SYNOPSIS

The project is entitled "**Medical Billing System**", which is developed using HTML, CSS, Bootstrap, and JavaScript as the Front-End, PHP as the Back-End, and MySQL as the database.

The main objective of this application is to streamline the medical billing process, improve inventory management, and ensure accurate customer record-keeping. It enables users to efficiently manage customer details, medicine stock, and billing operations through an intuitive and user-friendly interface.

The system allows users to add customer details, manage stock, and generate bills seamlessly. It features a well-structured billing module where users can create invoices by selecting a customer, brand, item, rate, quantity, and GST. Multiple products can be added to a single invoice, and the system also supports generating PDF invoices for better record-keeping.

The primary goal of the Medical Billing System is to provide an automated solution for billing and stock management in medical stores. It ensures accuracy, reduces manual errors, and saves time in handling transactions.

The system is designed to be **user-friendly, secure, and efficient**, making it an essential tool for medical businesses to manage their operations effectively.

INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The Medical Billing System is designed to provide an efficient, user-friendly, and seamless platform for managing customer details, inventory, and billing operations in medical stores. The main objective of this system is to simplify and automate the billing process, ensuring accuracy, reducing manual errors, and improving overall operational efficiency.

By integrating a structured database and an intuitive interface, the system allows users to add, update, and manage customer records, track stock availability, and generate invoices with ease. The billing module enables the creation of detailed invoices by selecting a customer, brand, item, quantity, applicable GST, and other relevant details. Users can add multiple products in a single bill and generate PDF invoices for better record-keeping and customer reference.

The system is designed to be highly responsive, accessible, and secure, ensuring smooth operations on various devices, including desktops, tablets, and mobile phones. It also includes stock management functionalities, allowing users to track and update inventory levels, generate stock reports, and maintain a stock ledger for different time periods.

This Medical Billing System serves as an essential tool for medical businesses, streamlining their workflow and ensuring that all transactions are recorded and processed efficiently. With its organized structure, automation features, and secure database management, the system enhances productivity and ensures seamless business operations for medical stores.

Modules

- Login
- Dashboard
- Home
- Manage Customers
- Manage Brands
- Manage Variant
- Manage Item
- Manage Product
- Manage Stock
- View Ledger
- Add Bill
- List Bill

1.2 LANGUAGE USED IN PROJECT

HTML

HyperText Markup Language (HTML) is a computer language devised which allows website creation. It is the HTML code that provides an overall framework of how the site will look. These websites can then be viewed by anyone else connected to the Internet.

CSS

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation and layout of HTML tags. It controls the presentation aspect of the site and allows your site to have its unique look. It does this by maintaining style sheets, which sit on top of other style rules and are triggered based on other inputs, such as device screen size and resolution.

JavaScript

JavaScript is an event-based imperative language that is used to transform a static HTML page into a dynamic interface. Some of the dynamic behaviors that can be generated by JavaScript are the following: can change HTML content; can change HTML attributes; can change HTML styles (CSS); and can validate data.

JQuery

JQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use application programming interface (API) that works across a multitude of browsers. With a combination of versatility and extensibility, JQuery has changed the way that millions of people write JavaScript.

Introduction to PHP:

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor.
- PHP is a server-side scripting language that is embedded in HTML
- Access cookies variables and set cookies.
- Using PHP, you can restrict users from accessing some pages of your website.

- It can encrypt data.
- PHP is one of the most widely used languages on the web
- You add, delete, and modify elements within your database through PHP.
- PHP code is enclosed within <?php ?> tags.
- Statements end with a semicolon ;.
- Variables in PHP start with the dollar sign $.
- They are loosely typed, meaning data types are automatically converted as needed.
- Arrays are versatile data structures in PHP.

Benefits of PHP:

- Free to use, modify, and distribute.
- Simple syntax, easy for beginners to grasp.
- Runs on various operating systems.
- Seamless integration with databases and other technologies.

About PHP:

PHP was created by Rasmus Lerdorf in 1994 as a set of Common Gateway Interface (CGI) binaries written in C. It evolved into a scripting language designed for web development. PHP scripts are executed on the server, generating HTML content that is then sent to the client's web browser. This allows for dynamic and interactive web pages.

PHP code is embedded within HTML, typically enclosed within <?php ?> tags. It's designed to be easy to learn and use, with a syntax similar to C and Perl. PHP is versatile and can be used for various purposes, including server-side scripting, command-line scripting, and writing desktop applications. PHP has extensive support for interacting with databases, including MySQL, PostgreSQL, SQLite, and others. It offers functions and extensions for connecting to and querying databases.

There are numerous PHP frameworks and content management systems (CMSs) available, such as Laravel, Symfony, CodeIgniter, and WordPress. These frameworks provide pre-built components and libraries to expedite development. PHP has a large and active community of developers who contribute to its development and provide support through forums, documentation, tutorials, and online resources.

.

SQL Server:

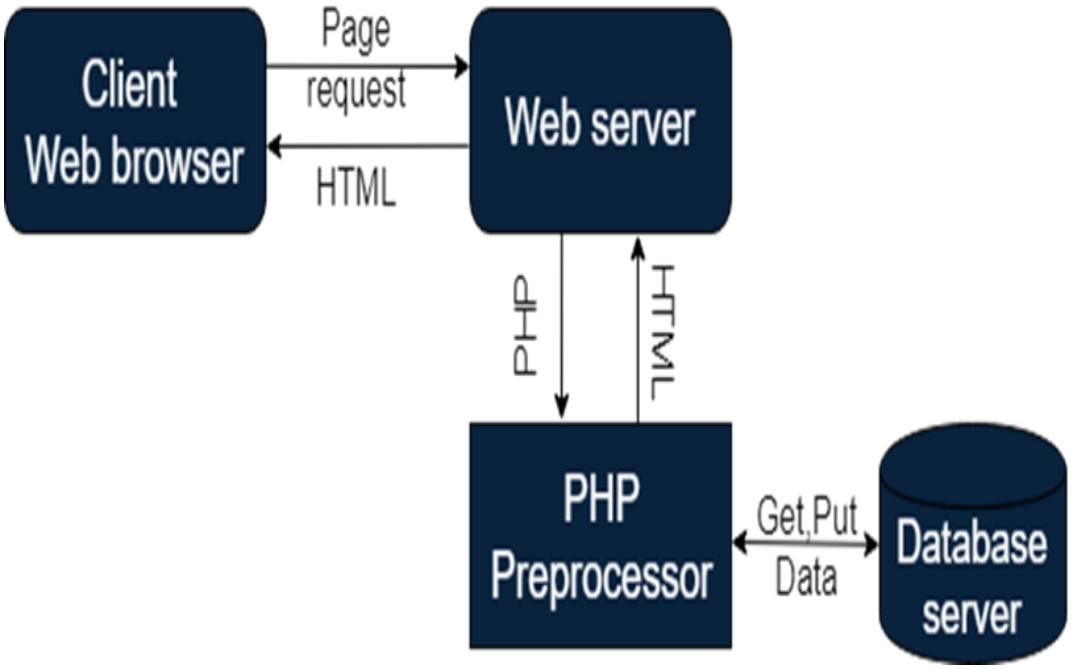
- SQL stands for Structured Query Language.
- SQL lets you access and manipulate databases.
- SQL can execute queries against a database.
- SQL can retrieve data from a database.
- SQL can insert records in a database.

SQL stores information which is called a database. To use SQL, you will need to follow these four steps −

- SQL server replication services are used by SQL servers to replicate and synchronize the database objects, either entirely or a subset of the objects database across the network.

- SQL server analyses add data capabilities for SQL server database. Analysis services support the XML for analysis standard as an underlying communication protocol.

- Originally introduced as a post-release for SQL server 20200, notification services were bundled as a part of the MS SQL server platform for the first and only time with SQL server 2005. .



1.3 SYSTEM SPECIFICATION

HARDWARE SPECIFICATION

Processor	Intel Core i3 Processor
Speed	1.20GHz
Memory	8 GB RAM
Storage	512 GB SSD
Keyboard	104 Keys Keyboard
Pointer device	USB mouse

SOFTWARE SPECIFICATION

Front End	HTML, CSS, Java Script
Back End	PHP
DataBase	MYSQL
Operating System	Windows 10
Server	C Panel

2. SYSTEM STUDY

2.1 Existing System:

The existing system is a manual process where customer details, stock, and billing information are recorded on paper or spreadsheets. Managing historical data is difficult, leading to errors despite repeated cross-checking. Stock mismanagement and billing inaccuracies are common, making the process inefficient. To address these issues, an automated Medical Billing System is required for accuracy, efficiency, and better record-keeping.

2.2 Proposed System

The proposed Medical Billing System automates the management of customer details, inventory, and billing processes. The system enables the admin to efficiently add, update, and retrieve customer and stock information while generating invoices accurately. It eliminates manual errors, improves data consistency, and ensures seamless inventory tracking. The system also allows for the generation of PDF invoices for record-keeping and better business management.

Drawbacks

- Inconsistency in data entry.
- Prone to human errors despite verification.
- Difficulty in tracking stock and sales.
- Time-consuming manual processes.
- Lack of security and data backup.
- Risk of duplication in records.

3. SYSTEM DESIGN

3.1 INPUT DESIGN

The Medical Billing System is designed to minimize errors, simplify data entry, and enhance efficiency. The input forms ensure accurate and secure data collection for managing customer details, inventory, and billing processes. The system provides a user-friendly interface that reduces manual effort, prevents duplication, and maintains data integrity.

Input fields are structured to validate entries, ensuring that only correct and complete information is stored. The design focuses on reducing delays, avoiding unnecessary steps, and maintaining security while keeping the process simple and efficient.

Input Design considered the following things:

- What data should be given as input?
- How should the data be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when errors occur.

3.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source of information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

Designing computer output should proceed in an organized, well-thought-out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can be used easily and effectively. When analyzing design computer output, they should

- Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.
- Create documents, reports, or other formats that contain information produced by the system

RELATIONAL DATABASE MANAGEMENT SYSTEM

A relational model represents the database as a collection of relational. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tipple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables each of which is assigned a unique name. A row in a tale represents a set of related values.

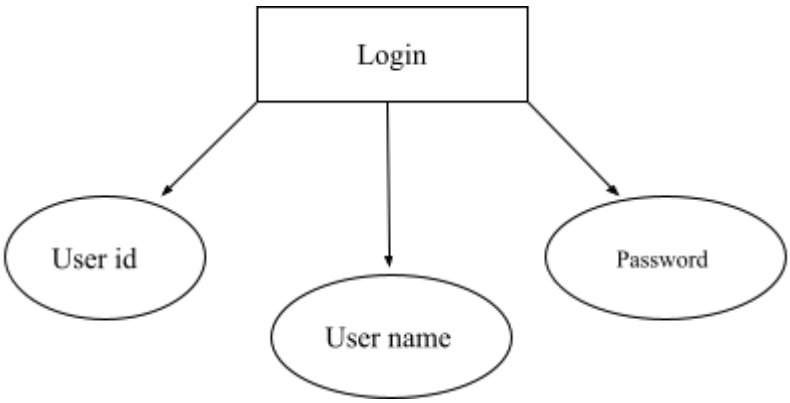
Relations, Domains & Attributes:

A table is a relation. The row in a table is called tipples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Relationship and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

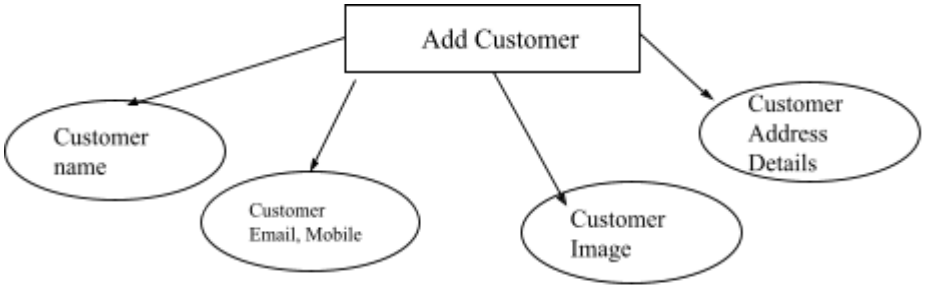
3.3 DATA FLOW DIAGRAM

A Data flow diagram (DFD) shows the functional relationships of the values A computed by a system, including input values, output values & internal data stores. A data flow diagram contains **processes** that transform data, **data flows** that move data, **external entity** objects that produce & data **store** objects that store data passively.

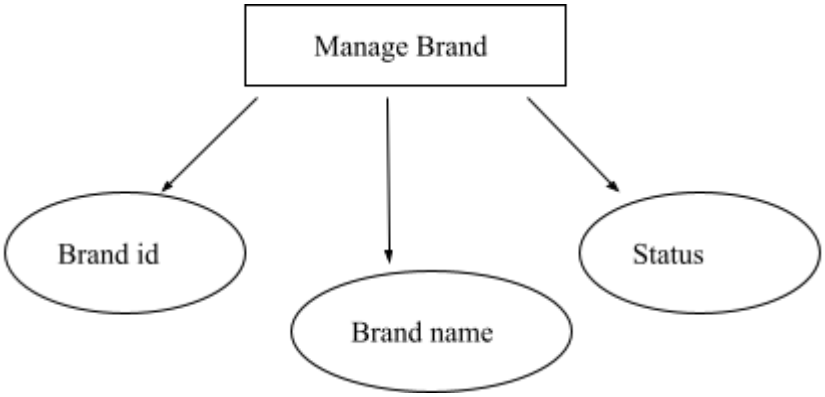
DFD Level-1:



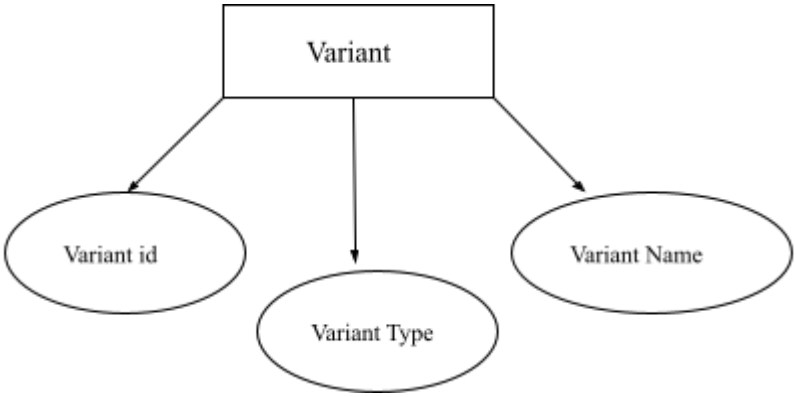
DFD Level-2:



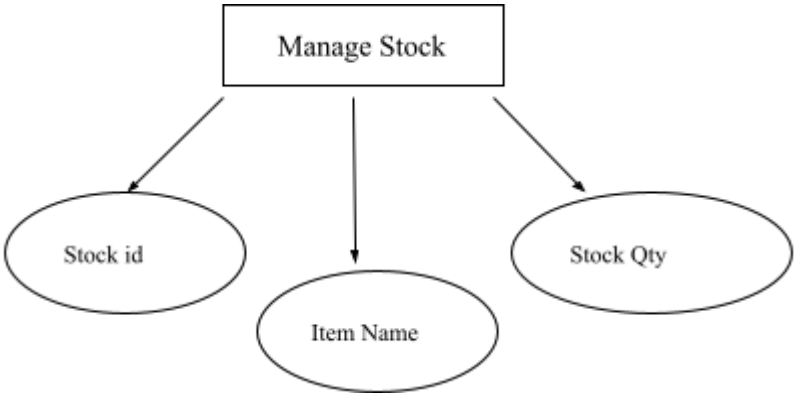
DFD Level-3:



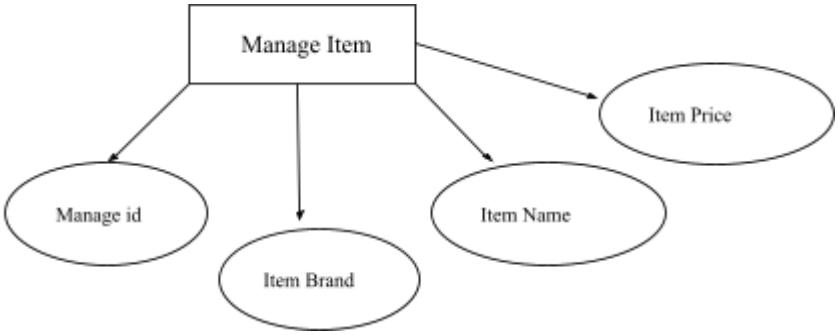
DFD Level-4:



DFD Level-5:



DFD Level-6:



3.4 DATABASE TABLES

The project **"Content Management System" is** fully connected with the database **SQL**. The main reason for choosing this database is the flexibility and the data security that it can provide to the entire program. The general theme behind databases is to handle information as a whole. No artificiality is normally embedded in separate files or applications. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive, and flexible for the user.

Database design is the most critical part of the Design phase. An elegantly designed, well-defined Database is a strong foundation for the whole system. The tables are classified and fully normalized by inputting Primary Keys and Foreign Keys for each table.

Table Name: Customers
Primary Key: ID

Attributes	Data Types	Length	Description
Id	INT	10	Default
First Name	VARCHAR	10	Default
Mobile	INT	10	Default
Image	VARCHAR	50	Default
Email	VARCHAR	2	Default
Address	VARCHAR	2	Default
State	VARCHAR	2	Default
Status	TINY INT	2	Default

Table Name: Item

Primary Key: ID

Attributes	Data Types	Length	Description
ID	INT	10	Default
Brand Id	VARCHAR	15	Default
Product Name	VARCHAR	80	Default
Price	TINY INT	3	Default
Variant Type	INT	15	Default
Variant	TINY INT	2	Default
Status	TINY INT	2	Default
Stock_warning	TINY INT	2	Default
Stocks	TINY INT	3	Default
Create date	DATE TIME	3	Default
Update by	TINY INT	3	Default

Table Name: Brand

Primary Key: ID

Attributes	Data Types	Length	Description
ID	INT	3	Default
Brand Name	INT	50	Default
Description	VARCHAR	2	Default
Updated by	TINY INT	3	Default
Updated date	DATE TIME	null	Default
status	TINY INT	3	Default

Table Name: Variant

Primary Key: ID

Attributes	Data Types	Length	Description
ID	INT	10	Default
Variant Type	VARCHAR	15	Default
Variant	VARCHAR	50	Default
Variant Name	VARCHAR	6	Default
Created by	TINY INT	3	Default
Created date	DATE TIME	null	Default
Status	TINY INT	3	Default

Table Name: Product Order

Primary Key: ID

Attributes	Data Types	Length	Description
ID	INT	10	Default
product_id	INT	100	Default
brand_id	INT	50	Default
Brand name	VARCHAR	50	Default
Product Id	VARCHAR	10	Default
Qty	VARCHAR	50	Default
Price	VARCHAR	100	Default
Variant	VARCHAR	10	Default
Variant Type	VARCHAR	100	Default
Total	INT	4	Default
Created date	DATE TIME	null	Default
Updated by	TINY INT	3	Default
Updated date	DATE TIME	null	Default
Status	TINY INT	3	Default

SYSTEM TESTING AND IMPLEMENTATION

4. SYSTEM TESTING AND IMPLEMENTATION

4.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies, and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail unacceptably. There are various types of tests. Each test type addresses a specific testing requirement.

UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases. Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages, and responses must not be delayed.

Features to be tested

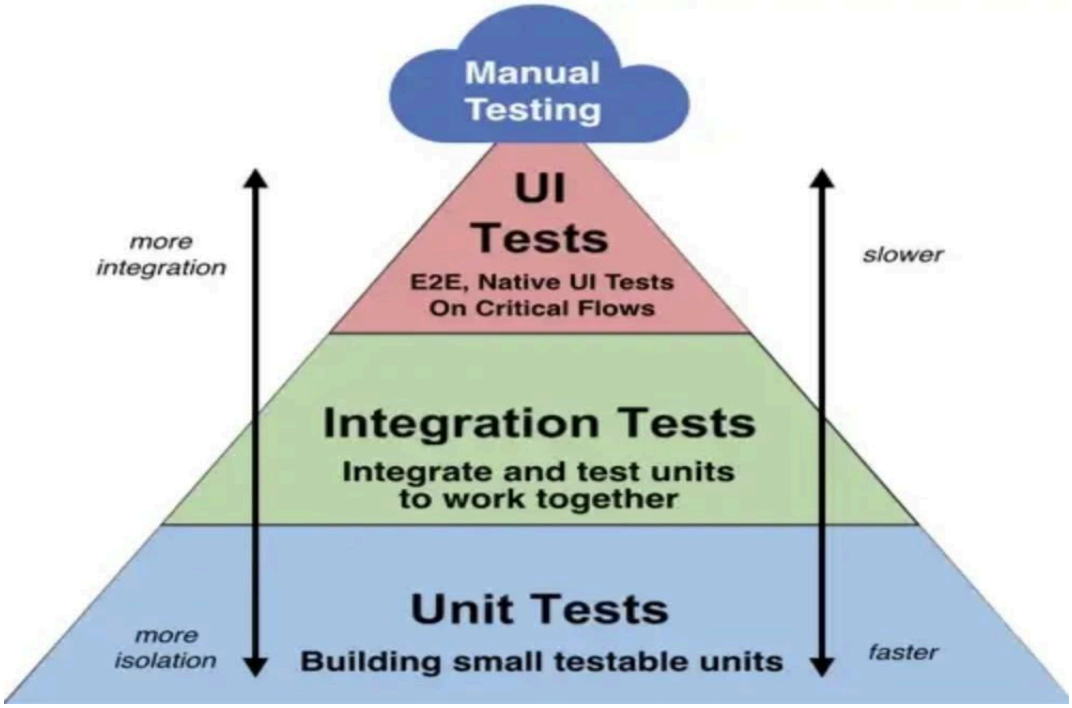
- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications.

Test Results:

All the test cases mentioned above passed successfully. No defects were encountered.



4.2 SYSTEM IMPLEMENTATION

System implementation is the important stage of a project when the theoretical design is tuned into a practical system. After proper testing and validation, system implementation should be done. System implementation includes all those activities that take place to convert an old system to a new one. The new system may be new. Replacing an existing manual or automated system may be a major modification to an existing system.

Implementation Procedure

Implementation is the stage, which is crucial in the life cycle of the new system design. The main stage in the implementation is planning, training, and system testing. Implementation is converting a new or revised system into an operational one. It is the process of changing from the old system to the new one. After the system is implemented, the user conducts a review of system.

4.3 MAINTENANCE OF THE SYSTEM

All systems are dynamic and subject to constantly changing requirements. Efforts must be devoted to adapting them and design must be flexibly specified so that such changes can be easily implemented. This activity is called system maintenance. It includes the improvement of system functions and the correction of errors. We may define system maintenance by describing four activities that are undertaken after a program is released for use.

CORRECTIVE MAINTENANCE

The first maintenance activity occurs since it is unreasonable to assume that system testing will uncover all errors in a large software system. The process of including the diagnosis and correction of one or more errors is called corrective maintenance.

ADAPTIVE MAINTENANCE

This activity that contributes to the definition of maintenance occurs since rapid change is encountered in every aspect of computing. Therefore, adaptive maintenance modifies software to properly interface with the changing environment.

PERFECTIVE MAINTENANCE

This activity involves recommendations for new capabilities modifications to the existing functions and general enhancements when the software is used to satisfy these requests, perfective maintenance is performed.

APPENDIX

5.APPENDIX

5.1 SAMPLE CODE

Login Page

```
<?php
ob_start();
ob_clean();
session_start();
extract($_REQUEST);
include 'dilg/cnt/join.php';
include 'global-functions.php';
$msg='';
if(isset($_SESSION['USERNAME']) || isset($_COOKIE['USERNAME'])){
$_SESSION['USERNAME'] = $_COOKIE['USERNAME'];
$_SESSION['UID'] = $_COOKIE['UID'];
$_SESSION['SName'] = $_COOKIE['CName'];
$_SESSION['Team'] = $_COOKIE['Team'];
header('location:home.php');
die();
}
if(isset($_POST['Login']))
{
$UserName=$_POST['username'];
$password=$_POST['password'];
$encrypted_password = encrypt_decrypt('encrypt', $password );
$login_select=mysqli_query($conn,"select * from  user where UserName = '$UserName' and Password =
'$encrypted_password' ");
if(mysqli_num_rows($login_select)>>0){
$row=mysqli_fetch_array($login_select);
$status= $row['status'];
if($status =="1"){

$_SESSION['USERNAME']=$row['UserName'];
```

```
$_SESSION['UID']=$row['Id'];
$_SESSION['USERTYPE']=$row['user_type'];
$_SESSION['SNAME']=$row['name'];
$_SESSION['Team']=$row['team'];
setcookie('CName',$row['name'],time()+60*60*24*30);
setcookie('USERNAME',$row['UserName'],time()+60*60*24*30);
setcookie('Team',$row['team'],time()+60*60*24*30);
setcookie('UID',$row['Id'],time()+60*60*24*30);
setcookie('USERTYPE',$row['user_type'],time()+60*60*24*30);
header('location:home.php');
}
else{
$msg = "Account deactivated! Kindly contact your admin to reactivate!!!";
}
}
else
{
$msg = "Incorrect Username or Password";
}
}

?>

<html lang="en">

<head>

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<!--favicon-->
<link rel="icon" href="assets/images/Our/fav-icon-1.PNG" type="image/png" />
<!--plugins-->
```

```
<link href="assets/plugins/simplebar/css/simplebar.css" rel="stylesheet" />
<link href="assets/plugins/perfect-scrollbar/css/perfect-scrollbar.css" rel="stylesheet" />
<link href="assets/plugins/metismenu/css/metisMenu.min.css" rel="stylesheet" />
<!-- loader-->
<link href="assets/css/pace.min.css" rel="stylesheet" />
<script src="assets/js/pace.min.js"></script>

<link href="assets/css/bootstrap.min.css" rel="stylesheet">
<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500&amp;display=swap"
rel="stylesheet">
<link href="assets/css/app.css" rel="stylesheet">
<link href="assets/css/icons.css" rel="stylesheet">
<title>Hair Transplant</title>
</head>

<body class="bg-login">
<!--wrapper-->
<div class="wrapper">
<div class="section-authentication-signin d-flex align-items-center justify-content-center my-5
my-lg-0">
<div class="container-fluid">
<div class="row row-cols-1 row-cols-lg-2 row-cols-xl-3">
<div class="col mx-auto">
<div class="mb-4 text-center">
<img src="assets/images/Our/logo.PNG" style="width:200px;">
</div>
<div class="card">
<div class="card-body">
<div class="border p-4 rounded">
<div class="text-center">
<h3 class="">Admin Sign in</h3>
</div>
```

```
<div class="login-separater text-center mb-4"> <span>Login</span>
<hr/>
</div>
<div class="form-body">
<form action="#" class="row g-3" method="post">
<? if($msg !=''){ ?>    <div class="alert alert-danger border-0 bg-danger alert-dismissible fade show
py-2">
<div class="d-flex align-items-center">
<div class="font-35 text-white"><i class="bx bxs-message-square-x"></i>
</div>
<div class="ms-3">
<h6 class="mb-0 text-white">Alerts</h6>
<div class="text-white"><?=$msg; ?></div>
</div>
</div>
<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
</div> <? } ?>
<div class="col-12">
<label for="inputEmailAddress" class="form-label">Enter Username</label>
<input type="text" name="username" class="form-control" id="inputEmailAddress"
placeholder="Username">
</div>
<div class="col-12">
<label for="inputChoosePassword" class="form-label">Enter Password</label>
<div class="input-group" id="show_hide_password">
<input type="password" class="form-control border-end-0" id="inputChoosePassword"
name="password" value="" placeholder="Enter Password"> <a href="javascript:;"
class='input-group-text bg-transparent'><i class='bx bx-hide'></i></a>
</div>
</div>
<div class="col-md-6">
<div class="form-check form-switch">
```

```
<input class="form-check-input" type="checkbox" id="flexSwitchCheckChecked" checked>
<label class="form-check-label" for="flexSwitchCheckChecked">Remember Me</label>
</div>
</div>
<div class="col-md-6 text-end">       <a href="#" class="color-146236">Forgot Password ?</a>
</div>
<div class="col-12">
<div class="d-grid">
<button type="submit" class="btn btn-primary" name="Login"><i class="bx bxs-lock-open"></i>Sign
in</button>
</div>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
<!--end row-->
</div>
</div>
</div>
<script src="assets/js/bootstrap.bundle.min.js"></script>
<!--plugins-->
<script src="assets/js/jquery.min.js"></script>
<script src="assets/plugins/simplebar/js/simplebar.min.js"></script>
<script src="assets/plugins/metismenu/js/metisMenu.min.js"></script>
<script src="assets/plugins/perfect-scrollbar/js/perfect-scrollbar.js"></script>
<!--Password show & hide js -->
<script>
$(document).ready(function () {
```

```
$("#show_hide_password a").on('click', function (event) {
event.preventDefault();
if ($('#show_hide_password input').attr("type") == "text") {
$('#show_hide_password input').attr('type', 'password');
$('#show_hide_password i').addClass("bx-hide");
$('#show_hide_password i').removeClass("bx-show");
} else if ($('#show_hide_password input').attr("type") == "password") {
$('#show_hide_password input').attr('type', 'text');
$('#show_hide_password i').removeClass("bx-hide");
$('#show_hide_password i').addClass("bx-show");
}
});
});
</script>
<!--app JS-->
<script src="assets/js/app.js"></script>
</body>

</html>
```

Variant Page

```
<?php
function main() {
extract($_REQUEST);
include 'dilg/cnt/join.php';
include 'global-functions.php';
$ID=$_GET['id'];
date_default_timezone_set('Asia/Kolkata');
$currentTime = date('Y-m-d H:i:s');



if($Submit=='Add')
{

//fetch balance stock

$select_item1=mysqli_query($conn,"select * from stock where item_id='$item_id' order by id desc limit
1");
$row_items1=mysqli_fetch_array($select_item1);
  $item_id1=$row_items1['item_id'];

  $balance_stock1=$row_items1['balance_stock'];
$new_balance_stock=$balance_stock1+$stock;

//item name
$select_item_name=mysqli_query($conn,"select * from item where id='$item_id' and type='0'");
$row_iname=mysqli_fetch_array($select_item_name);
$iname=$row_iname['item_name'];



if($item_id1!=$item_id){
```

```
$insert_stock=mysqli_query($conn,"insert into stock set
type='IN',brand_name='$brand_name',item_id='$item_id',item_name='$iname',stock='$stock',balance_st
ock='$stock',  created_by = ".$_SESSION['UID']."', created_datetime = '$currentTime'");

}else{

$insert_stock1=mysqli_query($conn,"insert into stock set
type='IN',brand_name='$brand_name',item_id='$item_id',item_name='$iname',stock='$stock',balance_st
ock='$new_balance_stock',  created_by = ".$_SESSION['UID']."', created_datetime = '$currentTime'");



}

if($insert_stock1 ||$insert_stock )
{
$msg = 'Stock Details Inseted Successfully';
header('Location:manage-stock.php?msg='.$msg);
}
else
{
$alert_msg = 'Could not able to Inserted try once again!!!';
  header('Location:manage-stock.php?alert_msg='.$alert_msg);
}

}


if($from_date!='' && $end_date!=''){
  $from_date=$from_date;
  $end_date=$end_date;
}else{
```

```
$from_date=date('Y-m-01');
$end_date=date('Y-m-t');
}

$subqry="and (date(created_datetime) between '".$from_date."' and '".$end_date."')";

?>



<div class="page-breadcrumb d-none d-sm-flex align-items-center mb-3">
  <h5 class="mb-0 text-dark">Manage Stock</h5>

<div class="ms-auto">
<div class="col">
<!-- Button trigger modal -->
<button type="button" class="btn btn-primary" data-bs-toggle="modal"
data-bs-target="#exampleExtraLargeModal" onClick="getedit(0)">Add Stock</button>

</div>
</div>
</div>
<hr/>
<?  $select_stock=mysqli_query($conn,"select * from stock where type='IN' $subqry order by
created_datetime desc ");

?>
<? if($msg !=''){ ?><div class="alert alert-success border-0 bg-success alert-dismissible fade show
py-2">
<div class="d-flex align-items-center">
<div class="font-35 text-white"><i class="bx bxs-check-circle"></i>
</div>
```

```
<div class="ms-3">
<h6 class="mb-0 text-white">Success Alerts</h6>
<div class="text-white"><?=$msg; ?></div>
</div>
</div>
<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
</div> <? } ?>
<? if($alert_msg !=''){ ?> <div class="alert alert-danger border-0 bg-danger alert-dismissible fade show
py-2">
<div class="d-flex align-items-center">
<div class="font-35 text-white"><i class="bx bxs-message-square-x"></i>
</div>
<div class="ms-3">
<h6 class="mb-0 text-white">Alerts</h6>
<div class="text-white"><?=$alert_msg; ?></div>
</div>
</div>
<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
</div> <? } ?>


<div class="card border-top border-0 border-4 border-primary">
<div class="card-body p-5">
<form action="" method="POST">
<div class="row g-3">


<div class="col-md-3">
<label for="inputFirstName" class="form-label">From</label>
<input type="date" name="from_date" class="form-control" value="<?=$from_date;?>" required>
</div>
```

```
<div class="col-md-3">
<label for="inputFirstName" class="form-label">To</label>
<input type="date" name="end_date" class="form-control" value="<?=$end_date;?>" required>
</div>


<div class="col-md-4 mt-3 align-self-end">
<input type="submit" name="Submit" class="btn btn-primary px-3" value="Search">
</div>

</div>
</form>
</div>
</div>

<div class="card">
<div class="card-body">
<div class="table-responsive">
<table id="example2" class="table table-striped table-bordered" style="width:100%">
<thead>
<tr>
<!-- <th>Date</th> -->
<th class="d-none">SNo</th>
<th>Date</th>
<th>Brand Name</th>
<th>Item Name</th>
<th>Stock Added</th>
<th>Balance Stock</th>

</tr>
</thead>
<tbody>
```

```
<?
if(mysqli_num_rows($select_stock)>>0){
$SNo = 0;
while($row_brand=mysqli_fetch_array($select_stock))
{
$SNo = $SNo + 1;
$id=$row_brand['id'];
$brand_name=$row_brand['brand_name'];
$item_id=$row_brand['item_id'];

$select_item2=mysqli_query($conn,"select item_name from item where id='$item_id'");
while($row_items=mysqli_fetch_array($select_item2)){
  $item_id=$row_items['item_name'];
}

$total_stock=$row_brand['stock'];
$balance_stock=$row_brand['balance_stock'];
$date = date('d-m-Y', strtotime($row_brand['created_datetime']));
?>
<tr>

<td class="d-none"><?=$SNo; ?></td>
<td ><?=$date; ?></td>
<td><?=$brand_name; ?></td>
<td><?= $item_id; ?></td>
<td><?=$total_stock; ?></td>
<td><?=$balance_stock; ?>

</tr>
<? } ?>
</tbody>
</table>
```

```
</div>
</div>
</div>
<? } else { echo "No Records Found";  } ?>

<div class="modal fade" id="exampleExtraLargeModal" tabindex="-1" aria-hidden="true">
<div class="modal-dialog modal-x">
<div class="modal-content">
<div class="modal-header">
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
</div>
<div class="modal-body">
<div class="card border-top border-0 border-4 border-primary">
<div class="card-body p-5">

<div id="output"></div>
</div>
</div>
</div>

</div>
</div>
</div>




<script>
function getedit(val){

$.ajax({
url: "ajax-modal.php",
type: "POST",
```

```
data: "id="+val+"&act=stock",
success: function(result){
$("#output").html(result);
}});
}

function getproduct(val){
$.ajax({
url: "ajax-dropdown.php",
type: "POST",
data: "brand_name="+val,
success: function(result){
$("#result").html(result);
}});
}

</script>
<?php
}
include 'template.php';
?>
```

Item Page

```
<?php
function main() {
extract($_REQUEST);
include 'dilg/cnt/join.php';
include 'global-functions.php';
$ID=$_GET['id'];
$type1 = $_GET['type'];
date_default_timezone_set('Asia/Kolkata');
$currentTime = date('Y-m-d H:i:s');if($type1!=''){
  $subquery=" and type='".$type1."'";
}
if($Submit=='Add')
{if($type == '1'){
$subqry=" id='11' ";}else{
  $subqry=" id='$brand_name' ";
}
$slect_srev=mysqli_query($conn,"select * from brand where $subqry");
$row_serv=mysqli_fetch_array($slect_srev);$brand_name1=$row_serv['brand_name'];
$brand_name=$row_serv['id']; $variant=$variant_name." $variant_type ";
if($type!="1"){ $fitem_name=$item_name." - ". $variant;
}else{$fitem_name=$item_name;}$select_item=mysqli_query($conn,"select * from item where
item_name='$fitem_name'");if(mysqli_num_rows($select_item)==0){
$insert_item=mysqli_query($conn,"insert into item set
brand_name='$brand_name1',brand_id='$brand_name',item_name='$fitem_name',item_price='$item
_price', type='$type',
variant_type='$variant_type',variant='$variant_name',stock_warning='$stock_warning', status =
'$status',  created_by = ".$_SESSION['UID']."', created_datetime = '$currentTime'");if($insert_item)
{
$msg = 'Item Details Added Successfully';
header('Location:manage-item.php?msg='.$msg);
```

```
}else
{
$alert_msg = 'Could not able to add try once again!!!';
  header('Location:manage-item.php?alert_msg='.$alert_msg);
}
}else{
   $alert_msg = 'Item Name is Already there!!!';
    header('Location:manage-item.php?alert_msg='.$alert_msg);
}}
if($Submit=='Update')
{if($type == '1'){
$subqry=" id='11' ";}else{
   $subqry=" id='$brand_name' ";
}$slect_srev=mysqli_query($conn,"select * from brand where $subqry");
$row_serv=mysqli_fetch_array($slect_srev);$brand_name1=$row_serv['brand_name'];
$brand_name=$row_serv['id']; $variant=$variant_name." $variant_type ";if($type!="1"){
$fitem_name=$item_name." - ". $variant; }else{$fitem_name=$item_name;}

$select_item=mysqli_query($conn,"select * from item where item_name='$fitem_name' and
id!='$MainId'");if(mysqli_num_rows($select_item)==0){
   $update_customer=mysqli_query($conn,"update item set
brand_name='$brand_name1',brand_id='$brand_name',item_name='$fitem_name',item_price='$item
_price',variant_type='$variant_type',variant='$variant_name',stock_warning='$stock_warning', status
= '$status', modified_by= ".$_SESSION['UID']."', modified_datetime ='$currentTime'  where
id='$MainId' ");
if($update_customer)
{
$msg = 'Item Details Updated Successfully';
header('Location:manage-item.php?msg='.$msg);
}
else
{
```

```
$alert_msg = 'Could not able to update try once again!!!';
  header('Location:manage-item.php?alert_msg='.$alert_msg);
}
}else{
    $alert_msg = 'Item Name is Already there!!!';
   header('Location:manage-item.php?alert_msg='.$alert_msg);
}
}if($act=='delete' && $ID>0)
{
$customer_DeleteValues = mysqli_query($conn,"delete from item where id ='$ID' ");
if($customer_DeleteValues)
{
$alert_msg = 'Item Details Deleted Successfully';
header('Location:manage-item.php?alert_msg='.$alert_msg);
}
else
{
$alert_msg = 'Could not able to delete try once again!!!';
header('Location:manage-item.php?alert_msg='.$alert_msg);
}
}
if($_POST['act']=='ust')
{
   ob_clean();
   if($id != '' && $status != '')
   {
      $rs_UpdReg = mysqli_query($conn,"update item set status = '$status' where id = '$id'");
   }
   ?>
<?
    $rs_SelReg = mysqli_query($conn,"select * from item where id = '$id'");
   if(mysqli_num_rows($rs_SelReg)>0)
```

```
  {
    $rows_Reg = mysqli_fetch_array($rs_SelReg);
  }
  ?>
<script>drwStatus('<?=$rows_Reg['id']?>', '<?=$rows_Reg['status']?>')</script>
<?
  exit();
}
?>
<script language="javascript">function chStatus(id,st){
        $.ajax({
        url:'manage-item.php',
        data:'act=ust&id='+id+"&status="+st,
        type:'POST',
        success:function(data){
        drwStatus(id, st)
        }
      });
}</script>
 <script>
function drwStatus(id, St){   if(St=='1'){

    document.getElementById("spSt"+id).innerHTML = '<span style="cursor:pointer"
onclick="chStatus(\"+id+'\',\'0\')" title="Click To Change Active" class="btn btn-success padx-5
radius-30">Active</span>';
  }
  else{
     document.getElementById("spSt"+id).innerHTML = '<span style="cursor:pointer"
onclick="chStatus(\"+id+'\',\'1\')" title="Click To change Inactive" class="btn btn-danger padx-5
radius-30">Inactive</span>';
  }
}
```

```
 </script>
<div class="page-breadcrumb d-none d-sm-flex align-items-center mb-3">
  <h5 class="mb-0 text-dark">Manage Item</h5><div class="ms-auto">
<div class="col">
<!-- Button trigger modal -->
<button type="button" class="btn btn-primary" data-bs-toggle="modal"
data-bs-target="#exampleExtraLargeModal" onClick="getedit(0)">Add Item</button>
</div>
</div>
</div>
<hr/><?
    $select_item = mysqli_query($conn, "select * from item where 1=1 $subquery order by id
desc");if(mysqli_num_rows($select_item)>>0){
?>
<? if($msg !=''){ ?><div class="alert alert-success border-0 bg-success alert-dismissible fade show
py-2">
<div class="d-flex align-items-center">
<div class="font-35 text-white"><i class="bx bxs-check-circle"></i>
</div>
<div class="ms-3">
<h6 class="mb-0 text-white">Success Alerts</h6>
<div class="text-white"><?=$msg; ?></div>
</div>
</div>
<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
</div> <? } ?>
<? if($alert_msg !=''){ ?> <div class="alert alert-danger border-0 bg-danger alert-dismissible fade
show py-2">
<div class="d-flex align-items-center">
<div class="font-35 text-white"><i class="bx bxs-message-square-x"></i>
</div>
<div class="ms-3">
```

```
<h6 class="mb-0 text-white">Alerts</h6>
<div class="text-white"><?=$alert_msg; ?></div>
</div>
</div>
<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
</div> <? } ?>
<div class="card">
<div class="card-body">
<div class="table-responsive">
<table id="example2" class="table table-striped table-bordered" style="width:100%">
<thead>
<tr>
<!-- <th>Date</th> -->
<th class="d-none">SNo</th>
<th>Brand Name</th>
<th>Item Name</th>
<th>Rate</th>
<th>Status</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<?
$SNo = 0;
while($row_brand=mysqli_fetch_array($select_item))
{
$SNo = $SNo + 1;
$id=$row_brand['id'];
$brand_name=$row_brand['brand_name'];
$item_name=$row_brand['item_name'];
$item_rate=$row_brand['item_price'];
$type=$row_brand['type'];$status=$row_brand['status'];?>
```

```
<tr><td class="d-none"><?=$SNo; ?></td>
<td><?=$brand_name; ?></td>
<td><?=$item_name; ?></td><td>
<?=$item_rate; ?></td><td><div id="spSt<?=$id?>"></div>
        <script>drwStatus('<?=$id?>', '<?=$status?>')</script></td>
<td>
<div class="d-flex order-actions">
 <a href="#" class="btn btn-add btn-sm" tooltip="Edit" data-bs-toggle="modal"
data-bs-target="#exampleExtraLargeModal" onClick="getedit(<?=$id; ?>)"><i class="bx
bxs-edit"></i></a><a href="#" class="ms-3" data-toggle="modal" tooltip="Delete"
data-target="#customer2" onClick="if(confirm('Are you sure want to delete this?')) {
window.location.href='manage-item.php?act=delete&id=<?=$id ?> ' }"><i class="bx
bxs-trash"></i></a>
</div></td></tr><? } ?>
</tbody>
</table>
</div>
</div>
</div>
<? } else { echo "No Records Found";  } ?><div class="modal fade" id="exampleExtraLargeModal"
tabindex="-1" aria-hidden="true">
<div class="modal-dialog modal-x">
<div class="modal-content">
<div class="modal-header">
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
</div>
<div class="modal-body">
<div class="card border-top border-0 border-4 border-primary">
<div class="card-body p-5"><div id="output"></div>
</div>
</div>
</div></div>
```

```
</div>
</div><script>
function gettype(val){if(val==0){
  $(".service-grid").show();
$(".service-grid").removeClass("d-none");
$("#stock_warning").attr("required", "required");
$("#brands").attr("required", "required");
$("#variant_type").attr("required", "required");
}else{
  $(".service-grid").hide();
$(".service-grid").addClass("d-none");
$("#stock_warning").removeAttr("required");
$("#brands").removeAttr("required");
$("#variant_type").removeAttr("required");
}
}function getproduct1(val){
$.ajax({
url: "ajax-get-variant.php",
type: "POST",
data: "variant_type="+val,
success: function(result){
$("#variant_name").html(result);
}});}function getedit(val){$.ajax({url: "ajax-modal.php", type: "POST",data:
"id="+val+"&act=item",
success: function(result){
$("#output").html(result);
}});
}</script>
<?php
}
include 'template.php';
?>
```

Header Page

```
<header>
<div class="topbar d-flex align-items-center">
<nav class="navbar navbar-expand">
<div class="mobile-toggle-menu"><i class='bx bx-menu'></i>
</div>

<?
if($Pagename == 'view-task.php'){
$UserMobileIds = $_GET['id'];
$select_usermobile=mysqli_query($conn,"select * from  registration where phone_number =
'$UserMobileIds' ");
$row_usermobile=mysqli_fetch_array($select_usermobile)
?>
<div class="txt-cent" id="submit" name="smt"  style=" margin-left: 10px;" ><a href="task-user.php"
class="btn btn-danger" style="margin-top: 10px;">Back</a>
</div>
<p class="text-task"><?=$row_usermobile['name']; ?> <span class="md-none"> -
<?=$row_usermobile['phone_number']; ?></span> - <?=$row_usermobile['team']; ?></p>
<? } ?>

<div class="top-menu ms-auto">
<ul class="navbar-nav align-items-center">
<li class="nav-item dropdown dropdown-large">
<div class="dropdown-menu dropdown-menu-end">
<div class="header-notifications-list">
</div>
</div>
</li>
<li class="nav-item dropdown dropdown-large">
<div class="dropdown-menu dropdown-menu-end">
```

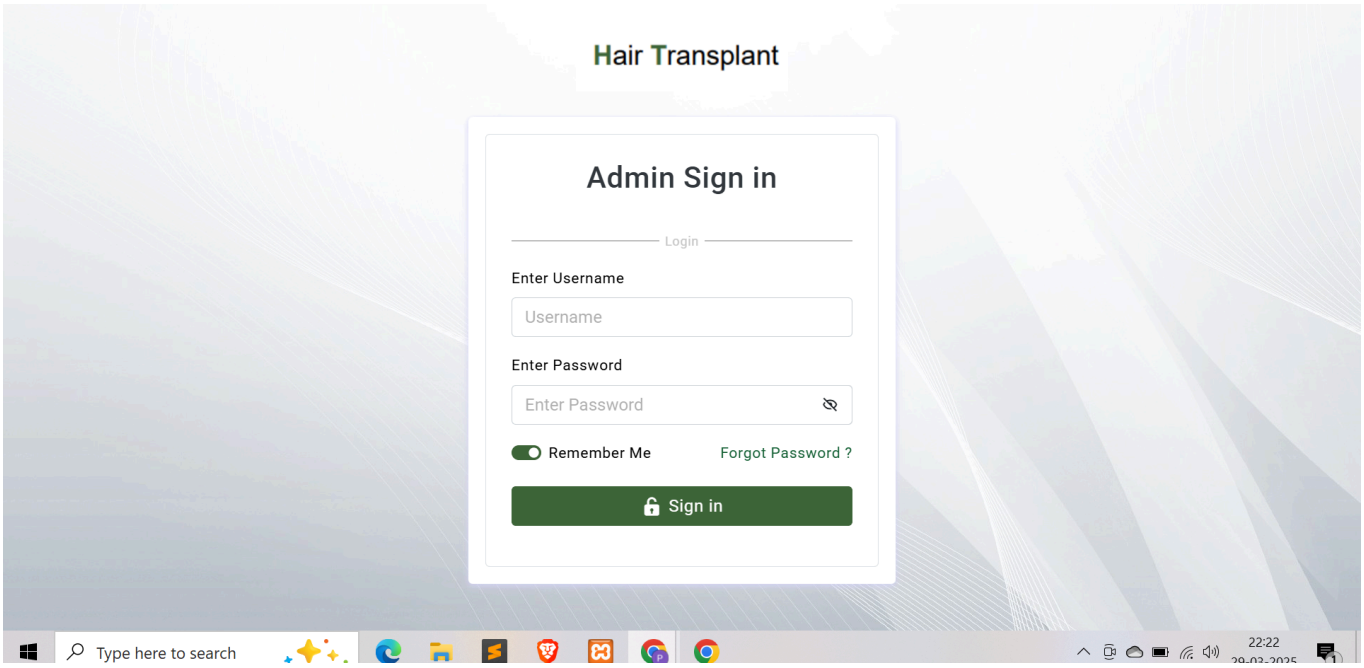
```
<div class="header-message-list">
</div>
</div>
</li>
</ul>
</div>
<div class="user-box dropdown">
<a class="d-flex align-items-center nav-link dropdown-toggle dropdown-toggle-nocaret" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false">
<!-- <img src="assets/images/avatars/avatar-2.png" class="user-img" alt="user avatar"> -->
<div class="user-info ps-3">

<p class="user-name mb-0">Super Admin</p>

<!-- <p class="designattion mb-0">Web Designer</p> -->
</div>
</a>
<ul class="dropdown-menu dropdown-menu-end">
<!-- <li><a class="dropdown-item" href="home.php"><i class='bx
bx-home-circle'></i><span>Dashboard</span></a>
</li> -->

<li><a class="dropdown-item" href="logout.php"><i class='bx
bx-log-out-circle'></i><span>Logout</span></a>
</li>
</ul>
</div>
</nav>
</div>
</header>
```

Footer Page

```
        <div class="overlay toggle-icon"></div>
                <!--end overlay-->
                <!--Start Back To Top Button--> <a href="javaScript:;" class="back-to-top"><i class='bx
bxs-up-arrow-alt'></i></a>
                <!--End Back To Top Button-->
                <footer class="page-footer">
                        <p class="mb-0">Copyright © 2025. All right reserved.</p>
                </footer>
```

5.2 SCREENSHOTS

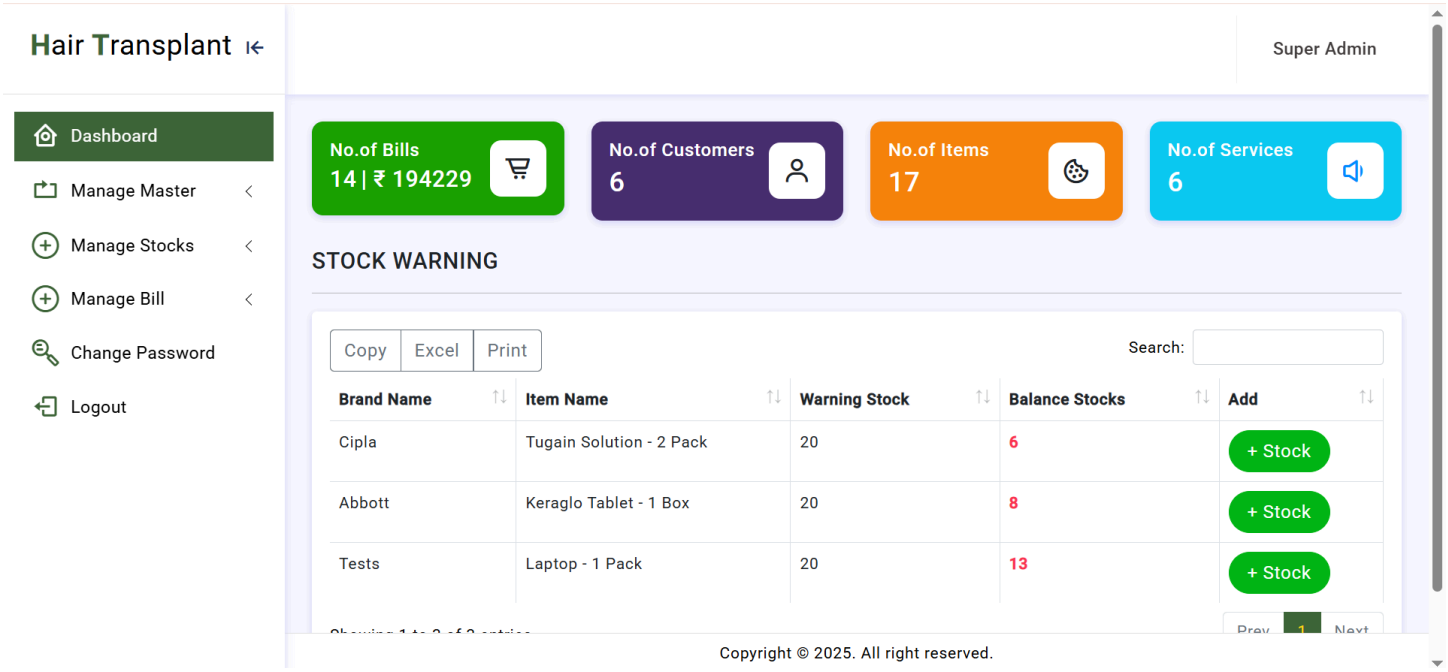
Login Form



The Login module is designed to establish the user identity with the credentials provided to them. This enables only the authenticated user to use the system. Hence the data in the system remains secret and not known to all. Only persons with the proper access privileges can view the sensitive data in the system.

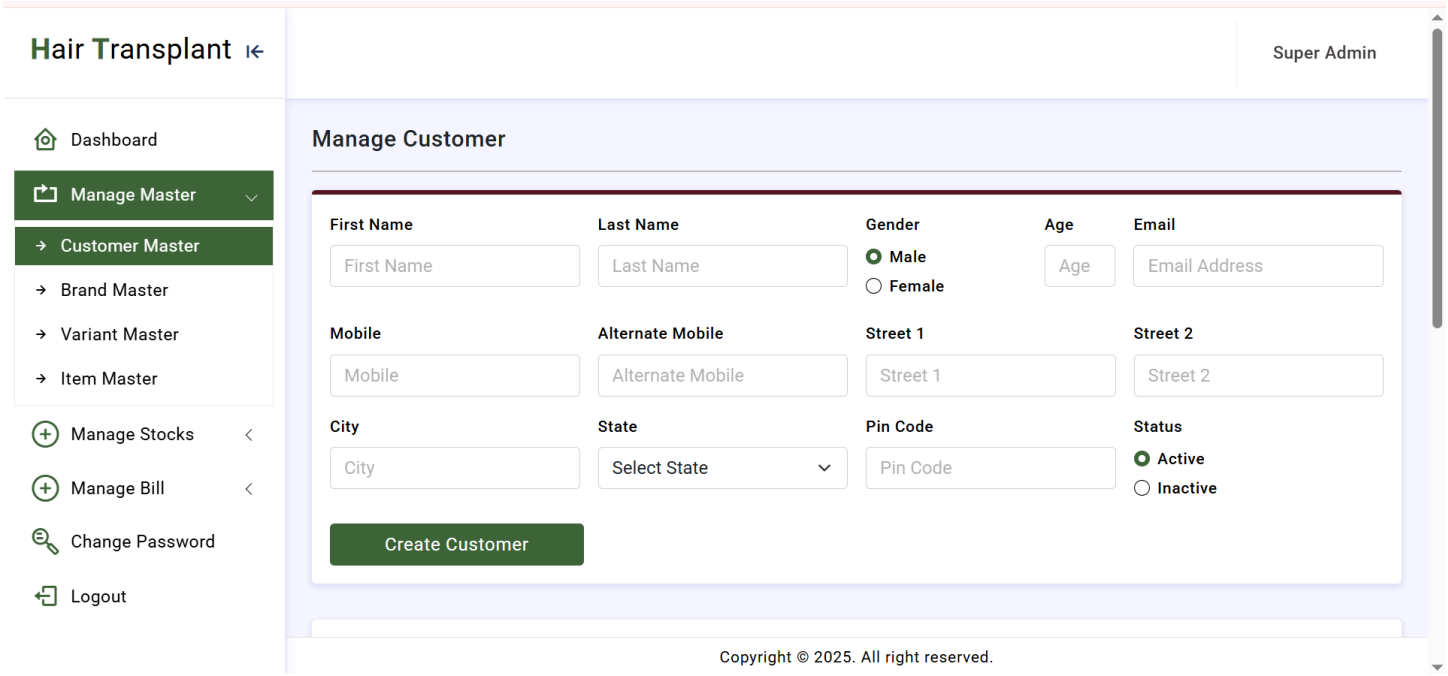
The page where the admin users will log in their system credentials to access the system. When the username and password match with the database, the next form will be displayed according to the role.

Dashboard



The dashboard of the Medical Billing System provides a centralized and real-time overview of key business metrics, ensuring efficient management and monitoring of essential operations. It displays important statistics such as the total number of bills generated, the number of registered customers, available items in stock, and the number of services provided. This allows the admin to quickly assess the business's performance and track daily transactions with ease. The user-friendly interface ensures that all critical information is accessible at a glance, reducing the need for manual record-keeping and improving decision-making.

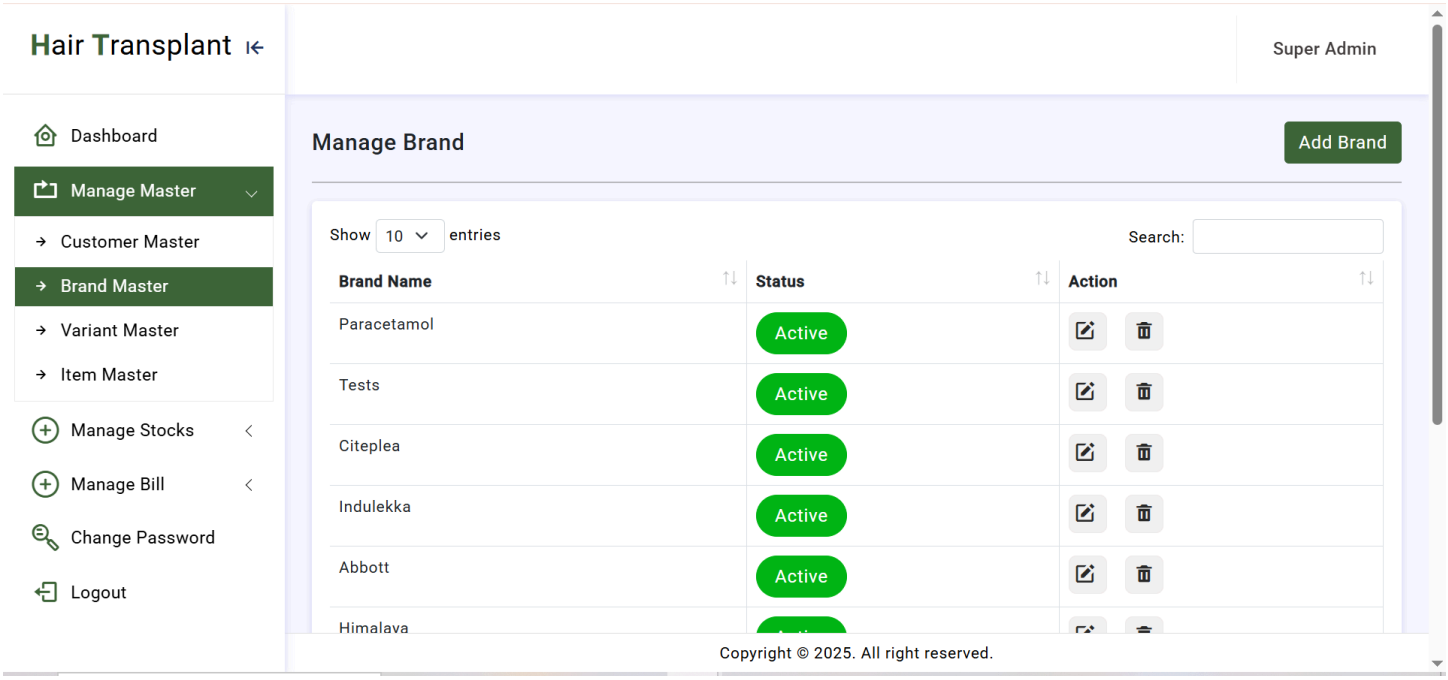
Manage Customer



The Manage Customer module is designed to store and organize customer information efficiently, ensuring seamless billing and service management. This module allows the admin to add new customers by entering their essential details, such as name, address, mobile number, email, and profile image.

Keeping a structured record of customers helps in generating accurate bills and maintaining a smooth transaction history.

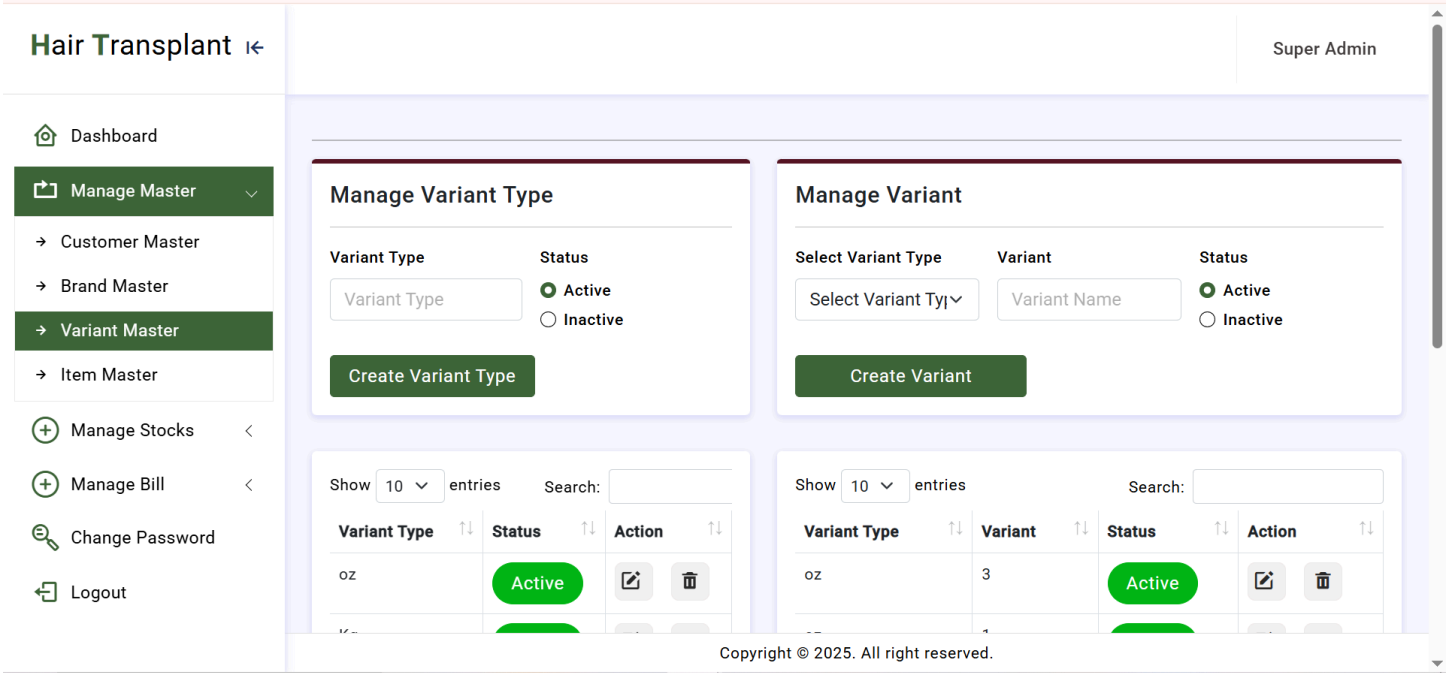
Manage Brand



The Brand Management module allows the admin to create, update, and manage different brands of medicines available in the system. This module ensures that products are categorized under their respective brands, making it easier to organize inventory and streamline the billing process.

The brand page displays a list of all registered brands along with their status, indicating whether they are active or inactive.

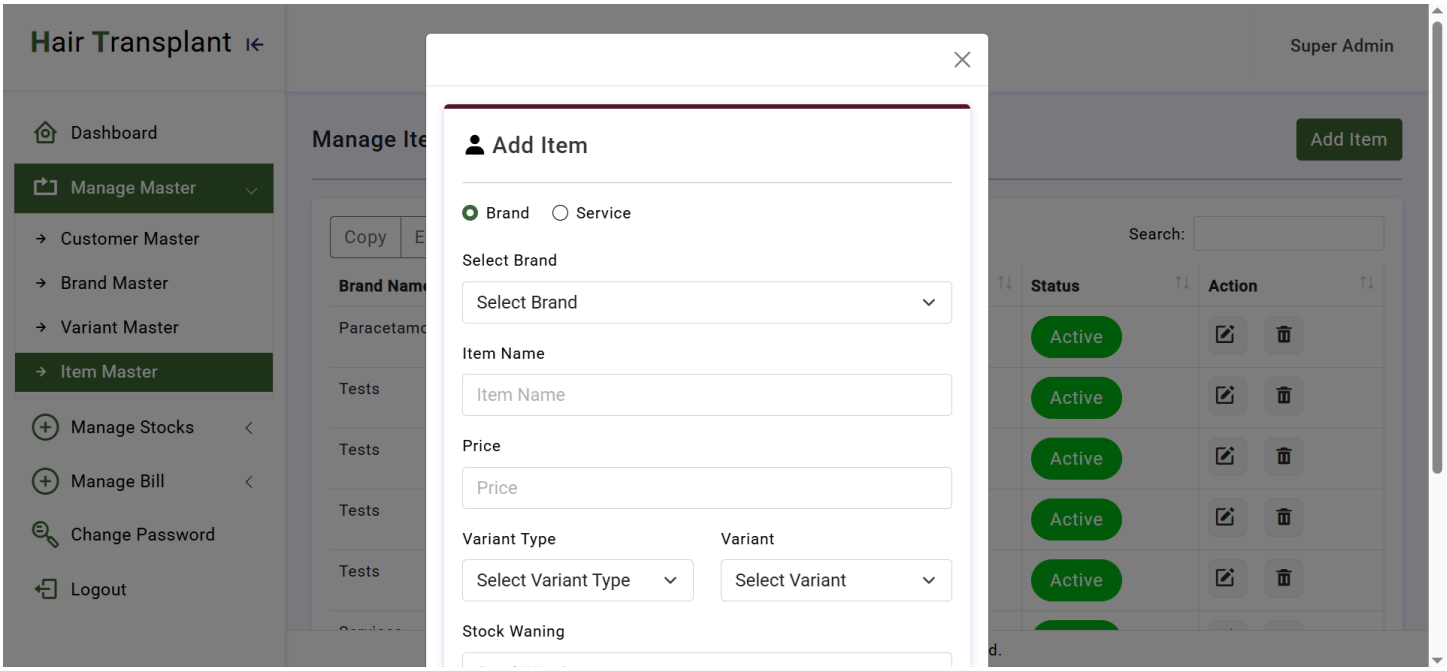
Manage Variant



The Manage Variant module enables the admin to create and manage different product variants, ensuring flexibility in inventory classification. Variants refer to different forms or measurements of a product, such as packaging sizes, weights, or dosages. This module allows the admin to define and organize product variations systematically, making stock management more efficient.

The variant page provides options to select a variant type and add specific variant names, such as different weights (e.g., 1kg, 2kg) or volumes (e.g., 100ml, 500ml).

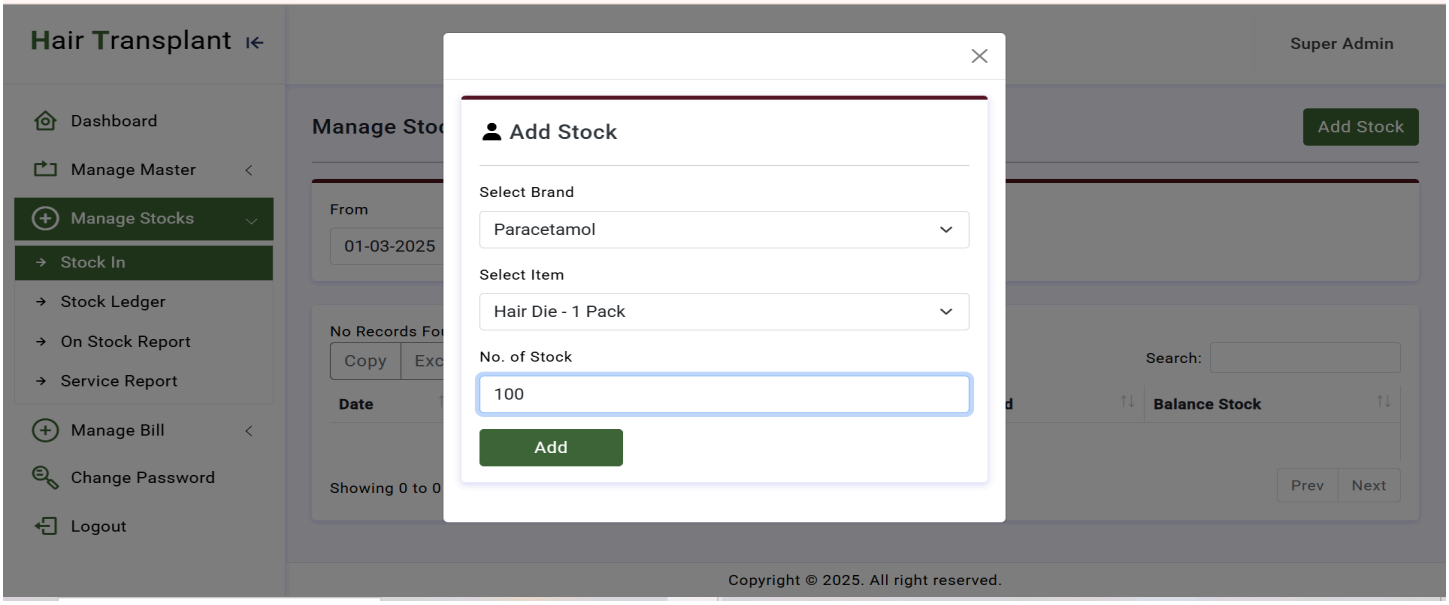
Manage Item



The Manage Item module is designed to efficiently handle product inventory by allowing the admin to add, update, and manage items available for billing. Each item is linked to a specific brand, ensuring proper categorization and easy retrieval during transactions.

On the item management page, the admin can view a list of items along with their associated brand name, item name, rate, and status.

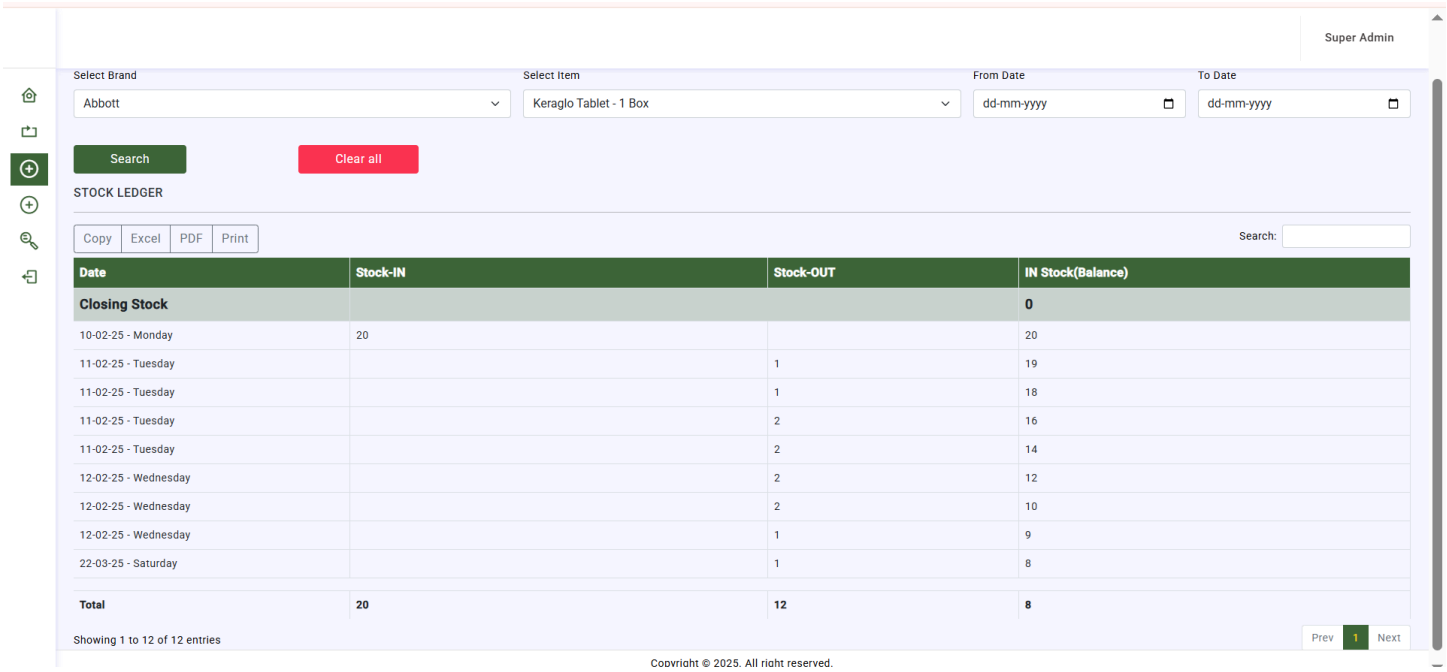
Manage Stock



The Stock Management module ensures efficient tracking and management of product inventory by allowing the admin to add and update stock levels. This module plays a crucial role in maintaining adequate product availability and preventing shortages.

On the stock management page, the admin can select a brand and then choose a specific item to update its stock quantity.

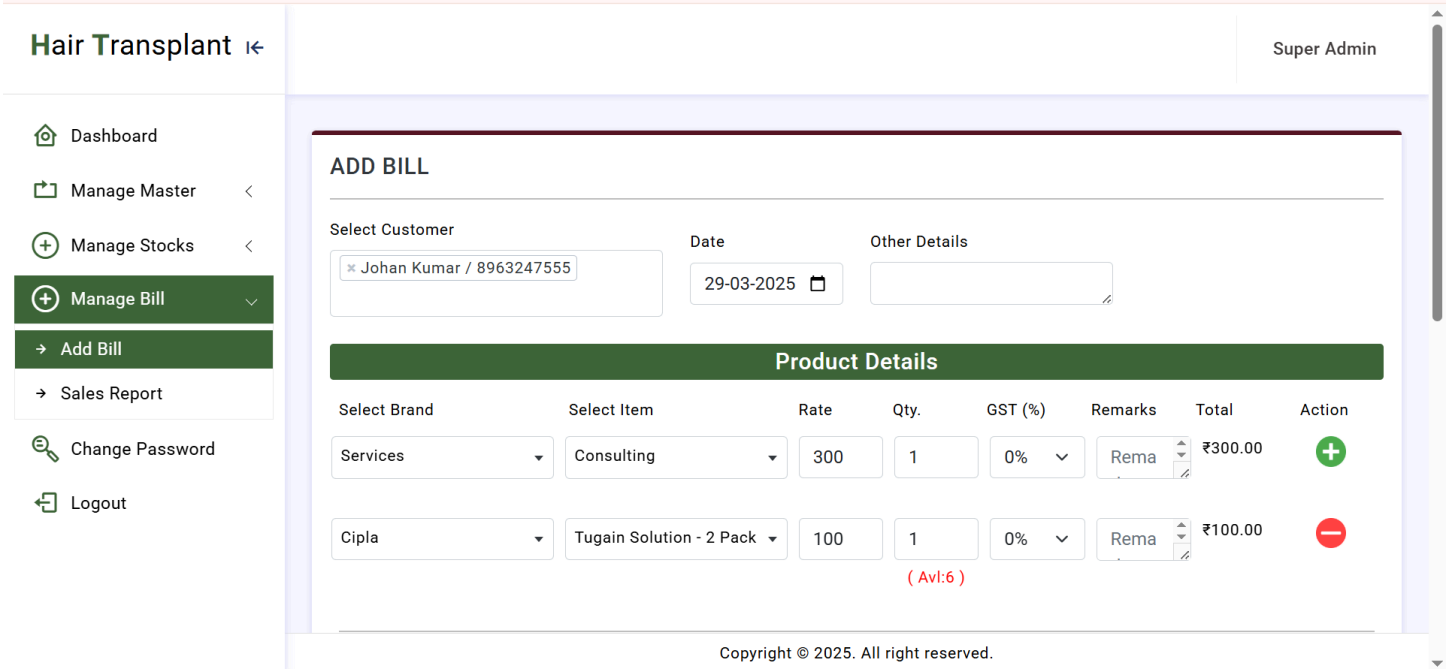
Stock Ledger



The Stock Ledger module provides a detailed view of stock movements, allowing the admin to track inventory inflow and outflow efficiently. This module helps in maintaining accurate records of stock transactions, ensuring proper inventory management.

The stock ledger displays date-wise entries along with the quantity of stock-in (added stock) and stock-out (sold or used stock).

Add Bill

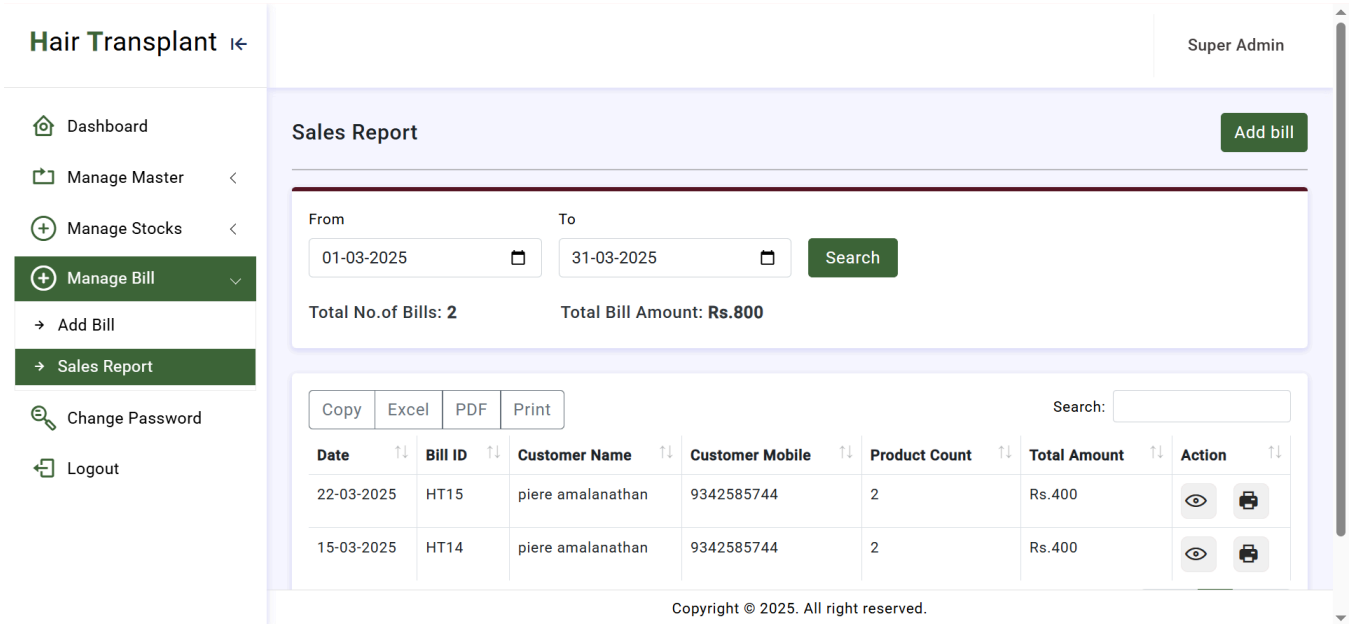


The Add Bill module allows the admin to generate and manage customer invoices efficiently. This module streamlines the billing process by enabling the selection of customer details, adding product information, and calculating totals in a structured manner.

The billing page includes fields to select a customer from the registered database, ensuring accurate billing records. The date field automatically captures the current date of the transaction. Additional fields allow entry of other details related to the bill.

In the Product Details section, the admin can select a brand and item, specify the rate and quantity, and apply the GST percentage if applicable.

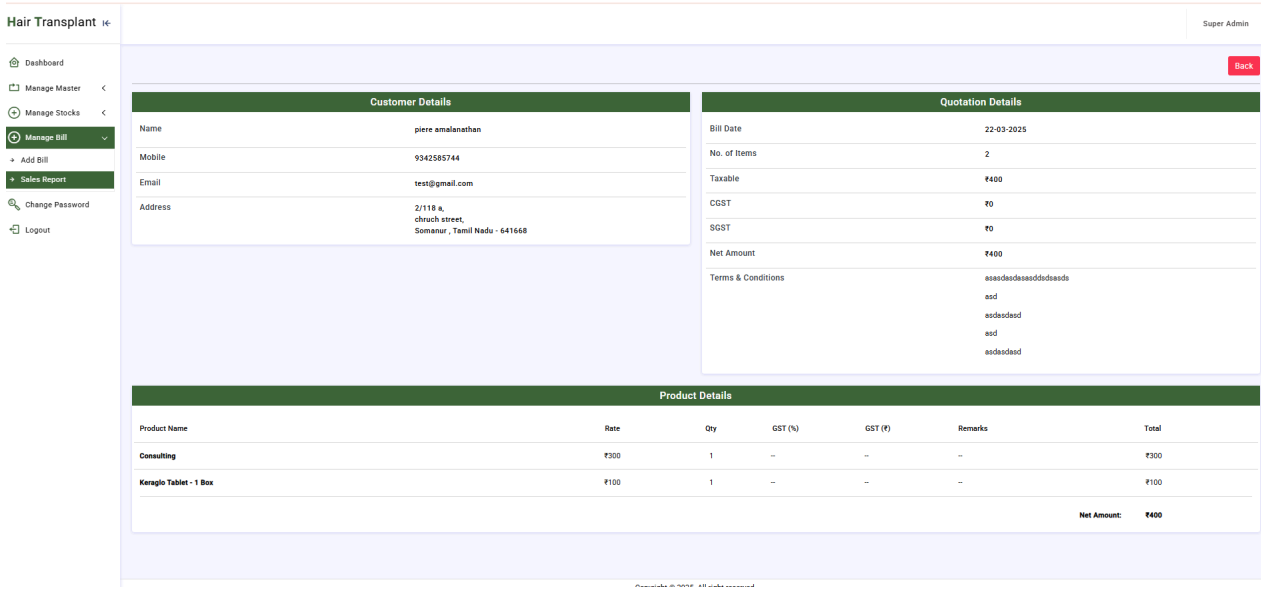
Sales Report



The Sales Report module provides a comprehensive summary of all billing transactions within a specified date range. This module helps in tracking sales performance, analyzing revenue trends, and maintaining financial records efficiently.

The report allows the admin to filter sales data by selecting a start date and end date, ensuring flexibility in generating reports for a specific period. It displays key details such as total number of bills generated and the total bill amount for the selected timeframe.

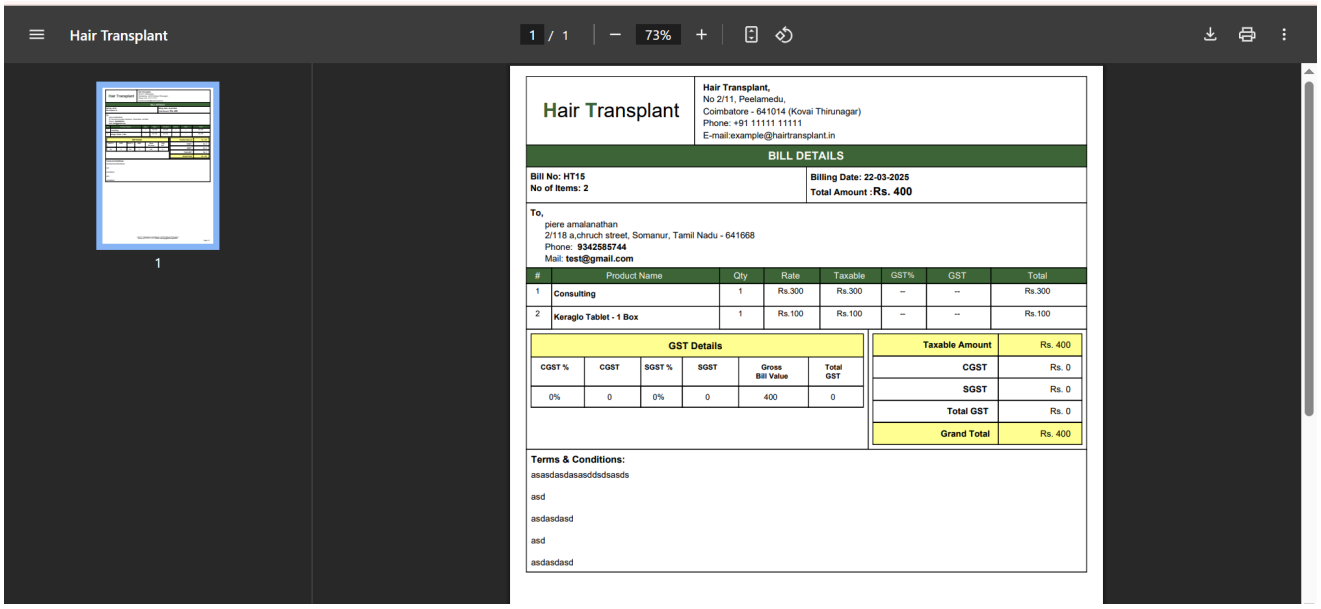
View Bill



The View Bill module allows the admin to access and review previously generated invoices, ensuring easy tracking of customer transactions. This module provides a detailed breakdown of each bill, helping in verifying sales records, handling customer queries, and managing financial reports.

On the view bill page, the admin can see all invoices along with essential details such as bill date, bill ID, customer name, mobile number, product details, total amount, and remarks. The system ensures that each bill can be easily retrieved and reviewed as needed.

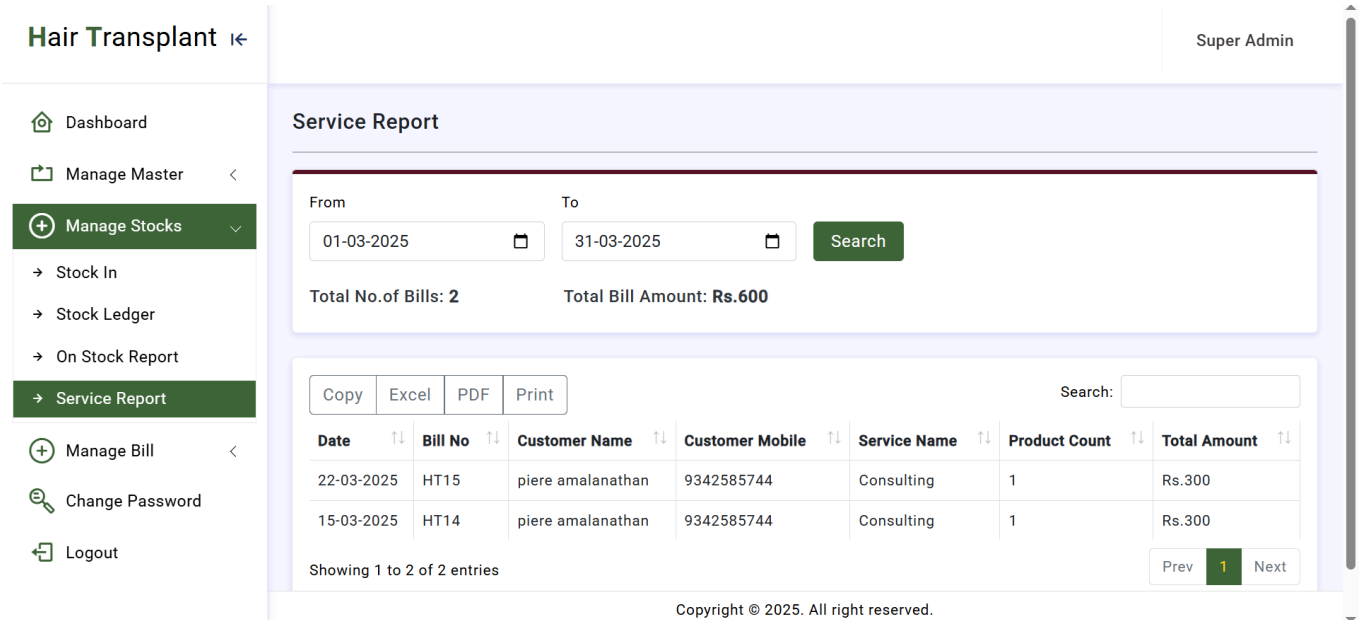
Generate PDF



The Generate PDF module allows the admin to create and download invoices in a structured and professional format. This feature ensures that bills can be easily shared, printed, and stored for future reference.

On the Generate PDF page, the system retrieves all relevant billing details, including bill ID, date, customer name, mobile number, product details, quantity, rate, total amount, and remarks. The invoice is formatted neatly, ensuring clarity and accuracy.

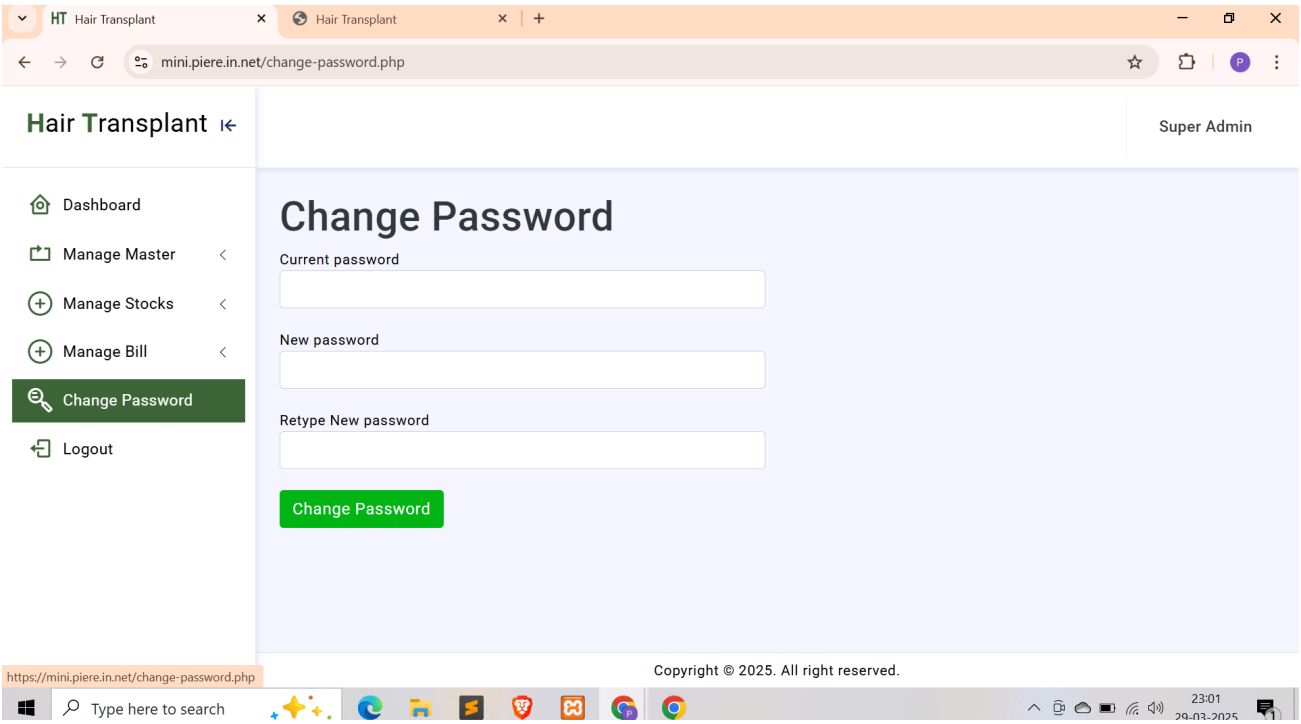
Service Report



The Service Report module provides a detailed summary of all service-related transactions within a specified period. This feature helps in tracking the number of services provided, analyzing customer interactions, and maintaining accurate service records.

On the Service Report page, the admin can filter data by selecting a start date and end date to generate reports for a specific timeframe. The report displays key details such as service date, service ID, customer name, mobile number, service type, status, and total charges.

Change Password



The Change Password module allows the admin to update their login credentials securely. This feature enhances system security by enabling password updates at regular intervals, preventing unauthorized access.

On the Change Password page, the admin must enter the current password followed by the new password and confirm it to ensure accuracy. The system validates the input to meet security requirements, such as minimum length and complexity.

CONCLUSION

6. CONCLUSION

The development of the Medical Billing System has streamlined the process of managing customer details, inventory, and billing operations. By replacing the manual system with an automated solution, the project enhances accuracy, efficiency, and data security. The system ensures smooth invoice generation, stock management, and customer record-keeping, reducing human errors and improving overall business operations.

With its user-friendly interface and well-structured database, the system provides a reliable and efficient solution for medical billing needs. As businesses grow, future enhancements can further improve functionality, ensuring adaptability and scalability. This system not only simplifies daily operations but also contributes to a more organized and effective billing process.

As technology evolves, this system can be further enhanced with features such as cloud integration, real-time analytics, and automated notifications to improve workflow and decision-making. With its robust framework and user-friendly interface, the **Medical Billing System** lays the groundwork for a more efficient, scalable, and modernized billing process.

FUTURE ENHANCEMENTS

7. FUTURE ENHANCEMENTS

The Medical Billing System has been designed with scalability in mind, allowing for future enhancements to improve functionality and efficiency. Some potential upgrades include:

- ❖ Multi-User Access: Expanding the system to allow multiple users with different roles, such as pharmacists, accountants, and managers, to streamline operations.
- ❖ Cloud Integration: Storing data on a secure cloud server to enable remote access and ensure data backup and security.
- ❖ Advanced Reporting: Implementing detailed sales and stock reports with graphical analysis to assist in decision-making.
- ❖ Automated Notifications: Adding features for low-stock alerts, payment reminders, and bill confirmations via email or SMS.
- ❖ Barcode Scanning: Integrating barcode scanning for quick product selection and billing, reducing manual data entry errors.
- ❖ Mobile Compatibility: Developing a mobile-friendly version or app to allow real-time billing and stock management on the go.

BIBLIOGRAPHY

8. BIBLIOGRAPHY

Reference Websites

1. https://www.w3schools.com/php/
2. https://www.tutorialspoint.com/sql/index.htm
3. www.geeksforgeeks.org/php/
4. www.php.net
5. https://jquery.com/
6. https://getbootstrap.com/https://getbootstrap.com/
7. https://www.ajax-engg.com/https://www.ajax-engg.com/
8. https://chatgpt.com/
9. https://github.com/https://github.com/
10. http://codepen.io/