VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi, Karnataka - 590 018

A DBMS Laboratory mini- Project Report
on

"MEDICINES SUPPLY MANAGEMENT SYSTEM"

Submitted in partial fulfilment of the requirements for the award of Course completion in degree of

Bachelor of Engineering

in

Computer Science and Engineering

Submitted By

NAMITHA C **1KI21CS066**

Under the Guidance of

Prof. Niranjan S.J

Assistant Professor.

Department of Computer Science and Engineering

Kalpataru Institute of Technology

Tiptur - 572 201.

2023-20

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590 018.

KALPATARU INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering

Tiptur-572 201.

CERTIFICATE

This is to certify that the DBMS Laboratory mini- Project report entitled **“MEDICINES SUPPLY MANAGEMENT SYSTEM ”** is a bonafied work carried out by **NAMITHA C** bearing USN:**1KI21CS066** in partial fulfilment for the completion of the laboratory work of the course subject **“DATA BASE MANAGEMENT SYSTEM ”** with course code **“21CSL55”**, during the year 2023-2024. It is certified that all the corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of course activity work prescribed for the said degree.

Guide	**HOD**
Prof. Niranjan S.J	**Prof. Shashidhara M S**
Assistant Professor.	**Associate Professor & HOD**
Dept of CSE	**Dept of CSE**

KALPATARU INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering

Tiptur-572 201.

Acknowledgement

The satisfaction and great happiness that accompany the successful completion of any task would be incomplete without mentioning about the people who made it possible. Here we make an honest attempt to express our deepest gratitude to all those who have been helpful and responsible for the successful completion of our project.

We would like to thank **Dr.G D GURUMURTHY, Principal**, Kalpataru Institute of Technology, Tiptur for his continuous support and encouragement throughout the course of this project.

We would like to thank **Prof. SHASHIDHARA M S, Head of Department**, Department of Computer Science and Engineering, Kalpataru Institute of Technology, Tiptur for his continuous support and encouragement throughout the course of this project.

We are immensely indebted to our internal guide Prof. Niranjan S and Dr Raviprakash M L, Department of Computer Science and Engineering, Kalpataru Institute of Technology, Tiptur for his support, technical assistance and constructive suggestions and guidance for successful completion of our project. We are very much thankful to him for the encouragement that has infused at most real into us to complete the project work.

We would like to thank all faculty members of Department of Computer Science and Engineering, KIT, Tiptur, our family members, and to our friends who are directly or indirectly responsible for our success.

Thanking you

NAMITHA C
1KI21CS066

KALPATARU INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering

Tiptur-572 201.

Declaration

I, **NAMITHA C**, student of Fifth semester Bachelor of Engineering, Department of Computer Science and Engineering, Kalpataru Institute of Technology, Tiptur, would hereby declare that the Course Activity work entitled **"DATA BASE MANAGEMENT SYSTEM LABORATORY - 21CSL55"** has been carried out by me at Kalpataru Institute of Technology and submitted in partial fulfillment of the course requirement for the award of degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi, during the academic year 2023-2024. I further declare that the work embodied in this report has not been submitted to any other university or institution for the award of any other degree.

Place: Tiptur
Date :

Signature of student

ABSTRACT

Medicines Supply Management in healthcare systems is evaluated with a particular focus on the distribution of medicines from a wholesaler to clinics. Currently, there are issues with service levels to clinics that need addressing. The value of the paper arises from providing a detailed analysis of a healthcare supply chain in the developing world and diagnosis the parameters involved in inventory. Pharmaceutical practices have evolved over time to become fully encompassed in all aspects of pharmacy itself. Such practices include: dispensing of drugs, consultation, drug regulation, and the sale of these drugs. Creating an Online Pharmaceutical Management System would help in pharmaceutical practices for all parties involved. It is eminent that the system provides a safe, secure and verified platform for all parties which help to bridge the communication gap and provide legitimate drugs. Therefore, if all recommendations are strictly adhered to, there will be strict monitoring and regulation of how drugs are circulated and a decrease in the spread of fake drugs.

TABLE OF CONTENTS

CHAPTER-1:

INTRODUCTION

1.1 OBJECTIVES:

• The main objective of the project is to design and develop a user friendly-system

• Easy to use and an efficient computerized system.

• To develop an accurate and flexible system, it will eliminate data redundancy.

• To study the functioning of pharmacy supply management System.

• To make a software fast in processing, with good user interface.

• To make software with good user interface so that user can change it and it should be used for a long time without error and maintenance.

• To provide synchronized and centralized farmer and seller database.

• Computerization can be helpful as a means of saving time and money.

• To provide better Graphical User Interface (GUI).

• Less chances of information leakage.

• Provides Security to the data by using login and password method.

• To provide immediate storage and retrieval of data and information.

• Improving arrangements for medicines coordination.

• Reducing paperwork.

CHAPTER-2

STUDY OF EXISTING SYSTEM

2.1 CASE STUDY:

Rising debt, cost-cutting, and layoffs in health care-delivery facilities, alluded to earlier. Models for the design and operation of supply chain networks may be steady state or dynamic and may be deterministic or deal with uncertainties (particularly in product demands). Research in this field started very early on, with location-allocation problems forming part of the earlier set of ‘‘classical’’ operations research problems. The gap between the growing demand and available supply of high-quality, cost effective, and timely health care continues to be a daunting challenge not only in developing and underdeveloped countries, but also in developed countries. Further, the issues involved with the supply chain design in developing countries are prevalent in developed countries, especially with the rising number of uninsured and jobless among the patient populations and with the budget deficits. Thus the project is a sincere effort in simplifying the tasks of administrators in an easily usable format.

2.2 PROPOSED SYSTEM:

While there has been no consensus on the definition of Pharmacy Supply Management in the literature, they have proposed that researchers adopt the below definition to allow for the coherent development of theory in the area. In order to have a successful supply management, we need to make many decisions related to the flow of information, product, and funds. Each decision should be made in a way to increase the whole supply chain profitability . Supply management is more complex in healthcare and other industries because of the impact on people’s health requiring adequate and accurate medical supply according to the patient’s need.

CHAPTER 3

SOFTWARE ENVIRONMENT

3.1 SOFTWARE REQUIREMENTS:

Frontend- HTML, CSS, Java Script, Bootstrap
Backend-Python flask (Python 3.7) , SQLAlchemy,
• Operating System: Windows 10
• Google Chrome/Internet Explorer
• AMPPS (Version-3.7)
• Python main editor (user interface): PyCharm Community
• workspace editor: Sublime text 3

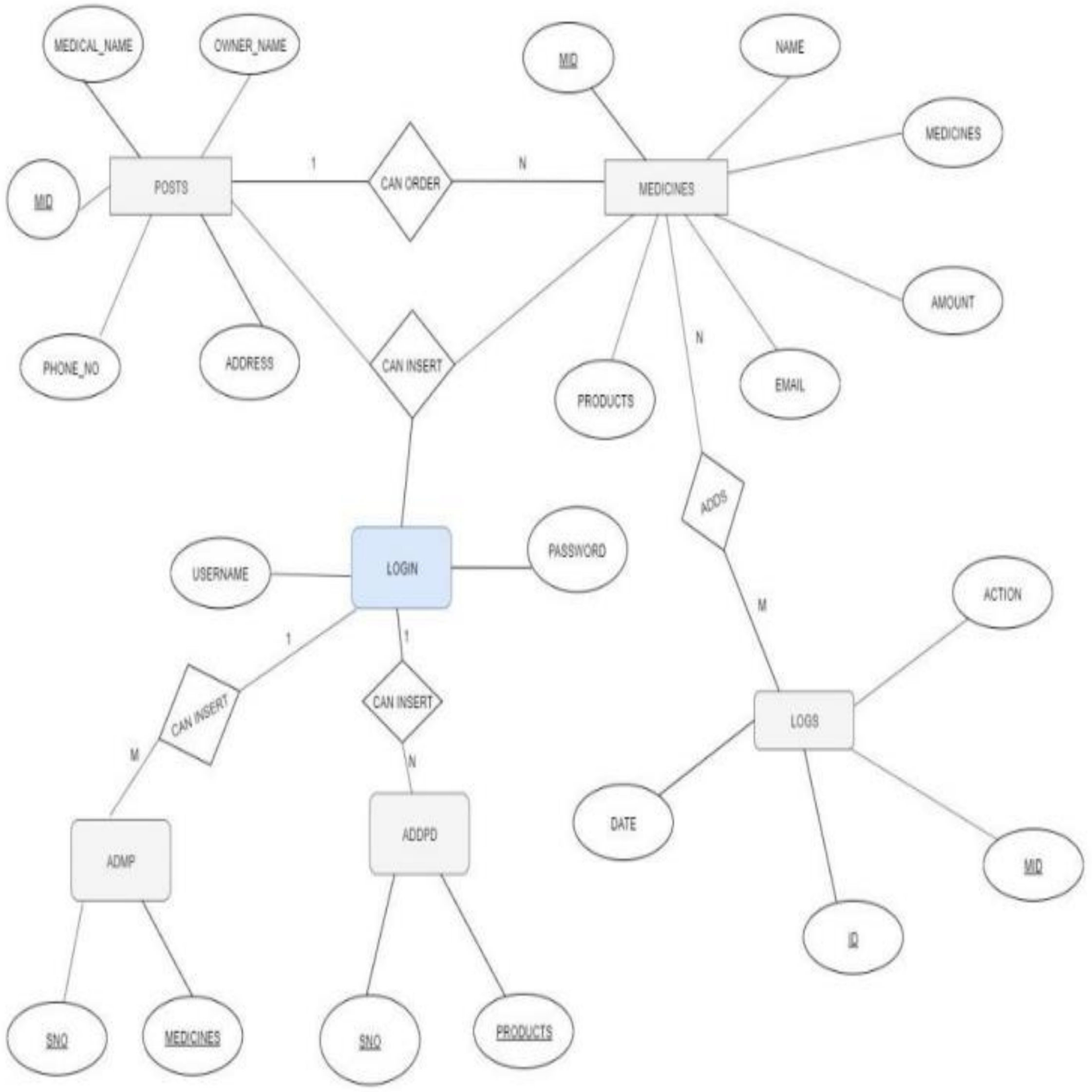
3.2 HARDWARE REQUIREMENTS:

• Computer with a 1.1 GHz or faster processor
• Minimum 2GB of RAM or more
• 2.5 GB of available hard-disk space
• 5400 RPM hard drive
• 1366 × 768 or higher-resolution display
• DVD-ROM drive

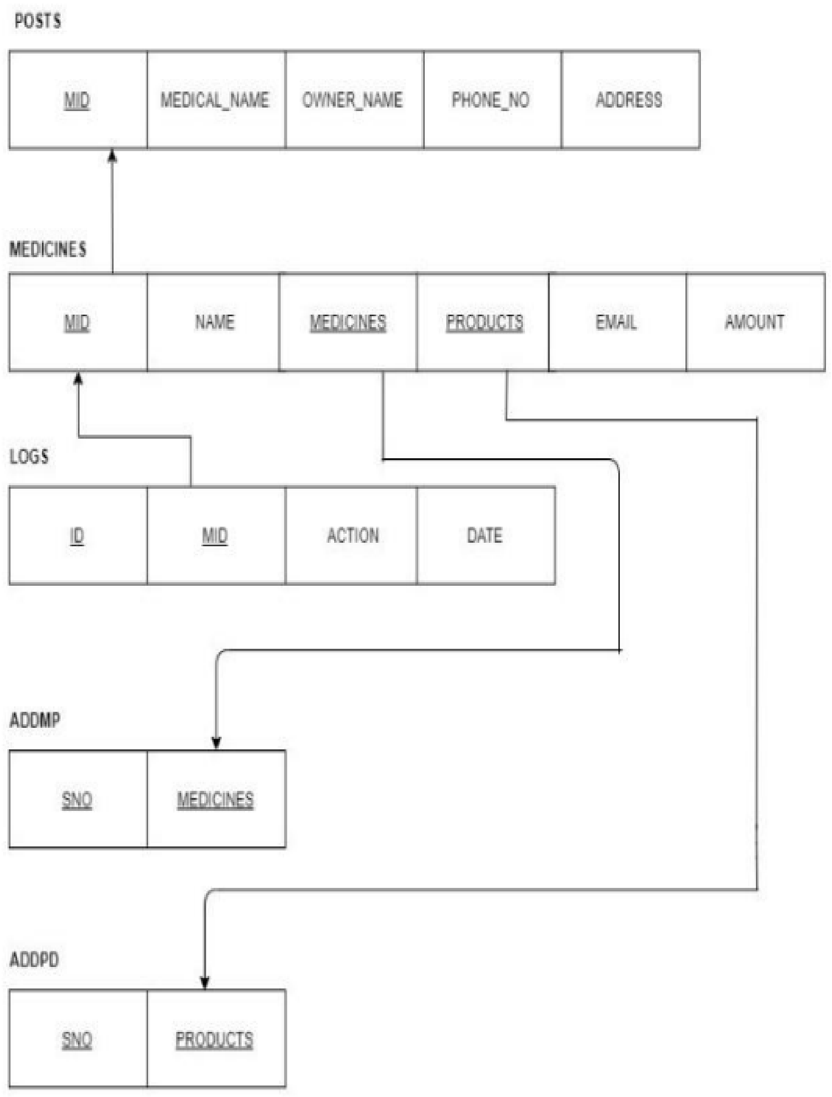
CHAPTER 4

SYSTEM DESIGN

4.1: E-R DIAGRAM:



4.2 SCHEMA DIAGRAM:



CHAPTER 5:

5.1 IMPLEMENTATION:

5.1.1 Python:

An "implementation" of Python should be taken to mean a program or environment which provides support for the execution of programs written in the Python language, as represented by the CPython reference implementation.
There have been and are several distinct software packages providing of what we all recognize as Python, although some of those are more like distributions or variants of some existing implementation than a completely new implementation of the language.

5.1.2 Back End (MySQL) Database:

A Database Management System (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems. Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. A DBMS includes:

➢ A modeling language to define the schema of each database hosted in the DBMS, according to the DBMS data model.

➢ The dominant model in use today is the ad hoc one embedded in SQL, despite the objections of purists who believe this model is a corruption of the relational model, since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.

➢ Data structures (fields, records, files and objects) optimized to deal with very large amounts of data stored on a permanent data storage device (which implies relatively slow access compared to volatile main memory).A database query language and report writer to allow users to interactively interrogate the database, analyze its data and update it according to the users privileges on data.

➢ Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called sub schemas. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and medical data.

➢ If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it may not leave an audit trail of actions or provide the kinds of controls necessary in a multi-user organization. These controls are only available when a set of application programs are customized for each data entry and updating function.

➢A transaction mechanism, that ideally would guarantee the ACID properties, in order to ensure data integrity, despite concurrent user accesses (concurrency control), and faults (fault tolerance).

- It also maintains the integrity of the data in the database.
- The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can help prevent duplicate records via unique index constraints; for example, no two customers with the same customer numbers (key fields) can be entered into the database. See ACID properties for more information (Redundancy avoidance)

When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. to the Organizations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and analysis. Overall systems design decisions are performed by data administrators and systems analysts. Detailed database design is performed by database administrators.

5.1.3 SQL:

Structured Query Language (SQL) is the language used to manipulate relational databases. SQL is tied very closely with the relational model. Structured Query Language (SQL) is the language used to manipulate relational databases.

SQL is tied very closely with the relational model.

- In the relational model, data is stored in structures called relations or tables.
- MySQL is a database server, ideal for both small and large application
- Data definition: Defining tables and structures in the database (DDL used to create, alter and drop schema objects such as tables and indexes).

5.1.4 Stored Procedure:

Routine name: proc 1, post proc

Type: procedure

Definition: Select * from posts;

Select * from medicines;

5.1.5 Triggers:

It is the special kind of stored procedure that automatically executes when an event occurs in the database. Triggers used:

1: Trigger name: on insert

Table: medicines

Time: after

Event: insert

Definition: INSERT INTO logs VALUES (null, new.mid, 'inserted', NOW ());

2: Trigger name: on delete

Table: medicines

Time: after

Event: delete

Definition: INSERT INTO logs VALUES (null, old.mid, 'deleted', NOW ());

5.2 SOURCE CODE:

5.2.1 BACKEND PYHTON WITH MYSQL CODE

```
from flask import Flask, render_template, request, session, redirect, flash
from flask_sqlalchemy import SQLAlchemy
from flask_mail import Mail
import json
with open('config.json','r') as c:
params = json.load(c)["params"]
local_server = True
app = Flask(__name__)
app.secret_key = 'super-secret-key'
app.config.update(
MAIL_SERVER='smtp.gmail.com',
MAIL_PORT='465',
MAIL_USE_SSL=True,
MAIL_USERNAME=params['gmail-user'],
MAIL_PASSWORD=params['gmail-password']
)
mail = Mail(app)
if(local_server):
app.config['SQLALCHEMY_DATABASE_URI'] = params['local_uri']
else:
app.config['SQLALCHEMY_DATABASE_URI'] = params['proud_uri'
db = SQLAlchemy(app)
class Medicines(db.Model):
id = db.Column(db.Integer, primary_key=True)
amount = db.Column(db.Integer, nullable=False)
name = db.Column(db.String(500), nullable=False)
medicines= db.Column(db.String(500), nullable=False)
products = db.Column(db.String(500), nullable=False)
email = db.Column(db.String(120), nullable=False)
mid = db.Column(db.String(120), nullable=False)
```

```
class Posts(db.Model):
mid = db.Column(db.Integer, primary_key=True)
medical_name = db.Column(db.String(80), nullable=False)
owner_name = db.Column(db.String(200), nullable=False)
phone_no = db.Column(db.String(200), nullable=False)
address = db.Column(db.String(120), nullable=False)
class Logs(db.Model):
id = db.Column(db.Integer, primary_key=True)
mid = db.Column(db.String, nullable=True)
action = db.Column(db.String(30), nullable=False
date = db.Column(db.String(100), nullable=False)
@app.route("/index")
def home():
return render_template('dashbord.html', params=params)
@app.route("/insert", methods = ['GET','POST'])
def insert():
if (request.method == 'POST'):
'''ADD ENTRY TO THE DATABASE'''
mid=request.form.get('mid')
medical_name = request.form.get('medical_name')
owner_name = request.form.get('owner_name')
phone_no = request.form.get('phone_no')
address = request.form.get('address')
push = Posts(mid=mid,medical_name=medical_name, owner_name=owner_name,
phone_no=phone_no, address=address)
db.session.add(push)
db.session.commit()
flash("Thanks for submitting your details","danger")
@app.route("/items",methods=['GET','POST'])
def items():
if ('user' in session and session['user'] == params['user']):
posts=Addmp.query.all()
return render_template('items.html', params=params,posts=posts)
@app.route("/logout")
def logout():
```

```
session.pop('user')
flash("You are logout", "primary")
return redirect('/login')
@app.route("/login",methods=['GET','POST'])
def login():
if ('user' in session and session['user'] == params['user']):
posts = Posts.query.all()
return render_template('dashbord.html',params=params,posts=posts)
if request.method=='POST':
username=request.form.get('uname')
userpass=request.form.get('password')
if(username==params['user'] and userpass==params['password']):
session['user']=username
posts=Posts.query.all()
flash("You are Logged in", "primary")
return render_template('index.html',params=params,posts=posts)
else:
flash("wrong password", "danger")
return render_template('login.html', params=params)
# if user is logged in
#delete
@app.route("/delete/<string:mid>", methods=['GET', 'POST'])
def delete(mid):
if ('user' in session and session['user']==params['user']):
post=Posts.query.filter_by(mid=mid).first()
db.session.delete(post)
db.session.commit()
flash("Deleted Successfully", "warning")
return redirect('/login')
post=Medicines.query.filter_by(id=id).first()
db.session.delete(post)
db.session.commit()
flash("Deleted Successfully", "primary")
@app.route("/medicines", methods = ['GET','POST'])
def medicine():
```

```
if(request.method=='POST'):
'''ADD ENTRY TO THE DATABASE'''
mid=request.form.get('mid')
name=request.form.get('name')
medicines=request.form.get('medicines')
products=request.form.get('products')
email=request.form.get('email')
amount=request.form.get('amount')
entry=Medicines(mid=mid,name=name,medicines=medicines,products=products,email=emai
l,amount=amount)
db.session.add(entry)
db.session.commit()
mail.send_message('new message send from ' + name, sender=email,
recipients=[params['gmail-user']],
body= 'Medicines ordered--->' +medicines + "\n" 'products ordered---> ' + products + "\n"
'customer id --->' + email + "\n"'total amount=' +amount)
flash("Data sent and Added Successfully","primary"
return render_template('medicine.html',params=params)
app.run(debug=True)
```

5.2.2 FRONT END CODE

```
<!DOCTYPE html>
<html lang="en">
<head
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<meta name="description" content="">
<meta name="author" content="">

<title>{{params['blog_name']}}</title>

<!-- Bootstrap core CSS -->
<link                    href="{{url_for('static',filename='vendor/bootstrap/css/bootstrap.min.css')}}"
rel="stylesheet">
```

```
<!-- Custom fonts for this template -->
<link                  href="{{url_for('static',filename='vendor/fontawesome-free/css/all.min.css')}}"
rel="stylesheet" type="text/css">
<link                  href='https://fonts.googleapis.com/css?family=Lora:400,700,400italic,700italic'
rel='stylesheet' type='text/css'>
<link
href='https://fonts.googleapis.com/css?family=Open+Sans:300italic,400italic,600italic,700ita
lic,800italic,400,300,600,700,800' rel='stylesheet' type='text/css'>

<!-- Custom styles for this template -->
<link href="{{url_for('static',filename='css/clean-blog.min.css')}}" rel="stylesheet">

</head>
<body>

<!-- Navigation -->
<nav class="navbar navbar-expand-lg navbar-light fixed-top" id="main Nav">
<div class="container">
<a class="navbar-brand" h ref="#">{{params['blog_name']}}</a>
<!--    <button class="navbar-toggler navbar-toggler-right" type="button" datatoggle="collapse"
data-target="#navbar Responsive" aria-controls="navbar Responsive" ariaexpanded="false" aria-
label="Toggle navigation">
Menu
<i class="f as fa-bars"></i>
</button> -->
<div class="collapse navbar-collapse" id="navbar Responsive">
<ul class="navbar-nav ml-auto">
<li class="nav-item">
<a class="nav-link" h ref="/login">home</a>
</li>
<li class="nav-item">
<a class="nav-link" h ref="/insert">add medical information</a>
</li>
<li class="nav-item">
```

```
<a class="nav-link" h ref="/list">view ordered list</a>
</li>
<li class="nav-item">
<a class="nav-link" h ref="/medicines ">ADD medicines/products</a>
</li>
<li class="nav-item">
<a class="nav-link" h ref="/details ">Details</a>
</li>
<li class="nav-item">
<a class="nav-link" h ref="/search ">add/search</a>
</li>

<li class="nav-item">
<a class="nav-link" href="/aboutus">about us</a>
</li>
<!-- <li class="nav-item">
<a class="nav-link" href="/add">search</a>
</li> -->
<li class="nav-item">
<a onclick="return confirm('Are you sure to logout?');" class="nav-link"
href="/logout">logout</a>
</li>

</ul>
</div>
</div>
</nav>

{% block body %}  {% endblock %}
<hr>
<!-- Bootstrap core JavaScript -->
<script src="{{url_for('static',filename='vendor/jquery/jquery.min.js')}}"></script>
<script
src="{{url_for('static',filename='vendor/bootstrap/js/bootstrap.bundle.min.js')}}"></script>
```

```
<!-- Custom scripts for this template -->
<script src="{{url_for('static',filename='js/clean-blog.min.js')}}"></script>

</body>
</html>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
</html>
```

Chapter 6:

Snapshots

6.1: LOGIN PAGE

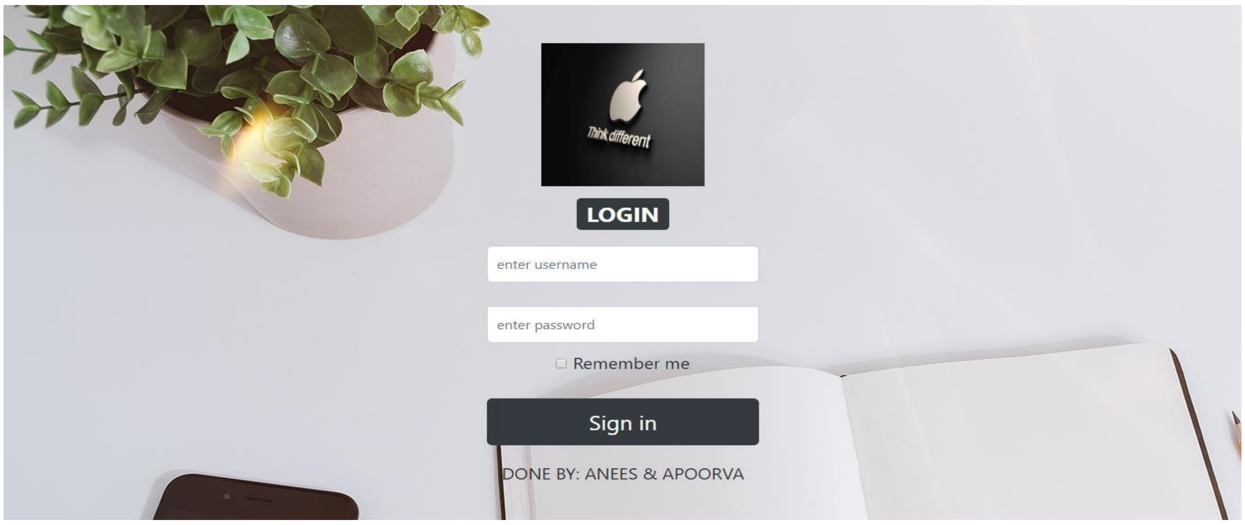
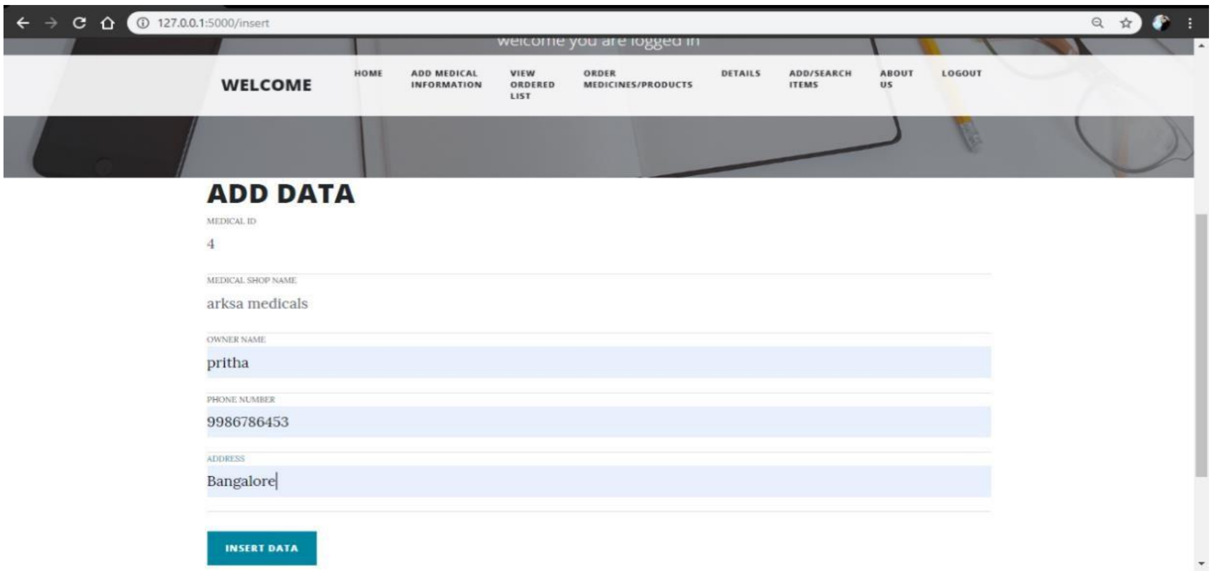


Fig6.1: LOGIN PAGE

6.2: ADD MEDICAL SHOP INFO :



ADD DATA

MEDICAL ID
4

MEDICAL SHOP NAME
arksa medicals

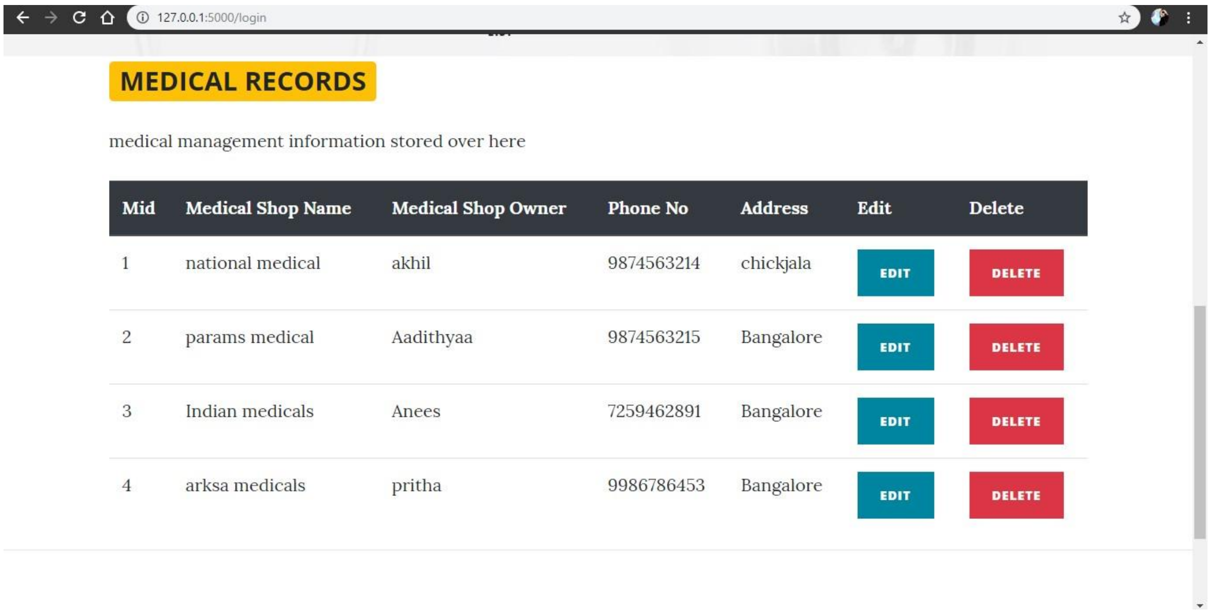
OWNER NAME
pritha

PHONE NUMBER
9986786453

ADDRESS
Bangalore

Fig 6.2: ADD MEDICAL SHOP INFO :

6.3:MEDICAL REPORTS:



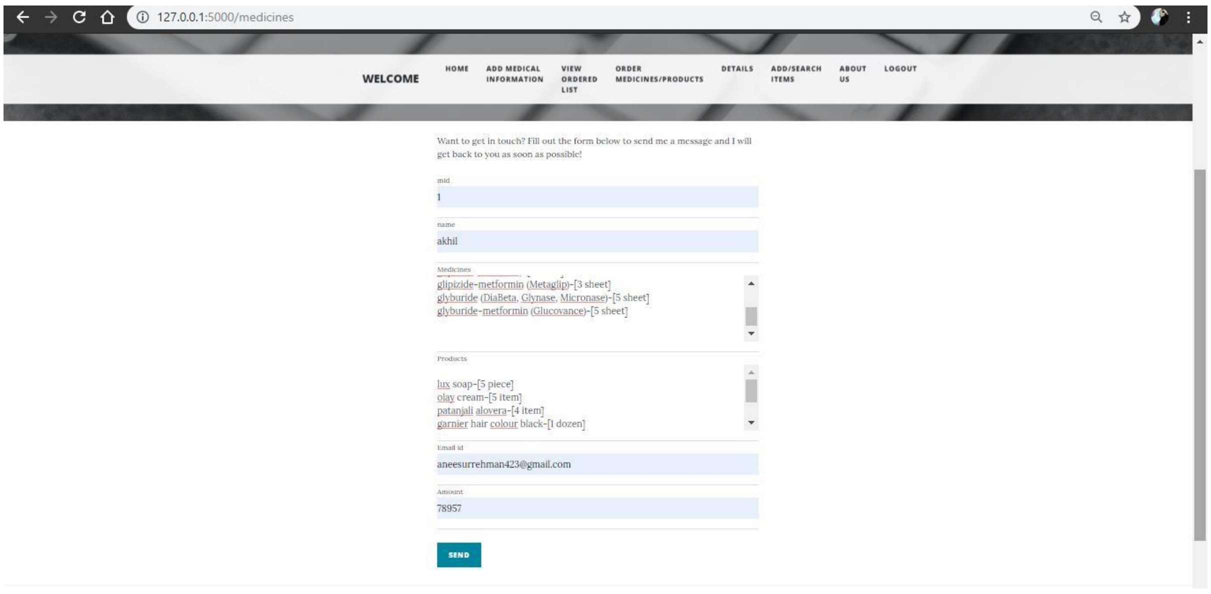
MEDICAL RECORDS

medical management information stored over here

Mid	Medical Shop Name	Medical Shop Owner	Phone No	Address	Edit	Delete
1	national medical	akhil	9874563214	chickjala	EDIT	DELETE
2	params medical	Aadithyaa	9874563215	Bangalore	EDIT	DELETE
3	Indian medicals	Anees	7259462891	Bangalore	EDIT	DELETE
4	arksa medicals	pritha	9986786453	Bangalore	EDIT	DELETE

Fig 6.3:MEDICAL REPORTS:

6.4: COLLECTING ORDERS

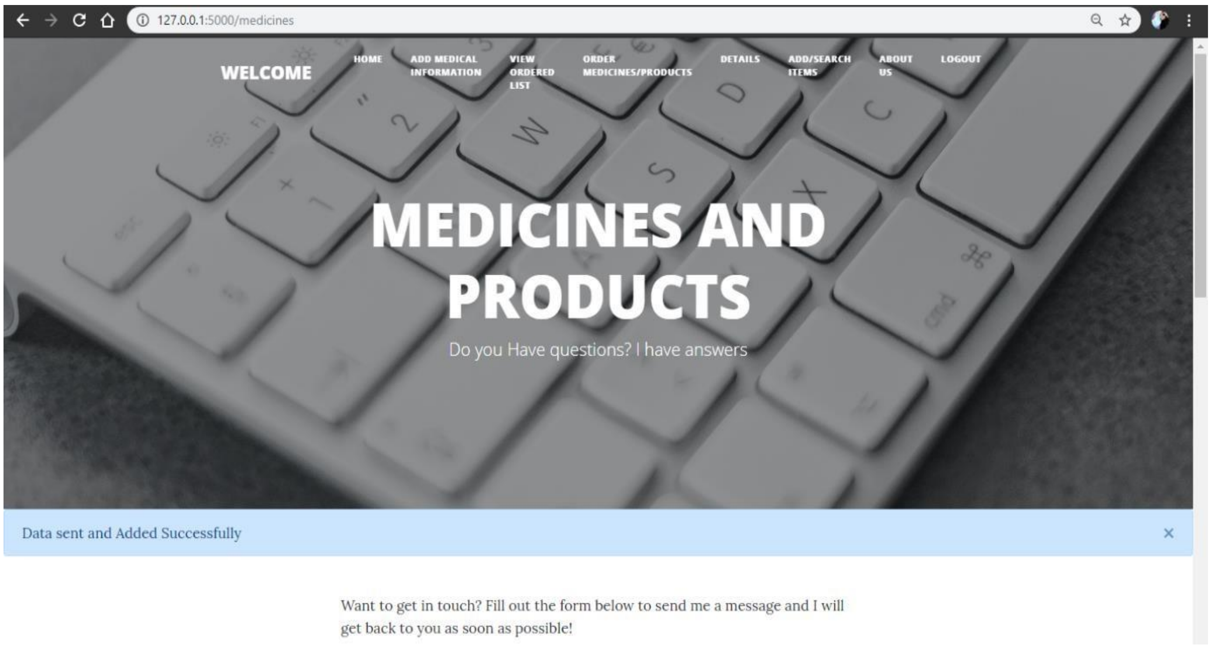


Want to get in touch? Fill out the form below to send me a message and I will get back to you as soon as possible!

mid	1
name	akhil
Medicines	glipizide-metformin (Metaglip)-[3 sheet] glyburide (DiaBeta, Glynase, Micronase)-[5 sheet] glyburide-metformin (Glucovance)-[5 sheet]
Products	lux soap-[5 piece] olay cream-[5 item] patanjali alovera-[4 item] garnier hair colour black-[1 dozen]
Email id	aneesurrehman423@gmail.com
Amount	78957

Fig 6.4: COLLECTING ORDERS :

6.5: ORDERED LIST:

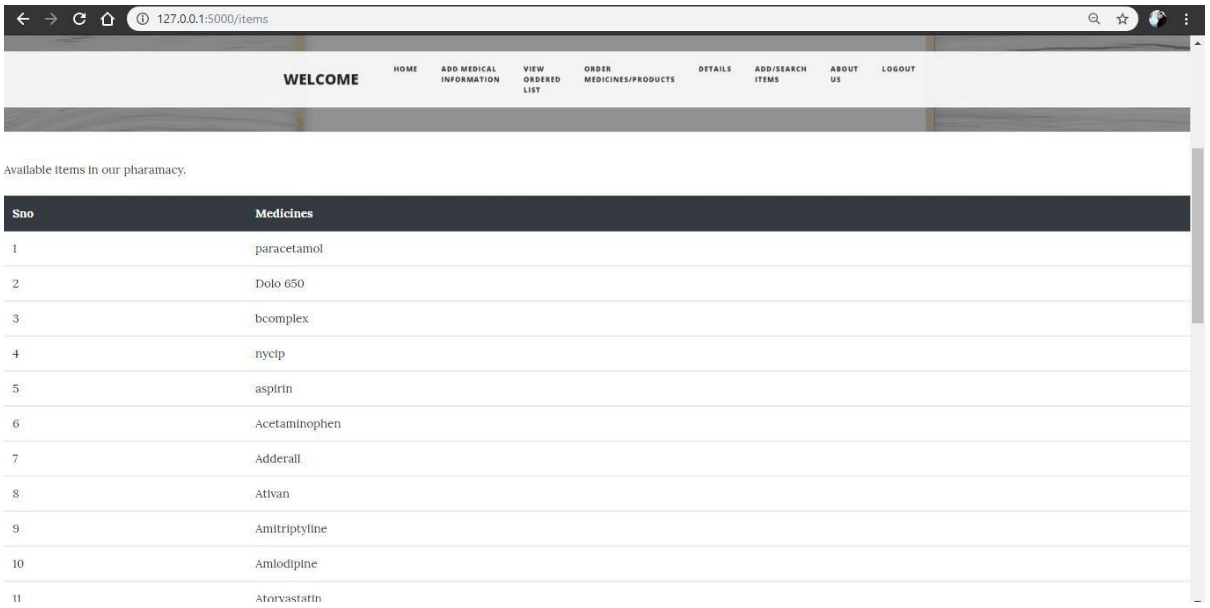


Data sent and Added Successfully

Want to get in touch? Fill out the form below to send me a message and I will get back to you as soon as possible!

Fig 6.5: ORDERED LIST:

6.6: ADDING MEDICINES AND PRODUCTS IN PHARMACY :



Available items in our pharamacy.

Sno	Medicines
1	paracetamol
2	Dolo 650
3	bcomplex
4	nycip
5	aspirin
6	Acetaminophen
7	Adderall
8	Ativan
9	Amitriptyline
10	Amlodipine
11	Atorvastatin

Fig6.6: ADDING MEDICINES AND PRODUCTS IN PHARMACY :

6.7: SEARCH MEDICINS AND PRODUCTS :

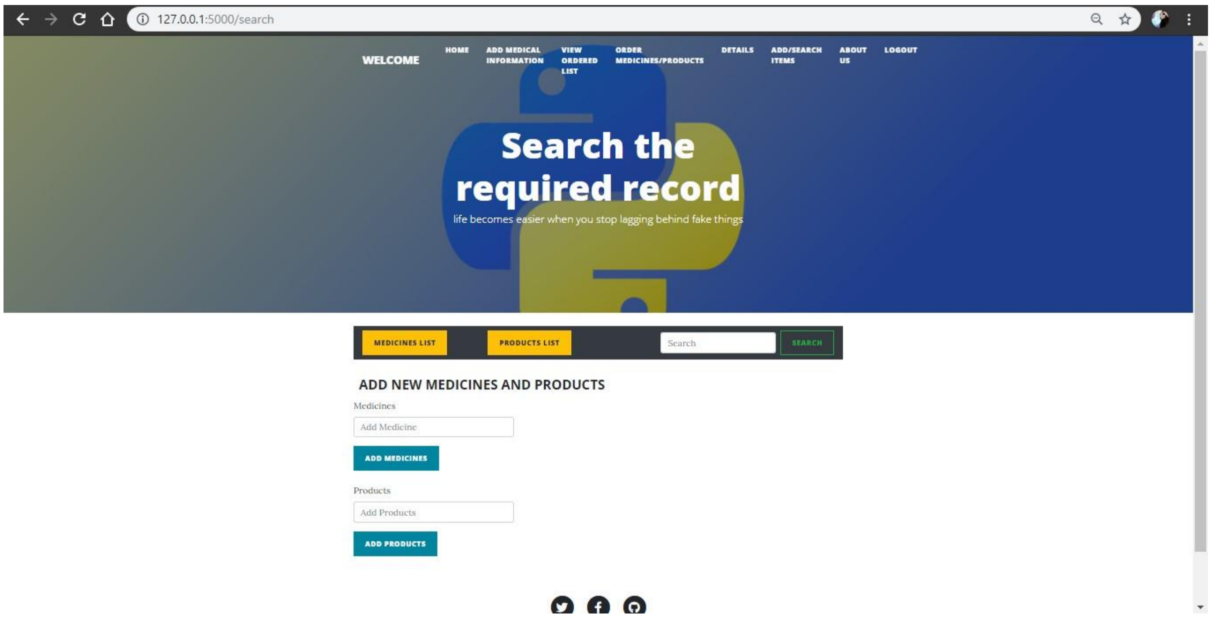
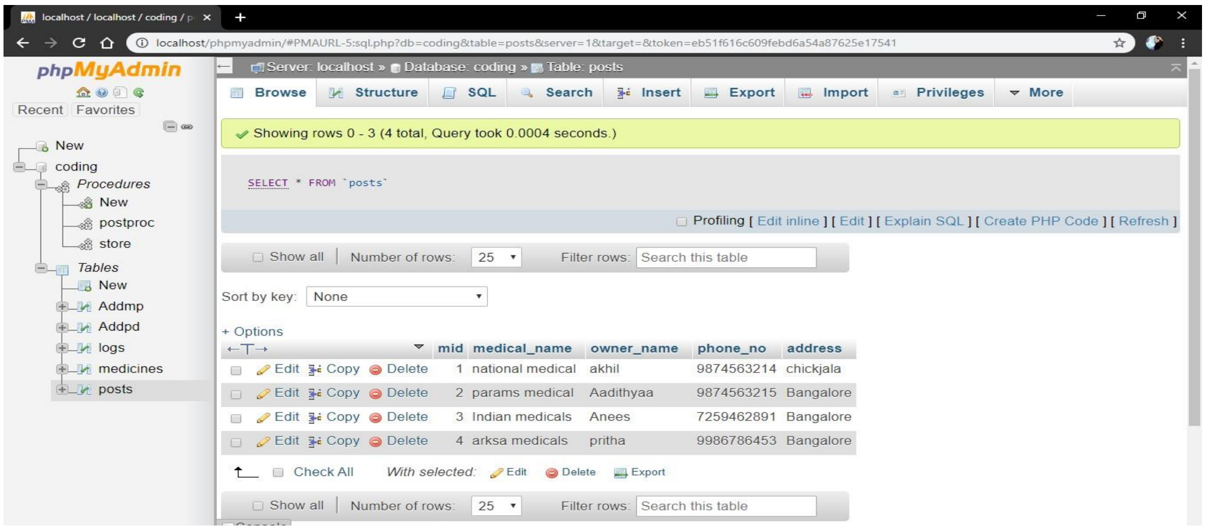


Fig 6.7: SEARCH MEDICINS AND PRODUCTS :

6.10: MEDICAL SHOP INFO(POSTS) :


localhost / localhost / coding / p
localhost/phpmyadmin/#PMAURL-5:sql.php?db=coding&table=posts&server=1&target=&token=eb51f616c609febd6a54a87625e17541
phpMyAdmin
Recent Favorites
New
coding
Procedures
New
postproc
store
Tables
New
Addmp
Addpd
logs
medicines
posts
Server: localhost » Database: coding » Table: posts
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
More

Showing rows 0 - 3 (4 total, Query took 0.0004 seconds.)

```
SELECT * FROM `posts`
```

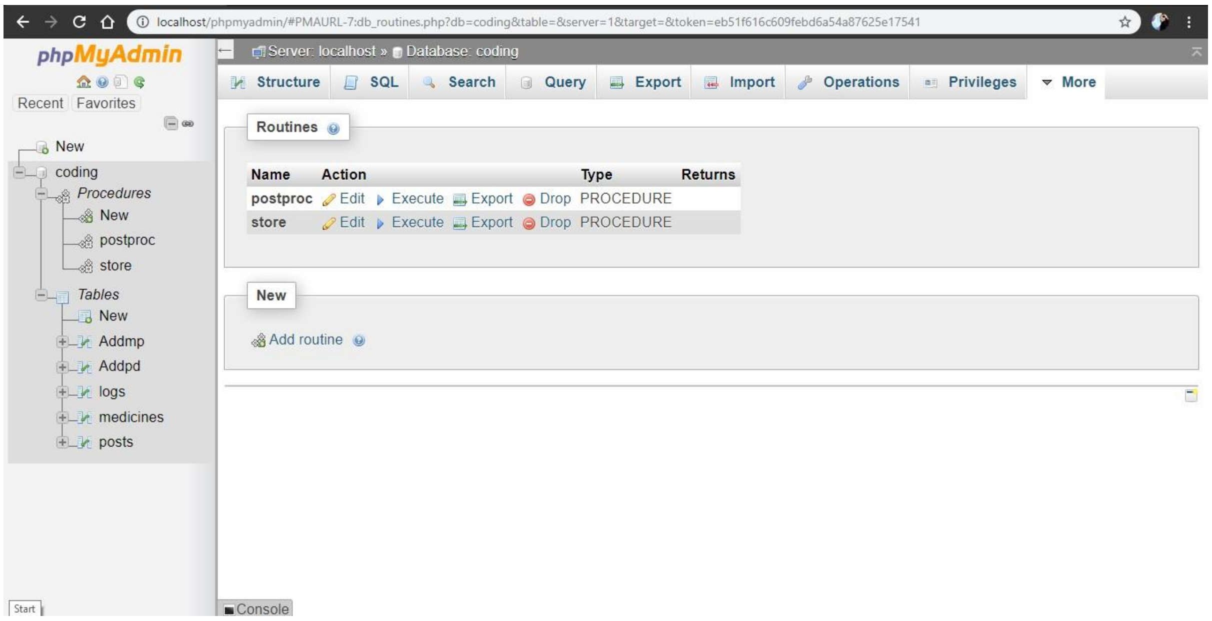
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP Code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options

mid	medical_name	owner_name	phone_no	address
1	national medical	akhil	9874563214	chickjala
2	params medical	Aadithyaa	9874563215	Bangalore
3	Indian medicals	Anees	7259462891	Bangalore
4	arksa medicals	pritha	9986786453	Bangalore

Check All
With selected:
Edit
Delete
Export
Show all
Number of rows: 25
Filter rows: Search this table

Fig 6.10: MEDICAL SHOP INFO(POSTS) :

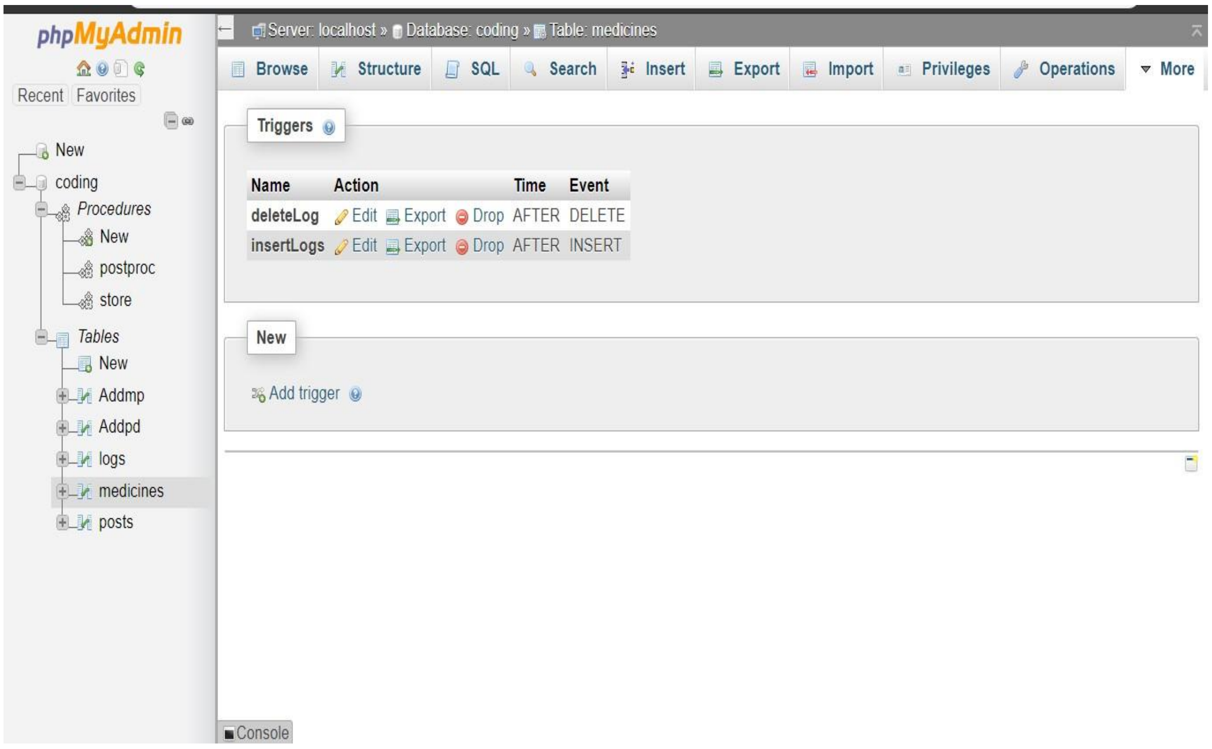
6.11: STORE PROCEDURE


localhost/phpmyadmin/#PMAURL-7:db_routines.php?db=coding&table=&server=1&target=&token=eb51f616c609febd6a54a87625e17541
Server: localhost » Database: coding
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
More
Routines
New
Add routine
Console

Name	Action	Type	Returns
postproc	Edit Execute Export Drop	PROCEDURE	
store	Edit Execute Export Drop	PROCEDURE	

Fig 6.11: STORE PROCEDURE

6.12: TRIGGERS :



Name	Action	Time	Event
deleteLog	Edit Export Drop	AFTER	DELETE
insertLogs	Edit Export Drop	AFTER	INSERT

Fig 6.12: TRIGGERS :

CHAPTER:7

7.1 CONCLUSION

PHARMACY MANAGEMENT SYSTEM successfully implemented offline medicines supply management database which helps us in administrating the data user for managing the tasks performed in medicines supply. The project successfully used various functionalities of Ampps and python flask and also create the fully functional database management system for offline pharmacy.

Using MySQL as the database is highly beneficial as it is free to download, popular and can be easily customized. The data stored in the MySQL database can easily be retrieved and manipulated according to the requirements with basic knowledge of SQL.

With the theoretical inclination of our syllabus, it becomes very essential to take the atmost advantage of any opportunity of gaining practical experience that comes along. The building blocks of this Major Project "Pharmacy Supply Management System" was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development:

• The planning that goes into implementing a project.

• The importance of proper planning and an organized methodology.

• The key element of team spirit and co-ordination in a successful project.

7.2 FUTURE ENANCEMENT

• Enhanced database storage facility

• Enhanced user friendly GUI

• more advanced transportation of medicines

• online Bill payments

7.3 Scope For Developing The Medical Store Management System

The detail of medicine which are available in medical store is easily managed and organized by using this system. The medical store management system also helps in keeping track of the available stock of medicine and also update them on a regular manner. It also helps in analyzing the location of the specific medicine available in the medical store. This project also enables to store the detail of suppliers and the stocks supplied by them.

The medical store management system also provides the facility of generated automatic bills for the customers. The database for the customers and employees can also be managed by using this project.

The designing of medical store management system saves the time which one spent in keeping records and managing payments of the stocks. The process of sailing medicine is very crucial as the medical stores must keep eye on the expiry date of every stock and medicine. This project helps the medical stores to keep track on the expiry date of every medicine they have in the store. After the designing and developing the software of medical store management system, it can be used by the medical stores for conducting, organizing and managing the different activities that are involved in keeping inventory, managing payments and others in the medical stores.

REFERENCES

- https://www.youtube.com/
- https://www.google.com/
- http://www.getbootstrap.com/