**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

# IST 659 Project Implementation Report

# ROOM RESERVATION MANAGEMENT SYSTEM

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

# TABLE OF CONTENTS:

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

# 1. PROJECT SUMMARY:

Our project is a database solution for Concord Housing which rents properties to customers in New York. Concord Housing has around 50 properties which it rents out to customers for a temporary period. It also provides additional facilities like food and other recreational facilities to the customers during their stay at Concord Housing. The organization is handled by the main owner of the company and caretakers of each property. The customer has to reserve a property by physically being present at the booking place with some amount of advance payment. The customer has to provide his ID proof at the time of making a reservation and also while check-in.

Currently, the administrator and the staff have to reply to email queries and accept room reservations over the phone. Moreover, a single person has to manually update the reservations coming in from all the distributed networks. Considering the fact that most of the reservation systems have a complicated distribution network with a wide range of agents, this can be an extremely difficult task. Apart from that, collection of data is an important factor to consider, especially in the hospitality sector where customers are demanding a more personalized experience. Without an online booking system, it is difficult to know in depth about the customer i.e. where the customers are coming from, what are their preferences and why are they visiting a particular destination. Collecting data about customers helps in improving the overall user experience. Another problem in using a manual system is that it is quite possible that more reservations are accepted than there is room for. Overbooking creates leads to hassles which affect the customer experience.

Room reservation management system can solve all of these issues. Firstly, it simplifies the administrative process of accepting reservations. This can help the rest of the staff to be more organized and efficient because everyone will have access to live information about the current occupancy at the hotel. All live rates and availability are automatically updated by the system. The room reservation system also reduces the risk of overbooking and increases the overall smoothness of the reservation process. Also, collecting data about customers becomes easy through an online room reservation management system. This data can be compiled into comprehensive reports that help in improving the customer experience.

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

# 2. ENTITY AND ATTRIBUTE TABLE:

## 1. Customer Table:

Stores the information of the customers of the system.

| **Entity Name: Customer** | **Attribute Name** | **Field Type** | **NULL/NOT NULL** | **Explanation** |
|---|---|---|---|---|
| Primary Key | CustomerID | VARCHAR(30) | NOT NULL | Unique identifier for the customer. |
| Other Attributes | CustomerPassword | VARCHAR(25) | NOT NULL | Password for each customerID which should be greater than 8 characters. It should have numbers, Upper case letter and lowercase letter |
| | CFName | VARCHAR(30) | NOT NULL | First Name of the customer |
| | CLName | VARCHAR(30) | NOT NULL | Last Name of the customer |
| | CPhone | CHAR(12) | NOT NULL | Customer's Phone Number |
| | CEmail | VARCHAR(50) | NOT NULL | Customer's Email id |
| | CAptNo | INT | NOT NULL | Apt No. of the Customer's address |
| | CStreetName | VARCHAR(30) | NOT NULL | Street Name of the Customer's address |
| | CCity | VARCHAR(30) | NOT NULL | City |
| | CState | CHAR(2) | NOT NULL | State |
| | CZip | CHAR(5) | NOT NULL | ZipCode |
| | IDProof | VARCHAR(15) | NOT NULL | Valid ID Proof (Driving License/ Passport) |
| | IDNumber | VARCHAR(15) | NOT NULL | ID number as per ID proof |

## 2. Property Table:

Stores the information about each of the prop

| **Entity Name: Property** | **Attribute Name** | **Field Type** | **NULL/NOT NULL** | **Explanation** |
|---|---|---|---|---|
| Primary Key | PropertyId | VARCHAR(10) | NOT NULL | Property_ID - Primary Key |
| Foreign Key | PropertyTypeId | VARCHAR(10) | NOT NULL | Property_Type_ID- Foreign Key |
| Other Attributes | AptNo | INT | NOT NULL | Apt. No. of the property |
| | PStreetName | VARCHAR(30) | NOT NULL | Street Name of the property |
| | PCity | VARCHAR(30) | NOT NULL | City |
| | PState | CHAR(2) | NOT NULL | State |
| | NoOfRooms | INT | NOT NULL | No. of rooms available in a property |
| | PropertyStatus | VARCHAR(10) | NOT NULL | Status of the property(Available/Reserved) |

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

## 3. Property Type Table:

Stores information about the types of property provided

| **Entity Name: Property_type** | **Attribute Name** | **Field Type** | **NULL/ NOT NULL** | **Explanation** |
|---|---|---|---|---|
| Primary Key | PropertyTypeId | VARCHAR(10) | NOT NULL | PropertyTypeID - Primary Key |
| Other Attributes | MaxGuestsAllowed | INT | NOT NULL | Maximum no. of guests allowed per property |
| | RatePerDay | DECIMAL | NOT NULL | Rate of the property per day |
| | PropertyType | VARCHAR(15) | NOT NULL | Type of property: Villa, apartment, bungalow, cottage |

## 4. Reservation Table:

Stores information about each of the reservations

| **Entity Name: Reservation** | **Attribute Name** | **Field Type** | **NULL/NOT NULL** | **Explanation** |
|---|---|---|---|---|
| Primary Key | ReservationID | INT | NOT NULL | ReservationID- Primary Key |
| Foreign Key | CustomerID | VARCHAR(30) | NOT NULL | CustomerID- Foreign Key |
| Foreign Key | PropertyId | VARCHAR(10) | NOT NULL | PropertyId-Foreign Key |
| Other Attributes | CancellationStatus | VARCHAR(10) | | Status of the reserved property- Reserved/Cancelled |
| | CheckinDate | DATE | NOT NULL | The date on which the customer would check-in the property |
| | CheckoutDate | DATE | NOT NULL | The date on which the customer would check-out the property |
| | ReservationDate | DATETIME | NOT NULL | The date on which the reservation is made |
| | TotalAmount | DECIMAL | NOT NULL | The total price based on the no. of days and property selected as well as the services selected |
| | NoOfGuests | INT | NOT NULL | Total No. of Guests |
| | CancellationDate | DATE | | Date on which the cancellation is made. |

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

## 5. Available Services Table:

Stores the details about the services provided by the administrator.

| **Entity Name: available_services** | **Attribute Name** | **Field Type** | **NULL/NOT NULL** | **Explanation** |
|---|---|---|---|---|
| Primary Key | ServiceID | VARCHAR(10) | NOT NULL | Service_ID- Primary Key |
| Other Attributes | ServiceName | VARCHAR(25) | NOT NULL | Services provided by the administrator (Example: Food, Clubhouse, Wi-Fi) |
| | ServiceDesc | VARCHAR(100) | NOT NULL | Description of the services in brief |
| | ServiceCost | DECIMAL | NOT NULL | Cost of the Service per day |

## 6. Reservation Services Table:

Stores the details about the services per reservation.

| **Entity Name: reservation_services** | **Attribute Name** | **Field Type** | **NULL/NOT NULL** | **Explanation** |
|---|---|---|---|---|
| Primary Key Foreign Key | ServiceID | VARCHAR(10) | NOT NULL | Service_ID- Primary Key & Foreign Key |
| Primary Key Foreign Key | ReservationID | INT | NOT NULL | Reservation_ID- Primary Key & Foreign Key |
| Other Attributes | ServiceDate | DATE | NOT NULL | Dates for which the service was requested |

## 7. Payment Method Table:

Stores information about the payment method of the customer for each reservation.

| **EntityName: Payment_method** | **Attribute Name** | **Field Type** | **NULL/NOT NULL** | **Explanation** |
|---|---|---|---|---|
| Primary Key | PaymentMethodID | INT | NOT NULL | PaymentMethodId- Primary Key |
| Foreign Key | ReservationID | INT | NOT NULL | Reservation_ID- Foreign Key |
| Other Attributes | CardNumber | Numeric(16) | NOT NULL | Details of the card with which the payment is being made i.e. CardNumber |
| | ExpiryDate | DATE | NOT NULL | The duration for which the card is valid |
| | NameonCard | VARCHAR(30) | NOT NULL | Card Holder's Name |
| | CVV | Numeric(16) | NOT NULL | Card Security Code |

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

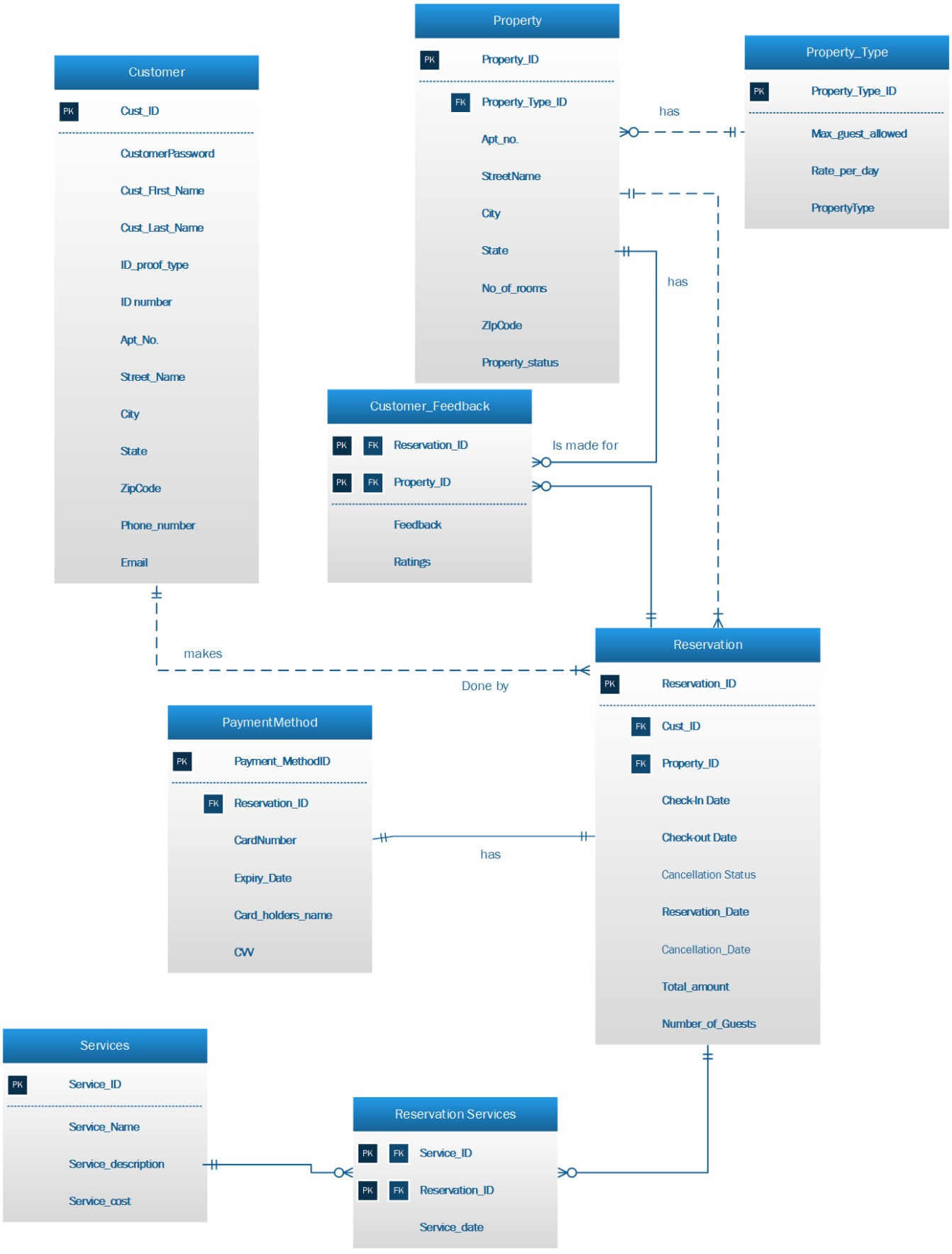
## 8. Customer Feedback Table:

Stores feedback of the property with respect to each reservation.

| **Entity Name: Participant** | **Attribute Name** | **Field Type** | **NULL/NOT NULL** | **Explanation** |
|---|---|---|---|---|
| Primary Key Foreign Key | PropertyID | VARCHAR(10) | NOT NULL | Unique identifier for property |
| Primary Key Foreign Key | ReservationID | INT | NOT NULL | Unique identifier for Reservation |
| Other Attributes | Feedback | VARCHAR(100) | NOT NULL | Detailed feedback about the property from the customer |
| | Ratings | Numeric(1) | NOT NULL | Ratings for the property |

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

# 3. ENTITY RELATIONSHIP DIAGRAM:



**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

# 4. DATABASE SYSTEM ARCHITECTURE:

We used the following tools to create and implement this project:

1. **MS Visio:** We created entity relationship diagram using MS Visio. Here, we created entities and defined their attributes along with the keys (primary & foreign). We established the relationships and dependencies between entities.
2. **SQL Server:** We used SQL Server as the database that stored all the tables and their data. We created tables in the database using SQL queries. Also, we used SQL server to create and deploy triggers required for the project.
3. **MS Access:** We used MS Access to create the interface for the system. Using Access, we link our tables that were created in SQL Server. Once, the tables were linked, we created forms that could take user input or display the necessary information to the users. Based on the data, we used MS Access to generate reports for the users of the system.

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

# 5. SQL SCRIPTS FOR CREATING TABLE AND INSERTING DATA:

**CREATING TABLES:**
We created tables in SQL Server using the following SQL queries:

1. **Creating Tables:**
    1. **Customer Table:**

```
CREATE TABLE customer (CustomerID varchar(30),
CustomerPassword varchar(25) NOT NULL,
CFName varchar(30) NOT NULL,
CLName varchar(30) NOT NULL,
CPhone char(12) NOT NULL,
CEmail varchar(50) NOT NULL,
CAptNo INT NOT NULL,
CStreetName varchar(30) NOT NULL,
CCity varchar(30) NOT NULL,
CState char(2) NOT NULL,
CZip char(5) NOT NULL,
IDProof varchar(15) NOT NULL,
IDNumber varchar(15) NOT NULL,

CONSTRAINT pass CHECK (len(CustomerPassword)>(8) AND [CustomerPassword] like '%[0-9]%' AND
[CustomerPassword] <> Lower([CustomerPassword]) COLLATE Latin1_General_CS_AI),
CONSTRAINT CID_PK PRIMARY KEY(CustomerID),
CONSTRAINT email_check CHECK (CEmail LIKE '%@%'),
CONSTRAINT IDP CHECK (IDProof='Driving License' OR IDProof='Passport'));
```

    2. **Property Table:**

```
CREATE TABLE property (PropertyId varchar(10),
PropertyTypeId varchar(10) NOT NULL,
AptNo int NOT NULL,
PStreetName varchar(30) NOT NULL,
PCity varchar(30) NOT NULL,
PState char(2) NOT NULL,
NoOfRooms int NOT NULL,
PropertyStatus varchar(10) NOT NULL,

CONSTRAINT PID_PK PRIMARY KEY(PropertyId),
CONSTRAINT PID_FK FOREIGN KEY(PropertyTypeId) REFERENCES property_type,
CONSTRAINT PS CHECK (PropertyStatus='Available' OR PropertyStatus='Reserved'));
```

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

3. **Property Type Table:**

```
CREATE TABLE property_type (
PropertyTypeId varchar(10) NOT NULL,
MaxGuestsAllowed int NOT NULL,
RatePerDay DECIMAL NOT NULL,
PropertyType varchar(15) NOT NULL,

CONSTRAINT PTID_PK PRIMARY KEY(PropertyTypeId),
CONSTRAINT PT CHECK(PropertyType='Villa' OR PropertyType='Bungalow' OR PropertyType='Apartment'
OR PropertyType='Cottage'));
```

4. **Reservation Table:**

```
CREATE TABLE reservation (
ReservationID INT IDENTITY(100,1) NOT NULL ,
CustomerID varchar(30) NOT NULL,
PropertyId varchar(10) NOT NULL,
CancellationStatus varchar(10) DEFAULT 'Reserved',
CheckinDate Date NOT NULL,
CheckoutDate Date NOT NULL,
ReservationDate DATETIME default getdate() NOT NULL,
TotalAmount DECIMAL DEFAULT 0 NOT NULL,
NoOfGuests INT NOT NULL,
CancellationDate DATE DEFAULT NULL,

CONSTRAINT RID_PK PRIMARY KEY(ReservationID),
CONSTRAINT RID_FK FOREIGN KEY(CustomerId) REFERENCES customer,
CONSTRAINT RID_FK1 FOREIGN KEY(PropertyId) REFERENCES property,
CONSTRAINT CS CHECK (CancellationStatus='Reserved' or CancellationStatus='Cancelled'));
```

5. **Available_services Table:**

```
CREATE TABLE available_services (
ServiceID varchar(10) NOT NULL,
ServiceName varchar(25) NOT NULL,
ServiceDesc varchar(100) NOT NULL,
ServiceCost Decimal NOT NULL,

CONSTRAINT SID_PK PRIMARY KEY(ServiceID));
```

6. **Reservation_services Table:**

```
CREATE TABLE reservation_services (
ServiceID varchar(10) NOT NULL,
ReservationID INT NOT NULL,
ServiceDate Date default getdate() NOT NULL,

CONSTRAINT RSID_PK PRIMARY KEY(ServiceId, ReservationId),
CONSTRAINT RSID_FK1 FOREIGN KEY(ServiceId) REFERENCES available_services,
CONSTRAINT RSID_FK2 FOREIGN KEY(ReservationId) REFERENCES reservation          );
```

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

7. **Payment_method Table:**

```
CREATE TABLE payment_method(
PaymentMethodID INT IDENTITY(10000,1) NOT NULL,
ReservationID INT NOT NULL,
CardNumber Numeric(16) NOT NULL,
ExpiryDate Date NOT NULL,
NameonCard varchar(30) NOT NULL,
CVV NUMERIC(3) NOT NULL

CONSTRAINT PM_PK PRIMARY KEY(PaymentMethodID),
CONSTRAINT PM_FK FOREIGN KEY(ReservationID) REFERENCES reservation);
```

8. **Customer_feedback Table:**

```
CREATE TABLE Customer_feedback(
PropertyID varchar(10) NOT NULL,
ReservationID INT NOT NULL,
Feedback varchar(100) NOT NULL,
Ratings numeric(1) NOT NULL

CONSTRAINT CF_PK PRIMARY KEY(PropertyID, ReservationID),
CONSTRAINT CF_FK1 FOREIGN KEY(PropertyID) REFERENCES property,
CONSTRAINT CF_FK2 FOREIGN KEY(ReservationID) REFERENCES reservation,
CONSTRAINT CFR CHECK(Ratings<=5 and Ratings>=0));
```

**INSERTING DATA:**

We used the following SQL queries to insert data in the tables created using the above queries:

**1. Customer Table:**

```
INSERT INTO customer values('akshay','Akshay123', 'akshay', 'sigar','315-913-
5157','akshays@gmail.com','112','Lafayeete Rd','Syracuse','NY','13205','Passport','P123465');
INSERT INTO customer values('nikita','Nikita567', 'Nikita', 'Dongare','315-913-
5125','nikita11@gmail.com','512','Lancaster Ave','Syracuse','NY','13210','Passport','P875263');
INSERT INTO customer values('rahul','Rahul2309', 'Rahul', 'Wable','315-628-
7280','rahulwable@gmail.com','203','Dell St','Syracuse','NY','13205','Driving License','S7215631');
```

**2. Property Type Table:**

```
INSERT INTO property_type values('PT01','8','200','Villa')
INSERT INTO property_type values('PT02','4','100','Apartment')
INSERT INTO property_type values('PT03','6','300','Bungalow')
INSERT INTO property_type values('PT04','5','250','Cottage')
```

**3. Property Table:**

```
INSERT INTO property values('P1','PT01', '12', 'East Gennese St','Syracuse','NY','6','Available');
INSERT INTO property values('P2','PT02', '53', 'Columbus Ave','Syracuse','NY','3','Available');
INSERT INTO property values('P3','PT01', '414', 'South Beech St','Syracuse','NY','6','Available');
INSERT INTO property values('P4','PT03', '115', 'Maryland Ave','Syracuse','NY','6','Available');
```

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

**4. Reservation Table:**

```
INSERT INTO reservation values('akshay', 'P2','Reserved','2019-08-10','2019-08-13','2019-08-08','435','3',DEFAULT);
INSERT INTO reservation values('nikita', 'P5','Reserved','2019-10-12','2019-10-14','2019-10-10','515','4',DEFAULT);
INSERT INTO reservation values('rahul', 'P5','Reserved','2016-12-12','2016-12-14','2016-12-10','520','4',DEFAULT);
```

**5. Available Services Table:**

```
INSERT INTO available_services values('S1','Gym','Well equiped professional gym and other recreational activities','50')
INSERT INTO available_services values('S2','WiFi','24*7 Internet Access','20')
INSERT INTO available_services values('S3','Laundry','Laundry within the house is charged at $15 per load','15')
INSERT INTO available_services values('S4','Breakfast, Lunch & Dinner','Food buffets charged per person','50')
```

**6. Reservation_Services Table:**

```
INSERT INTO reservation_services values('S1','100','2018-11-12')
INSERT INTO reservation_services values('S2','100','2018-11-12')
INSERT INTO reservation_services values('S4','101','2019-08-10')
INSERT INTO reservation_services values('S2','101','2019-08-10')
```

**7. Payment_Method Table:**

```
INSERT INTO payment_method values('101','9000525242417896','2023-11-05','Akshay Sigar','101')
INSERT INTO payment_method values('102','6001002504050789','2021-10-30','Nikita Dongare','875')
INSERT INTO payment_method values('103','5278142563529874','2019-12-31','Rahul Wable','485')
```

**8. Customer Feedback Table:**

```
INSERT INTO customer_feedback values('P5','100','Amazing experience','5')
INSERT INTO customer_feedback values('P2','101','Good Host! Clean property','3')
INSERT INTO customer_feedback values('P5','102','Great experience','4')
```

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

# OUTPUT FOR CREATE AND INSERT SQL SCRIPTS:

The below tables contain data which was inserted using SQL scripts, Access forms and triggers

## 1. Customer Table:

| | CustomerID | CustomerPassword | CFName | CLName | CPhone | CEmail | CAptNo | CStreetName | CCity | CState | CZip | IDProof | IDNumber |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | akshay | Akshay123 | akshay | sigar | 315-913-5157 | akshays@gmail.com | 112 | Lafayeete Rd | Syracuse | NY | 13205 | Passport | P123465 |
| 2 | Joey | Joeyt1997 | Joey | Johnson | 315-728-0578 | joeyt@gmail.com | 270 | Westcott St | Syracuse | NY | 13210 | Passport | P745892 |
| 3 | josh | Magnitude56 | Josh | Steve | 315-913-2880 | josh97@gmail.com | 720 | Westcott St | Syracuse | NY | 13210 | Passport | S2503674 |
| 4 | Mark | Chocolates579 | Mark | Bing | 315-913-7847 | bingmark@gmail.com | 120 | Comstock Ave | Syracuse | NY | 13210 | Passport | D7854697 |
| 5 | meghan | Sqlite127 | Meghan | James | 315-913-2057 | meghanj@gmail.com | 920 | Comstock Ave | Syracuse | NY | 13210 | Driving License | D8567561 |
| 6 | neha | Nehad1234 | Neha | D | 315-913-7458 | neha123@gmail.com | 540 | Lafayeete Rd | Syracuse | NY | 13205 | Passport | P785741 |
| 7 | nick | Thomas2597 | Nick | Thomas | 315-417-0012 | nickth@gmail.com | 741 | Lancaster Ave | Syracuse | NY | 13210 | Driving License | D7458741 |
| 8 | nikita | Nikita567 | Nikita | Dongare | 315-913-5125 | nikita11@gmail.com | 512 | Lancaster Ave | Syracuse | NY | 13210 | Passport | P875263 |
| 9 | rahul | Rahul2309 | Rahul | Wable | 315-628-7280 | rahulwable@gmail.com | 203 | Dell St | Syracuse | NY | 13205 | Driving License | S7215631 |
| 10 | sophia | SophiaJo45 | Sophia | Jaden | 315-621-7880 | sophiaj@gmail.com | 23 | Dell St | Syracuse | NY | 13210 | Driving License | D4215042 |

## 2. Property Type Table:

| | PropertyTypeId | MaxGuestsAllowed | RatePerDay | PropertyType |
|---|---|---|---|---|
| 1 | PT01 | 8 | 200 | Villa |
| 2 | PT02 | 4 | 100 | Apartment |
| 3 | PT03 | 6 | 300 | Bungalow |
| 4 | PT04 | 5 | 250 | Cottage |
| 5 | PT05 | 10 | 500 | Villa |
| 6 | PT06 | 5 | 200 | Apartment |

## 3. Property Table:

| | PropertyId | PropertyTypeId | AptNo | PStreetName | PCity | PState | NoOfRooms | PropertyStatus |
|---|---|---|---|---|---|---|---|---|
| 1 | P1 | PT01 | 12 | East Gennese St | Syracuse | NY | 6 | Available |
| 2 | P10 | PT06 | 148 | Westcotte St | Syracuse | NY | 4 | Available |
| 3 | P2 | PT02 | 53 | Columbus Ave | Syracuse | NY | 3 | Available |
| 4 | P3 | PT01 | 414 | South Beech St | Syracuse | NY | 6 | Available |
| 5 | P4 | PT03 | 115 | Maryland Ave | Syracuse | NY | 6 | Available |
| 6 | P5 | PT04 | 531 | East Gennese St | Syracuse | NY | 4 | Available |
| 7 | P6 | PT05 | 258 | Lancaster Ave | Syracuse | NY | 6 | Available |
| 8 | P7 | PT01 | 721 | Ostrom Ave | Syracuse | NY | 5 | Available |
| 9 | P8 | PT05 | 400 | Liverpool | Syracuse | NY | 6 | Available |
| 10 | P9 | PT02 | 95 | Maryland Ave | Syracuse | NY | 2 | Available |

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

## 4. Reservation Table:

| | ReservationID | CustomerID | PropertyId | CancellationStatus | CheckinDate | CheckoutDate | ReservationDate | TotalAmount | NoOfGuests | CancellationDate |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 100 | nikita | P5 | Reserved | 2018-11-12 | 2018-11-14 | 2018-11-10 00:00:00.000 | 570 | 4 | NULL |
| 2 | 101 | akshay | P2 | Reserved | 2019-08-10 | 2019-08-13 | 2019-08-08 00:00:00.000 | 435 | 3 | NULL |
| 3 | 102 | nikita | P5 | Reserved | 2019-10-12 | 2019-10-14 | 2019-10-10 00:00:00.000 | 515 | 4 | NULL |
| 4 | 103 | rahul | P5 | Reserved | 2016-12-12 | 2016-12-14 | 2016-12-10 00:00:00.000 | 520 | 4 | NULL |
| 5 | 104 | josh | P3 | Reserved | 2018-12-25 | 2018-12-31 | 2018-12-23 00:00:00.000 | 1220 | 6 | NULL |
| 6 | 105 | meghan | P1 | Reserved | 2018-12-27 | 2018-12-31 | 2018-12-25 00:00:00.000 | 850 | 8 | NULL |
| 7 | 106 | neha | P3 | Reserved | 2019-10-25 | 2019-10-27 | 2019-10-15 00:00:00.000 | 500 | 7 | NULL |
| 8 | 107 | nick | P7 | Reserved | 2019-10-12 | 2019-10-14 | 2019-09-30 00:00:00.000 | 435 | 5 | NULL |
| 9 | 108 | nikita | P1 | Reserved | 2017-12-12 | 2017-12-16 | 2017-12-01 00:00:00.000 | 800 | 8 | NULL |
| 10 | 109 | sophia | P1 | Reserved | 2019-10-02 | 2019-10-05 | 2019-09-23 00:00:00.000 | 600 | 6 | NULL |
| 11 | 110 | mark | P6 | Reserved | 2019-10-27 | 2019-10-31 | 2019-09-25 00:00:00.000 | 2070 | 4 | NULL |
| 12 | 111 | joey | P4 | Reserved | 2019-10-06 | 2019-10-14 | 2019-09-15 00:00:00.000 | 2400 | 5 | NULL |
| 13 | 112 | akshay | P8 | Reserved | 2019-10-12 | 2019-10-16 | 2019-09-01 00:00:00.000 | 2020 | 8 | NULL |
| 14 | 113 | sophia | P9 | Reserved | 2018-10-02 | 2018-10-05 | 2018-09-23 00:00:00.000 | 315 | 6 | NULL |
| 15 | 114 | mark | P10 | Reserved | 2015-10-27 | 2015-10-31 | 2015-09-25 00:00:00.000 | 850 | 4 | NULL |
| 16 | 115 | nick | P1 | Reserved | 2019-10-20 | 2019-10-22 | 2019-09-25 00:00:00.000 | 400 | 5 | NULL |

## 5. Available Services Table:

| | ServiceID | ServiceName | ServiceDesc | ServiceCost |
|---|---|---|---|---|
| 1 | S1 | Gym | Well equiped professional gym and other recreati... | 50 |
| 2 | S2 | WiFi | 24*7 Internet Access | 20 |
| 3 | S3 | Laundry | Laundry within the house is charged at $15 per load | 15 |
| 4 | S4 | Breakfast, Lunch & Dinner | Food buffets charged per person | 50 |

## 6. Reservation Services Table:

| | ServiceID | ReservationID | ServiceDate |
|---|---|---|---|
| 1 | S1 | 100 | 2018-11-12 |
| 2 | S1 | 101 | 2019-08-10 |
| 3 | S1 | 105 | 2018-12-27 |
| 4 | S1 | 106 | 2019-10-25 |
| 5 | S1 | 110 | 2019-10-27 |
| 6 | S1 | 114 | 2015-10-27 |
| 7 | S2 | 100 | 2018-11-12 |
| 8 | S2 | 101 | 2019-08-10 |
| 9 | S2 | 103 | 2016-12-12 |
| 10 | S2 | 104 | 2018-12-25 |
| 11 | S2 | 107 | 2019-10-12 |
| 12 | S2 | 110 | 2019-10-27 |
| 13 | S2 | 112 | 2019-10-12 |
| 14 | S3 | 101 | 2019-08-10 |
| 15 | S3 | 102 | 2019-10-12 |
| 16 | S3 | 107 | 2019-10-12 |
| 17 | S3 | 113 | 2018-10-02 |
| 18 | S4 | 101 | 2019-08-10 |
| 19 | S4 | 106 | 2019-10-25 |

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

## 7. Payment_Method Table

| | PaymentMethodID | ReservationID | CardNumber | ExpiryDate | NameonCard | CVV |
|---|---|---|---|---|---|---|
| 1 | 10000 | 100 | 6011524011201111 | 2024-10-25 | Nikita Dongare | 251 |
| 2 | 10001 | 101 | 9000525242417896 | 2023-11-05 | Akshay Sigar | 101 |
| 3 | 10002 | 102 | 6001002504050789 | 2021-10-30 | Nikita Dongare | 875 |
| 4 | 10003 | 103 | 5278142563529874 | 2019-12-31 | Rahul Wable | 485 |
| 5 | 10004 | 104 | 9785100050004102 | 2020-07-25 | Josh Steve | 633 |
| 6 | 10005 | 105 | 7412853695221000 | 2024-05-05 | Meghan James | 114 |
| 7 | 10006 | 106 | 9896152045894447 | 2022-11-05 | Neha Dongare | 310 |
| 8 | 10007 | 107 | 5078902557850012 | 2021-11-03 | Nick Thomas | 785 |
| 9 | 10008 | 108 | 6001002504050789 | 2021-10-30 | Nikita Dongare | 875 |
| 10 | 10009 | 109 | 5201258442261002 | 2024-01-25 | Sophia Jaden | 753 |
| 11 | 10010 | 110 | 9875632100257586 | 2023-07-05 | Mark Bing | 856 |
| 12 | 10011 | 111 | 6006657842531025 | 2024-01-30 | Joey Johnson | 505 |
| 13 | 10012 | 112 | 9000525242417896 | 2023-11-05 | Akshay Sigar | 101 |
| 14 | 10013 | 113 | 5201258442261002 | 2024-01-25 | Sophia Jaden | 753 |
| 15 | 10014 | 114 | 9875632100257586 | 2023-07-05 | Mark Bing | 856 |
| 16 | 10015 | 115 | 5078902557850012 | 2021-11-03 | Nick Thomas | 785 |

## 8. Customer Feedback Table

| | PropertyID | ReservationID | Feedback | Ratings |
|---|---|---|---|---|
| 1 | P1 | 105 | Great experience! Would definitely visit again | 5 |
| 2 | P1 | 108 | Amazing location, definitely recommended | 4 |
| 3 | P1 | 109 | Value for money, Good Host | 5 |
| 4 | P1 | 115 | Great experience! Would definitely visit again | 5 |
| 5 | P10 | 114 | Great experience! | 4 |
| 6 | P2 | 101 | Good Host! Clean property | 3 |
| 7 | P3 | 104 | Unclean property | 1 |
| 8 | P3 | 106 | Unclean property, Shady location | 2 |
| 9 | P4 | 111 | Lovely experience! Would definitely visit ag... | 5 |
| 10 | P5 | 100 | Amazing experience | 5 |
| 11 | P5 | 102 | Great experience | 4 |
| 12 | P5 | 103 | Amazing experience | 4 |
| 13 | P6 | 110 | Friendly caretaker, Services provided were ... | 3 |
| 14 | P7 | 107 | Enjoyed the stay, value for money | 3 |
| 15 | P8 | 112 | Amazing experience | 5 |
| 16 | P9 | 113 | Location is not good | 1 |

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

# 6. BUSINESS RULES:

1. The customer must be a registered user on the system.
2. The user must have a valid id proof for reservations.
3. The property type can be an apartment, cottage, villa or bungalow.
4. A user can make one or more reservations.
5. The user must cancel a reservation 24 hours prior to check-in.
6. The user must complete the payment process in order to make a reservation.
7. Guests can be an individual person or a group of people.
8. The ratings in customer feedback must be given on a scale of 0 to 5.

# 7. MAJOR DATA QUESTIONS

We answered the major data questions using SQL queries in MS Access. We created forms and reports to display the same to the user for easy understanding.

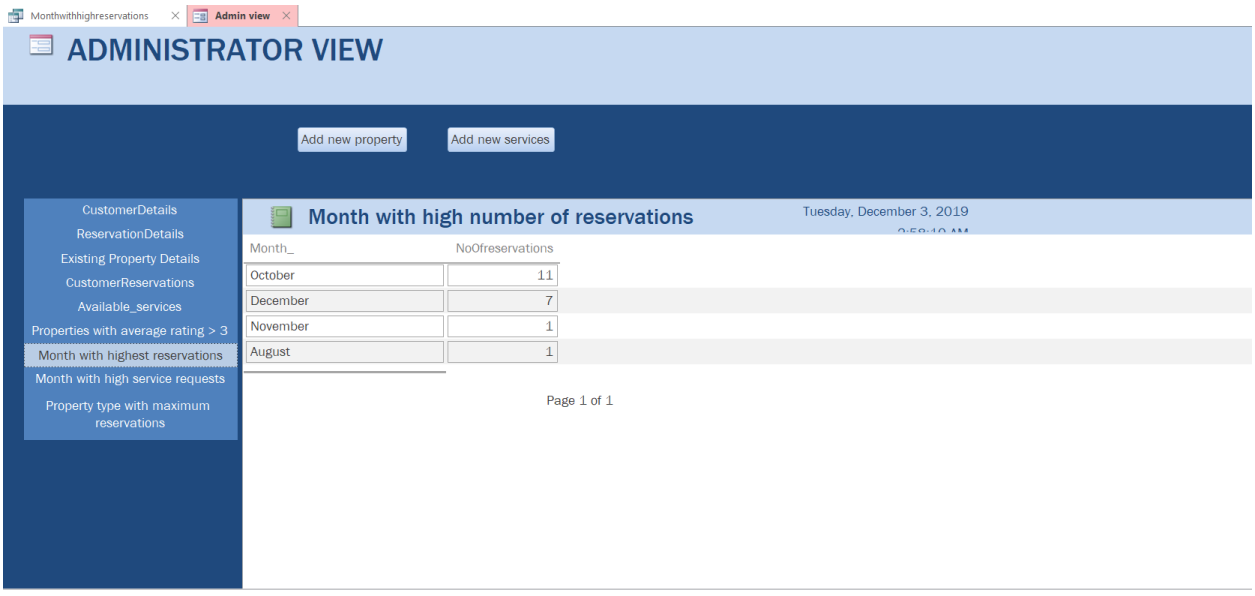
1. **Which month had the maximum number of reservations?**

**SQL Query in MS Access:**

```
SELECT MONTHNAME(DATEPART("m",dbo_reservation.CheckinDate)) AS Month_,
   COUNT(MONTHNAME(DATEPART("m",dbo_reservation.CheckinDate))) AS NoOfreservations
FROM dbo_reservation
GROUP BY MONTHNAME(DATEPART("m",dbo_reservation.CheckinDate));
```

**Output (Report):**

From the report, we can see different months and the number of reservations in each month. We can see that October has maximum number of reservations.



**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

**SQL Query in SQLserver:**

**The query below displays the different months and number of reservations in each month.**

```
SELECT DATENAME(MONTH,(CheckinDate)) as Month_, COUNT(DATENAME(MONTH,CheckinDate)) as
NoOfreservations
FROM Reservation
GROUP BY DATENAME(MONTH,CheckinDate)
```

Results | Messages

| | Month_ | NoOfreservations |
|---|---|---|
| 1 | August | 1 |
| 2 | December | 7 |
| 3 | November | 1 |
| 4 | October | 11 |

**The query below displays the month with maximum number of reservations which answers our data question.**

```
SELECT DATENAME(MONTH,CheckinDate) as Month_ ,COUNT(DATENAME(MONTH,CheckinDate)) as
NumberofReservations
FROM Reservation
GROUP BY DATENAME(MONTH,CheckinDate)
HAVING COUNT(DATENAME(MONTH,CheckinDate)) = (SELECT MAX(b.NoOfreservations) as
NoOfreservations
FROM (SELECT DATENAME(MONTH,(CheckinDate)) as Month_, COUNT(DATENAME(MONTH,CheckinDate))
as NoOfreservations
FROM Reservation
GROUP BY DATENAME(MONTH,CheckinDate)) b )
```

Results | Messages

| | Month_ | NumberofReservations |
|---|---|---|
| 1 | October | 11 |

## 2. Which month had the greatest number of service requests in 2019?

**SQL Query in MS Access:**

```
SELECT MONTHNAME(DATEPART("m",dbo_reservation_services.ServiceDate)) AS Month_,
   COUNT(MONTHNAME(DATEPART("m",dbo_reservation_services.ServiceDate))) AS NoOfservicereservations
FROM dbo_reservation_services
WHERE YEAR(ServiceDate)=2019
GROUP BY MONTHNAME(DATEPART("m",dbo_reservation_services.ServiceDate));
```

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

## Output (Report):

From the report, we can see that different months and number of service requests in each month. We can see that there were 8 service requests in October which is the maximum.


## SQL Query in SQLserver:

**The query below shows the different months and number of service requests in each month in 2019.**

```
SELECT DATENAME(MONTH,(ServiceDate)) as Month_, COUNT(DATENAME(MONTH,ServiceDate)) as
NoOfservicereservations
FROM Reservation_services
WHERE YEAR(ServiceDate)=2019
GROUP BY DATENAME(MONTH,ServiceDate)
```

Results | Messages

| | Month_ | NoOfservicereservations |
|---|---|---|
| 1 | August | 4 |
| 2 | December | 5 |
| 3 | October | 8 |

**The query below displays the month with maximum number of service requests i.e. October which answers our data question.**

```
SELECT DATENAME(MONTH,ServiceDate) as Month_ ,COUNT(DATENAME(MONTH,ServiceDate)) as
NumberofserviceReservations
FROM Reservation_services
WHERE YEAR(ServiceDate)=2019
GROUP BY DATENAME(MONTH,ServiceDate)
HAVING COUNT(DATENAME(MONTH,ServiceDate)) = (SELECT MAX(b.NoOfservicereservations) as
NoOfservicereservations
FROM (SELECT DATENAME(MONTH,(ServiceDate)) as Month_, COUNT(DATENAME(MONTH,ServiceDate)) as
NoOfservicereservations
FROM Reservation_services
WHERE YEAR(ServiceDate)=2019
GROUP BY DATENAME(MONTH,ServiceDate)) b )
```

Results | Messages

| | Month_ | NumberofserviceReservations |
|---|---|---|
| 1 | October | 8 |

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

3. **Which properties have received an average rating of more than 3?**

**SQL Query in MS Access:**

**Query1: AvgRating (This query filters out propertId with average rating greater than 3)**

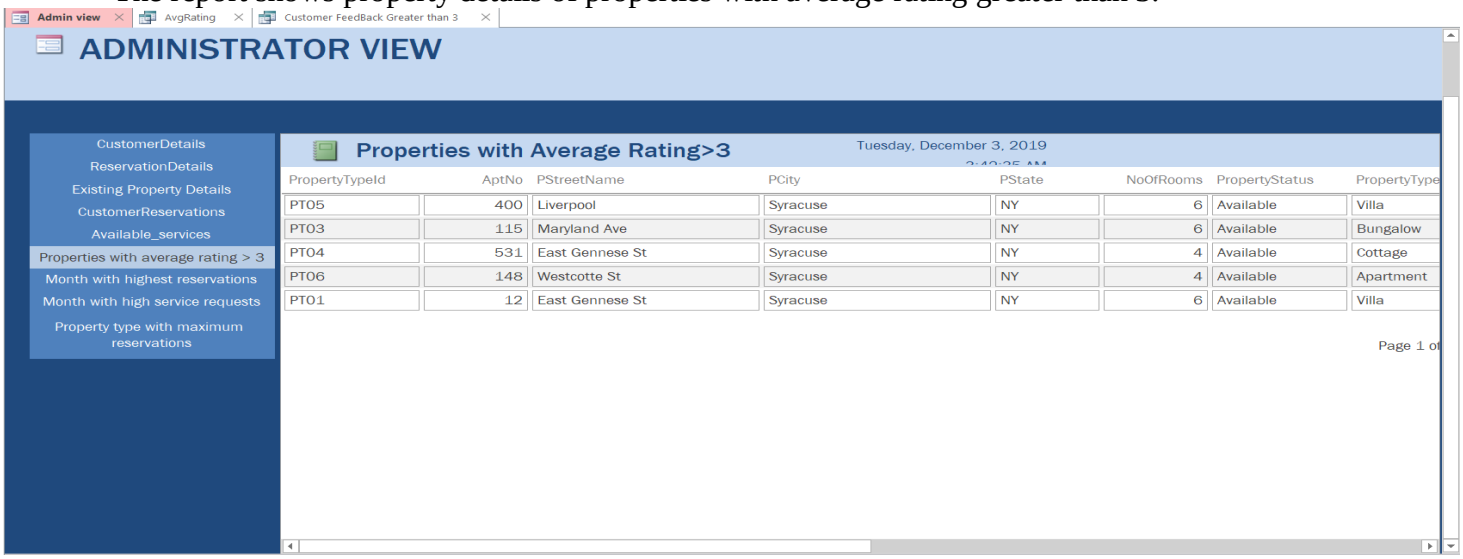
```
SELECT dbo_Customer_feedback.PropertyID, Avg(dbo_Customer_feedback.Ratings) AS AvgOfRatings
FROM dbo_Customer_feedback
GROUP BY dbo_Customer_feedback.PropertyID
HAVING Avg(dbo_Customer_feedback.Ratings)>3;
```

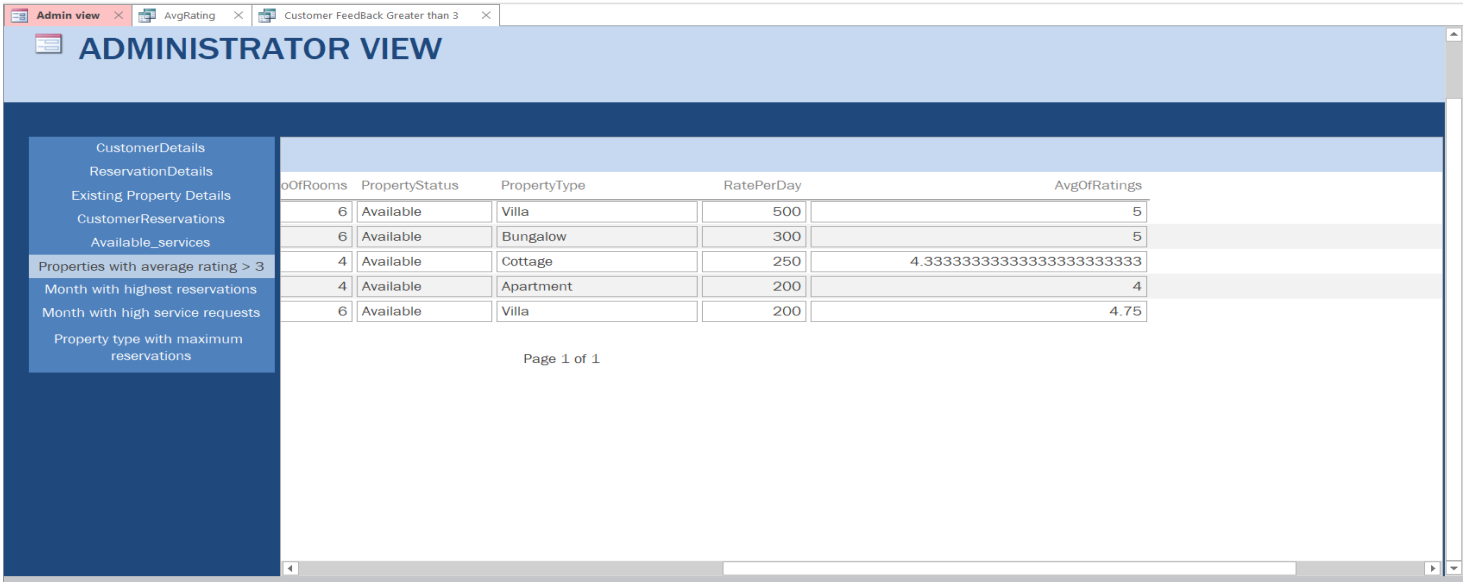
**Query 2: Property with rating greater than 3(This query displays all the information related to properties with average rating greater than 3)**

```
SELECT dbo_property.*, dbo_property_type.PropertyType, dbo_property_type.RatePerDay,
    AvgRating.AvgOfRatings
FROM (AvgRating INNER JOIN dbo_property ON AvgRating.PropertyID = dbo_property.PropertyId) INNER
JOIN dbo_property_type ON dbo_property.PropertyTypeId = dbo_property_type.PropertyTypeId;
```

**Output (Report):**

The report shows property details of properties with average rating greater than 3.



**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

**SQL Query in SQLserver:**

**The query below displays the properties with their average rating greater than 3.**

```
SELECT P.PropertyId, P.PropertyTypeId,PT.PropertyType,
P.AptNo,P.PStreetName,P.PCity,P.PState,P.NoOfRooms,PT.Rateperday, a.Average_Rating
FROM (SELECT AVG(CF.Ratings) as Average_Rating, CF.PropertyId
from Customer_feedback as CF
GROUP BY CF.PropertyId
HAVING AVG(CF.Ratings) > 3) a
JOIN Property as P
ON a.PropertyId=P.PropertyId
JOIN Property_type as PT
ON P.PropertyTypeId=PT.PropertyTypeId
```

Results | Messages

| | PropertyId | PropertyTypeId | PropertyType | AptNo | PStreetName | PCity | PState | NoOfRooms | Rateperday | Average_Rating |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | P1 | PT01 | Villa | 12 | East Gennese St | Syracuse | NY | 6 | 200 | 4.750000 |
| 2 | P10 | PT06 | Apartment | 148 | Westcotte St | Syracuse | NY | 4 | 200 | 4.000000 |
| 3 | P4 | PT03 | Bungalow | 115 | Maryland Ave | Syracuse | NY | 6 | 300 | 5.000000 |
| 4 | P5 | PT04 | Cottage | 531 | East Gennese St | Syracuse | NY | 4 | 250 | 4.333333 |
| 5 | P8 | PT05 | Villa | 400 | Liverpool | Syracuse | NY | 6 | 500 | 5.000000 |

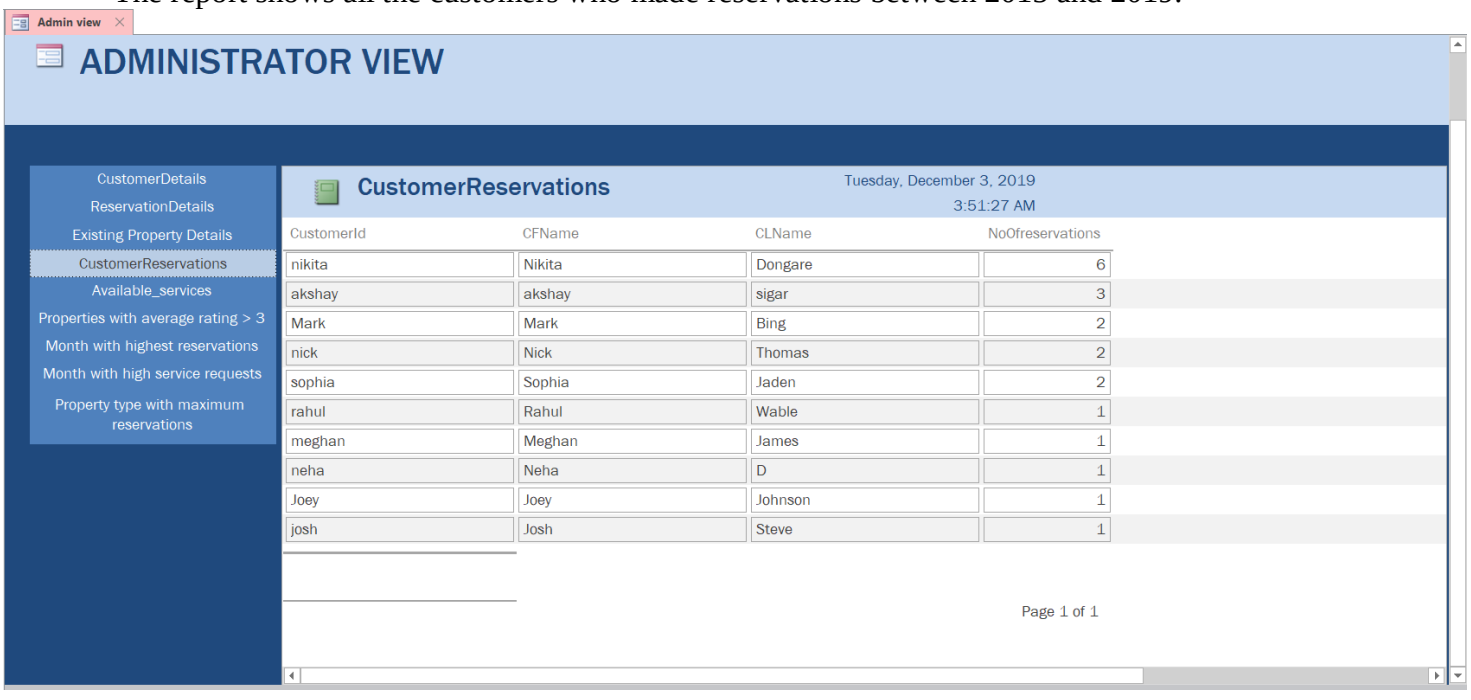
4. **Which customer has made the maximum reservations between 2015 and 2019?**

**SQL Query in MS Access:**

```
SELECT dbo_customer.CustomerId, dbo_customer.CFName, dbo_customer.CLName,
   Count(dbo_customer.CustomerId) AS NoOfreservations
FROM dbo_reservation INNER JOIN dbo_customer ON dbo_reservation.CustomerID = dbo_customer.CustomerID
WHERE DATEPART("yyyy",dbo_reservation.ReservationDate) BETWEEN 2015 and 2019
GROUP BY dbo_customer.CustomerId, dbo_customer.CFName, dbo_customer.CLName;
```

**Output(report):**

The report shows all the customers who made reservations between 2015 and 2019.



**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

**SQL Query in SQLserver:**

**The query below displays the customer and their number of reservations between 2015 and 2019.**

```
SELECT C.CustomerId,C.CFName,C.CLName, COUNT(C.CustomerId) as NoOfreservations
FROM Reservation as R
JOIN Customer as C
ON C.CustomerId=R.CustomerId
WHERE YEAR(R.ReservationDate)>=2015 and YEAR(R.ReservationDate)<=2019
GROUP BY C.CustomerId,C.CFName,C.CLName
ORDER BY NoOfreservations DESC
```

Results | Messages

| | CustomerId | CFName | CLName | NoOfreservations |
|---|---|---|---|---|
| 1 | nikita | Nikita | Dongare | 6 |
| 2 | akshay | akshay | sigar | 3 |
| 3 | Mark | Mark | Bing | 2 |
| 4 | nick | Nick | Thomas | 2 |
| 5 | sophia | Sophia | Jaden | 2 |
| 6 | rahul | Rahul | Wable | 1 |
| 7 | meghan | Meghan | James | 1 |
| 8 | neha | Neha | D | 1 |
| 9 | Joey | Joey | Johnson | 1 |
| 10 | josh | Josh | Steve | 1 |

**The query below displays the customer with maximum number of reservations.**

```
SELECT C.CustomerId,C.CFName,C.CLName, COUNT(R.CustomerId) as NumberofReservations
FROM Customer as C
JOIN Reservation as R
ON C.CustomerId=R.CustomerId
WHERE YEAR(R.ReservationDate)>=2015 and YEAR(R.ReservationDate)<=2019
GROUP BY C.CustomerId,C.CFName,C.CLName
HAVING COUNT(R.CustomerId)= (SELECT MAX(a.NoOfreservations) FROM (SELECT CustomerId,
COUNT(CustomerId) as NoOfreservations FROM Reservation
GROUP BY CustomerId)a)
```

Results | Messages

| | CustomerId | CFName | CLName | NumberofReservations |
|---|---|---|---|---|
| 1 | nikita | Nikita | Dongare | 6 |

5. **Which type of property was booked maximum number of times e.g. villa, cottage, apartment or bungalows?**

**SQL Query in MS Access:**

**Query1: Subqueryproptype1(This query produces the propertyId and no.of reservations related to each propertyId)**

```
SELECT dbo_reservation.PropertyId, Count(dbo_reservation.PropertyId) AS NoofReservations
FROM dbo_reservation
GROUP BY dbo_reservation.PropertyId;
```

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

**Query2: subquery2 (This query displays the property type of different propertyIds and number of reservations associated with each propertyId)**

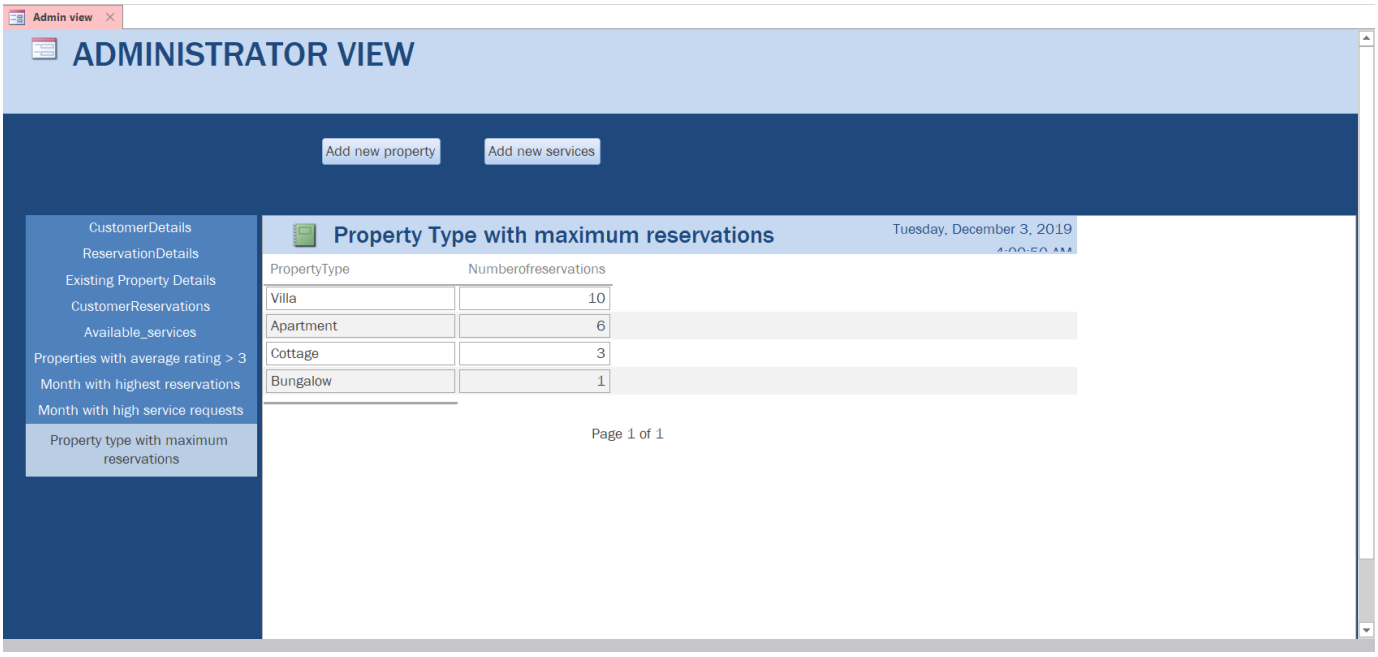
```
SELECT subqueryproptype1.*, dbo_property_type.PropertyTypeId, dbo_property_type.PropertyType
FROM (subqueryproptype1 INNER JOIN dbo_property ON subqueryproptype1.PropertyId = dbo_property.PropertyId)
INNER JOIN dbo_property_type ON dbo_property.PropertyTypeId = dbo_property_type.PropertyTypeId;
```

**Query 3: PropertyTypewithmaxreserv (This query displays the property type and number of reservations associated with each property type)**

```
SELECT subquery2.PropertyType, SUM(subquery2.NoofReservations) AS Numberofreservations
FROM subquery2
GROUP BY subquery2.PropertyType;
```

**Output(report):**

The report shows the property types and no of reservations associated with each property type. Thus, we can see that property type: Villa have been booked the most.



**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

## SQL Query in SQLserver:

**The query below displays the property type with maximum number of reservations. Based on the data, villa property type is reserved maximum number of times which answers our data question.**

```
Select x.PropertyType as PropertyType, Sum(x.NoOfReservation) as TotalNoOfReservation from
(
SELECT  C.Numberofreservations as NoOfReservation, PT.PropertyType as PropertyType
FROM
(SELECT PropertyId, Count(PropertyId) as Numberofreservations
FROM Reservation
GROUP BY PropertyId)C
JOIN Property as P
ON C.PropertyId=P.PropertyId
JOIN Property_type as PT
ON PT.PropertyTypeId=P.PropertyTypeId
)x group by x.PropertyType having SUM(x.NoOfReservation)=

(Select Max(y.TotalNoOfReservation) from
(Select Sum(x.NoOfReservation) as TotalNoOfReservation,x.PropertyType as PropertyType from
(
SELECT  C.Numberofreservations as NoOfReservation, PT.PropertyType as PropertyType
FROM
(SELECT PropertyId, Count(PropertyId) as Numberofreservations
FROM Reservation
GROUP BY PropertyId)C
JOIN Property as P
ON C.PropertyId=P.PropertyId
JOIN Property_type as PT
ON PT.PropertyTypeId=P.PropertyTypeId
)x group by x.PropertyType)y)
```

Results | Messages

| | PropertyType | TotalNoOfReservation |
|---|---|---|
| 1 | Villa | 10 |

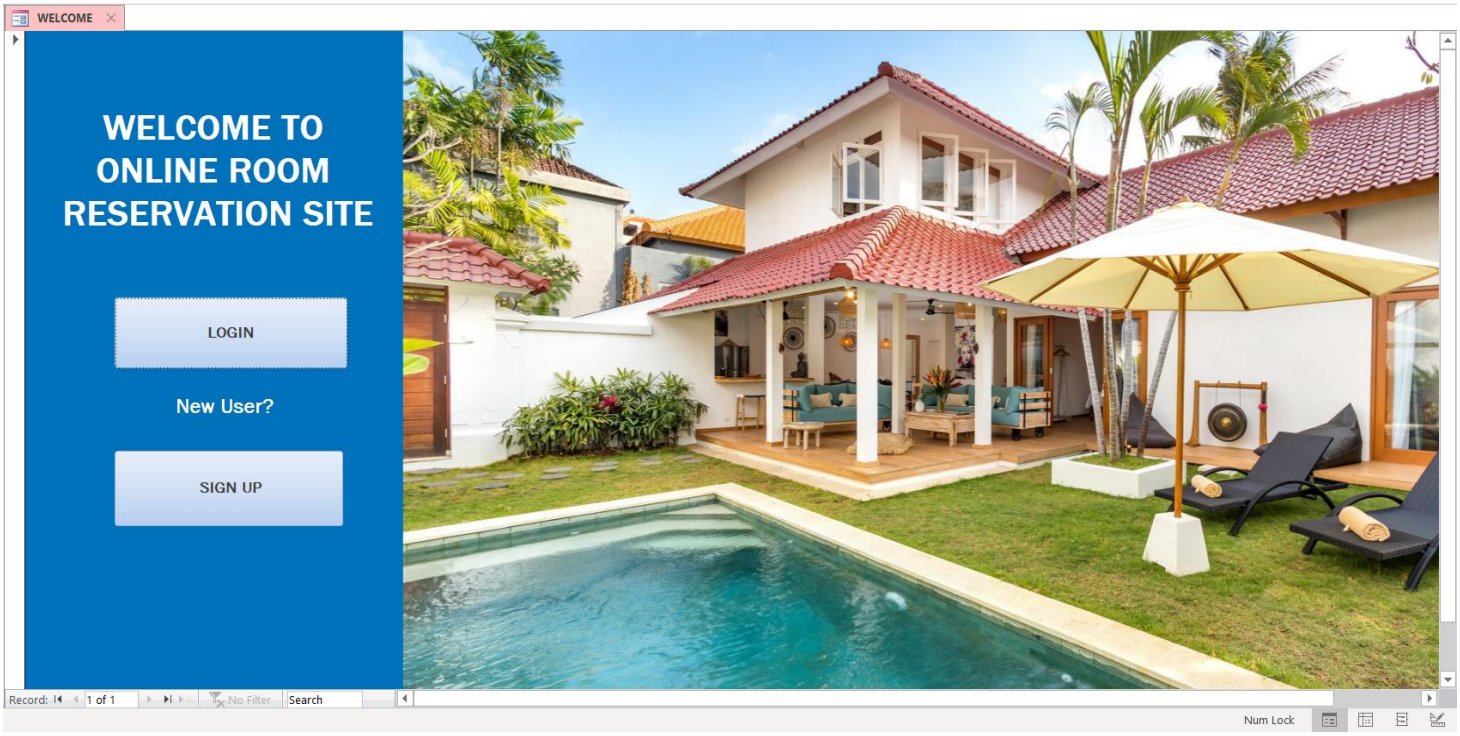
**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

# 8. INTERFACES:

The users of the system are categorized into two groups:

1. Customers: View Properties, Reserve Property, Cancel a reserved property
2. Admin: Maintains the database and performs create, read, update and delete operations

A login page is created which helps to control the access provided to the users. Different dashboards are created for the two types of users.




Validations are provided to ensure User ID and password are entered and they match with the ID and password present in the database.
Based on the type of login, we redirect the user to the next page.

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

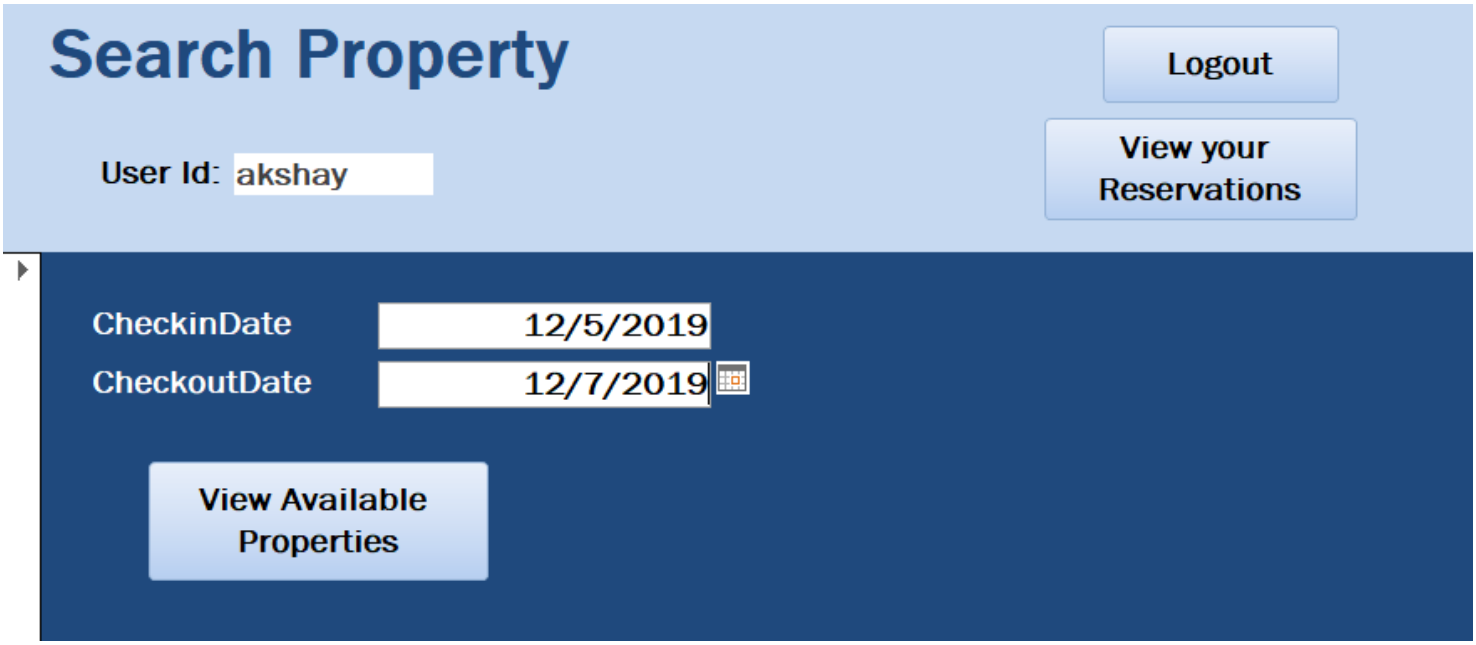
**Scenario 1: Customer login**

For a customer to reserve a property, he/she must be a registered customer in the system.
Thus, a new user can register on the system by clicking on the sign-up button which will load the registration form for the new customer.

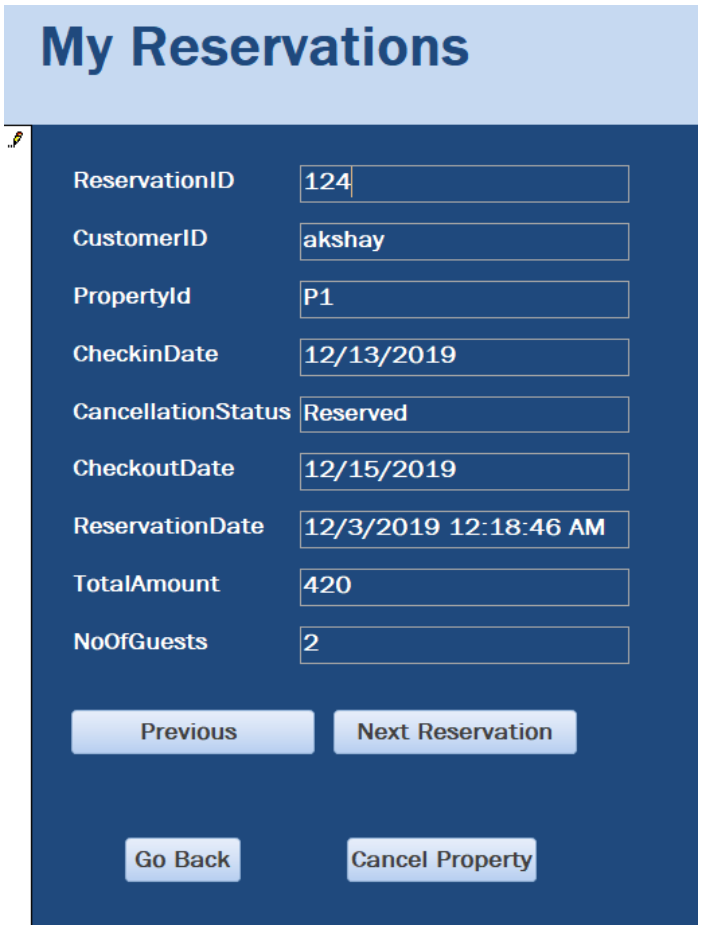

If the customer is already registered on the system, then he/she can login by clicking on login button which will load the login page:

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

On successful login, the customer can either start to search a property or can view his previous reservations:

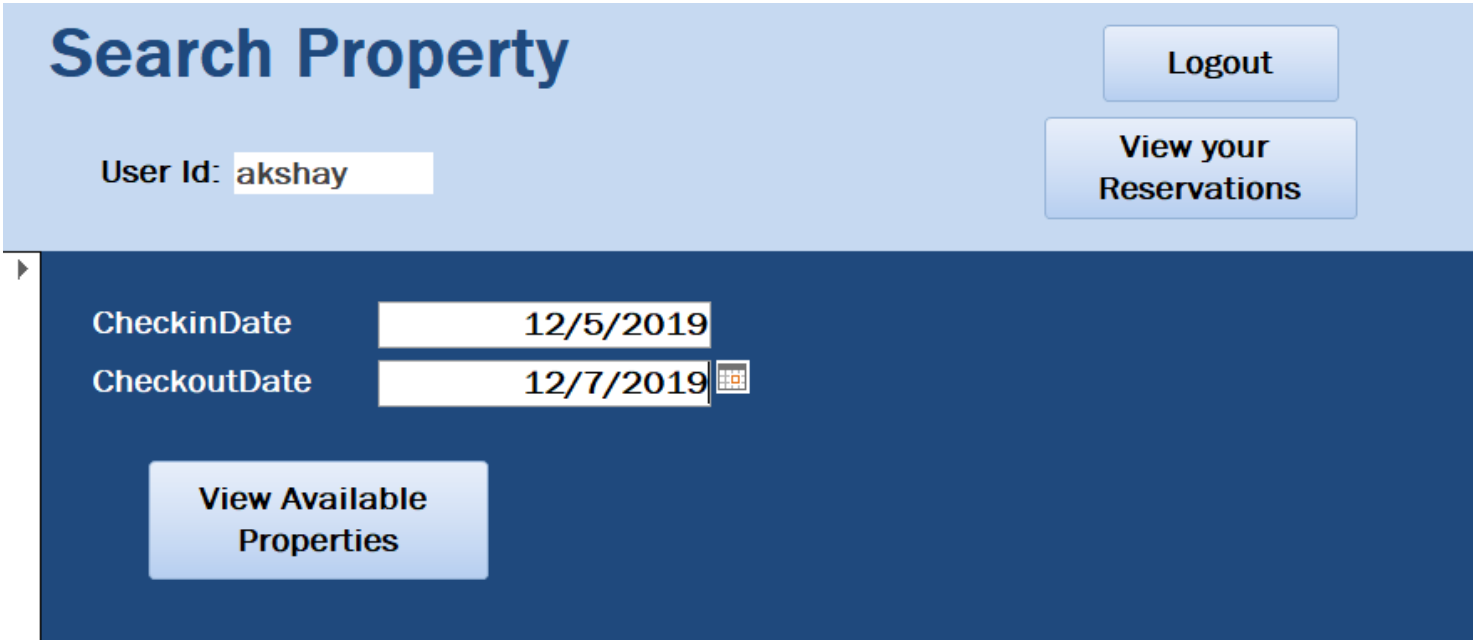



Here, by clicking on **VIEW YOUR RESERVATIONS** it will load the form where the user can view his reservations and can cancel the reservations if the user cancels the reservation 24 hours prior to check-in.



**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

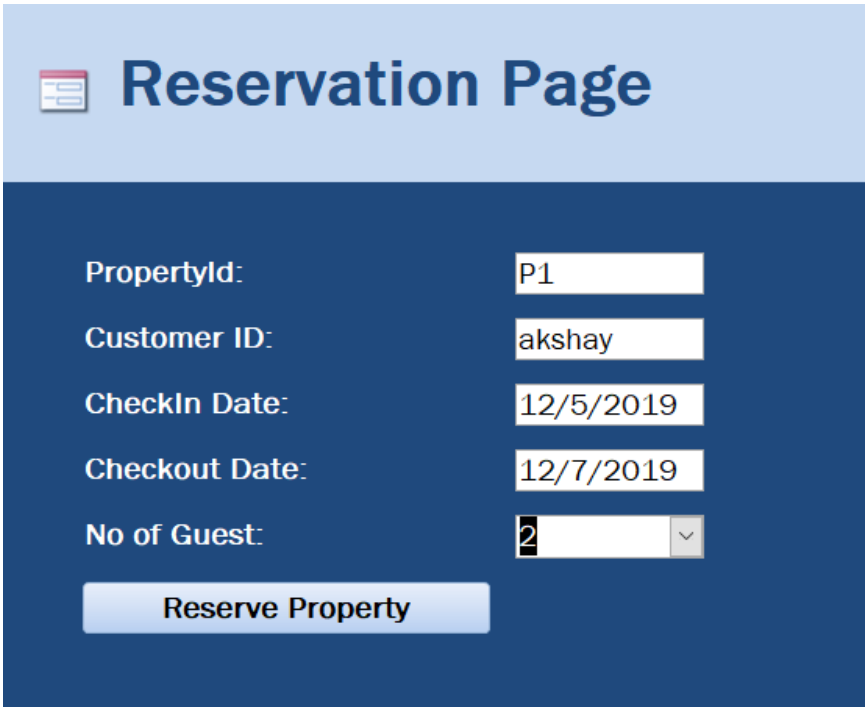
Else the user can enter a check-in and check-out date to search for available property between those days:



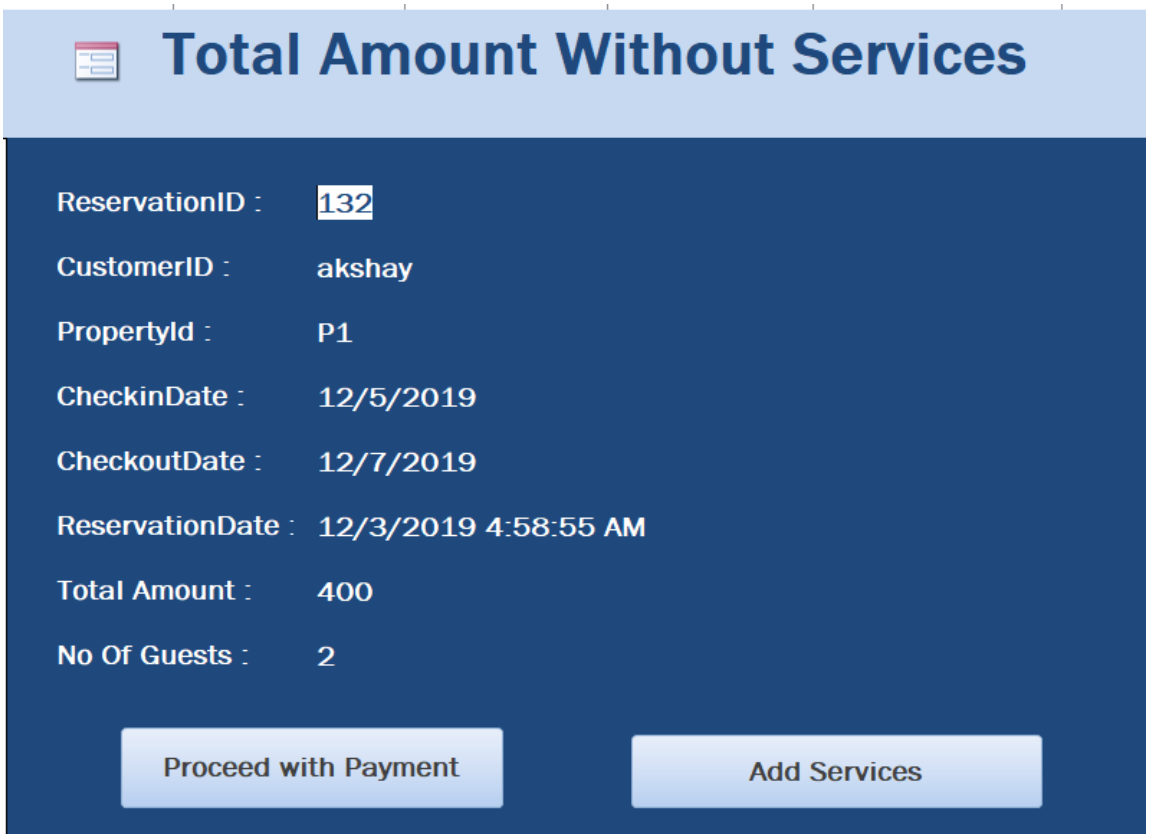

It will display the properties to the user based on the availability on the entered check-in and check-out date. The user can select from the available properties:

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

The User then has to enter the number of guests to reserve the property:

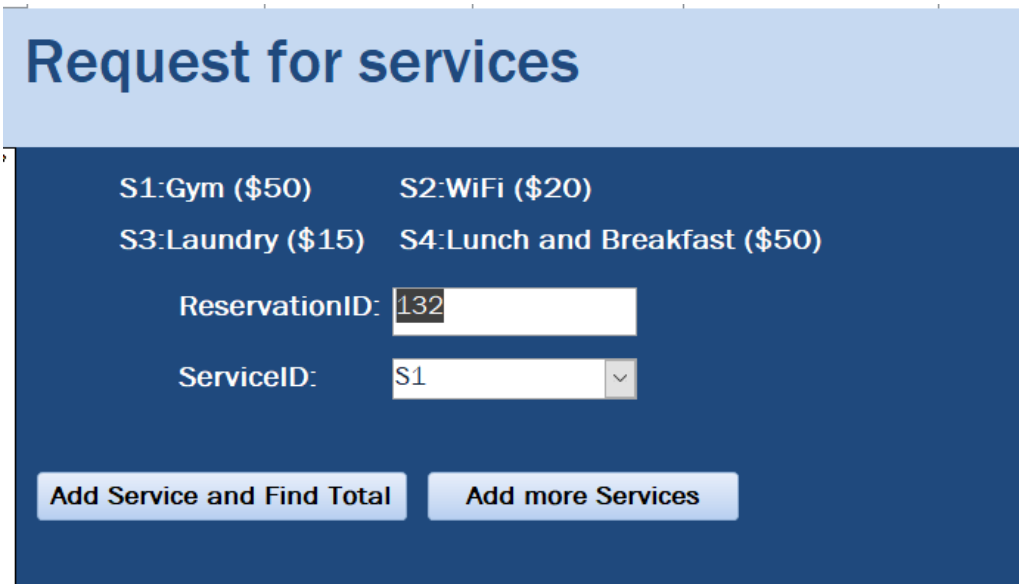



As the user clicks on Reserve Property Button, it will load a summary page with all the details about the reservation. Then the user can directly pay the total amount of reservation or can add services to his reservation.

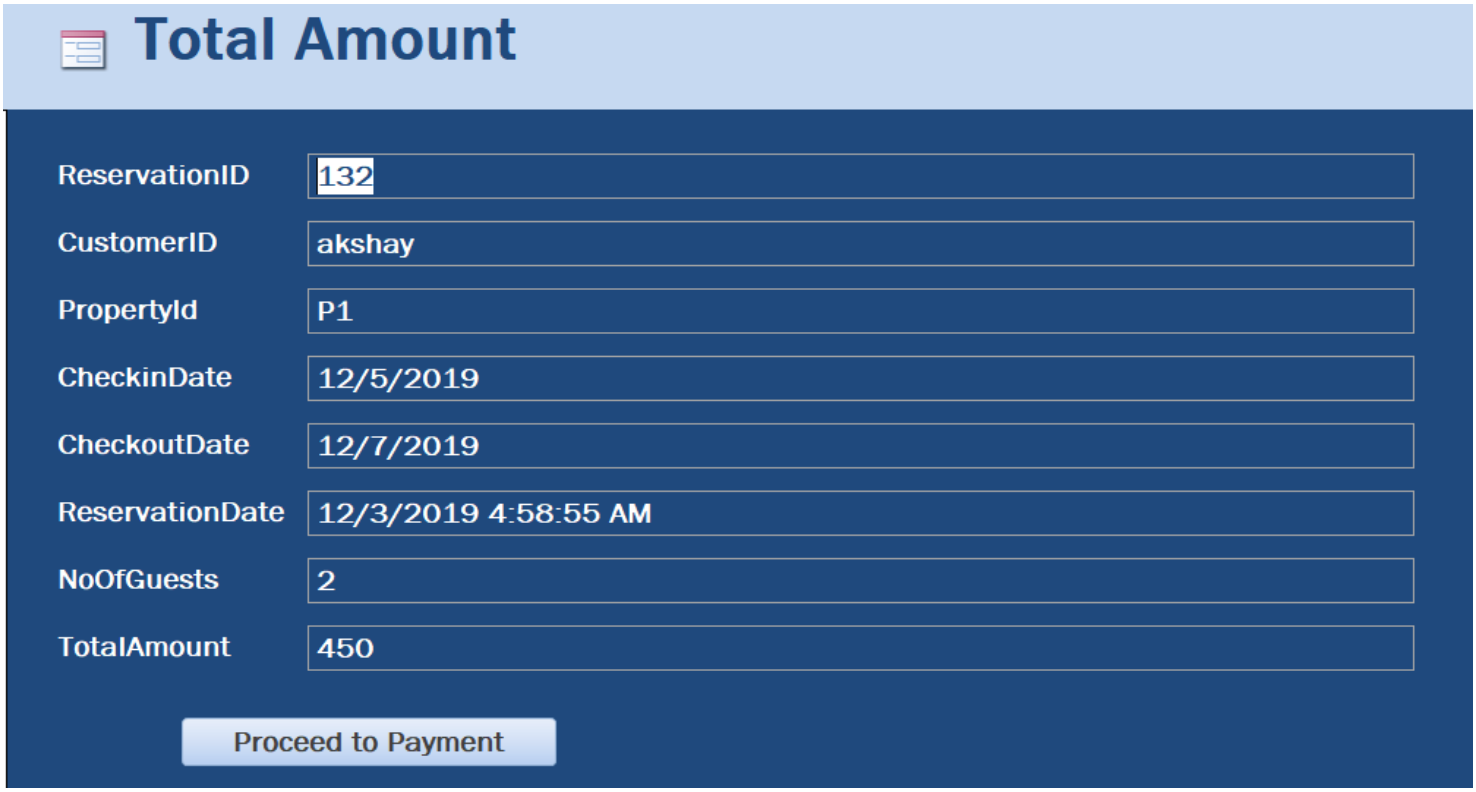


**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

If the user clicks on add services, it will load a page where the user can add services to his reservation:

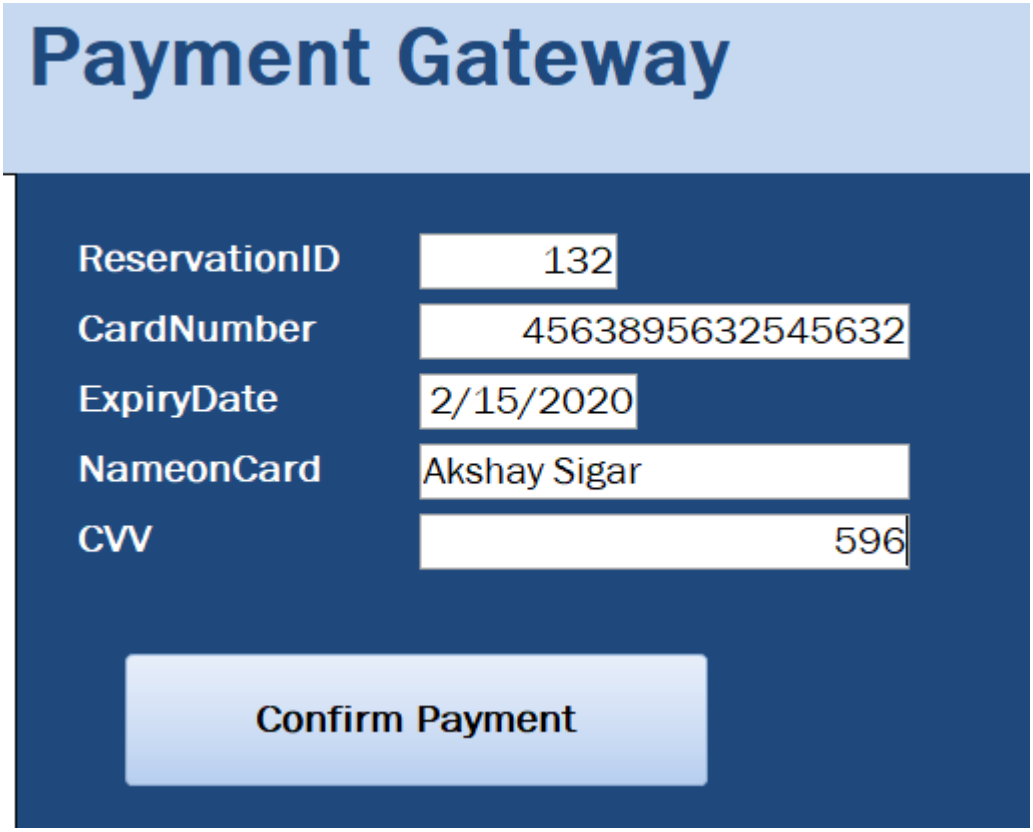



The User have two options here, he can either add more services by clicking on **ADD MORE SERVICES** button or else the user can click on **ADD SERVICE AND FIND TOTAL** button to load a summary page with total amount after adding the cost of the selected services.

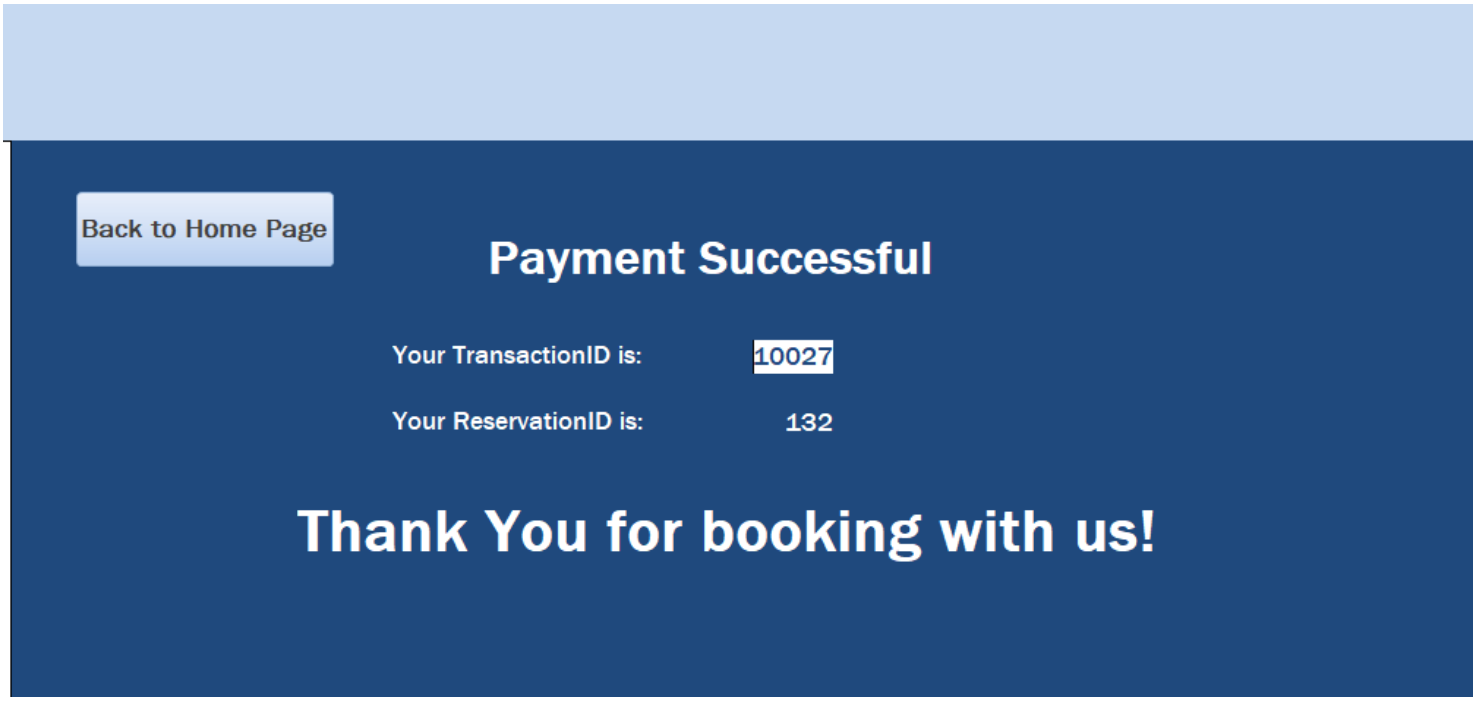


**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

As the User clicks on the **PROCEED TO PAYMENT** button, it will load a page where the user can add his credit card details to complete the payment.

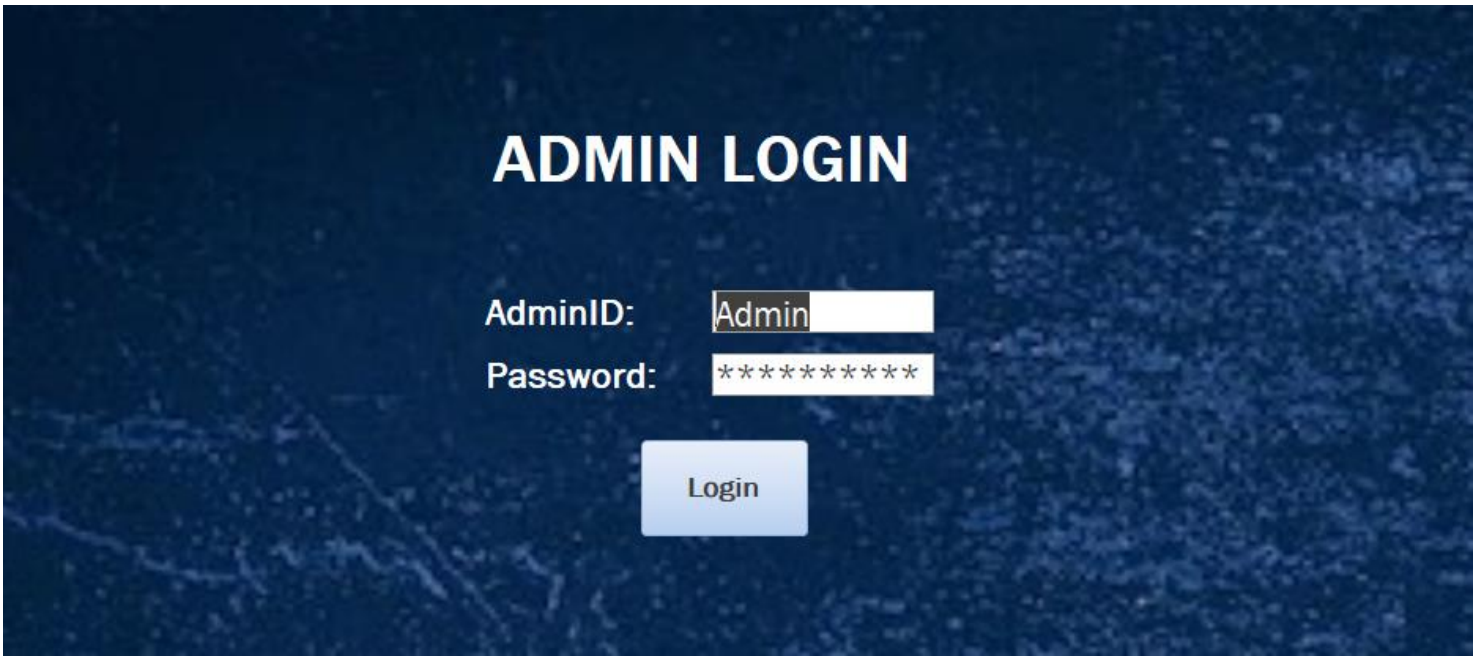



This will complete the entire process and the user will reach the final page where he can find his transaction number and his reservation number.

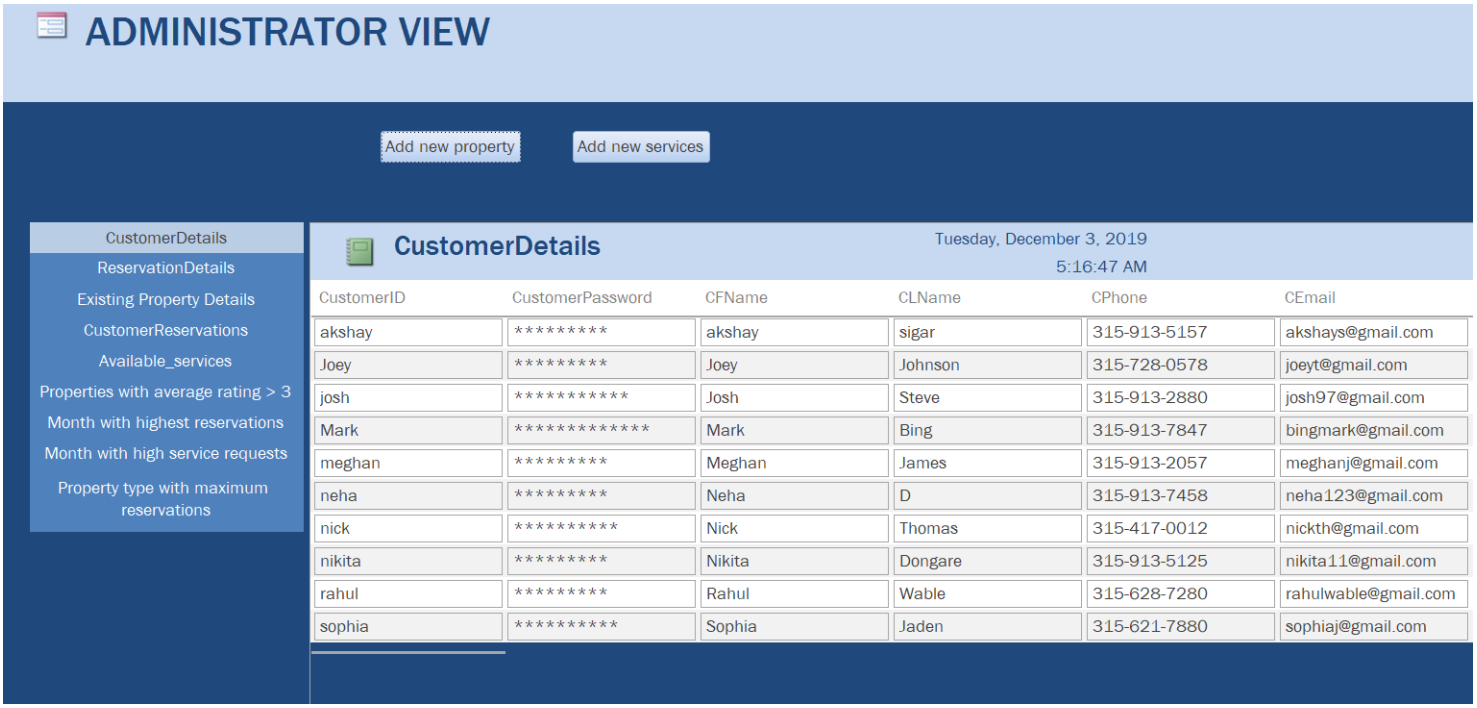


**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

**Scenario 2: Admin Login**
The admin enters the credentials using the login form.



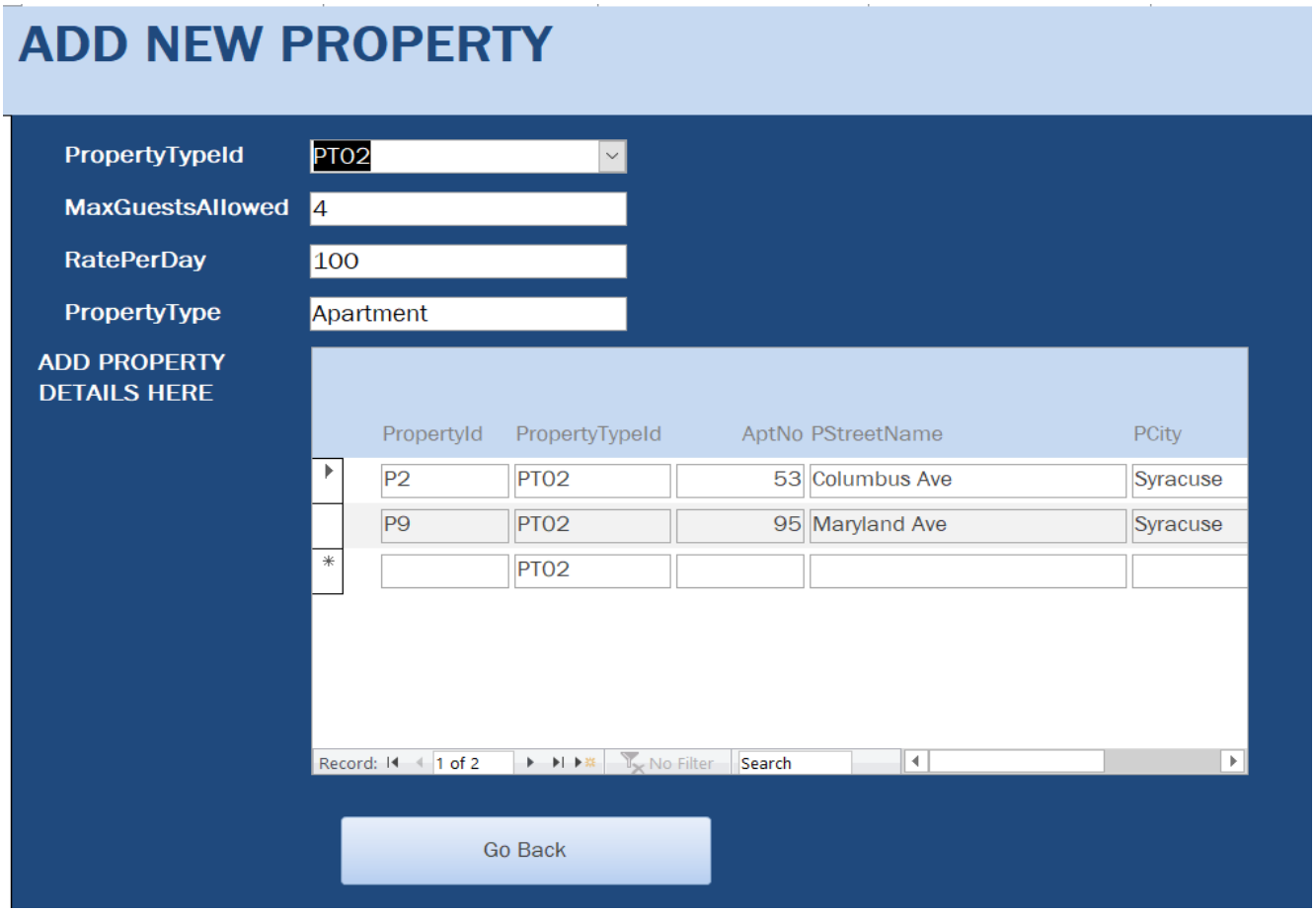

The ID and password are validated in the database and on successful login, the admin is redirected to a different dashboard. It contains different admin functions to add data to the database. The navigation bar displays all the admin functionalities.




| CustomerID | CustomerPassword | CFName | CLName | CPhone | CEmail |
|---|---|---|---|---|---|
| akshay | ********* | akshay | sigar | 315-913-5157 | akshays@gmail.com |
| Joey | ********* | Joey | Johnson | 315-728-0578 | joeyt@gmail.com |
| josh | ********** | Josh | Steve | 315-913-2880 | josh97@gmail.com |
| Mark | ********** | Mark | Bing | 315-913-7847 | bingmark@gmail.com |
| meghan | ********* | Meghan | James | 315-913-2057 | meghanj@gmail.com |
| neha | ********* | Neha | D | 315-913-7458 | neha123@gmail.com |
| nick | ********** | Nick | Thomas | 315-417-0012 | nickth@gmail.com |
| nikita | ********* | Nikita | Dongare | 315-913-5125 | nikita11@gmail.com |
| rahul | ********* | Rahul | Wable | 315-628-7280 | rahulwable@gmail.com |
| sophia | ********** | Sophia | Jaden | 315-621-7880 | sophiaj@gmail.com |

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

**1. Add New Property:**
The Admin can add new properties on the database by clicking on the **ADD NEW PROPERTY** button



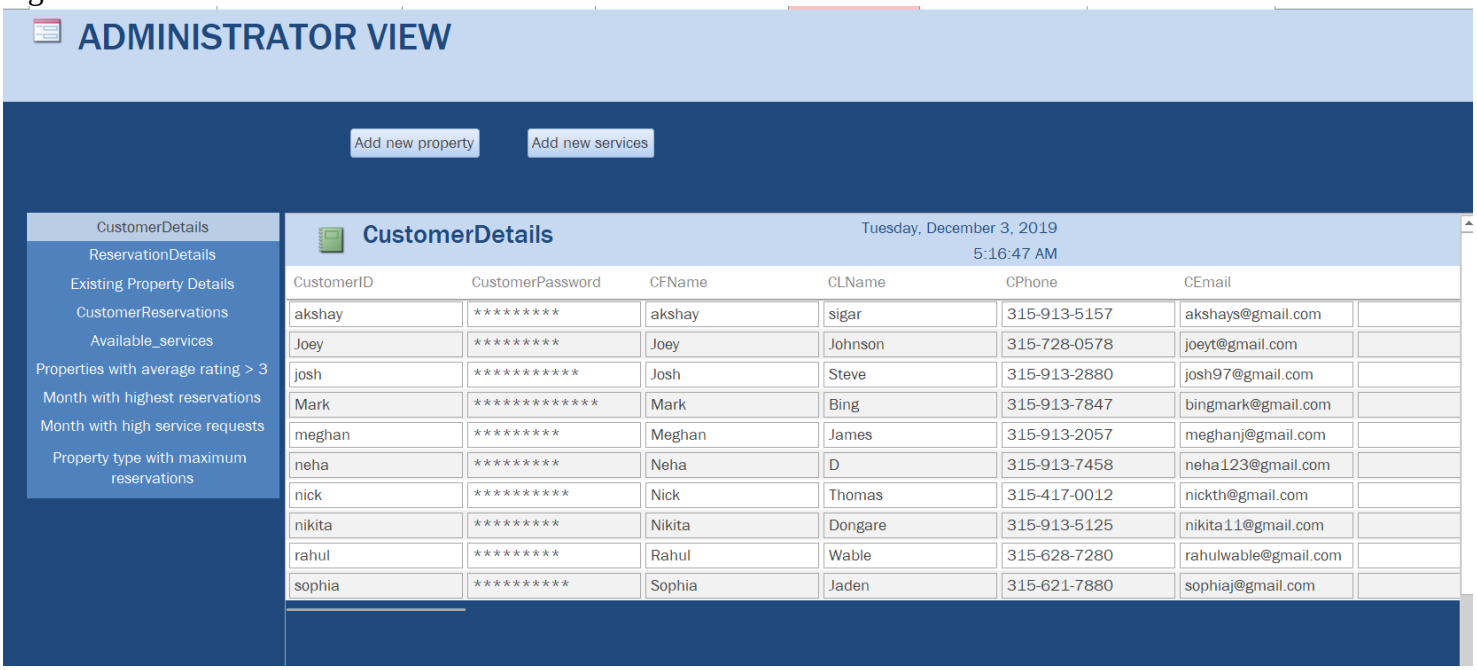

**2. Add New Services:** The admin can add new services into the database by using it.

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

## REPORTS:

### 1. Customer Details:

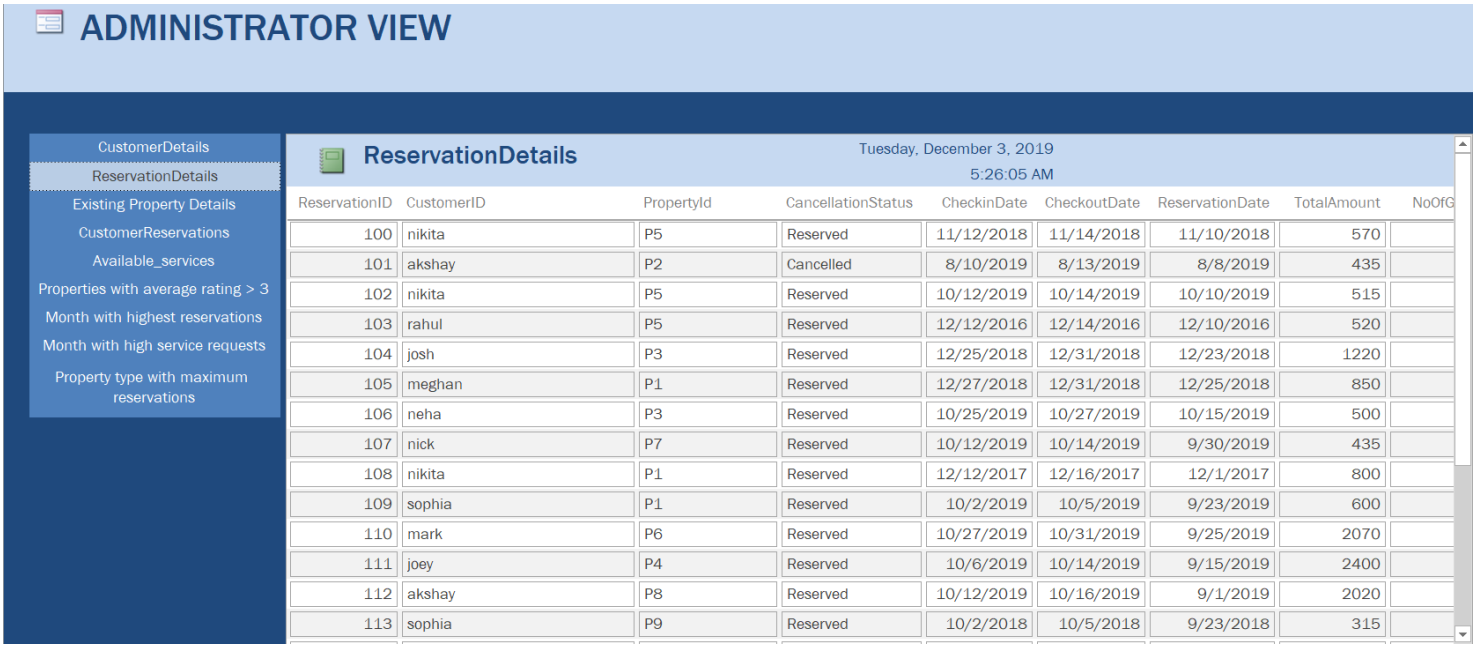
This report displays all the registered users in our system. It provides all the information about each of the registered user.




| CustomerID | CustomerPassword | CFName | CLName | CPhone | CEmail |
|---|---|---|---|---|---|
| akshay | ******** | akshay | sigar | 315-913-5157 | akshays@gmail.com |
| Joey | ******** | Joey | Johnson | 315-728-0578 | joeyt@gmail.com |
| josh | ********** | Josh | Steve | 315-913-2880 | josh97@gmail.com |
| Mark | ********** | Mark | Bing | 315-913-7847 | bingmark@gmail.com |
| meghan | ******** | Meghan | James | 315-913-2057 | meghanj@gmail.com |
| neha | ******** | Neha | D | 315-913-7458 | neha123@gmail.com |
| nick | ********** | Nick | Thomas | 315-417-0012 | nickth@gmail.com |
| nikita | ******** | Nikita | Dongare | 315-913-5125 | nikita11@gmail.com |
| rahul | ******** | Rahul | Wable | 315-628-7280 | rahulwable@gmail.com |
| sophia | ********** | Sophia | Jaden | 315-621-7880 | sophiaj@gmail.com |

### 2. Reservation Details Report:

This report displays all the details about each of the reservation made on the system.

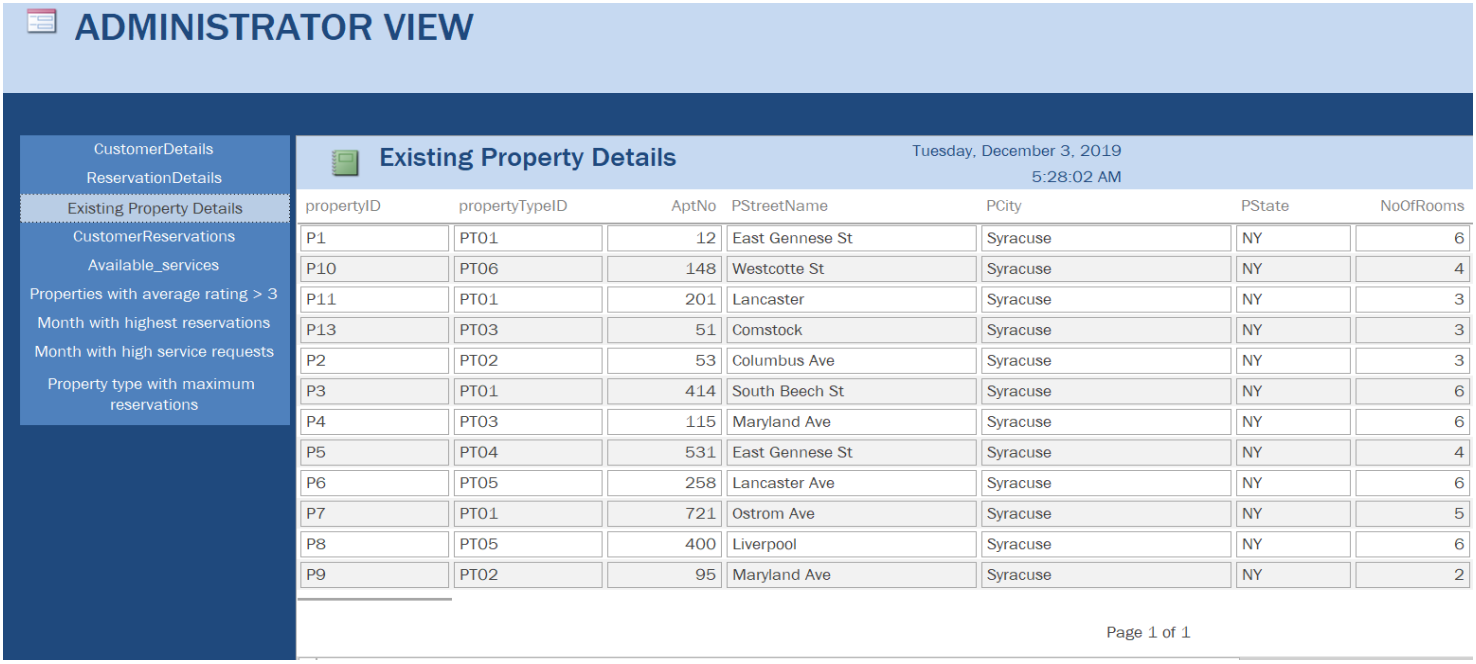



| ReservationID | CustomerID | PropertyId | CancellationStatus | CheckinDate | CheckoutDate | ReservationDate | TotalAmount | NoOfG |
|---|---|---|---|---|---|---|---|---|
| 100 | nikita | P5 | Reserved | 11/12/2018 | 11/14/2018 | 11/10/2018 | 570 | |
| 101 | akshay | P2 | Cancelled | 8/10/2019 | 8/13/2019 | 8/8/2019 | 435 | |
| 102 | nikita | P5 | Reserved | 10/12/2019 | 10/14/2019 | 10/10/2019 | 515 | |
| 103 | rahul | P5 | Reserved | 12/12/2016 | 12/14/2016 | 12/10/2016 | 520 | |
| 104 | josh | P3 | Reserved | 12/25/2018 | 12/31/2018 | 12/23/2018 | 1220 | |
| 105 | meghan | P1 | Reserved | 12/27/2018 | 12/31/2018 | 12/25/2018 | 850 | |
| 106 | neha | P3 | Reserved | 10/25/2019 | 10/27/2019 | 10/15/2019 | 500 | |
| 107 | nick | P7 | Reserved | 10/12/2019 | 10/14/2019 | 9/30/2019 | 435 | |
| 108 | nikita | P1 | Reserved | 12/12/2017 | 12/16/2017 | 12/1/2017 | 800 | |
| 109 | sophia | P1 | Reserved | 10/2/2019 | 10/5/2019 | 9/23/2019 | 600 | |
| 110 | mark | P6 | Reserved | 10/27/2019 | 10/31/2019 | 9/25/2019 | 2070 | |
| 111 | joey | P4 | Reserved | 10/6/2019 | 10/14/2019 | 9/15/2019 | 2400 | |
| 112 | akshay | P8 | Reserved | 10/12/2019 | 10/16/2019 | 9/1/2019 | 2020 | |
| 113 | sophia | P9 | Reserved | 10/2/2018 | 10/5/2018 | 9/23/2018 | 315 | |

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

### 3. Existing Property Report:

This report displays details about the property in our database. It also provides details about each of the property type present in the system.



ADMINISTRATOR VIEW

CustomerDetails
ReservationDetails
Existing Property Details
CustomerReservations
Available_services
Properties with average rating > 3
Month with highest reservations
Month with high service requests
Property type with maximum reservations

Existing Property Details — Tuesday, December 3, 2019 5:28:02 AM

| propertyID | propertyTypeID | AptNo | PStreetName | PCity | PState | NoOfRooms |
|---|---|---|---|---|---|---|
| P1 | PT01 | 12 | East Gennese St | Syracuse | NY | 6 |
| P10 | PT06 | 148 | Westcotte St | Syracuse | NY | 4 |
| P11 | PT01 | 201 | Lancaster | Syracuse | NY | 3 |
| P13 | PT03 | 51 | Comstock | Syracuse | NY | 3 |
| P2 | PT02 | 53 | Columbus Ave | Syracuse | NY | 3 |
| P3 | PT01 | 414 | South Beech St | Syracuse | NY | 6 |
| P4 | PT03 | 115 | Maryland Ave | Syracuse | NY | 6 |
| P5 | PT04 | 531 | East Gennese St | Syracuse | NY | 4 |
| P6 | PT05 | 258 | Lancaster Ave | Syracuse | NY | 6 |
| P7 | PT01 | 721 | Ostrom Ave | Syracuse | NY | 5 |
| P8 | PT05 | 400 | Liverpool | Syracuse | NY | 6 |
| P9 | PT02 | 95 | Maryland Ave | Syracuse | NY | 2 |

Page 1 of 1

### 4. Most Valuable Customer Report:

This report provides information about the number of reservations made by a customer on our system.

ADMINISTRATOR VIEW

CustomerDetails
ReservationDetails
Existing Property Details
CustomerReservations
Available_services
Properties with average rating > 3
Month with highest reservations
Month with high service requests
Property type with maximum reservations

CustomerReservations — Tuesday, December 3, 2019 5:28:37 AM

| CustomerId | CFName | CLName | NoOfreservations |
|---|---|---|---|
| nikita | Nikita | Dongare | 6 |
| akshay | akshay | sigar | 4 |
| Mark | Mark | Bing | 2 |
| nick | Nick | Thomas | 2 |
| sophia | Sophia | Jaden | 2 |
| rahul | Rahul | Wable | 1 |
| meghan | Meghan | James | 1 |
| neha | Neha | D | 1 |
| Joey | Joey | Johnson | 1 |
| josh | Josh | Steve | 1 |

Page 1 of 1

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

# 9. TRIGGERS AND VIEWS:

In order to simplify task of calculating and displaying the total amount when the user selects a particular property with or without services, we created 2 triggers in SQL server.

**Without Trigger:**

Once a user reserves the property and requests for services, the admin would have to calculate the total amount with services or without services based on the number of days and rate of the property as well as the service rate.

This is a redundant process that the admin would have to do it every time. To avoid this, a trigger would save the time as it would be automatically executed every time a property and services are reserved.

**With Trigger:**

The below trigger is written in SQL and deployed on SQL Server. The trigger is created to execute when an insert command is executed on Reservation table.

Once the trigger is executed, update command would be executed. Using checkindate and checkoutdate, the number of days is calculated and multiplied with the rate per day of the property reserved. Totalamt trigger calculates the totalamount when no services are selected and updates the totalamount in the reservation table. Totalamountwservices trigger calculates the totalamount when services are requested and updates the totalamount in the reservation table.

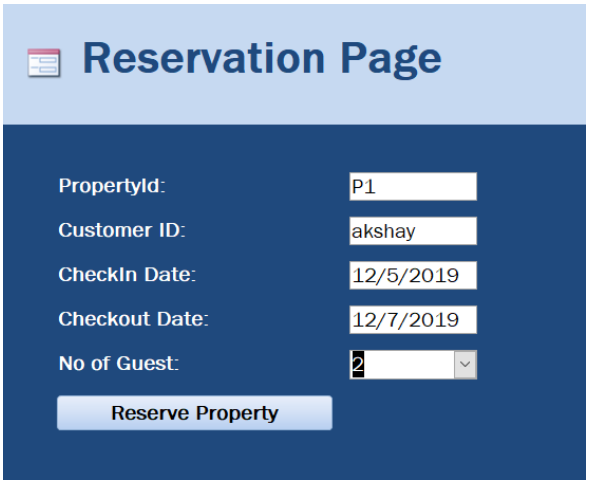
**Code:**

```
--Trigger 1:
CREATE TRIGGER totalamt
ON reservation
AFter INSERT
AS
BEGIN

   UPDATE reservation
   SET TotalAmount=(Select PT.Rateperday*(DATEDIFF(d, i.CheckinDate, i.CheckoutDate))
FROM inserted i
JOIN property P
ON P.PropertyId=i.propertyid
JOIN property_type PT
ON PT.propertytypeId= P.propertytypeId
) where reservation.ReservationID=(select ReservationID from inserted)
END
```

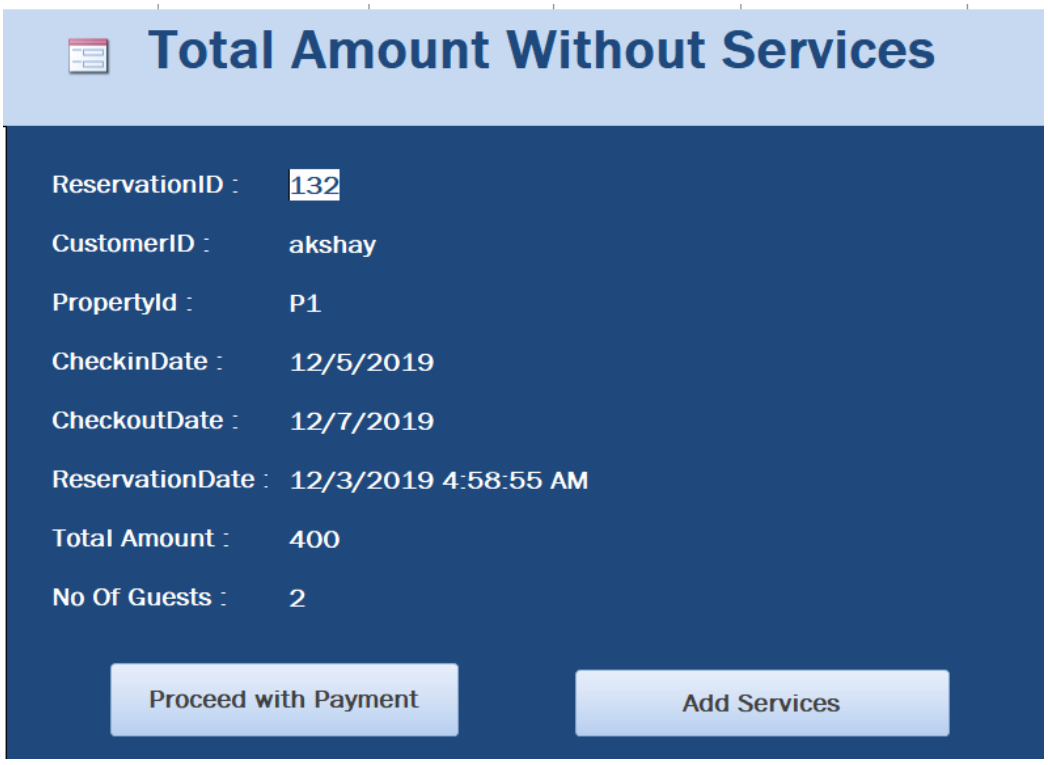
**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

**Execution:**

1. When the user reserves a property as shown below without adding any services, the totalamt trigger is executed.




2. The totalamt trigger updates the totalamount value in Reservation table and creates a summary of the reservation with the updated value of total amount.

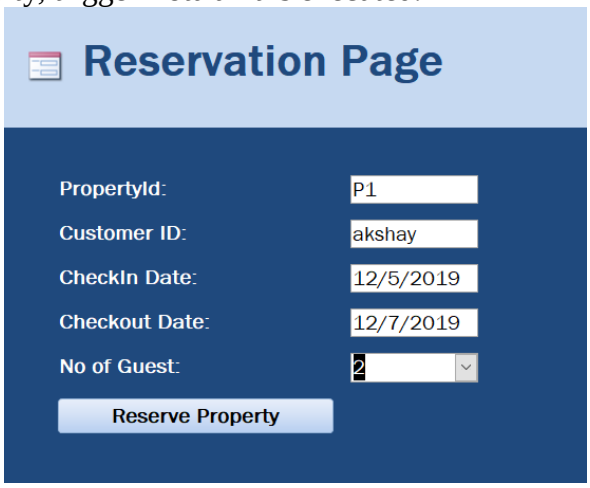


**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

```
--Trigger2

CREATE TRIGGER totalamtwservices
ON reservation_services
AFter INSERT
AS
BEGIN

   UPDATE reservation
   SET TotalAmount=(
        Select x.total+y.total from

(Select distinct PT.Rateperday*(DATEDIFF(d, i.CheckinDate, i.CheckoutDate))as total
FROM reservation i
JOIN
inserted rs
ON
i.reservationID=rs.reservationID
JOIN
property P
ON P.PropertyId=i.propertyid
JOIN property_type PT
ON PT.propertytypeId= P.propertytypeId)y,

(select sum(a.ServiceCost)as total from available_services a
JOIN
reservation_services rs
ON a.serviceID=rs.serviceID
where rs.reservationID=(select ReservationID from inserted)
group by rs.reservationID)x

) where reservation.ReservationID=(select ReservationID from inserted)
END
```

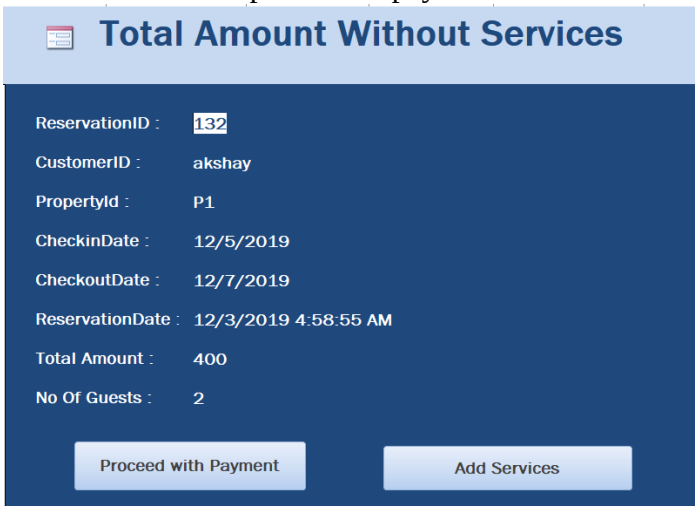
**EXECUTION:**

1. When the user reserves a property, trigger Totalamt is executed.



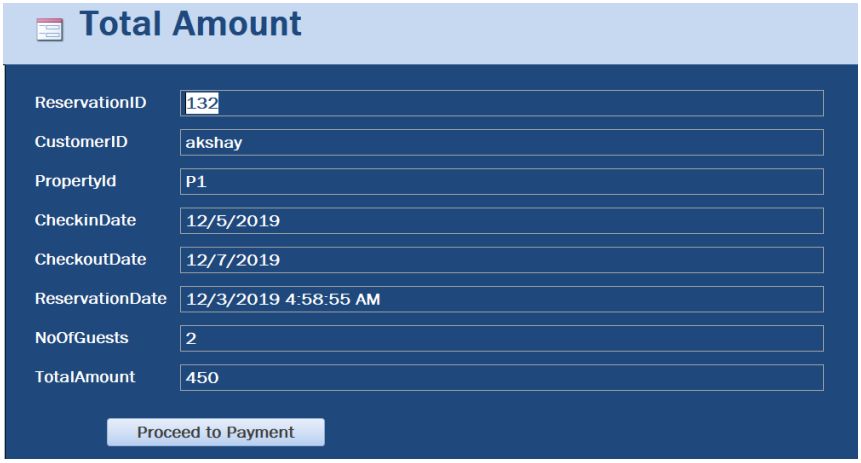
**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

2. The user now has an option to add services or proceed to payment as shown below.




3. If the user chooses to add services, then the following page pop ups. The user can add multiple services and then choose the add services and find total button. When this button is clicked, the trigger Totalamountwservices is executed.


4. The trigger updates the Totalamount with service cost in the reservation table and the following reservation summary is generated.




Thus, the trigger was executed successfully, and the totalamount was added, thus saving the redundant work that had to be done by the admin.

**Akshay Jaipal Sigar- 865154216**
**Nikita Pralhad Dongare- 651838408**

## Views:

The view PropertyPresent is created to join the two tables property and property_type in order to related the properties with their property_type.

```
Create View PropertyPresent as(
Select
p.propertyID,p.propertyTypeID,p.AptNo,p.PStreetName,p.PCity,p.PState,p.NoOfRooms,pt.MaxGuestsAllowed,
pt.RateperDay,pt.PropertyType
from property p
JOIN
Property_type pt
ON
p.propertyTypeID=pt.propertyTypeID)
```

**Output(Datasheet):**

Results Messages

| | propertyID | propertyTypeID | AptNo | PStreetName | PCity | PState | NoOfRooms | MaxGuestsAllowed | RateperDay | PropertyType |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | P1 | PT01 | 12 | East Gennese St | Syracuse | NY | 6 | 8 | 200 | Villa |
| 2 | P10 | PT06 | 148 | Westcotte St | Syracuse | NY | 4 | 5 | 200 | Apartment |
| 3 | P11 | PT01 | 201 | Lancaster | Syracuse | NY | 3 | 8 | 200 | Villa |
| 4 | P13 | PT03 | 51 | Comstock | Syracuse | NY | 3 | 6 | 300 | Bungalow |
| 5 | P2 | PT02 | 53 | Columbus Ave | Syracuse | NY | 3 | 4 | 100 | Apartment |
| 6 | P3 | PT01 | 414 | South Beech St | Syracuse | NY | 6 | 8 | 200 | Villa |
| 7 | P4 | PT03 | 115 | Maryland Ave | Syracuse | NY | 6 | 6 | 300 | Bungalow |
| 8 | P5 | PT04 | 531 | East Gennese St | Syracuse | NY | 4 | 5 | 250 | Cottage |
| 9 | P6 | PT05 | 258 | Lancaster Ave | Syracuse | NY | 6 | 10 | 500 | Villa |
| 10 | P7 | PT01 | 721 | Ostrom Ave | Syracuse | NY | 5 | 8 | 200 | Villa |
| 11 | P8 | PT05 | 400 | Liverpool | Syracuse | NY | 6 | 10 | 500 | Villa |
| 12 | P9 | PT02 | 95 | Maryland Ave | Syracuse | NY | 2 | 4 | 100 | Apartment |

**Output (Report):**

Admin view | dbo_PropertyPresent

**Existing Property Details**

Tuesday, December 3, 2019
3:59:20 PM

| propertyID | propertyTypeID | AptNo | PStreetName | PCity | PState | NoOfRooms | MaxGuestsAllowed | RateperDay | PropertyType |
|---|---|---|---|---|---|---|---|---|---|
| P1 | PT01 | 12 | East Gennese St | Syracuse | NY | 6 | 8 | 200 | Villa |
| P10 | PT06 | 148 | Westcotte St | Syracuse | NY | 4 | 5 | 200 | Apartment |
| P11 | PT01 | 201 | Lancaster | Syracuse | NY | 3 | 8 | 200 | Villa |
| P13 | PT03 | 51 | Comstock | Syracuse | NY | 3 | 6 | 300 | Bungalow |
| P2 | PT02 | 53 | Columbus Ave | Syracuse | NY | 3 | 4 | 100 | Apartment |
| P3 | PT01 | 414 | South Beech St | Syracuse | NY | 6 | 8 | 200 | Villa |
| P4 | PT03 | 115 | Maryland Ave | Syracuse | NY | 6 | 6 | 300 | Bungalow |
| P5 | PT04 | 531 | East Gennese St | Syracuse | NY | 4 | 5 | 250 | Cottage |
| P6 | PT05 | 258 | Lancaster Ave | Syracuse | NY | 6 | 10 | 500 | Villa |
| P7 | PT01 | 721 | Ostrom Ave | Syracuse | NY | 5 | 8 | 200 | Villa |
| P8 | PT05 | 400 | Liverpool | Syracuse | NY | 6 | 10 | 500 | Villa |
| P9 | PT02 | 95 | Maryland Ave | Syracuse | NY | 2 | 4 | 100 | Apartment |

Page 1 of 1