

**School of Computer Science and Engineering**

DBMS Course Project Report

On

**PHARMACY DATABASE MANAGEMENT SYSTEM**

Submitted by

**Mr. Abhishek P.M 01FE18BCS010**

**Mr. Abhishek S 01FE18BCS012**

**Mr. Aditya Mishra 01FE18BCS018**

**Mr. Aman Khan Athani 01FE18BCS028**

**TEAM LEADER: Abhishek P.M**

KLETech/SoCSE(2019-20)/DBMS/15ECSC208/Course Project/Report

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| S.no | Content |
| 1. | Introduction |
| 2. | Proposal Phase |
| 3. | Design phase |
| 4. | Implementation phase |
| 5. | User Manual/README |
| 6. | Snap shots of Client and Office and Workplace  (Venkateshwara Medical and General Stores) |
| 7. | Acceptance letter |
| 8. | Final acceptance letter |
| 9. | Customer Feedback w.r.t the application |

**1.INTRODUCTION**

**Problem Description:**

The Problem relates to our client Mr. Mallikarjun Savanur who owns Pharmacy store called ‘Venkateshwara Medical and General Store ’ of a health care centre located in Chennamma Circle, Hubli that is still abiding the rules of the book and is going with the age old process of bill books and accounts maintained in ledger books. With most of the bills not being properly recorded, and the stock analysis not being properly done, data related to medicines that are expired , data related to the distributor from whom the medicine is purchased, neither there exists a record on the amount of sales done by a particular doctor, the medical history of the patient doesn’t exist either. There exists a problem that requires a database that helps in generating bills, keeping a record and helps us in analyzing the patient’s medical history, gives the total transactions between distributers, and also helps us retrieve any other necessary data

According to the client requirement we have planned to build an application which has user friendly interface and allows the client to store and retrieve any data in a very sequential and as easily as possible, there are a few factors that we have taken into consideration to build our final application, that we have tried to explain in a crisp and easy manner, we have also tried to automate the whole process so that the process of maintenance and administration becomes very easy for the client and is not strenuous.

All the course knowledge that we have learnt was put into application and everything was applied to a way that the process is simplified, the ER diagram and relational mapping was formulated and queries related to the above sql code were executed on oracle Sql database and the application was implemented along a python console on Django on Visual Studio Code IDE

**2. PROPOSAL PHASE**

**Proposal Phase Report**

**Responsibilities:**

Abhishek PM made the first contact with the client and was responsible for the collaborative planning of our project; he came up with an ER diagram and is in constant contact with the client for gathering all the required data/info that will be required for the project.

Abhishek S made an ER diagram and was involved in the collaborative planning of our project. He is tasked with organizing the raw data so as to make it compatible with our optimal ER diagram.

Aditya Mishra also made an ER diagram and made the schema for our project. He is also tasked with organizing the data so as to make it compatible with our optimal ER diagram.

Aman Khan was also responsible for an ER diagram and planning of our project, contacting the client about his needs and wants. He was also tasked with the final editing and formatting of this document

**Problem Description:**

The Problem relates to the Pharmacy store of a health care centre that is still abiding the rules of the book and is going with the age old process of bill books and accounts maintained in ledger books. With most of the bills not being properly recorded, and the stock analysis not being properly done, data related to medicines that are expired , data related to the distributor from whom the medicine is purchased, neither there exists a record on the amount of sales done by a particular doctor, the medical history of the patient doesn’t exist either. There exists a problem that requires a database that helps in generating bills, keeping a record and helps us in analyzing the patient’s medical history, gives the total transactions between distributers, and also helps us retrieve any other necessary data

**Requirements:**

**Feasibility Study**:

A feasibility analysis involves a detailed assessment of the need, value and practicality of a proposed enterprise, the process of designing and implementing record and has significant accountability and resource implications for the organization

**Operational Feasibility:**

People are inherently resistant to change, and computers have been known to facilitate change, an estimate should be made. It has to be user friendly and make way for smooth and a more centralized application that helps make the process more reliable.

The pharmacy keeps patient records, including information like phone no, referrals Bill no, medicines bought, etc There is also a record of medicines which is connected to a larger record of stocks, Which keeps track of all the medicines in the store, their suppliers, their quantities etc. There is a separate record of all the payments done by the store to the suppliers which holds other details like date, and amount and mode of payment. With the use of our database application our client will be benefitted the following ways:

* The client will be able to provide automatic-printed bills and not written ones which take time and effort
* The client will have a proper database for his sales and purchases of all of his drugs making it easier for him to look up information about anything with ease
* The client will be able to check his stock and will be notified about the drugs which need to be restocked
* The client will be able to calculate his gross profit, maintain a record on the sales of the following month and analyze the data in the most suitable manner
* The expiry stock, sales order, purchase order and sales summary are obtained.

**Design Questions to be answered**

**Question 1:** From the problem description, identify the entities that need to be represented in the database, the attributes of each entity, the relationships between the entities, and the cardinality ratios of each relationship.

Entity 1: Medicine

* M\_id
* MRP
* Stock

Entity 2: Patient

* P\_id
* Phone No
* Name

F\_name

L\_name

Entity 3: Bill Record

* Bill\_no
* Date
* Amount
* Medicines

Entity 4: Stock

* Supplier
* Stock\_no
* Mfd\_date
* Exp\_date

Entity 5: Supplier

* Company
* Phone\_no
* Address
* Batch\_no

Realtionships ,participation and cardinality:

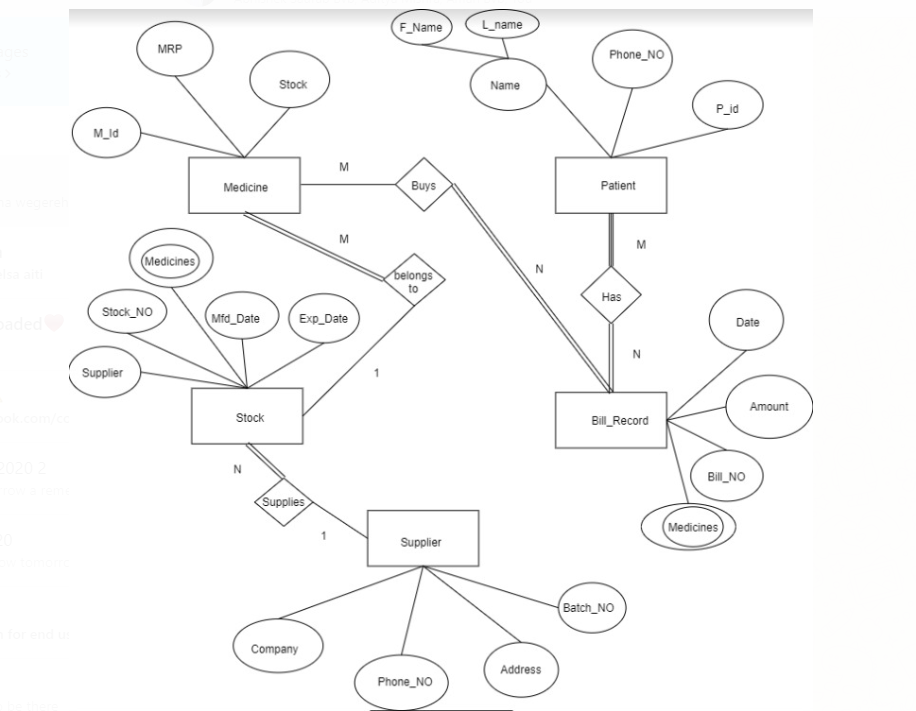
Patient Buys Medicine : m:n cardinality Ratio

Medicine Belongs to Stock: m:1 cardinality Ratio

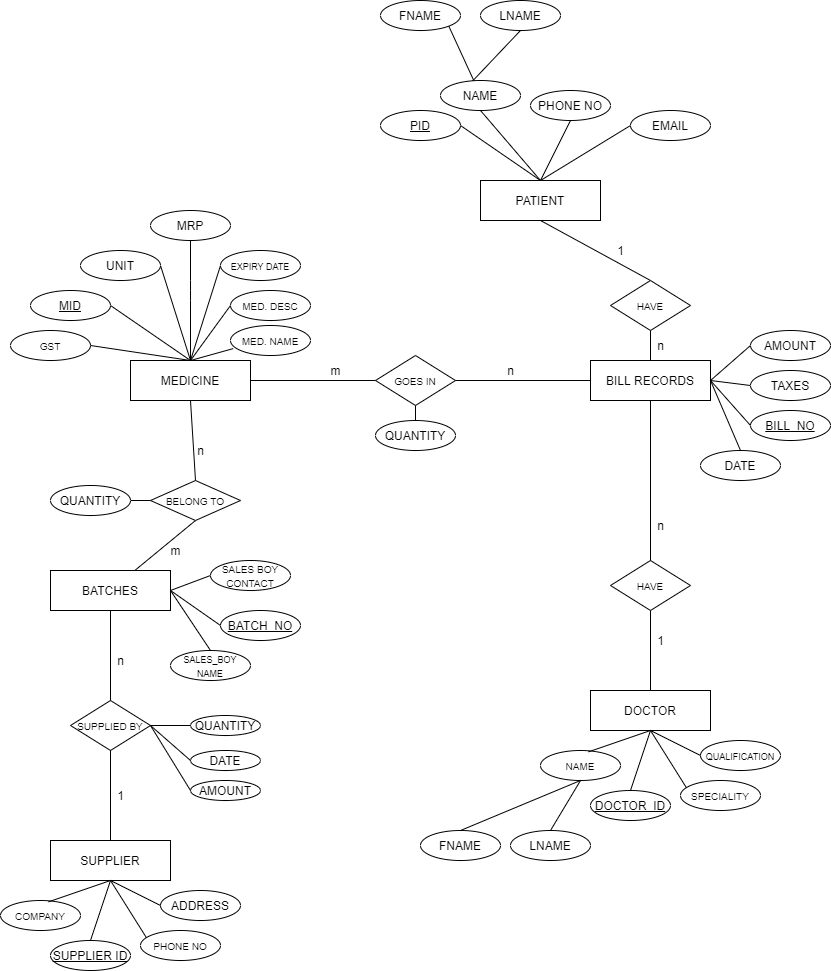
Stock Supplies Supplier: n:1 cardinality Ratio

Patient has Bill Record: 1:n Cardinality Ratio

**Question 2:** Draw an Entity-Relationship Diagram illustrating the information you have identified in Question 1.

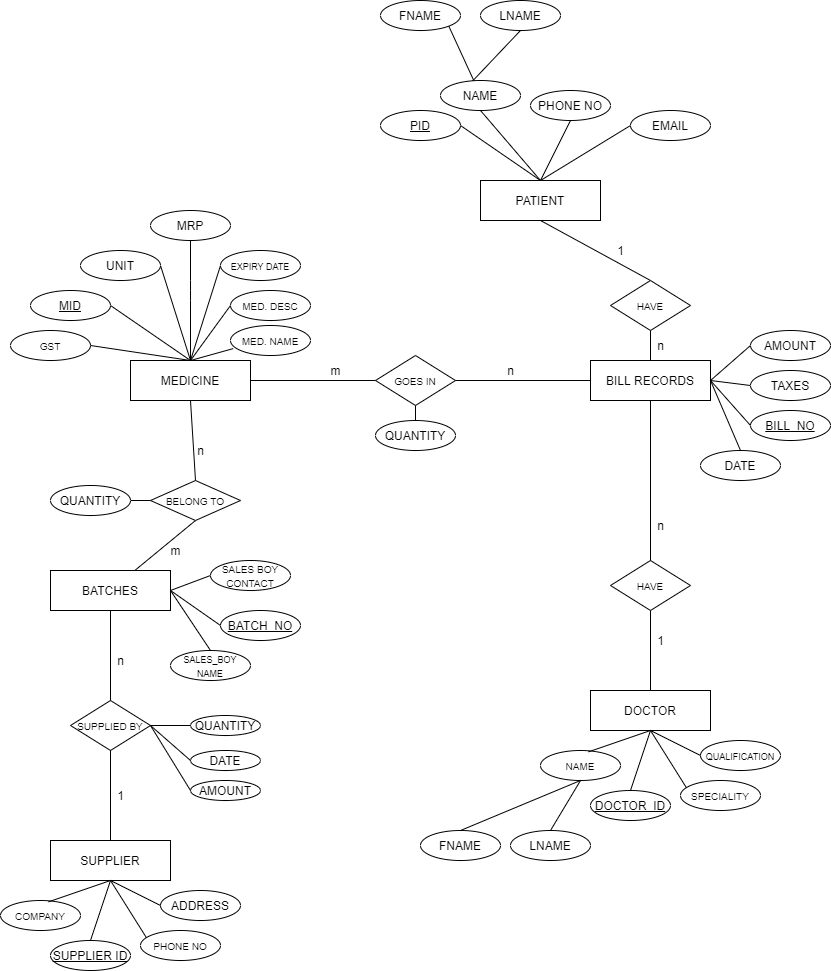


**Question 3:** Draw **alternate** Entity-Relationship Diagram illustrating the information you have identified in Question 1 that you think are most likely to occur.

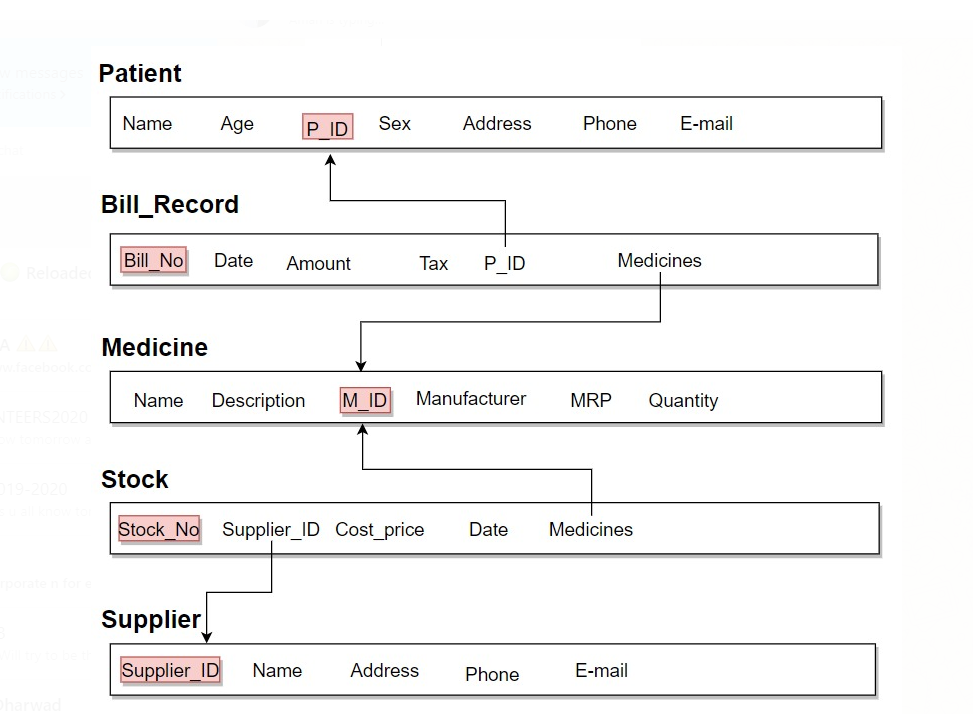
****

**Question 4:** Choose the **optimal** Entity-Relationship Diagram from the designs provided above and justify why you think this is an optimal solution for your identified problem specification.

**This Is our optimal ER-Diagram**

****

**Question 5:** Draw an ER to Relation Mapping illustrating the information you have identified in Question 4.



**Question 6:** Draw a Data Dictionary illustrating the information you have identified in Question 6.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Object (Entity)** | **Name (Attribute)** | **Type**  **(Data type)** | **Description** | **Primary Key** | **Foreign Key** |
| Patient | P\_id | Integer | Unique number that identifies the patient | Yes | No |
|  | Phone\_no | Integer | Contact details of the patient | No | Yes |
|  | Name | string | Name details of the patient | No | Yes |
| Bill Record | Bill\_no | Integer | Uniquely identifies a particular bill | Yes | No |
|  | Date | Integer | Date of Purchase | No | Yes |
|  | Amount | Integer | Amount debited by a patient | No | Yes |
|  | Medicines | Integer | Medicine Id for a particular Medicine | No | Yes |
| Supplier | Supplier\_id | Integer | Uniquely identifies the Supplier | Yes | No |
|  | Phone No | Integer | Contact details of the supplier | No | No |
|  | Company | String | Medicine Id for a particular | No | Yes |
|  | Address | String | Details of the supplier | No | Yes |
|  | Batch No | Integer | Batch distribution of supplier | No | Yes |
| Stock | Stock\_no | integer | Uniquely identifies a particular stock | Yes | No |
|  | Supplier\_id | Integer | To recognize a particular Supplier | No | Yes |
|  | Cost\_price | Integer | Cost of a particular medicine | No | Yes |
|  | Date | Integer | The in date of the stock | No | Yes |
|  | Medicine\_id | Integer | Id to identify the medicine | No | Yes |
| Medicines | Medicine\_id | Integer | Uniquely Identifies a medicine | Yes | No |
|  | Name | Integer | Identifies the Medicine | No | Yes |
|  | Description | string | Defines a drescription about a string | No | Yes |
|  | Manifacturer | String | Specifies the manifacturar | No | Yes |
|  | Mrp | Integer | The cost price of the medicine | No | Yes |
|  | Quantity | Integer | The Quantity of medicine purchased | No | Yes |

**References:**

**-Tools used:**

**1; draw.io**

**2: erdplus.com**

**-Books:**

**1:Fundamentals of database systems**

**2:Class notes**

**Submission Date: 13th February 2020**

**3.DESIGN PHASE**

**Design Phase Report**

**Responsibilities:**

Abhishek P.M: Rectifying issues In ER diagram, Worked on Normalization of the data, Creating User interfaces, Documentation and Creating Database

Abhishek S: Rectifying issues In ER diagram, creating a new relation mapping and implementing GUI.

Aditya Mishra: Rectifying issues In ER diagram, creating a new relation mapping, and creating Database and Normalization.

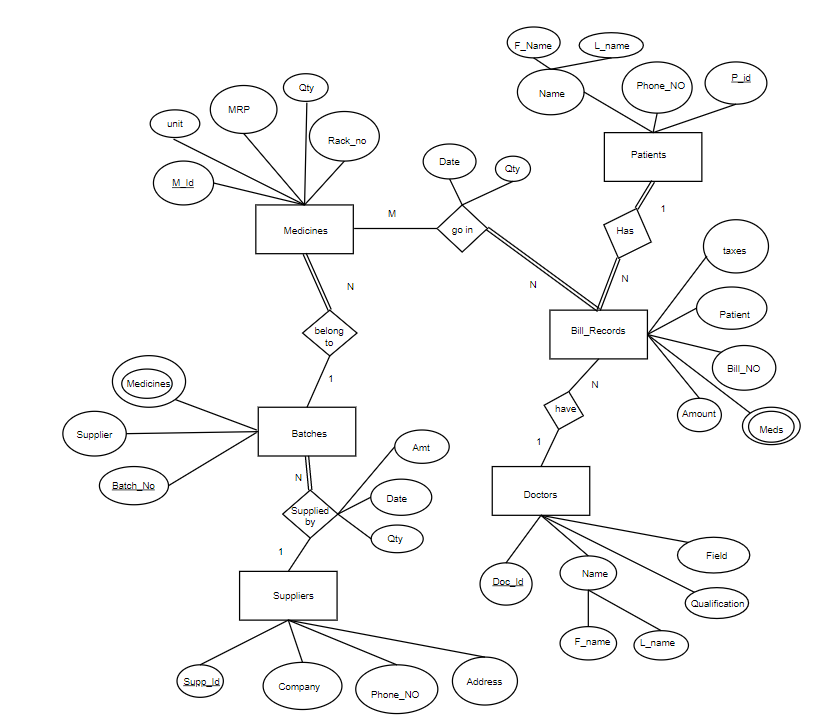
Aman Khan Athani: Rectifying issues In ER diagram and creating GUI.

**ER Design:**

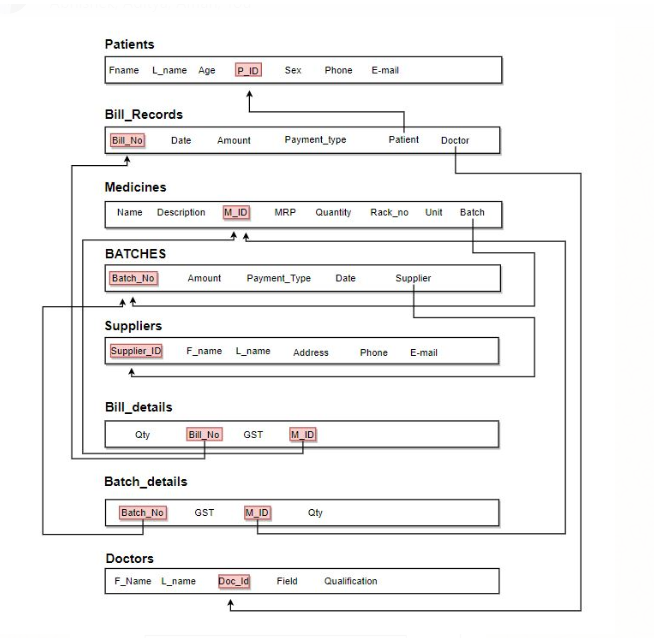
**CHANGES IN THE ER- DIAGRAM**

1. A few entities were missing in the first review, taking into consideration the various other relations that could be included, the new ER diagram has been formulated with new entities and attributes for all the old entities and the new ones are reconsidered and added.
2. The entity names were in a clash with the attributes of the entities. All the entities are renamed with proper standard taken into consideration.
3. A few attributes were in anomaly, and did not fit in to the entity, so such anomalous attributes were removed and an optimal ER diagram with the most suitable attributes are taken into consideration in this ER-Diagram.
4. The relationships between the entities were anomaly, so everything reconsidered; the relations between the entities have been changed.

Hence All the Above mentioned changes have been implemented in the following ER diagram and the relation mapping has been done



**ER to Relation Mapping:**



**DataDictionary:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Object (Entity)** | **Name (Attribute)** | **Type**  **(Data type)** | **Description** | **Primary Key** | **Foreign Key** |
| Patients | Fname | String | First name of the patient | Yes | No |
| Patients | Lname | String | Last name of the patient | No | No |
| Patients | Age | Int | Age of the patient | No | No |
| Patients | P\_id | String | Unique ID to determine the patient | No | No |
| Patients | Sex | String | Sex of the patient | No | No |
| Patients | Phone | Int | Contact number of the patient | No | No |
| Patients | Email | String | Email id of the patient | No | No |
| Bill\_Records | Bill\_no | Int | Unique Bill\_no of each Bill\_record | Yes | No |
| Bill\_Records | Date | Int | Date of the Purchase | No | No |
| Bill\_Records | Amount | Int | Total purchase cost | No | No |
| Bill\_Records | Payment\_type | String | Mode of payment | No | No |
| Bill\_Records | P\_id | Int | Unique ID to determine the patient | No | Yes |
| Bill\_Records | Doc\_id | Int | Unique ID to determine each doctor | No | Yes |
| Medicines | Name | String | Name of the Medicine | No | No |
| Medicines | Description | String | Description of the Medicine | No | No |
| Medicines | M\_Id | String | Unique id to determine a Medicine | Yes | No |
| Medicines | MRP | Int | The cost of the Medicine | No | No |
| Medicines | Quantity | Int | The Quantity of medicines available | No | No |
| Medicines | Rack\_No | Int | The Rack where the medicine is kept | No | No |
| Medicines | Unit | Int | Total available Units of the Medicine | No | No |
| Medicines | Batch | int | The batch in which the medicine arrived | No | Yes |
| Batches | Batch\_No | Int | The Batch in which the medicine comes in | Yes | No |
| Batches | Amount | Int | The amount of purchase by the pharmacy | No | No |
| Batches | Payment\_type | String | The payment mode by the pharmacy | No | No |
| Batches | Date | Inr | The date of purchase by the pharmacy | No | No |
| Batches | Supplier | String | The id of the supplier | No | Yes |
| Bill\_Details | Bill\_No | Int | The Bill no of the Particular Bill | No | Yes |
| Bill\_Details | Qty | Int | The Quantity of medicines purchased | No | No |
| Bill\_Details | Gst | Int | The Goods and Service tax applied | No | No |
| Bill\_Details | M\_Id | Int | The medicine id of the medicine purchased | No | Yes |
| Suppliers | Supplier\_Id | Int | Uniquely determines the supplier | Yes | No |
| Suppliers | F\_name | String | First name of the incharge in supplies | No | No |
| Suppliers | L\_name | String | Last name of the incharge in supplies | No | No |
| Suppliers | Address | String | The address of the Supplier | No | No |
| Suppliers | Phone | Int | Contact number of the Supplier | No | No |
| Suppliers | Email | String | Email address of the Supplier | No | No |
| Batch\_details | Batch\_No | Int | The Batch Number of the medicine intake | No | Yes |
| Batch\_details | Gst | Int | The goods and service tax applied on the supplier | No | No |
| Batch\_details | M\_ID | Int | The medicine id of each medicine | No | Yes |
| Batch\_details | Qty | Int | The Quantity that the suppliers supplies | No | No |
| Doctors | F\_Name | String | First name of the doctor | No | No |
| Doctors | L\_Name | String | Last Name of the Doctor | No | No |
| Doctors | Doc\_id | Int | TheUnique Id of the Doctor | Yes | No |
| Doctors | Field | String | The field of pursuit of Doctor | No | No |
| Doctors | Qualification | String | The Qualification of the Doctor | No | No |

**Question 1: Normalization:**



In the above Entity, P\_Id is the primary key and hence, it determines all other attributes present in the tuple.

* As all the attributes of the entity are individual or atomic values the entity is already said to be in 1NF
* The above entity is in 2NF, because there are no multiple key attributes, so there is no need to check whether any of the non key attributes depend partially on the key attributes of the entity
* The above entity is in 3NF, because there exists only one FD such that it is a primary key and determines all other attributes and hence it is already in 3NF
* In the above entity, we already have the relation A->B, where A is the primary key satisfied and hence the relation is already in BCNF.

Batches

In the above Entity, Batch\_no is the primary key and hence, it determines all other attributes present in the tuple.

* As all the attributes of the entity are individual or atomic values the entity is already said to be in 1NF
* The above entity is in 2NF, because there are no multiple key attributes, so there is no need to check whether any of the non key attributes depend partially on the key attributes of the entity
* The above entity is in 3NF, because there exists only one FD such that it is a primary key and determines all other attributes and hence it is already in 3NF
* In the above entity, we already have the relation A->B, where A is the primary key satisfied and hence the relation is already in BCNF.

**Bill Records**

Bill records

In the above Entity, Bill\_no is the primary key and hence, it determines all other attributes present in the tuple.

* As all the attributes of the entity are individual or atomic values the entity is already said to be in 1NF
* The above entity is in 2NF, because there are no multiple key attributes, so there is no need to check whether any of the non key attributes depend partially on the key attributes of the entity
* The above entity is in 3NF, because there exists only one FD such that it is a primary key and determines all other attributes and hence it is already in 3NF
* In the above entity, we already have the relation A->B, where A is the primary key satisfied and hence the relation is already in BCNF.

doctors

In the above Entity, Doc\_id is the primary key and hence, it determines all other attributes present in the tuple.

* As all the attributes of the entity are individual or atomic values the entity is already said to be in 1NF
* The above entity is in 2NF, because there are no multiple key attributes, so there is no need to check whether any of the non key attributes depend partially on the key attributes of the entity
* The above entity is in 3NF, because there exists only one FD such that it is a primary key and determines all other attributes and hence it is already in 3NF
* In the above entity, we already have the relation A->B, where A is the primary key satisfied and hence the relation is already in BCNF.

medicines

In the above Entity, M\_Id is the primary key and hence, it determines all other attributes present in the tuple.

* As all the attributes of the entity are individual or atomic values the entity is already said to be in 1NF
* The above entity is in 2NF, because there are no multiple key attributes, so there is no need to check whether any of the non key attributes depend partially on the key attributes of the entity
* The above entity is in 3NF, because there exists only one FD such that it is a primary key and determines all other attributes and hence it is already in 3NF
* In the above entity, we already have the relation A->B, where A is the primary key satisfied and hence the relation is already in BCNF.

suppliers

In the above Entity, Supplier\_id is the primary key and hence, it determines all other attributes present in the tuple.

* As all the attributes of the entity are individual or atomic values the entity is already said to be in 1NF
* The above entity is in 2NF, because there are no multiple key attributes, so there is no need to check whether any of the non key attributes depend partially on the key attributes of the entity
* The above entity is in 3NF, because there exists only one FD such that it is a primary key and determines all other attributes and hence it is already in 3NF
* In the above entity, we already have the relation A->B, where A is the primary key satisfied and hence the relation is already in BCNF.

bill details

In the above Entity, Bill\_no and M\_ID the primary key and hence, it determines all other attributes present in the tuple.

* As all the attributes of the entity are individual or atomic values the entity is already said to be in 1NF
* The above entity should be bought done to 2NF, the functional dependencies in the above relation are as follows

{Bill\_no, M\_id} -> {Qty, GST}

{Bill\_ No} -> {Qty}

{M\_ID} -> {GST}

As there are multiple key attributes we have to check whether any of the non key attributes depend partially on the key attributes of the entity.

So to bring it in 2NF, the schema would be normalized as

|  |  |
| --- | --- |
| **Bill\_No** | Qty |

|  |  |
| --- | --- |
| **M\_ID** | GST |

Here to normalize to 2NF, the keys have been made individual, and separate tables have been formed and hence, the relations are now in 2NF.

* As there exists no functional dependencies such that transitivity exists, the FD’s are also in 3NF.
* In the above entity, after the normalization to 2NF we already have the relation A->B, where A is the primary key satisfied and hence the relation is already in BCNF.

batch details

In the above Entity, Batch\_no and M\_ID the primary key and hence, it determines all other attributes present in the tuple.

* As all the attributes of the entity are individual or atomic values the entity is already said to be in 1NF
* The above entity should be bought done to 2NF, the functional dependencies in the above relation are as follows

{Batch\_no, M\_id} -> {Qty, GST}

{Batch\_ No} -> {Qty}

{M\_ID} -> {GST}

As there are multiple key attributes we have to check whether any of the non key attributes depend partially on the key attributes of the entity.

So to bring it in 2NF, the schema would be normalized as

|  |  |
| --- | --- |
| **Batch\_No** | Qty |

|  |  |
| --- | --- |
| **M\_ID** | GST |

Here to normalize to 2NF, the keys have been made individual, and separate tables have been formed and hence, the relations are now in 2NF.

* As there exists no functional dependencies such that transitivity exists, the FD’s are also in 3NF.
* In the above entity, after the normalization to 2NF we already have the relation A->B, where A is the primary key satisfied and hence the relation is already in BCNF.

**Question 2:** Choose the **optimal** normalized schema from Question 1 and justify why you think this is an optimal solution.

The Optimal Solution for the normalized schema would be:

Most of the entities such as patients, Bill\_records, Medicines, Batches, Suppliers and Doctors have only one primary key, and hence as all the attributes are already atomic values, they are in 1NF. As the entries are now optimal and there is only one functional dependency, and no multiple key, the entities are also in further normalized forms.

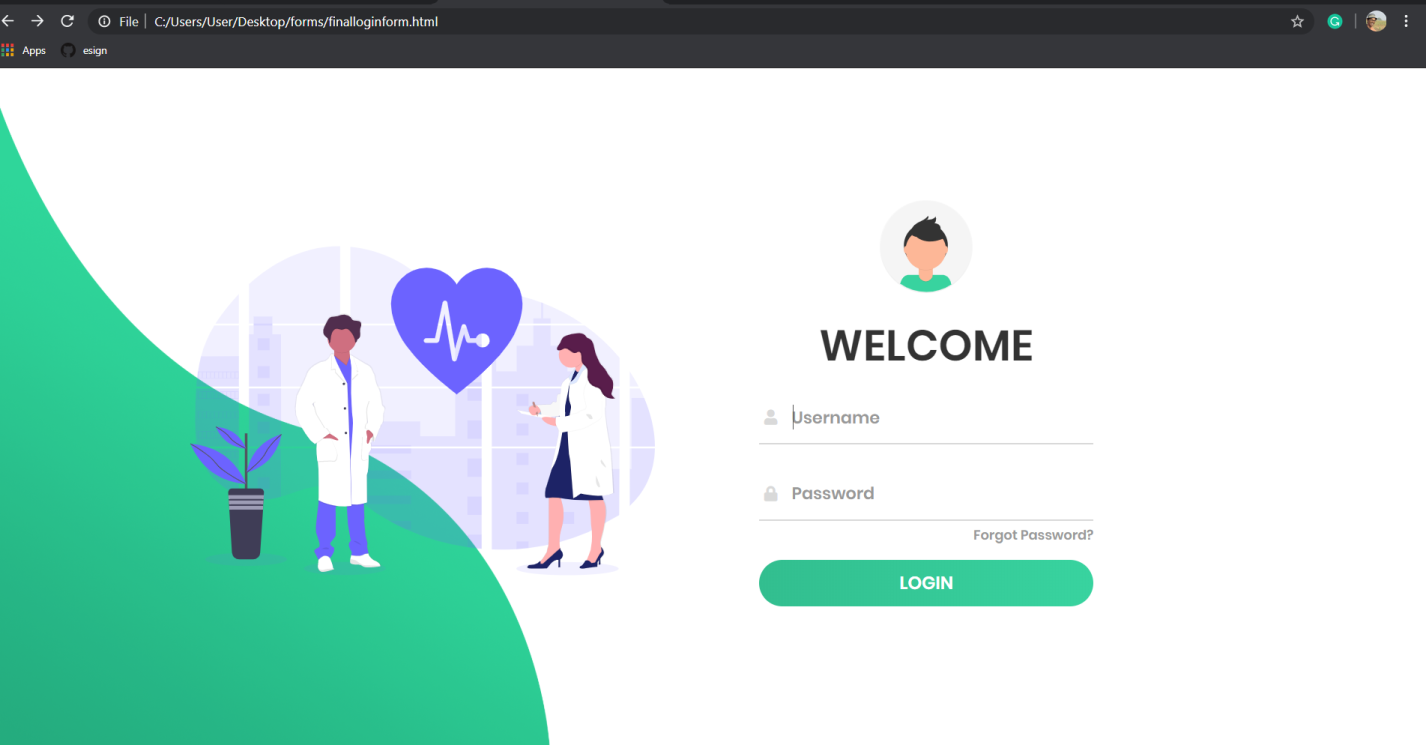
The other two entities are bill details and batch details, have all their attributes which are atomic values and hence it is in 1NF, Now as the above entities have two primary keys, To check if they are in 2NF, we need to check if the partial keys determine the attributes and hence as we have multiple functional dependencies, they do, so altering the table has to be done and hence, We obtain 2NF.

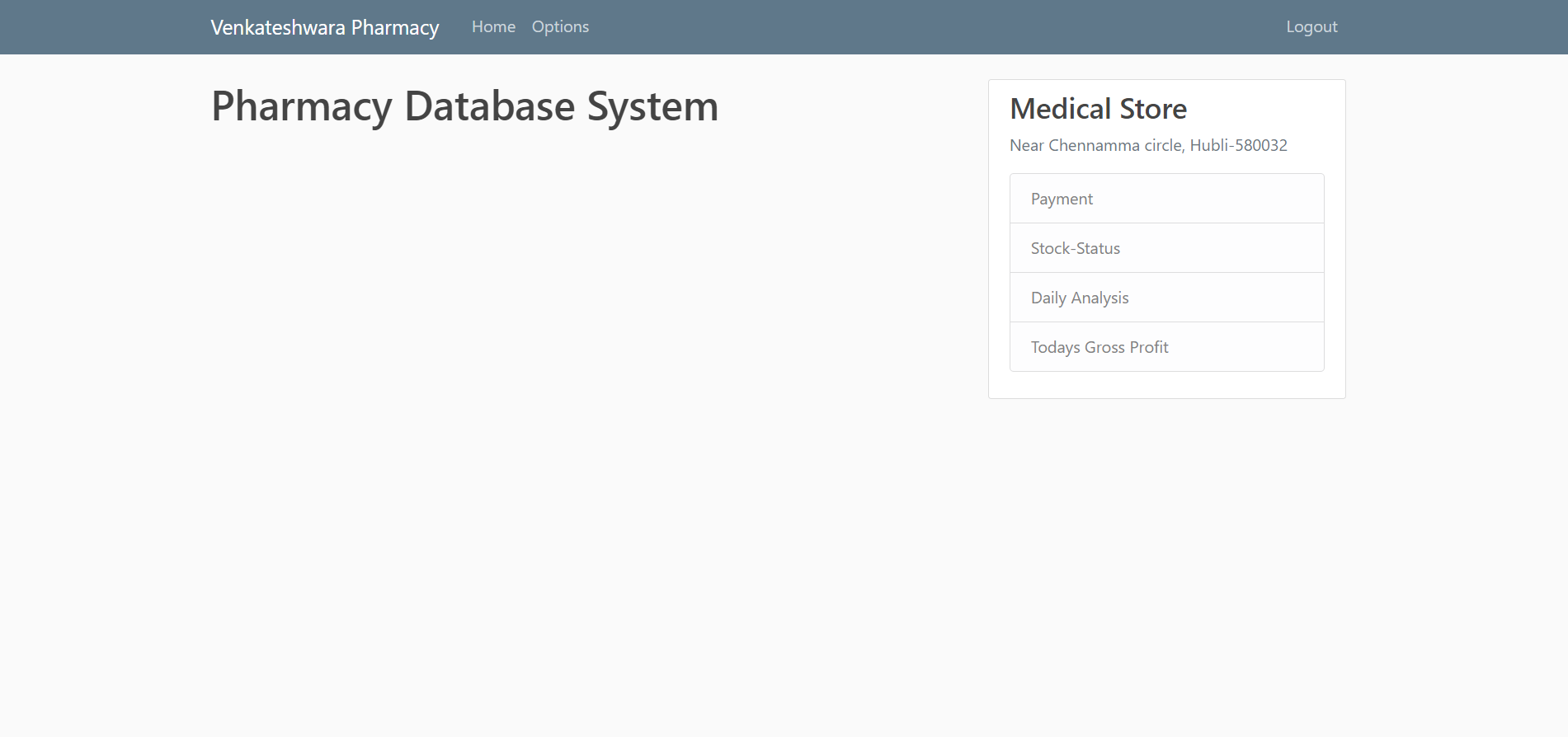
We Keep the normalized schema as the redundancy is reduced, as compared to the previous table.

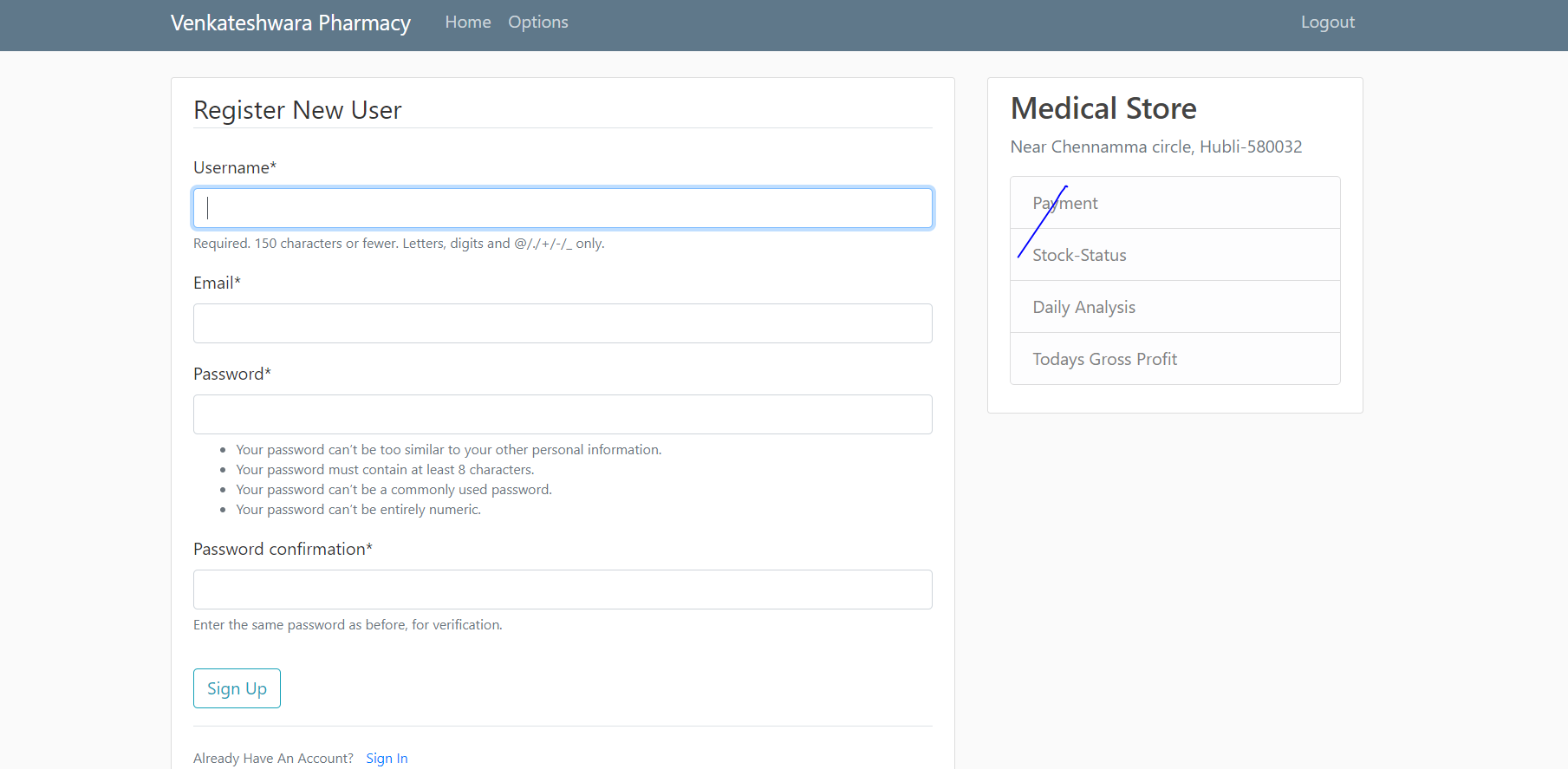
Further more as there doesn’t exist transitivity, and we have obtained tables with single primary keys that determine all other attributes and hence they are in 3NF, and further on in BCNF.

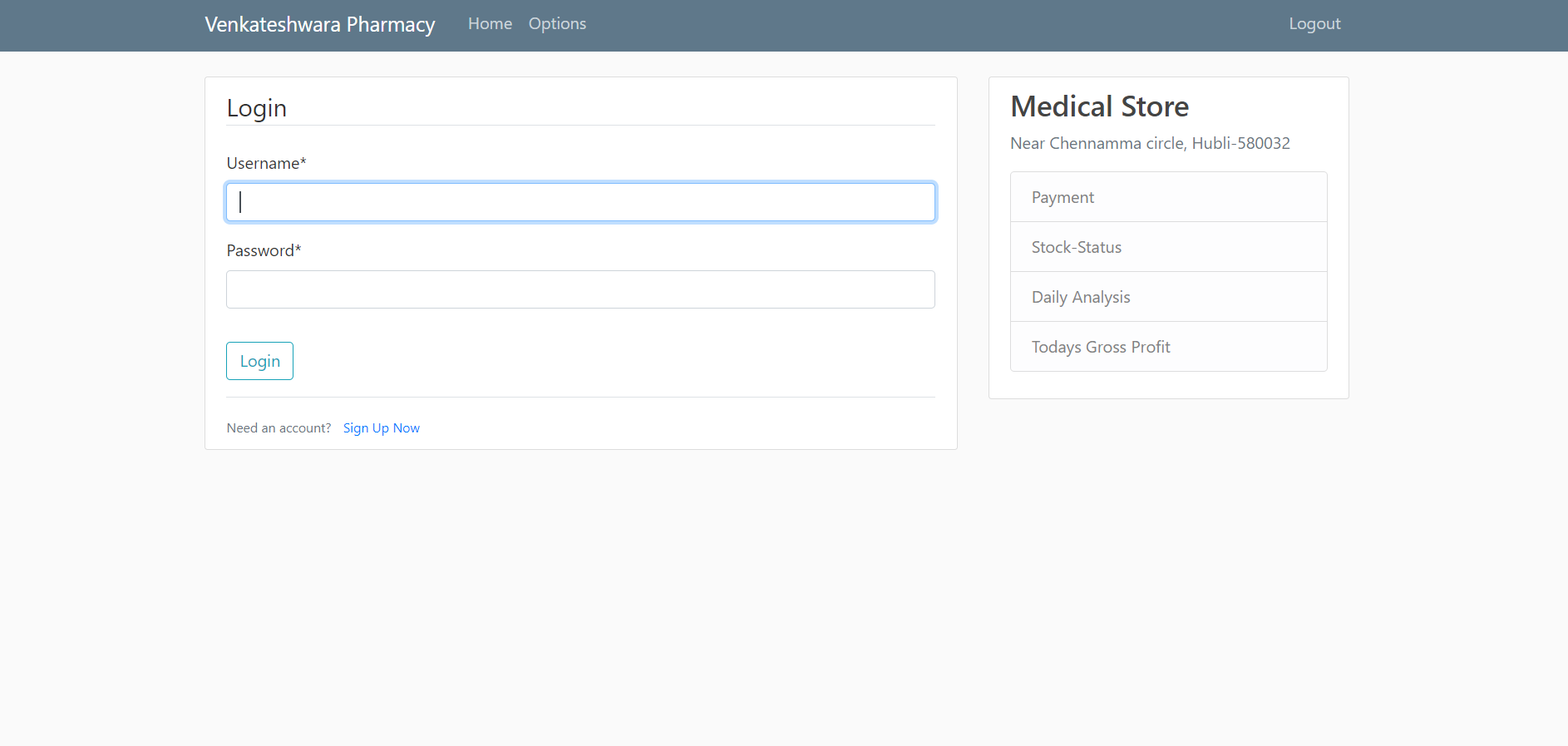
**Question 3:** User Interface (UI) design

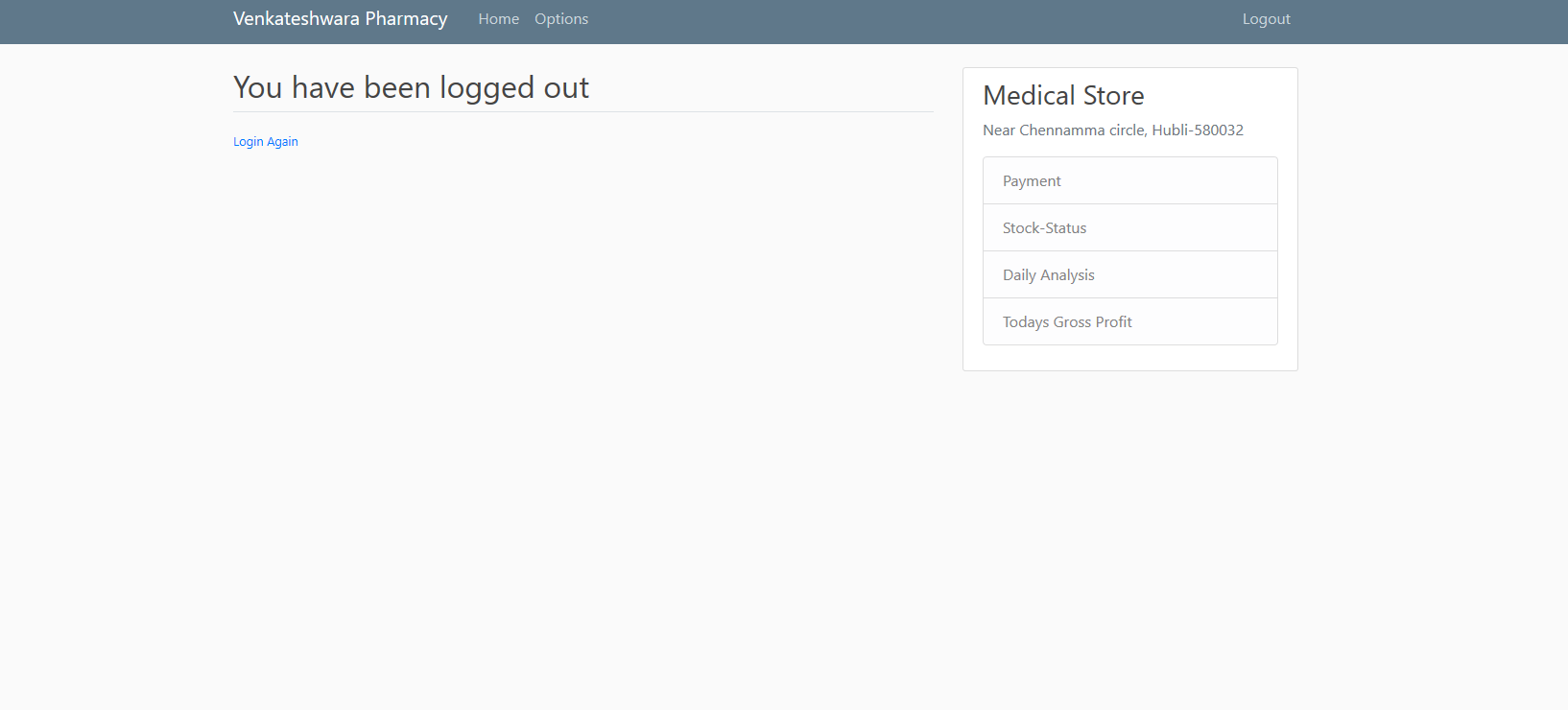
**USER-LOGIN FORM**

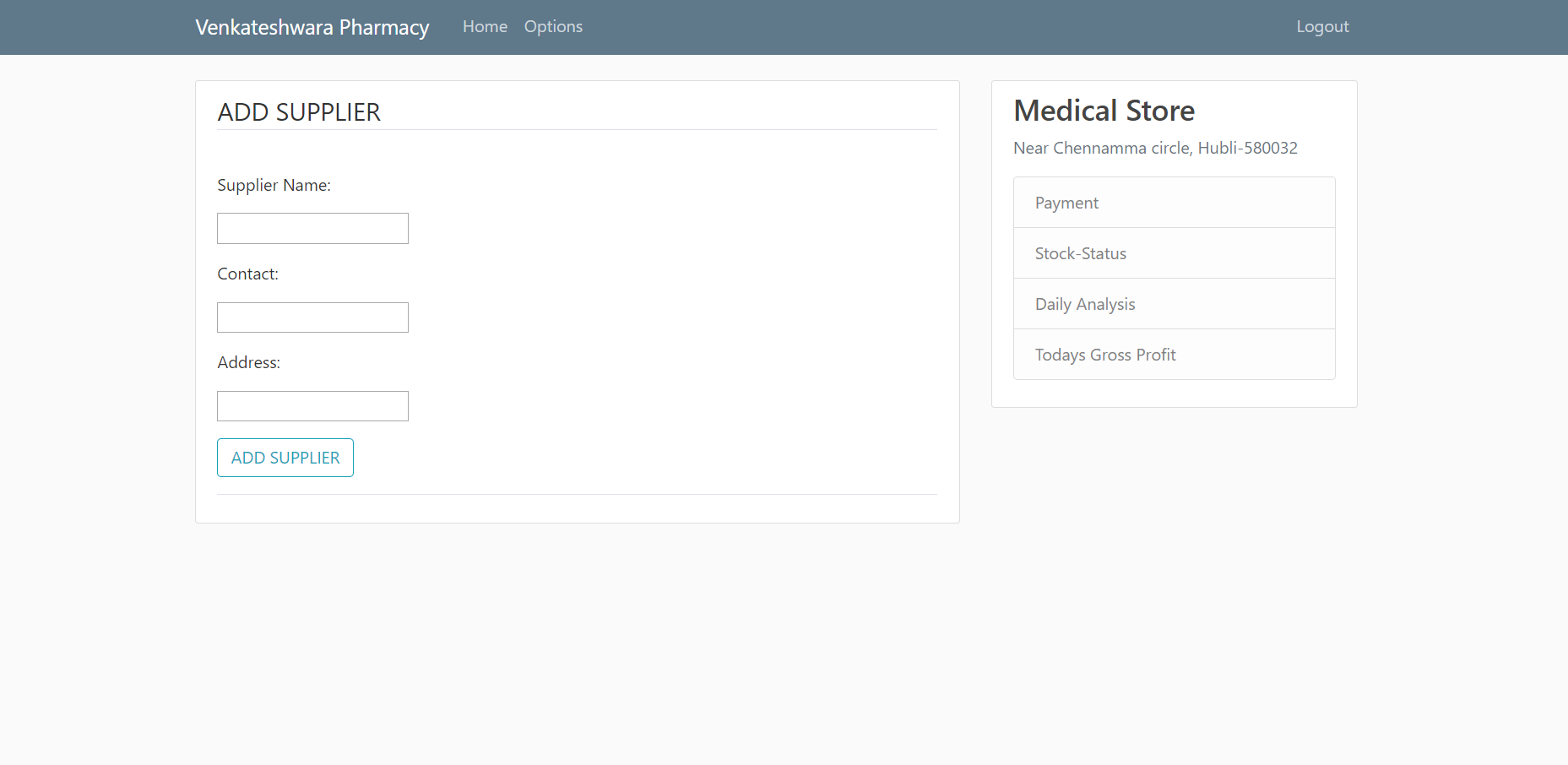


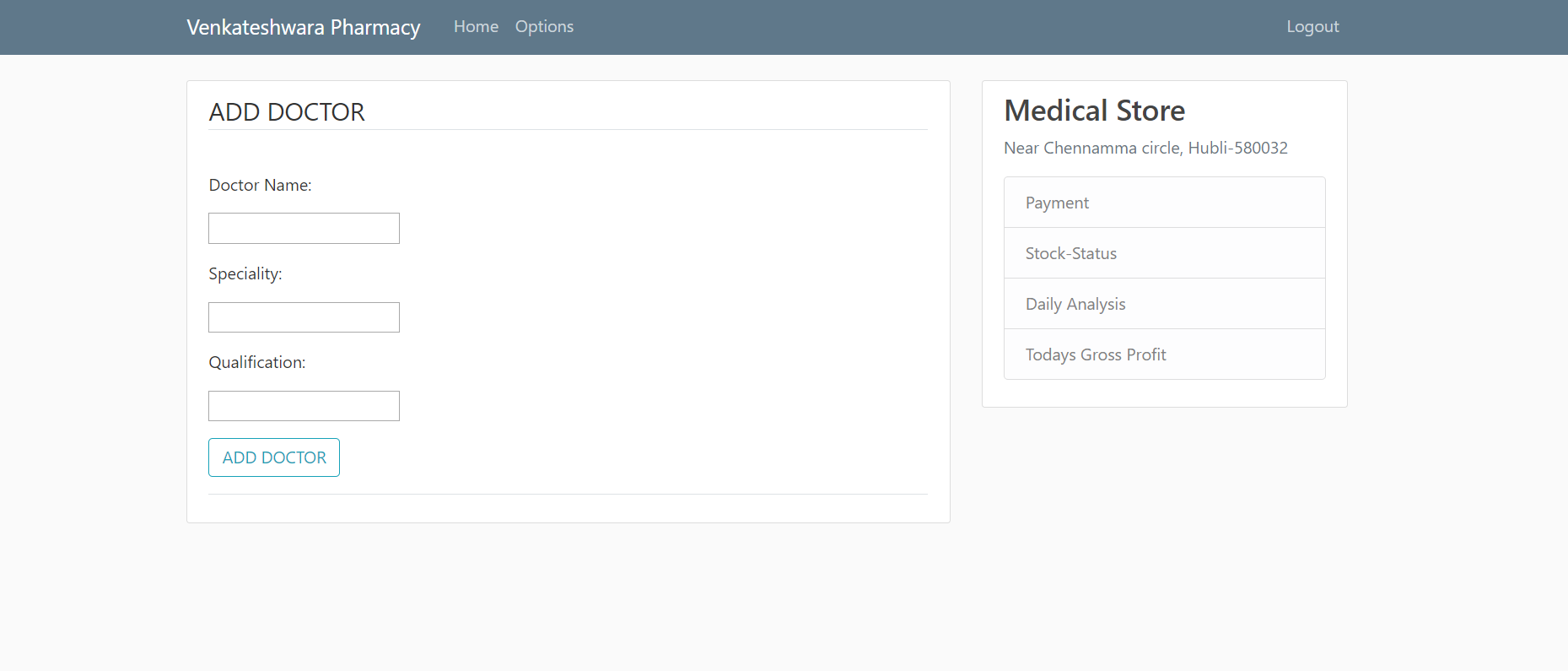


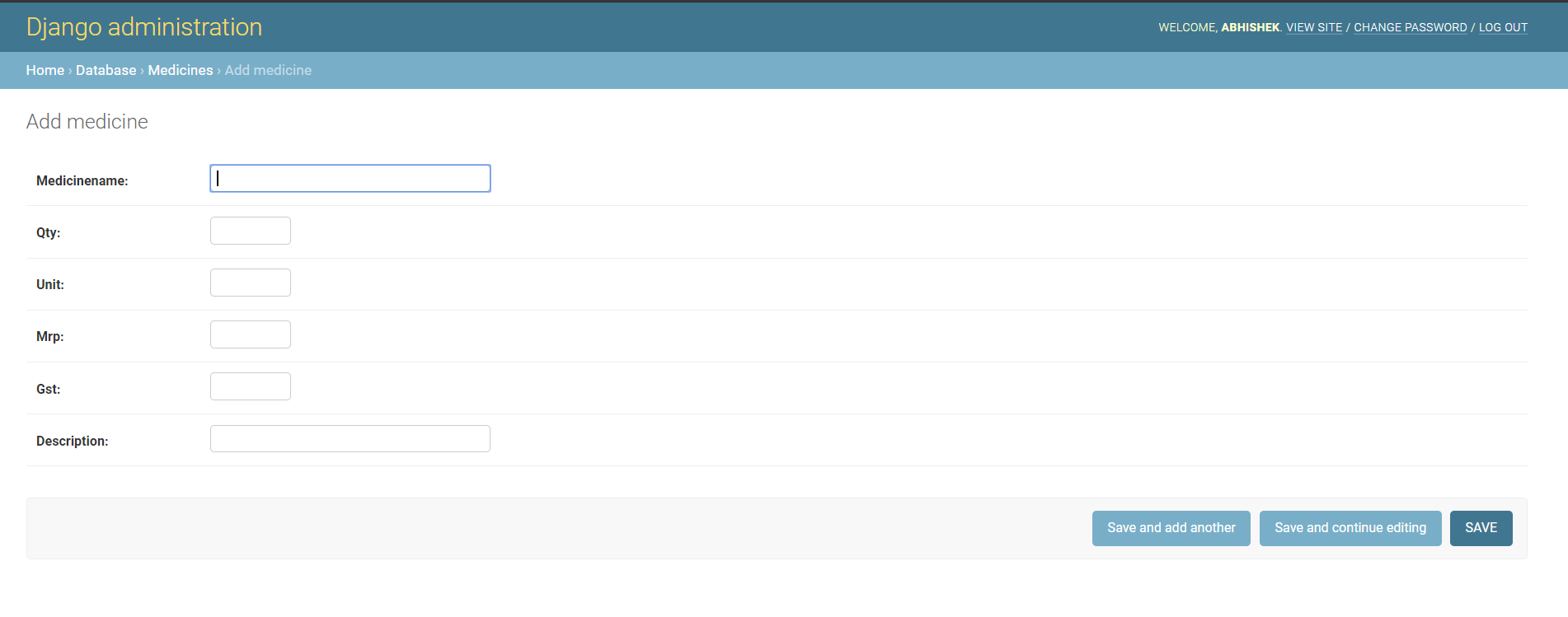


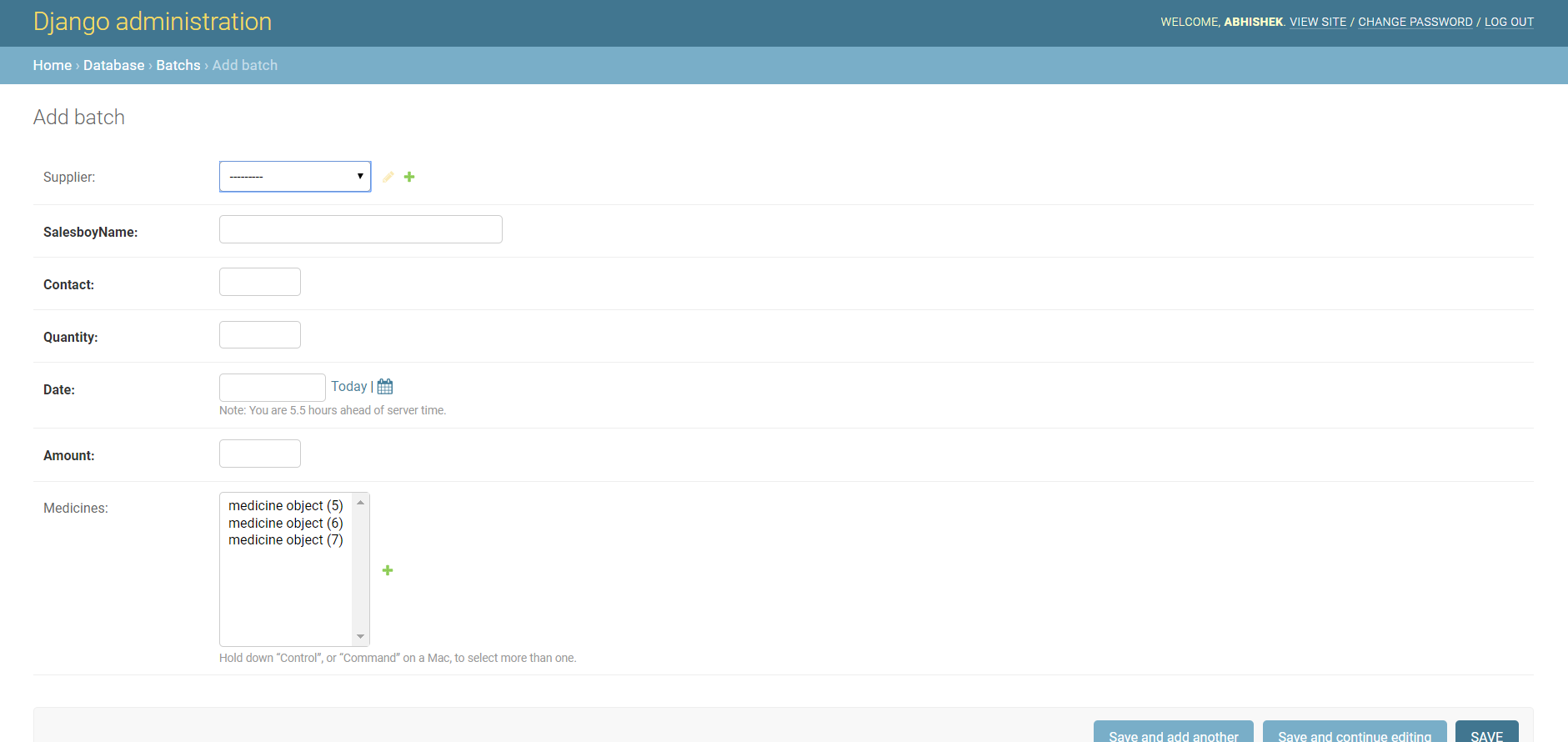


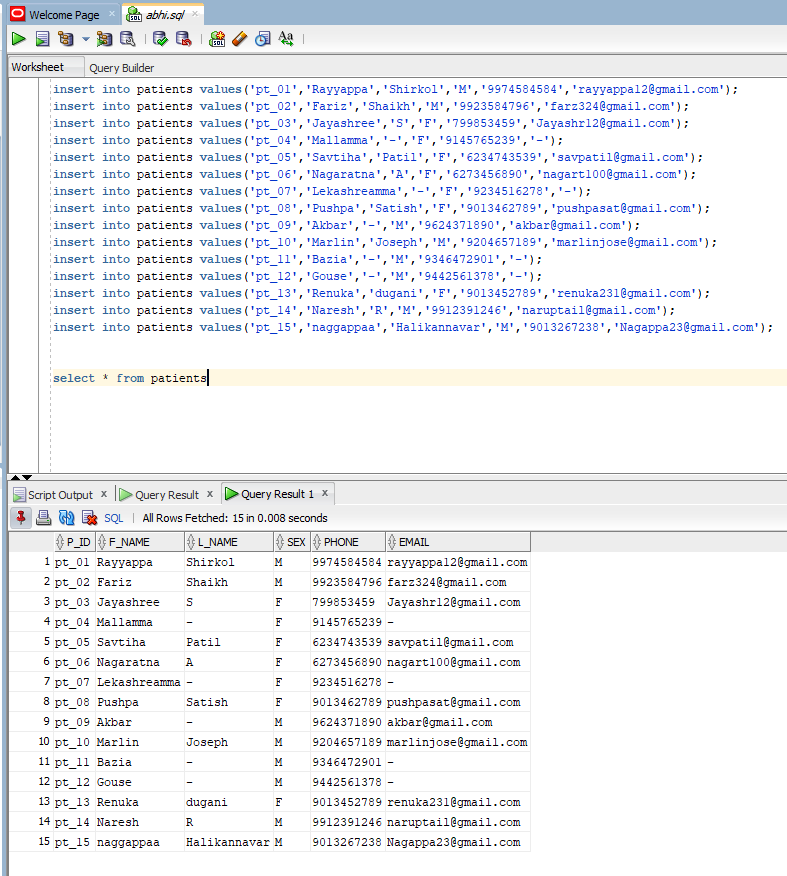


****

****

****

****

****

**Question 4:** Implemented requirements completed till date

The ER diagram and relation mapping have been reconsidered and been modified so as to have a most effective database system. Further normalization has been done to reduce redundancy and anomalous tuples, the user interfaces have been designed in accordance to the clients need, so as to meet the requirements as effectively as possible. The Database has been completely created and we have started entering the tuples. In the course of the next review everything would be done in the most suffice way

**References:**

**1.**Elmasri R. and Navathe S. , Fundamentals of Database Systems, 7th edition, Pearson Education, 2015.

2.draw.io.com

3.Django

**Submission Date: 22-03-2020**

**4.IMPLIMENTATION PHASE**

**Implementation Phase Report**

**Responsibilities:**

**Abhishek P.M: Responsible for creating the application and handling the frontend and backend of the system and modeling the relational database of the system according to the relational schema.**

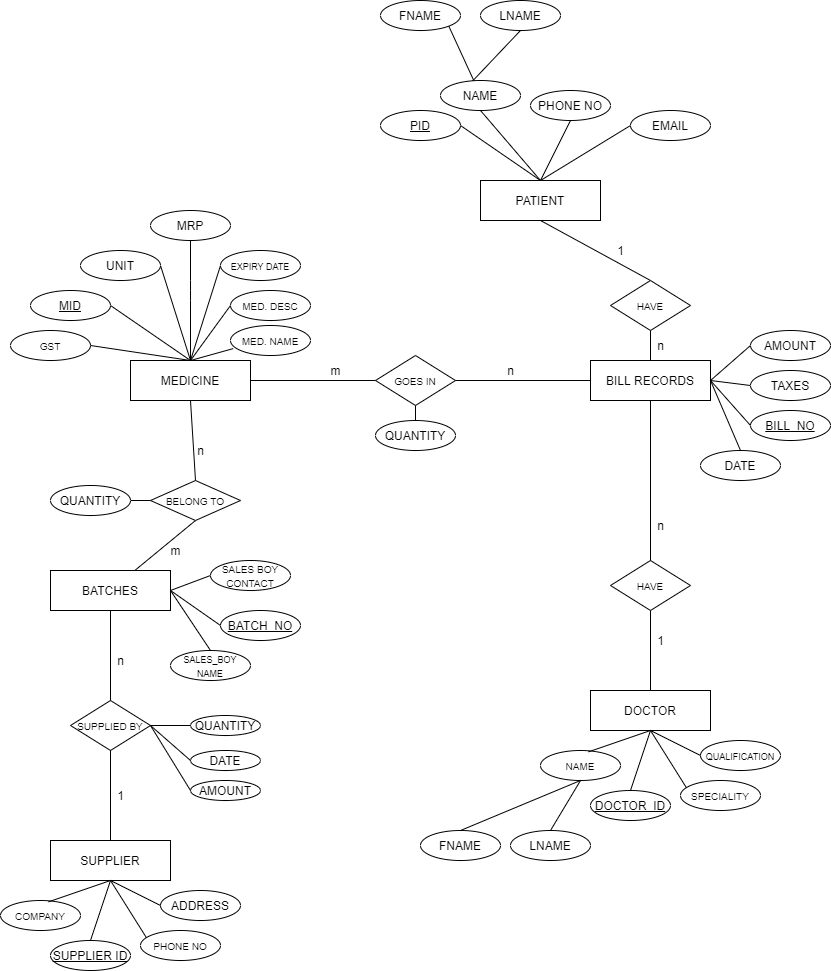
**Abhishek S: Responsible for handling the frontend and backend of the system and adding data into the database of the system and modeling the relational schema.**

**Aditya Mishra: Responsible for handling the database of the system and adding queries into the system, so that user has easy to use interfaces for the user to have easy access to all the required information.**

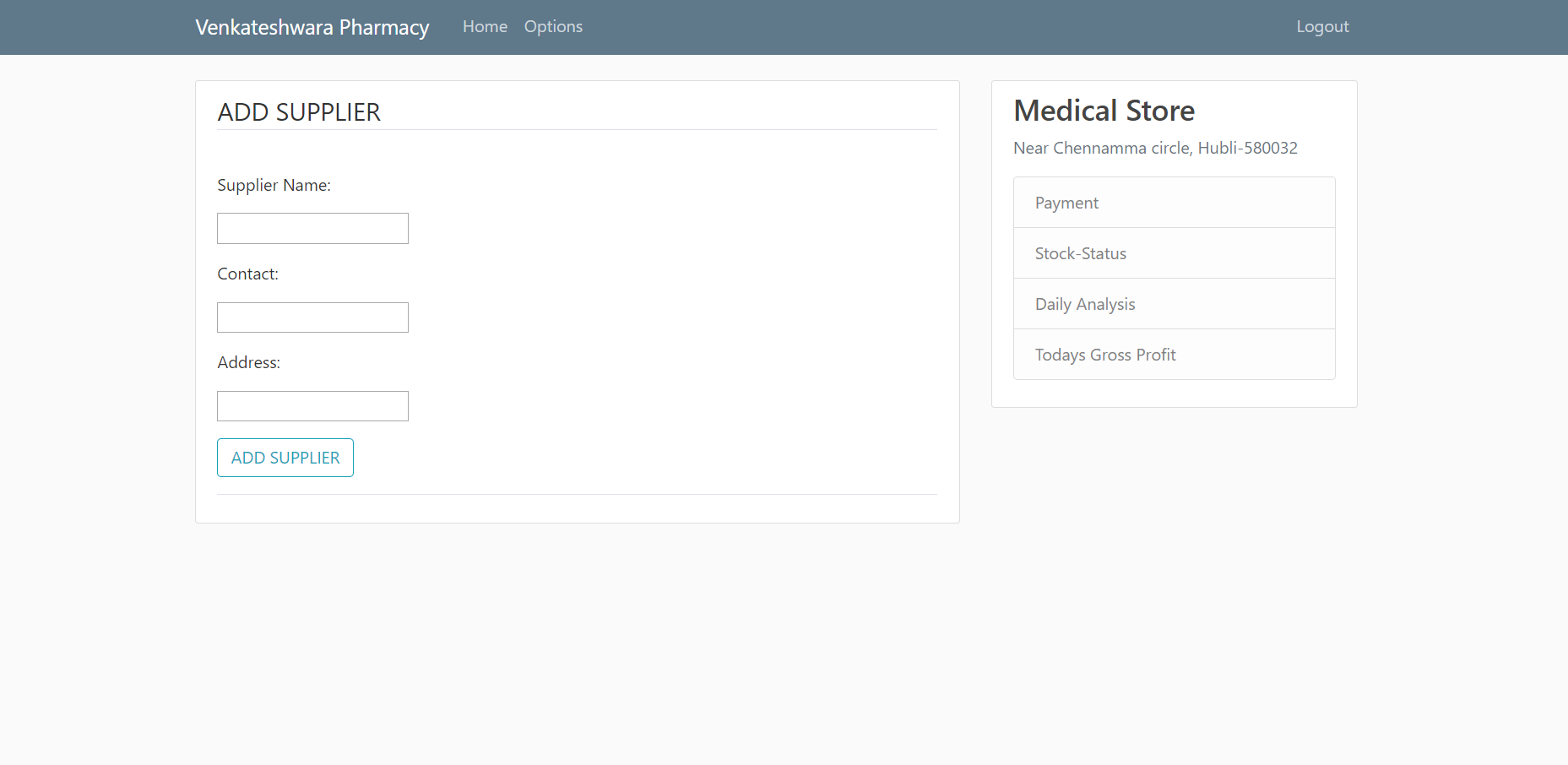
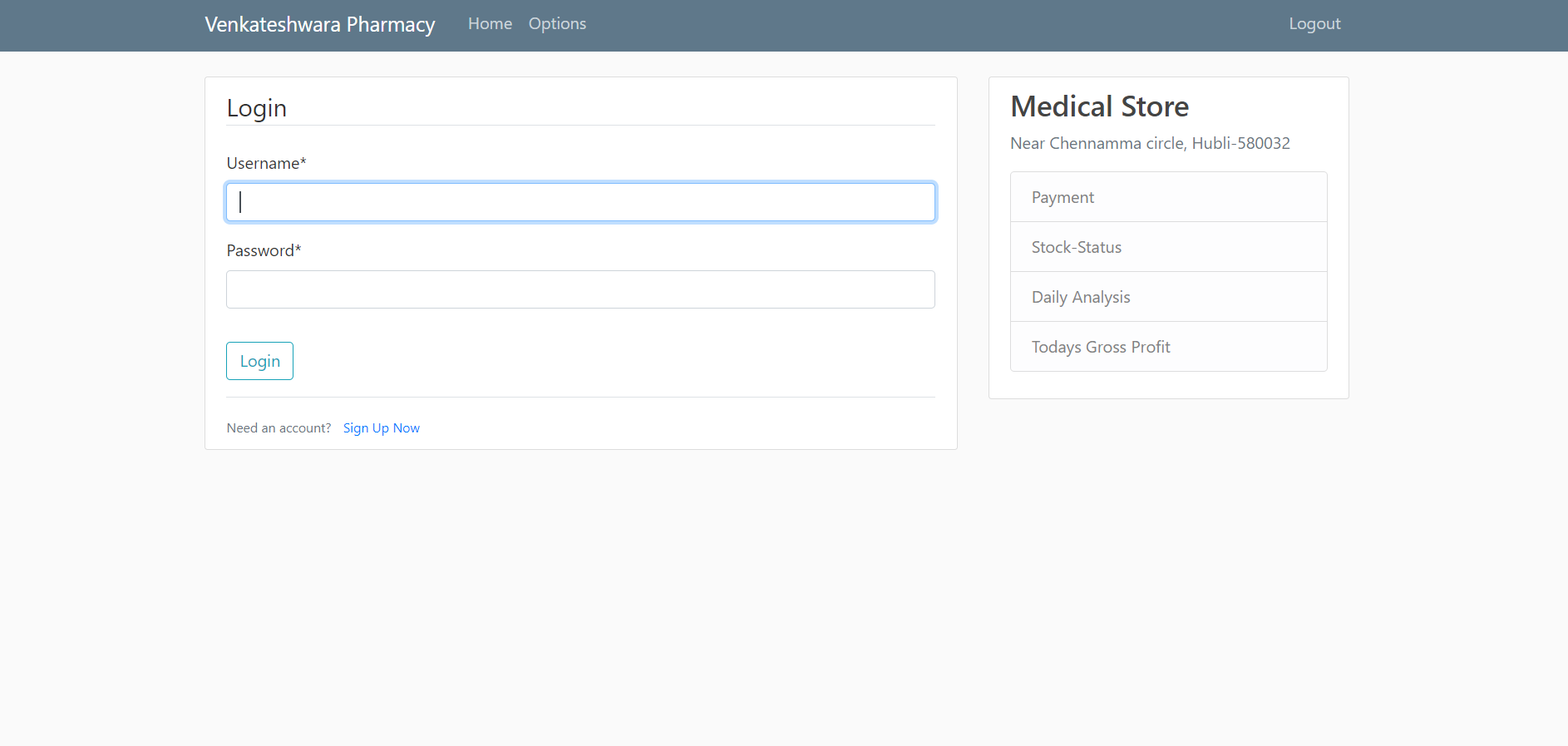
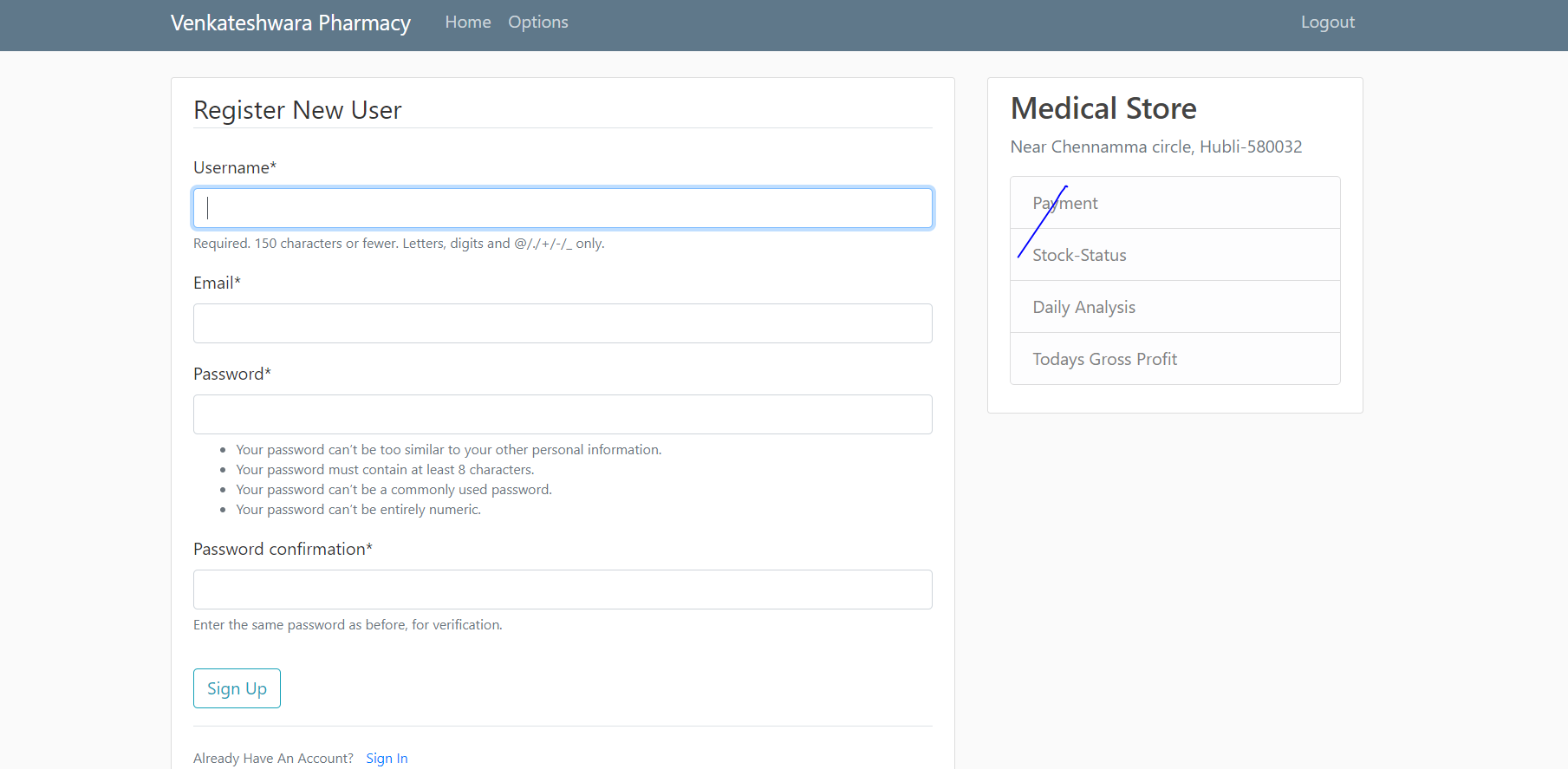
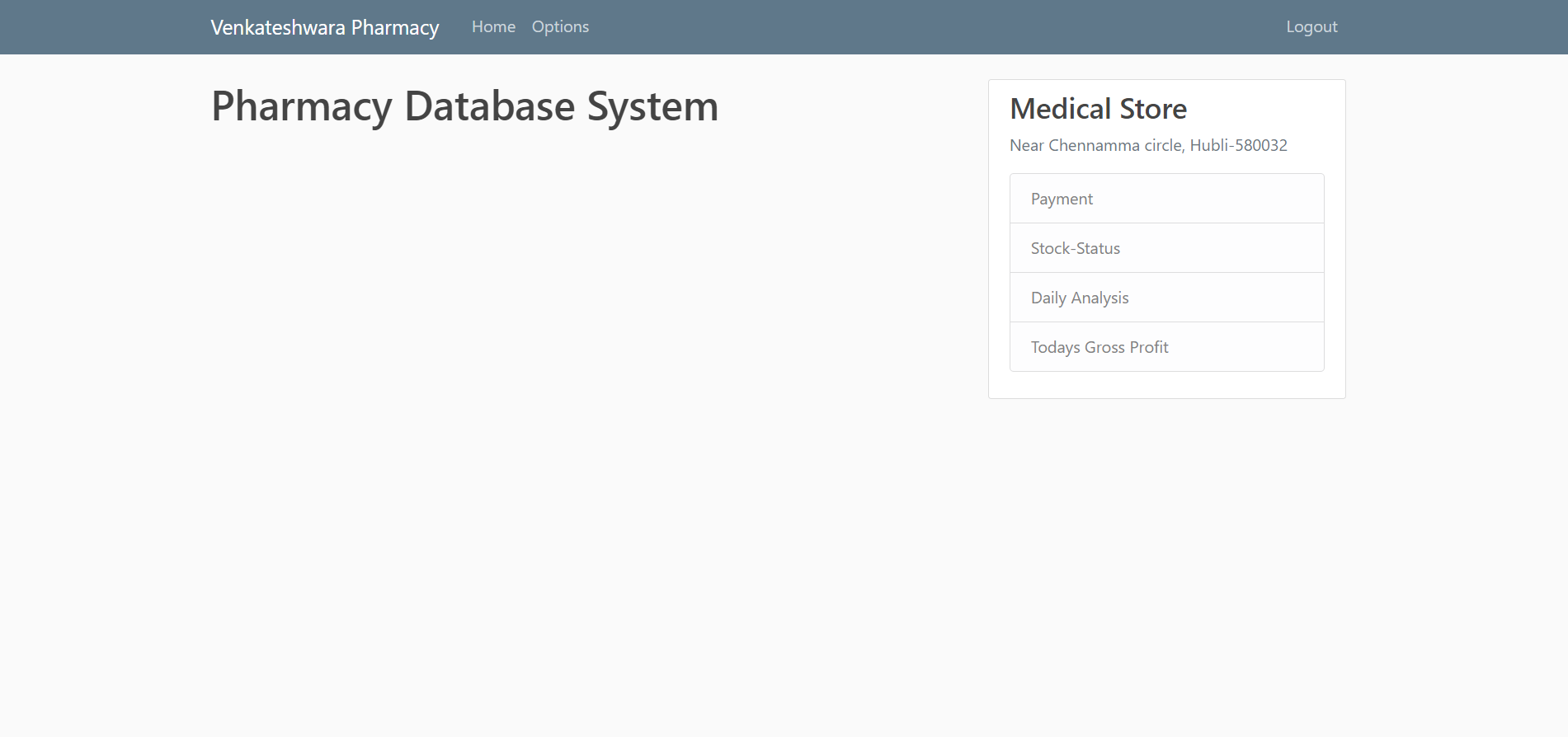
**Aman Khan: Responsible for adding data into the database and frontend of the system. And adding queries into the database.**

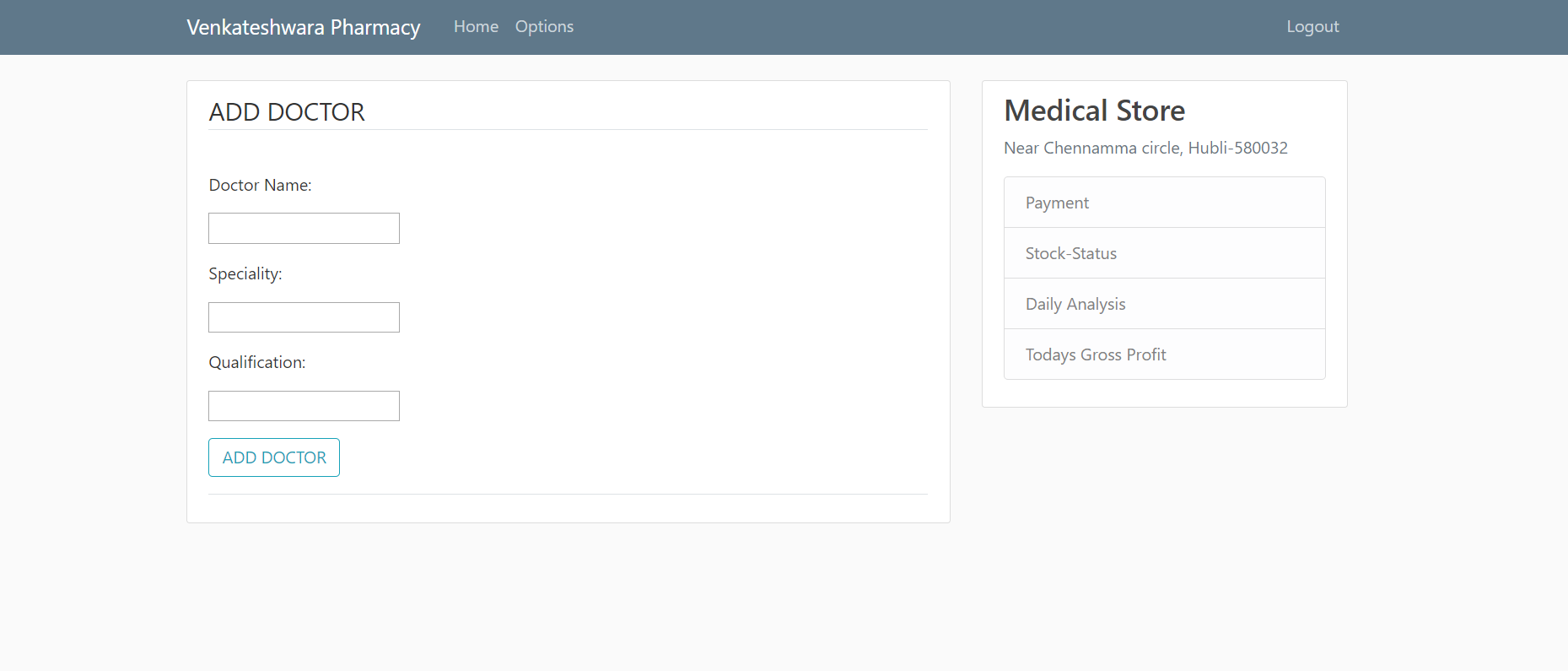
**Design updating**

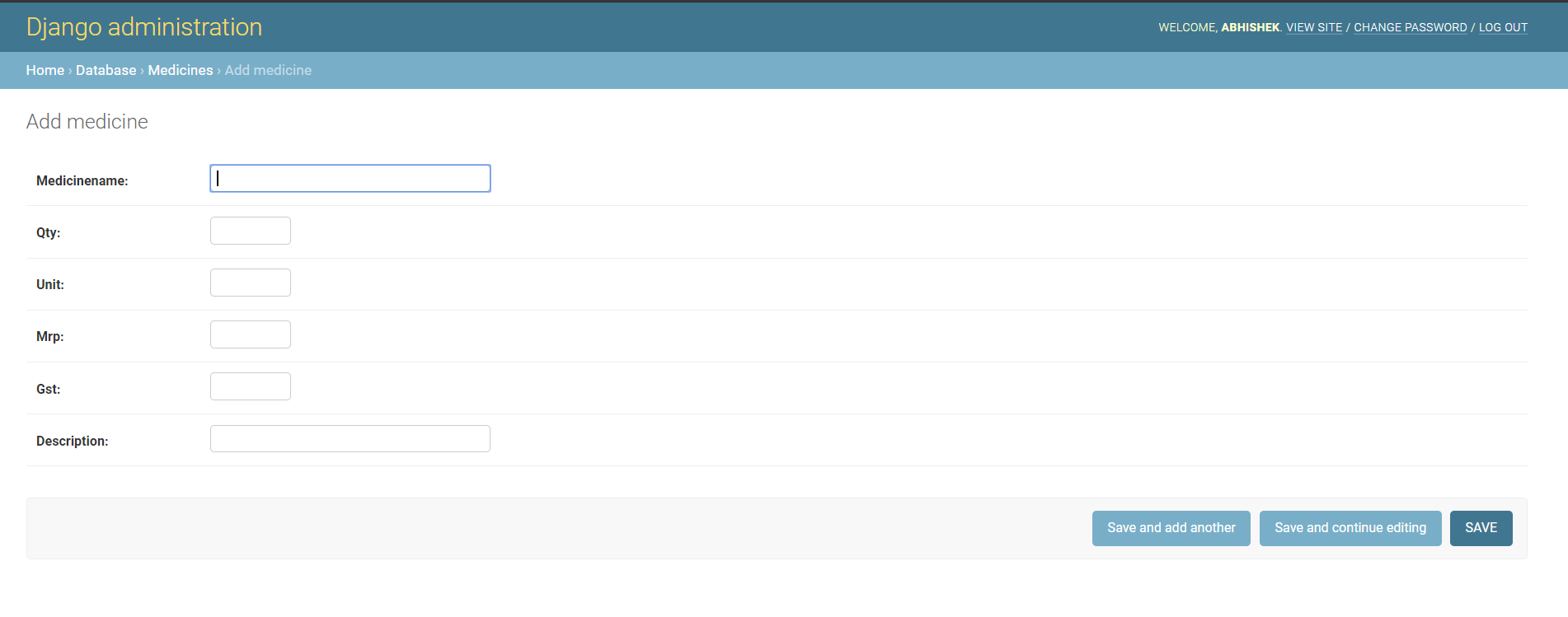
Our ER diagram remains unaltered as compared to the previous review



**Implementation Phase Questions to be answered**

******

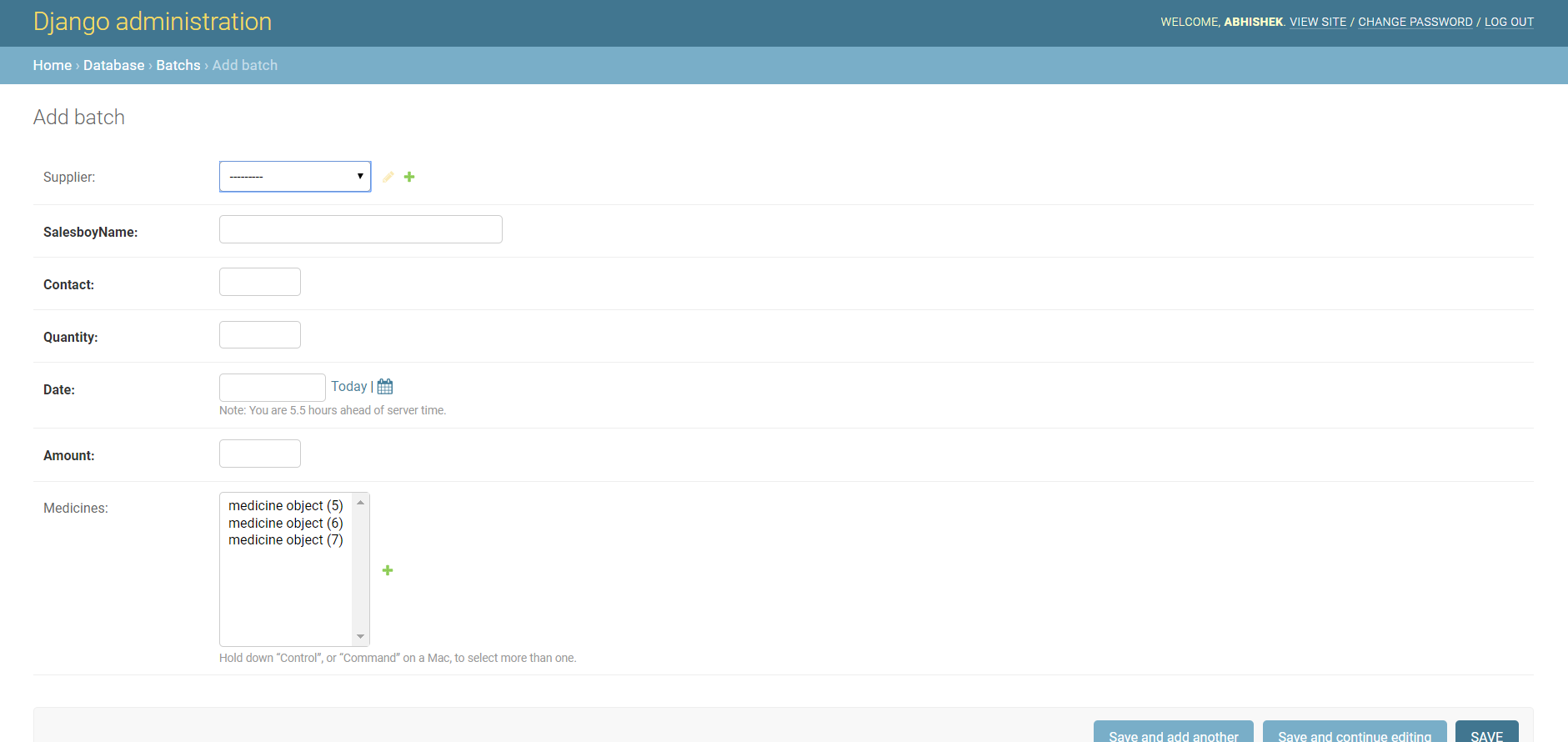
******

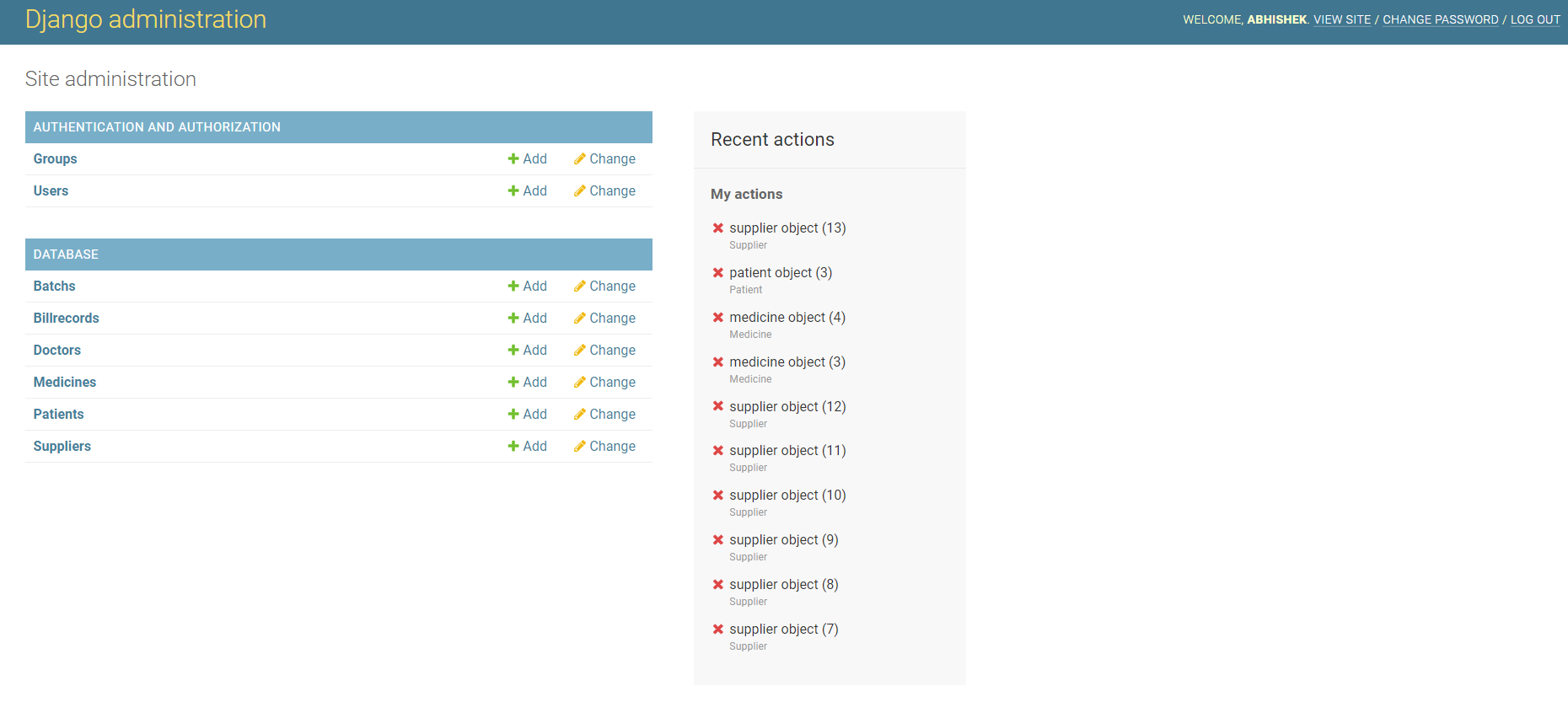
******

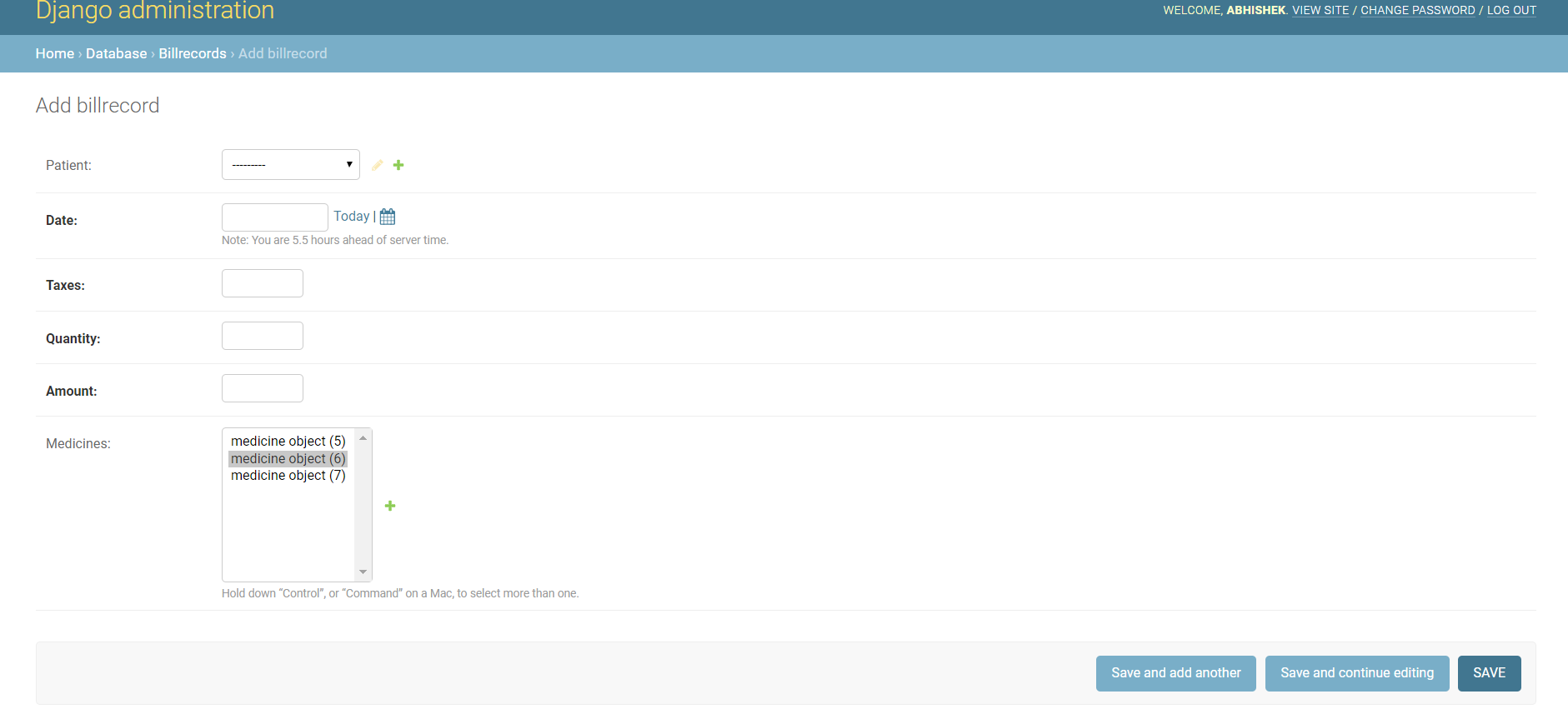
**Normalization:** [Any change in normal form as compared to design phase do include]

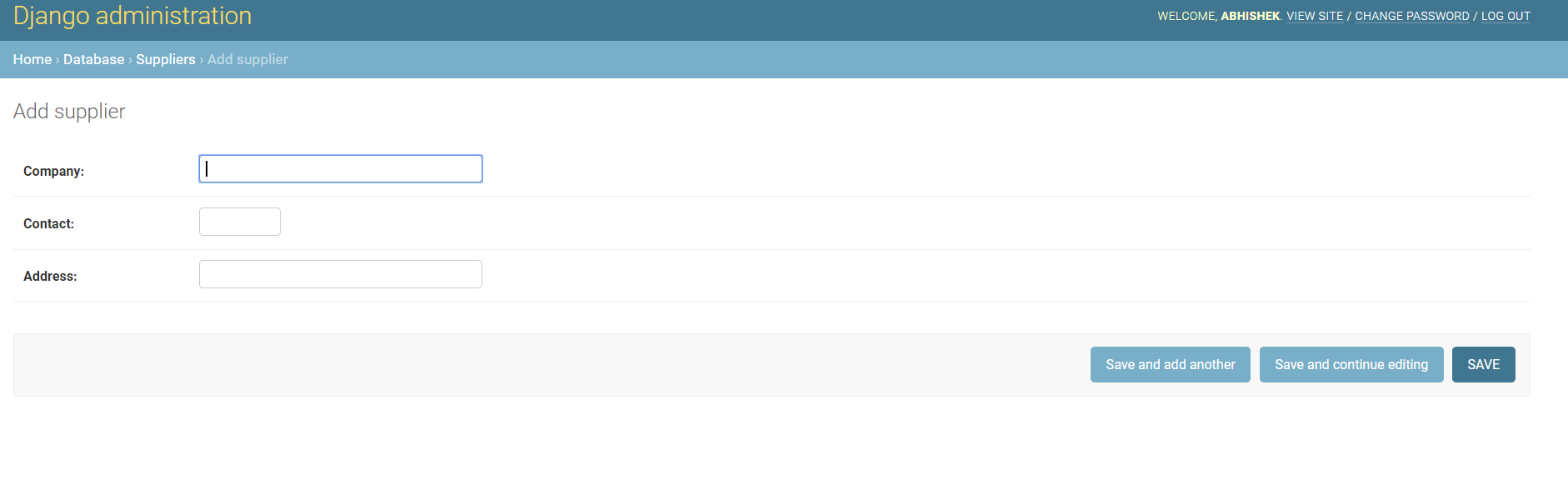
The Normalization remains the same as the previous review

**Question1:** Give the SQL statement(s) used to create the Oracle/MySQL database tables needed to implement the normalized relational schema.

******

******

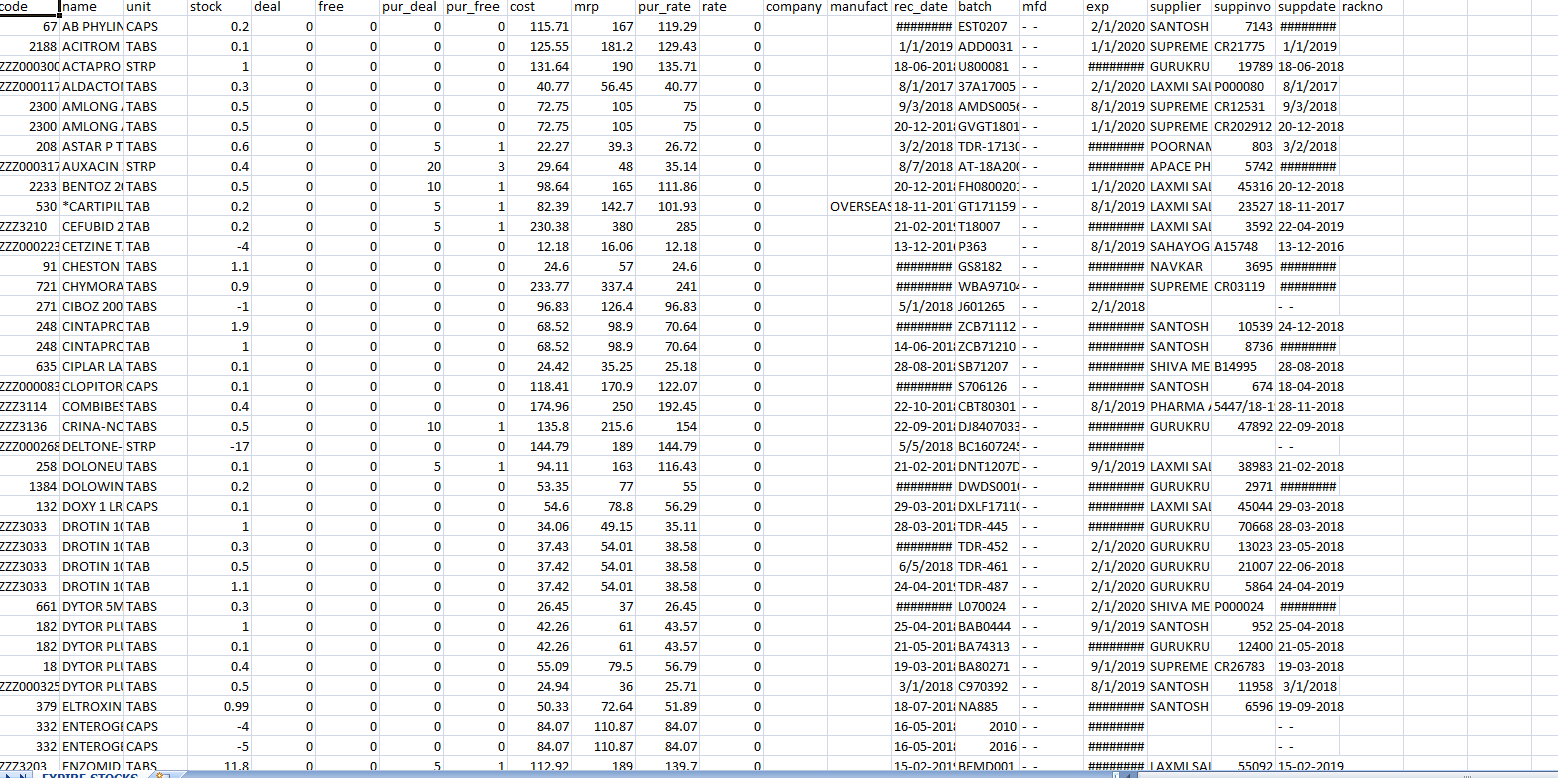
******

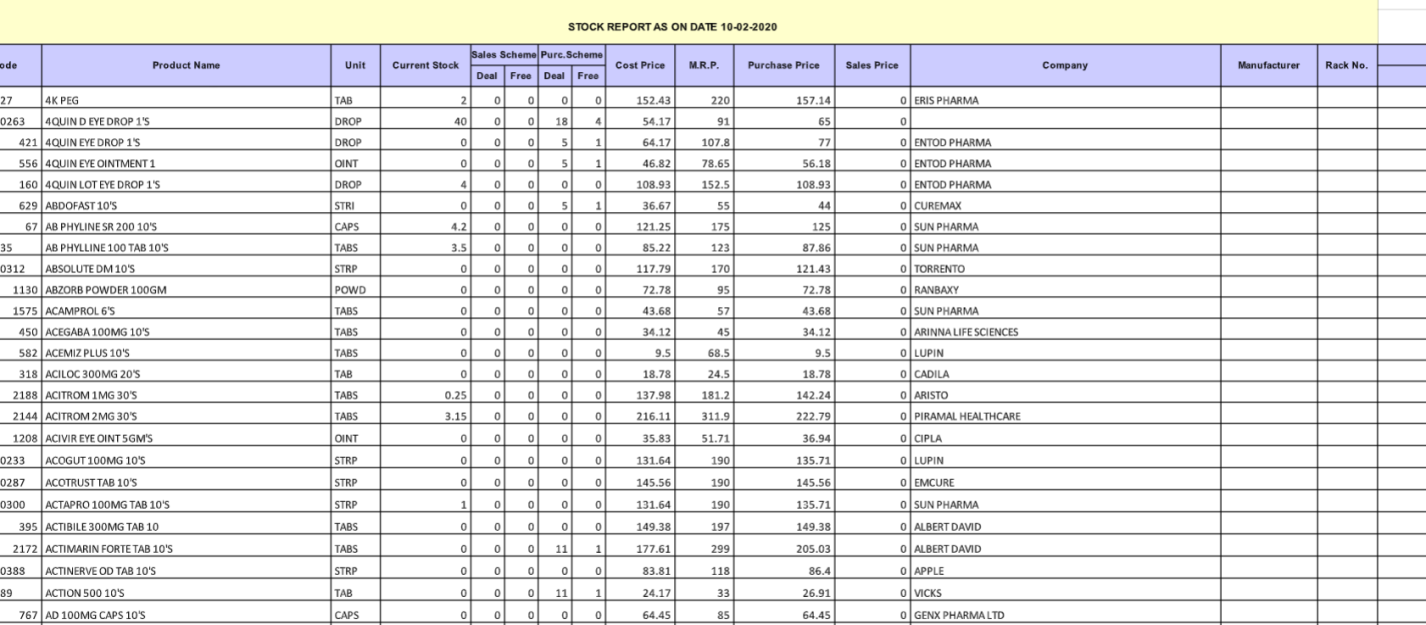
******

**Question2:** Give the actual data stored in each table of the database. (real sample data)

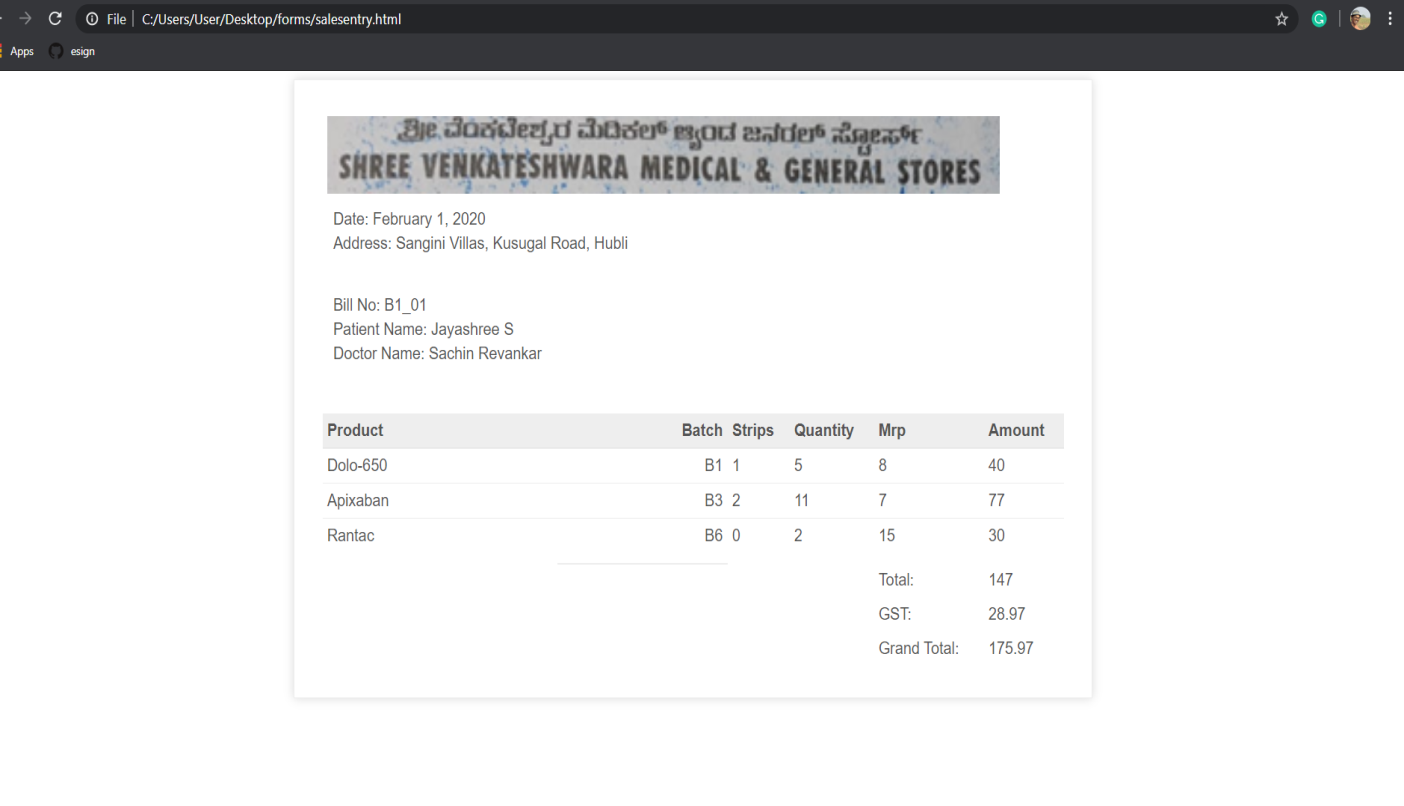
WRITE THE ANSWER HERE







**Question3:** Give all possible final bill reports/other forms of ledger reports summarized etc and graphs obtained by your application.



**Submission Date:05/04/2020**

**REFERENCES:**

1) Undraw.co

2) Bootstrap documentation

3) Web-development course on Udemy

4) Django documentation

5) Fundamentals of DataBase System, 7th Edition

**5. User Manual/README**

USERNAME: abhishek PASSWORD: abhipm123

The Following is a easy to use user manual for our client or anyone in need to make the best use of our application in times of need and enhance performance in their accountancy.

For a client who is a new user to our application. The client shall be able to create a new account for all his employees and shall be able to fetch the sales that every particular employee of his does, and will be able to properly manage accounts. The New user has to validate his existence by filling the form, by entering a non existing username and a strong password that is validated and then accepted. The users can now login and logout from the application.

Once the user is in the application, all the WebPages are according to the client’s needs. The Client will be able to make all or any changes that he wished to do. Out of all the available options that are listed in our navbar for the user at the top.

At first, the user will be able to add medicines to the list of existing medicines to the shop. Once this is done, the user will be able to add the shop stocks and view all the stocks that are available in the shop at any given point of time. The Batches that the medicines come in and the expiry date of each and every medicine is also stored in the database which can be accessed for later use. There can be orders that can be added online through all the suppliers that are listen in the system of the database and also for all the orders, transactions can be added and can be made a list of. We have also added bar and pie graph visualisations for the customer to easily make sure of all the transactions and orders are easily maintained and analysed from the client point of view.

We have also added to the list, all the expenses that the client of ours has undergone and that will have a detailed analysis of all the expenses that the client has undergone and will give us analysis on a bar graph basis on what expenditure how much amount is spent and also gives us a top 5 analysis of the highest expenditures that happen at the pharmacy.

The Billing problem that existed with our client was to be rectified has he had people who used to keep debts of money to pay later and also had costumers who pay on the spot itself and hence it is necessary for us to properly give him a detailed analysis of the debt and the amount that is in account of the shopkeeper at any point of time.

We can query the bills at any given point of time. That is all the bills that exists or purchases that have happened between particular dates are displayed.

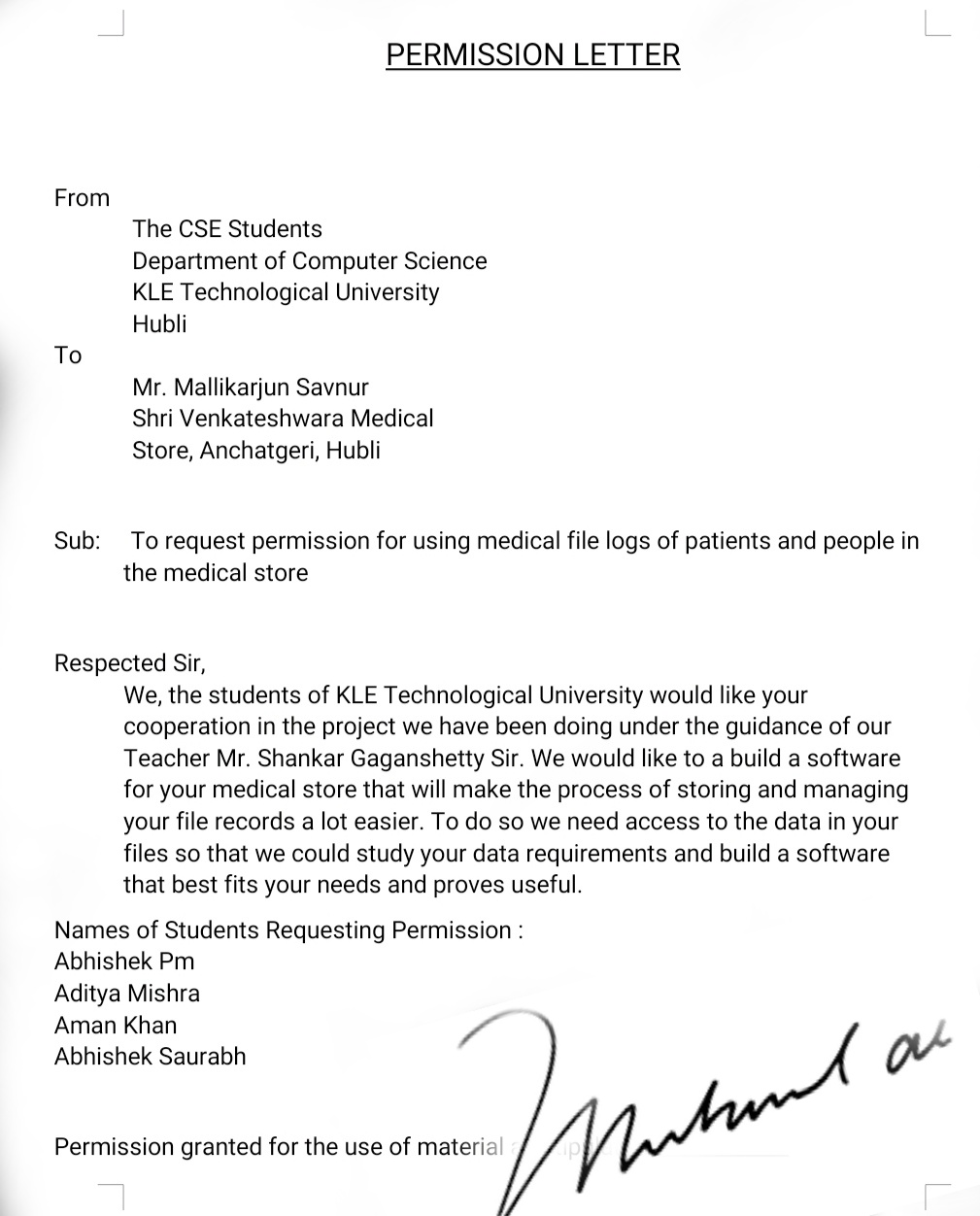
Once the customer enters the details of the customer and the nearby doctor that has treated him. The client will now be able to enter the details of purchase of the medicines and the quantities and the price is automatically generated and the option of paid or not is available through which hee can keep an account. And hence all the requirements are satisfied.

1. **Snap shots of Client and Office and Workplace (Venkateshwara Medical Pharmacy)**





**INTIAL ACCEPTANCE LETTER**



**CUSTOMER FEEDBACK FORM**

Here are a few questionnaires that we had prepared for pour customer after we sent him the application for test.

Q1) Does the application satisfy all your needs?

Ans. To some extent we might rely on the application, but the process is still not simplified as it is still tedious to select every medicine every time there is an in stock that comes in from any of the supplier, nor there is a option to maintain the doctor service number and add gst number to the suppliers. Then it would be quite easy for us to simplify the whole process.

Q2) what features did you like in the application?

Ans. The particular feature of analysis and the feature of bills that are to be accessible between particular dates are useful and bill details of any particular customer at any point of time are accessible and made easy to use. We even liked the feature of the amount analysis that shoes the percentage of money that is in dept and the amount of money that has been paid in the store.

Q3) What Changes if in future would you recommend us to make?

Ans. We would want you’ll to add features where we can print the bill and export sales to a detailed analysis and also fetch the details of patients who have purchased from our store and also add essential discount percentages that we sometimes need.

As the circumsantances weren’t well, we could only talk to our client on call and only take this input and just give him a online demo .