# INTRODUCTION

The Project on Medical Shop Management System is to manage the details of Sells, Medicines, Stocks, Company , billing. It manages all the information about Sells, Medical Shop, Inventory, Sells. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Sells, Medicines, Medical Shop, and Stocks. It tracks all the details about the Stocks, Company, and Inventory.

The main objective of the application is to automate the existing system of manually maintained records of the counter sales, purchases, reorder levels, Supplier and Customer monetary positions and other related transactions made by the seller.

This application can be used by any other store to automate the process of manually maintaining the records related to the subject of maintaining the stock and liquid flows. The main goal of this project is that managing the medicine details, stock details and billing details very clearly. The project is aimed to develop by JAVA as Front end and SQL as Back end

## 1.1. Organization Profile

TestPress helps educational academies create web based online exams that lets their students take exams from anywhere. Testpress partners with different entrance exam coaching institutes to help students prepare for exams effectively. Some of the core features include online exam management, course management, analytics and custom domain mapping.

**Website**

[**http://testpress.in**](https://www.linkedin.com/redir/redirect?url=http%3A%2F%2Ftestpress%2Ein&urlhash=xYA6&trk=about_website)

**Industries**

IT Services and IT Consulting

**Company size**

11-50 employees

**Headquarters**

Chennai

**Type**

Partnership

**Founded**

2014

**Specialties**

Online exam software, course management, exam management, and Learning Management System

## System Specifications

### HARDWARE CONFIGURATION

**Processor** : Pentium -IV

**Speed** : 1 GHz

**Hard Disk Capacity** : 40GB

**RAM Capacity** : 1GB RAM

**CD-ROM Drive** : 52x speed

**Keyboard** : 104 keys

**Mouse** : Logitech

**Printer** : HP3745 series DeskJet printer

### SOFTWARE SPECIFICATION

**Operating System** : Windows 7/8/10

**Front End** : JAVA

**Back End** : SQL

**Feasibility Study**

# SYSTEM STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility of a proposed solution is evaluated in teams of its components. These components are:

* + - * Economic feasibility
      * Technical feasibility

## Economic Feasibility

The economic feasibility study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development or the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

## Technical Feasibility

The technical feasibility study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The will lead to high demands on the available technical resources. This will lead to high demands being places on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

## EXISTING SYSTEM

Currently, the medical works are based on the manual process, and each work is maintained in the paper. The details of purchasing drugs, audits, sell reports maintained on the paper while anyone can enter into the system and can make changes in these reports, so it is not a safe method to keep the information on the paper. The pharmacist faces problem in searching the products from the self as it is not an easy method to remember about the place of each medicine. There no Pharmacy Management System which can alert the pharmacist about the end of the drugs.

### DRAWBACKS

* Can’t able to handle stock details
* More number of resources can be handling single medical shop
* It’s not user friendly for the customer
* Customer need to wait long time for getting billing report

## PROPOSED SYSTEM

The design of the medical shop management system is based on the computer which will simplify the maintenance of the information, accessible and efficient. The Pharmacy Management System will provide the information about the end of the drugs in the medical so that the physician can order them drugs before the end. The pharmacist and nurses will get more accurate results at the time sell, about the details of the use of medicines and the dosages so that the system will become more reliable to use than the present system. The records of each work will be secure as to access the information the user must have to provide the ID and password in the system.

### FEATURES

* + - * Systematically provide all the details up-to-date.
      * Easily find the daily billing details
      * We can collect the all-customer details in single application.

# SYSTEM DESIGN AND DEVELOPMENT

Design is concerned with identifying software components specifying relationship Among components. Specifying software structure and providing blue print for the document phase. Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner, the interaction between parts is Minimal clearly specified. Design will explain software components in details. This will help the implementation of the system. Moreover, this will guide the further changes in the system to satisfy the further requirements.

The design document describes how to transform, the requirement and the functional design into more technical system design specification. This design involves conceiving and planning out in the mind and making a drawing pattern of sketch of. It includes type of activities, External Design, Architectural Design and Detailed Design. The architectural design and detailed design collectively referred to as internal design.

The external design involves specifying the externally observable characteristics of a software product and the internal design involves specifying the internal structure and processing details of the system. The fundamental concept of the design includes abstraction structure, information hiding Modularity, concurrency, verification and design aesthetics.

## FILE DESIGN

In computing, a file design (or file system) is used to control how data is stored and retrieved. Without a file system, information placed in a storage area would be one large body of data with no way to tell where one piece of information stops and the next begins. By separating the data into individual pieces, and giving each piece a name, the information is easily separated and identified. Taking its name from the way paper-based information systems are named, each group of data is called a "file". The structure and logic rules used to manage the groups of information and their names are called a "file system".

Some file systems are used on local data storage devices; others provide file access via a network protocol. Some file systems are "virtual", in that the "files" supplied are computed on request or are merely a mapping into a different file system used as a backing store. The file system manages access to both the content of files and the metadata about those files. It is responsible for arranging storage space; reliability, efficiency, and tuning with regard to the physical storage medium are important design considerations.

## INPUT DESIGN

The input design is the process of entering data to the system. The input design goal is to enter to the computer as accurate as possible. Here inputs are designed effectively so that errors made by the operations are minimized.

The inputs to the system have been designed in such a way that manual forms and the inputs are coordinated where the data elements are common to the source document and to the input. The input is acceptable and understandable by the users who are using it.

Input design is the process of converting user-originated inputs to a computer-based format input data are collected and organized into group of similar data. Once identified, appropriate input media are selected for processing.

The input design also determines the user to interact efficiently with the system. Input design is a part of overall system design that requires special attention because it is the common source for data processing error. The goal of designing input data is to make entry easy and free from errors.

Input design is the process of connecting the user-originated inputs into a computer to used format.

The goal of the input design is to make the data entry logical & free from errors.

## OUTPUT DESIGN

Output design is the process of converting computer data into hard copy that is understood by all. The various outputs have been designed in such a way that they represent the same format that the office and management used to.

Computer output is the most important and direct source of information to the user. Efficient, intelligible output design should improve the systems relationships with the user and help in decision making. A major form of output is the hardcopy from the printer.

Output requirements are designed during system analysis. A good starting point for the output design is the Data Flow Diagram (DFD). Human factors educe issues for design involves addressing internal controls to ensure readability.

The output form in the system is either by screen or by hard copies. Output design aims at communicating the results of the processing of the users. The reports are generated to suit the needs of the users. The reports have to be generated with appropriate levels.

All reports are output formats, maintained details can be reported over crystal reports, this project sustain following reports

## DATABASE DESIGN

The most important consideration in designing the database is how information will be used.

The main objectives of designing a database are:

### Data Integration

In a database, information from several files are coordinated, accessed and operated upon as through it is in a single file. Logically, the information are centralized, physically, the data may be located on different devices, connected through data communication facilities.

### Data Integrity

Data integrity means storing all data in one place only and how each application to access it. This approach results in more consistent information, one update being sufficient to achieve a new record status for all applications, which use it. This leads to less data redundancy; data items need not be duplicated; a reduction in the direct access storage requirement.

### Data Independence

Data independence is the insulation of application programs from changing aspects of physical data organization. This objective seeks to allow changes in the content and organization of physical data without reprogramming of applications and to allow modifications to application programs without reorganizing the physical data.

The tables needed for each module were designed and the specification of each and every column was given based on the records and details collected during record specification of the system study.

## SYSTEM DEVELOPMENT

The key to control maintenance costs is to design systems that are easy to change, so the link between development and maintenance is very strong. Many of the analysis and design methodologies, tools, and techniques employed during system development can be applied to system maintenance, but there are significant differences between development and maintenance. Maintainability is the ease with which software can be understood, corrected, adopted and enhanced.

### DESCRIPTION OF MODULES

To develop this project several step should be followed. There are various modules in this proposed system they are listed below.

* + - * Admin
      * Manager
      * Cashier
      * Billing Details
      * Customer Details

### Admin:

This module has access to create manager, only admin can able to create manager. The admin collects the managers details register manager entry. Once the manager has created, he gave access to login.

### Manager:

This module manager can create the cashier account, once cashier has created, he/she can able to login and register entry for billing details. The manager creates account as well as for pharmacist.

### Cashier:

The cashier entered into the login and promotes the billing. The main work for the cashier is for check the medicine and generates the bill and collects money.

### Billing Details:

This module will show the all the billing details as a date wise, once the cashier collects the money and generate the bill it has to be shown in billing module. Manager or cashier can check the billing details for a date wise.

### Customer Details:

This module will be shown the all the customer details, this customer details has collected by the cashier before entering the bill. This module we can easily track the regular customer.

# TESTING AND IMPLEMENTATION

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client\server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. System testing is the state of implementation that is aimed at assuring that the system works accurately and efficiently. Testing is the vital to the success of the system. System testing makes the logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

### Unit Testing

Unit testing focuses verification efforts on the smallest unit of software design of the module. This is also known as “module testing”. This testing is carried out during programming stage itself. In this testing step, each module is found to be working satisfactorily as regards to the expected output of the modules.

### Integration Testing

Data can be lost across an interface, one module can have adverse effect on another sub function when combined it may not produce the desired major functions. Integration testing is a systematic testing for constructing test to uncover errors associated within an interface.

The objectives taken from unit tested modules and a program structure is built for integrated testing.

All the modules are combined and the test is made.

A correction made in this testing is difficult because the vast expenses of the entire program complicated the isolation of causes. In this integration testing step, all the errors are corrected for next testing process.

### Validation Testing

After the completion of the integrated testing, software is completely assembled as a package; interfacing error has been uncovered and corrected and a final series of software test validation begins.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software function in a manner that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists:

### Output Testing

The next process of validation testing, is output testing of the proposed system, since no system could be successful if it does not produce the required output in the specified format. Asking the user about the format required, list the output to be generated or displayed by the system under considerations.

Output testing is a different test whose primary purpose is to fully exercise the computer based system although each test has a different purpose all the work should verify that all system elements have been properly integrated and perform allocated functions.

The output format on the screen is found to be corrected as the format was designed in the system design phase according to the user needs for the hard copy also; the output testing has not resulted in any correction in the system.

**IMPLEMENTATION**

System implementation is the stage of the project that the theoretical design is turned into a working system. If the implementation stage is not properly planned and controlled, it can cause error. Thus it can be considered to be the most crucial stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective.

Normally this stage involves setting up a coordinating committee, which will act as a sounding board for ideas; complaints and problem. The first task is implementation planning; i.e., deciding on the methods and time scale to be adopted. Apart from planning two major task of preparing for implementation are, education takes place much earlier in the project; at the implementation stage the emphasis must be on training in new skills to give staff confidence they can use the system. Once staff has been trained, the system can be tested.

After the implementation phase is completed and the user staff is adjusted to the changes created by the candidate system, evaluation and maintenance is to bring the new system to standards.

# CONCLUSION

In conclusion, fuel automation systems, or automatic tank gauging systems, offer several advantages to fuel retailers. These systems provide accurate and real-time fuel level information, improve safety, and reduce labor costs. They also help fuel retailers manage their inventory more efficiently and comply with regulations. While fuel automation systems do have some drawbacks, such as high initial costs and maintenance requirements, their benefits generally outweigh the drawbacks, making them a worthwhile investment for many fuel retailers. By implementing fuel automation systems, fuel retailers can improve their operations, reduce risks, and increase profitability.

.

## BIBLIOGRAPY

**Books Referred :**

* Shildt, Herbert. Java: A Beginner's Guide. McGraw-Hill Education, 2018.
* Sierra, Kathy, and Bert Bates. Head First Java, 2nd Edition. O'Reilly Media, 2005.
* Subramaniam, Venkat. Functional Programming in Java: Harnessing the Power of Java 8 Lambda Expressions. Pragmatic Bookshelf, 2014.
* Walrath, Kathy, et al. The Java Tutorial: A Short Course on the Basics. Addison-Wesley Professional, 2018.
* Roger S. Pressman, “ Software Engineering”, Tata me Graw Hill, 2000, Fifth Edition.
* Davin Reader, “Master SQL server 2000” BPB Publications, Third Edition.
* Elias M. Award, “System Analysis and design", Tata McGraw Hill , 3rd edition.
* Jeffrey A. Hoffer, Joey F. George and Joseph S. Valacich (2005), “Modern systems Analysis & Design”, pearson Education, 3rd Edition.

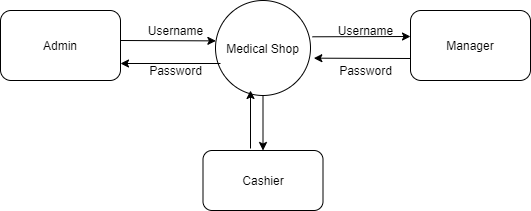
## Websites:

* Baeldung. "Java Tutorials and Articles." Baeldung, 2023, https://www.baeldung.com/java-tutorials.
* GeeksforGeeks. "Java Programming Language." GeeksforGeeks, 2023, <https://www.geeksforgeeks.org/java/>.

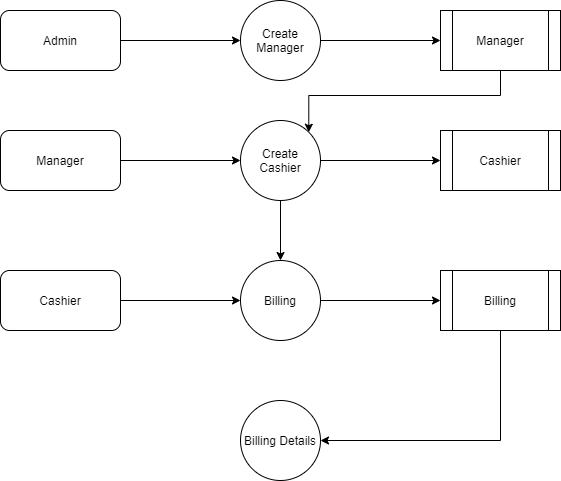
# APPENDICES

## Data Flow Diagram

**Level 0**



**Level 1**

****

## TABLE STRUCTURE

**Table Name :** Admin

**Primary Key :** Admin\_id

**Table Description :** This table is used to maintain the details about Admin

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| Admin\_id | Int | 8 | Primary Key |
| First name | Varchar | 20 | Not null |
| Last name | Varchar | 20 | Not null |
| Email | Varchar | 20 | Not null |
| Phone | Varchar | 10 | Not null |
| Password | Varchar | 20 | Not null |
| Role | Varchar | 10 | Not null |

**Table Name :** Manager

**Primary Key :** Manager\_id

**Table Description :** This table is used to maintain the details about Manager

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| Manager\_id | Int | 8 | Primary Key |
| First name | Varchar | 20 | Not null |
| Last name | Varchar | 20 | Not null |
| Email | Varchar | 20 | Not null |
| Phone | Varchar | 10 | Not null |
| Password | Varchar | 20 | Not null |
| Role | Varchar | 10 | Not null |

**Table Name :** Pharmacist

**Primary Key :** Pharmacist\_id

**Table Description :** This table is used to maintain the details about Admin

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| Pharmacist\_id | Int | 8 | Primary Key |
| First name | Varchar | 20 | Not null |
| Last name | Varchar | 20 | Not null |
| Email | Varchar | 20 | Not null |
| Phone | Varchar | 10 | Not null |
| Password | Varchar | 20 | Not null |
| Role | Varchar | 10 | Not null |

**Table Name :** Salesman

**Primary Key :** Salesman\_id

**Table Description :** This table is used to maintain the details about Salesman

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| Salesman\_id | Int | 8 | Primary Key |
| First name | Varchar | 20 | Not null |
| Last name | Varchar | 20 | Not null |
| Email | Varchar | 20 | Not null |
| Phone | Varchar | 10 | Not null |
| Password | Varchar | 20 | Not null |
| Role | Varchar | 10 | Not null |

## B. Sample Coding

. <?java

session\_start();

$sessionId = $\_SESSION['id'] ?? '';

$sessionRole = $\_SESSION['role'] ?? '';

echo "$sessionId $sessionRole";

if ( !$sessionId && !$sessionRole ) {

header( "location:login.java" );

die();

}

ob\_start();

include\_once "config.java";

$connection = mysqli\_connect( DB\_HOST, DB\_USER, DB\_PASSWORD, DB\_NAME );

if ( !$connection ) {

echo mysqli\_error( $connection );

throw new Exception( "Database cannot Connect" );

}

$id = $\_REQUEST['id'] ?? 'dashboard';

$action = $\_REQUEST['action'] ?? '';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=1024">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="assets/css/bootstrap.min.css">

<link rel="stylesheet" href="assets/css/all.min.css">

<link href="https://fonts.googleapis.com/css2?family=Raleway:wght@400;500;600;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="assets/css/style.css">

<title>Dashboard</title>

</head>

<body>

<!--------------------------------- Secondary Navber -------------------------------->

<section class="topber">

<div class="topber\_\_title">

<span class="topber\_\_title--text">

<?java

if ( 'dashboard' == $id ) {

echo "DashBoard";

} elseif ( 'addManager' == $id ) {

echo "Add Manager";

} elseif ( 'allManager' == $id ) {

echo "Managers";

} elseif ( 'addPharmacist' == $id ) {

echo "Add Pharmacist";

} elseif ( 'allPharmacist' == $id ) {

echo "Pharmacists";

} elseif ( 'addSalesman' == $id ) {

echo "Add Salesman";

} elseif ( 'allSalesman' == $id ) {

echo "Salesmans";

} elseif ( 'userProfile' == $id ) {

echo "Your Profile";

} elseif ( 'editManager' == $action ) {

echo "Edit Manager";

} elseif ( 'editPharmacist' == $action ) {

echo "Edit Pharmacist";

} elseif ( 'editSalesman' == $action ) {

echo "Edit Salesman";

}

?>

</span>

</div>

<div class="topber\_\_profile">

<?java

$query = "SELECT fname,lname,role,avatar FROM {$sessionRole}s WHERE id='$sessionId'";

$result = mysqli\_query( $connection, $query );

if ( $data = mysqli\_fetch\_assoc( $result ) ) {

$fname = $data['fname'];

$lname = $data['lname'];

$role = $data['role'];

$avatar = $data['avatar'];

?>

<img src="assets/img/<?java echo "$avatar"; ?>" height="25" width="25" class="rounded-circle" alt="profile">

<div class="dropdown">

<button class="btn dropdown-toggle" type="button" id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

<?java

echo "$fname $lname (" . ucwords( $role ) . " )";

}

?>

</button>

<div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

<a class="dropdown-item" href="index.java">Dashboard</a>

<a class="dropdown-item" href="index.java?id=userProfile">Profile</a>

<a class="dropdown-item" href="logout.java">Log Out</a>

</div>

</div>

</div>

</section>

<!--------------------------------- Secondary Navber -------------------------------->

<!--------------------------------- Sideber -------------------------------->

<section id="sideber" class="sideber">

<ul class="sideber\_\_ber">

<h3 class="sideber\_\_panel"><i id="left" class="fas fa-laugh-wink"></i> Medical Shop</h3>

<li id="left" class="sideber\_\_item<?java if ( 'dashboard' == $id ) {

echo " active";

}?>">

<a href="index.java?id=dashboard"><i id="left" class="fas fa-tachometer-alt"></i>Dashboard</a>

</li>

<?java if ( 'admin' == $sessionRole ) {?>

<!-- Only For Admin -->

<li id="left" class="sideber\_\_item sideber\_\_item--modify<?java if ( 'addManager' == $id ) {

echo " active";

}?>">

<a href="index.java?id=addManager"><i id="left" class="fas fa-user-plus"></i></i>Add Manager</a>

</li><?java }?>

<li id="left" class="sideber\_\_item<?java if ( 'allManager' == $id ) {

echo " active";

}?>">

<a href="index.java?id=allManager"><i id="left" class="fas fa-user"></i>All Manager</a>

</li>

<?java if ( 'admin' == $sessionRole || 'manager' == $sessionRole ) {?>

<!-- For Admin, Manager -->

<li id="left" class="sideber\_\_item sideber\_\_item--modify<?java if ( 'addPharmacist' == $id ) {

echo " active";

}?>">

<a href="index.java?id=addPharmacist"><i id="left" class="fas fa-user-plus"></i></i>Add

Pharmacist</a>

</li><?java }?>

<li id="left" class="sideber\_\_item<?java if ( 'allPharmacist' == $id ) {

echo " active";

}?>">

<a href="index.java?id=allPharmacist"><i id="left" class="fas fa-user"></i>All Pharmacist</a>

</li>

<?java if ( 'admin' == $sessionRole || 'manager' == $sessionRole || 'pharmacist' == $sessionRole ) {?>

<!-- For Admin, Manager, Pharmacist-->

<li id="left" class="sideber\_\_item sideber\_\_item--modify<?java if ( 'addSalesman' == $id ) {

echo " active";

}?>">

<a href="index.java?id=addSalesman"><i id="left" class="fas fa-user-plus"></i>Add Salesman</a>

</li><?java }?>

<li id="left" class="sideber\_\_item<?java if ( 'allSalesman' == $id ) {

echo " active";

}?>">

<a href="index.java?id=allSalesman"><i id="left" class="fas fa-user"></i>All Salesman</a>

</li>

</ul>

<footer class="text-center"><span>PMS</span><br>©2020 PMS All right reserved.</footer>

</section>

<!--------------------------------- #Sideber -------------------------------->

<!--------------------------------- Main section -------------------------------->

<section class="main">

<div class="container">

<!-- ---------------------- DashBoard ------------------------ -->

<?java if ( 'dashboard' == $id ) {?>

<div class="dashboard p-5">

<div class="total">

<div class="row">

<div class="col-3">

<div class="total\_\_box text-center">

<h1>2453</h1>

<h2>Total Sell</h2>

</div>

</div>

<div class="col-3">

<div class="total\_\_box text-center">

<h1>

<?java

$query = "SELECT COUNT(\*) totalManager FROM managers;";

$result = mysqli\_query( $connection, $query );

$totalManager = mysqli\_fetch\_assoc( $result );

echo $totalManager['totalManager'];

?>

</h1>

<h2>Manager</h2>

</div>

</div>

<div class="col-3">

<div class="total\_\_box text-center">

<h1>

<?java

$query = "SELECT COUNT(\*) totalPharmacist FROM pharmacists;";

$result = mysqli\_query( $connection, $query );

$totalPharmacist = mysqli\_fetch\_assoc( $result );

echo $totalPharmacist['totalPharmacist'];

?>

</h1>

<h2>Pharmacist</h2>

</div>

</div>

<div class="col-3">

<div class="total\_\_box text-center">

<h1><?java

$query = "SELECT COUNT(\*) totalSalesman FROM salesmans;";

$result = mysqli\_query( $connection, $query );

$totalSalesman = mysqli\_fetch\_assoc( $result );

echo $totalSalesman['totalSalesman'];

?></h1>

<h2>Salesman</h2>

</div>

</div>

</div>

</div>

</div>

<?java }?>

<!-- ---------------------- DashBoard ------------------------ -->

<!-- ---------------------- Manager ------------------------ -->

<div class="manager">

<?java if ( 'allManager' == $id ) {?>

<div class="allManager">

<div class="main\_\_table">

<table class="table">

<thead>

<tr>

<th scope="col">Avater</th>

<th scope="col">Name</th>

<th scope="col">Email</th>

<th scope="col">Phone</th>

<?java if ( 'admin' == $sessionRole ) {?>

<!-- Only For Admin -->

<th scope="col">Edit</th>

<th scope="col">Delete</th>

<?java }?>

</tr>

</thead>

<tbody>

<?java

$getManagers = "SELECT \* FROM managers";

$result = mysqli\_query( $connection, $getManagers );

while ( $manager = mysqli\_fetch\_assoc( $result ) ) {?>

<tr>

<td>

<center><img class="rounded-circle" width="40" height="40" src="assets/img/<?java echo $manager['avatar']; ?>" alt=""></center>

</td>

<td><?java printf( "%s %s", $manager['fname'], $manager['lname'] );?></td>

<td><?java printf( "%s", $manager['email'] );?></td>

<td><?java printf( "%s", $manager['phone'] );?></td>

<?java if ( 'admin' == $sessionRole ) {?>

<!-- Only For Admin -->

<td><?java printf( "<a href='index.java?action=editManager&id=%s'><i class='fas fa-edit'></i></a>", $manager['id'] )?></td>

<td><?java printf( "<a class='delete' href='index.java?action=deleteManager&id=%s'><i class='fas fa-trash'></i></a>", $manager['id'] )?></td>

<?java }?>

</tr>

<?java }?>

</tbody>

</table>

</div>

</div>

<?java }?>

<?java if ( 'addManager' == $id ) {?>

<div class="addManager">

<div class="main\_\_form">

<div class="main\_\_form--title text-center">Add New Manager</div>

<form action="add.java" method="POST">

<div class="form-row">

<div class="col col-12">

<label class="input">

<i id="left" class="fas fa-user-circle"></i>

<input type="text" name="fname" placeholder="First name" required>

</label>

</div>

<div class="col col-12">

<label class="input">

<i id="left" class="fas fa-user-circle"></i>

<input type="text" name="lname" placeholder="Last Name" required>

</label>

</div>

<div class="col col-12">

<label class="input">

<i id="left" class="fas fa-envelope"></i>

<input type="email" name="email" placeholder="Email" required>

</label>

</div>

<div class="col col-12">

<label class="input">

<i id="left" class="fas fa-phone-alt"></i>

<input type="number" name="phone" placeholder="Phone" required>

</label>

</div>

<div class="col col-12">

<label class="input">

<i id="left" class="fas fa-key"></i>

<input id="pwdinput" type="password" name="password" placeholder="Password" required>

<i id="pwd" class="fas fa-eye right"></i>

</label>

</div>

<input type="hidden" name="action" value="addManager">

<div class="col col-12">

<input type="submit" value="Submit">

</div>

</div>

</form>

</div>

</div>

<?java }?>

<?java if ( 'editManager' == $action ) {

$managerId = $\_REQUEST['id'];

$selectManagers = "SELECT \* FROM managers WHERE id='{$managerId}'";

$result = mysqli\_query( $connection, $selectManagers );

$manager = mysqli\_fetch\_assoc( $result );?>

<div class="addManager">

<div class="main\_\_form">

<div class="main\_\_form--title text-center">Update Manager</div>

<form action="add.java" method="POST">

<div class="form-row">

<div class="col col-12">

<label class="input">

<i id="left" class="fas fa-user-circle"></i>

<input type="text" name="fname" placeholder="First name" value="<?java echo $manager['fname']; ?>" required>

</label>

</div>

<div class="col col-12">

<label class="input">

<i id="left" class="fas fa-user-circle"></i>

<input type="text" name="lname" placeholder="Last Name" value="<?java echo $manager['lname']; ?>" required>

</label>

</div>

<div class="col col-12">

<label class="input">

<i id="left" class="fas fa-envelope"></i>

<input type="email" name="email" placeholder="Email" value="<?java echo $manager['email']; ?>" required>

## D. Sample Input

## Login page admin, pharmacist and salesman

## 

## Input of Manager registration

## 

## Input of pharmacist registration

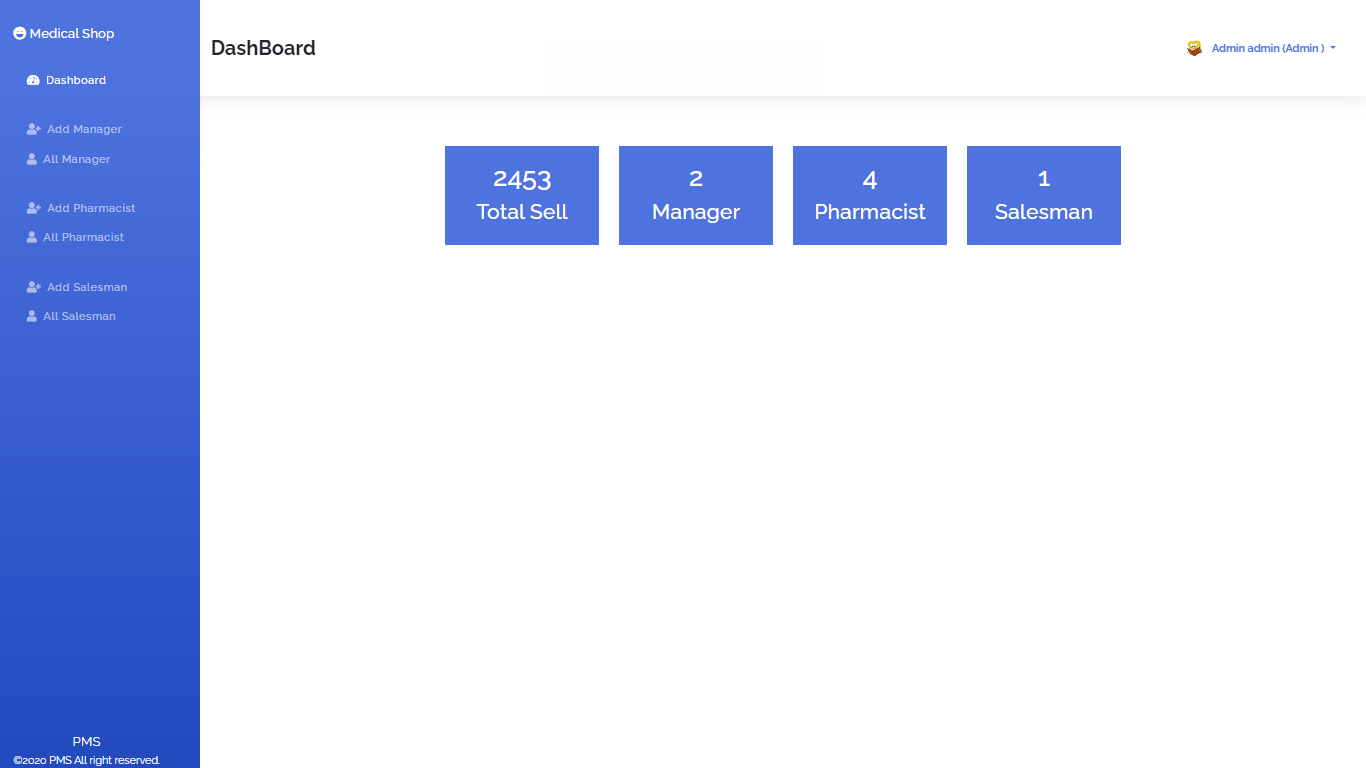
## 

## Input of salesman registration

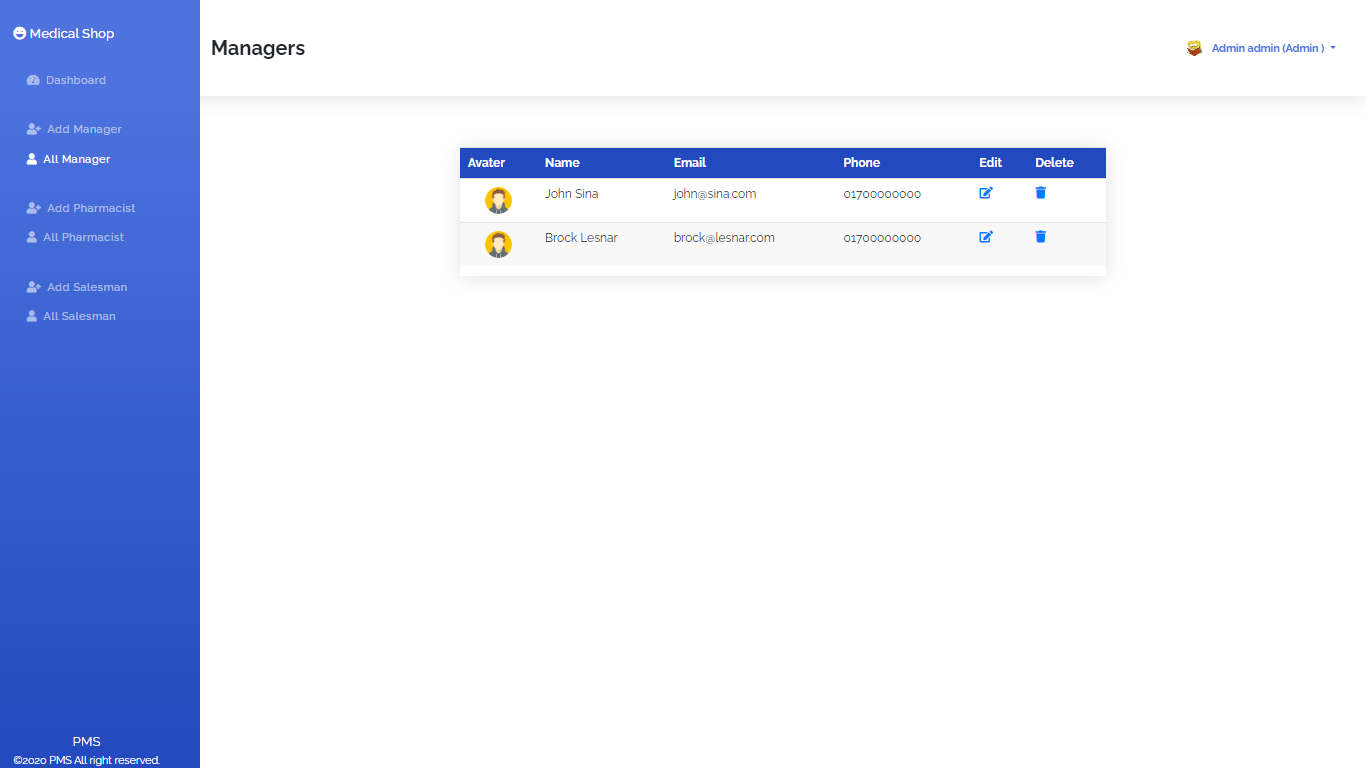
## 

## E. Sample Output

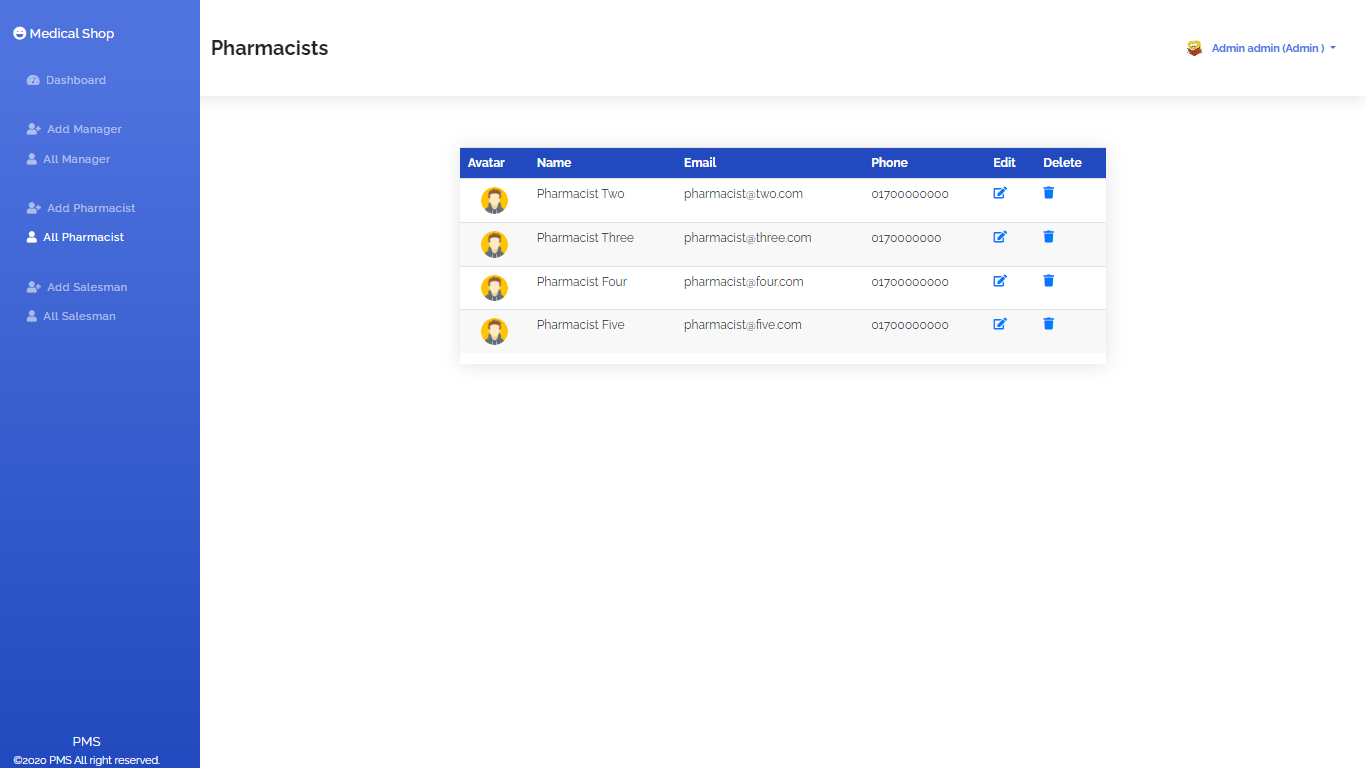
**Output of Dashboard**



Output of Manager details



Output of pharmacist details



Output of sales man details

