

**SVKM's NMIMS**  
**Mukesh Patel School of Technology Management & Engineering**  
A.Y. 2022 - 23  
**Course: Database Management Systems**

**Project Report**

Program	B TECH ARTIFICIAL INTELLIGENCE	
Semester	III	
Name of the Project:	FLIGHT TICKET BOOKING SYSTEM	
Details of Project Members		
Batch	Roll No.	Name
B3	I060	Mohammad Adil Shaikh
B3	I064	Mohammed Az Syed
B3	I066	Srihari Thyagarajan
B3	I068	Jash Vasa
Date of Submission: 16/10/2022		

**Contribution of each project Members:**

Roll No.	Name:	Contribution
I060	Mohammad Adil Shaikh	Database design, addition of required entries, fetching data for Flights, Bug testing.
I064	Mohammed Az Syed	Database design, addition of required entries, determining user flow for GUI, Bug testing.
I066	Srihari Thyagarajan	Relation designs, Python – MYSQL connection, adding OTP verification system for new account creation, Project Source Code, PyQt5 GUI Screens implementation for displaying appropriate data, ER diagram, Relational Schema, Bug Testing.
I068	Jash Vasa	Database design, addition of required entries, ER diagram, Relational Schema, Creating tables, Normalization, Bug testing.

**Note:**

1. Create a readme file if you have multiple files

2. All files must be properly named (I004\_DBMSProject)
3. Submit all relevant files of your work (Report, all SQL files, Any other files)
4. **Plagiarism is highly discouraged (Your report will be checked for plagiarism)**

**Rubrics for the Project evaluation:**

- Innovative Ideas and self-learning (5 Marks) Idea should not be regular such as Hotel, Library Management system etc.
- Implementation and Design (10 Marks) It includes ER model, Relational model, and Normalization of tables.
- Project Demonstration and Viva (5 Marks)

# **Project Report**

## **FLIGHT TICKET BOOKING SYSTEM**

**BY**

**Student 1 - Mohammad Adil Shaikh , Roll number: I060**

**Student 2 - Mohammad Az Syed, Roll number: I064**

**Student 3 - Srihari Thyagarajan, Roll number: I066**

**Student 4 – Jash Vasa, Roll number: I068**

**Course: DBMS AY: 2022-23**

<b>Sr no.</b>	<b>Topic</b>	<b>Page no.</b>
<b>1</b>	Storyline	5
<b>2</b>	Components of Database Design	6
<b>3</b>	Entity Relationship Diagram	7
<b>4</b>	Relational Model	8
<b>5</b>	Normalization	10
<b>6</b>	SQL Queries	12
<b>7</b>	Learning from the Project	33
<b>8</b>	Challenges you faced while doing the Project	34
<b>9</b>	Conclusion	35

# **I. Storyline**

This section should describe the requirements for the chosen database topic. Form a storyline and describe in detail.

## **Flight Ticket Management System -**

In a world where tourism is on the rise and is a major addition to a country's GDP, it is utterly important to have an assorted booking system. In our project we focus on only Flight booking system via python integrated SQL program with basic UI for easy and seamless flight booking for your perfect vacation or business-related travel.

Rajeev is a business advisor from Mumbai. He owns a firm that takes tasks related to business setup, financial plans, business expansion procedures, and many other business tasks. Since this job requires him to travel frequently, he would expect to use a platform that requires the least booking efforts is exactly what our flight booking platform, FlyHigh is designed for. Rajeev has to login/register, for login it takes 1 click and registering takes 3 clicks. This now directs him to the main booking page where he has to enter his flight details (origin, destination, time, company). After you click on the "Find Flights" it searches the companies database for available flights on that date and displays all the available option, he should note down the flight ID through which he wished to travel. He is asked if he wants to update his information, if he clicks next, we proceed to confirm flight details page where you must enter the flight id and load the details of that flight. After entering that he proceeds to summary page which asks additional questions and directs him to the payment page where he enters his payment details and confirms his booking. If he wishes to cancel his booking, he can do so with one effortless click.

## II. Components of Database Design

Description of all entities along with their attributes here along with the primary keys for each entity.

Description of all relationships among various entities along with the specification of the cardinality and participation for all relationships (present in ERD and Relational Schema).

DESC payment;

Field	Type	Null	Key	Default	Extra
Payment_ID	int	NO	PRI	NULL	
Payment_Customer_ID	int	NO		NULL	
Payment_Cost	float	NO		NULL	
Payment_Tax	float	NO		NULL	
Payment_Date	date	NO		NULL	
Payment_Type	varchar(15)	NO		NULL	
Payment_Card_No	int	NO		NULL	

DESC Flights;

Field	Type	Null	Key	Default	Extra
F_ID	int	NO	PRI	NULL	
F_Dept_Location	varchar(25)	NO		NULL	
F_Arr_Location	varchar(25)	NO		NULL	
F_Company	varchar(25)	NO		NULL	
F_Duration	int	NO		NULL	
F_Dept_Time	datetime	NO		NULL	
F_Arr_Time	datetime	NO		NULL	
F_Seats	int	NO		NULL	
C_ID	int	YES	MUL	NULL	

DESC Cancellation;

Field	Type	Null	Key	Default	Extra
Canc_ID	int	NO	PRI	NULL	
Canc_Payment_ID	int	NO		NULL	
Canc_Refund	float	NO		NULL	
Canc_Date	date	NO		NULL	

DESC Company;

Result Grid						
Filter Rows:						
Export:						
Wrap Cell Content:						
	Field	Type	Null	Key	Default	Extra
▶	C_Name	varchar(25)	NO		NULL	
	C_Type	varchar(25)	NO		NULL	
	C_ID	int	NO	PRI	NULL	

DESC Initial\_Info\_Account;

Result Grid						
Filter Rows:						
Export:						
Wrap Cell Content:						
	Field	Type	Null	Key	Default	Extra
▶	Username	varchar(20)	YES		NULL	
	Password	varchar(15)	YES		NULL	

DESC Full\_Profile\_Account;

Result Grid						
Filter Rows:						
Export:						
Wrap Cell Content:						
	Field	Type	Null	Key	Default	Extra
▶	Customer_ID	int	YES		NULL	
	Customer_Name	varchar(20)	YES		NULL	
	Customer_State	varchar(15)	YES		NULL	
	Customer_Country	varchar(15)	YES		NULL	
	Customer_Pincode	int	YES		NULL	
	Date_Of_Birth	date	YES		NULL	
	Customer_Gender	varchar(40)	YES		NULL	
	PhoneNumber	varchar(10)	YES		NULL	

DESC Indivo;

Result Grid						
Filter Rows:						
Export:						
Wrap Cell Content:						
	Field	Type	Null	Key	Default	Extra
▶	C_ID	int	NO	PRI	NULL	
	COMPANY_NAME	varchar(20)	YES		NULL	
	COMPANY_HISTORY	varchar(200)	YES		NULL	
	C_TYPE	varchar(20)	YES		NULL	

DESC Nistara;

Result Grid						
Filter Rows:						
Export:						
Wrap Cell Content:						
	Field	Type	Null	Key	Default	Extra
▶	C_ID	int	NO	PRI	NULL	
	COMPANY_NAME	varchar(20)	YES		NULL	
	COMPANY_HISTORY	varchar(200)	YES		NULL	
	C_TYPE	varchar(20)	YES		NULL	

DESC MetAirways;

Result Grid						
Filter Rows:		Export:		Wrap Cell Content:		
	Field	Type	Null	Key	Default	Extra
►	C_ID	int	NO	PRI	NULL	
	COMPANY_NAME	varchar(20)	YES		NULL	
	COMPANY_HISTORY	varchar(200)	YES		NULL	
	C_TYPE	varchar(20)	YES		NULL	

DESC Picejet;

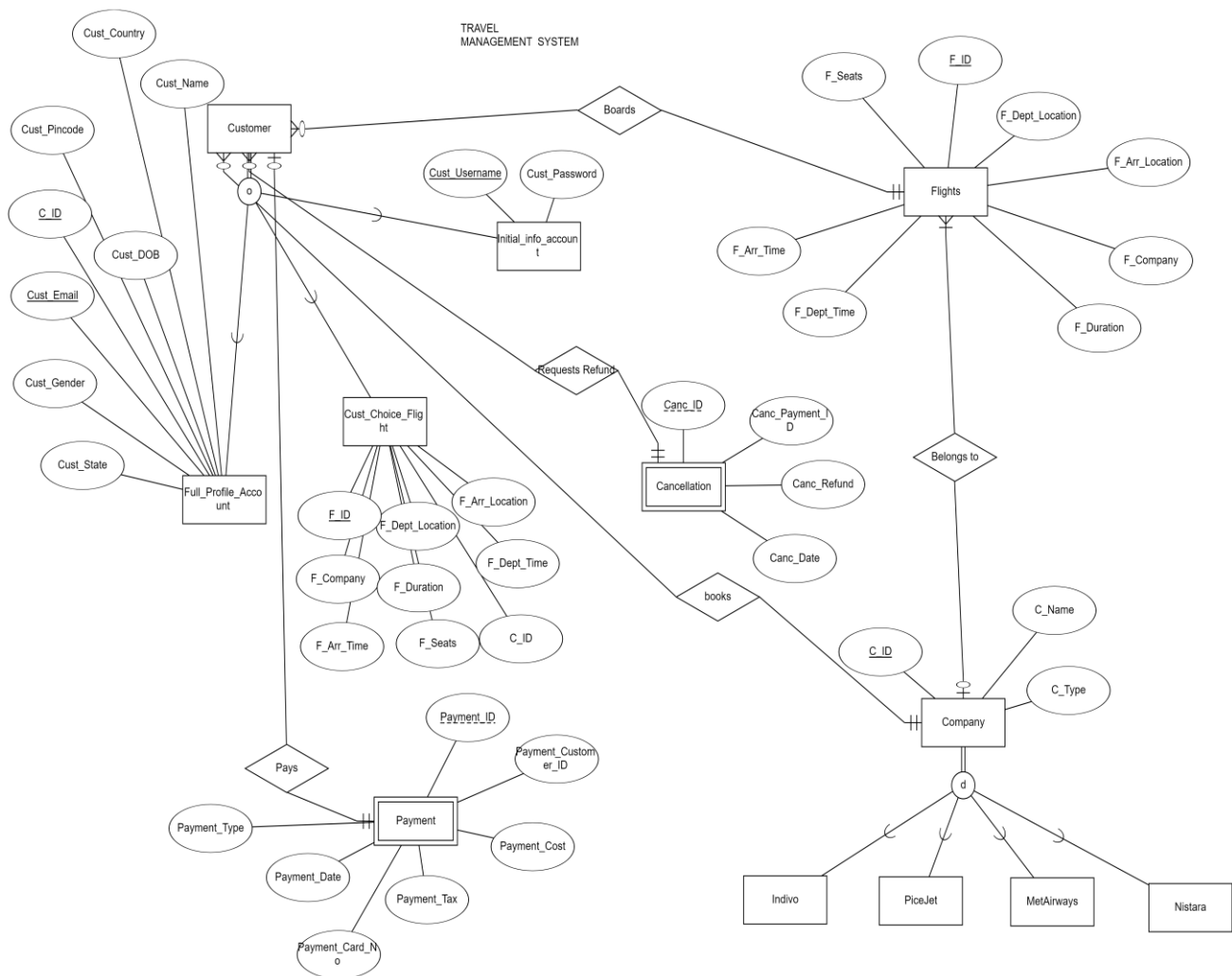
Result Grid						
Filter Rows:		Export:		Wrap Cell Content:		
	Field	Type	Null	Key	Default	Extra
►	C_ID	int	NO	PRI	NULL	
	COMPANY_NAME	varchar(20)	YES		NULL	
	COMPANY_HISTORY	varchar(200)	YES		NULL	
	C_TYPE	varchar(20)	YES		NULL	

DESC Cust\_Choice\_Flight;

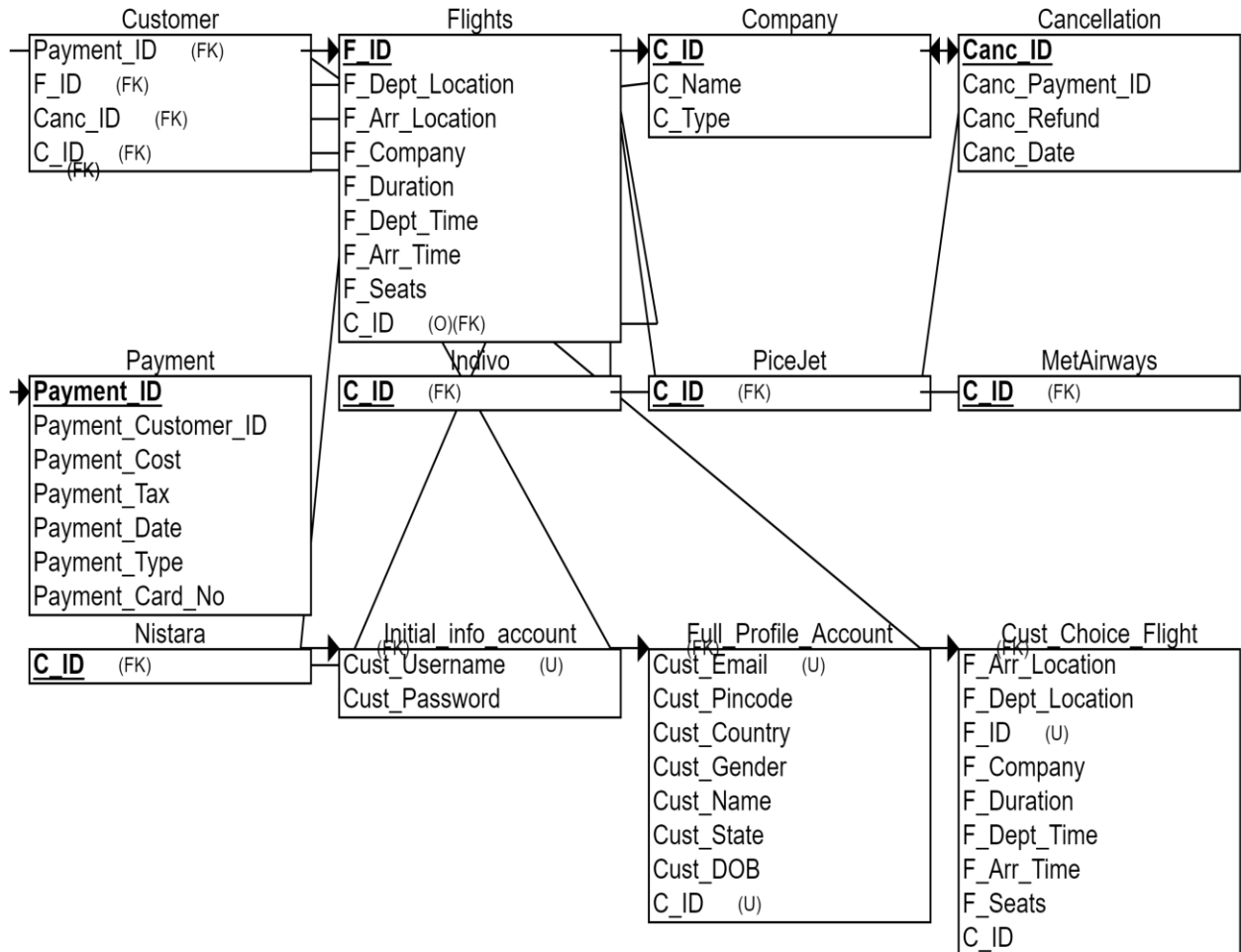
Result Grid						
Filter Rows:		Export:		Wrap Cell Content:		
	Field	Type	Null	Key	Default	Extra
►	F_ID	int	NO		NULL	
	F_Dept_Location	varchar(25)	NO		NULL	
	F_Arr_Location	varchar(25)	NO		NULL	
	F_Company	varchar(25)	NO		NULL	
	F_Duration	int	NO		NULL	
	F_Dept_Time	datetime	NO		NULL	
	F_Arr_Time	datetime	NO		NULL	
	F_Seats	int	NO		NULL	
	C_ID	int	YES	MUL	NULL	



# III. Entity Relationship Diagram



## IV. Relational Model



## V. Normalization

Perform normalization (1NF, 2NF, 3NF, BCNF) as applicable for the entire database.

Table **Payment:** -

- All key attributes are defined.
- No repeating groups in the table.
- All attributes depend on the primary key.  
Hence in **1NF**.
- No partial dependencies.

Hence in **2NF**.

- Since there are transitive dependencies (Payment\_tax is dependent on Payment\_cost)

Hence not in **3NF**.

Table **Flights:** -

- All key attributes are defined.
- No repeating groups in the table.
- All attributes depend on the primary key.

Hence in **1NF**.

- Since there are partial dependencies (**F\_company** is dependent on **C\_ID** as well as **F\_ID**)  
Hence not **2NF**.

Table **Cancellation:** -

- All key attributes are defined.
- No repeating groups in the table.
- All attributes depend on the primary key.  
Hence in **1NF**.
- No partial dependencies.

Hence in **2NF**.

- Contains no partial dependencies.

Hence in **3NF**.

- Since not all determinants are candidate keys (**Canc\_Date**)

Hence not in **BCNF**.

Table **Company:** -

- All key attributes are defined.
- No repeating groups in the table.
- All attributes depend on the primary key.  
Hence in **1NF**.
- No partial dependencies.

Hence in **2NF**.

- Contains no partial dependencies.

Hence in **3NF**.

- Since not all determinants are candidate keys (**C\_Type**)  
Hence not in **BCNF**.

Table **Indivo**: -

- All key attributes are defined.
- No repeating groups in the table.
- All attributes depend on the primary key.  
Hence in **1NF**.
- No partial dependencies.

Hence in **2NF**.

- Contains no partial dependencies.

Hence in **3NF**.

- Since not all determinants are candidate keys (**C\_Type, Company\_History**)  
Hence not in **BCNF**.

Table **Nistara**: -

- All key attributes are defined.
- No repeating groups in the table.
- All attributes depend on the primary key.  
Hence in **1NF**.
- No partial dependencies.

Hence in **2NF**.

- Contains no partial dependencies.

Hence in **3NF**.

- Since not all determinants are candidate keys (**C\_Type, Company\_History**)  
Hence not in **BCNF**.

Table **MetAirways**: -

- All key attributes are defined.
- No repeating groups in the table.
- All attributes depend on the primary key.  
Hence in **1NF**.
- No partial dependencies.

Hence in **2NF**.

- Contains no partial dependencies.

Hence in **3NF**.

- Since not all determinants are candidate keys (**C\_Type, Company\_History**)  
Hence not in **BCNF**.

Table **Picejet**: -

- All key attributes are defined.
- No repeating groups in the table.
- All attributes depend on the primary key.  
Hence in **1NF**.
- No partial dependencies.

Hence in **2NF**.

- Contains no partial dependencies.

Hence in **3NF**.

- Since not all determinants are candidate keys (**C\_Type, Company\_History**)  
Hence not in **BCNF**.

## VI. SQL Queries

Using a DBMS software (SQLite3 or MySQL or any other of your choice):

- Create the tables
- Populate the tables (insert some meaningful data, at least 10 tuples for each relation)
- Run SQL queries (minimum 15) covering **all concepts** learned in the class

This section should contain the question, SQL code, and the output snapshot for each query.

Creation of required tables -

```
CREATE TABLE IF NOT EXISTS Company
```

```
(  
  C_Name VARCHAR(25) NOT NULL,  
  C_Type VARCHAR(25) NOT NULL,  
  C_ID INT NOT NULL,  
  PRIMARY KEY (C_ID)  
);
```

```
CREATE TABLE IF NOT EXISTS Cancellation
```

```
(  
  Canc_ID INT NOT NULL,  
  Canc_Payment_ID INT NOT NULL,  
  Carcerand FLOAT NOT NULL,  
  Canc_Date DATE NOT NULL,  
  PRIMARY KEY (Canc_ID)  
);
```

```
CREATE TABLE IF NOT EXISTS Payment
```

```
(  
  Payment_ID INT NOT NULL,  
  Payment_Customer_ID INT NOT NULL,  
  Payment_Cost FLOAT NOT NULL,
```

```
Payment_Tax FLOAT NOT NULL,  
Payment_Date DATE NOT NULL,  
Payment_Type VARCHAR(15) NOT NULL,  
Payment_Card_No INT NOT NULL,  
PRIMARY KEY (Payment_ID)  
);
```

```
CREATE TABLE IF NOT EXISTS Indivo  
(  
    C_ID INT NOT NULL,  
    Company_Name VARCHAR(20) NOT NULL,  
    Company_History VARCHAR(200) NOT NULL,  
    C_Type INT NOT NULL,  
    PRIMARY KEY (C_ID),  
    FOREIGN KEY (C_ID) REFERENCES Company(C_ID)  
);
```

```
CREATE TABLE IF NOT EXISTS Picejet  
(  
    C_ID INT NOT NULL,  
    Company_Name VARCHAR(20) NOT NULL,  
    Company_History VARCHAR(200) NOT NULL,  
    C_Type INT NOT NULL,  
    PRIMARY KEY (C_ID),  
    FOREIGN KEY (C_ID) REFERENCES Company(C_ID)  
);
```

```
CREATE TABLE IF NOT EXISTS Metairways  
(  
    C_ID INT NOT NULL,
```

```
Company_Name VARCHAR(20) NOT NULL,  
Company_History VARCHAR(200) NOT NULL,  
C_Type INT NOT NULL,  
PRIMARY KEY (C_ID),  
FOREIGN KEY (C_ID) REFERENCES Company(C_ID)  
);
```

```
CREATE TABLE IF NOT EXISTS Nistara
```

```
(  
    C_ID INT NOT NULL,  
    Company_Name VARCHAR(20) NOT NULL,  
    Company_History VARCHAR(200) NOT NULL,  
    C_Type INT NOT NULL,  
    PRIMARY KEY (C_ID),  
    FOREIGN KEY (C_ID) REFERENCES Company(C_ID)  
);
```

```
CREATE TABLE IF NOT EXISTS Flights
```

```
(  
    F_ID INT NOT NULL,  
    F_Dept_Location VARCHAR(25) NOT NULL,  
    F_Arr_Location VARCHAR(25) NOT NULL,  
    F_Company VARCHAR(25) NOT NULL,  
    F_Duration INT NOT NULL,  
    F_Dept_Time DATETIME NOT NULL,  
    F_Arr_Time DATETIME NOT NULL,  
    F_Seats INT NOT NULL,  
    C_ID INT,
```



```
PRIMARY KEY (F_ID),  
FOREIGN KEY (C_ID) REFERENCES Company(C_ID)  
);
```

```
CREATE TABLE IF NOT EXISTS Cust_Choice_Flight  
(  
    F_ID INT NOT NULL,  
    F_Dept_Location VARCHAR(25) NOT NULL,  
    F_Arr_Location VARCHAR(25) NOT NULL,  
    F_Company VARCHAR(25) NOT NULL,  
    F_Duration INT NOT NULL,  
    F_Dept_Time DATETIME NOT NULL,  
    F_Arr_Time DATETIME NOT NULL,  
    F_Seats INT NOT NULL,  
    C_ID INT,  
    PRIMARY KEY (F_ID),  
    FOREIGN KEY (C_ID) REFERENCES Company(C_ID)  
);
```

Insertion of data to respective tables (Population of data) -

Initial Info account relation -

```
INSERT INTO initial_info_account VALUES("Sri", "a");
```

Full Profile Account relation -

```
INSERT INTO full_profile_account VALUES(2, "Sri", "MH", "IND", 403407, "2003-01-10", "Male",  
"3432");
```

Flights relation -

1<sup>st</sup> Oct data sample -

```
INSERT INTO `dbms_project`.`flights` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`, `F_Company`,  
`F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2001', 'Bangalore', 'Mumbai',  
'Indivo', '2', '2022-10-01 10:10:00', '2022-10-01 12:10:00', '300', '1');
```

```
INSERT INTO `dbms_project`.`flights` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`, `F_Company`,  
`F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2002', 'Bangalore', 'Mumbai',  
'Indivo', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '1');
```

```
INSERT INTO `dbms_project`.`flights` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`, `F_Company`,  
`F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2003', 'Bangalore', 'Mumbai',  
'MetAirways', '2', '2022-10-01 10:10:00', '2022-10-01 12:10:00', '300', '2');
```

```
INSERT INTO `dbms_project`.`flights` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`, `F_Company`,  
`F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2004', 'Bangalore', 'Mumbai',  
'MetAirways', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '2');
```

```
INSERT INTO `dbms_project`.`flights` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`, `F_Company`,  
`F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2005', 'Bangalore', 'Mumbai',  
'Picejet', '2', '2022-10-01 10:10:00', '2022-10-01 12:10:00', '300', '3');
```

```
INSERT INTO `dbms_project`.`flights` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`, `F_Company`,  
`F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2006', 'Bangalore', 'Mumbai',  
'Picejet', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '3');
```

```
INSERT INTO `dbms_project`.`flights` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`, `F_Company`,  
`F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2007', 'Bangalore', 'Mumbai',  
'Nistara', '2', '2022-10-01 10:10:00', '2022-10-01 12:10:00', '300', '4');
```

```
INSERT INTO `dbms_project`.`flights` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`, `F_Company`,  
`F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2008', 'Bangalore', 'Mumbai',  
'Nistara', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '4');
```

```
INSERT INTO `dbms_project`.`flights` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`, `F_Company`,  
`F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2009', 'Bangalore', 'Delhi',  
'Indivo', '2', '2022-10-01 10:10:00', '2022-10-01 12:10:00', '300', '1');
```

```
INSERT INTO `dbms_project`.`flights` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`, `F_Company`,  
`F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2010', 'Bangalore', 'Delhi',  
'Indivo', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '1');
```

```
INSERT INTO `dbms_project`.`flights` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`, `F_Company`,  
`F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2011', 'Bangalore', 'Delhi',  
'MetAirways', '2', '2022-10-01 10:10:00', '2022-10-01 12:10:00', '300', '2');
```

```
INSERT INTO `dbms_project`.`flights` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`, `F_Company`,  
`F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2012', 'Bangalore', 'Delhi',
```

'MetAirways', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '2');

INSERT INTO `dbms\_project`.`flights` (`F\_ID`, `F\_Dept\_Location`, `F\_Arr\_Location`, `F\_Company`, `F\_Duration`, `F\_Dept\_Time`, `F\_Arr\_Time`, `F\_Seats`, `C\_ID`) VALUES ('2013', 'Bangalore', 'Delhi', 'Picejet', '2', '2022-10-01 10:10:00', '2022-10-01 12:10:00', '300', '3');

INSERT INTO `dbms\_project`.`flights` (`F\_ID`, `F\_Dept\_Location`, `F\_Arr\_Location`, `F\_Company`, `F\_Duration`, `F\_Dept\_Time`, `F\_Arr\_Time`, `F\_Seats`, `C\_ID`) VALUES ('2014', 'Bangalore', 'Delhi', 'Picejet', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '3');

INSERT INTO `dbms\_project`.`flights` (`F\_ID`, `F\_Dept\_Location`, `F\_Arr\_Location`, `F\_Company`, `F\_Duration`, `F\_Dept\_Time`, `F\_Arr\_Time`, `F\_Seats`, `C\_ID`) VALUES ('2015', 'Bangalore', 'Delhi', 'Nistara', '2', '2022-10-01 10:10:00', '2022-10-01 12:10:00', '300', '4');

INSERT INTO `dbms\_project`.`flights` (`F\_ID`, `F\_Dept\_Location`, `F\_Arr\_Location`, `F\_Company`, `F\_Duration`, `F\_Dept\_Time`, `F\_Arr\_Time`, `F\_Seats`, `C\_ID`) VALUES ('2016', 'Bangalore', 'Delhi', 'Nistara', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '4');

INSERT INTO `dbms\_project`.`flights` (`F\_ID`, `F\_Dept\_Location`, `F\_Arr\_Location`, `F\_Company`, `F\_Duration`, `F\_Dept\_Time`, `F\_Arr\_Time`, `F\_Seats`, `C\_ID`) VALUES ('2017', 'Bangalore', 'Chennai', 'Indivo', '2', '2022-10-01 10:10:00', '2022-10-01 12:10:00', '300', '1');

INSERT INTO `dbms\_project`.`flights` (`F\_ID`, `F\_Dept\_Location`, `F\_Arr\_Location`, `F\_Company`, `F\_Duration`, `F\_Dept\_Time`, `F\_Arr\_Time`, `F\_Seats`, `C\_ID`) VALUES ('2018', 'Bangalore', 'Chennai', 'Indivo', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '1');

INSERT INTO `dbms\_project`.`flights` (`F\_ID`, `F\_Dept\_Location`, `F\_Arr\_Location`, `F\_Company`, `F\_Duration`, `F\_Dept\_Time`, `F\_Arr\_Time`, `F\_Seats`, `C\_ID`) VALUES ('2019', 'Bangalore', 'Chennai', 'MetAirways', '2', '2022-10-01 10:10:00', '2022-10-01 12:10:00', '300', '2');

INSERT INTO `dbms\_project`.`flights` (`F\_ID`, `F\_Dept\_Location`, `F\_Arr\_Location`, `F\_Company`, `F\_Duration`, `F\_Dept\_Time`, `F\_Arr\_Time`, `F\_Seats`, `C\_ID`) VALUES ('2020', 'Bangalore', 'Chennai', 'MetAirways', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '2');

INSERT INTO `dbms\_project`.`flights` (`F\_ID`, `F\_Dept\_Location`, `F\_Arr\_Location`, `F\_Company`, `F\_Duration`, `F\_Dept\_Time`, `F\_Arr\_Time`, `F\_Seats`, `C\_ID`) VALUES ('2021', 'Bangalore', 'Chennai', 'Picejet', '2', '2022-10-01 10:10:00', '2022-10-01 12:10:00', '300', '3');

INSERT INTO `dbms\_project`.`flights` (`F\_ID`, `F\_Dept\_Location`, `F\_Arr\_Location`, `F\_Company`, `F\_Duration`, `F\_Dept\_Time`, `F\_Arr\_Time`, `F\_Seats`, `C\_ID`) VALUES ('2022', 'Bangalore', 'Chennai', 'Picejet', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '3');

INSERT INTO `dbms\_project`.`flights` (`F\_ID`, `F\_Dept\_Location`, `F\_Arr\_Location`, `F\_Company`, `F\_Duration`, `F\_Dept\_Time`, `F\_Arr\_Time`, `F\_Seats`, `C\_ID`) VALUES ('2023', 'Bangalore', 'Chennai', 'Nistara', '2', '2022-10-01 10:10:00', '2022-10-01 12:10:00', '300', '4');

INSERT INTO `dbms\_project`.`flights` (`F\_ID`, `F\_Dept\_Location`, `F\_Arr\_Location`, `F\_Company`, `F\_Duration`, `F\_Dept\_Time`, `F\_Arr\_Time`, `F\_Seats`, `C\_ID`) VALUES ('2024', 'Bangalore', 'Chennai', 'Nistara', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '4');

Customer Flight Choice relation -

```
INSERT INTO `dbms_project`.`Cust_Choice_Flight` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`,  
`F_Company`, `F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2024',  
'Bangalore', 'Chennai', 'Nistara', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '4');
```

```
INSERT INTO `dbms_project`.`Cust_Choice_Flight` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`,  
`F_Company`, `F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2024',  
'Bangalore', 'Chennai', 'Metairways', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '4');
```

```
INSERT INTO `dbms_project`.`Cust_Choice_Flight` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`,  
`F_Company`, `F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2024',  
'Bangalore', 'Chennai', 'Picejet', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '4');
```

```
INSERT INTO `dbms_project`.`Cust_Choice_Flight` (`F_ID`, `F_Dept_Location`, `F_Arr_Location`,  
`F_Company`, `F_Duration`, `F_Dept_Time`, `F_Arr_Time`, `F_Seats`, `C_ID`) VALUES ('2024',  
'Bangalore', 'Chennai', 'Indivo', '2', '2022-10-01 18:10:00', '2022-10-01 20:10:00', '300', '4');
```

Company relation -

```
INSERT INTO COMPANY VALUES('Indivo', 'Airbus 380', 1);
```

```
INSERT INTO COMPANY VALUES('Picejet', 'Airbus 380', 3);
```

```
INSERT INTO COMPANY VALUES('Nistara', 'Airbus 380', 4);
```

```
INSERT INTO COMPANY VALUES('Metairways', 'Airbus 380', 2);
```

Cancellation Relation -

```
INSERT INTO Cancellation VALUES(543534, 345345, 2233.3, '2022-10-16');
```

```
INSERT INTO Cancellation VALUES(756, 23, 543.3, '2022-09-16');
```

```
INSERT INTO Cancellation VALUES(654, 43, 234.3, '2022-11-16');
```

```
INSERT INTO Cancellation VALUES(543, 54, 563.3, '2022-12-16');
```

Payment Relation -

```
INSERT INTO `dbms_project`.`Payment` (`Payment_ID`, `Payment_Customer_ID`, `Payment_Cost`,  
`Payment_Tax`, `Payment_Date`, `Payment_Type`, `Payment_Card_No`) VALUES ('42334', '5434', '3242',  
'234', '2022-10-16', 'Cash', '435');
```

```
INSERT INTO `dbms_project`.`Payment` (`Payment_ID`, `Payment_Customer_ID`, `Payment_Cost`,
`Payment_Tax`, `Payment_Date`, `Payment_Type`, `Payment_Card_No`) VALUES ('4334', '5434', '3242',
'234', '2022-11-16', 'Cash', '435');
```

```
INSERT INTO `dbms_project`.`Payment` (`Payment_ID`, `Payment_Customer_ID`, `Payment_Cost`,
`Payment_Tax`, `Payment_Date`, `Payment_Type`, `Payment_Card_No`) VALUES ('424', '5434', '3242',
'234', '2022-09-16', 'Cash', '435');
```

```
INSERT INTO `dbms_project`.`Payment` (`Payment_ID`, `Payment_Customer_ID`, `Payment_Cost`,
`Payment_Tax`, `Payment_Date`, `Payment_Type`, `Payment_Card_No`) VALUES ('443534', '5434',
'3242', '234', '2022-08-16', 'Cash', '435');
```

Question, Queries and respective output -

Select Queries:

```
SELECT * FROM dbms_project.initial_info_account;
```

	Username	Password
►	Srihari	aa
	GG	yo
	Haleshot	ggs

```
SELECT * FROM dbms_project.full_profile_account;
```

Customer_ID	Customer_Name	Customer_State	Customer_Country	Customer_Pincode	Date_Of_Birth	Customer_Gender	PhoneNumber
3	Srihari	JH	IND	49323	2004-01-01	Non Binary	645645
4	GG	CH	IND	3242	2009-02-01	I prefer not to say	543
5	Haleshot	PY	IND	234	2000-02-19	Non Binary	523

WHERE Query:

```
SELECT *, FROM dbms_project.flights WHERE F_ID = '108';
```

F_ID	F_Dept_Location	F_Arr_Location	F_Company	F_Duration	F_Dept_Time	F_Arr_Time	F_Seats	C_ID
108	Delhi	Mumbai	Nistara	2	2022-10-01 04:00:00	2022-10-01 06:00:00	300	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Update Query -

```
UPDATE full_profile_account SET Customer_ID = 234 WHERE Customer_ID = 34534
```

ORDER BY:

Customer_ID	Customer_Name	Customer_State	Customer_Country	Customer_Pincode	Date_Of_Birth	Customer_Gender	PhoneNumber
5	Haleshot	PY	IND	234	2000-02-19	Non Binary	523
4	GG	CH	IND	3242	2009-02-01	I prefer not to say	543
3	Srihari	JH	IND	49323	2004-01-01	Non Binary	645645

VIEW CREATED FOR CUSTOMER:

CREATE VIEW FULL\_CUSTOMER\_INFORMATION as

(SELECT Customer\_Name, Customer\_State, Customer\_Country, Customer\_Pincode, Date\_Of\_Birth, Customer\_Gender, PhoneNumber FROM initial\_info\_account iia, FULL\_PROFILE\_ACCOUNT ffa WHERE iia.Username = ffa.Customer\_Name);

SELECT \* FROM FULL\_CUSTOMER\_INFORMATION;

Customer_Name	Customer_State	Customer_Country	Customer_Pincode	Date_Of_Birth	Customer_Gender	PhoneNumber
Srihari	JH	IND	49323	2004-01-01	Non Binary	645645
GG	CH	IND	3242	2009-02-01	I prefer not to say	543
Haleshot	PY	IND	234	2000-02-19	Non Binary	523

Nested Query:

SELECT \* FROM full\_profile\_account where Customer\_Name IN (SELECT Username FROM initial\_info\_account);

Customer_ID	Customer_Name	Customer_State	Customer_Country	Customer_Pincode	Date_Of_Birth	Customer_Gender	PhoneNumber
3	Srihari	JH	IND	49323	2004-01-01	Non Binary	645645
4	GG	CH	IND	3242	2009-02-01	I prefer not to say	543
5	Haleshot	PY	IND	234	2000-02-19	Non Binary	523

Delete with nested subquery:

DELETE iia, ffa FROM initial\_info\_account iia INNER JOIN FULL\_PROFILE\_ACCOUNT ffa ON(iia.Username = ffa.Customer\_Name)

WHERE ffa.Customer\_Name =

(SELECT \* FROM (SELECT Customer\_Name FROM FULL\_PROFILE\_ACCOUNT ffa WHERE ffa.Customer\_Name = 'user' LIMIT 1) FULL\_PROFILE\_ACCOUNT);

	Username	Password
▶	Srihari	aa
	GG	yo
	Haleshot	ggs

	Customer_ID	Customer_Name	Customer_State	Customer_Country	Customer_Pincode	Date_Of_Birth	Customer_Gender	PhoneNumber
▶	3	Srihari	JH	IND	49323	2004-01-01	Non Binary	645645
	4	GG	CH	IND	3242	2009-02-01	I prefer not to say	543
	5	Haleshot	PY	IND	234	2000-02-19	Non Binary	523

The row user is deleted from both tables.

Join Query:

SELECT \* FROM COMPANY NATURAL JOIN metairways;

Company Type	Company ID	Company Name	Company History
Airbus 380	2	Met Airways	Launched in 1993, Jet Airways grew to become the airline ...

LIKE Query:

SELECT \* FROM flights WHERE F\_Dept\_Location = 'Mumbai' AND F\_Arr\_Location = 'Chennai' AND F\_Company = 'Indivo' AND (F\_Dept\_Time LIKE '%18:10%' or F\_Dept\_Time LIKE '%19:30%' or F\_Dept\_Time LIKE '%20:00%' or F\_Dept\_Time LIKE '%21:00%')

Flight ID	Flight Departure	Flight Arrival	Flight Company	Flight Duration	Departure Time	Arrival Time	Seats	Company ID
5	Mumbai	Chennai	Indivo	2	2022-10-01 ...	2022-10-01 ...	300	1
29	Mumbai	Chennai	Indivo	2	2022-10-02 ...	2022-10-02 ...	300	1
53	Mumbai	Chennai	Indivo	2	2022-10-03 ...	2022-10-03 ...	300	1
77	Mumbai	Chennai	Indivo	2	2022-10-04 ...	2022-10-04 ...	300	1
1001	Mumbai	Chennai	Indivo	2	2022-10-05 ...	2022-10-05 ...	300	1
1025	Mumbai	Chennai	Indivo	2	2022-10-06 ...	2022-10-06 ...	300	1
1049	Mumbai	Chennai	Indivo	2	2022-10-07 ...	2022-10-07 ...	300	1
1073	Mumbai	Chennai	Indivo	2	2022-10-08 ...	2022-10-08 ...	300	1
1097	Mumbai	Chennai	Indivo	2	2022-10-09 ...	2022-10-09 ...	300	1
1121	Mumbai	Chennai	Indivo	2	2022-10-10 ...	2022-10-10 ...	300	1

SELECT \* FROM flights WHERE F\_Dept\_Location = 'Delhi' AND F\_Arr\_Location = 'Chennai'

Flight ID	Flight Departure	Flight Arrival	Flight Company	Flight Duration	Departure Time	Arrival Time	Seats	Company ID
109	Delhi	Chennai	Indivo	4	2022-10-01 ...	2022-10-01 ...	300	1
110	Delhi	Chennai	Indivo	4	2022-10-01 ...	2022-10-01 ...	300	1
111	Delhi	Chennai	MetAir	4	2022-10-01 ...	2022-10-01 ...	300	2
112	Delhi	Chennai	MetAir	4	2022-10-01 ...	2022-10-01 ...	300	2
113	Delhi	Chennai	Picejet	4	2022-10-01 ...	2022-10-01 ...	300	3
114	Delhi	Chennai	Picejet	4	2022-10-01 ...	2022-10-01 ...	300	3
115	Delhi	Chennai	Nistara	4	2022-10-01 ...	2022-10-01 ...	300	4
116	Delhi	Chennai	Nistara	4	2022-10-01 ...	2022-10-01 ...	300	4
133	Delhi	Chennai	Indivo	4	2022-10-02 ...	2022-10-02 ...	300	1
134	Delhi	Chennai	Indivo	4	2022-10-02 ...	2022-10-02 ...	300	1
135	Delhi	Chennai	MetAir	4	2022-10-02 ...	2022-10-02 ...	300	2

```
SELECT * FROM flights WHERE F_Dept_Location = 'Mumbai' AND F_Arr_Location = 'Chennai' AND
F_Company = 'Indivo' AND (F_Dept_Time LIKE '%18:10%' or F_Dept_Time LIKE '%19:30%' or
F_Dept_Time LIKE '%20:00%' or F_Dept_Time LIKE '%21:00%')
```

	Flight ID	Flight Departure	Flight Arrival	Flight Company	Flight Duration	Departure Time	Arrival Time	Seats	Company ID
5		Mumbai	Chennai	Indigo	2	2022-10-01 ...	2022-10-01 ...	300	1
29		Mumbai	Chennai	Indigo	2	2022-10-02 ...	2022-10-02 ...	300	1
53		Mumbai	Chennai	Indigo	2	2022-10-03 ...	2022-10-03 ...	300	1
77		Mumbai	Chennai	Indigo	2	2022-10-04 ...	2022-10-04 ...	300	1
1001		Mumbai	Chennai	Indigo	2	2022-10-05 ...	2022-10-05 ...	300	1
1025		Mumbai	Chennai	Indigo	2	2022-10-06 ...	2022-10-06 ...	300	1
1049		Mumbai	Chennai	Indigo	2	2022-10-07 ...	2022-10-07 ...	300	1
1073		Mumbai	Chennai	Indigo	2	2022-10-08 ...	2022-10-08 ...	300	1
1097		Mumbai	Chennai	Indigo	2	2022-10-09 ...	2022-10-09 ...	300	1
1121		Mumbai	Chennai	Indigo	2	2022-10-10 ...	2022-10-10 ...	300	1

```
SELECT * FROM flights WHERE F_Dept_Location = 'Mumbai' AND F_Arr_Location = 'Chennai' AND
F_Company = 'Nistara' AND (F_Dept_Time LIKE '%18:10%' or F_Dept_Time LIKE '%19:30%' or
F_Dept_Time LIKE '%20:00%' or F_Dept_Time LIKE '%21:00%')
```

Flight ID	Flight Departure	Flight Arrival	Flight Company	Flight Duration	Departure Time	Arrival Time	Seats	Company ID
11	Mumbai	Chennai	Nistara	2	2022-10-01 ...	2022-10-01 ...	300	4
35	Mumbai	Chennai	Nistara	2	2022-10-02 ...	2022-10-02 ...	300	4
59	Mumbai	Chennai	Nistara	2	2022-10-03 ...	2022-10-03 ...	300	4
83	Mumbai	Chennai	Nistara	2	2022-10-04 ...	2022-10-04 ...	300	4
1007	Mumbai	Chennai	Nistara	2	2022-10-05 ...	2022-10-05 ...	300	4
1031	Mumbai	Chennai	Nistara	2	2022-10-06 ...	2022-10-06 ...	300	4
1055	Mumbai	Chennai	Nistara	2	2022-10-07 ...	2022-10-07 ...	300	4
1079	Mumbai	Chennai	Nistara	2	2022-10-08 ...	2022-10-08 ...	300	4
1103	Mumbai	Chennai	Nistara	2	2022-10-09 ...	2022-10-09 ...	300	4
1127	Mumbai	Chennai	Nistara	2	2022-10-10 ...	2022-10-10 ...	300	4

```
SELECT * FROM dbms_project.Payment
```

GROUP BY Payment\_Type;

[illegible]



## VI. Project demonstration

- Tools/software/ libraries used –

PyQt5 - a combination of Python language and the Qt library, with the help of ‘QtDesigner’ it becomes easier to create seamless user interface.

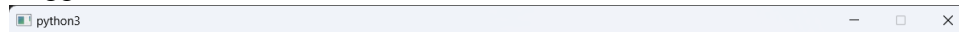
Twilio - a module used for communicating with the Twilio API. it is used to send SMS through python (we have used it to send the user an otp as they register with the application)

MySQL.Connector - enables python applications to access MySQL databases.

Random - used to generate random numbers in the specified limits for otp/Customer\_id/Payment\_id.

- Screenshot and Description of the Demonstration of project (If GUI is made)

Logged in User:



### Flight Booking Management System

Login/Sign Up Page

Login

Create an Account

python3

# Login

Sign in to your existing account

Username/Email ID \*

Haleshot

Password \*

...

Login

(If Find Flight Buttons is clicked without choosing From and To Drop Down boxes)

python3

## FLIGHT BOOKING TICKET SYSTEM

FROM \*

Select

TO \*

Select

Please input all \* fields!

Timings

Select

Airline Company

Select

Find Flights

Flight ID	Flight Departure	Flight Arrival	Flight Company	Flight Duration	Departure Time	Arrival Time	Seats	Company ID
-----------	------------------	----------------	----------------	-----------------	----------------	--------------	-------	------------

Next

python3

FLIGHT BOOKING TICKET SYSTEM

FROM \*

Mumbai

TO \*

Chennai

Timings

Select

Airline Company

Select

Find Flights

Flight ID	Flight Departure	Flight Arrival	Flight Company	Flight Duration	Departure Time	Arrival Time	Seats	Company ID
4	Mumbai	Chennai	Indivo	2	2022-10-01 10:10:00	2022-10-01 12:10:00	300	1
5	Mumbai	Chennai	Indivo	2	2022-10-01 18:10:00	2022-10-01 20:10:00	300	1
6	Mumbai	Chennai	MetAirways	2	2022-10-01 10:10:00	2022-10-01 12:10:00	300	2
7	Mumbai	Chennai	MetAirways	2	2022-10-01 18:10:00	2022-10-01 20:10:00	300	2
8	Mumbai	Chennai	Picejet	2	2022-10-01 10:10:00	2022-10-01 12:10:00	300	3
9	Mumbai	Chennai	Picejet	2	2022-10-01 18:10:00	2022-10-01 20:10:00	300	3
10	Mumbai	Chennai	Nistara	2	2022-10-01 10:10:00	2022-10-01 12:10:00	300	4
11	Mumbai	Chennai	Nistara	2	2022-10-01 18:10:00	2022-10-01 20:10:00	300	4
28	Mumbai	Chennai	Indivo	2	2022-10-02 10:10:00	2022-10-02 12:10:00	300	1
29	Mumbai	Chennai	Indivo	2	2022-10-02 18:10:00	2022-10-02 20:10:00	300	1

Next

With Timings and Airline Company Filter applied:

python3

FLIGHT BOOKING TICKET SYSTEM

FROM \*

Mumbai

TO \*

Chennai

Timings

Day

Airline Company

Indivo

Find Flights

Flight ID	Flight Departure	Flight Arrival	Flight Company	Flight Duration	Departure Time	Arrival Time	Seats	Company ID
4	Mumbai	Chennai	Indivo	2	2022-10-01 10:10:00	2022-10-01 12:10:00	300	1
28	Mumbai	Chennai	Indivo	2	2022-10-02 10:10:00	2022-10-02 12:10:00	300	1
52	Mumbai	Chennai	Indivo	2	2022-10-03 10:10:00	2022-10-03 12:10:00	300	1
76	Mumbai	Chennai	Indivo	2	2022-10-04 10:10:00	2022-10-04 12:10:00	300	1
1000	Mumbai	Chennai	Indivo	2	2022-10-05 10:10:00	2022-10-05 12:10:00	300	1
1024	Mumbai	Chennai	Indivo	2	2022-10-06 10:10:00	2022-10-06 12:10:00	300	1
1048	Mumbai	Chennai	Indivo	2	2022-10-07 10:10:00	2022-10-07 12:10:00	300	1
1072	Mumbai	Chennai	Indivo	2	2022-10-08 10:10:00	2022-10-08 12:10:00	300	1
1096	Mumbai	Chennai	Indivo	2	2022-10-09 10:10:00	2022-10-09 12:10:00	300	1
1120	Mumbai	Chennai	Indivo	2	2022-10-10 10:10:00	2022-10-10 12:10:00	300	1

Next

python3

# Update Customer Information

Customer Information:

Name	State	Country	Pincode	Date Of Birth	Gender	Phone Number
Haleshot	PY	IND	234	2000-02-19	Non Binary	523

Do you want to change Customer Information?

Yes

No

On clicking No:

python3

# CONFIRM FLIGHT DETAILS

Enter Flight ID : 

Load Details

(To proceed to Payment)

Flight ID	Flight Departure	Flight Arrival	Flight Company	Flight Duration	Departure Time	Arrival Time	Seats	Company ID

Proceed to Summary

Note - Selects the displayed Flight ID. Can be changed later in the Summary Screen.

python3

## CONFIRM FLIGHT DETAILS

Enter Flight ID :  [Load Details](#)

(To proceed to Payment)

Flight ID	Flight Departure	Flight Arrival	Flight Company	Flight Duration	Departure Time	Arrival Time	Seats	Company ID
121	Delhi	Bangalore	Picejet	3	2022-10-01 ...	2022-10-01 ...	300	3

[Proceed to Summary](#)

Note - Selects the displayed Flight ID. Can be changed later in the Summary Screen.

python3

## Summary

### Customer Information:

Name	State	Country	Pincode	Date Of Birth	Gender	Phone Number
Haleshot	PY	IND	234	2000-02-19	Non Binary	523

Do you want to change Customer Information? [Yes](#)

### Flight Information:

Flight ID	Flight Departure	Flight Arrival	Flight Company	Flight Duration	Departure Time	Arrival Time
121	Delhi	Bangalore	Picejet	3	2022-10-01 12:40:00	2022-10-01 15:00:00

Do you want to change Flight Information? [Yes](#) [See additional Flight Information](#)

Do you want to proceed to Payment ?

[Yes](#)

[No](#)

(Takes you to see list of available flights to choose from)

On clicking Additional Flight Information:

python3

Additional Flight Information

Company Type	Company ID	Company Name	Company History
Airbus 380	3	PiceJet	The origins of SpiceJet can be tracked back to March 198...

Back to Summary

python3

Summary

Customer Information:

Name	State	Country	Pincode	Date Of Birth	Gender	Phone Number
Haleshot	PY	IND	234	2000-02-19	Non Binary	523

Do you want to change Customer Information?

Yes

Flight Information:

Flight ID	Flight Departure	Flight Arrival	Flight Company	Flight Duration	Departure Time	Arrival Time
121	Delhi	Bangalore	PiceJet	3	2022-10-01 12:40:00	2022-10-01 15:00:00

Do you want to change Flight Information?

Yes

See additional Flight Information

Do you want to proceed to Payment ?

Yes

No

(Takes you to see list of available flights to choose from)

On clicking Yes:

python3

# Payment Info

Ticket Fare	Net Tax	Date
4139	413	2022-10-16

Customer ID \*

Enter your Customer ID...

Card Type \*

Card Number \*

Enter your Card Number...

Confirm Payment

python3

# Payment Info

Ticket Fare	Net Tax	Date
4139	413	2022-10-16

Customer ID \*

345

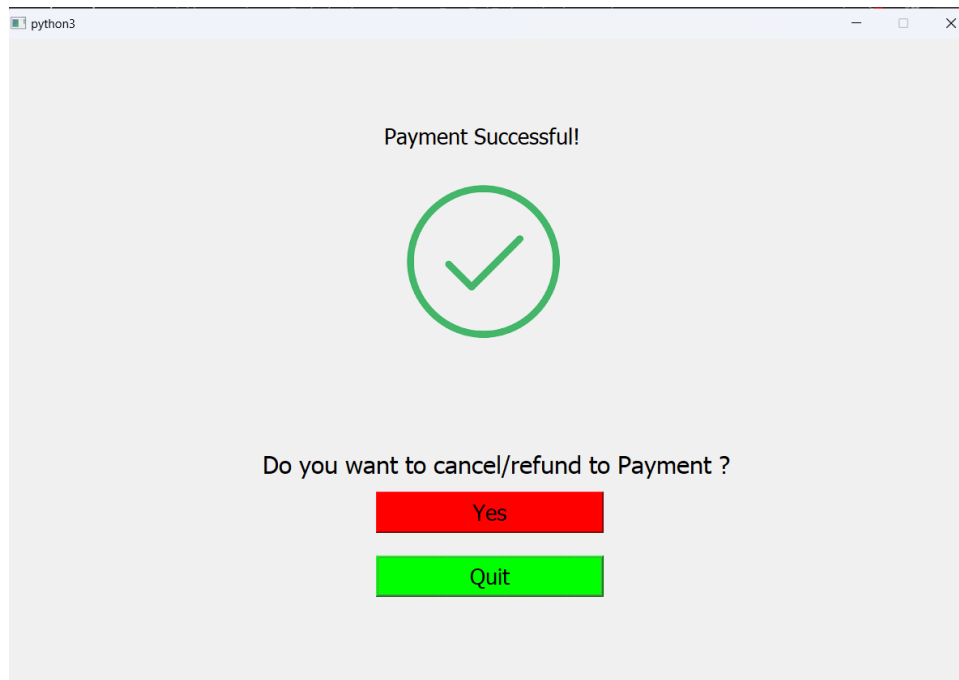
Card Type \*

Rupay

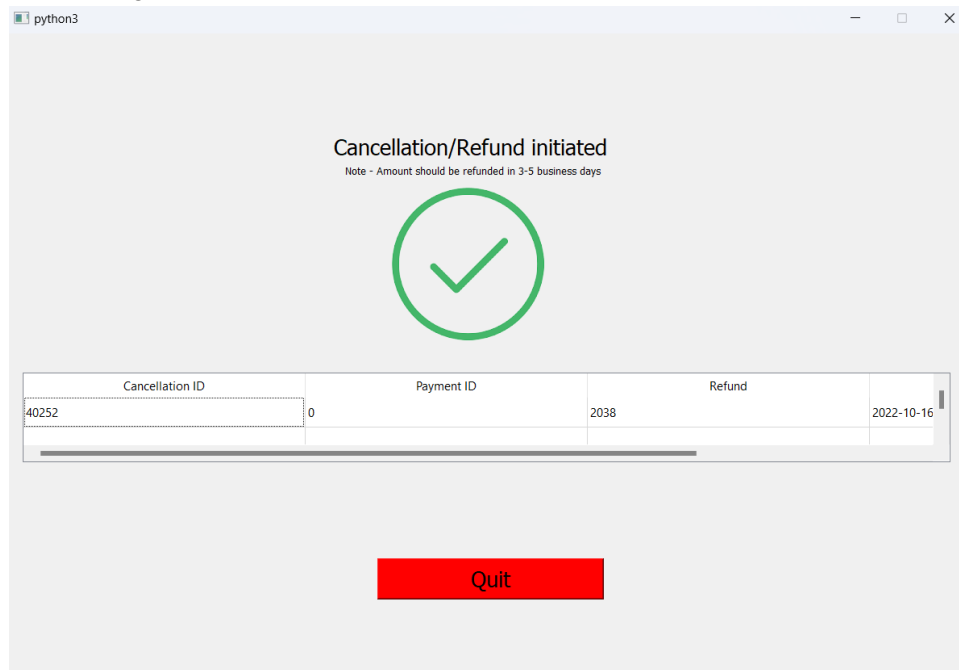
Card Number \*

5345

Confirm Payment



On clicking Yes:



Create Account User:



python3

# Create an Account

Username/Email ID \*

Password \*

Confirm your Password \*

Next

python3

# Create an Account

Username/Email ID \*

Password \*

Confirm your Password \*

Next

python3

## Profile Details

Customer Name\*

Pincode\*

State\*

Gender\*

Country\*

Date Of Birth\*

Phone Number\*

OTP Confirmation:

python3

## OTP Confirmation

Phone Number\*

OTP\*

python3

# OTP Confirmation

Phone Number\*

Send OTP

OTP\*

Check OTP

Next

Success! Account Created!

On Clicking Next, it goes to Flight Booking Ticket System Screen (Screenshot 3).

## **VII. Learning from the Project**

Include learning from the project:

- How this project helped you?

This project helped us in understanding the Ticket Management/ Booking system and process on the whole. The integration of front end (GUI) and backend (Database) along with the efficient retrieval and modification of data according to the wishes of the user. This project has made our concepts of database management systems clearer as we have understood various concepts such as normalization, weak entities, SQL commands, relationship cardinalities, attributes constraints, etc. We also came across various new errors and problems that we had not witnessed in our weekly labs and the resolution of these errors have helped us get a better grasp of MySQL.

- What new aspects did you learn?

Key takeaways and learning from the project include learning the PyQt5 + Qt Designer interface which helped in building seamless UI screens (screenshots of respective screens shown above).

Integrating front end with backend, extracting data from the user through GUI (text boxes, interactive buttons) and merging them to the respective relations in the database.

## **VIII. Challenges Faced**

Challenges faced include -

- Integrating Front end with Backend.
- OTP verification used to fail.
- Data loss/modification while merging to backend.
- MySQL Connector library connection issues.
- Some relations used to fetch incorrect tuples.
- Adding certain values resulted in unexpected program crashes due to Primary Key error.
- Normalization of tables.
- Certain buttons weren't working, labels in GUI weren't displayed in the respective screens.
- Importing the appropriate libraries for the project.

## **IX. Conclusion**

- What are the key takeaways from the project?

Implemented Flight Ticket Booking system which helps a customer in booking flights as per their requirement.

Learnt integration of front end (PyQt5) with backend (Database); enhanced coding in Python; improvised and implemented concepts learnt in the DBMS course in the project which led to better understanding of the queries in MySQL.

Understood the importance of various applications that helps us in our day-to-day life and the working at front end as well as the back end of these applications.

