

**DELHI PUBLIC SCHOOL DEHRADUN  
SESSION 2023-24**

**COMPUTER SCIENCE**

**PYTHON PROJECT**

**Submitted by :-  
  
(i) Rakshit Tyagi – XII**

**(ii) Tejas Kumawat – XII**



**DEPARTMENT OF COMPUTER SCIENCE  
DELHI PUBLIC SCHOOL, DEHRADUN   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***Certificate***

This is to certify that the original and genuine content of this project entitled “**Flight Management System**” has been satisfactorily executed and prepared by **Tejas Kumawat** and **Rakshit Tyagi** the students of class **XII** , under the direct supervision of **Dr. Pooja Vashista** for the academic session 2023-24. This bona fide work of theirs have been submitted to **Delhi Public School, Dehradun** for consideration in partial fulfilment of **Computer Science** **Practical Examination** conducted by AISSCE, New Delhi

The project report is up to the expectation and a result of their efforts and endeavour.

**Internal Examiner External Examiner**

**Principal**

**ACKNOWLEDGEMENT**

Project is like a bridge between theoretical and practical learning and now on the accomplishment of this project successfully, we would like to thank all the people who have been concerned with it and bestowed upon us their blessings and the heart pledged support.

Primarily, we would like to thank God, who always guided us to work on the right path of life. Without his grace this project could hot become a reality. Next to him are our parents, to whom we are greatly indebted for their love and encouragement to this stage.

We are feeling highly obliged in taking the opportunity to express our special thanks and gratitude to our Respected Principal Sir, Mr. B. K. Singh and Vice Principal Ma'am, Mrs. Sujata Singh, for their valuable guidance, keen interest and encouragement.

We express our deep and sincere gratitude to my subject teacher **Dr. Pooja Vashista**, whose guidance, encouragement, suggestions and constructive criticism have contributed immensely to the evolution of our ideas on the project.

At last, but not the least, we are thankful to all our teachers and friends who have been always helped us and encouraged us throughout the year. No more we have the words to express our thanks, but our heart is still full of favours received from every person.

|  |  |  |
| --- | --- | --- |
| **Table Of Contents** | | |
| **SNO.** | **Description** | **Page No.** |
| **1** | Certificate | 1 |
| **2** | Acknowledgement | 2 |
| **3** | Introduction | 4 |
| **4** | System Requirements | 7 |
| **5** | Source code | 8 |
| **6** | Output | 10 |
| **7** | BackEnd Output | 12 |
| **8** | Bibliography |  |

**Introduction to the Flight Management System**

In an era where air travel has become an integral part of our lives, the need for a reliable and user-friendly Flight Management System has never been greater. Whether you're an airline striving to streamline your operations or a traveller seeking a hassle-free booking experience, our Flight Management System is designed to meet your needs. This project is a comprehensive and efficient solution that empowers both airlines and passengers alike, offering the ability to manage passengers, display flight information, explore available services, and simplify the flight booking process.

The Flight Management System represents a dynamic blend of functionality and convenience, bringing together a range of features to enhance the experience of both travellers and aviation service providers. Powered by SQL and Python, this project is designed to offer a seamless and robust platform for storing, retrieving, and managing data while providing passengers with an effortless way to explore flights and available services.

**Key Features of the Flight Management System:**

**Passenger Management**:

Our system allows airlines to efficiently add, update, and manage passenger information, ensuring a smooth and organized check-in process.

**Flight Information**:

Passengers can easily access a list of available flights, including essential details such as departure times, destinations, and aircraft type, making it convenient to plan their journeys.

**Food Menu**:

Travelers can view the onboard food menu, pre-order meals, and make their in-flight dining experience as enjoyable as possible.

**Flight Booking:**

Passengers can effortlessly browse available flights, select their preferences, and complete the booking process. The system simplifies the entire booking journey, from flight selection to payment.

**Availability Checks:**

Passengers can check the availability of flights, seats, and services, helping them make informed decisions about their travel plans.

**Reporting and Analytics:**

Airlines can gain insights into passenger data, flight bookings, and service preferences, enabling them to make data-driven decisions and optimize their operations.

**Conclusion:**

The Flight Management System is not just a project; it's your ticket to an enhanced flight booking experience. By harnessing the power of SQL and Python, we aim to address the intricate requirements of the aviation industry, ensuring precision, efficiency, and security. Whether you're an airline looking to streamline your operations or a passenger seeking a straightforward booking process, our project offers a scalable and adaptable solution.

**SYSTEM REQUIRMENTS**

**Recommended System Requirements**

* Processor- Intel® Core™ i3-4300M at 2.60 GHz
* Disk Space- 2 to 4GB
* Operating System- Windows 10, UBUNTU, MACOS
* Python Version: 3.10 64-bit

**Minimum System Requirements**

* Processor- Intel Atom® Processor
* Disk Space- 1 GB
* Operating System- Windows 7, MACOS, UBUNTU
* Python Versions: 2.7.X , 3.6.X

**Prerequisites before installing MySQL Connector Python :-**

• Required root or administrative privileges to perform the installation

• Python and MySQL must be installed on the device

• MySQL Connector Python needs to be in the system’s PATH for code to run else it fails

• If Python doesn’t exist in system’s PATH, please manually add the directory containing python.exe

**Source code**

**import mysql.connector as mc**

**mdb=mc.connect(host='localhost',user='root',password='1234',charset='utf8')**

**mcur=mdb.cursor()**

**mcur.execute("CREATE DATABASE IF NOT EXISTS fms")**

**mcur.execute("USE fms")**

**if mdb.is\_connected():**

**print("Connected")**

**else:**

**print("Not Connected")**

**def createtable():**

**q1 ="create table if not exists passengers (Name varchar(20),Age int, Seat\_no int PRIMARY KEY,Destination varchar(20),No\_bag char(2),Wt\_lug char(2))"**

**mcur.execute(q1)**

**q2="create table if not exists flights (Starting\_from varchar(20),Going\_to varchar(20),Takeoffdate datetime,Land\_date datetime,Type varchar(13))"**

**mcur.execute(q2)**

**q3="create table if not exists Cake (Srno int(2) PRIMARY KEY, Type varchar(15), Price varchar(3))"**

**mcur.execute(q3)**

**q4="create table if not exists Mocktail (Srno int(2) PRIMARY KEY, Type varchar(15), Price varchar(3))"**

**mcur.execute(q4)**

**q5="create table if not exists Coffee (Srno int(2) PRIMARY KEY, Type varchar(15), Price varchar(3))"**

**mcur.execute(q5)**

**q6="create table if not exists Tea (Srno int(2) PRIMARY KEY, Type varchar(15), Price varchar(3))"**

**mcur.execute(q6)**

**q7="create table if not exists Meal (Srno int(2) PRIMARY KEY, Type varchar(15), Price varchar(3))"**

**mcur.execute(q7)**

**q8="create table if not exists Snacks (Srno int(2) PRIMARY KEY, Type varchar(15), Price varchar(3))"**

**mcur.execute(q8)**

**q13="create table if not exists forder (Srno int(2) PRIMARY KEY, Name varchar(20), Item varchar(15))"**

**mcur.execute(q13)**

**print("Tables created sucessfully")**

**def addpassenger():**

**name=input("Enter passenger's name :: :")**

**age=input("Enter age :: ")**

**seat\_no=int(input("Enter alloted seat number :: "))**

**destination=input("Enter destination :: ")**

**wtallowed=5**

**nobag=int(input("Enter number of bags :: "))**

**print("Allowed Luggage Weight = 5kg ")**

**wtbag=int(input("Enter weight of luggage (kg) :: "))**

**if wtbag<wtallowed:**

**print("Passed")**

**mcur.execute("INSERT INTO passengers (Name, Age, Seat\_no, Destination, No\_bag, Wt\_lug) VALUES (%s, %s, %s, %s, %s, %s)", (name, age, seat\_no, destination, nobag, wtbag))**

**mdb.commit()**

**print("executed")**

**else:**

**print("Your luggage is over weight")**

**def food():**

**print("l .Cake")**

**print("2.Mocktail")**

**print("3.Coffee")**

**print("4.Tea")**

**print("5.Meal")**

**print("6.Snacks")**

**choice=int(input("Enter your choice :: "))**

**print(choice)**

**if choice==1:**

**q10="Select \* from cake"**

**mcur.execute(q10)**

**dl = mcur.fetchall()**

**print(dl)**

**elif choice==2:**

**q11 ="Select \* from Mocktail"**

**mcur.execute(q11)**

**d2=mcur.fetchall()**

**print(d2)**

**elif choice==3:**

**q12="Select \* from Coffee"**

**mcur.execute(q12)**

**d3=mcur.fetchall()**

**print(d3)**

**elif choice==4:**

**q15 ="Select \* from Tea"**

**mcur.execute(q15)**

**d4=mcur.fetchall()**

**print(d4)**

**elif choice==5:**

**q14="Select \* from Meal"**

**mcur.execute(q14)**

**d5=mcur.fetchall()**

**print(d5)**

**elif choice==3:**

**q12="Select \* from Snacks"**

**mcur.execute(q12)**

**d6=mcur.fetchall()**

**print(d6)**

**else:**

**print("You entered invalid choice")**

**def avail\_flights():**

**q16="select Starting\_from,Going\_to from flights"**

**mcur.execute(q16)**

**d7=mcur.fetchall()**

**print("From ---> To")**

**print(d7)**

**def display\_passengers():**

**q17="select \* from passengers"**

**mcur.execute(q17)**

**d8=mcur.fetchall()**

**print(d8)**

**def passengers\_goingto():**

**q18=" select Name, Destination from passengers order by Destination"**

**mcur.execute(q18)**

**d9=mcur.fetchall()**

**print(d9)**

**createtable()**

**while True:**

**print("l .add passenger")**

**print("2.flights available")**

**print("3.To display all passengers")**

**print("4.passengers going to common place")**

**print("5.View Food Menu")**

**print("")**

**ch=int(input("Enter your choice :: "))**

**if ch==1 :**

**addpassenger()**

**elif ch==2:**

**avail\_flights()**

**elif ch==3:**

**display\_passengers()**

**elif ch==4:**

**passengers\_goingto()**

**elif ch==5:**

**food()**

**c=input("Do you want to exit (y/n)::")**

**if c=='y':**

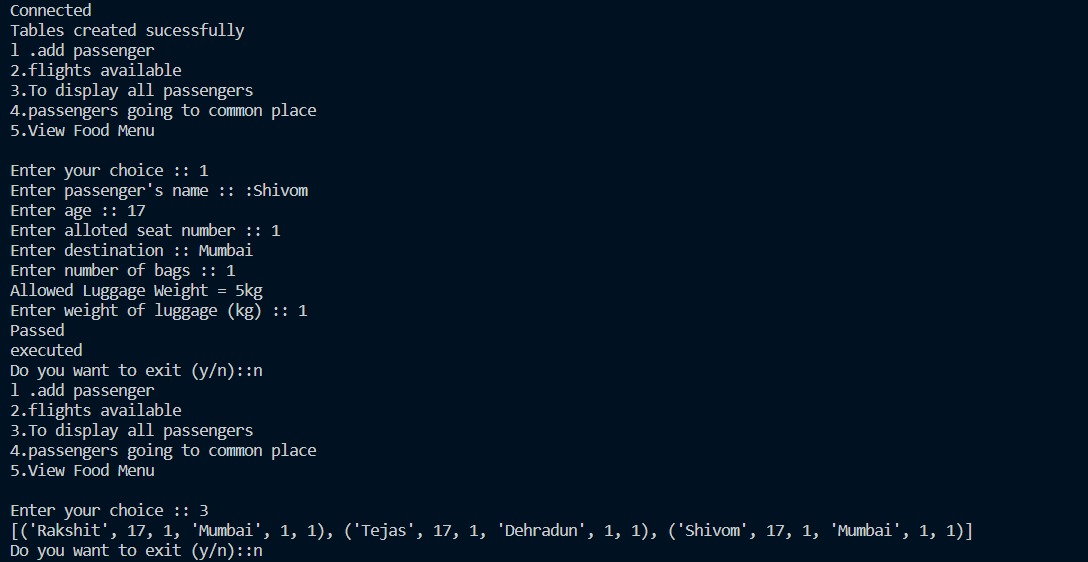
**print("Thanks for visting..")**

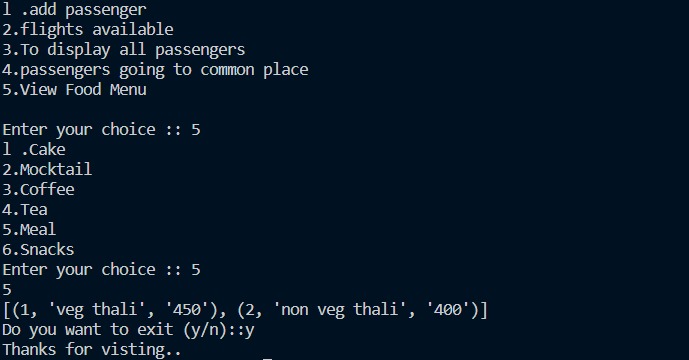
**break**

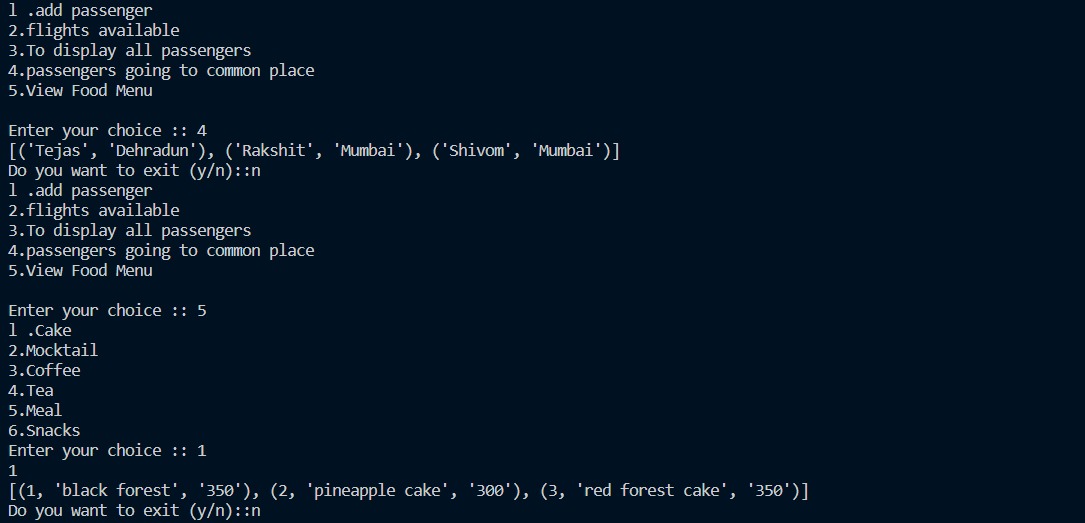
**else:**

**continue**

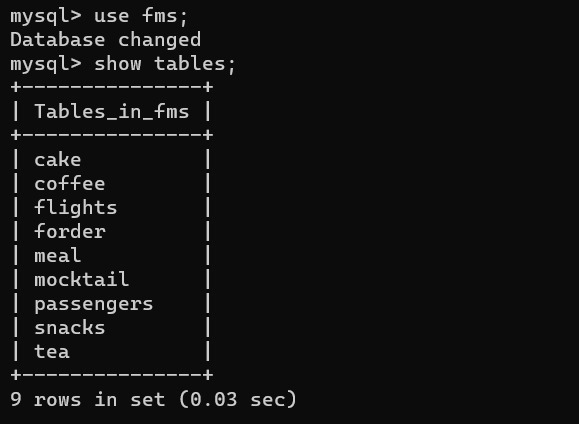
**OUTPUT**

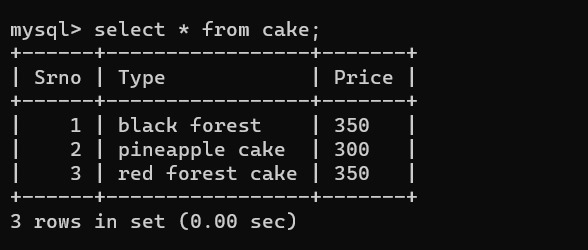


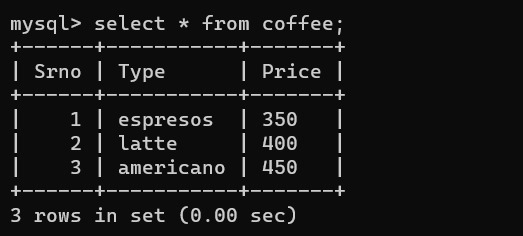


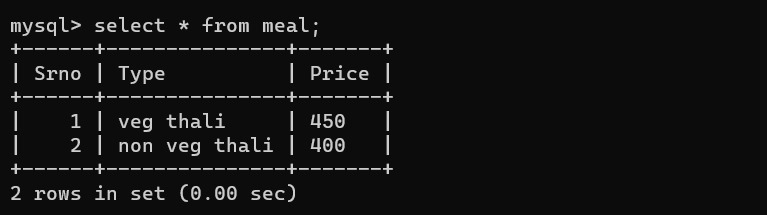


**Backend Output**

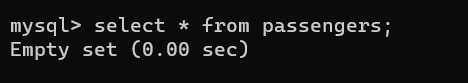




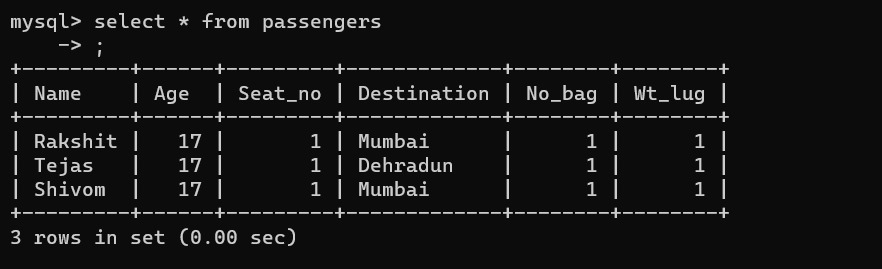




**Before**



**After**



**Bibliography**

* **Brilliant Minds-Youtube**
* **Tech-Gram Academy – Youtube**
* **Chronics – Youtube**