SkyTracks: An Airport Management System

A Project Report Submitted to

MANIPAL ACADEMY OF HIGHER EDUCATION

For Partial Fulfillment of the Requirement for the

Award of the Degree

Of

Bachelor of Technology

in

Computer and Communication Engineering

by

Suvan Kumar Shee, Gokul Nair, Armaan Gupta

230953380, 230953440, 230953382

Under the guidance of

Dr. Akshay KC (Lab Faculty 1)
Assistant Professor - Senior scale
Department of I&CT
Manipal Institute Of Technology
Manipal, Karnataka, India

Dr Shweta Pai
Assistant Professor
Department of I&CT
Manipal Institute of Technology
Manipal, Karnataka, India

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Unit of MAHE, Manipal

April 2025

ABSTRACT

The *Airport Management System (AMS)* is a comprehensive software solution designed to streamline airport operations through the integration of multiple functional modules. The system manages end-to-end processes such as flight scheduling, gate allocation, passenger handling, staff coordination, and baggage tracking to ensure seamless service delivery and operational efficiency.

Key features include real-time management of flight details such as departure and arrival schedules, delays, and route information. Passenger data is securely handled, supporting identity verification, travel details, and baggage association. The system also tracks gate availability, terminal usage, and facilitates smooth boarding processes. Lost and Found management ensures that misplaced items are recorded and traceable.

To maintain operational control and security, the system implements **Staff Registration and Login** functionalities. Staff can log in based on designated roles, enabling access to specific modules such as flight updates, passenger services, or baggage management, ensuring a role-based permission structure.

SkyTracks is developed using MySQL for data handling, with a user-friendly frontend built in HTML, CSS, and JavaScript. SkyTracks aims to enhance the passenger experience while ensuring airport operations remain efficient and secure.

[Information Systems]: Data Management; Database Design; Normalization; Query Optimization. [Human-Centred Computing]: User Interfaces; Personalization; Recommendation Systems.

[SDG9]: Industry, Innovation and Infrastructure

Table of Contents

List of Tables

1. Relations After Normalization

The final relations derived after applying normalization (up to BCNF) to the database schema

Relation Name	Attributes
Airport	AirportID, Name, Location, Terminals
Airline	AirlineID, Name, Code
Passenger	PassengerID, Name, DOB, PhoneNumber
Flight	FlightID, AirlineID, DepartureAirportID, ArrivalAirportID, DepartureTime, ArrivalTime
Gate	GateID, AirportID, Status, Terminal
Staff	StaffID, AirportID, Name
Works_For	StaffID, AirportID
Flight_Passenger	FlightID, PassengerID
Airport_Airline	AirportID, AirlineID
LostAndFound	ItemID, AirportID, DateFound

2. Functional Dependencies

Listing all functional dependencies identified during normalization.

Functional Dependency (FD)
FD1: AirportID → {AirportName, Location,Terminals}
FD2: StaffID → {StaffName, Role, AirportID}
FD3: AirlineID → {AirlineName, Code}
FD4: ItemID → {DateFound, AirportID, Description}
FD5: GateID → {AirportID, Terminal}
FD6: PassengerID → {PassengerName, DOB, PhoneNumber}
FD7: GateID → {AirportID, Terminal}
FD8: (BaggageTag, PassengerID) → {Weight, DepartureAirport, ArrivalAirport}

3. **Normalization Steps**

   Explaining how the universal relation was decomposed into smaller relations during normalization.

Step	Description
1st Normal Form (1NF)	Staff have a multivalued attribute {Role}
2nd Normal Form (2NF)	Removed partial dependencies by decomposing into smaller relations such as Airport, Staff
3rd Normal Form (3NF)	No transitive dependencies.
BCNF	Verified that all relations satisfy BCNF rules.

4. **Entity Descriptions**

   Providing a detailed description of each entity in the database.

Entity Name	Description
Staff	Represents administrators who manage Baggage, FlightUpdates .
User	Represents end-users who interact with the system to get flight Info.
Flight	Contains details about each flight, timings, departure and arrival airport.

Abbreviations

Here is a list of all the abbreviations used in our Airport Management System project:

- ERD: Entity-Relationship Diagram
- SDG: Sustainable Development Goals
- FD: Functional Dependency
- BCNF: Boyle-Codd Normal Form
- SRS: Software Requirements Specification
- GUI: Graphical User Interface

Chapter 1:
Introduction

1.1 Background and Context

Airports are complex hubs of activity involving passengers, staff, airlines, flights, baggage, and security operations. Managing these components efficiently is critical to ensuring smooth day-to-day operations and a positive travel experience. With the rise of smart infrastructure and digital transformation, there is a growing need for integrated systems that can automate and streamline airport processes.

However, many existing airport systems are either fragmented across departments or lack real-time coordination and flexibility. The SkyTracks Airport Management System addresses these challenges by providing a unified web-based platform that manages airport operations such as flight scheduling, passenger handling, baggage tracking, staff allocation, and gate management — all in a centralized and user-friendly environment.

1.2 Purpose of the Project

The primary purpose of this project is to develop a full-fledged airport management application that enables seamless coordination between different departments of an airport. The system supports features like flight and passenger tracking, real-time gate and terminal management, lost and found log monitoring, baggage handling, and secure role-based staff login and registration.

By digitizing and automating operational workflows, this system aims to improve overall airport efficiency, enhance user experience, and support sustainable infrastructure development in line with Sustainable Development Goal (SDG) 9: Industry, Innovation and Infrastructure.

1.3 Problem Statement

Airports often face issues such as flight delays, baggage mismanagement, inefficient staff coordination, and poor passenger service due to the lack of integrated management systems. Existing solutions may not provide centralized data handling, real-time updates, or role-based access for staff operations.

The SkyTracks Airport Management System seeks to solve these problems by offering a dynamic, modular platform that allows departments to collaborate effectively. It ensures real-time data tracking, secure staff authentication, centralized management of airport operations, and enhanced service delivery.

1.4 Objectives

The key objectives of this project are:

• To design and develop a centralized airport management system with modular support for flights, staff, passengers, baggage, and gate management.

• To implement secure staff registration and login with role-based access to system functionalities.

• To provide real-time updates on flights, delays, baggage tracking, and terminal assignments.

• To enhance service efficiency and reduce resource mismanagement through automated workflows.

1.5 Significance of the Project

This project is significant for several reasons:

• It improves operational efficiency by digitizing key airport processes.

• It enhances passenger experience by providing accurate and timely information on flights and services.

• It ensures staff accountability and secure access management through dedicated login systems.

• It supports scalable airport infrastructure aligned with digital transformation initiatives.

• It contributes to SDG 9: Industry, Innovation and Infrastructure and indirectly supports SDG 11: Sustainable Cities and Communities through efficient urban transport hubs.

1.6 Scope of the Project

The scope of this project includes:

• Developing a web-based application with modules for flight management, passenger handling, baggage tracking, gate/terminal assignment, lost and found, and staff management.

• Implementing secure authentication systems for staff login and role-specific access.

• Designing a relational database structure to manage airport entities including flights, passengers, staff, baggage, and terminals.

• Creating an intuitive user interface for seamless interaction across departments.

Chapter 2:
Data Design

2.1 Entity-Relationship (ER) Diagram

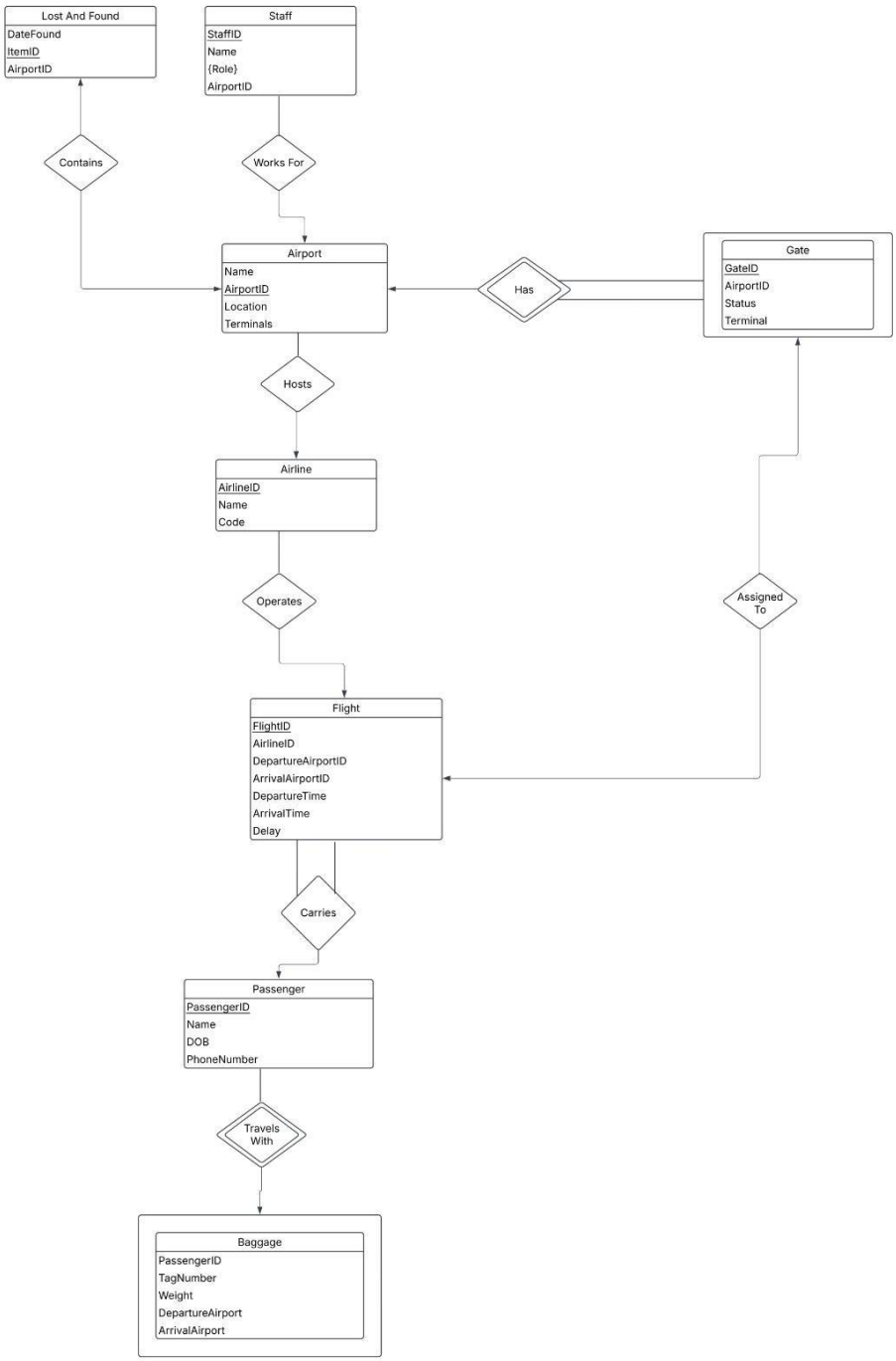

Figure 2.1: Entity-Relationship Diagram.

Entities:

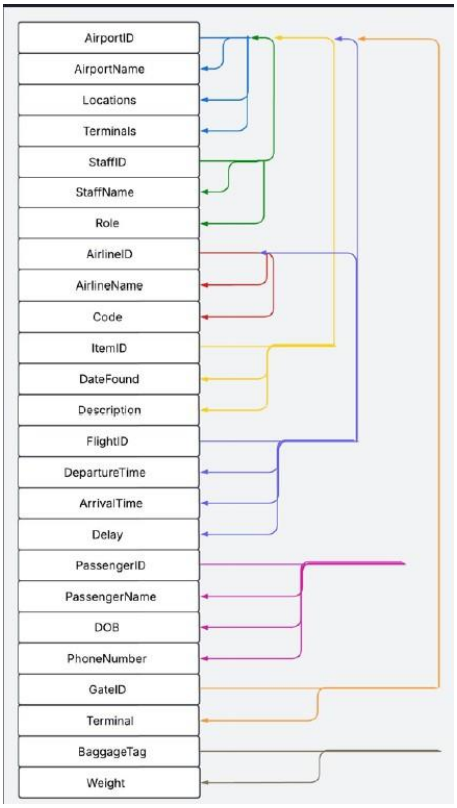

Fig 2.2 - Functional Dependencies

Airport: AirportID, Name, Location, Terminals
Staff: StaffID, Name, {Role}, AirportID **Airline**:
AirlineID, Name, Code
Lost And Found: ItemID, DateFound, AirportID, Description
Flight: FlightID, AirlineID, DepartureAirportID, ArrivalAirportID, DepartureTime, ArrivalTime,Delay)
Passenger: PassengerID, Name, DOB, PhoneNumber
Gate : GateID (PK), AirportID (FK), Terminal
Baggage : BaggageTag (PK), PassengerID (FK), Weight, DepartureAirport, ArrivalAirport

Relationships:
Works_For(StaffID, AirportID) - N:1

Has(GateID, AirportID) - N:1

Contains(AirportID, ItemID) - 1:M

Hosts(AirportID, AirlineID) - 1:N

Operates(AirlineID, FlightID) - 1:N

Assigned_To(GateID, FlightID) - 1:1

Carries(FlightID, PassengerID) - 1:N

Travels_With(PassengerID, BaggageTag) - 1:N

2.2 Normalization

Before Normalization:

Airport: AirportID, AirportName, Location, Terminals
- **Staff**: StaffID, StaffName, {Role}, AirportID
- **Airline**: AirlineID, AirlineName, Code
- **Lost And Found**: ItemID, DateFound, AirportID, Description
- **Flight**: FlightID, AirlineID, DepartureAirportID, ArrivalAirportID, DepartureTime, ArrivalTime, Delay)
- **Passenger**: PassengerID, PassengerName, DOB, PhoneNumber
- **Gate :** GateID (PK), AirportID (FK), Terminal
- **Baggage :** BaggageTag (PK), PassengerID (FK), Weight, DepartureAirport, ArrivalAirport
- **Flight_Passenger**(FlightID, PassengerID)
- **Flight_Gate**(FlightID, GateID)
- **Works_For**(StaffID, AirportID)
- **Airport_Airline**(AirportID, AirlineID)

The data has been checked for 1NF,2NF,3NF and BCNF. The derived relations after normalization are:

- **Airport** (AirportID, AirportName, Location, Terminals)
- **Staff** (StaffID, StaffName, Role, AirportID)
- **Airline** (AirlineID, AirlineName, Code)
- **LostAndFound** (ItemID, DateFound, AirportID, Description)
- **Flight** (FlightID, AirlineID, DepartureAirportID, ArrivalAirportID, DepartureTime, ArrivalTime, Delay)
- **Passenger** (PassengerID, PassengerName, DOB, PhoneNumber)
- **Gate** (GateID, AirportID, Terminal)
- **Baggage** (BaggageTag,PassengerID,Weight, DepartureAirport, ArrivalAirport)
- **Flight_Passenger** (FlightID, PassengerID)
- **Flight_Gate** (FlightID, GateID)
- **Works_For** (StaffID, AirportID)
- **Airport_Airline** (AirportID, AirlineID)
- **Staff_Role** (StaffID, Role)

2.3 Final Database Schema

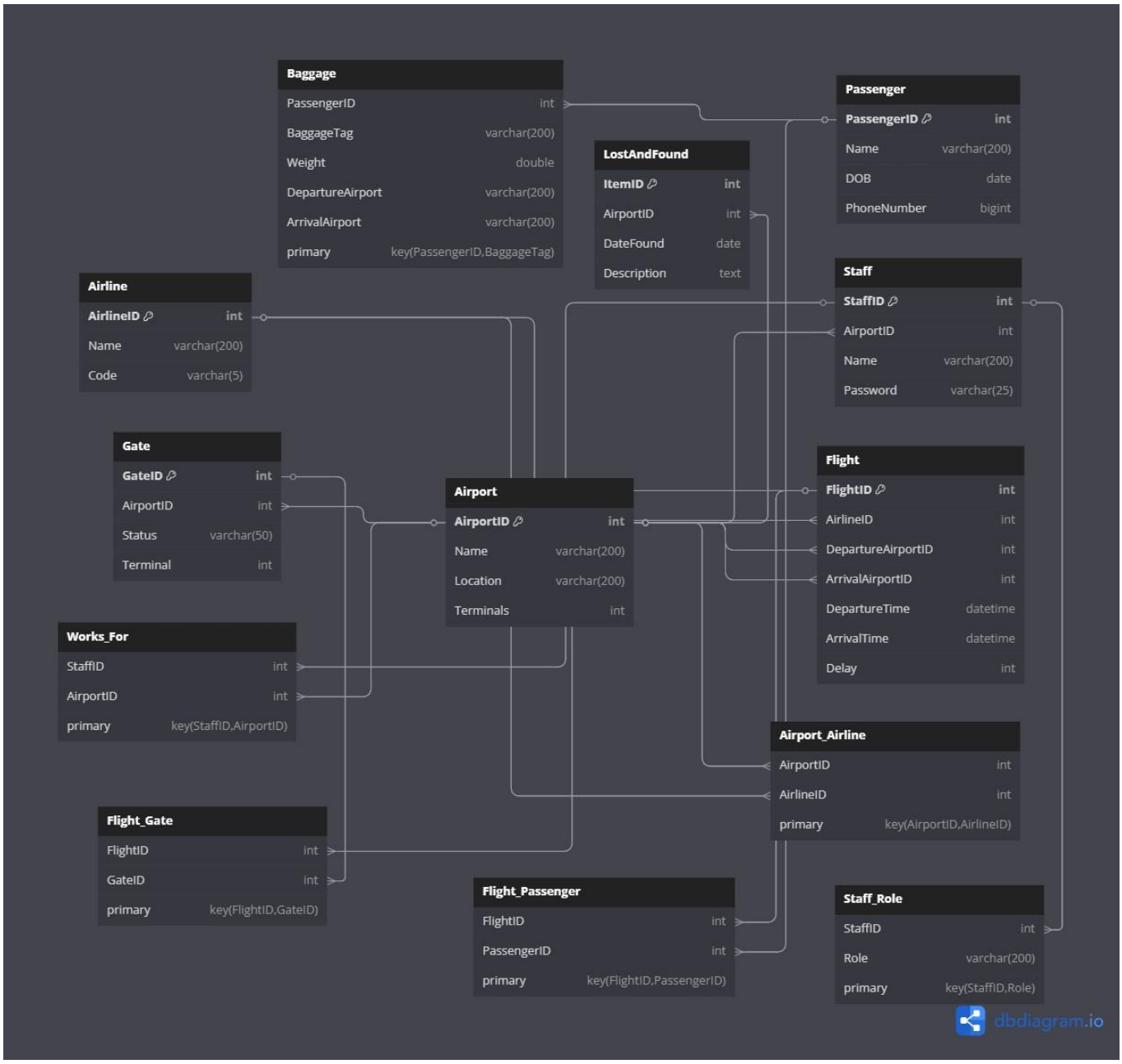

Figure 2.3: Finalized database Schema.

Chapter 3:
Methodology

3.1 Research Design

- Efficient relational database design using normalization techniques to avoid redundancy and ensure data consistency.
- Modular design of airport operations such as flight scheduling, passenger handling, baggage tracking, and gate assignments.
- User-friendly interface for staff login, registration, and operational management.
- Secure authentication and role-based access to ensure safe and authorized system usage.

3.2 Operational Flow

- Real-time data entry and updates from multiple departments (e.g., flight info, baggage updates).
- Staff access modules: Role-specific dashboards for airline operators, gate controllers, and lost & found personnel.
- Integration of passenger information and flight records for streamlined boarding and baggage processes.
- Gate and terminal assignments are managed dynamically based on availability and operational efficiency.

3.3 Implementation Detail

Frontend Development:

- Technologies used: HTML, CSS, and JavaScript for creating an interactive and responsive user interface.
- Features include login/registration forms, dashboards for flight and baggage management, and real-time data views.

Backend Development:

- Framework: Node.js, Express.js for server-side logic.
- Database Management System: MySQL for structured storage of flight data, passenger records, staff details, and gate information.
- Integration with Git and GitHub for collaborative development and version control.

3.4 System Workflow

- **Staff Login:** Staff members log in securely and access their designated operational module.
- **Data Input:** Depending on role, users can input or update information about flights, gates, passengers, or lost items.
- **Database Processing:** The backend processes data inputs and fetches/upgrades records as necessary.
- **Output Display:** Updated information is reflected on dashboards and accessible through role-based views.
- **Continuous Monitoring:** Modules like flight status and lost & found are dynamically updated in real-time.

3.5 Block Diagram of system architecture

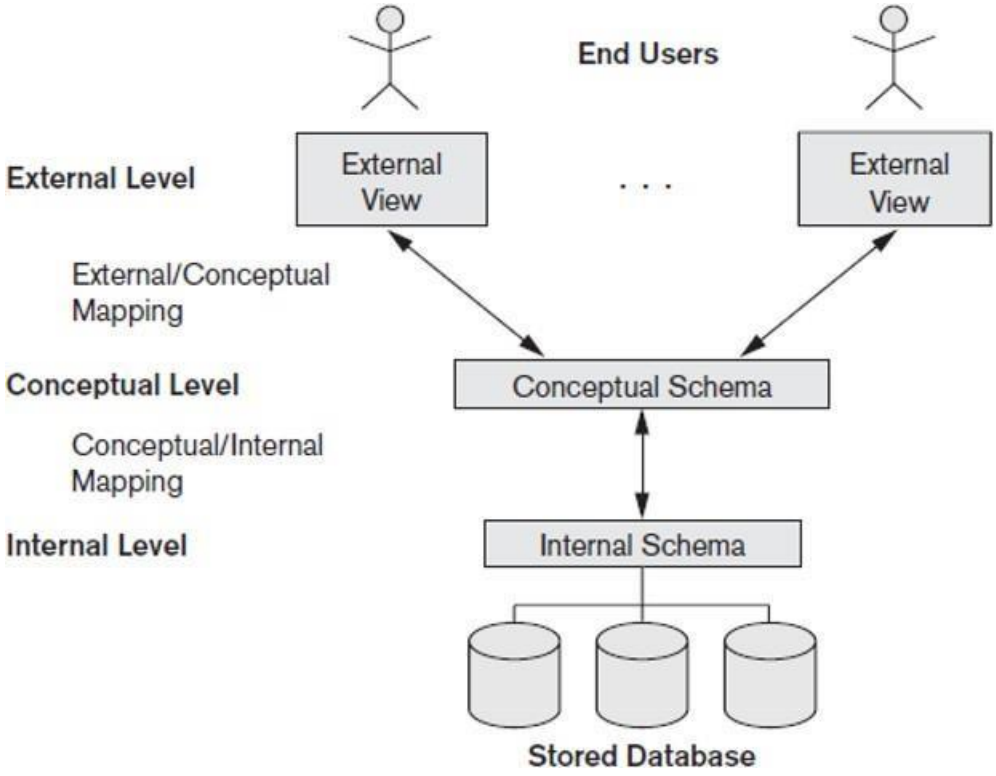

Fig 3.5 System Architecture

Chapter 4:
Results

4.1: Following screenshots describe the UI of the Airport Management System.

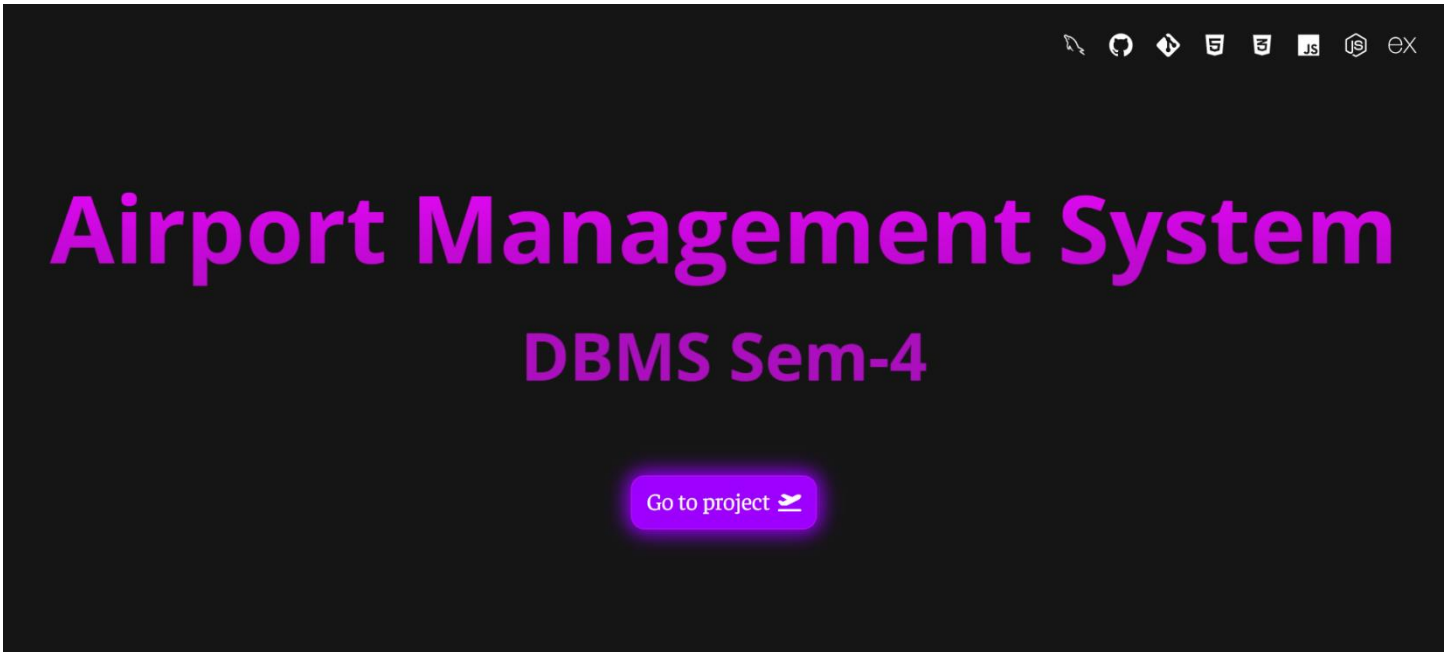

Figure 4.1: Home page

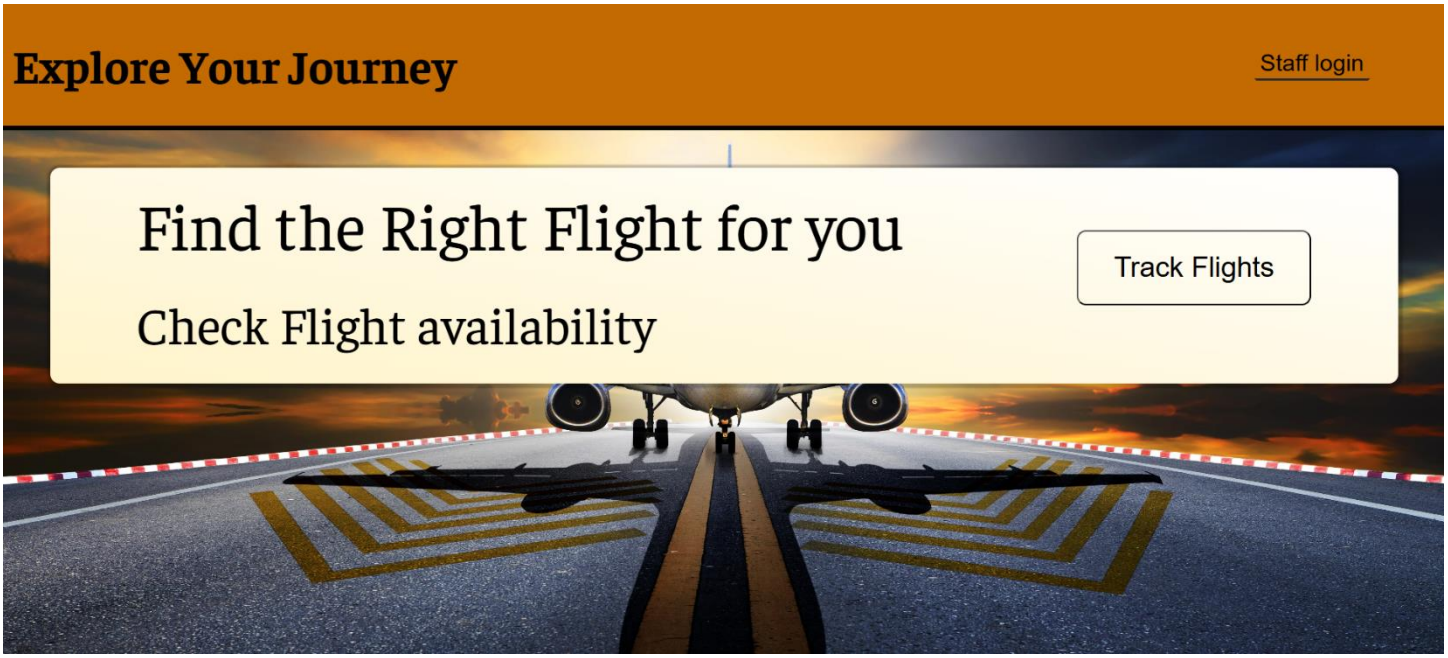

Figure 4.2: Flight details page

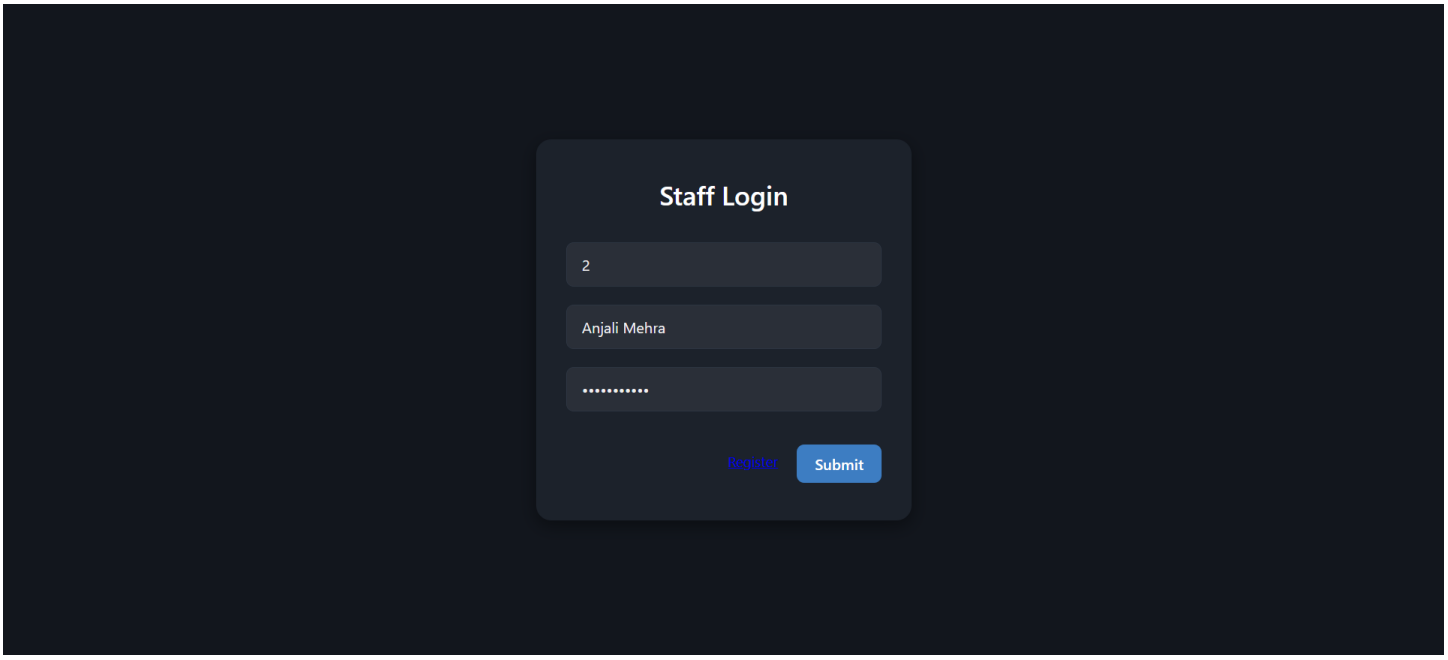

Figure 4.3: Staff login page

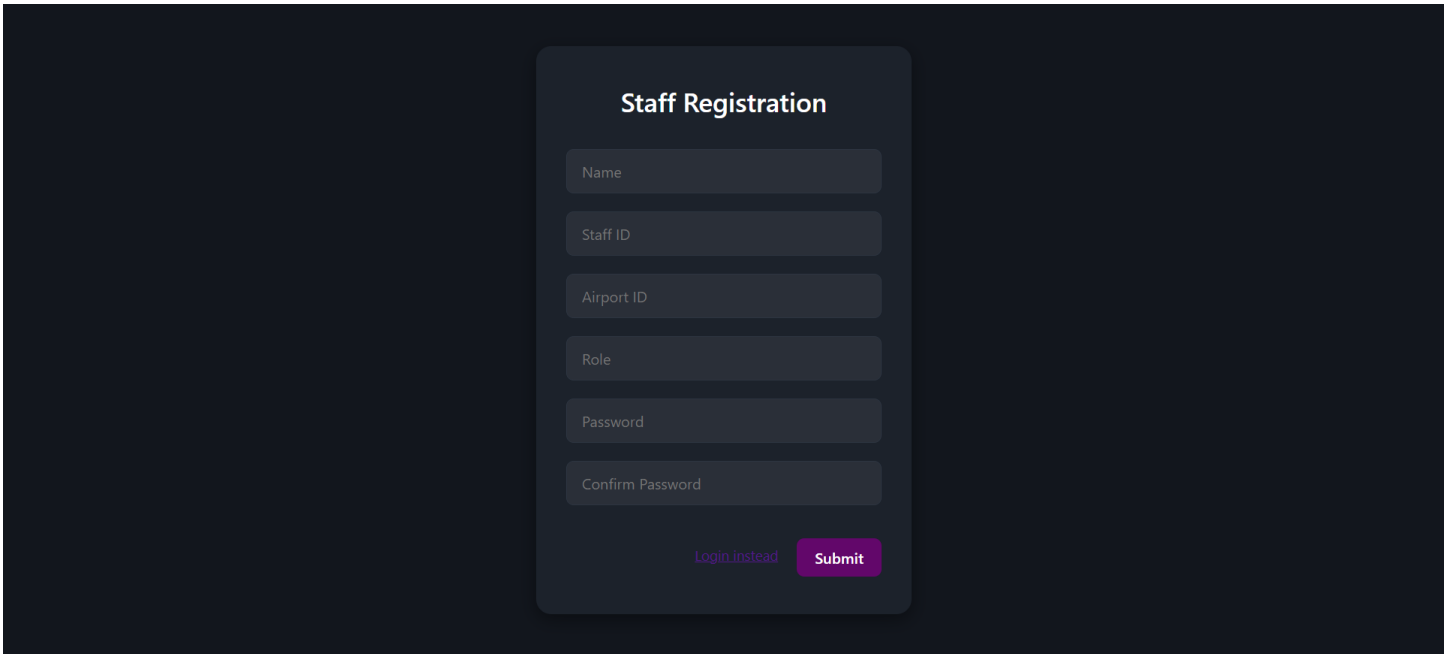

Figure 4.4: Staff Registration page

Figure 4.5: Post-Login page for staff

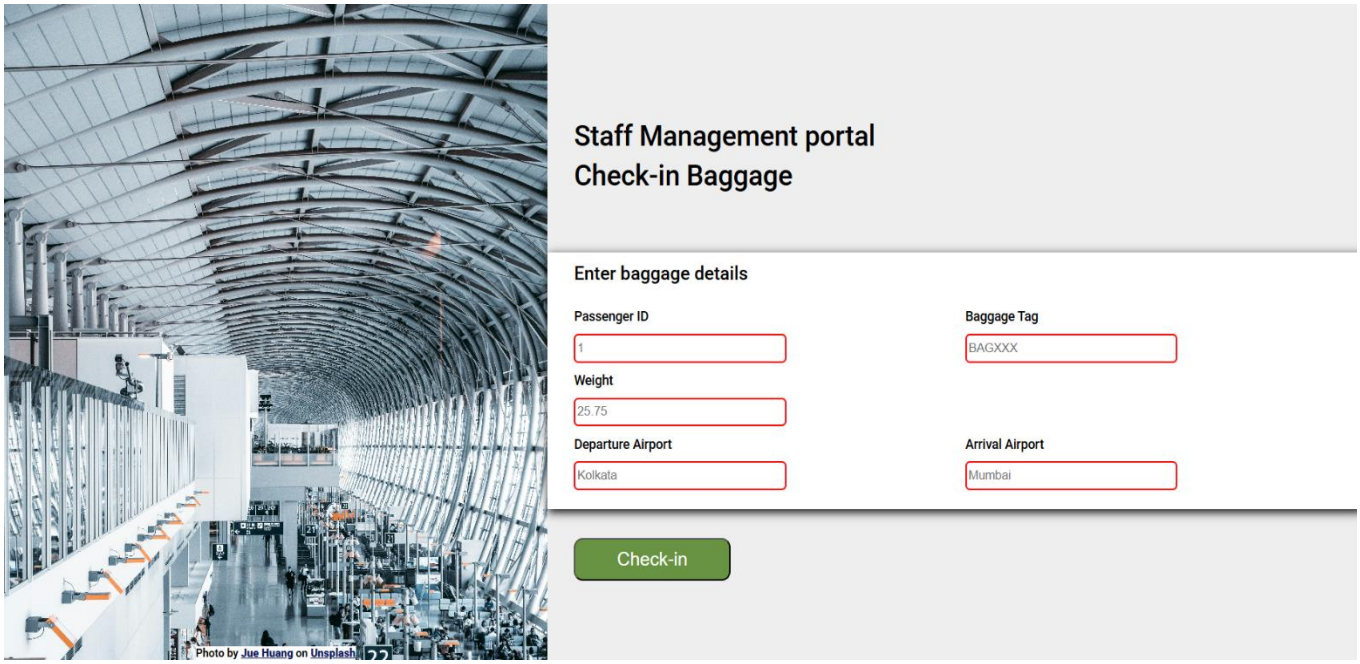

Figure 4.6: Check-in Baggage Page

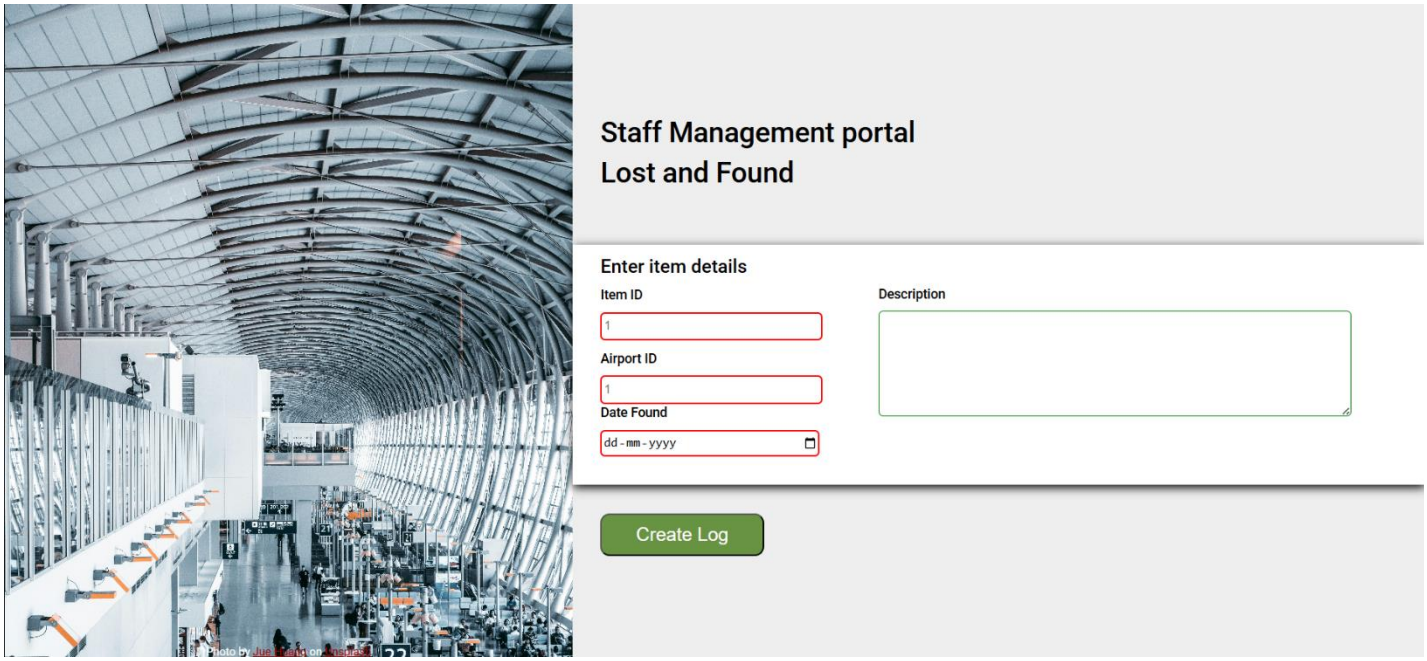

Figure 4.7: Lost and Found page

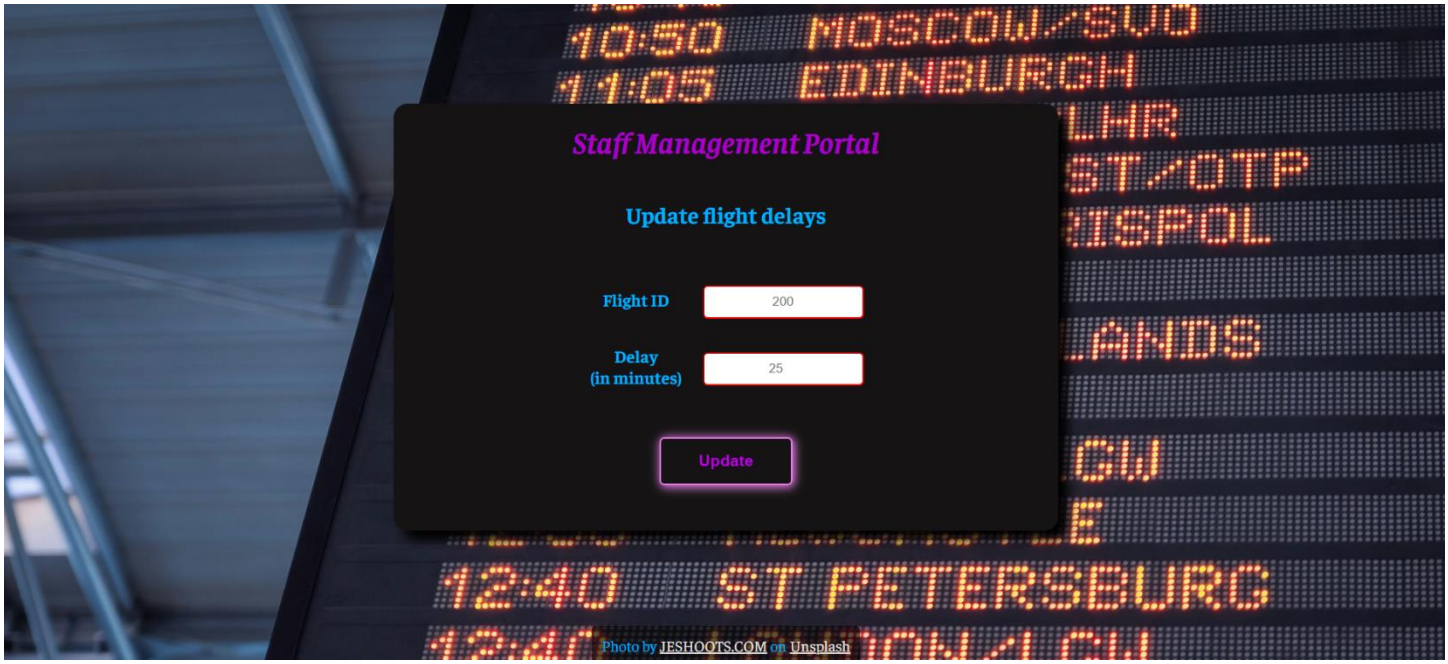

Figure 4.8: Flight Delay Update Page

Chapter 5:

Conclusion and Future Work

5.1: Conclusion

The Airport Management System (AMS) project successfully developed a centralized, web-based application that streamlines various airport operations including flight scheduling, passenger management, baggage tracking, and staff coordination. With an efficient relational database and a role-based secure access system, AMS enhances operational efficiency, reduces manual errors, and improves the overall travel experience for passengers.

5.2: Future Work

- **Integrate Real-Time Data Feeds**
  Incorporate real-time flight tracking and weather updates through API integrations to enhance dynamic scheduling and delay management.
- **Implement Mobile Compatibility**
  Develop a mobile version of the AMS platform for on-the-go access by staff and airport personnel.
- **Add Analytics and Reporting Dashboard**
  Introduce data visualization tools for generating performance reports, passenger trends, and operational metrics.

5.3: References

[1] Elmasri, R., & Navathe, S. B. (2017).
Fundamentals of Database Systems (7th Edition). Pearson.

[2] MySQL 8.0 Reference Manual.
https://dev.mysql.com/doc/refman/8.0/en/

[3] United Nations Sustainable Development Goals (SDG). *Goal 9: Industry, Innovation, and Infrastructure.*
https://sdgs.un.org/goals/goal9

[4] Directorate General Of Civil Aviation https://www.dgca.gov.in/digigov-portal/?page=4264/4206/sericename

[5] Airports Data
5.1 Bengaluru Airport : https://www.bengaluruairport.com/flights
5.2 GMR Airports Group : https://www.gmrgroup.in/airports/