NEU Healthcare Management System

Divya Bhupinder Suri - 002737591
Kirubagar Thiagarajan - 002745789
Mansi Dabriwal - 002774233
Siddhant Burse - 002761979
Vinayak Kiranji - 002726550

Table of contents

01 Problem Statement

02 Objective & Goals

03 ER Diagram

04 Server-Side Modules

05 Reports & Visualization

06 Future Scope

01 Problem Statement

Develop a comprehensive database system to efficiently manage healthcare for students, faculty, and staff at the university. Streamline tracking processes, improve stock management, and enhance patient care by providing accurate and up-to-date health-related information.

02 Objective & Goals

Objective

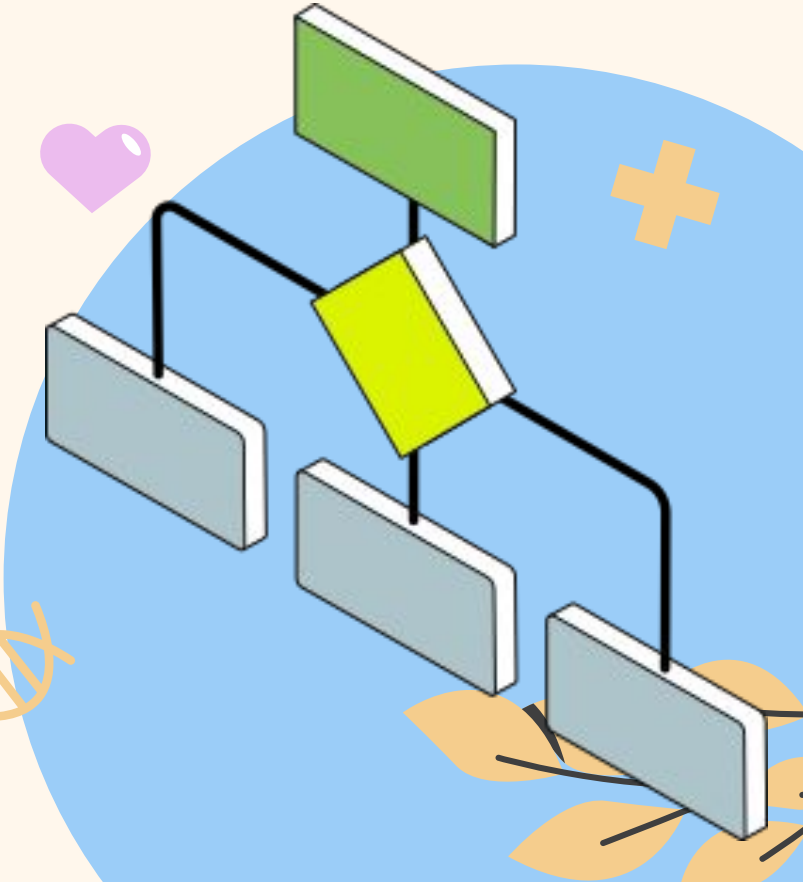
- To develop a comprehensive healthcare management system for our university, with a centralized database for storing and managing health-related information of students, faculty, and staff.
- To streamline the process of tracking doctor availability, student vaccinations, and health-related information, and improve the efficiency of the healthcare system.
- To create an easily accessible system for authorized personnel to access accurate and up-to-date health-related information of patients, and enhance the quality of patient care.
- To enable users to search for and enquire about stock and availability of medications, vaccines, and tests in the university's pharmacy, vaccine centers, and test centers.
- To design a system that allows for efficient tracking of a student's health through their appointment history, as well as tracking of expenses related to their healthcare.
- To develop a system for efficient management of insurance claims and statuses, and enable users and the university to keep track of them easily.

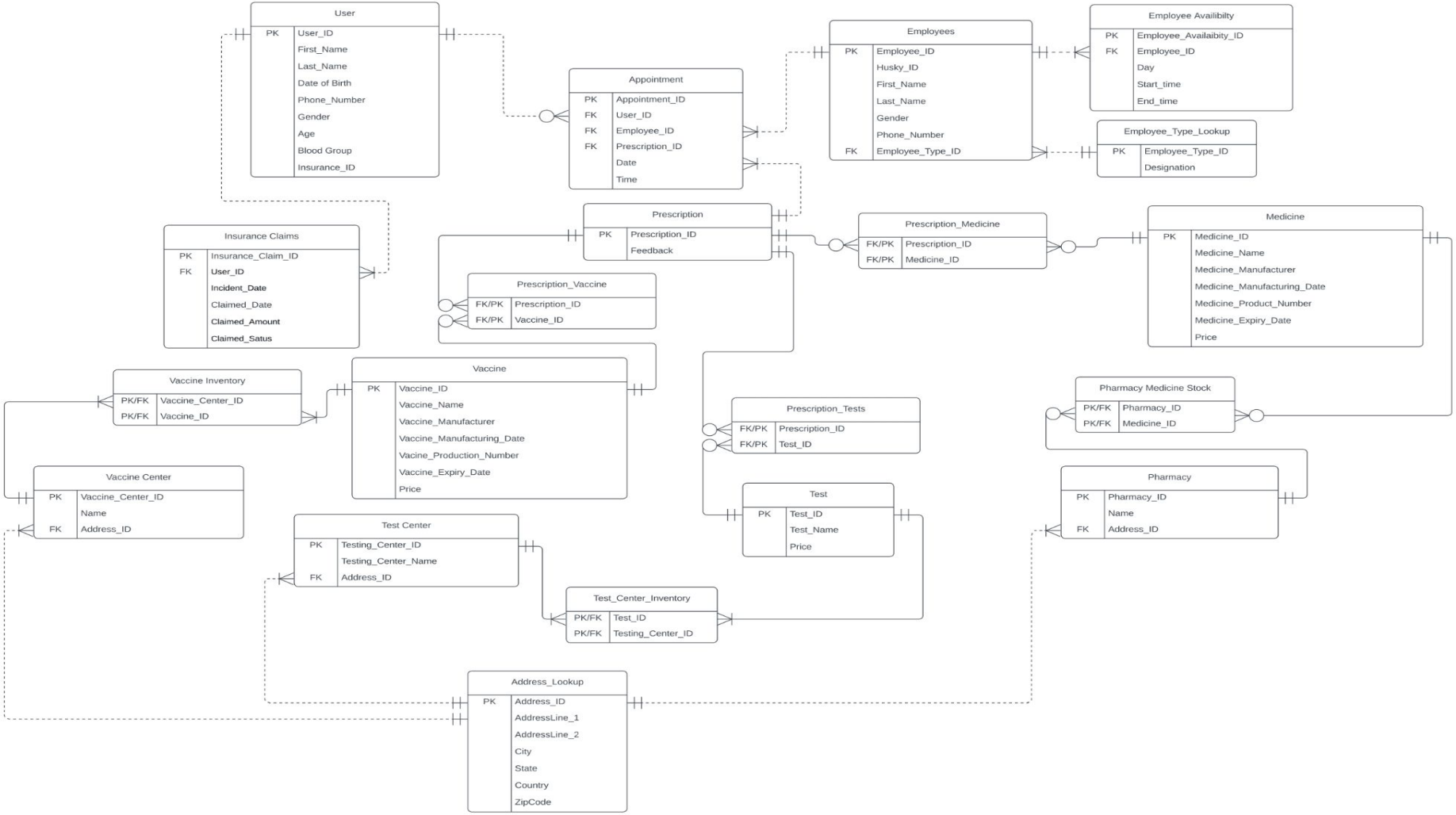
Goals

- Develop a user-friendly interface that allows easy navigation and interaction with the healthcare management system.
- Design the database schema and tables using Crow's Foot notation and establish proper relationships between tables.
- Implement CRUD (Create, Read, Update, Delete) functionality for managing health-related data of users, appointments, employees, prescription, insurance claims, etc.
- Provide search functionality for users to search for medications, vaccines, and tests, and enable users to book appointments with available doctors.
- Develop a reporting feature for generating insights and trends related to healthcare data, such as employee availability, patient history, and expenses.
- Test the system thoroughly to ensure proper functionality and reliability, and address any issues or bugs found during testing.

03

ERDiagram




User
PK User_ID
First_Name
Last_Name
Date of Birth
Phone_Number
Gender
Age
Blood Group
Insurance_ID
Appointment
PK Appointment_ID
FK User_ID
FK Employee_ID
FK Prescription_ID
Date
Time
Employees
PK Employee_ID
Husky_ID
First_Name
Last_Name
Gender
Phone_Number
FK Employee_Type_ID
Employee Availibilty
PK Employee_Availaibity_ID
FK Employee_ID
Day
Start_time
End_time
Employee_Type_Lookup
PK Employee_Type_ID
Designation
Insurance Claims
PK Insurance_Claim_ID
FK User_ID
Incident_Date
Claimed_Date
Claimed_Amount
Claimed_Satus
Prescription
PK Prescription_ID
Feedback
Prescription_Medicine
FK/PK Prescription_ID
FK/PK Medicine_ID
Medicine
PK Medicine_ID
Medicine_Name
Medicine_Manufacturer
Medicine_Manufacturing_Date
Medicine_Product_Number
Medicine_Expiry_Date
Price
Prescription_Vaccine
FK/PK Prescription_ID
FK/PK Vaccine_ID
Vaccine Inventory
PK/FK Vaccine_Center_ID
PK/FK Vaccine_ID
Vaccine
PK Vaccine_ID
Vaccine_Name
Vaccine_Manufacturer
Vaccine_Manufacturing_Date
Vacine_Production_Number
Vaccine_Expiry_Date
Price
Prescription_Tests
FK/PK Prescription_ID
FK/PK Test_ID
Pharmacy Medicine Stock
PK/FK Pharmacy_ID
PK/FK Medicine_ID
Vaccine Center
PK Vaccine_Center_ID
Name
FK Address_ID
Test
PK Test_ID
Test_Name
Price
Pharmacy
PK Pharmacy_ID
Name
FK Address_ID
Test Center
PK Testing_Center_ID
Testing_Center_Name
FK Address_ID
Test_Center_Inventory
PK/FK Test_ID
PK/FK Testing_Center_ID
Address_Lookup
PK Address_ID
AddressLine_1
AddressLine_2
City
State
Country
ZipCode

04

Database Implementation & Server Side Module

Table Creation & Insertion

```sql
--User Table--
CREATE TABLE [User] (
 User_ID INT PRIMARY KEY NOT NULL,
 First_Name VARCHAR (30) NOT NULL,
 Last_Name VARCHAR (30) NOT NULL,
 Date_of_Birth DATE NOT NULL,
 Phone_Number VARCHAR (30) NOT NULL,
 Gender VARCHAR (30) NOT NULL,
 Blood_Group VARCHAR (5) NOT NULL,
 Insurance_ID VARCHAR (30) NOT NULL,
 Age AS DATEDIFF(hour,Date_of_Birth,GETDATE())/8766);



--Employee Type Lookup Table--
CREATE TABLE Employee_Type_Lookup(
Employee_Type_ID INT PRIMARY KEY NOT NULL,
Designation VARCHAR (30) NOT NULL);

--Employee Table--
CREATE TABLE Employees(
Employee_ID INT PRIMARY KEY NOT NULL,
Husky_ID VARCHAR (30) NOT NULL,
First_Name VARCHAR (30) NOT NULL,
Last_Name VARCHAR (30) NOT NULL,
Gender VARCHAR (30) NOT NULL,
Phone_Number VARCHAR (30) NOT NULL,
Employee_Type_ID INT NOT NULL
FOREIGN KEY (Employee_Type_ID) REFERENCES Employee_Type_Lookup(Employee_Type_ID),
);
```

110 %

```sql
--Prescription Table Insertion--
INSERT INTO Prescription (Prescription_ID, Feedback)
VALUES (1111, 'Prescription verified and processed. Medication delivery scheduled for tomorrow.'),
       (1121, 'Patient needs to take proper test for correct diagnosis.'),
       (1131, '101 Degree Celsius Fever. Need to take proper rest and visit again after a week'),
       (1141, 'Prescription filled and medication given to patient.'),
       (1151, 'Patient needs to rest and complete medication cycle'),
       (1161, 'X-ray scheduled for patient.'),
       (1171, 'Patient is all well. Needs to take rest and visit after a week.'),
       (1181, 'High fever and throat infection. Prescribed medication'),
       (1191, 'Migraine Problem. Patient needs to sleep and take rest'),
       (1110, 'Breathing problem because of pollution. Prescribed inhaler');


--Appointment Table Insertion--
DROP PROCEDURE IF EXISTS dbo.Appointment_Table_Insertion;
GO
CREATE PROCEDURE Appointment_Table_Insertion
    @Appointment_ID INT,
    @User_ID INT,
    @Employee_ID INT,
    @Prescription_ID INT,
    @Date DATE,
    @Time TIME

AS
BEGIN
    INSERT INTO Appointment(Appointment_ID, User_ID, Employee_ID, Prescription_ID, [Date],[Time])
    VALUES (@Appointment_ID, @User_ID, @Employee_ID, @Prescription_ID, @Date,@Time);
END;
GO
```

```sql
EXEC Appointment_Table_Insertion 1,1001,1111,1111,'01-02-2023','4:00PM';
EXEC Appointment_Table_Insertion 2,1002,1111,1121,'02-03-2023','5:00PM';
EXEC Appointment_Table_Insertion 3,1003,1111,1131,'03-04-2023','6:00PM';
EXEC Appointment_Table_Insertion 4,1004,1118,1141,'04-05-2023','7:00PM';
EXEC Appointment_Table_Insertion 5,1005,1118,1151,'05-06-2023','8:00AM';
EXEC Appointment_Table_Insertion 6,1006,1118,1161,'06-07-2023','9:00AM';
EXEC Appointment_Table_Insertion 7,1007,1112,1171,'07-08-2023','10:00AM';
EXEC Appointment_Table_Insertion 8,1008,1112,1181,'08-09-2023','1:00PM';
EXEC Appointment_Table_Insertion 9,1009,1119,1191,'09-10-2023','2:00PM';
EXEC Appointment_Table_Insertion 10,1010,1119,1110,'10-01-2023','3:00PM';
```

Constraints

```sql
USE GROUP_16;
                                                    --Table Level Check Based Function for USERS TABLE--
--Adding Function Based Column Check on USER--

ALTER TABLE [User]
DROP CONSTRAINT IF EXISTS CheckValidFirstNames;

ALTER TABLE [User]
DROP CONSTRAINT IF EXISTS CheckValidLastNames;

DROP FUNCTION IF EXISTS dbo.CheckValidFirstName;
DROP FUNCTION IF EXISTS dbo.CheckValidLastName;

GO


--Creating a function that performs a Column Check on First Name to check if it contians only Alphabets.
CREATE FUNCTION CheckValidFirstName(@FName varchar(30))
RETURNS smallint
AS
BEGIN
   DECLARE @Count smallint=0;
   SELECT @Count = COUNT(First_Name)
          FROM [User]
          WHERE First_Name = @FName
          AND First_Name LIKE '%[^a-zA-Z]%' OR First_Name = '';
   RETURN @Count;
END;

GO

--Creating a function that performs a Column Check on Last Name to check if it contians only Alphabets.
CREATE FUNCTION CheckValidLastName(@LName varchar(30))
RETURNS smallint
AS
BEGIN
   DECLARE @Count smallint=0;
   SELECT @Count = COUNT(Last_Name)
          FROM [User]
          WHERE Last_Name = @LName
          AND Last_Name LIKE '%[^a-zA-Z]%' OR Last_Name = '';
   RETURN @Count;
END;

GO
```

```sql
--Adding the constraints to the User table--
ALTER TABLE [User] ADD CONSTRAINT CheckValidFirstNames CHECK (dbo.CheckValidFirstName(First_Name) = 0);
ALTER TABLE [User] ADD CONSTRAINT CheckValidLastNames CHECK (dbo.CheckValidLastName(Last_Name) = 0);


--Testing the Function By Inserting Invalid First Names--
INSERT INTO [User](User_ID, First_Name, Last_Name, Date_of_Birth, Phone_Number, Gender, Blood_Group, Insurance_ID)
VALUES
(2008,'','Thiagarajan','04-12-1997','8248598448','Male','A+','A201');

INSERT INTO [User](User_ID, First_Name, Last_Name, Date_of_Birth, Phone_Number, Gender, Blood_Group, Insurance_ID)
VALUES
(2008,'Kiruba6a87','Thiagarajan','04-12-1997','8248598448','Male','A+','A201');

INSERT INTO [User](User_ID, First_Name, Last_Name, Date_of_Birth, Phone_Number, Gender, Blood_Group, Insurance_ID)
VALUES
(2008,'Kiruba%&*gar','Thiagarajan','04-12-1997','8248598448','Male','A+','A201');


--Testing the Function By Inserting Invalid Last Names--
INSERT INTO [User](User_ID, First_Name, Last_Name, Date_of_Birth, Phone_Number, Gender, Blood_Group, Insurance_ID)
VALUES
(2008,'Kirubagar','Thiagar6jan','04-12-1997','8248598448','Male','A+','A201');

INSERT INTO [User](User_ID, First_Name, Last_Name, Date_of_Birth, Phone_Number, Gender, Blood_Group, Insurance_ID)
VALUES
(2008,'Kirubagar','Thiaga#r%ajan','04-12-1997','8248598448','Male','A+','A201');

INSERT INTO [User](User_ID, First_Name, Last_Name, Date_of_Birth, Phone_Number, Gender, Blood_Group, Insurance_ID)
VALUES
(2008,'Kirubagar','','04-12-1997','8248598448','Male','A+','A201');
```

```sql
--Table Level Check Based Function for EMPLOYEES TABLE--

ALTER TABLE Employees
DROP CONSTRAINT IF EXISTS CheckValidEmployeeFirstNames;

ALTER TABLE Employees
DROP CONSTRAINT IF EXISTS CheckValidEmployeeLastNames;

ALTER TABLE Employees
DROP CONSTRAINT IF EXISTS CheckValidHusky_ID;

DROP FUNCTION IF EXISTS dbo.CheckValidEmployeeFirstName;
DROP FUNCTION IF EXISTS dbo.CheckValidEmployeeLastName;
DROP FUNCTION IF EXISTS dbo.CountInvalidHuskyIDs;

GO

--Creating a function that performs a Column Check on First Name to check if it contians only Alphabets.
CREATE FUNCTION CheckValidEmployeeFirstName(@FName varchar(30))
RETURNS smallint
AS
BEGIN
   DECLARE @Count smallint=0;
   SELECT @Count = COUNT(First_Name)
          FROM Employees
          WHERE First_Name = @FName
          AND First_Name LIKE '%[^a-zA-Z]%' OR First_Name = '';
   RETURN @Count;
END;


GO

--Creating a function that performs a Column Check on Last Name to check if it contians only Alphabets.
CREATE FUNCTION CheckValidEmployeeLastName(@LName varchar(30))
RETURNS smallint
AS
BEGIN
   DECLARE @Count smallint=0;
   SELECT @Count = COUNT(Last_Name)
          FROM Employees
          WHERE Last_Name = @LName
          AND Last_Name LIKE '%[^a-zA-Z]%' OR Last_Name = '';
   RETURN @Count;
END;

GO
```

```sql
--Adding the constraints to the Employee table--
ALTER TABLE Employees ADD CONSTRAINT CheckValidEmployeeFirstNames CHECK (dbo.CheckValidEmployeeFirstName(First_Name) = 0);
ALTER TABLE Employees ADD CONSTRAINT CheckValidEmployeeLastNames CHECK (dbo.CheckValidEmployeeLastName(Last_Name) = 0);

GO

--Testing the Function By Inserting Invalid First Names--
INSERT INTO Employees(Employee_ID,Husky_ID,First_Name,Last_Name,Gender,Phone_Number,Employee_Type_ID)
VALUES
(2222,'002774233','Dan1e1','Craig','Male','9876543210',001);

INSERT INTO Employees(Employee_ID,Husky_ID,First_Name,Last_Name,Gender,Phone_Number,Employee_Type_ID)
VALUES
(2222,'002774233','Dan$ie*l%','Mishra','Male','9876543210',001);

INSERT INTO Employees(Employee_ID,Husky_ID,First_Name,Last_Name,Gender,Phone_Number,Employee_Type_ID)
VALUES
(2222,'002774233','','Craig','Male','9876543210',001);


--Testing the Function By Inserting Invalid Last Names--
INSERT INTO Employees(Employee_ID,Husky_ID,First_Name,Last_Name,Gender,Phone_Number,Employee_Type_ID)
VALUES
(2222,'002774233','Daniel','Cra19','Male','9876543210',001);

INSERT INTO Employees(Employee_ID,Husky_ID,First_Name,Last_Name,Gender,Phone_Number,Employee_Type_ID)
VALUES
(2222,'002774233','Daniel','Cr@!ag','Male','9876543210',001);

INSERT INTO Employees(Employee_ID,Husky_ID,First_Name,Last_Name,Gender,Phone_Number,Employee_Type_ID)
VALUES
(2222,'002774233','Daniel','','Male','9876543210',001);
```

```sql
--Creating a function that checks if the husky ID is valid (9 digit number starting with 2 0's)--
GO
CREATE FUNCTION CountInvalidHuskyIDs()
RETURNS smallint
AS
BEGIN
   DECLARE @Count smallint=0;
   SELECT @Count = COUNT(Husky_ID)
          FROM Employees
          WHERE LEN(Husky_ID) != 9
          OR Husky_ID NOT LIKE '%00[0-9][0-9][0-9][0-9][0-9][0-9][0-9]%'
   RETURN @Count;
END;

GO

--Testing by inserting invalid values--
ALTER TABLE Employees ADD CONSTRAINT CheckValidHusky_ID CHECK (dbo.CountInvalidHuskyIDs()=0);

INSERT INTO Employees(Employee_ID,Husky_ID,First_Name,Last_Name,Gender,Phone_Number,Employee_Type_ID)
VALUES
(2228,'qwertyujik','Daniel','Craig','Male','9876543210',001);

select * from Employees;

INSERT INTO Employees(Employee_ID,Husky_ID,First_Name,Last_Name,Gender,Phone_Number,Employee_Type_ID)
VALUES
(2221,'!@#456789','Daniel','Craig','Male','9876543210',001);
```

Computed Column

```sql
-- Creating a function to calculate total estimated price of a prescription, which includes prices of vaccines, medicines and tests--
--attached with each prescirption--
CREATE FUNCTION dbo.GetPrescriptionTotalPrice
(
    @PrescriptionId INT
)
RETURNS TABLE
AS
RETURN
(
    SELECT
        SUM(COALESCE(V.Price, 0)) + SUM(COALESCE(T.Price, 0)) + SUM(COALESCE(M.Price, 0)) AS TotalPrice
    FROM
        Prescription_Vaccine PV
    FULL OUTER JOIN
        Prescription_Tests PT ON PV.Prescription_ID = PT.Prescription_ID
    FULL OUTER JOIN
        Prescription_Medicine PM ON PV.Prescription_ID = PM.Prescription_ID
    LEFT JOIN
        Vaccine V ON PV.Vaccine_ID = V.Vaccine_ID
    LEFT JOIN
        Test T ON PT.Test_ID = T.Test_ID
    LEFT JOIN
        Medicine M ON PM.Medicine_ID = M.Medicine_ID
    WHERE
        PV.Prescription_ID = @PrescriptionId
);
GO
```

```sql
--Creating a new column called EstimatedTotalPrice, and populating the value by passing the prescription ID to the function.
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'Prescription' AND COLUMN_NAME = 'EstimatedTotalPrice'
BEGIN
    ALTER TABLE Prescription
    ADD EstimatedTotalPrice DECIMAL(10, 2);
END
GO
UPDATE Prescription
SET EstimatedTotalPrice = (SELECT TotalPrice FROM dbo.GetPrescriptionTotalPrice(Prescription_ID));
GO
--Displaying the updated columns with the populated values.
SELECT * FROM Prescription;
```

Results 1

SELECT * FROM Prescription; Enter a SQL expression to filter results (use Ctrl+Space)

	123 Prescription_ID	ABC Feedback	123 EstimatedTotalPrice
1	1,110	Breathing problem because of pollution. Prescribed inhaler	3,649.98
2	1,111	Prescription verified and processed. Medication delivery scheduled for tomorrow.	506
3	1,121	Patient needs to take proper test for correct diagnosis.	389
4	1,131	101 Degree Celsius Fever. Need to take proper rest and visit again after a week	371
5	1,141	Prescription filled and medication given to patient.	380.23
6	1,151	Patient needs to rest and complete medication cycle	5,879.96
7	1,161	X-ray scheduled for patient.	2,549.98
8	1,171	Patient is all well. Needs to take rest and visit after a week.	1,115.49
9	1,181	High fever and throat infection. Prescribed medication	3,141.96
10	1,191	Migraine Problem. Patient needs to sleep and take rest	2,612.49

Panels

Views

Students who are vaccinated with Moderna Vaccine

First_name	Last_name	Date_of_Birth	Phone_Number
Mansi	Dabriwal	1998-04-07	8573708868
Atharva	Kulkarni	1997-01-05	9876543210
Shrey	Sinha	1997-12-12	7876543210
Animesh	Jain	1994-05-01	8907654321

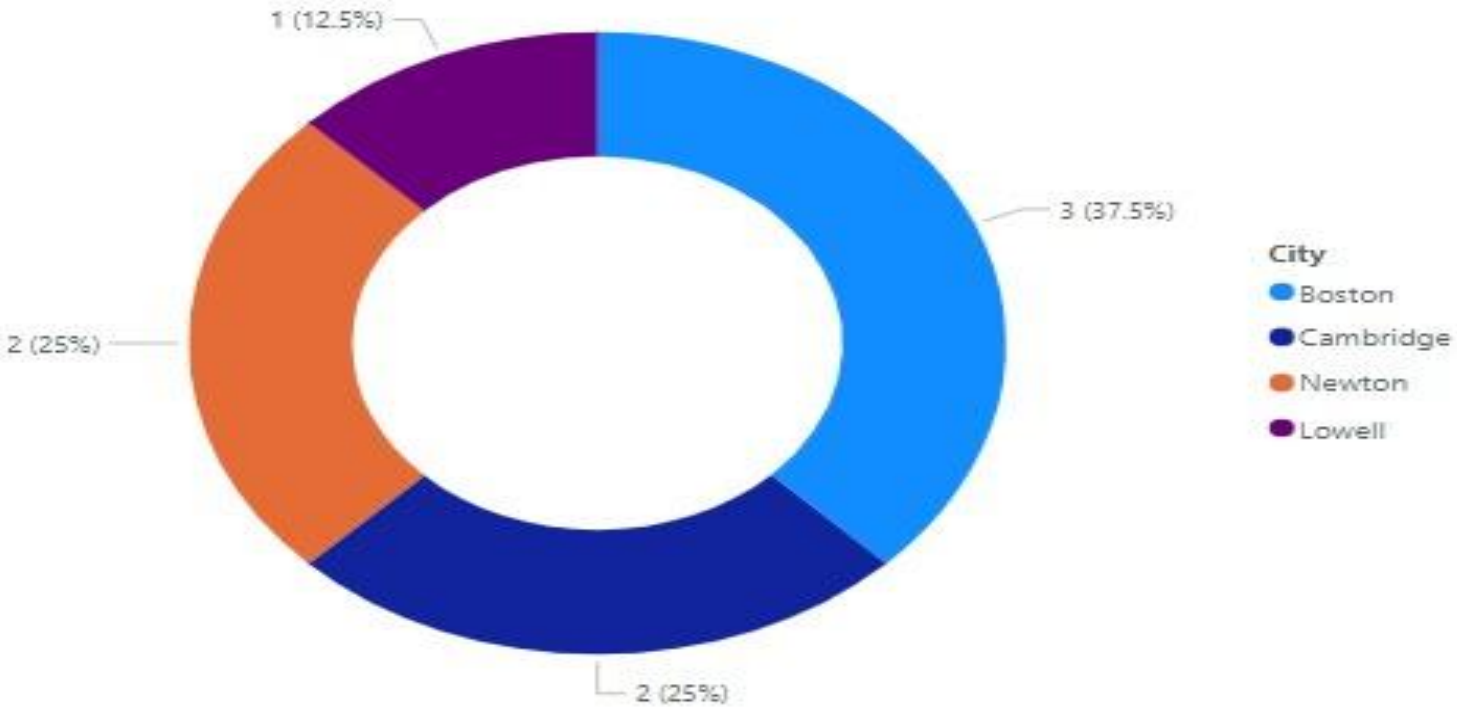
All vaccine and test centers, and pharmacies that are situated in Massachusetts

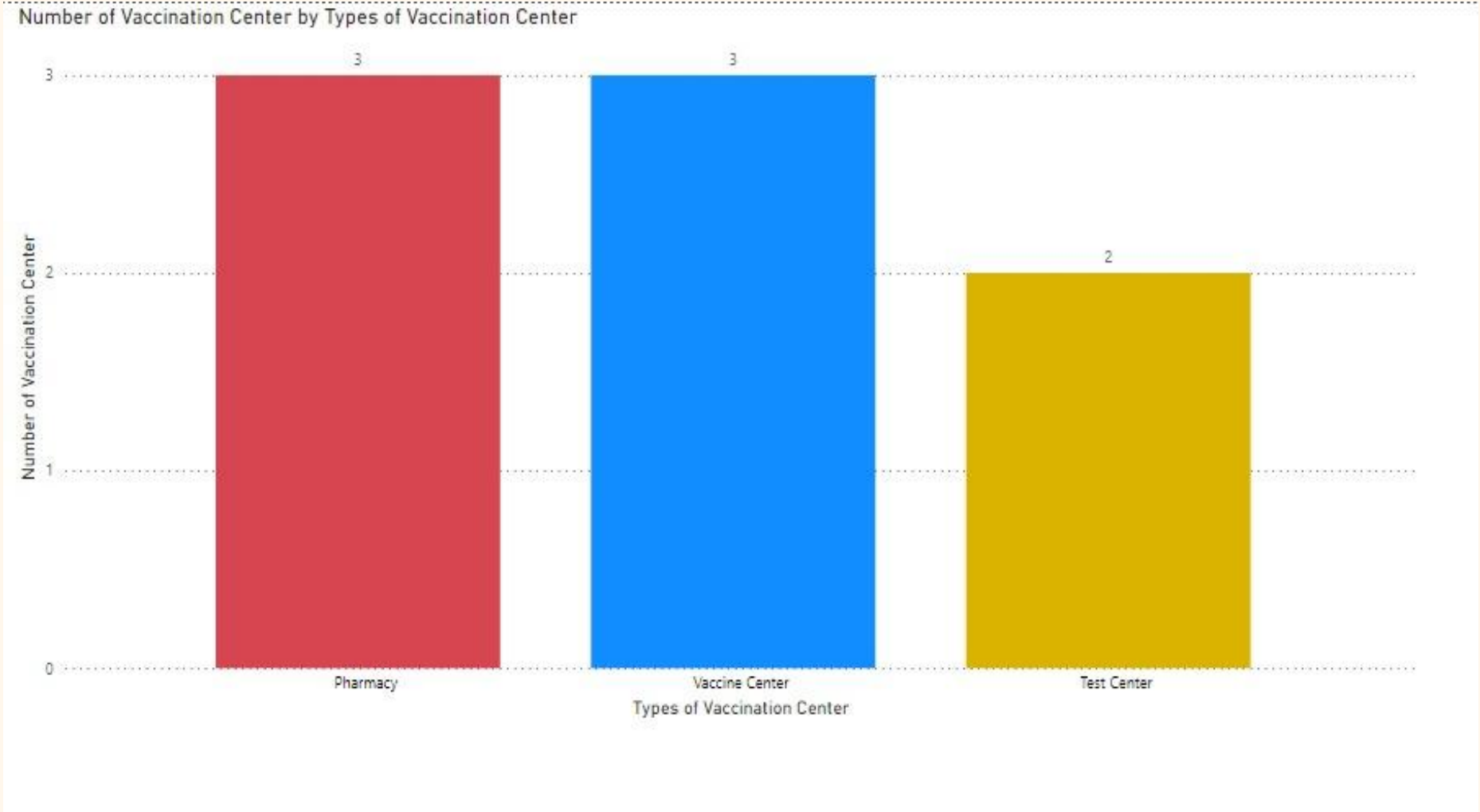
Vaccine_Center_ID	Center_Name	AddressLine_1	AddressLine_2	City	ZipCode	Center_Type
23	Rhode Vaccination Center	999 Cedar Rd	Unit B	Boston	2108	Vaccine Center
26	Boston Urgent Care Center	555 Pine St	Suite 245	Cambridge	27854	Vaccine Center
28	Rhode Pediatric Clinic	999 Cedar Rd	Unit B	Boston	2108	Vaccine Center
54	Heart Health Services	777 Walnut St	Apt 5A	Newton	2456	Test Center
55	Top Clinical Labs	999 Cedar Rd	Unit B	Boston	2108	Test Center
203	Cigna	321 Maple Blvd	Suite 100	Lowell	2953	Pharmacy
204	UnitedHealth Group	555 Pine St	Suite 245	Cambridge	27854	Pharmacy
206	Humana	777 Walnut St	Apt 5A	Newton	2456	Pharmacy

05

Reports & Data visualization

Number of Centers by City




Number of Vaccination Center by Types of Vaccination Center
3
3
2
Number of Vaccination Center
3
2
1
0
Pharmacy
Vaccine Center
Test Center
Types of Vaccination Center

Number of Patients Who Have taken Moderna Vaccine

06

FutureScope

- The healthcare system's database could be used for research and analysis purposes to identify patterns and trends in healthcare data, which could help healthcare professionals to improve their services and develop better treatments for patients.
- The database could be enhanced with predictive analytics capabilities to predict potential health risks and recommend preventive measures based on the user's medical history and current health status.

Thanks