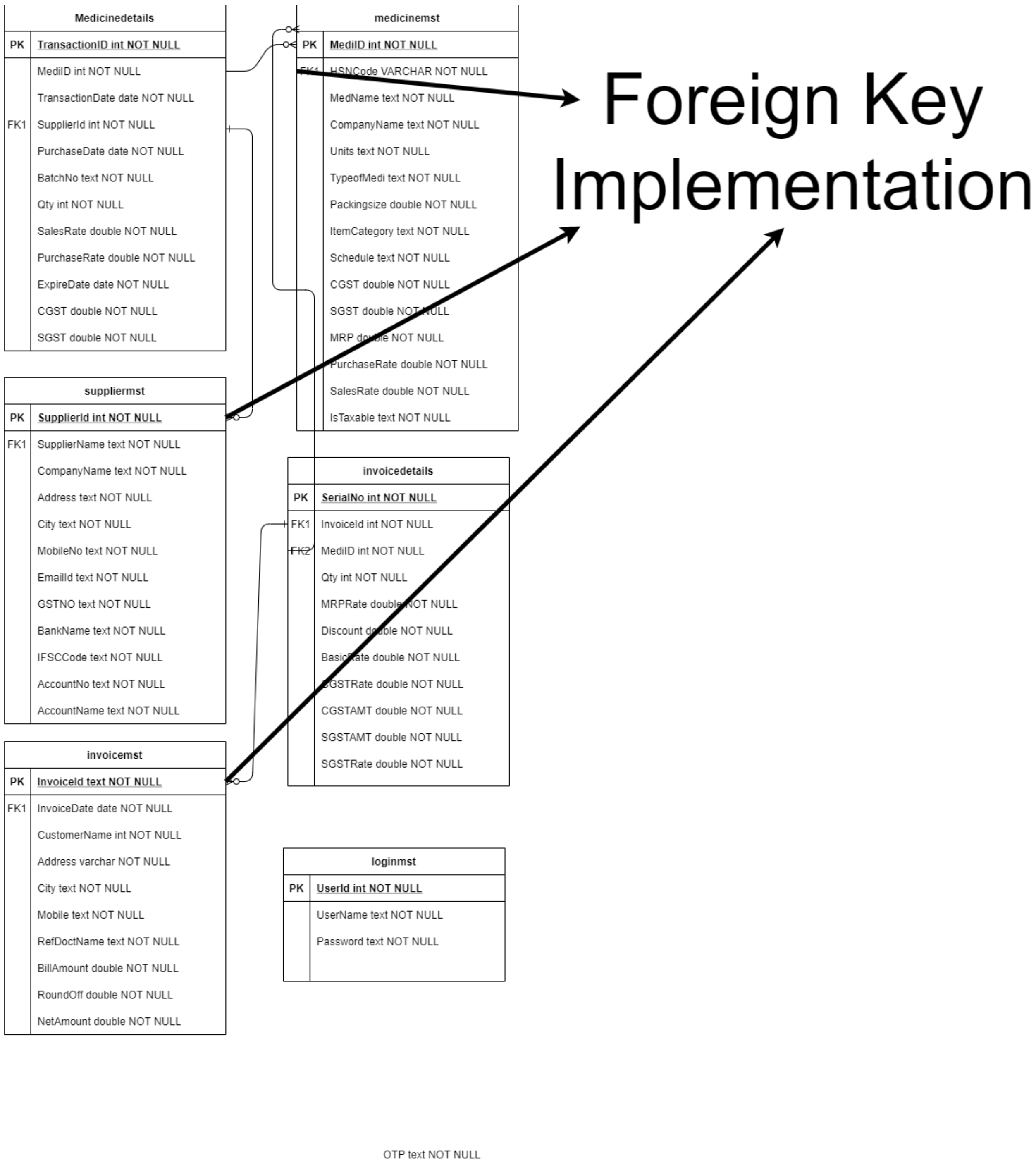
Computer Science- Criteria C: Product Design

Techniques used for development:

- Language Used: PHP, HTML and JavaScript
- Database Management System: PhpMyAdmin
- Primary and Foreign Key
- Login Session Authentication
- Database design methods(ER Diagram,functional dependency)
- Security of Database-PDO Authentication
- Use of JavaScript functions in Invoice
- Exception Handling



	Medicinedetails
PK	TransactionID int NOT NULL
	MediID int NOT NULL
	TransactionDate date NOT NULL
FK1	SupplierId int NOT NULL
	PurchaseDate date NOT NULL
	BatchNo text NOT NULL
	Qty int NOT NULL
	SalesRate double NOT NULL
	PurchaseRate double NOT NULL
	ExpireDate date NOT NULL
	CGST double NOT NULL
	SGST double NOT NULL

	medicinemst
PK	MediID int NOT NULL
FK1	HSNCode VARCHAR NOT NULL
	MedName text NOT NULL
	CompanyName text NOT NULL
	Units text NOT NULL
	TypeofMedi text NOT NULL
	Packingsize double NOT NULL
	ItemCategory text NOT NULL
	Schedule text NOT NULL
	CGST double NOT NULL
	SGST double NOT NULL
	MRP double NOT NULL
	PurchaseRate double NOT NULL
	SalesRate double NOT NULL
	IsTaxable text NOT NULL

	suppliermst
PK	SupplierId int NOT NULL
FK1	SupplierName text NOT NULL
	CompanyName text NOT NULL
	Address text NOT NULL
	City text NOT NULL
	MobileNo text NOT NULL
	EmailId text NOT NULL
	GSTNO text NOT NULL
	BankName text NOT NULL
	IFSCCode text NOT NULL
	AccountNo text NOT NULL
	AccountName text NOT NULL

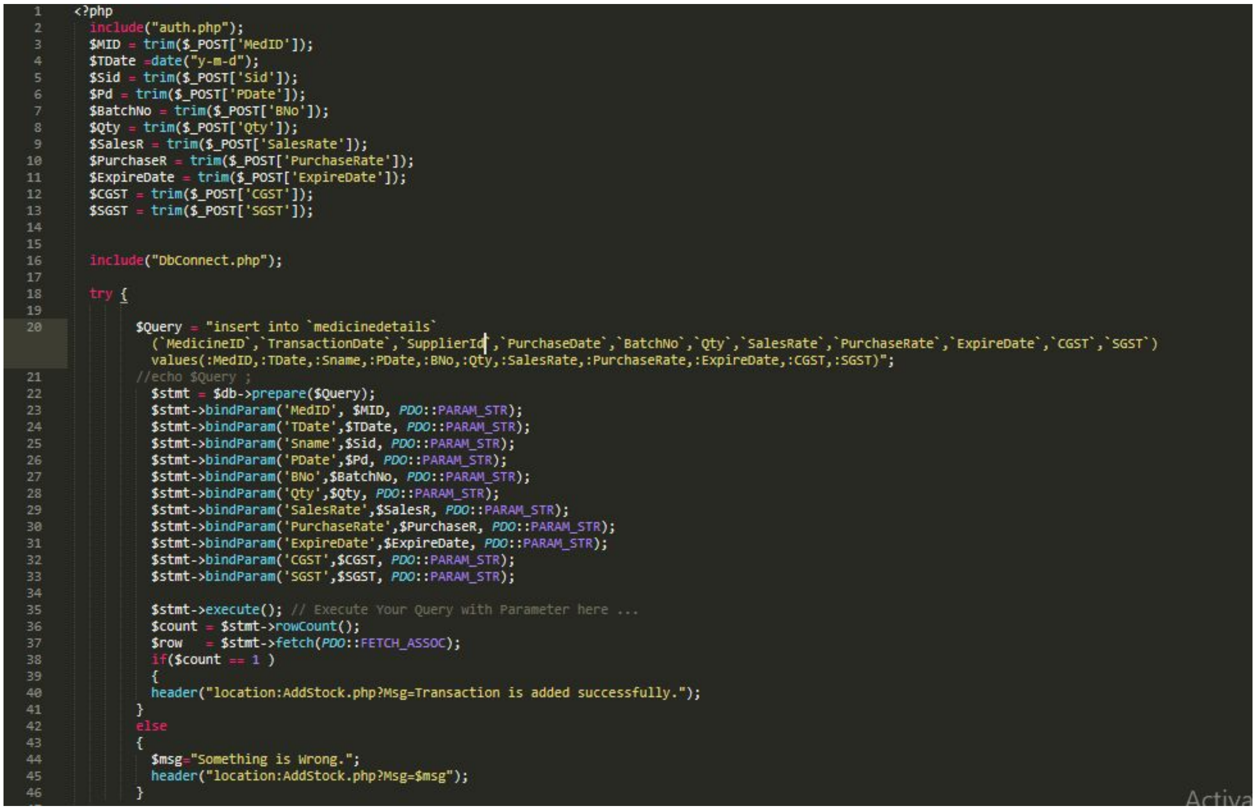
	invoicedetails
PK	SerialNo int NOT NULL
FK1	InvoiceId int NOT NULL
FK2	MediID int NOT NULL
	Qty int NOT NULL
	MRPRate double NOT NULL
	Discount double NOT NULL
	BasicRate double NOT NULL
	CGSTRate double NOT NULL
	CGSTAMT double NOT NULL
	SGSTAMT double NOT NULL
	SGSTRate double NOT NULL

	invoicemst
PK	InvoiceId text NOT NULL
FK1	InvoiceDate date NOT NULL
	CustomerName int NOT NULL
	Address varchar NOT NULL
	City text NOT NULL
	Mobile text NOT NULL
	RefDoctName text NOT NULL
	BillAmount double NOT NULL
	RoundOff double NOT NULL
	NetAmount double NOT NULL

	loginmst
PK	UserId int NOT NULL
	UserName text NOT NULL
	Password text NOT NULL

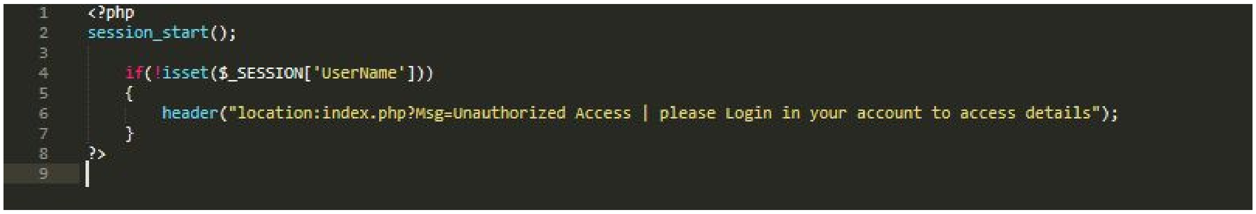
OTP text NOT NULL

1. **Foreign Key Implementation:** Foreign Keys are integral to database design. They extract data(usually Primary Keys) from other database tables and specifies the data field values in other tables.



```
<?php
  include("auth.php");
  $MID = trim($_POST['MedID']);
  $TDate =date("y-m-d");
  $Sid = trim($_POST['Sid']);
  $Pd = trim($_POST['PDate']);
  $BatchNo = trim($_POST['BNo']);
  $Qty = trim($_POST['Qty']);
  $SalesR = trim($_POST['SalesRate']);
  $PurchaseR = trim($_POST['PurchaseRate']);
  $ExpireDate = trim($_POST['ExpireDate']);
  $CGST = trim($_POST['CGST']);
  $SGST = trim($_POST['SGST']);


  include("DbConnect.php");

  try {

        $Query = "insert into `medicinedetails`
          (`MedicineID`,`TransactionDate`,`SupplierId`,`PurchaseDate`,`BatchNo`,`Qty`,`SalesRate`,`PurchaseRate`,`ExpireDate`,`CGST`,`SGST`)
          values(:MedID,:TDate,:Sname,:PDate,:BNo,:Qty,:SalesRate,:PurchaseRate,:ExpireDate,:CGST,:SGST)";
        //echo $Query ;
          $stmt = $db->prepare($Query);
          $stmt->bindParam('MedID', $MID, PDO::PARAM_STR);
          $stmt->bindParam('TDate',$TDate, PDO::PARAM_STR);
          $stmt->bindParam('Sname',$Sid, PDO::PARAM_STR);
          $stmt->bindParam('PDate',$Pd, PDO::PARAM_STR);
          $stmt->bindParam('BNo',$BatchNo, PDO::PARAM_STR);
          $stmt->bindParam('Qty',$Qty, PDO::PARAM_STR);
          $stmt->bindParam('SalesRate',$SalesR, PDO::PARAM_STR);
          $stmt->bindParam('PurchaseRate',$PurchaseR, PDO::PARAM_STR);
          $stmt->bindParam('ExpireDate',$ExpireDate, PDO::PARAM_STR);
          $stmt->bindParam('CGST',$CGST, PDO::PARAM_STR);
          $stmt->bindParam('SGST',$SGST, PDO::PARAM_STR);

          $stmt->execute(); // Execute Your Query with Parameter here ...
          $count = $stmt->rowCount();
          $row   = $stmt->fetch(PDO::FETCH_ASSOC);
          if($count == 1 )
          {
          header("location:AddStock.php?Msg=Transaction is added successfully.");
        }
        else
        {
          $msg="Something is Wrong.";
          header("location:AddStock.php?Msg=$msg");
        }
```

2. **Generating Queries:** The snippet above is an application of using queries to create database tables.bindParam is used to insert data fields in the database. PhpMyAdmin uses variations such as update, insert, drop and truncate to manage database records. Creating queries requires attention to detail with parameter names and variable identifiers should match with pre-existing fields. These queries are generated through inbuild functions in PHP which have to be learnt from online resources and guides.



```
<?php
session_start();

    if(!isset($_SESSION['UserName']))
    {
        header("location:index.php?Msg=Unauthorized Access | please Login in your account to access details");
    }
?>
```

3. **Login Page Authentication:** Using file “auth.php”, user requires to enter username and password in order to access the dashboard and all other files

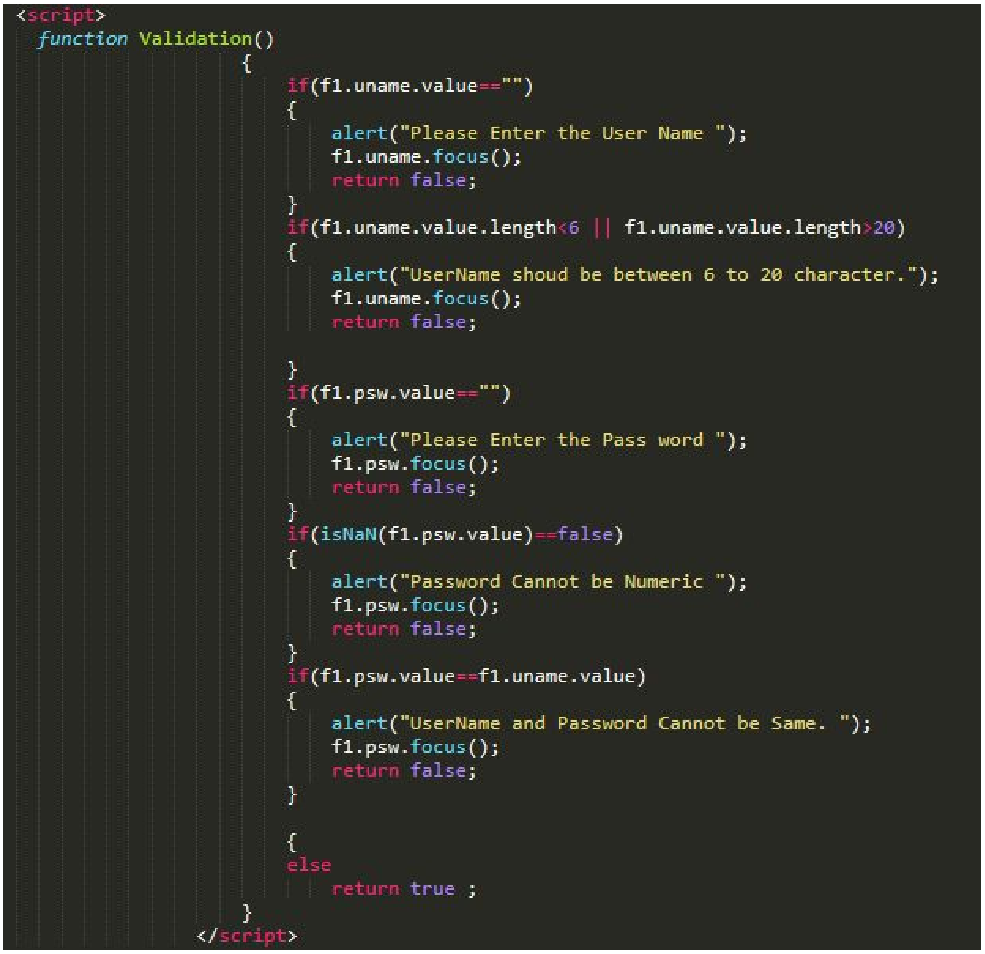
in the application. If an intruder tries to access the URL directly, then access is blocked.

Techniques used for minimizing errors during data entry:

Validation:

a.**Use of default values** to ensure a reduction in the number of errors. E.g the Tdate(Transaction Date) in AddStock defaults to DATE and enters date from the computer system.

b.**Restriction to data inputs:** Data inputs have a number of characters limit and also have appropriate validation e.g Password should be between 8 to 20 characters, Password and Confirm Password should match.

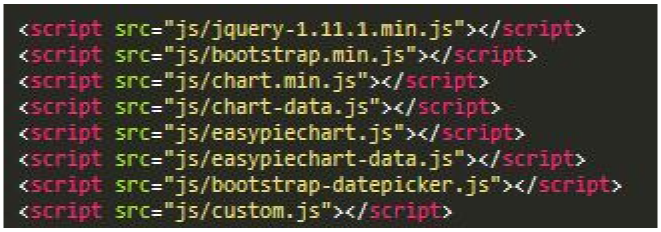


```
<script>
  function Validation()
                    {
                        if(f1.uname.value=="")
                        {
                            alert("Please Enter the User Name ");
                            f1.uname.focus();
                            return false;
                        }
                        if(f1.uname.value.length<6 || f1.uname.value.length>20)
                        {
                            alert("UserName shoud be between 6 to 20 character.");
                            f1.uname.focus();
                            return false;

                        }
                        if(f1.psw.value=="")
                        {
                            alert("Please Enter the Pass word ");
                            f1.psw.focus();
                            return false;
                        }
                        if(isNaN(f1.psw.value)==false)
                        {
                            alert("Password Cannot be Numeric ");
                            f1.psw.focus();
                            return false;
                        }
                        if(f1.psw.value==f1.uname.value)
                        {
                            alert("UserName and Password Cannot be Same. ");
                            f1.psw.focus();
                            return false;
                        }

                        {
                        else
                            return true ;
                    }
                </script>
```

c.**Appropriate data types:** The data types chosen in the database complement the data taken by users and reduces errors.

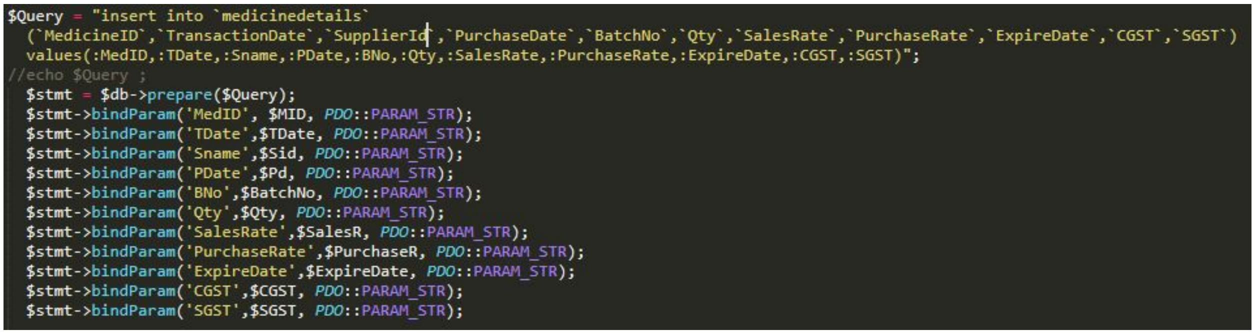
Techniques used for creating an application:

1. **Additional Libraries:** Certain js libraries were used such as:
    a. .jquery- used to create methods to execute certain tasks
    b. Bootstrap.min.js- introduces the CSS bootstrap to the program
    c. Bootstrap-datepicker.js-it adds the functionality of selecting time without the necessity of using custom JavaScript code.



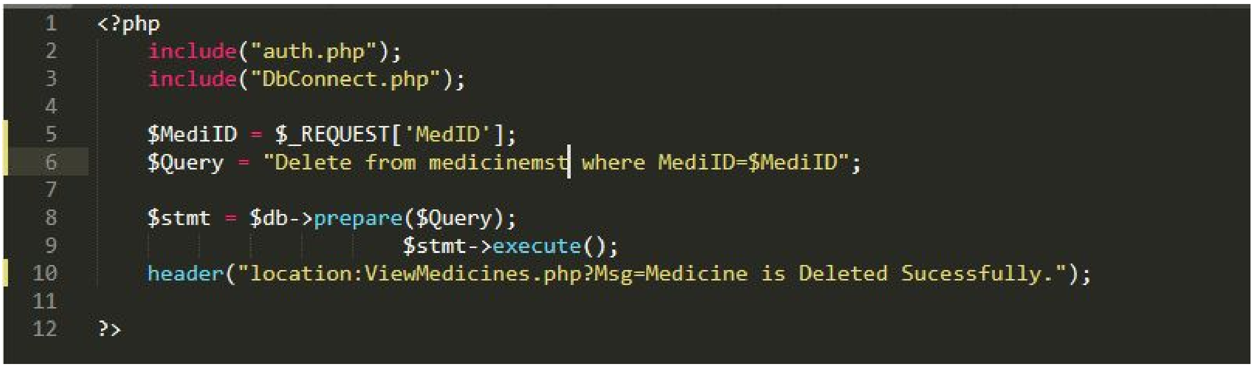
```
<script src="js/jquery-1.11.1.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/chart.min.js"></script>
<script src="js/chart-data.js"></script>
<script src="js/easypiechart.js"></script>
<script src="js/easypiechart-data.js"></script>
<script src="js/bootstrap-datepicker.js"></script>
<script src="js/custom.js"></script>
```

2. **Insert Data:** Data is added through text fields in PhpMyAdmin and $ variables as identifiers. Here, the data is being inserted into "medicinedetails" table using 'insert' command. The $Query is executed function prepare($Query).



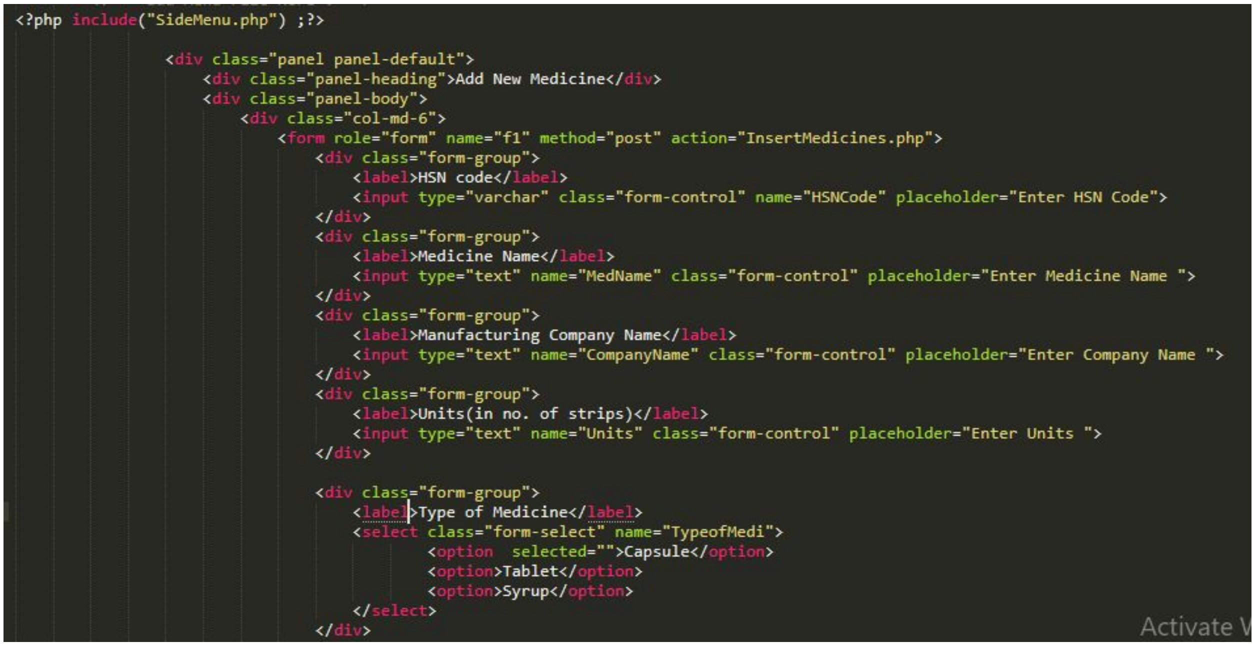
```
$Query = "insert into `medicinedetails`
  (`MedicineID`,`TransactionDate`,`SupplierId`,`PurchaseDate`,`BatchNo`,`Qty`,`SalesRate`,`PurchaseRate`,`ExpireDate`,`CGST`,`SGST`)
  values(:MedID,:TDate,:Sname,:PDate,:BNo,:Qty,:SalesRate,:PurchaseRate,:ExpireDate,:CGST,:SGST)";
//echo $Query ;
  $stmt = $db->prepare($Query);
  $stmt->bindParam('MedID', $MID, PDO::PARAM_STR);
  $stmt->bindParam('TDate',$TDate, PDO::PARAM_STR);
  $stmt->bindParam('Sname',$Sid, PDO::PARAM_STR);
  $stmt->bindParam('PDate',$Pd, PDO::PARAM_STR);
  $stmt->bindParam('BNo',$BatchNo, PDO::PARAM_STR);
  $stmt->bindParam('Qty',$Qty, PDO::PARAM_STR);
  $stmt->bindParam('SalesRate',$SalesR, PDO::PARAM_STR);
  $stmt->bindParam('PurchaseRate',$PurchaseR, PDO::PARAM_STR);
  $stmt->bindParam('ExpireDate',$ExpireDate, PDO::PARAM_STR);
  $stmt->bindParam('CGST',$CGST, PDO::PARAM_STR);
  $stmt->bindParam('SGST',$SGST, PDO::PARAM_STR);
```

3. **Delete Data:** Data is selected through text fields in PhpMyAdmin and value as identifiers. In this case, the data is being deleted from"medicinemst" table using DELETEcommand.



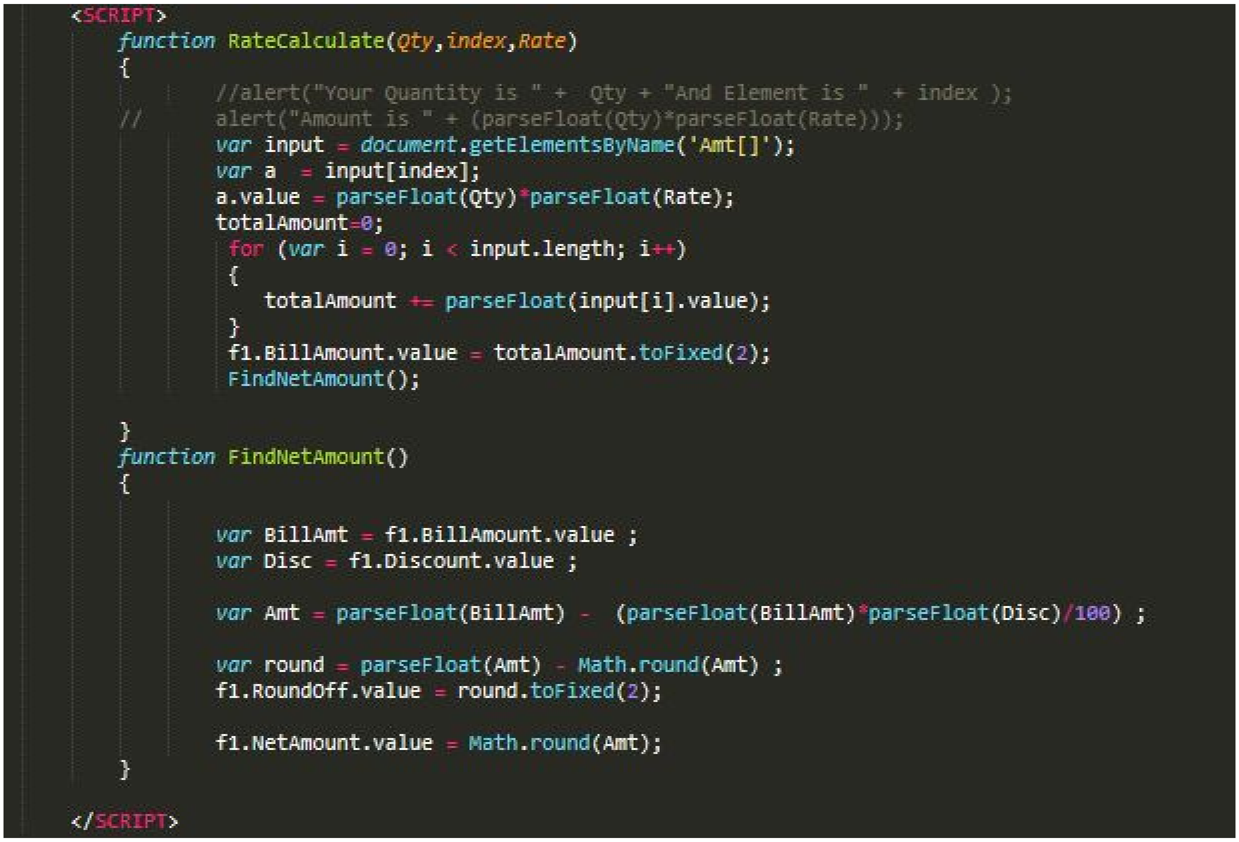
```
<?php
    include("auth.php");
    include("DbConnect.php");

    $MediID = $_REQUEST['MedID'];
    $Query = "Delete from medicinemst where MediID=$MediID";

    $stmt = $db->prepare($Query);
                        $stmt->execute();
    header("location:ViewMedicines.php?Msg=Medicine is Deleted Sucessfully.");

?>
```

4. **User-defined Classes:** There are user-defined classes such as HSN code, Medicine Name.



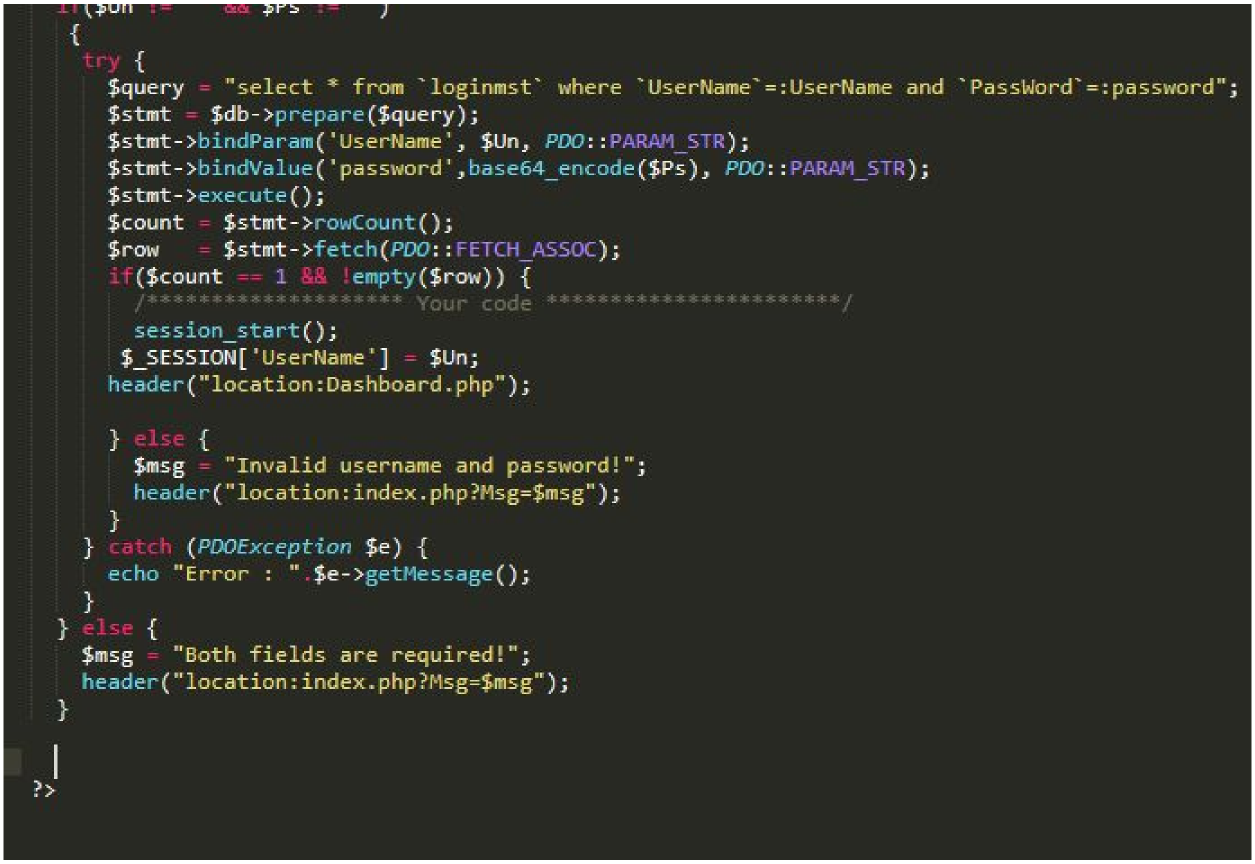
```
<?php include("SideMenu.php") ;?>

                <div class="panel panel-default">
                    <div class="panel-heading">Add New Medicine</div>
                    <div class="panel-body">
                        <div class="col-md-6">
                            <form role="form" name="f1" method="post" action="InsertMedicines.php">
                                <div class="form-group">
                                    <label>HSN code</label>
                                    <input type="varchar" class="form-control" name="HSNCode" placeholder="Enter HSN Code">
                                </div>
                                <div class="form-group">
                                    <label>Medicine Name</label>
                                    <input type="text" name="MedName" class="form-control" placeholder="Enter Medicine Name ">
                                </div>
                                <div class="form-group">
                                    <label>Manufacturing Company Name</label>
                                    <input type="text" name="CompanyName" class="form-control" placeholder="Enter Company Name ">
                                </div>
                                <div class="form-group">
                                    <label>Units(in no. of strips)</label>
                                    <input type="text" name="Units" class="form-control" placeholder="Enter Units ">
                                </div>

                                <div class="form-group">
                                    <label>Type of Medicine</label>
                                    <select class="form-select" name="TypeofMedi">
                                            <option  selected="">Capsule</option>
                                            <option>Tablet</option>
                                            <option>Syrup</option>
                                    </select>
                                </div>
```

5. **Implementing Javascript functions for calculations:** Use of javascript in Invoice generation. Users only have to enter the quantity and discount for required medicine when generating invoices. This allows invoice generation of multiple medicines in one time as all records are available in one place.



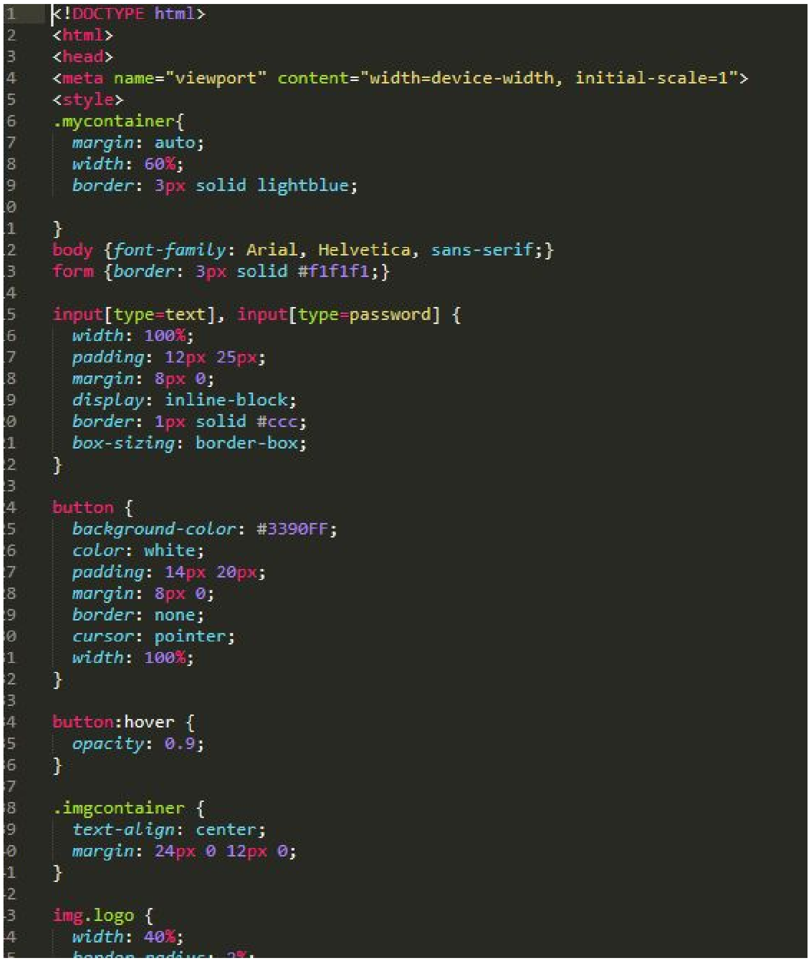
```
<SCRIPT>
    function RateCalculate(Qty,index,Rate)
    {
                //alert("Your Quantity is " +  Qty + "And Element is "  + index );
    //          alert("Amount is " + (parseFloat(Qty)*parseFloat(Rate)));
                var input = document.getElementsByName('Amt[]');
                var a  = input[index];
                a.value = parseFloat(Qty)*parseFloat(Rate);
                totalAmount=0;
                 for (var i = 0; i < input.length; i++)
                 {
                    totalAmount += parseFloat(input[i].value);
                 }
                 f1.BillAmount.value = totalAmount.toFixed(2);
                 FindNetAmount();

    }
    function FindNetAmount()
    {

                var BillAmt = f1.BillAmount.value ;
                var Disc = f1.Discount.value ;

                var Amt = parseFloat(BillAmt) -  (parseFloat(BillAmt)*parseFloat(Disc)/100) ;

                var round = parseFloat(Amt) - Math.round(Amt) ;
                f1.RoundOff.value = round.toFixed(2);

                f1.NetAmount.value = Math.round(Amt);
    }

</SCRIPT>
```

6. **Exception Handling:** Use of Try and Catch Statements in SQL queries. If queries are not executed in the Try block, the PDO Exception and error statement is shown to the user. In my case, I created an error message which is displayed.



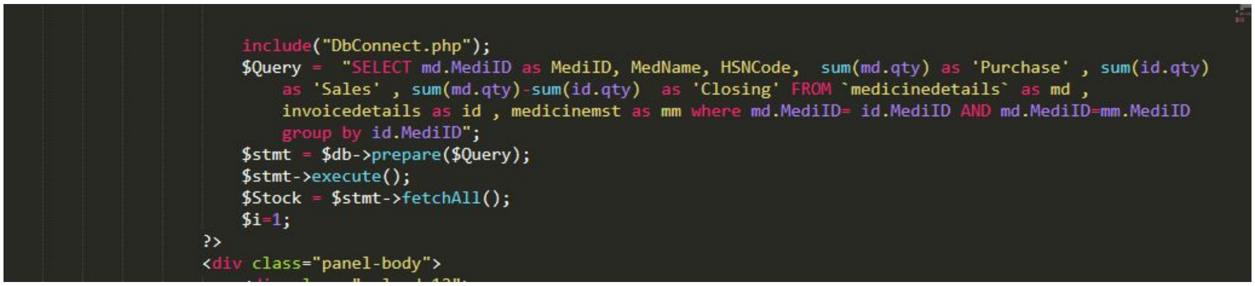
```
 if($Un !=     && $Ps !=    )
  {
   try {
     $query = "select * from `loginmst` where `UserName`=:UserName and `PassWord`=:password";
     $stmt = $db->prepare($query);
     $stmt->bindParam('UserName', $Un, PDO::PARAM_STR);
     $stmt->bindValue('password',base64_encode($Ps), PDO::PARAM_STR);
     $stmt->execute();
     $count = $stmt->rowCount();
     $row   = $stmt->fetch(PDO::FETCH_ASSOC);
     if($count == 1 && !empty($row)) {
       /******************** Your code ***********************/
       session_start();
      $_SESSION['UserName'] = $Un;
     header("location:Dashboard.php");

     } else {
       $msg = "Invalid username and password!";
       header("location:index.php?Msg=$msg");
     }
   } catch (PDOException $e) {
     echo "Error : ".$e->getMessage();
   }
 } else {
   $msg = "Both fields are required!";
   header("location:index.php?Msg=$msg");
 }

?>
```

7. **CSS Design:** Use of CSS to design elements in the Login Page to add personalization. Identification by “Dwarkesh Logo”, use of button functions like hover and span to ensure accurate design. Additional functions like opacity and border-radius add to a good visual appearance.CSS helps in personalizing the user interface and design throughout the application.



```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.mycontainer{
  margin: auto;
  width: 60%;
  border: 3px solid lightblue;

}
body {font-family: Arial, Helvetica, sans-serif;}
form {border: 3px solid #f1f1f1;}

input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 25px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

button {
  background-color: #3390FF;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.9;
}

.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}

img.logo {
  width: 40%;
  border-radius: 2%;
```

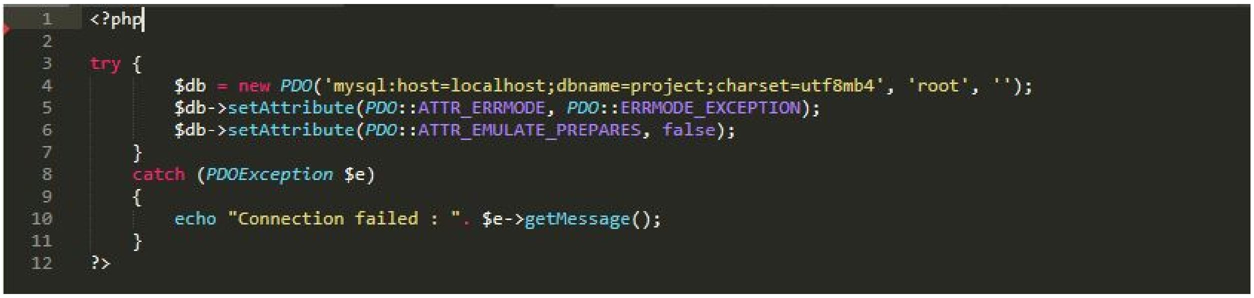
8. **Data entries from multiple tables for reports**- For the stock report, selecting data from three different database tables.For example, MedicineID is taken from medicinemst, InvoiceId from invoicemst and Sales from medicinedetails.



```
        include("DbConnect.php");
        $Query =  "SELECT md.MediID as MediID, MedName, HSNCode,  sum(md.qty) as 'Purchase' , sum(id.qty)
            as 'Sales' , sum(md.qty)-sum(id.qty)  as 'Closing' FROM `medicinedetails` as md ,
            invoicedetails as id , medicinemst as mm where md.MediID= id.MediID AND md.MediID=mm.MediID
            group by id.MediID";
        $stmt = $db->prepare($Query);
        $stmt->execute();
        $Stock = $stmt->fetchAll();
        $i=1;
    ?>
    <div class="panel-body">
```

9. **Other elements**:

- If…….elseif………..else statements
- For loop
- Database connect: Using file “Dbconnect” to ensure valid connection for the user.



```php
<?php

try {
        $db = new PDO('mysql:host=localhost;dbname=project;charset=utf8mb4', 'root', '');
        $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $db->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
    }
    catch (PDOException $e)
    {
        echo "Connection failed : ". $e->getMessage();
    }
?>
```

Word Count: 526 words