

### Bachelor of Technology in

**COMPUTER SCIENCE AND ENGINEERING**

**22CS2403 – DATABASE MANAGEMENT SYSTEMS**

**MINI PROJECT REPORT**

On

### Airport Management System

Submitted By

## Hrushikesh Raghvendra (ENG22CS0321)

## Gagan Nagarjuna (ENG22CS0304)

## Dhruv Ravi (ENG22CS0292)

### UNDER THE SUPERVISION

**Dr. J Sebastian Nixon**

**Professor, CSE, DSU**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SCHOOL OF ENGINEERING DAYANANDA SAGAR UNIVERSITY**

### (2023-2024)



**School of Engineering**

**Department of Computer Science & Engineering**

Harihalli, Ramanagara - 562112

Karnataka, India

**CERTIFICATE**

This is to certify that the DBMS MINI PROJECT titled “**Airport Management System**” carried out by **Hrushikesh Raghvendra (ENG22CS0321), Gagan Nagarjuna (ENG22CS0304), Dhruv Ravi (ENG22CS0292)** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year 2023-2024.

|  |  |  |
| --- | --- | --- |
| **Dr. Sebastian Nixon** | **Dr. Girisha G S** | **Dr. Uday Kumar Reddy K R** |
| Professor  Dept. of CSE  School of Engineering Dayananda Sagar University | Chairman, CSE School of Engineering Dayananda Sagar University | Dean  School of Engineering Dayananda Sagar University |

**DATE:**

# ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this DBMS MINI PROJECT

First, we take this opportunity to express our sincere gratitude to the School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our bachelor’s degree in this institution.

We would like to thank **Dr. Uday Kumar Reddy K R, Dean**, **School of Engineering & Technology**, **Dayananda Sagar University For** his constant encouragement and expert advice. It is immense pleasure to express our sincere thanks to **Dr. Girisha G S, Chairman**, **Department of Computer Science, and Engineering**, **Dayananda Sagar University,** for providing the right academic guidance that made our task possible.

We would like to thank our teacher **Dr. Sebastian Nixon**, Professor, **Department of Computer Science and Engineering**, **Dayananda Sagar University**, for sparing his valuable time to extend help in every step of our DBMS MINI PROJECT, which paved the way for smooth progress and the fruitful culmination of the project.

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the DBMS MINI PROJECT.

# ABSTRACT

This project is a comprehensive web-based Airport Management System designed to streamline and automate the processes related to flight scheduling, arrivals, and passenger management. Built using Django and MySQL, it offers functionalities to add, update, and delete flight arrivals, departures, and passenger records. The system features a user-friendly interface with Bootstrap, ensuring easy navigation and interaction. By integrating secure database access and efficient data handling, the project aims to enhance operational efficiency, improve data accuracy, and provide real-time updates for airport staff and passengers.

V

## TABLE OF CONTENTS

**PAGE NO**

ABSTRACT

CHAPTER 1 INTRODUCTION <<1>>

CHAPTER 2 PROBLEM STATEMENT <<2>>

CHAPTER 3 PROJECT DESCRIPTION <<3>>

CHAPTER 4 DESIGN <<4-5>>

CHAPTER 5 METHODOLOGY <<6>>

CHAPTER 6 TESTING AND RESULT <<7-12>>

CHAPTER 7 CONCLUSION <<13>>

CHAPTER 8 REFERENCES <<14>>

# INTRODUCTION

This project is an advanced Airport Management System developed to enhance the efficiency and accuracy of managing airport operations. Leveraging the Django framework and MySQL for database management, the system facilitates seamless handling of flight schedules, arrivals, departures, and passenger information. The core functionalities include adding, updating, and deleting records for flights and passengers, ensuring that the airport operations are streamlined and well-coordinated. The user interface is designed with Bootstrap to provide a responsive and intuitive experience for users, making it easy to navigate through various features. Security measures are integrated to ensure safe database access and data integrity. This system not only reduces manual workload but also minimizes errors, providing real-time updates and ensuring that all stakeholders have access to the most current information. By implementing this system, airports can significantly improve their operational workflows and enhance the overall travel experience for passengers.

# PROBLEM STATEMENT

Develop a web application for managing airport operations. The system should allow users to view and manage flight schedules, arrivals, departures, passengers, and gates. Features include adding, updating, and deleting arrivals, departures, and passengers. The application must provide a user-friendly interface, real-time data updates, and ensure data integrity and security.

# PROJECT DESCRIPTION

The Airport Management System is a comprehensive web application designed to streamline and optimize airport operations. Here's an overview of its key features and functionalities:

1. Flight Schedule Management:
   * View and manage flight schedules, including departure and arrival times, destinations, and current status.
   * Real-time updates on flight statuses, gate assignments, and delays.
2. Arrivals and Departures Management:
   * Add, update, and delete arrival and departure records.
   * Track arrival and departure times, origin/destination airports, gate assignments, and current statuses (e.g., landed, boarding, departed).
3. Passenger Management:
   * Add, update, and delete passenger information, including name, age, gender, contact information, and seat assignments.
   * Manage passenger baggage details, such as weight and status.
4. User-friendly Interface:
   * Intuitive and easy-to-use interface for airport staff to navigate and perform various tasks efficiently.
   * Responsive design to ensure compatibility across different devices and screen sizes.

# DESIGN

The Airport Management System project follows a structured design with a clear separation of concerns, utilizing Django as the web framework and MySQL as the database management system. Here's an explanation of the project's design elements:

1. File Structure:
   * airport/: Root directory of the Django project.
   * airport/airport/: Contains project-specific settings, URLs, and WSGI configuration.
   * airport/security/: Django app for managing airport operations.
   * airport/db.sqlite3: SQLite database file used for development.
   * airport/manage.py: Django's command-line utility for administrative tasks.
2. Django App (security):
   * admin.py, apps.py, models.py, tests.py: Files for Django app configuration, database models, and testing.
   * database\_access.py: Module containing functions to interact with the MySQL database.
   * urls.py, views.py: URL routing and view functions for handling HTTP requests and rendering HTML templates.
   * migrations/: Directory for storing database migration files generated by Django.
   * templates/security/: Contains HTML templates for rendering user interfaces.
3. MySQL Database:
   * The MySQL server stores data related to airport operations, including flight schedules, arrivals, departures, passengers, gates, etc.
   * The database\_access.py module provides functions to perform CRUD operations on the MySQL database using the mysql.connector library.
   * Functions such as insert\_arrivals, insert\_passenger, get\_flight\_schedule, etc., interact directly with the MySQL server to insert, retrieve, update, or delete data.
4. Views and Templates:
   * Views in views.py fetch data from the database using functions from database\_access.py and pass it to HTML templates for rendering.
   * HTML templates in templates/security/ contain the user interface markup, including forms, tables, buttons, etc.
   * Templates are extended and customized as needed for specific functionalities like adding arrivals, updating passengers, etc.
5. URL Routing:
   * URLs are mapped to view functions using the urls.py file in the Django app.
   * Each URL pattern corresponds to a specific view function responsible for handling the corresponding HTTP request.
6. Connections to MySQL Server:
   * Database connection parameters (host, user, password, database) are defined in database\_access.py.
   * The mysql.connector library is used to establish connections to the MySQL server and execute SQL queries.
   * Functions in database\_access.py manage database connections, execute SQL queries, fetch data, and handle transactions.

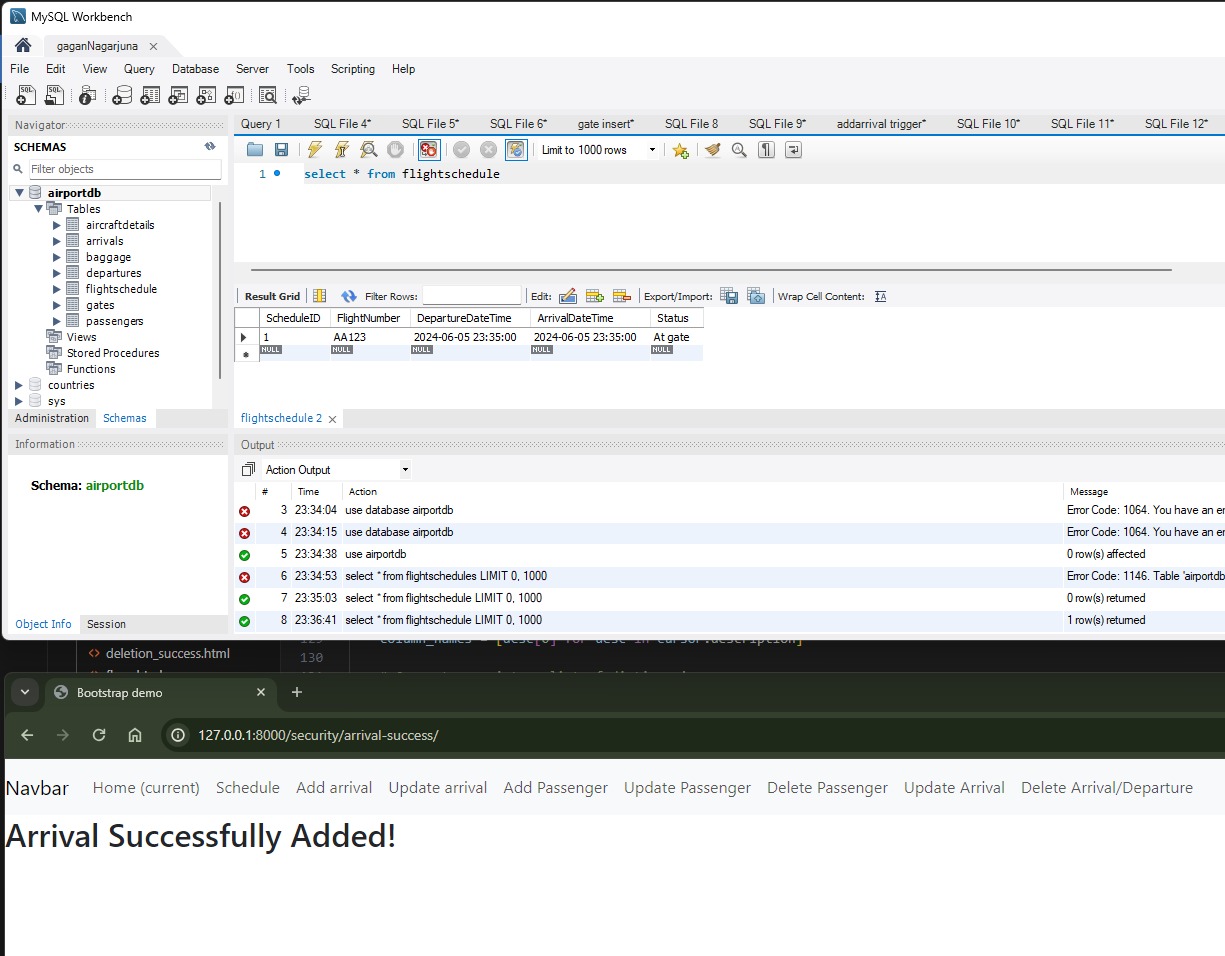
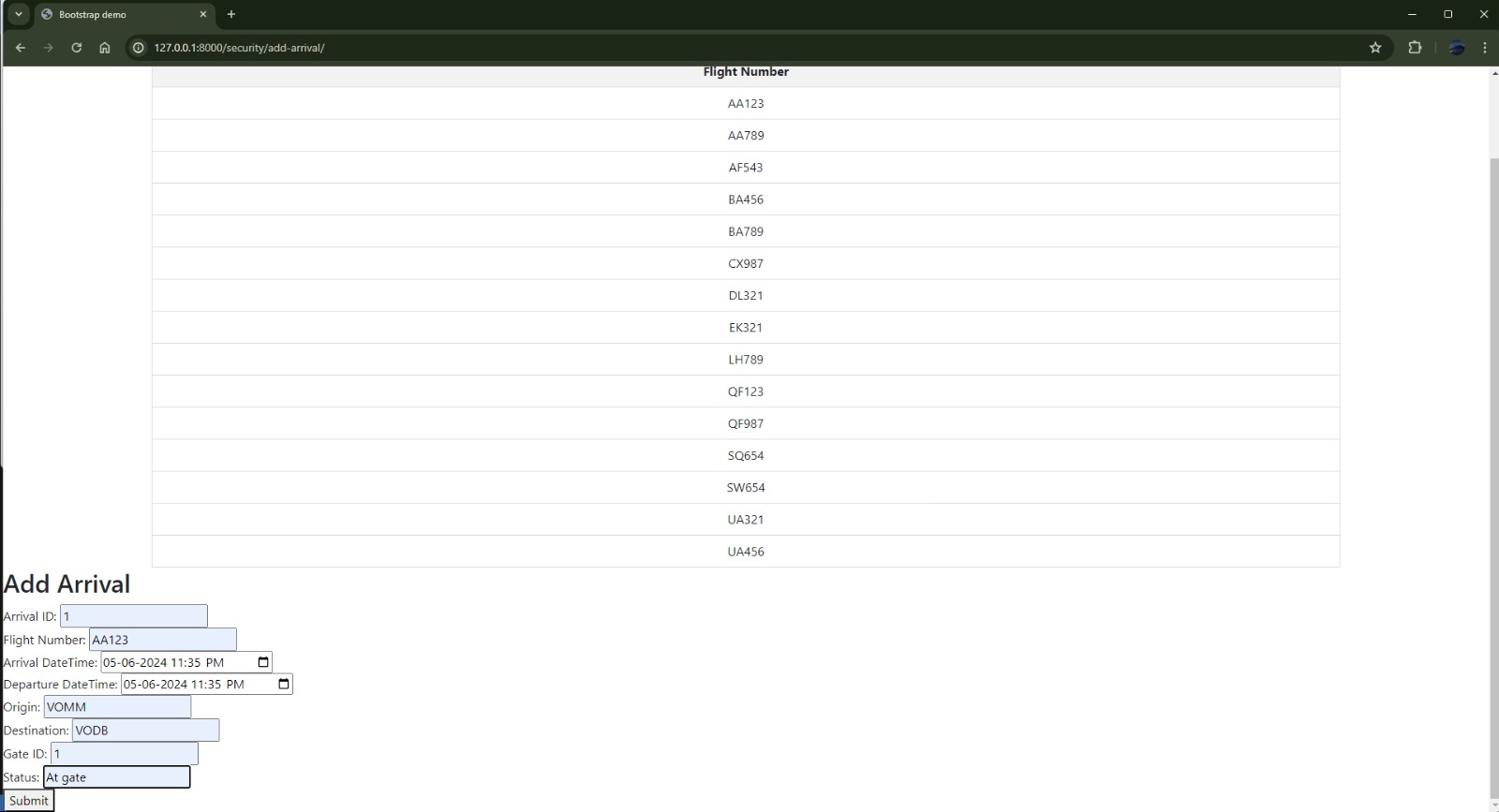
Overall, the project's design promotes modularity, separation of concerns, and scalability, making it easier to maintain, extend, and enhance the Airport Management System.

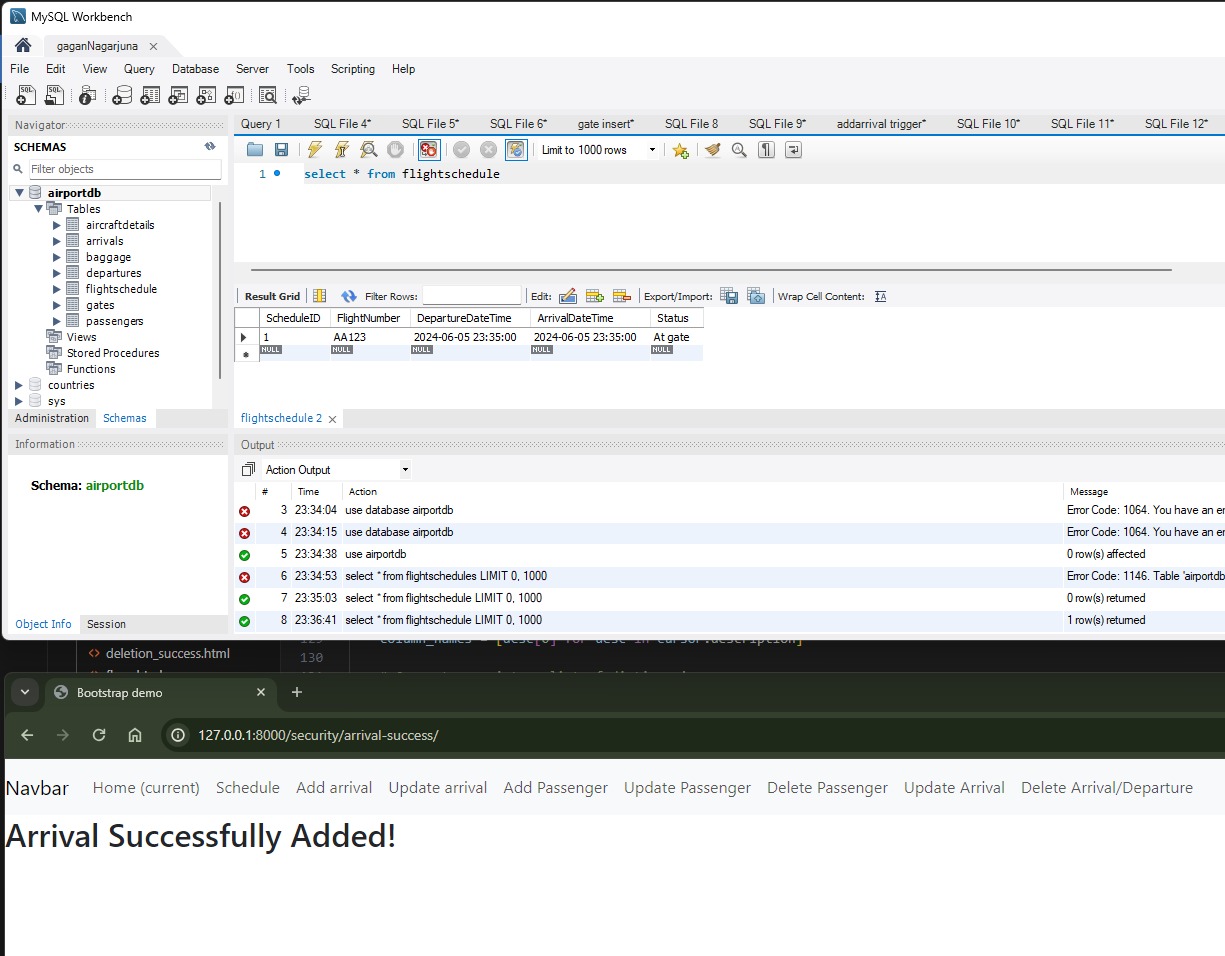
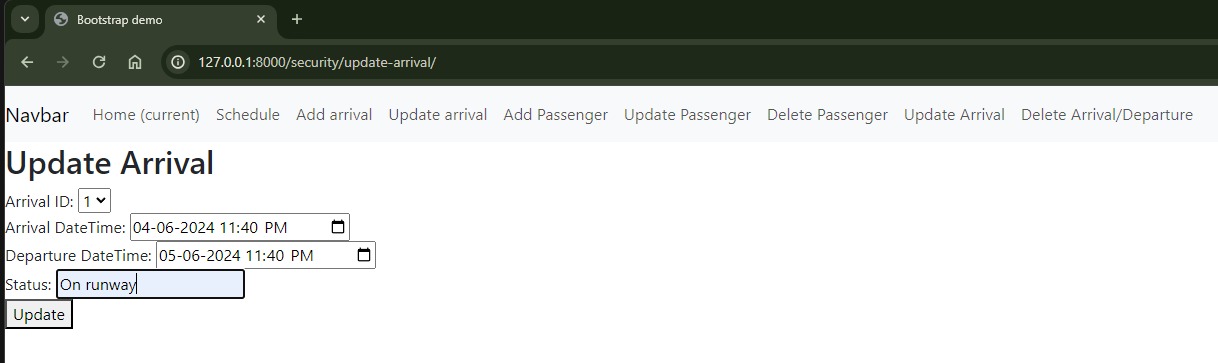
# METHODOLOGY

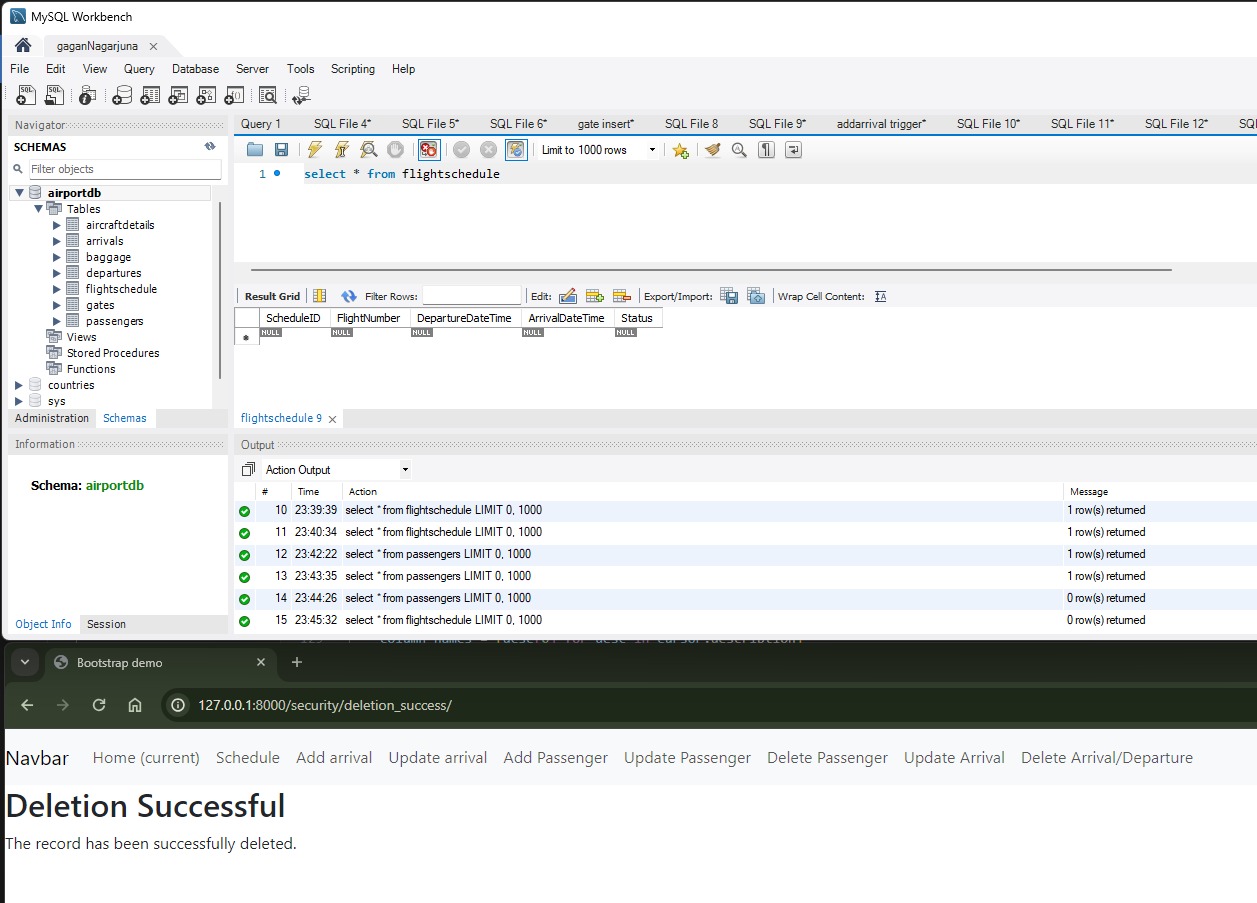
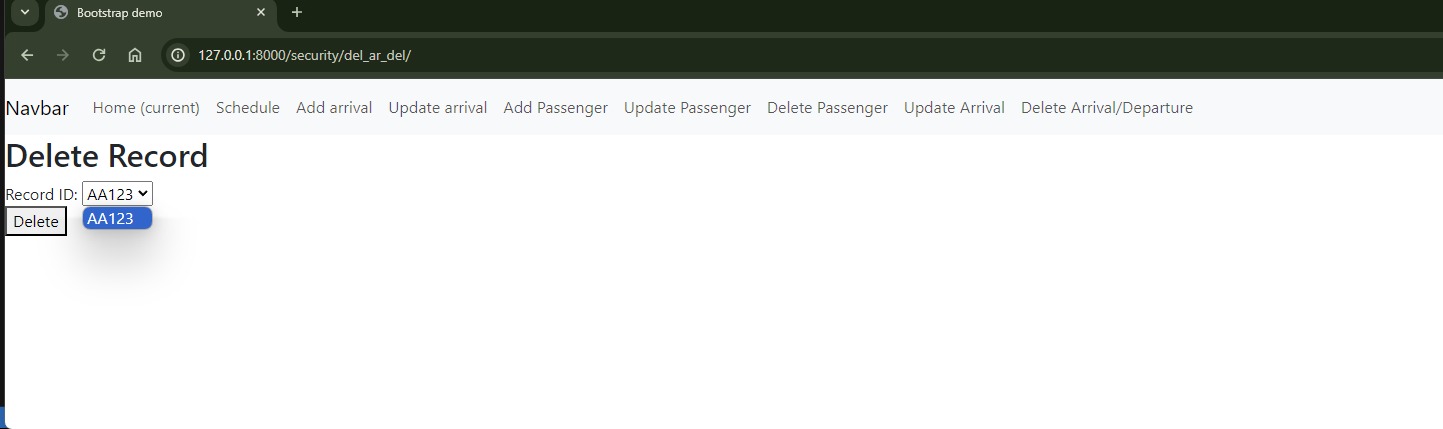
The Airport Management System project follows an iterative and incremental development methodology, leveraging Django as the web framework and MySQL as the database management system. Here's an explanation of the project methodology:

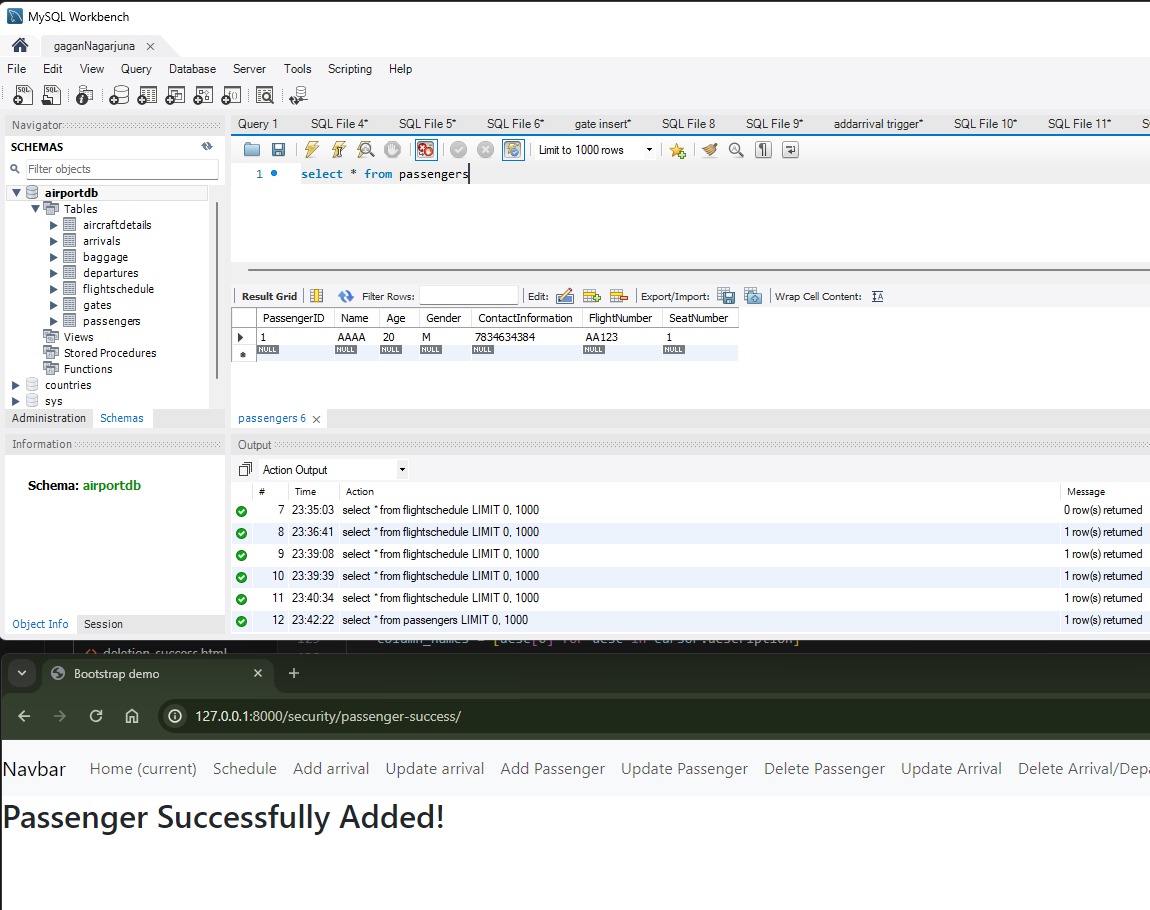
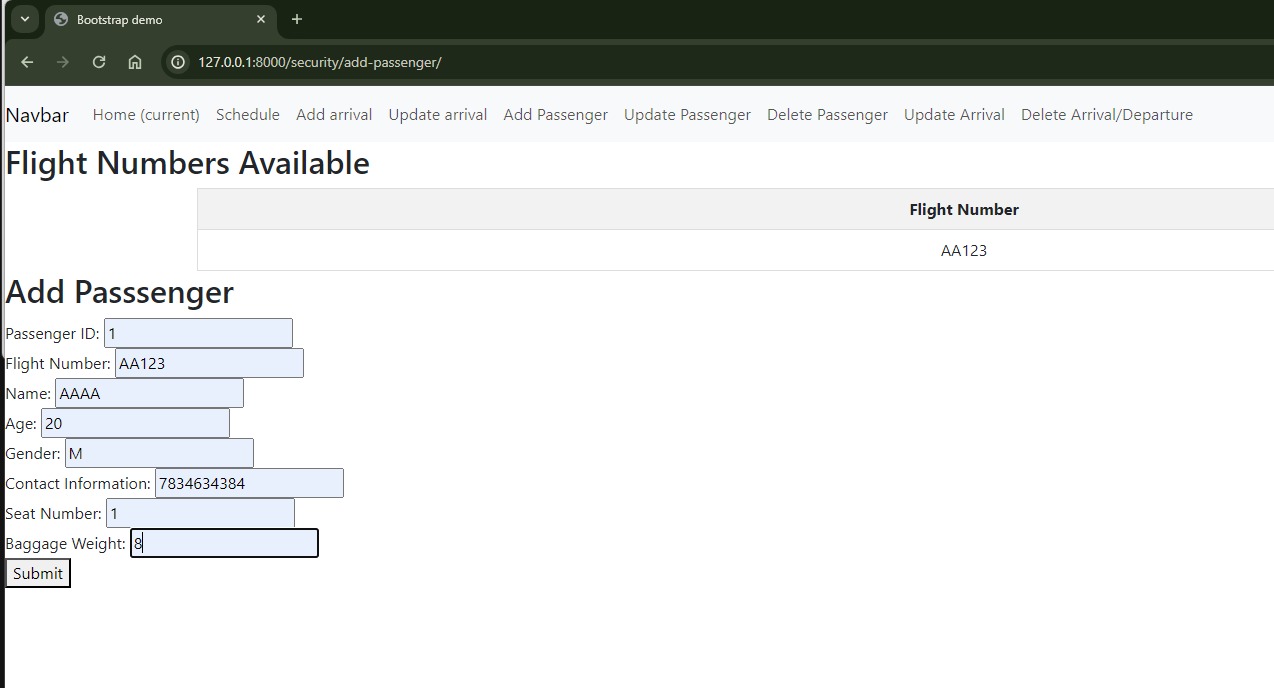
1. Requirement Analysis:
   * Identified stakeholders' needs and requirements for managing airport operations, including flight schedules, arrivals, departures, passengers, gates, etc.
2. Design:
   * Designed the system architecture, database schema, and user interface based on the identified requirements.
   * Developed a clear file structure for the Django project, separating concerns into apps, models, views, templates, and static files.
3. Implementation:
   * Developed the system incrementally, starting with basic functionalities and gradually adding more features.
   * Utilized Django's built-in features for URL routing, view functions, form handling, and template rendering.
   * Implemented database access functions in Python using the mysql.connector library to interact with the MySQL server.

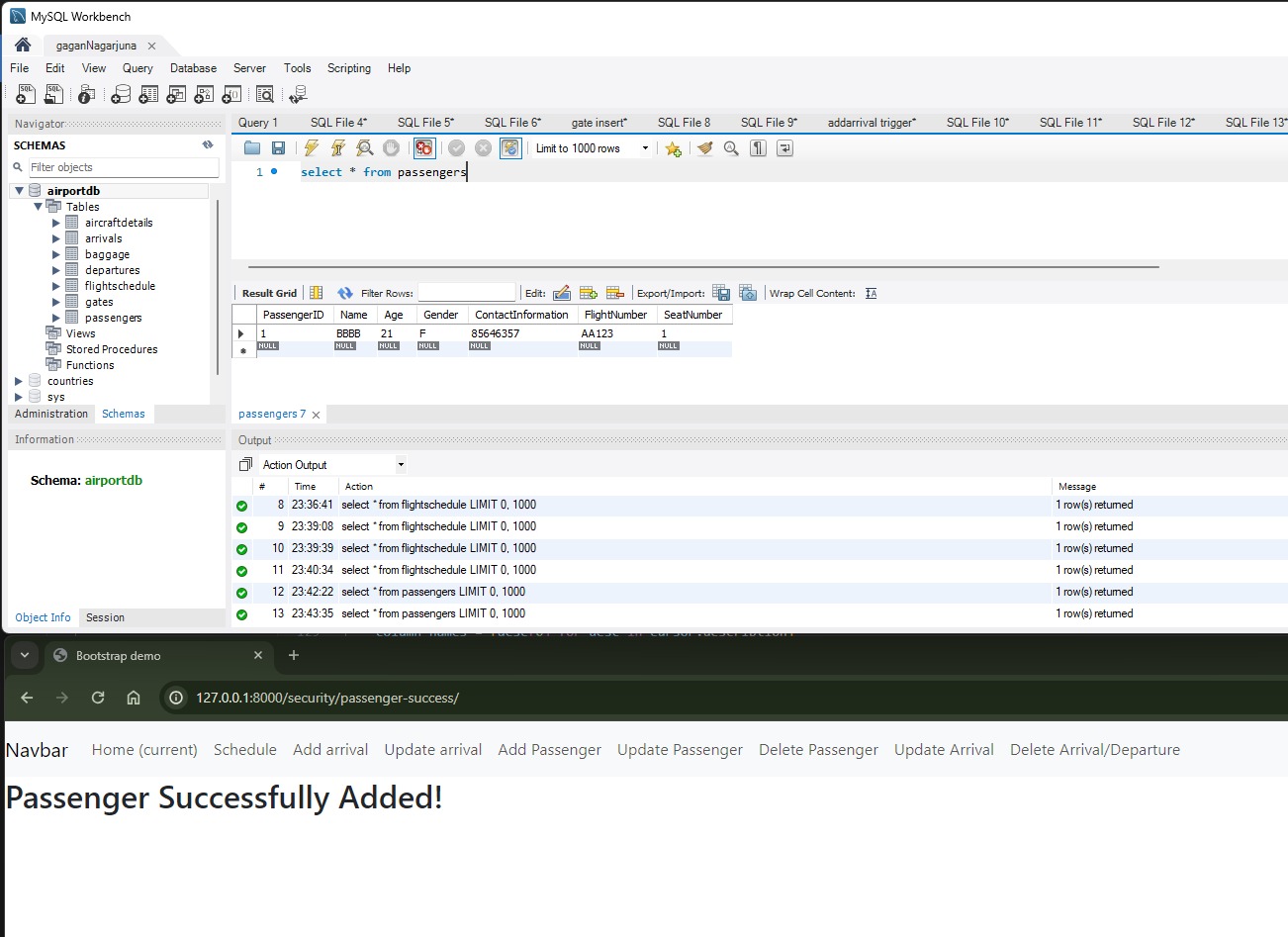
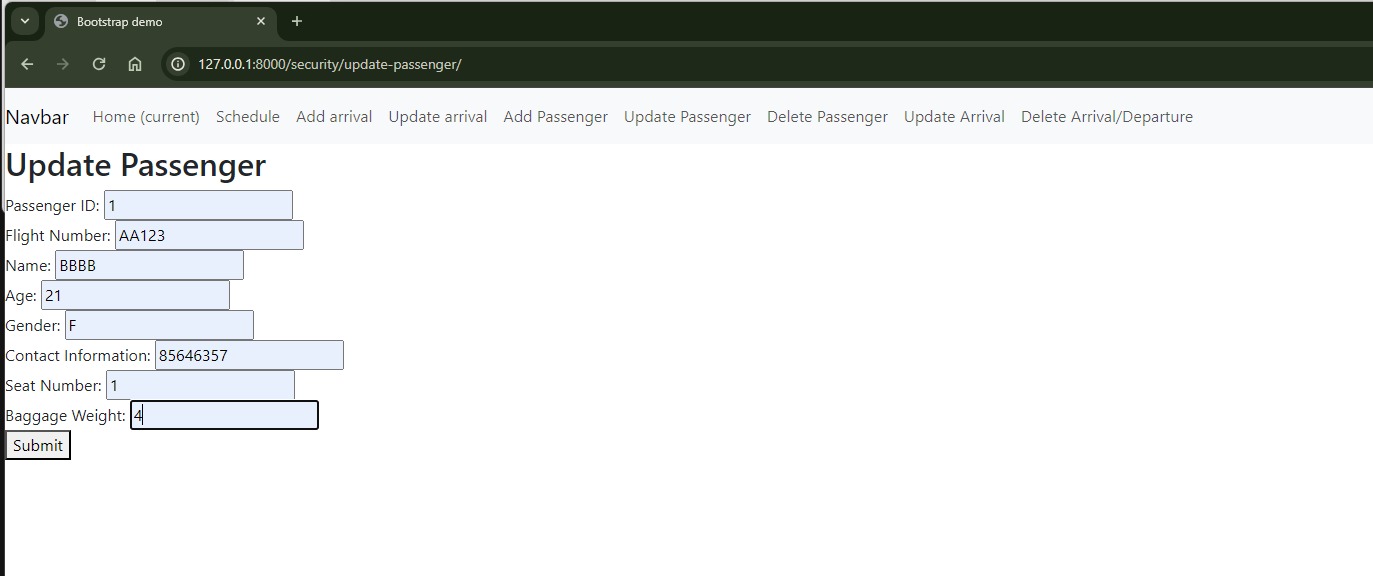
# TESTING AND IMPLEMENTATION

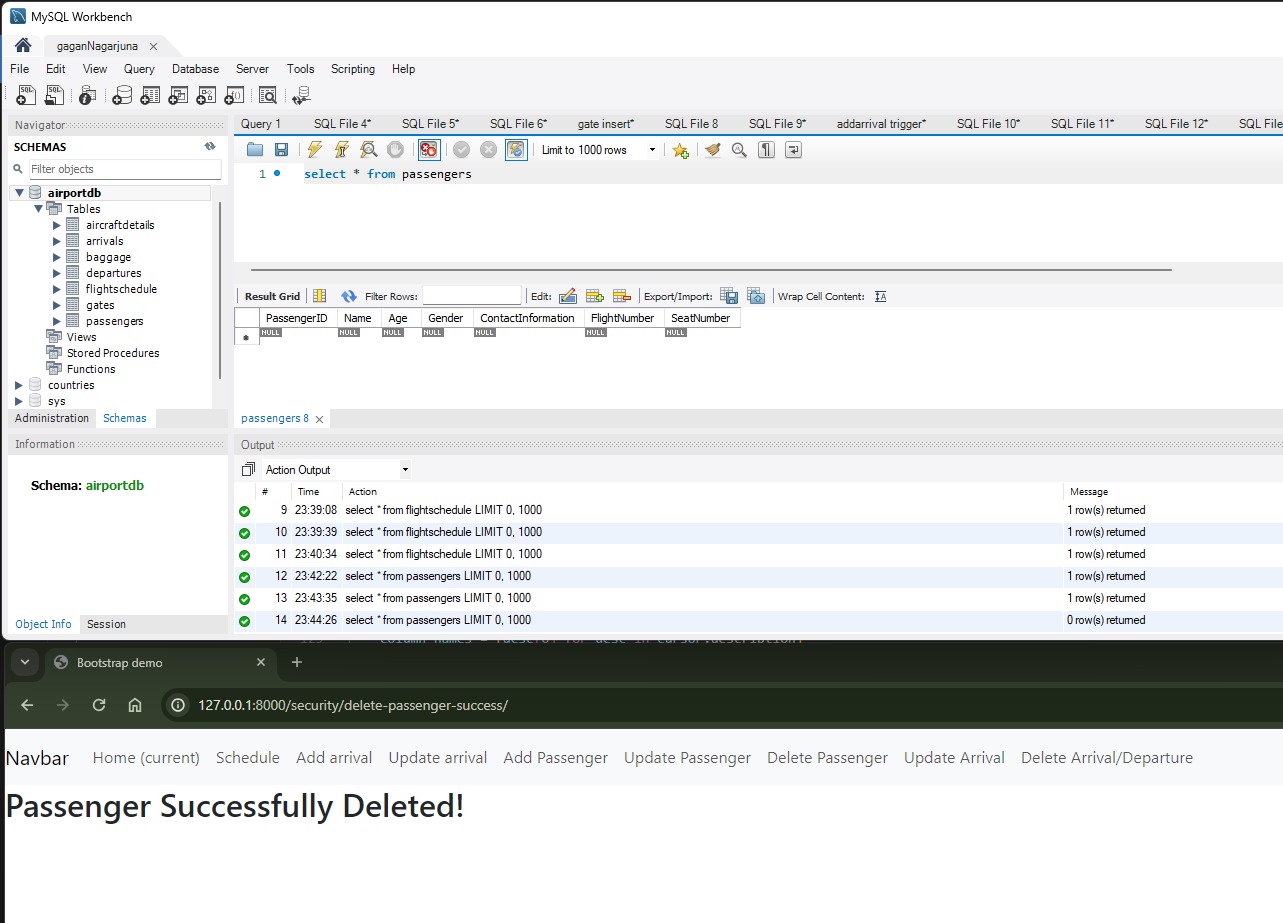
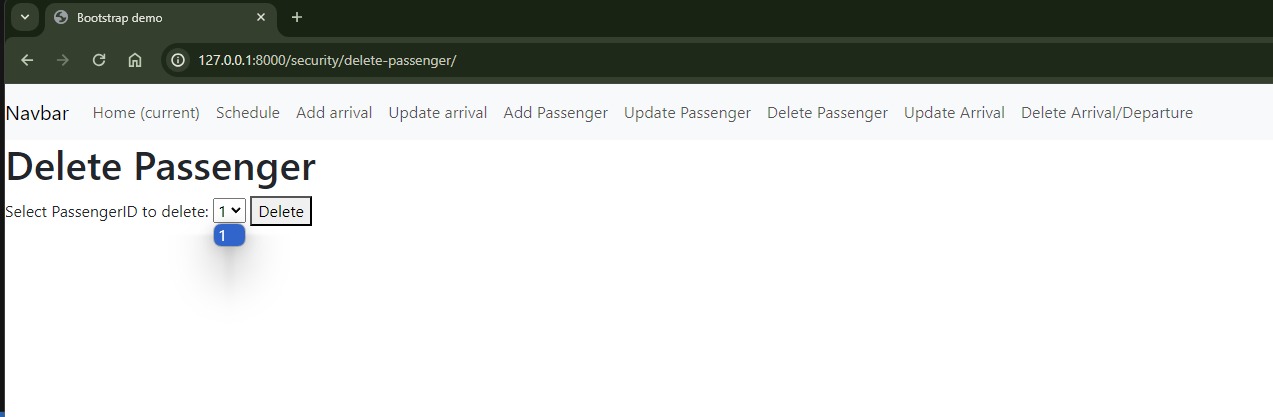
Add arrival and display flightschedule

Update arrival

Delete Flight

Add passenger

Update passenger

Delete passenger

# CONCLUSION

In conclusion, the Airport Management System project has successfully addressed the complex requirements of managing airport operations efficiently. By leveraging Django's robust web framework and MySQL as the database management system, we have developed a scalable and user-friendly solution. The system allows airport staff to manage flight schedules, arrivals, departures, passengers, gates, and more, streamlining operations and improving overall efficiency.

Through iterative development and continuous feedback, we have refined the system to meet the evolving needs of stakeholders and end-users. The clear file structure and modular design of the project facilitate ease of maintenance and future enhancements. As the project concludes, we remain committed to providing ongoing support and maintenance, ensuring that the Airport Management System remains reliable, secure, and adaptable to future requirements.

Overall, the project represents a significant step forward in modernizing airport management processes, offering a comprehensive solution that enhances productivity and customer satisfaction. We look forward to further advancements and refinements in the field of airport management, building upon the foundation laid by this project.

# REFERENCES

* “<https://www.w3schools.com/django/>” Django tutorial on w3schools website
* “<https://dev.mysql.com/doc/connector-python/en/>” Mysql connector documentation
* “<https://getbootstrap.com/>” Bootstrap premade styles for webpages