN MENU FORM
MAIN REPORTS
MAIN REPORTS
1
AdminLogin
2
FoundItemByDate
3
FindStudentByRegNo
4
RecentClaims
5
SearchItemByCategory
6
SearchItemByLocation
7
StudentContactList
8
SearchFoundItems


STUDENT FORM
REG NO
2217
STUDENT NAME
Naeem
EMAIL
naeem@yahoo.com
CONTACT NUMBER
03987600034
DEPARTMENT
Information Technology
Save
New
Delete
Exit


TABLES MENU
ADMIN
STUDENT
ITEM
LOCATION
LOST ITEM
FOUND ITEM
CLAIM



Reports

FIND ITEM BY COLOR

ITEM ID	ITEM NAME	CATEGORY	DESCRIPTION	COLOR
2	jacket	clothing	leather jacket with two outer pockets.	black

FOUND ITEM BY NAME

FOUND ID	FOUND DATE	ITEM NAME	CATEGORY
6	08/05/2025	register	books

FIND STUDENTS BY DEPARTMENT

REG NO	STUDENT NAME	EMAIL	CONTACT NUMBER	DEPARTMENT
5028	abuzar	abuzar@gmail.com	03185579734	cs

Saturday, 31 May 2025 Page 1 of 1

FIND STUDENT BY REG NO

REG NO	STUDENT NAME	EMAIL	CONTACT NUMBER	DEPARTMENT
5028	abuzar	abuzar@gmail.com	03185579734	cs

RECENT CLAIMS

CLAIM ID	CLAIM DATE	ADMIN ID	FOUND ID
2	28/05/2025	6	2

Page 1 of 1

REPORT END!