**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION)

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA)



**DBMS PROJECT**

**ON**

**“PHARMACEUTICALS WEBSITE”**

***SUBMITTED BY:***  **USN:**

MOHAN SAI KIRAN 1NT16CS056

PAWAN R TANKSALI 1NT16CS070

*Submitted in partial fulfillment of the requirement for the award of Degree of BACHELOR OF COMPUTER SCIENCE AND ENGINEERING*

Department of Computer Science and Engineering

NitteMeenakshi Institute of Technology,

Yelahanka, Bangalore– 560064

Academic Year 2018-2019

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION)

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA)



**CERTIFICATE**

**This is to certify that the DBMS Project Report entitled**

**“PHARMACEUTICALS WEBSITE”**

is an authentic work carried out by,

***SUBMITTED BY:***  **USN:**

MOHAN SAI KIRAN 1NT16CS056

PAWAN R TANKSALI 1NT16CS070

In partial fulfillment of the requirements for the completion of DBMS Project for the academic year 2018-2019.

Name & Signature of the Guide Name & Signature of HOD

**Mrs. Jagdevi N Kalshetty Dr. Thippeswamy**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guide and encouragement crowned our efforts with success.

We would like express our gratitude to our **Principal**, **Dr. H C NAGARAJ**.

We would like to thank our **HOD, Dr. Thippeswamy MN** for his valuable support without whose support this project would not have been possible.

We would also like to express our greatest thanks to our guide, **DBMS Teacher, Mrs. Jagdevi N Kalshetty** for giving us such a concept to work on and for her valuable effort, support and guidance without which this project would not have been possible.

We would like to thank our friends and parents for providing inspiration to successfully complete the project.

We would like to mention special thanks to the staff of **CSE Department, NMIT** for their support.

**ABSTRACT**

Now a day’s Information and communication technology (ICT) plays a great role in different fields and the Health care system belongs to this. This leads to various studies and researches being conducted to selected health care facilities. It is necessary to ensure a technologically appropriate, equitable, affordable and environmentally adaptable and consumer friendly system, designed to fully utilize the ICT for maximum benefit.

Here computers have great relevant on storing data’s securely and ease access on them in short period of time. For this access of data we use Databases either in a local server or a web server. To access all this data we use a webpage that is responsive and is easily accessible.

The Pharmacy website is robust, integrated and easily usable for everyone. Pharmacy website deals with maintenance of drugs in the databases and the properties or the details of the drugs like their batch number, expiry date, manufacture date, dosage, etc. The website will ensure the availability of sufficient quantity of drugs and consumable materials for the patient. This will enhance the efficiency of clinical work and ease patient’s convenience and providing a better care for the patients.

This project is an introduction or just a basic beginning of the Pharmaceuticals website where the Database platform is “MongoDB” and have used Web Tech languages like HTML, PHP.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **Description** | **Page No.** |
|  | INTRODUCTION  1.1 OBJECTIVE | 1 |
|  | SYSTEM REQUIREMENTS | 2 |
|  | IMPLEMENTATION | 3 |
|  | CORRELATING THEORETICAL CONCEPT TO PRACTICAL IMPLEMENTATION | 4 |
|  | FLOW DESIGN/DESIGN OF THE SYSTEM | 7 |
|  | RESULS/SNAPSHOTS | 8 |
|  | CONCLUSION | 17 |
| 8. | REFERENCES | 18 |

# INTRODUCTION

MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemata. MongoDB is developed by MongoDB Inc., and is published under a combination of the Server-Side Public License and the Apache License.

MongoDB supports field, range query, and regular expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions. Queries can also be configured to return a random sample of results of a given size.

As of October 2018, MongoDB is released under the Server-Side Public License, a license developed by the project. It replaces the GNU Affero General Public License, and is nearly identical to the GNU General Public License version 3, but requires that those using the software as part of a service available to third-parties must make the service's source code available under this license. The language drivers are available under an Apache License. In addition, MongoDB Inc. offers proprietary licenses for MongoDB.

## 1.1 OBJECTIVE

The main purpose of this project is to simplify the process of searching and adding medicine by providing a web interface for admin and user.

Listed below are the modules which will be included and further modifications can take place

1. To insert a new medicine with the appropriate details.

2. To update medicine details.

3. To remove medicine details.

4. To show medicine details.

5. To search for a particular medicine.

**SYSTEM REQUIREMENTS**

**SOFTWARE REQUIREMENTS:**

* HTML/CSS
* PHP
* NoSQL Database(MongoDB)
* XAMPP server
* JavaScript
* Notepad++
* Google Chrome or any other Browsers
* Windows or a Ubuntu operating system

**HARDWARE REQUIREMENTS:**

* Processor –Core i3+
* Hard Disk – 160 GB+
* Memory – 1GB RAM+
* Monitor

**IMPLEMENTATION**

Listed below are the modules which will be included and further modifications can take place

**1. Admin Login/Registration:** Admin need to login by providing the login credentials to access the below given admin modules or new admin can register himself and login.

**a. Entry of medicine:**

i. Admin can enter details about new medicine products details.

**b. View Medicines:**

i. Admin can view details about the medicines.

**c. Deletion of a Medicine:**

i. Admin can delete the details of a particular medicine based on the medicine name.

**d. Updating a medicine:**

i. Admin can update the details of a particular medicine.

**e. Search for a medicine:**

i. Admin can also employ search operation to search for a medicine.

**2. User:** Users can view all medicines and search for a particular medicine of interest.

**a. View Medicines:**

i. The medicines details are arranged and can be viewed by any user.

**b. Search for a medicine:**

i. Users can search for a particular medicine based on the medicine name.

**CORRELATING THEORETICAL CONCEPT TO PRACTICAL IMPLEMENTATION**

* **use <db>**:

Switch current database to <db>. The mongo shell variable db is set to the current database.

* **show collections**

Print a list of all collections for current database.

* **show users**

Print a list of users for current database.

* **show databases**

Print a list of all available databases.

**MONGODB CRUD OPERATIONS**

1. **Create Operations**

Create or insert operations add new documents to a collection. If the collection does not currently exist, insert operations will create the collection.

MongoDB provides the following methods to insert documents into a collection:

* db.collection.insertOne()
* db.collection.insertMany()

**SYNTAX:**

**db.collection.insertOne(**

**<document>,**

**{**

**writeConcern: <document>**

**}**

**)**

In MongoDB, insert operations target a single collection. All write operations in MongoDB are atomic on the level of a single document.

Example:

db.inventory.insertOne(

{ item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } }

)

* **\_id Field**

In MongoDB, each document stored in a collection requires a unique [\_id](https://docs.mongodb.com/manual/reference/glossary/#term-id) field that acts as a [primary key](https://docs.mongodb.com/manual/reference/glossary/#term-primary-key). If an inserted document omits the \_id field, the MongoDB driver automatically generates an [ObjectId](https://docs.mongodb.com/manual/reference/bson-types/" \l "objectid) for the \_id field.

**2. Read Operations:**

Read operations retrieves [documents](https://docs.mongodb.com/manual/core/document/#bson-document-format) from a [collection](https://docs.mongodb.com/manual/core/databases-and-collections/#collections) i.e. queries a collection for documents. MongoDB provides the following method to read documents from a collection.

**SYNTAX:**

* [**db.collection.find(query,projection)**](https://docs.mongodb.com/manual/reference/method/db.collection.find/#db.collection.find)

You can specify [query filters or criteria](https://docs.mongodb.com/manual/tutorial/query-documents/#read-operations-query-argument) that identify the documents to return.

EXAMPLE:

The following example selects from the inventory collection all documents where the status equals "D":

db.inventory.find( { status: "D" } )

**3. Update Operations:**

Update operations modify existing [documents](https://docs.mongodb.com/manual/core/document/#bson-document-format) in a [collection](https://docs.mongodb.com/manual/core/databases-and-collections/#collections). MongoDB provides the following methods to update documents of a collection:

* [db.collection.updateOne()](https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/#db.collection.updateOne)
* [db.collection.updateMany()](https://docs.mongodb.com/manual/reference/method/db.collection.updateMany/#db.collection.updateMany)
* [db.collection.replaceOne()](https://docs.mongodb.com/manual/reference/method/db.collection.replaceOne/#db.collection.replaceOne)

**SYNTAX:**

**db.collection.updateOne(**

**<filter>,**

**<update>,**

**{**

**upsert: <boolean>,**

**writeConcern: <document>,**

**collation: <document>,**

**arrayFilters: [ <filterdocument1>, ... ]**

**}**

**)**

In MongoDB, update operations target a single collection. All write operations in MongoDB are [atomic](https://docs.mongodb.com/manual/core/write-operations-atomicity/) on the level of a single document.

You can specify criteria, or filters, that identify the documents to update. These [filters](https://docs.mongodb.com/manual/core/document/#document-query-filter) use the same syntax as read operations.

EXAMPLE:

The following example updates the collection inventory to update the first document where item is paper

db.inventory.updateOne(

{ item: "paper" },

{

$set: { "size.uom": "cm", status: "P" },

$currentDate: { lastModified: true }

}

)

**4. Delete Operations:**

Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection:

* [db.collection.deleteOne()](https://docs.mongodb.com/manual/reference/method/db.collection.deleteOne/#db.collection.deleteOne)
* [db.collection.deleteMany()](https://docs.mongodb.com/manual/reference/method/db.collection.deleteMany/#db.collection.deleteMany)

In MongoDB, delete operations target a single [collection](https://docs.mongodb.com/manual/reference/glossary/#term-collection). All write operations in MongoDB are [atomic](https://docs.mongodb.com/manual/core/write-operations-atomicity/) on the level of a single document.

**SYNTAX:**

**db.collection.deleteOne(**

**<filter>,**

**{**

**writeConcern: <document>,**

**collation: <document>**

**}**

**)**

**EXAMPLE:**

The following example deletes the first document where status is "D":

db.inventory.deleteOne( { status: "D" } )

**FLOWCHART**

LOGIN PAGE

USER

ADMIN

View Medicines

Search for a medicine

Medicine Entry

View Medicines

Delete Medicine

Update Medicine

Search a Medicine

**RESULTS AND SNAPSHOTS**

MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time.

“Pharmaceutical website” mainly focusses on inserting, deleting, updating and displaying medicine details.

This project front end is developed using HTML, PHP, CSS and backend is developed using “MongoDB”.

**LOGIN PAGE:**

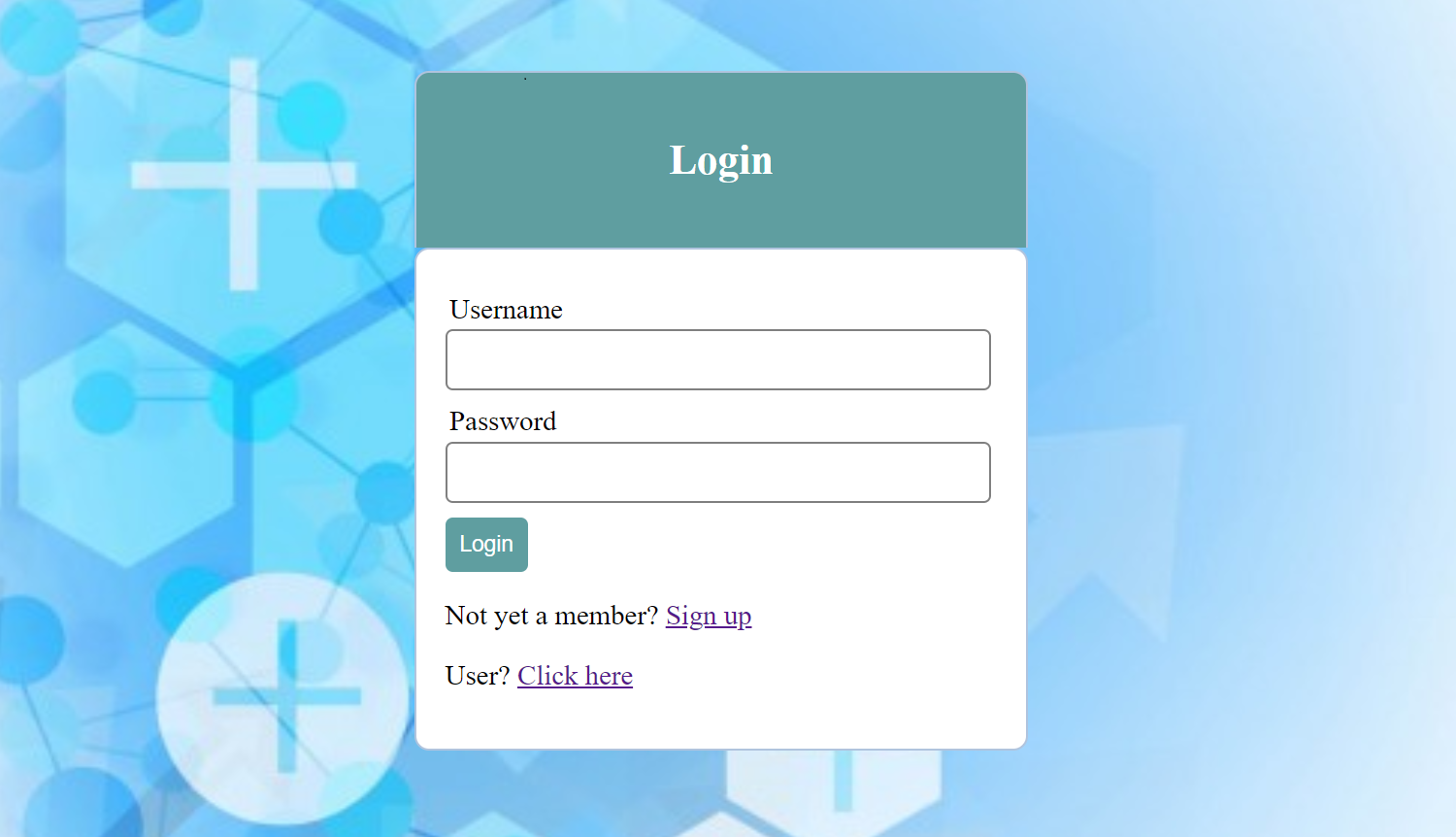
****

Fig 1: Login page

Upon entering the website login page appears with username and password fields to enter. If the person is an administrator then he/she can provide the login credentials and if the person is not registered he/she can register. It also consists of a link for normal user operations

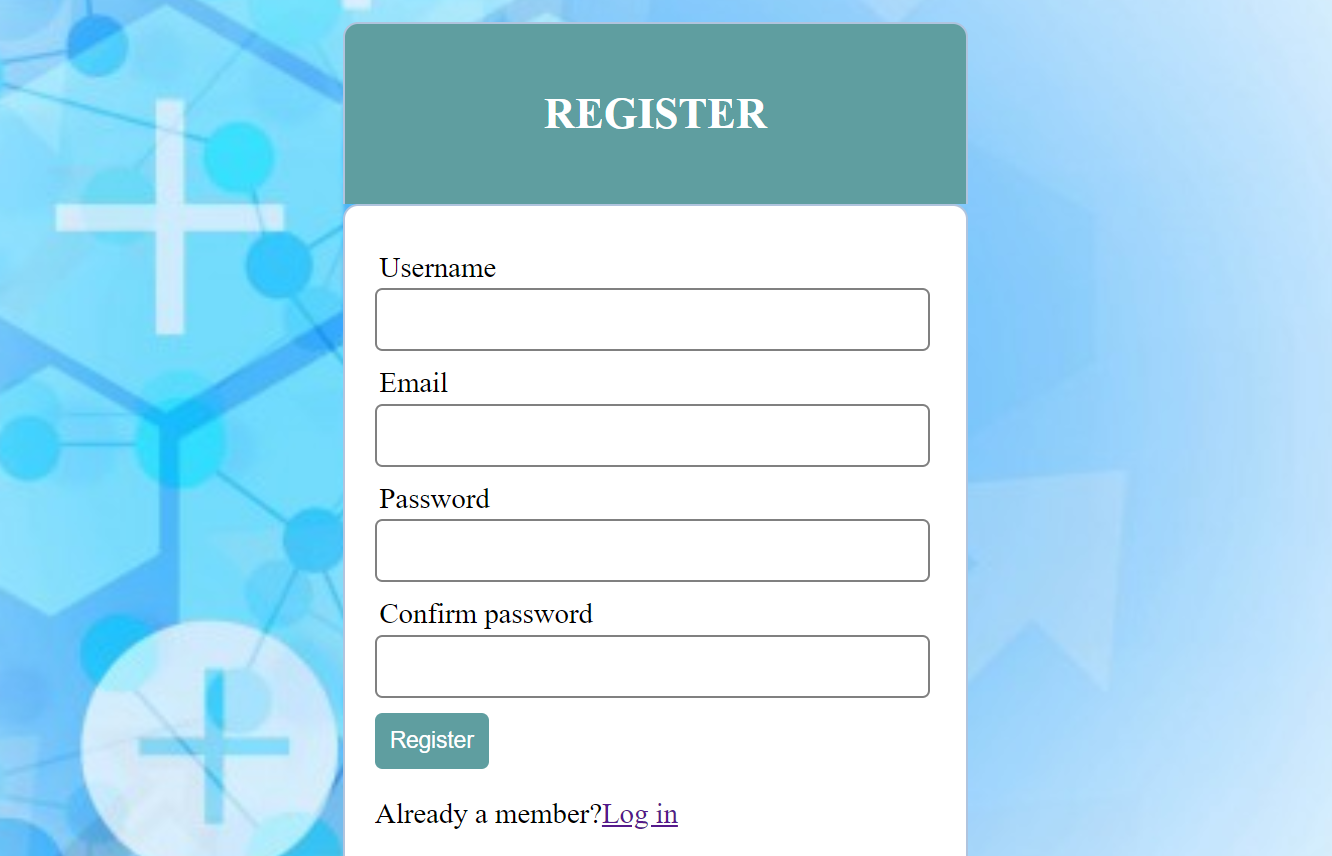
****

Fig 2: Registration

If the person is not registered as an admin then he/she will be provided with a registration form which prompts to enter all the valid details. If the person is already a member and already registered then he/she may proceed with the login.

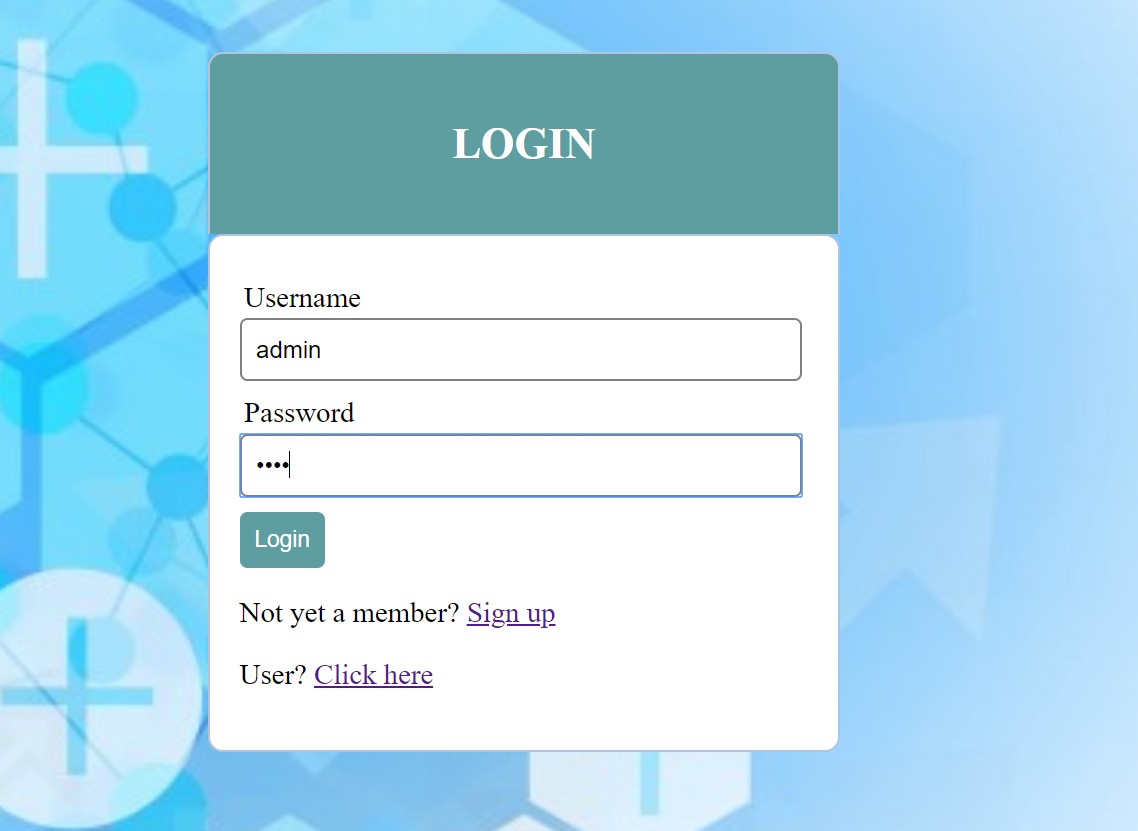


Fig 3: Registered admin login

The administrator who is already registered and a member will be provided with a login page wherein they have to enter the username and password.

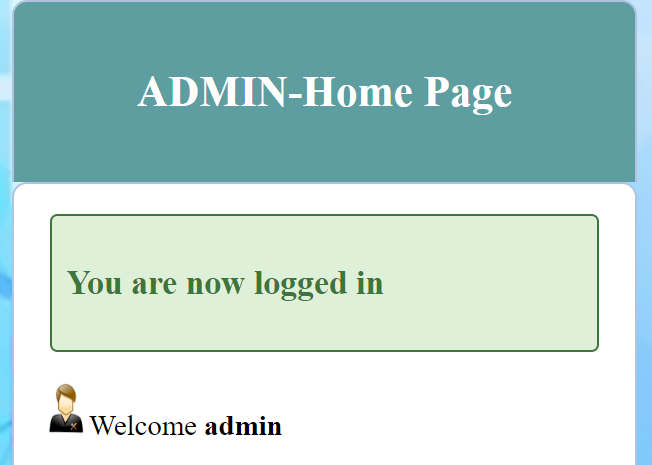


Fig 4: Admin home page

On providing correct credentials the admin will be taken to the home page where admin can perform various operations.

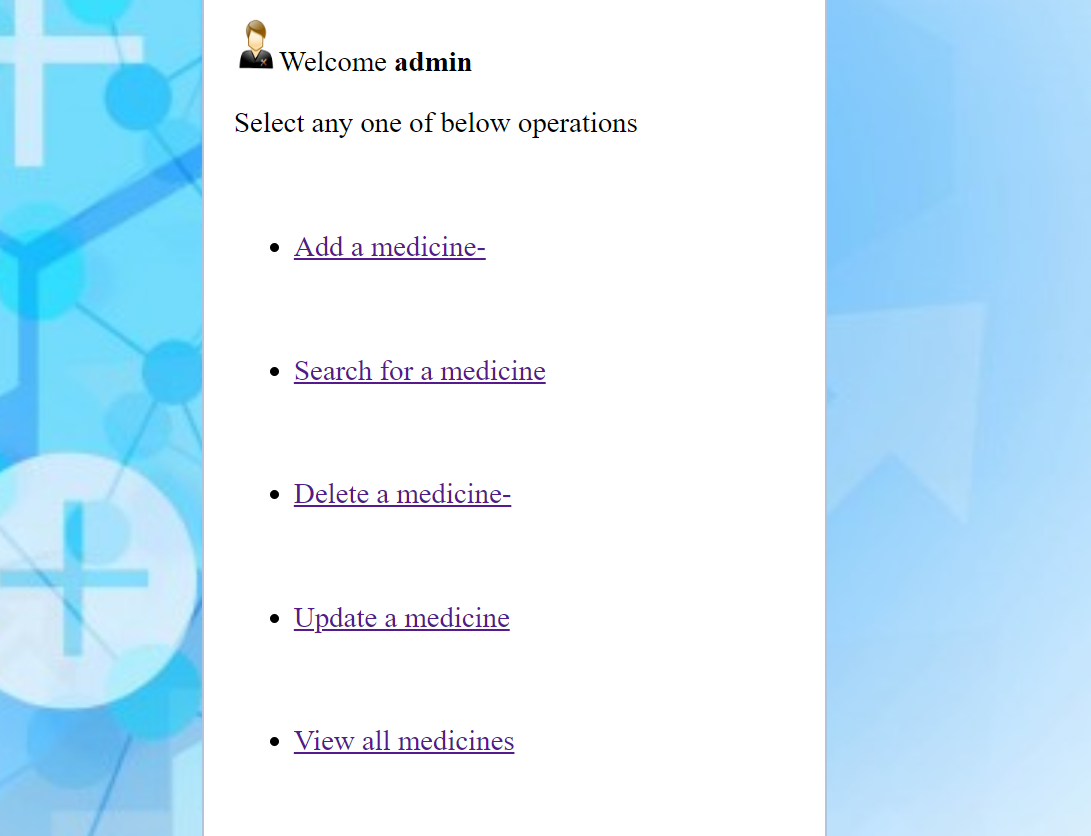


Fig 5: CRUD operations

Figure shows the various operations that the administrator can perform such as adding a medicine,deleting a medicine,updating a medicine,search and to view all the medicines.

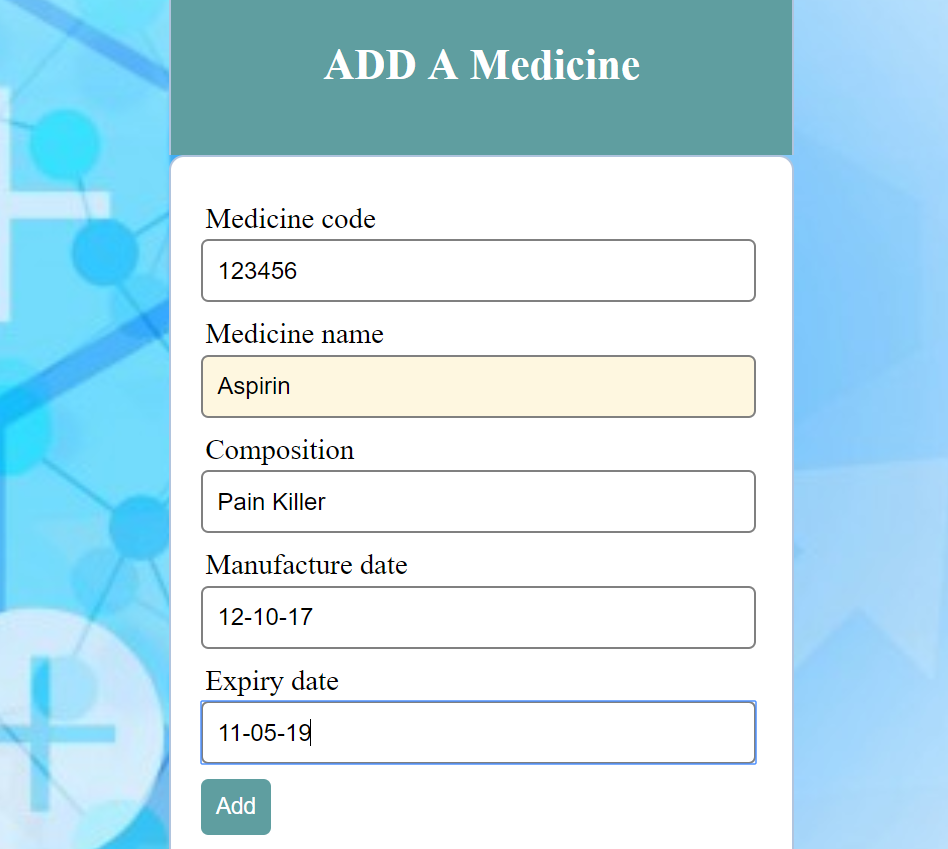


Fig 6: Add a medicine

Figure shows insertion of a medicine details like medicine code,medicine name,composition,manufacture date and expiry date.



Fig 7: Delete a medicine

Figure shows the deletion of a medicine using the medicine code

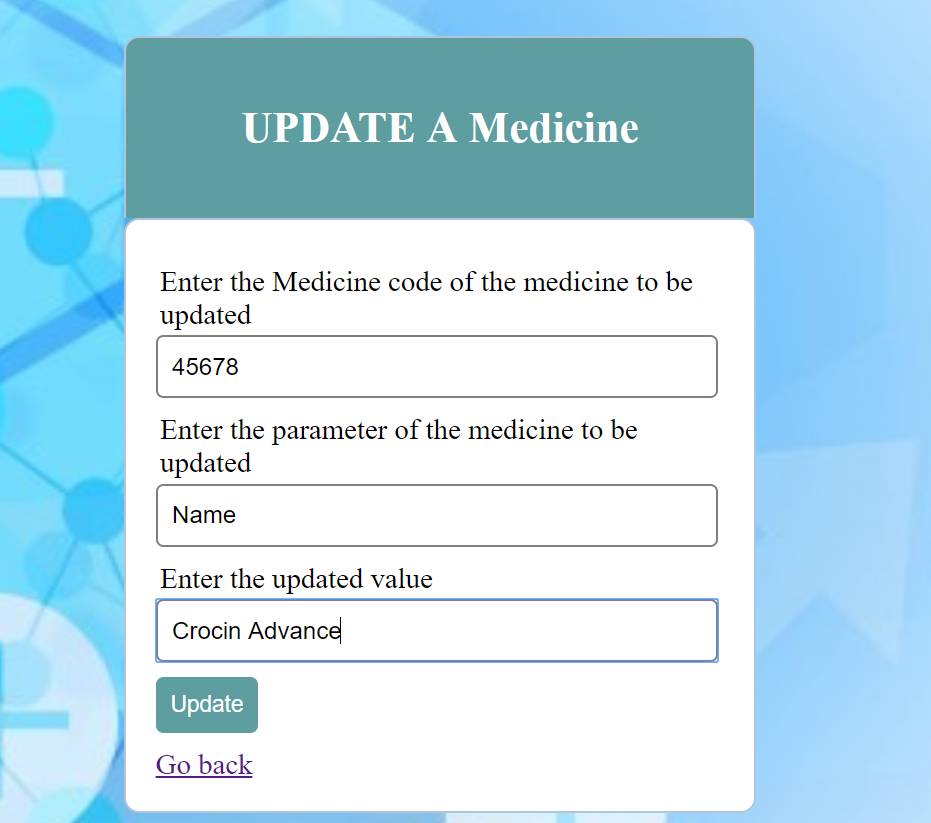


Fig 8: Updating a medicine

Figure shows the updation of a medicine using medicine code we enter the parameter to be changed and the updated value.

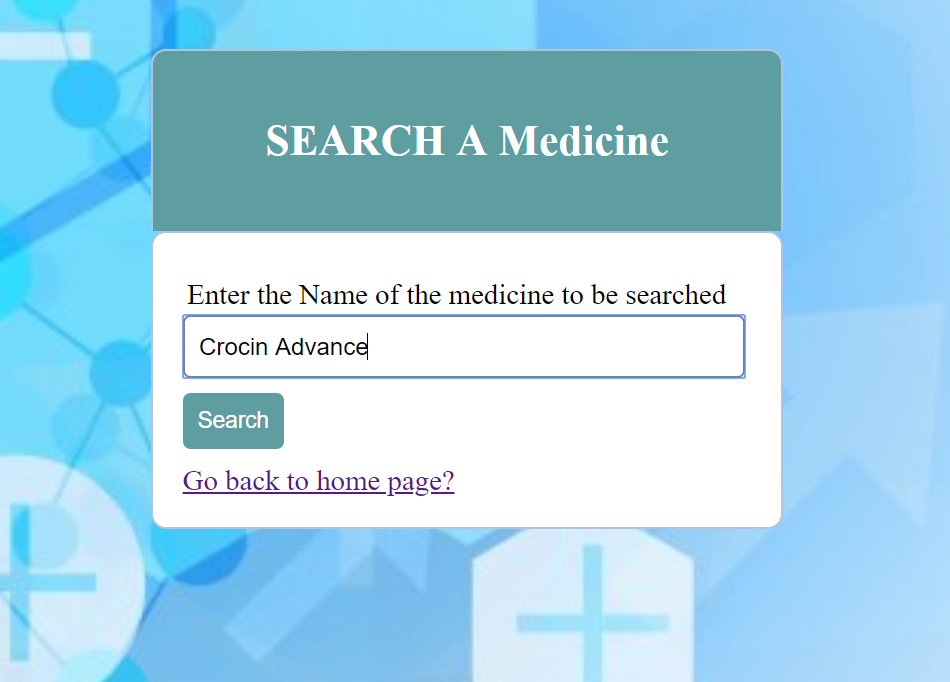


Fig 9: Search for a medicine

Figure shows the searching for a medcine using the name of the medicine.

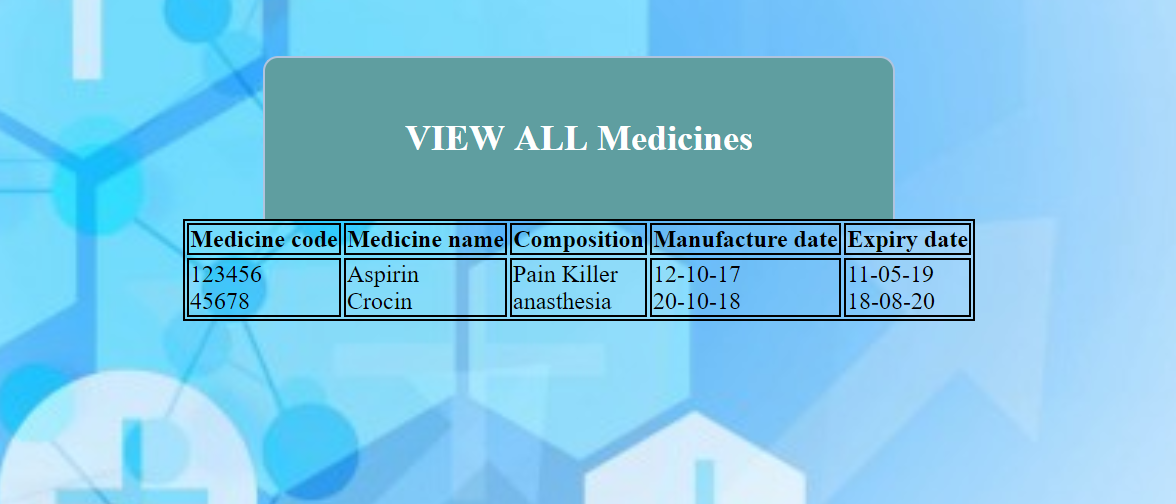


Fig 10: Display of all the medicines

Figure shows the entries of all the medicines along with their details

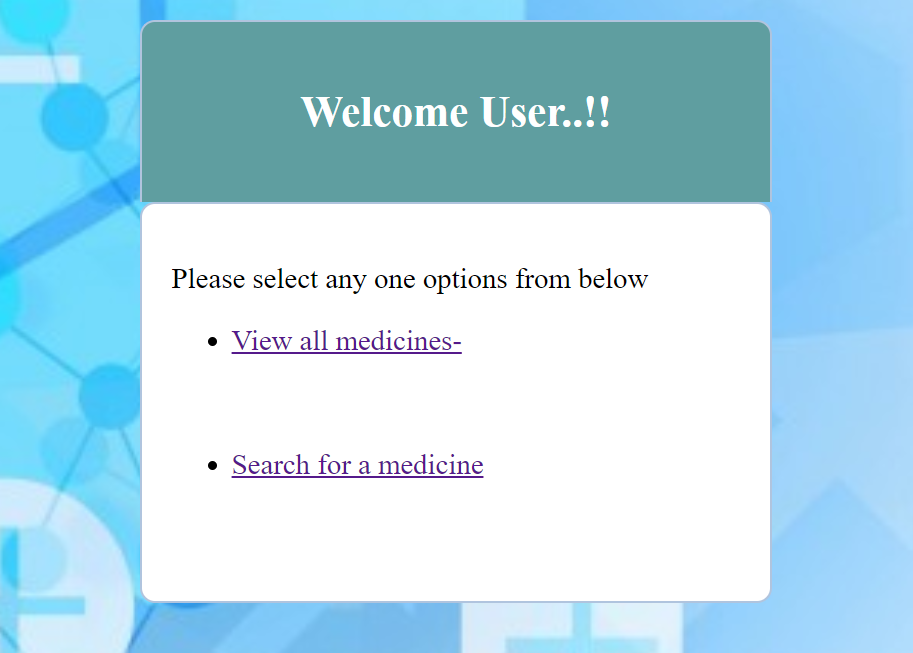
****

Fig 11: User page

Figure shows the page for the users who wishes to view all the medicines as well as search for a medicine.

# CONCLUSION

From this project, we have learnt the importance of the Web Tech languages and Database platforms. There are numerous benefits from learning this, however, the most important benefit is that being in a century where people are heading towards ‘Digital World’, this project helps people in simplifying their lives.

We have also, through this project understood the working of the above-mentioned concepts to a thorough extent. We have learnt different tip and tricks in coding and using the multiple languages as a powerful tool in making the understanding of Web Tech languages easier.

The project will be helpful for pharmacist to automate the system instead of maintaining manual records and is also very user friendly to run various queries.

# 

# REFERENCES

1. <https://www.w3schools.com>

* HTML
* PHP
* CSS
* JS

1. <https://docs.mongodb.com>
2. <https://stackoverflow.com/>