CS355- DBMS Mini Project

Team Members:

- Ashutosh Kumar - 2001CS11
- Kaushal Raj - 2001CS36
- Kunwar Abhay Rai - 2001CS40
- Vipul Kumar Gond - 2001CS77

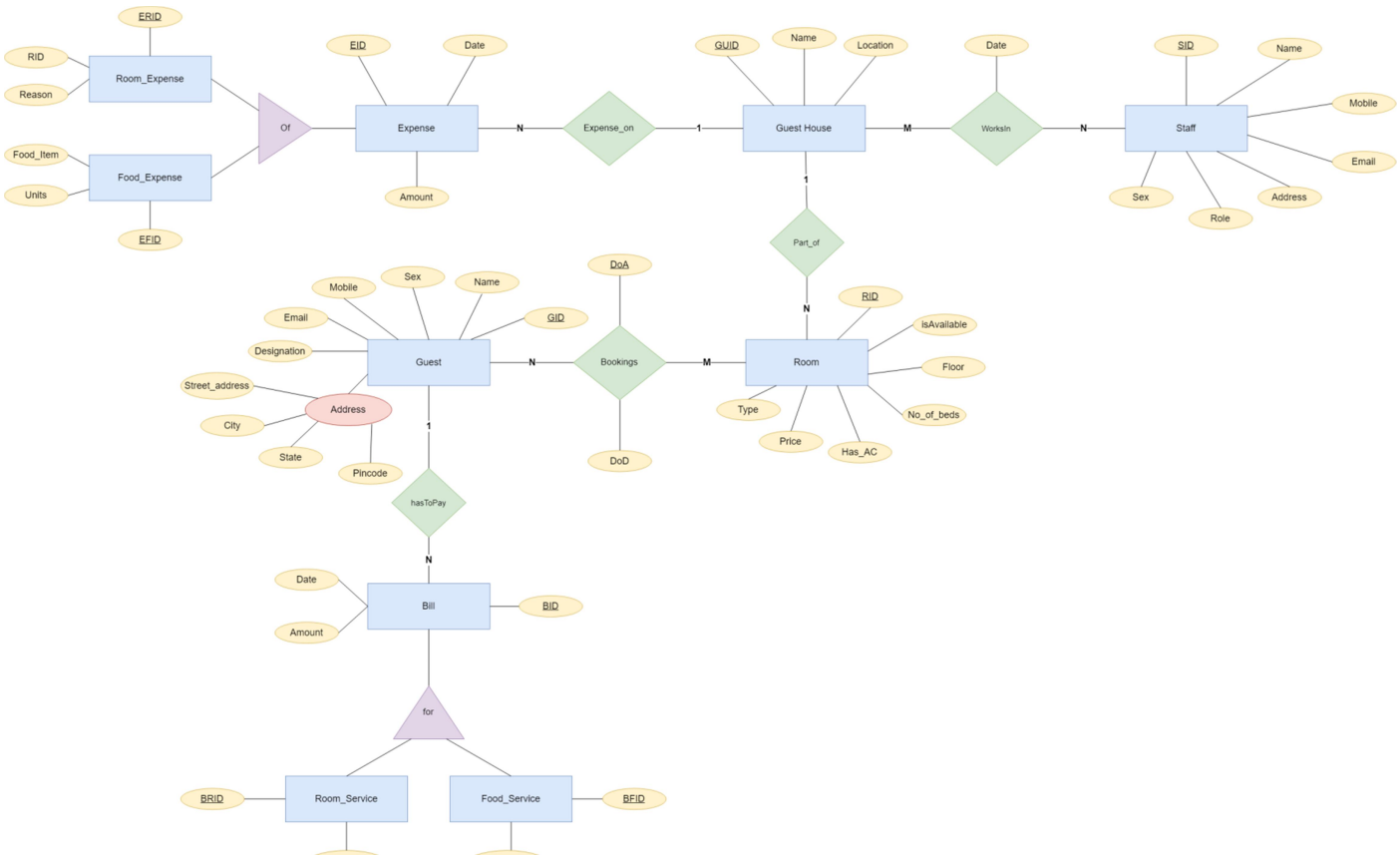
Goal: The goal of this mini project is to provide a realistic experience in the conceptual design, logical design, implementation, operation and maintenance of a small relational database.

Types of Services in our Application:

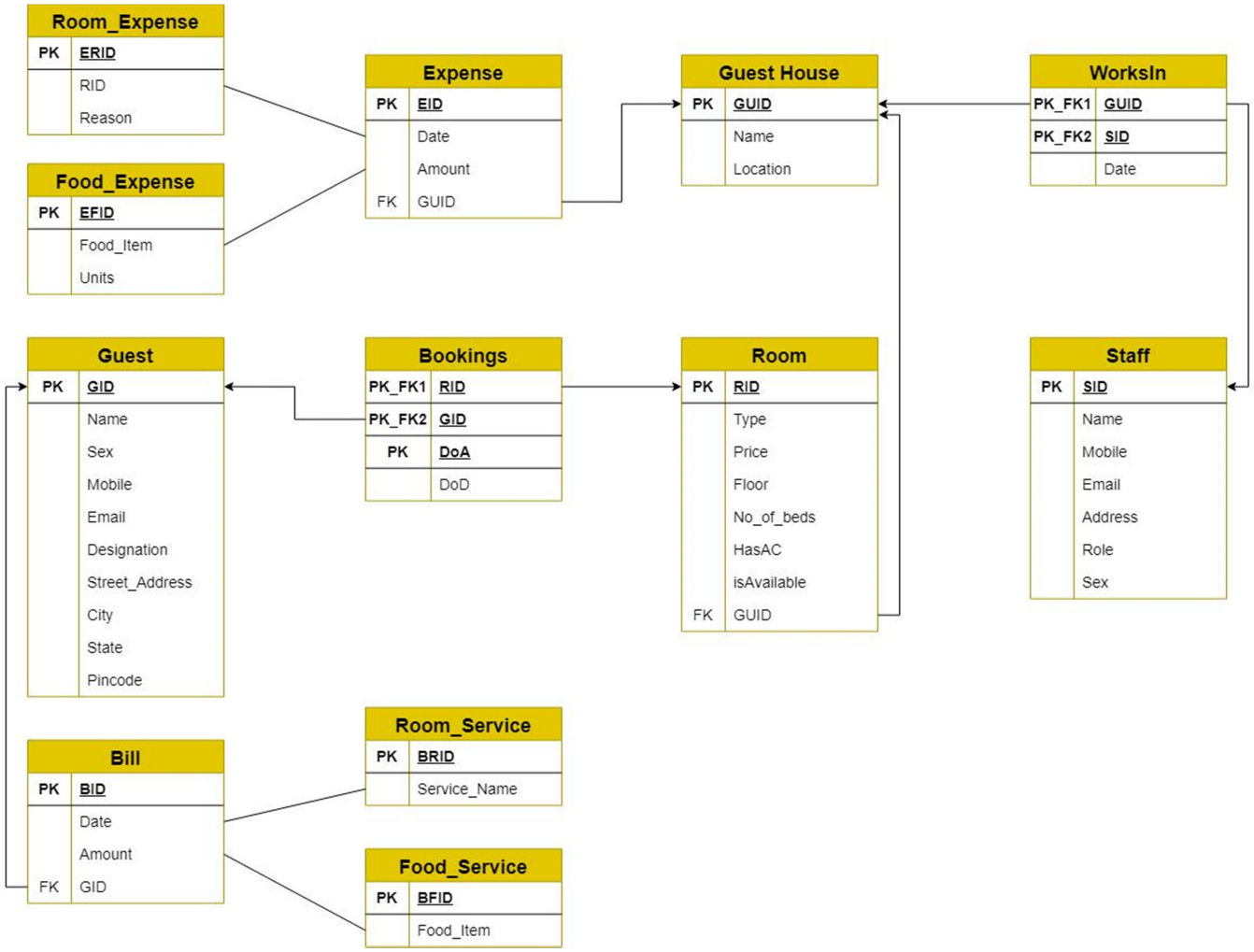
- Guest House
- Market Shop

Guest House

ER Diagram:



Relation Schema:



Description of database structure:

This database is made for guest house related services. It consists of total 10 entities:

- Bill
- Expense
- Food_Expense
- Food_service
- Guest
- Guest_house
- Room
- Room_expense
- Room_service
- Staff

It also consists of 5 relations:

- Expense_on: Connects expense and guest_house entities.
- WorksIn: Connects guest_house and staff entities.
- Part_of: Connects guest_house and room entities.
- Bookings: Connects guest and room entities.
- HasToPay: Connects guest and bill entities.

Entity 1: Bill

Attribute	Description	Type	Key
BID	Bill ID	int	PRI
Date	Date of Billing	date	-
Amount	Bill Amount	int	-
GID	Guest ID	int	MUL

Entity 2: Expense

Attribute	Description	Type	Key
EID	Expense ID	int	PRI
Date	Expense Date	date	-
Amount	Expense Amount	int	-
GUID	Guest House ID	int	MUL

Entity 3: Food_Expense

Attribute	Description	Type	Key
EFID	Expense Food ID	int	PRI
Food_item	Food Item	varchar(20)	-
Units	No. of units of Food	smallint	-

Entity 4: Food_Service

Attribute	Description	Type	Key
BFID	Bill Food ID	int	PRI
Food_item	Food Item Billed	varchar(20)	-

Entity 5: Guest

Attribute	Description	Type	Key
GID	Guest ID	int (auto increment)	PRI
Name	Guest Name	varchar(20)	-
Sex	Guest Gender	char(1)	-
Mobile	Guest Mobile No.	bigint	-
Email	Guest Email	varchar(20)	-
Designation	Guest Designation	varchar(20)	-
Street_Address	Guest Address	varchar(50)	-
City	Guest City	varchar(20)	-
State	Guest State	varchar(20)	-
Pincode	Guest Pincode	int	-

Entity 6: Guest_House

Attribute	Description	Type	Key
GUID	Guest House ID	int (auto increment)	PRI
Name	Guest House Name	varchar(20)	-
Location	Guest House Location	varchar(20)	-

Entity 7: Room

Attribute	Description	Type	Key
RID	Room ID	Int (auto increment)	PRI
Type	Room Type	Varchar(20)	-
Price	Room Pricing	Int	-
Floor	Floor no. of Room	smallint	-
No_of_beds	No. of beds in room	smallint	-
hasAC	Does room has AC	char(1)	-
isAvailable	Is room available	char(1)	-
GUID	Guest House ID	int	MUL

Entity 8: Room_Expense

Attribute	Description	Type	Key
ERID	Expense Room ID	int	PRI
RID	Room ID	int	-
Reason	Reason for expense	varchar(20)	-

Entity 9: Room_Service

Attribute	Description	Type	Key
BRID	Bill Room ID	int	PRI
Service_Name	Service for which billing is done	varchar(20)	-

Entity 10: Staff

Attribute	Description	Type	Key
SID	Staff ID	int (auto increment)	PRI
Name	Staff Name	varchar(20)	-
Sex	Staff Gender	char(1)	-
Mobile	Staff Mobile No.	bigint	-
Email	Staff Email	varchar(20)	-
Address	Staff Address	varchar(20)	-
Role	Staff Role	varchar(20)	

Relation 1: Bookings

Attribute	Description	Type	Key
RID	Room ID	int	PRI
GID	Guest ID	int	PRI
DoA	Date of Arrival	date	PRI
DoD	Date of Departure	date	-

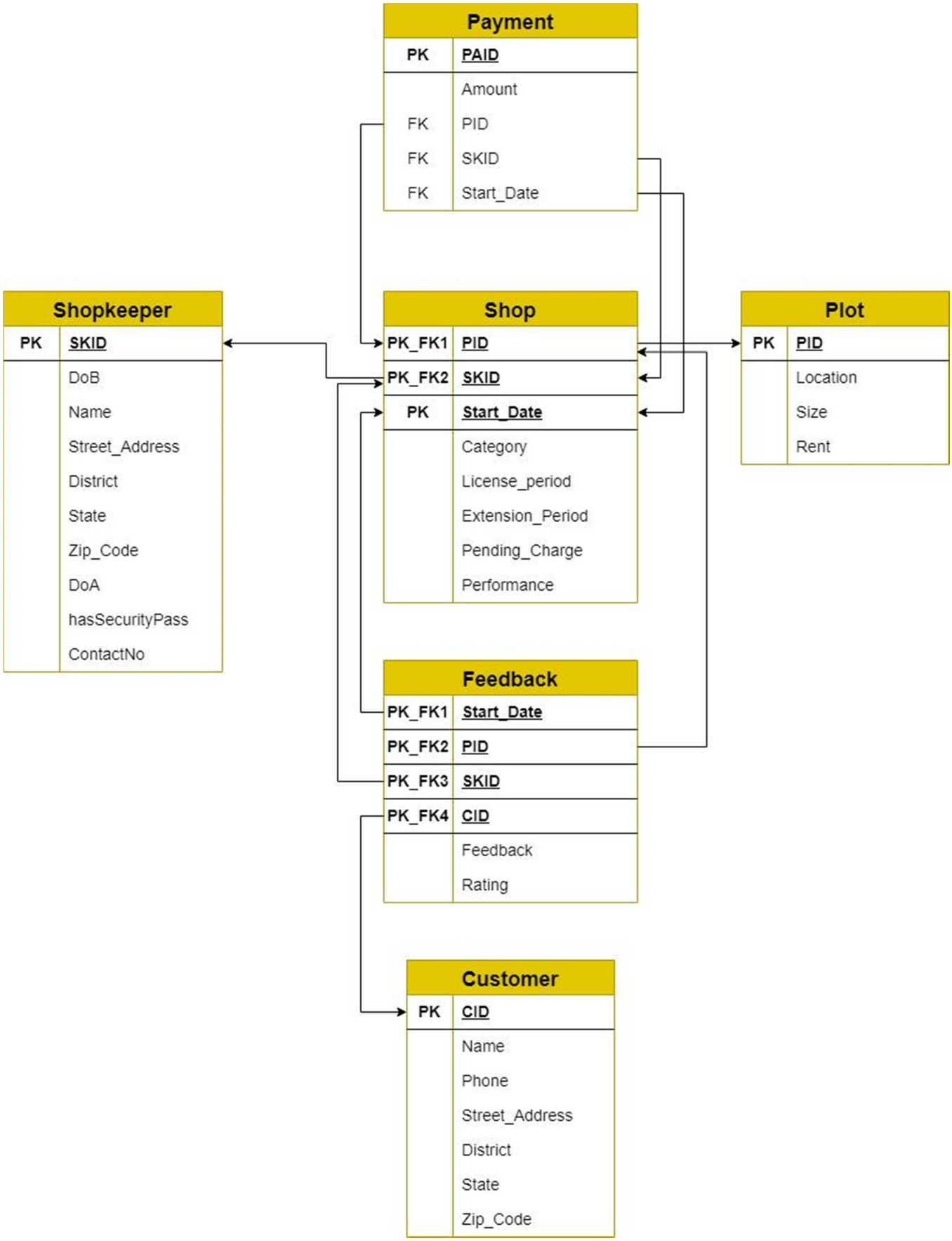
Relation 2: WorksIn

Attribute	Description	Type	Key
GUID	Guest House ID	int	PRI
SID	Staff ID	int	PRI
Date	Worked On Date	date	PRI

Market Shop:

ER Diagram

Relation Schema



Description of database structure:

This database is made for market shop related services. It consists of total 5 entities:

- Payment
- Shopkeeper
- Shop
- Customer
- Plot

It also consists of 4 relations:

- Payment_for: Connects payment and shop entities.
- Located_on: Connects shop and plot entities.
- Owner: Connects shop and shopkeeper entities.
- Feedback: Connects shop and customer entities.

Entity 1: Customer

Attribute	Description	Type	Key
CID	Customer ID	int (auto increment)	PRI
Name	Customer Name	varchar(20)	-
Phone	Customer Phone No.	bigint	-
Street_Address	Customer Address	varchar(50)	-
District	Customer District	varchar(20)	-
State	Customer State	varchar(20)	-
Zip_Code	Customer ZipCode	int	-

Entity 2: Payment

Attribute	Description	Type	Key
PAID	Guest ID	int (auto increment)	PRI
Amount	Guest Name	int	-
PID	Guest Gender	int	MUL
SKID	Guest Mobile No.	int	-
Start_Date	Guest Email	date	-

Entity 3: Plot

Attribute	Description	Type	Key
PID	Plot ID	int (auto increment)	PRI
Location	Plot Location	varchar(20)	-
Size	Plot Size	float	-
Rent	Plot Rent Amount	Int	-

Entity 4: Shop

Attribute	Description	Type	Key
PID	Plot ID	int	PRI
SKID	Shopkeeper ID	int	PRI
Start_Date	Shop Start Date	date	PRI
Category	Shop Category	varchar(20)	-
License_Period	Shop licensing period	Smallint	-
Extension_Period	Shop duration extension period	smallint	-
Pending_Charge	Pending charge from shop	int	-
Performance	Shop Performance	foat	-

Entity 5: Shopkeeper

Attribute	Description	Type	Key
SKID	Shopkeeper ID	Int (auto increment)	PRI
Name	Shopkeeper Name	varchar(20)	-
DoB	Shopkeeper Date of Birth	date	-
Street_Address	Shopkeeper Address	varchar(50)	-
District	Shopkeeper District	varchar(20)	-
State	Shopkeeper State	varchar(20)	-
Zip_Code	Shopkeeper Zipcode	int	-
DoA	Date of Application of Shopkeeper	date	-
hasSecurityPass	Does Shopkeeper has security pass	char(1)	-
ContactNo	Shopkeeper Contact No.	bigint	-

Relation 1: Feedback

Attribute	Description	Type	Key
Start_Date	Shop Start Date	date	PRI
PID	Plot ID	int	PRI
SKID	Shopkeeper ID	int	PRI
CID	Customer ID	int	PRI
Feedback	Feedback by customer on shop	text	-
Rating	Rating by customer to shop	smallint	-

Queries to create various tables:

Database: **Guest House**

```sql
CREATE DATABASE guest_house;

USE guest_house;
CREATE TABLE Room_Expense
(
  ERID INT NOT NULL,
  RID INT NOT NULL,
  Reason VARCHAR(20),
  PRIMARY KEY (ERID)
);

CREATE TABLE Food_Expense
(
  EFID INT NOT NULL,
  Food_item VARCHAR(20) NOT NULL,
  Units SMALLINT NOT NULL,
  PRIMARY KEY (EFID)
);


CREATE TABLE Guest
(
  GID INT NOT NULL AUTO_INCREMENT,
  Name VARCHAR(20) NOT NULL,
  Sex CHAR,
  Mobile BIGINT,
  Email VARCHAR(20),
  Designation VARCHAR(20),
  Street_Address VARCHAR(50),
  City VARCHAR(20),
  State VARCHAR(20),
  Pincode INT,
  PRIMARY KEY (GID)
);
```

```sql
CREATE TABLE Bill
(
  BID INT NOT NULL,
  Date DATE,
  Amount INT NOT NULL,
  GID INT NOT NULL,
  PRIMARY KEY (BID),
  CONSTRAINT Bill_fk FOREIGN KEY (GID) REFERENCES Guest(GID)
);

CREATE TABLE Guest_House
(
  GUID INT NOT NULL auto_increment,
  Name VARCHAR(20) NOT NULL,
  Location VARCHAR(20) NOT NULL,
  PRIMARY KEY (GUID)
);

CREATE TABLE Expense
(
  EID INT NOT NULL,
  Date DATE,
  Amount INT NOT NULL,
  GUID INT NOT NULL,
  PRIMARY KEY (EID),CONSTRAINT Expense_fk FOREIGN KEY(GUID) REFERENCES Guest_House(GUID)
);

CREATE TABLE Room
(
  RID INT NOT NULL auto_increment,
  Type VARCHAR(20),
  Price INT,
  Floor SMALLINT,
  No_of_beds SMALLINT,
  hasAC CHAR,
  isAvailable CHAR,
  GUID INT NOT NULL,
  PRIMARY KEY (RID),
  CONSTRAINT Room_fk FOREIGN KEY(GUID) REFERENCES Guest_House(GUID)
);

CREATE TABLE Bookings
(
  RID INT NOT NULL,
  GID INT NOT NULL,
  DoA DATE NOT NULL,
  DoD DATE NOT NULL,
  PRIMARY KEY (RID, GID, DoA),
  CONSTRAINT Bookings_fk1 FOREIGN KEY (RID) REFERENCES Room(RID), CONSTRAINT Bookings_fk2 FOREIGN KEY (GID) REFERENCES Guest(GID)
);

CREATE TABLE Room_Service
(
  BRID INT NOT NULL,
  Service_Name VARCHAR(20) NOT NULL,
  PRIMARY KEY (BRID)
);
```

```sql
CREATE TABLE Food_Service
(
  BFID INT NOT NULL,
  Food_item VARCHAR(20) NOT NULL,
  PRIMARY KEY (BFID)
);

CREATE TABLE Staff
(
  SID INT NOT NULL AUTO_INCREMENT,
  Name VARCHAR(20) NOT NULL,
  Sex CHAR,
  Mobile BIGINT,
  Email VARCHAR(20),
  Address VARCHAR(20),
  Role VARCHAR(20),
  PRIMARY KEY (SID)
);

CREATE TABLE WorksIn(
  GUID INT NOT NULL,
  SID INT NOT NULL,
  Date DATE NOT NULL,
  PRIMARY KEY (GUID, SID, Date),
  CONSTRAINT WorksIn_fk1 FOREIGN KEY (GUID) REFERENCES Guest_House(GUID), CONSTRAINT WorksIn_fk2 FOREIGN KEY (SID) REFERENCES Staff(SID)
);
```

Database: **Market**

```sql
CREATE DATABASE market;
USE market;

CREATE TABLE Shopkeeper
(
  SKID INT NOT NULL AUTO_INCREMENT,
  Name VARCHAR(20) NOT NULL,
  DoB DATE,
  Street_Address VARCHAR(50),
  District VARCHAR(20),
  State VARCHAR(20),
  Zip_Code INT,
  DoA DATE NOT NULL,
  hasSecurityPass CHAR(1),
  ContactNo BIGINT,
  PRIMARY KEY (SKID)
);


CREATE TABLE Plot
(
  PID INT NOT NULL AUTO_INCREMENT,
  Location VARCHAR(20),
  Size FLOAT,
  Rent INT,
  PRIMARY KEY (PID)
);


CREATE TABLE Shop
(
  PID INT NOT NULL,
  SKID INT NOT NULL,
  Start_Date DATE NOT NULL,
  Category VARCHAR(20),
  License_Period SMALLINT,
  Extension_Period SMALLINT,
  Pending_Charge INT,
  Performance FLOAT,
  PRIMARY KEY (PID, SKID, Start_Date),
  CONSTRAINT Shop_fk1 FOREIGN KEY (PID) REFERENCES Plot(PID), CONSTRAINT Shop_fk2 FOREIGN KEY (SKID) REFERENCES Shopkeeper(SKID)
);
```

```sql
CREATE TABLE Customer
(
  CID INT NOT NULL AUTO_INCREMENT,
  Name VARCHAR(20) NOT NULL,
  Phone BIGINT,
  Street_Address VARCHAR(50),
  District VARCHAR(20),
  State VARCHAR(20),
  Zip_Code INT,
  PRIMARY KEY (CID)
);


CREATE TABLE Feedback(
  Start_Date DATE NOT NULL,
  PID INT NOT NULL,
  SKID INT NOT NULL,
  CID INT NOT NULL,
  Feedback TEXT,
  Rating SMALLINT,
  PRIMARY KEY (Start_Date, PID, SKID, CID),
  CONSTRAINT Feedback_fk1 FOREIGN KEY (CID) REFERENCES Customer(CID), CONSTRAINT Feedback_fk2 FOREIGN KEY (PID, SKID, Start_Date) REFERENCES
  Shop(PID, SKID, Start_Date)
);


CREATE TABLE Payment
(
  PAID INT NOT NULL auto_increment,
  Amount INT NOT NULL,
  PID INT NOT NULL,
  SKID INT NOT NULL,
  Start_Date DATE NOT NULL,
  PRIMARY KEY (PAID),
  CONSTRAINT Payment_fk1 FOREIGN KEY (PID, SKID, Start_Date) REFERENCES
  Shop(PID, SKID, Start_Date)
);
```

Triggers

1. **after_bookings_insert:** After trigger for marking room as booked on booking a room
   Database: Guest House

```
DELIMITER $$
CREATE TRIGGER after_bookings_insert
 AFTER INSERT
    ON Bookings FOR EACH ROW
    BEGIN
UPDATE Room SET isAvailable = 'F' WHERE Room.RID = new.RID;
  END $$
DELIMITER ;

```

This trigger is used to auto-update the room availability once someone books a room. The “isAvailable” attribute is changed to ‘F ’once a room is successfully booked.

2. **after_feedback_insert:** After trigger to calculate performance from rating
   Database: Market Shop

```
DELIMITER $$
 CREATE TRIGGER after_feedback_insert
 AFTER INSERT
 ON Feedback FOR EACH ROW
 BEGIN
DECLARE avg_rating FLOAT DEFAULT 0;
SELECT SUM(Rating)/COUNT(*) INTO avg_rating
FROM Feedback
WHERE Feedback.SKID = new.SKID
AND Feedback.PID = new.PID
AND Feedback.Start_Date = new.Start_Date;
UPDATE Shop SET Performance = avg_rating WHERE Shop.SKID = new.SKID
AND Shop.PID = new.PID AND Shop.Start_Date = new.Start_Date;
END $$
DELIMITER ;

```

This trigger is used to automatically update the shop performance once a new feedback is inserted. Performance is simply the average rating of the shop.

3. **after_payment_insert:** After trigger to update pending charge
   Database: Market Shop

```
DELIMITER $$
CREATE TRIGGER after_payment_insert
  AFTER INSERT
  ON Payment FOR EACH ROW
  BEGIN
    UPDATE Shop
    SET Pending_Charge = Pending_Charge - new.Amount
    WHERE Shop.SKID = new.SKID
    AND Shop.PID = new.PID
    AND Shop.Start_Date = new.Start_Date;
  END $$
DELIMITER ;
```

This trigger is used to automatically update the pending charge of a shop. Once a payment has been made by the shopkeeper, the same amount is deducted from the pending charge stored in the database to update the pending charge.

Scheduled Event

1. **update_room_availability:** Scheduled event to update room availability status
   Database: Guest House

```
CREATE EVENT update_room_availability
  ON SCHEDULE EVERY 1 HOUR
  STARTS CURRENT_TIMESTAMP
  DO
  UPDATE Room
  SET isAvailable = 'T' WHERE Room.RID NOT IN(
  SELECT DISTINCT(Bookings.RID) FROM Bookings WHERE Bookings.DoD > CURDATE()
);
```

This event is scheduled to run at an interval of one hour, indefinitely. If for a booked room, the date of departure is greater than the current date, the room is marked as available for new booking.

Procedures

1. **generate_bill :** This procedure is used to generate bills for a guest. It takes the guest ID as input and gives the Bill ID, Guest ID, Billing Date and the Bill Amount for the guest.

Database: Guest House

```
DELIMITER $$
  CREATE PROCEDURE generate_bill(IN iGID INT)
  BEGIN
  SELECT BID, GID, Date, Amount FROM Bill WHERE GID = iGID;
  END $$
DELIMITER ;
```

2. **monthly_expenditure :** This procedure gives the total expenditure in the given month of a year for the guest houses. It takes yearMonth as input and gives the Guest House ID and total monthly expenditure for that guest house.

Database: Guest House

```
DELIMITER $$
  CREATE PROCEDURE monthly_expenditure(IN iYear_Month INT)
  BEGIN
SELECT GUID, CONCAT(EXTRACT(YEAR FROM Date),"-",EXTRACT(MONTH FROM Date))    as YM, SUM(Amount) AS Total_Expenditure

FROM Expense WHERE CONCAT(EXTRACT(YEAR_MONTH FROM Date),"") = CONCAT(iYear_Month,"") GROUP BY GUID;
END $$
DELIMITER ;
```

Total Entries in various tables

Table Name	Total_Entries
Bill	20
Bookings	10
Expense	16
Food_Expense	8
Food_Service	10
Guest	20
Guest_House	4
Room	30
Room_Expense	8
Room_Service	10
Staff	30
WorksIn	30
Customer	15
Feedback	2
Payment	2
Plot	15
Shop	15
Shopkeeper	15

Sample Queries

Guest House:

1. **Monthly Bookings for guest houses in different categories:** Returns all the booking information (guest information, room information and booking details) of a particular month of a year.

```
SELECT Guest.GID, Guest.Name, Room.RID, Room.Type, EXTRACT(YEAR_MONTH FROM Bookings.DoA) as Booking_Month, Bookings.DoD
FROM Guest INNER JOIN Bookings ON Bookings.GID = Guest.GID INNER JOIN Room ON Room.RID = Bookings.RID
GROUP BY Booking_month;
```

2. **The total monthly expenditure for the guest house:** Uses a procedure call to find the expenditure on guest houses of a particular month of a year. The procedure takes year and month as input.

```
CREATE PROCEDURE monthly_expenditure(IN iYear_Month INT)
BEGIN
SELECT GUID, CONCAT(EXTRACT(YEAR FROM Date),"-",EXTRACT(MONTH FROM Date)) as YM, SUM(Amount) AS Total_Expenditure
FROM Expense
WHERE CONCAT(EXTRACT(YEAR_MONTH FROM Date),"") = CONCAT(iYear_Month,"") GROUP BY GUID;
END $$
DELIMITER ;
CALL monthly_expenditure($query_yearMonth);
```

3. **Generation of bills:** Uses a procedure call to generate bills for the guests. The procedure takes a guest ID to generate bills.

```
DELIMITER $$
CREATE PROCEDURE generate_bill(IN iGID INT)
BEGIN
  SELECT BID, GID, Date, Amount FROM Bill WHERE GID = iGID;
END $$
DELIMITER ;
CALL generate_bill('$query_sagid');
```

4. **Availability of rooms:** Shows the details of all the available rooms.

```sql
SELECT * FROM Room WHERE isAvailable = 'T';
```

5. **Monthly food billing:** Shows the food billing information (Billing information and food information) for all possible months and years.

```sql

SELECT Bill.BID, Bill.GID, EXTRACT(YEAR_MONTH FROM Bill.Date) AS Food_Billing_Month, Bill.Amount, Food_Service.Food_Item
FROM Bill INNER JOIN Food_Service ON Bill.BID = Food_Service.BFID GROUP BY Food_Billing_Month;
```

6. **Staff schedule:** Shows the staff work schedule for the guest houses.

```sql
SELECT Staff.SID, Staff.Name, Guest_House.GUID, Guest_House.Name, WorksIn.Date
FROM Guest_House
INNER JOIN WorksIn ON Guest_House.GUID = WorksIn.GUID
INNER JOIN Staff ON Staff.SID = WorksIn.SID
ORDER BY WorksIn.Date DESC;
```

Market Shop:

1. **Current shop details of different areas of the campus:** Shows the shop's details (Shopkeeper details, shop details and plot details) of all shops.

```
SELECT Shopkeeper.SKID, Shopkeeper.Name,
Plot.PID, Plot.Location, Plot.Rent,
Shop.Category, Shop.Start_Date, Shop.License_Period, Shop.Extension_Period, Shop.Pending_Charge, Shop.Performance
FROM Shop
INNER JOIN Shopkeeper ON Shopkeeper.SKID = Shop.SKID INNER JOIN Plot ON Plot.PID = Shop.PID;
```

2. **Details of shopkeepers and their security pass validity:** Shows the Shopkeeper details such as their name, address and security pass.

```
SELECT * FROM Shopkeeper;
```

3. **Reminders for expiring license agreement period:** Shows all the shops whose licenses are about to expire (within 30 days) or have already expired.

```
SELECT SKID, PID, Start_Date, DATEDIFF(DATE_ADD(Start_Date,INTERVAL
(License_Period+Extension_Period) MONTH), CURDATE()) as Diff FROM Shopc WHERE DATEDIFF(DATE_ADD(Start_Date,INTERVAL (License_Period+Extension_Period)
MONTH), CURDATE()) <= 30;
```

4. **Pending charges from each shop:** Shows the shop information and pending charge for the shops.

```
SELECT SKID, PID, Start_Date, Pending_Charge FROM Shop;
```

5. **Summary of performances of each shop:** Shows the shop information and shop performance for each shop.

```sql
SELECT SKID, PID, Start_Date, Performance FROM Shop;
```

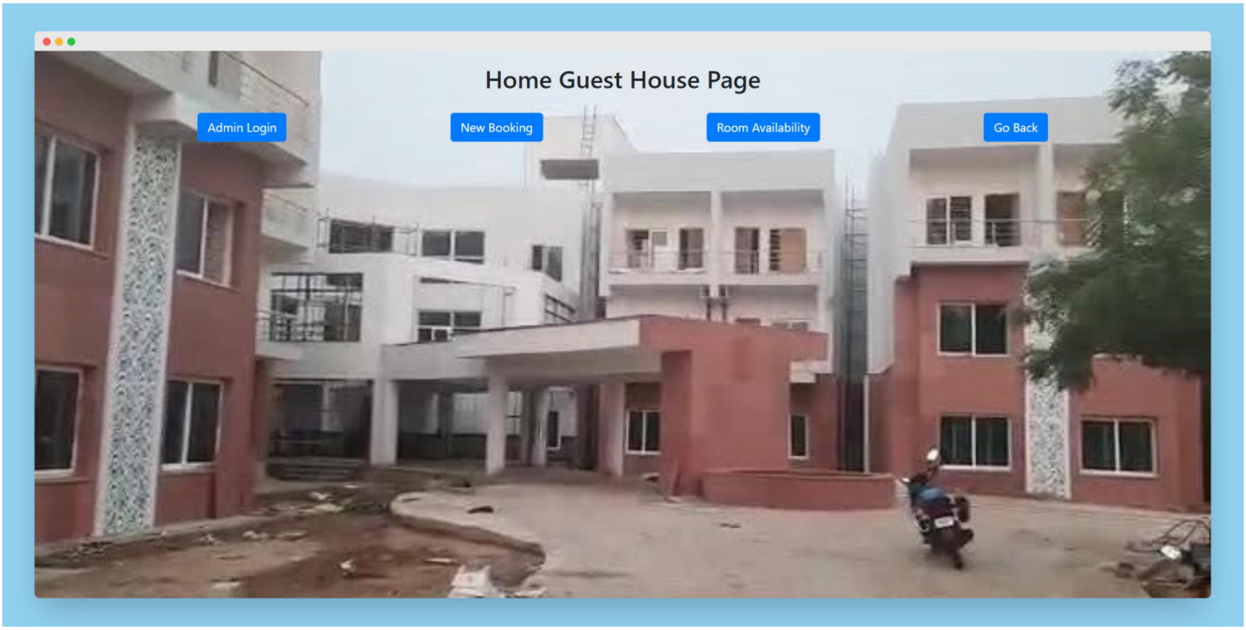
6. **Available plots to open shop:** Shows all the plots which are currently free to open shop.

```sql
SELECT * FROM Plot WHERE PID NOT IN (
  SELECT PID FROM Shop
  WHERE DATE_ADD(Start_Date,INTERVAL (License_Period+Extension_Period)
  MONTH) > CURDATE()
);
```

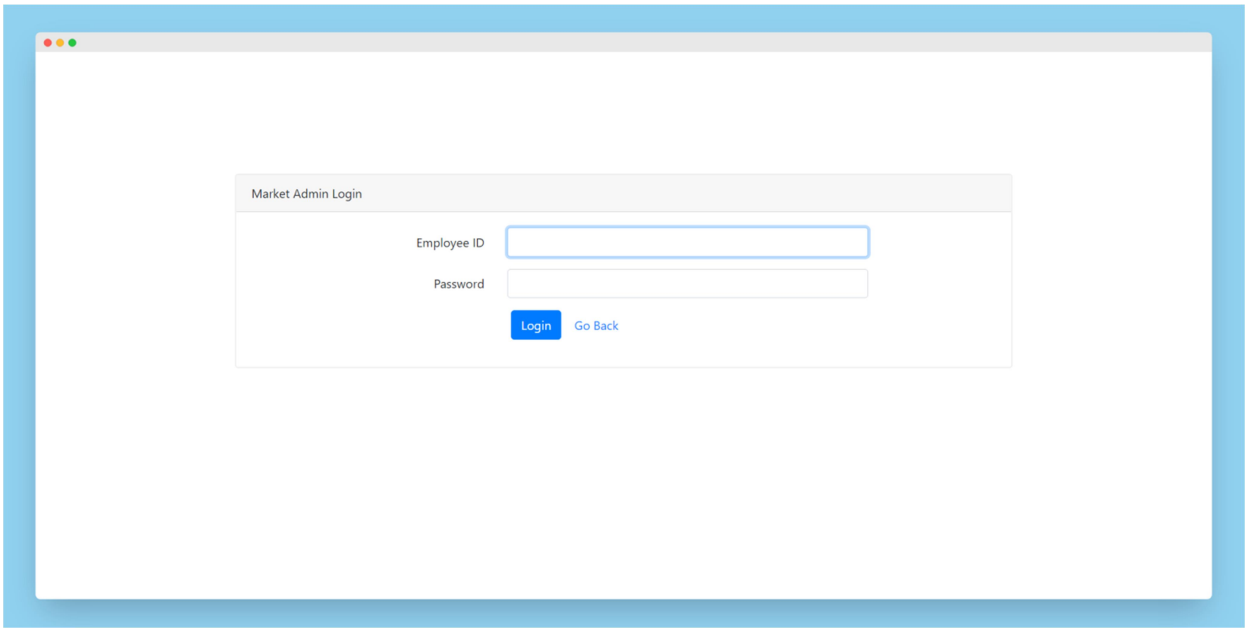
Web Page Screenshots

Home Page:

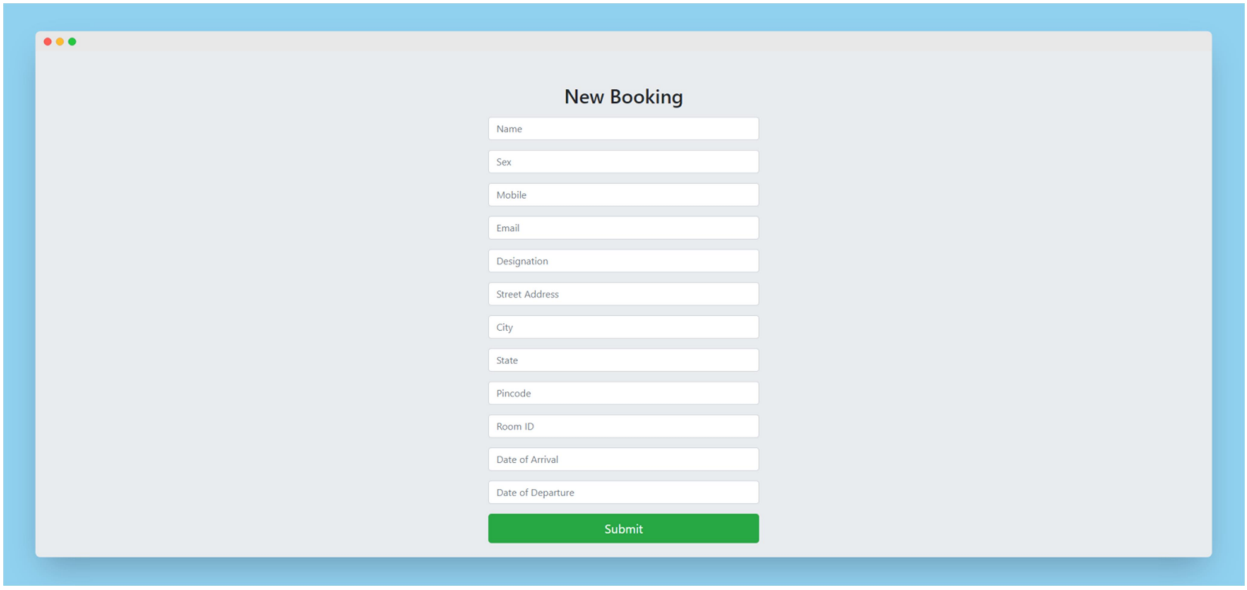
Guest House Menu:



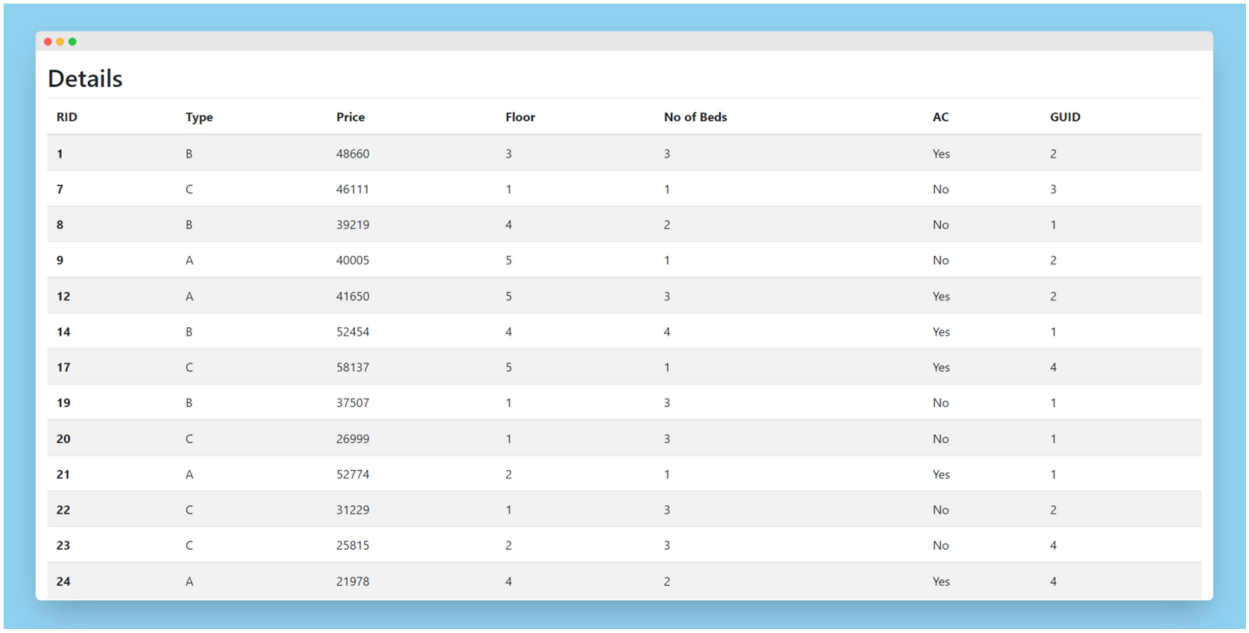
Admin Login:



New Booking:



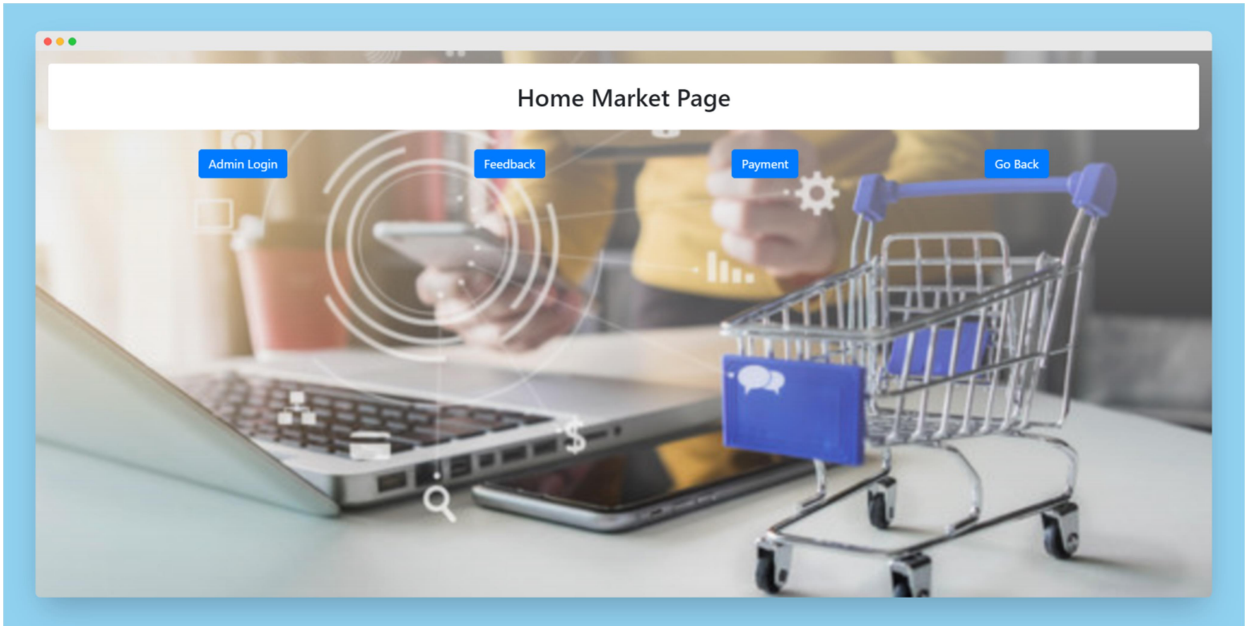
Room Availability



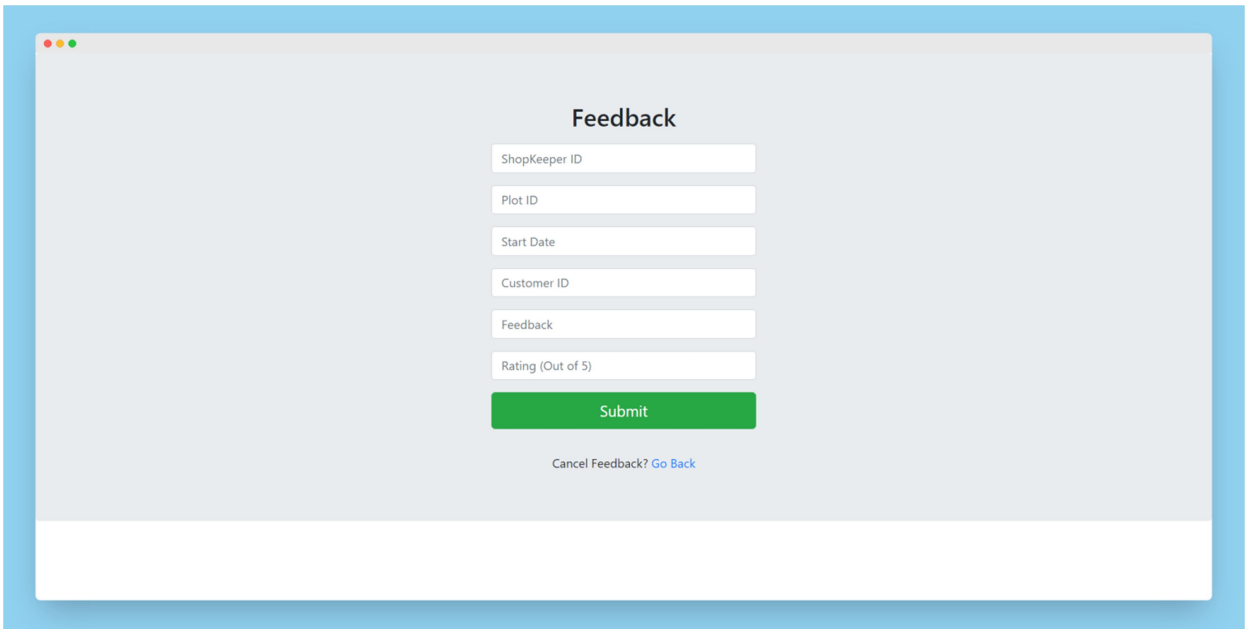
Details

RID	Type	Price	Floor	No of Beds	AC	GUID
1	B	48660	3	3	Yes	2
7	C	46111	1	1	No	3
8	B	39219	4	2	No	1
9	A	40005	5	1	No	2
12	A	41650	5	3	Yes	2
14	B	52454	4	4	Yes	1
17	C	58137	5	1	Yes	4
19	B	37507	1	3	No	1
20	C	26999	1	3	No	1
21	A	52774	2	1	Yes	1
22	C	31229	1	3	No	2
23	C	25815	2	3	No	4
24	A	21978	4	2	Yes	4

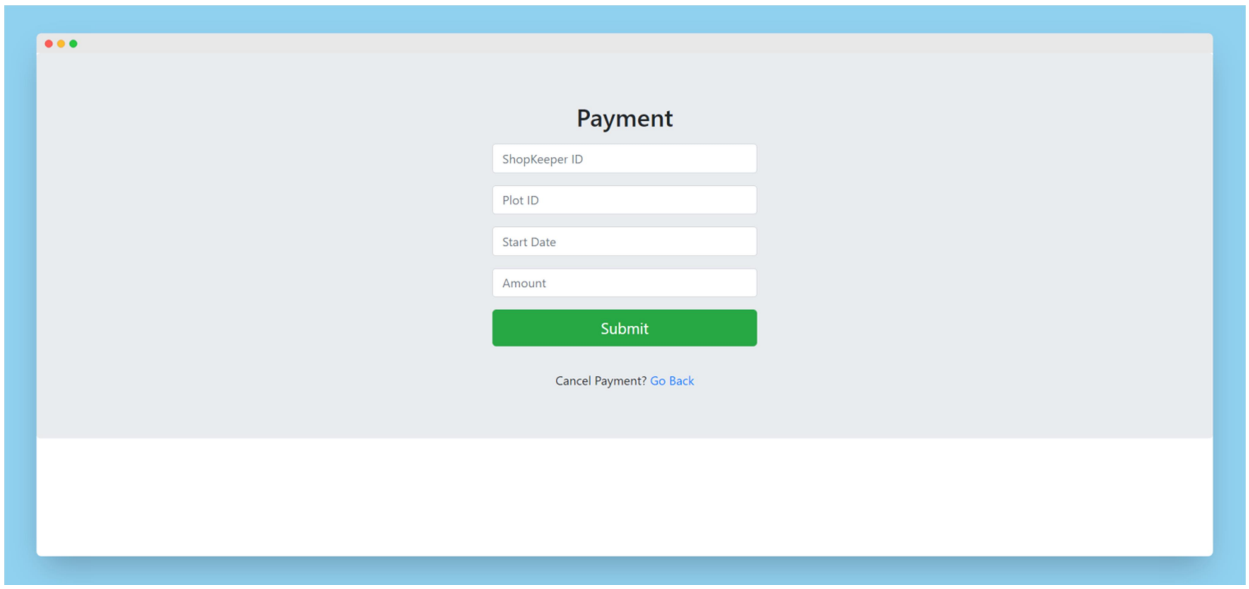
Home Market Menu:



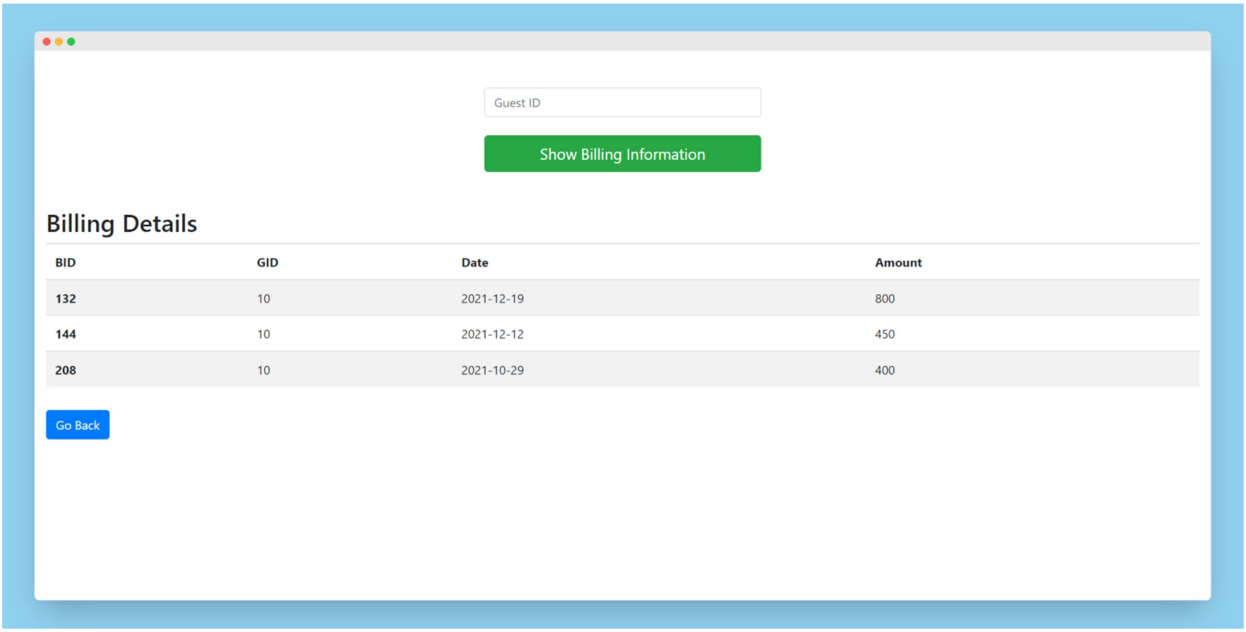
Feedback:



Payment



Generate Bill



Source Code:

https://github.com/kunwarAbhay/campus_amenities

Google Drive Link:

https://drive.google.com/drive/folders/1o3Yi_pSlr2mVM1bO9Q0Td4_RjAhBBiFZ