Final Project Report:

Pharmacy Database Overview:

This is a Pharmaceutical database that allows pharmacists to view and manage medication. A pharmacist can view the medication a patient needs, the amount of prescription a pharmacist would give to a patient. Consequently, this database allows patient to view their prescription medications, see when they bought it, from whom they bought it from, the amount they need, and the cost of their medication.

MySQL Section:

In this section I will show the code implemented when creating the pharmaceutical database.

To establish the database the following code was executed:

- Create database pharmacy_portal_db; - this creates the database
- Use pharmacy_portal_db; - this uses the database for further use

The Users table stores information about the user:

- **userId**- this stores the user's id number
- **userName**- this stores the first name and last name of the user
- **contactInfo**-stores the user's phone number or email
- **userType**- this notifies whether the user is a pharmacist or a patient

Below is the code for the Users table:

```
Create table Users (
userId int not null unique auto_increment,
userName varchar(45) not null unique,
contactInfo varchar(200),
userType enum('pharmacists', 'patient') not null,
Primary key (userId)
);
```

The Medication table stores information about medication:

- **medicationId**- this stores the id number of medication
- **medicationName**- this stores the name of the medication (i.e. adderall or robitussin)
- **dosage**-stores dosage information (i.e. how much should be described to the patient)
- **manufacturer**- this shows who made the medication

Below is the code for the Medication table:

```
Create table Medication (
medicationId int not null unique auto_increment,
medicationName varchar(45) not null,
dosage varchar(45) not null,
maufacturer varchar(100),
Primary key (medicationId)

);
```

The Prescriptions table stores information about prescriptions:

- **prescriptionId**- this stores the id of the prescription
- **userId**- a foreign key that stores the user's id number
- **medicationId**- another foreign key that stores the id number of medication
- **dosageInstructions**-stores instructions on how and when to take the prescribed medication
- **quantity**- this shows the number of pills, or injections available to the patient
- **refillCount**- stores information about how many times a patient can have refills

Below is the code for the Prescriptions table:

```
Create table Prescriptions(
prescriptionId int not null unique auto_increment,
userId int not null,
medicationId int not null,
prescribedDate Datetime not null,
dosageInstructions varchar(200),
quantity int not null,
refillCount int default 0,
primary key (prescriptionId),
Foreign Key (userId) references Users(userId),
foreign key (medicationId) references Medication(medicationId)
);
```

The Inventory table stores information about all inventory:

- **inventoryId**- this stores the id number of medication
- **medicationId**-a foreign key that stores the id number of medication
- **quantityAvailable**-stores the quantity available for each medication
- **lastUpdated**- gives information for when the inventory was last updated

Below is the code for the Inventory table:

```
create table Inventory (
inventoryId int not null unique auto_increment,
medicationId int not null,
quantityAvailable int not null,
lastUpdated datetime not null,
Primary key (inventoryId),
foreign key (medicationId) references Medication(medicationId)

);
```

The Sales table stores information about all the sales:

- **salesId**- this stores the sales id
- **prescriptionId**- a foreign key that stores the id of the prescription
- **saleDate**- stores information for when the prescribed medication was sold
- **quantitySold**- stores information about the how much was sold
- **saleAmount**- stores information about the amount sold

Below is the code for the Inventory table:

```
Create table Sales (
saleId int not null unique auto_increment,
prescriptionId int not null,
saleDate datetime not null,
quantitySold int not null,
saleAmount decimal(10, 2) not null,
primary key (saleId),
foreign key (prescriptionId) references Prescriptions(prescriptionId)
);
```

PHP Section:

In this section I have used php and html to implement a UI so patients can view their prescriptions, and for pharmacists to view or update patients into the database. A patient can login using their username and password to view their prescriptions or create a new account. For pharmacists, they can login in with their credentials in the pharmacy portal and update existing patients or add new patients as well as view their prescriptions, the pharmacy's inventory and the sales.