VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi – 590 018

A Mini Project Report on

"CAFÉ BILLING MANAGEMENT SYSTEM"

Submitted in partial fulfillment of the requirement for the DBMS Laboratory with Miniproject (21CSL55) of V Semester

Bachelor of Engineering In

Computer Science and Engineering

Submitted By

RAHUL SUNKAD	**[2JH21CS076]**
SHASHANK S SHETTY	**[2JH21CS089]**
SHRIVATSA D DESAI	**[2JH21CS094]**
UMESH LAXMAN NAYAK	**[2JH21CS0115]**

Under the Guidance of

Prof. RAJESHWARI PATIL

Designation, Dept. of CSE

JAIN COLLEGE OF ENGINEERING AND TECHNOLOGY, Hubballi 2023- 2024

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO SQL

SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database. MySQL is the most popular Open Source Relational SQL Database Management System. MySQL is one of the best RDBMS being used for developing various web-based software applications. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds. Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems. Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. MySQL is released under an open-source license. So you have nothing to pay to use it. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. MySQL uses a standard form of the well-known SQL data language. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. MySQL works very quickly and works well even with large data sets.

1.2 INTRODUCTION TO FRONT END SOFTWARE

- PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.
- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage
- dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle,
- Sybase. Informix, and Microsoft SQL Server.
- .PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier.
- Development a possibility for the first time. PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.
- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email
  you can send data, return data to the user.
- You add, delete, modify elements within your database through PHP. Access cookies variables and Set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.
- We used HTML.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 SOFTWARE REQUIREMENT

Operating System : 64/32 Bit Operating System, X64-Based Processor

Database : Mysql

Tools : Xampp

Scripting Language : PHP Admin

2.2 HARDWARE REQUIREMENTS

Processor : Any Processor Of Speed 1.60ghz

RAM : 1.00 GB Or More

Hard Disk : 20 GB Or More

Compact Disk : ANY DISKInput

Device : Keyboard

Output Devic : Laptop Display Screen

CHAPTER 3

OBJECTIVE OF THE PROJECT

The main objective of this Billing cafe Management System is to develop and implement a Reservation and Billing System for Cafe to help the management in making reservations, billing and other transactions needed to be made.

The main operations of the system are check-in, check-out, and reservation which are separated into different modules for functionality. The system can be used only if a registered user login to the system, only the admin can register a user for the system.

The proposed system has a user-friendly environment and has all the transactions of the business understudy like the reservation, billing, monitoring, and maintenance.

The system will make the employee's job easier and faster with error-free transactions for the good of the business.

The proponents will develop a computerized system containing the entirety of the business transaction..

The Main objectives of this project are:

- Reduction of paper work
- Automation of existing manual information systems.
- Reduction of manual processing
- Keep track of daily information exchange at the server by the administrator.
- Increase in processing and transfer speeds of information over the network.
- Decrease in processing time
- Reduction of errors and viruses due to absence of internet
- Keeping track that message should be delivered at the correct destination.
- Fast retrieval of all type of information
- Good efficiency and response time
- More consistent data handling
- A user-friendly system which do not require any special training or expertise of computer

CHAPTER 4

IMPLEMENTATION

4. 1 ER DIAGRAM

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation,normalizing relation and finally obtaining a relation database.

- ***ENTITIES:*** Which specify distinct real-world items in an application.
- ***PROPERTIES/ATTRIBUTES:*** Which specify properties of an entity and relationships.
- single-valued

  Most attributes have a single value for a particular entityFor

  eg:Age is a single-valued attribute of a person

- Multivalued

  An entity having multiple values for that attribute

  For eg: color of a color color={black,red} Person's

  degree degree={BE, MTech, PhD} Stored and

  Derived attribute

  Two (or more) attribute values are related—

  for eg: the Age and Birth_dateattributes of a person

  The value of Age can be determined from the current (today's) date and the valueof that person's Birth_date

  The Age attribute is hence called a derived attribute.

  Birth_date attribute is called a stored attribute

- NULL Values

  In some cases, a particular entity may not have an applicable value for an attribute.foreg, the Apartment_number attribute of an address applies only to addresses that are in apartment buildings and not to other types of residences, such as single-family homesCollege_degrees attribute applies only to people with college degrees

- Complex Attribute

composite and multivalued attributes can be nested arbitrarily arbitrary nesting bygrouping components of a composite attribute

between parentheses ( ) and separating the components with commas, and by displaying multivalued attributes between braces { }.

Such attributes are called complex attributes

For example, if a person can have more than one residence and each residence canhave a single

address and multiple phones, an attribute Address_phone for a person

- **RELATIONSHIPS:** Which connect entities and represent meaningful dependencies between them.

  whenever an attribute of one entity type refers to another entity type, some relationship exists

  for example, the attribute Manager of DEPARTMENT refers to an employee who manages the department,the

  attribute Controlling_department of PROJECT refers to the department that controls the project

  in the ER model, these references should not be represented as attributes but as relationships

  Relationship Types, Sets, and Instances

  A **r**elationship type R among n entity types E1, E2, . . . , En defines a set of associations—or a relationship

set—among entities from these entity types

entity types and entity sets, a relationship type and its corresponding relationship set are customarily referred
to by the same name, R

Mathematically, the relationship set R is a set of relationship instances ri, where each riassociates n individual
entities (e1, e2, . . . , en), and each entity ej in ri is a member of entity set Ej , $1 \leq j \leq n$

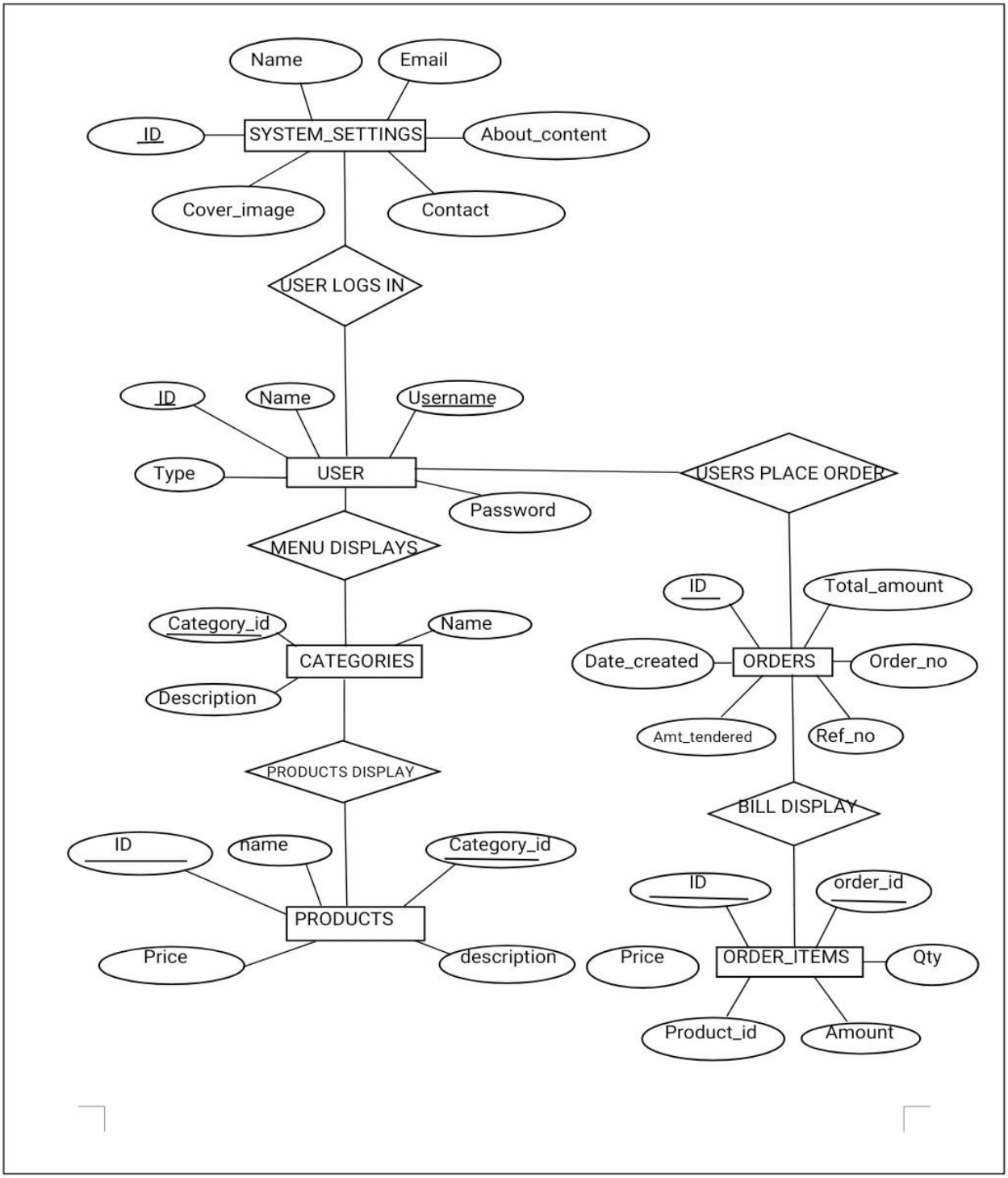
a relationship set is a mathematical relation on E1, E2, . . . , En; alternatively, it can be defined as a subset of
the Cartesian product of the entity sets $E1 \times E2 \times \ldots \times En$

each of the entity types E1, E2, . . . , En is said to participate in the relationship type R

each of the individual entities e1, e2, . . , en is said to participate in the relationship instance

Degree of a Relationship Type: The degree of a relationship type is the number of participating entity types.

ENTITY – RELATIONSHIP DIAGRAM :



4.2 MAPPING OF ER DIAGRAM TO SCHEMA

STEP1:

For each regular entity type E in the ER Schema, create relation R that includes all simple attributes of E.

STEP 2:

For each weak entity type W in the ER Schema with owner entity type E create a RelationR and include all simple attributes of Was attributes. In addition include as foreign key attribute of R the primary key attribute that correspond to owner entity.

STEP 3: MAPPING OF 1-N RELATIONSHIP

For each regular binary 1-N relationship type R identify the relation S that represents the participating entity type at the N side of the relation type include as foreign key in S theprimary key of relation T that represents the other entity type participating in R.

STEP 4: MAPPING OF M-N RELATONSHIP TYPE

Create a new Relation S to represent R include as foreign key in S1. The primary keys of relation S that represents the participating entity types their combination will formprimary key of S. Also include any simple attributes of the M-N relationship types as attributes of S.

SYSTEM SETTINGS

ID	Name	Email	About_content	Content	Cover_image

USER

ID	Username	Password	Type

CATEGORIES

Category_id	Name	Description

PRODUCT

Product_id	**Category_id**	Name	Price	Description

ORDERS

ID	Total_amount	Order_no	Ref_no	Amt_tendered	Date_created

ORDER_ITEMS

ID	**Order_id**	Product_id	Amount	Price	Qty

4.4 NORMALIZE THE RELATIONS

Normalization is the process of reorganizing data in a database so that it meets two basic requirements: (1) There is no redundancy of data (all data is stored in only one place), and (2) data dependencies are logical (all related data items are stored together). Normalization is important for many reasons, but chiefly because it allows databases to take up as little disk space as possible, resulting in increased performance.

Second Normal Form (2NF)

All the Relations are in 2NF because all the non-prime attribute in R are fully functionally dependent on the primary key of R.

Third Normal Form (3NF)

According to Codd's definition the relation schema is in 3NF because it satisfies 2NF and no non-prime attribute of R is transitively dependent on the primary key. The entire project is in Third Normal Form(3NF). 3NF is the highest normal form for this project.

4.5 CREATION AND INSERTION OF TABLES

■ **CREATE TABLE**

CATEGORIES (`ID` INT(30)

NOT NULL, `NAME`

VARCHAR(200) NOT NULL,

`DESCRIPTION` TEXT NOT

NULL) ENGINE=INNODB

DEFAULT

CHARSET=utf8mb4;

■ **DUMPING DATA FOR TABLE `CATEGORIES`**

INSERT INTO `CATEGORIES` (`ID`, `NAME`, `DESCRIPTION`)

VALUES(1, 'MEALS', 'SAMPLE CATEGORY '),

(2, 'BREAD', 'BREAD'),

(3, 'BEVERAGES', 'BEVERAGES');

■ **TABLE STRUCTURE FOR TABLE `ORDERS`**

CREATE TABLE `ORDERS` (

`ID` INT(11) NOT NULL,

`REF_NO` VARCHAR(50) NOT NULL,

`TOTAL_AMOUNT` FLOAT NOT NULL,

`AMOUNT_TENDERED` FLOAT NOT NULL,

`ORDER_NUMBER` INT(30) NOT NULL,

`DATE_CREATED` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP()

```
) ENGINE=INNODB DEFAULT CHARSET=UTF8MB4;
```

■ **DUMPING DATA FOR TABLE `ORDERS`**

```
INSERT INTO `ORDERS` (`ID`, `REF_NO`, `TOTAL_AMOUNT`, `AMOUNT_TENDERED`,
`ORDER_NUMBER`, `DATE_CREATED`) VALUES

(1, '989742510629', 170, 300, 1, '2020-11-02 15:29:46'),

(2, '590267910401', 115, 0, 0, '2020-11-02 16:06:07');
```

■ **TABLE STRUCTURE FOR TABLE `ORDER ITEMS`**

```
CREATE TABLE `ORDER_ITEMS` (

 `ID` INT(30) NOT NULL,

 `ORDER_ID` INT(30) NOT NULL,

 `PRODUCT_ID` INT(30) NOT NULL,

 `QTY` INT(30) NOT NULL,

 `PRICE` FLOAT NOT NULL,

 `AMOUNT` FLOAT NOT NULL

) ENGINE=INNODB DEFAULT CHARSET=UTF8MB4;
```

■ **DUMPING DATA FOR TABLE `ORDER ITEMS`**

```
INSERT INTO `ORDER_ITEMS` (`ID`, `ORDER_ID`, `PRODUCT_ID`, `QTY`, `PRICE`,
`AMOUNT`)

VALUES(1, 1, 4, 1,

70, 70),

(2, 1, 3, 1, 100, 100),

(3, 2, 3, 1, 100, 100),

(4, 2, 2, 1, 15, 15);
```

■ **TABLE STRUCTURE FOR TABLE 'PRODUCTS'**

```
CREATE TABLE `PRODUCTS` (
 `ID` INT(30) NOT NULL,
 `CATEGORY_ID` INT(30) NOT NULL,
 `NAME` VARCHAR(200) NOT NULL,
 `DESCRIPTION` TEXT NOT NULL,
 `PRICE` FLOAT NOT NULL,
 `STATUS` TINYINT(1) NOT NULL DEFAULT 1
COMMENT'0=UNAVAILABLE,1=AVAILABLE'
) ENGINE=INNODB DEFAULT CHARSET=UTF8MB4;
```

■ **DUMPING DATA FOR TABLE `PRODUCTS`**

```
INSERT INTO `PRODUCTS` (`ID`, `CATEGORY_ID`, `NAME`, `DESCRIPTION`, `PRICE`,
`STATUS`) VALUES
(1, 1, 'SAMPLE 1', 'SAMPLE 1', 500, 1),
(2, 3, 'ICED TEA', 'ICED TEA', 15, 1),
(3, 1, 'SAMPLE 2', 'SAMPLE 2', 100, 1),
(4, 1, 'FRIED CHICKEN SOLO', 'FRIED CHICKEN SOLO W/ RICE', 70, 1);
```

■ **TABLE STRUCTURE FOR TABLE `SYSTEM_SETTINGS`**

```
CREATE TABLE `SYSTEM_SETTINGS` (
 `ID` INT(30) NOT NULL,
 `NAME` TEXT NOT NULL,
 `EMAIL` VARCHAR(200) NOT NULL,
 `CONTACT` VARCHAR(20) NOT NULL,
```

```
  `COVER_IMG` TEXT NOT NULL,
  `ABOUT_CONTENT` TEXT NOT NULL
) ENGINE=INNODB DEFAULT CHARSET=UTF8MB4;
```

- **DUMPING DATA FOR TABLE `SYSTEM_SETTINGS'**

```
INSERT INTO `SYSTEM_SETTINGS` (`ID`, `NAME`, `EMAIL`, `CONTACT`, `COVER_IMG`, `ABOUT_CONTENT`) VALUES
(1, 'SIMPLE CAFE BILLING SYSTEM', '', '', '', '')
```

- **TABLE STRUCTURE FOR TABLE `USERS**

```
CREATE TABLE `USERS` (
  `ID` INT(30) NOT NULL,
  `NAME` TEXT NOT NULL,
  `USERNAME` VARCHAR(200) NOT NULL,
  `PASSWORD` TEXT NOT NULL,
  `TYPE` TINYINT(1) NOT NULL DEFAULT 3 COMMENT '1=ADMIN,2=STAFF'
) ENGINE=INNODB DEFAULT CHARSET=UTF8MB4;
```

- **DUMPING DATA FOR TABLE `USERS`**

```
INSERT INTO `USERS` (`ID`, `NAME`, `USERNAME`, `PASSWORD`, `TYPE`)
VALUES(1, 'ADMIN', 'ADMIN', '0192023A7BBD73250516F069DF18B500', 1),
(2, 'STAFF', 'STAFF', 'DE9BF5643EABF80F4A56FDA3BBB84483', 2)
-- INDEXES FOR DUMPED TABLES
```

- **INDEXES FOR TABLE `CATEGORIES`**

```
ALTER TABLE
 `CATEGORIES`ADD
 PRIMARY KEY (`ID`);
```

- **INDEXES FOR TABLE `ORDERS`**

```
ALTER TABLE
 `ORDERS` ADD
 PRIMARY KEY (`ID`);
```

- **INDEXES FOR TABLE `ORDER_ITEMS`**

```
ALTER TABLE
 `ORDER_ITEMS`ADD
 PRIMARY KEY (`ID`);
```

- **INDEXES FOR TABLE `PRODUCTS'**

```
ALTER TABLE
 `PRODUCTS`ADD
 PRIMARY KEY (`ID`);
```

- **INDEXES FOR TABLE `SYSTEM_SETTINGS`**

```
ALTER TABLE
 `SYSTEM_SETTINGS`ADD
 PRIMARY KEY (`ID`);
```

- **INDEXES FOR TABLE `USERS`**

```
ALTER TABLE `USERS`
 ADD PRIMARY KEY
 (`ID`);
```

```
-- AUTO_INCREMENT FOR DUMPED TABLES

-- AUTO_INCREMENT FOR TABLE

`CATEGORIES`ALTER TABLE `CATEGORIES`

  MODIFY `ID` INT(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

-- AUTO_INCREMENT FOR TABLE `ORDERS`

ALTER TABLE `ORDERS`

  MODIFY `ID` INT(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

-- AUTO_INCREMENT FOR TABLE `ORDER_ITEMS`

ALTER TABLE `ORDER_ITEMS`

  MODIFY `ID` INT(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

-- AUTO_INCREMENT FOR TABLE

`PRODUCTS`ALTER TABLE `PRODUCTS`

  MODIFY `ID` INT(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

-- AUTO_INCREMENT FOR TABLE

`SYSTEM_SETTINGS`ALTER TABLE

`SYSTEM_SETTINGS`

  MODIFY `ID` INT(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

-- AUTO_INCREMENT FOR TABLE `USERS`

ALTER TABLE `USERS`

  MODIFY `ID` INT(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

COMMIT;
```

CHAPTER 5

FRONT END DESIGN

5.1 SYSTEM DESIGN

INSERTION FROM FRONT END:

```
<!DOCTYPE html>
<html lang="en">

<?php session_start(); ?>
<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title><?php echo isset($_SESSION['system']['name']) ?
$_SESSION['system']['name'] : " ?></title>



<?php
 if(!isset($_SESSION['login_id']))
 header('location:login.php');
 include('./header.php');
 // include('./auth.php');
 ?>

</head>
<style>
```

```
        body{
      background: #80808045;
  }
  .modal-dialog.large {
    width: 80% !important;
    max-width: unset;
  }
  .modal-dialog.mid-large {
    width: 50% !important;
    max-width: unset;
  }
  #viewer_modal .btn-close {
    position: absolute;
    z-index: 999999;
    /*right: -4.5em;*/
    background: unset;
    color: white; border:
    unset;
    font-size: 27px;
    top: 0;
}
#viewer_modal  .modal-dialog  {
      width: 80%;
    max-width:    unset;
    height:   calc(90%);
    max-height: unset;
}
```

```
  #viewer_modal .modal-content {background:
      black;
    border: unset;
    height: calc(100%);
    display: flex;
    align-items: center;
    justify-content: center;
  }
  #viewer_modal img,#viewer_modal video{
    max-height: calc(100%);
    max-width: calc(100%);
  }
</style>

<body>
        <?php include 'topbar.php' ?>
        <?php include 'navbar.php' ?>
  <div class="toast" id="alert_toast" role="alert" aria-live="assertive" aria-atomic="true">
    <div class="toast-body text-white">
    </div>
  </div>

  <main id="view-panel" >
      <?php $page = isset($_GET['page']) ? $_GET['page'] :'home'; ?>
          <?php include $page.'.php' ?>
```

```
</main>

<div id="preloader"></div>
<a href="#" class="back-to-top"><i class="icofont-simple-up"></i></a>

    <div class="modal fade" id="confirm_modal" role='dialog'>
        <div class="modal-dialog modal-md" role="document">
  <div class="modal-content">
    <div class="modal-header">
    <h5 class="modal-title">Confirmation</h5>
  </div>
  <div class="modal-body">
    <div id="delete_content"></div>
  </div>
  <div class="modal-footer">
    <button type="button" class="btn btn-primary" id='confirm'
onclick="">Continue</button>
    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
  </div>
  </div>
 </div>
</div>
<div class="modal fade" id="uni_modal" role='dialog'>
 <div class="modal-dialog modal-md" role="document">
  <div class="modal-content">
```

```
        <div class="modal-header">
        <h5 class="modal-title"></h5>
      </div>
      <div class="modal-body">
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-primary" id='submit'onclick="$('#uni_modal
form').submit()">Save</button>
        <button type="button" class="btn btn-secondary" data-
dismiss="modal">Cancel</button>
      </div>
      </div>
    </div>
  </div>
  <div class="modal fade" id="viewer_modal" role='dialog'>
    <div class="modal-dialog modal-md" role="document">
      <div class="modal-content">
            <button type="button" class="btn-close" data- dismiss="modal"><span
class="fa fa-times"></span></button>
            <img src="" alt="">
      </div>
    </div>
  </div>
</body>
<script>
          window.start_load = function(){
    $('body').prepend('<di  id="preloader2"></di>')
```

```
 }
 window.end_load = function(){
  $('#preloader2').fadeOut('fast',  function()  {
      $(this).remove();
    })
 }
window.viewer_modal = function($src = ''){
  start_load()
  var t = $src.split('.')t
  = t[1]
  if(t =='mp4'){
   var view = $("<video src='"+$src+"' controls autoplay></video>")
  }else{
   var view = $("<img src='"+$src+"' />")
  }
  $('#viewer_modal .modal-content video,#viewer_modal .modal-contentimg').remove()
  $('#viewer_modal  .modal-content').append(view)
  $('#viewer_modal').modal({
         show:true,
         backdrop:'static',
         keyboard:false, focus:true
        })
        end_load()

}
```

```
window.uni_modal = function($title = " , $url='',$size=''){
  start_load()
  $.ajax({
    url:$url,
    error:err=>{
      console.log()
      alert("An error occured")
    },
    success:function(resp){ if(resp){
        $('#uni_modal .modal-title').html($title)
        $('#uni_modal .modal-body').html(resp)if($size !=
        ''){
          $('#uni_modal .modal-dialog').addClass($size)
        }else{
          $('#uni_modal .modal-
dialog').removeAttr("class").addClass("modal-dialog modal-md")
        }
        $('#uni_modal').modal({
          show:true,
          backdrop:'static',
          keyboard:false,
          focus:true
        })
        end_load()
      }
    }
```

```
    })
}
window._conf = function($msg='',$func='',$params = []){
    $('#confirm_modal
#confirm').attr('onclick',$func+"("+$params.join(',')+")")
    $('#confirm_modal  .modal-body').html($msg)
    $('#confirm_modal').modal('show')
 }
  window.alert_toast= function($msg = 'TEST',$bg = 'success'){
     $('#alert_toast').removeClass('bg-success')
     $('#alert_toast').removeClass('bg-danger')
     $('#alert_toast').removeClass('bg-info')
     $('#alert_toast').removeClass('bg-warning')

   if($bg == 'success')
     $('#alert_toast').addClass('bg-success') if($bg ==
   'danger')
     $('#alert_toast').addClass('bg-danger') if($bg ==
   'info')
     $('#alert_toast').addClass('bg-info') if($bg
   == 'warning')
     $('#alert_toast').addClass('bg-warning')
   $('#alert_toast  .toast-body').html($msg)
   $('#alert_toast').toast({delay:3000}).toast('show');
 }
 $(document).ready(function(){
   $('#preloader').fadeOut('fast',  function()  {
```

```
        $(this).remove();
      })
  })
  $('.datetimepicker').datetimepicker({ format:'Y/m/d
     H:i',
     startDate: '+3d'
  })
  $('.select2').select2({ placeholder:"Please
    select here",width: "100%"
  })
</script>
</html>
```

CONNECTIVITY TO BACK END FROM FRONT END TO INSERT USING XAMPP

```
<?php
include('db_connect.php');
session_start();
if(isset($_GET['id'])){
$user = $conn->query("SELECT * FROM users where id
="$_GET['id']);foreach($user->fetch_array() as $k =>$v){
```

```
        $meta[$k] = $v;
    }
    }
    ?>
    <div class="container-fluid">
        <div id="msg"></div>

        <form action="" id="manage-user">
            <input type="hidden" name="id" value="<?php echo
isset($meta['id']) ? $meta['id']: '' ?>">
            <div class="form-group">
                <label for="name">Name</label>
                <input type="text" name="name" id="name"
class="form-control" value="<?php echo isset($meta['name']) ?
$meta['name']: '' ?>" required>
            </div>
            <div class="form-group">
                <label for="username">Username</label>
                <input type="text" name="username" id="username"
```

```
class="form-control" value="<?php echo isset($meta['username']) ?
$meta['username']: '' ?>" required  autocomplete="off">
            </div>
            <div class="form-group">
                <label for="password">Password</label>
                <input type="password" name="password"
id="password"class="form-control" value="" autocomplete="off">
                <?php if(isset($meta['id'])): ?>
                <small><i>Leave this blank if you dont want to change the
password.</i></small>
            <?php endif; ?>
            </div>
            <?php if(isset($meta['type']) && $meta['type'] == 3): ?>
                <input type="hidden" name="type" value="3">
            <?php else: ?>
            <?php if(!isset($_GET['mtype'])): ?>
            <div class="form-group">
                <label for="type">User Type</label>
                <select name="type" id="type" class="custom-select">
```

```
                        <option value="2" <?php echo isset($meta['type']) &&
$meta['type'] == 2 ? 'selected': '' ?>>Staff</option>
                        <option value="1" <?php echo isset($meta['type']) &&
$meta['type'] == 1 ? 'selected': '' ?>>Admin</option>
                    </select>
                </div>
                <?php endif; ?>
                <?php endif; ?>


        </form>
</div>
<script>

        $('#manage-user').submit(function(e){
                e.preventDefault();
                start_load()
                $.ajax({
                        url:'ajax.php?action=save_user',
```

```
                method:'POST',
                data:$(this).serialize(),
                success:function(resp){
                    if(resp ==1){
                        alert_toast("Data successfully
                        saved",'success')setTimeout(function(){
                            location.reload()
                        },1500)
                    }else{
                        $('#msg').html('<div class="alert
alert-danger">Username already exist</div>')
                        end_load()
                    }
                }
            })
        })
</script>
```

CHAPTER 6

TESTING

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. 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 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 in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TESTING PROCESS

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested.

TESTING OBJECTIVES

The main objectives of testing process are as follows.

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has high probability of finding undiscovered error.
- A successful test is one that uncovers the undiscovered error.

Test cases for the project

In my system the following testing is done

- When a record is inserted in the front end, it gets saved in the back end.
- Deletion of a record will delete individual tables are tested, hence unit testingdone

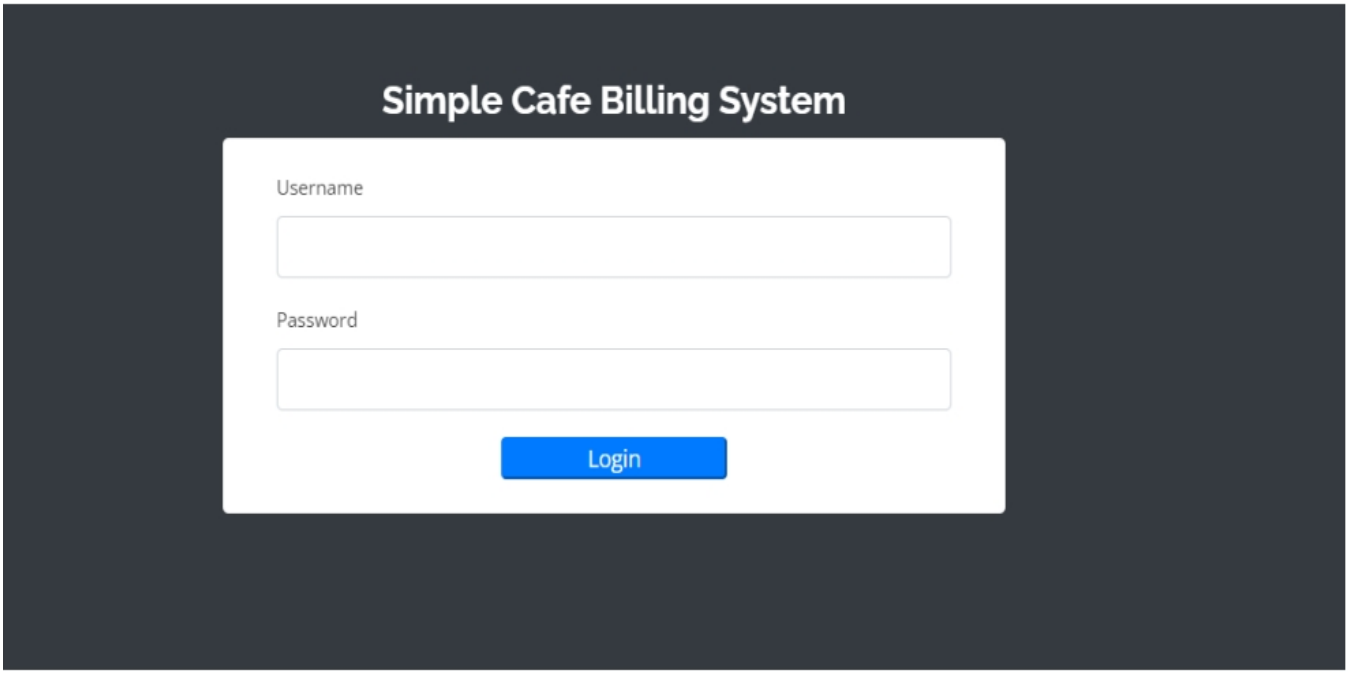
- All forms are linked and checked therefore integration testing is done. Finallysystem testing is done on deployment.

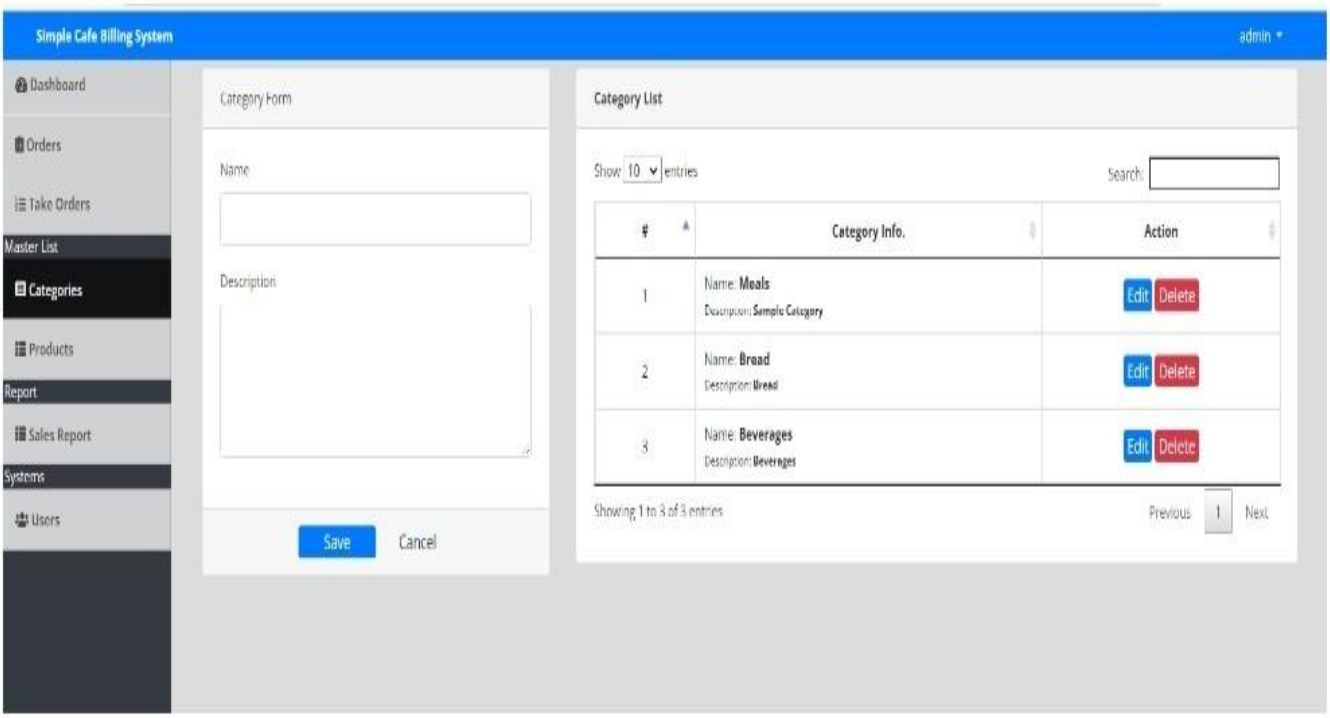
Table 6.1: Test cases for the project

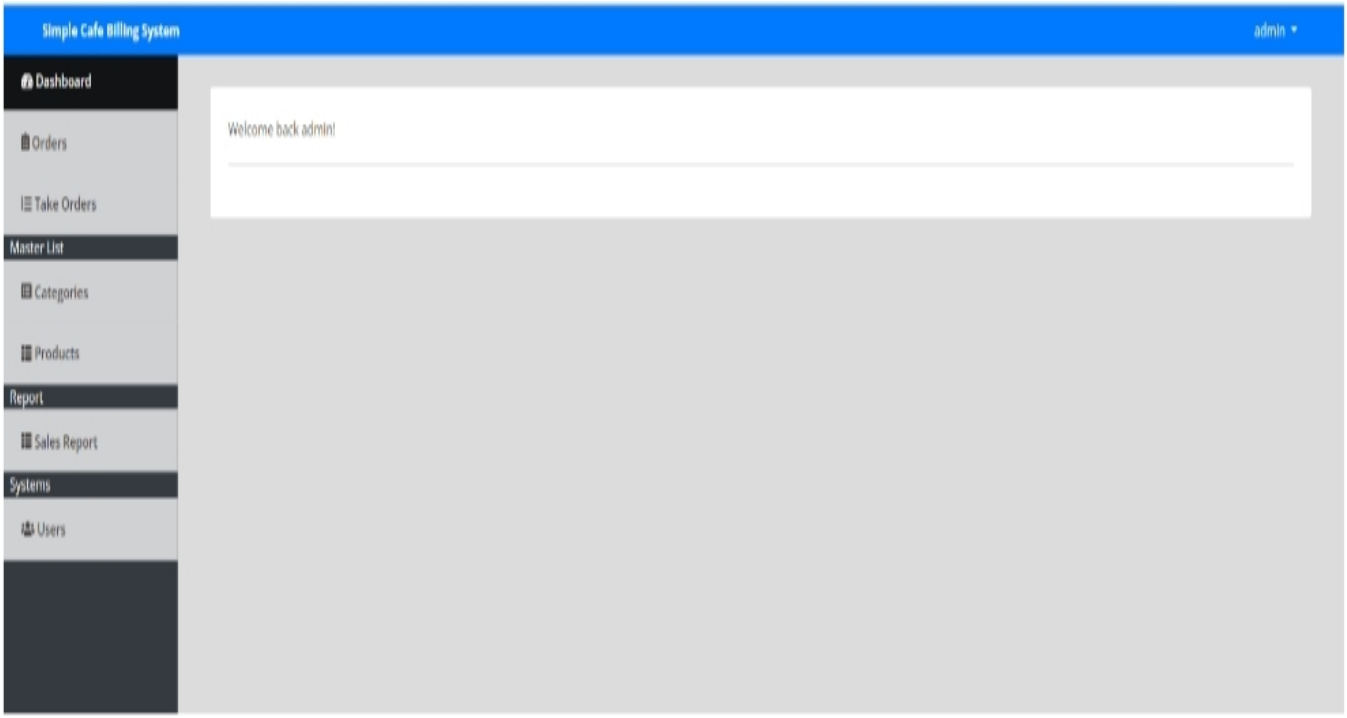
Sl No	Test Input	Expected Results	Observed Results	Remarks
1	Insert a Record	New tuple should be	Query OK 1 row	PASS
		inserted	affected or inserted	
2	Insert a Record	New tuple should be	ERROR	FAIL
		inserted		
3	Search a Record	Display the Record	Required Record	TRUE
			Displayed	
4	Search a Record	Display the Record	No Record Found	FAIL
5	Delete a Record	Delete the Record	Query OK 1 row	PASS
			affected or Record	
			Deleted	
6	Create trigger	Trigger Created	Query OK Trigger	PASS
			created	
7	Create Stored	Stored Procedures	Query OK Stored	PASS
	Procedures	Created	Procedures Created	

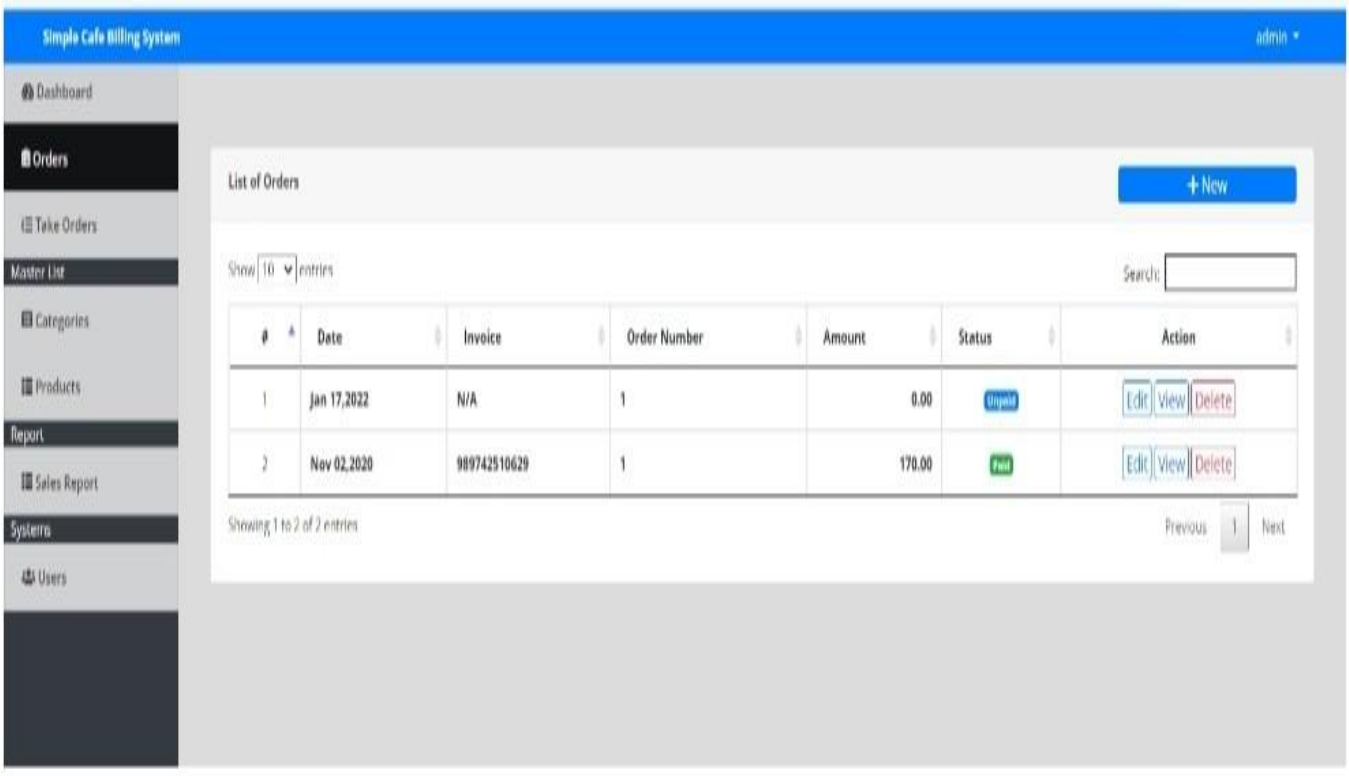
CHAPTER 7

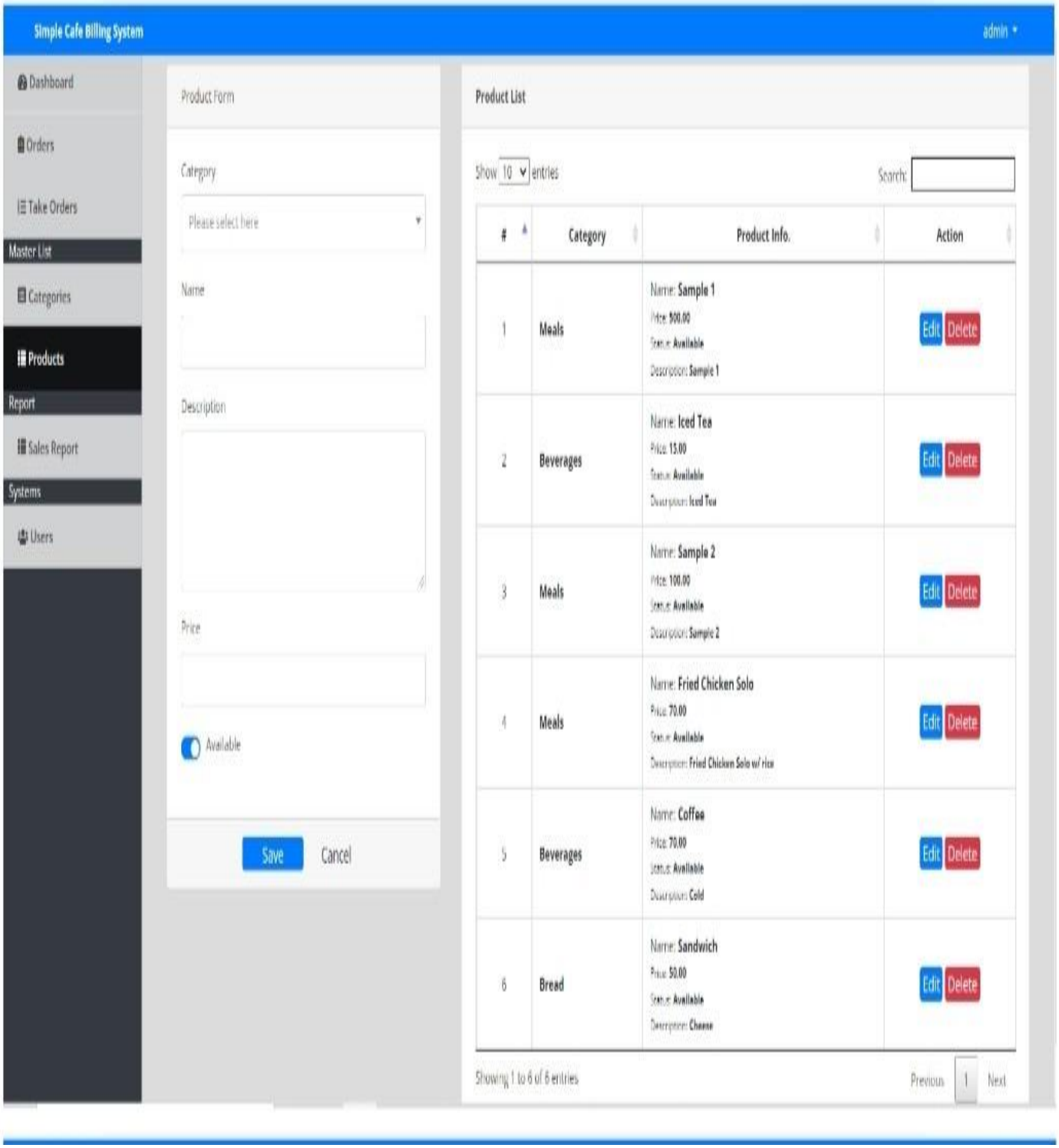
SNAPSHOTS

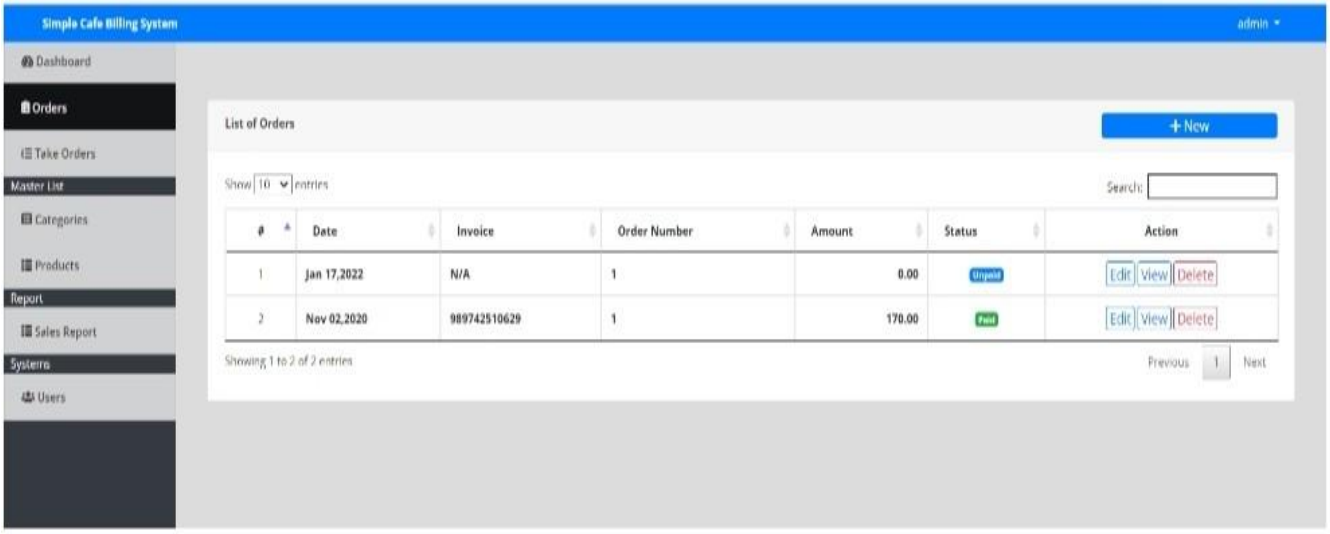


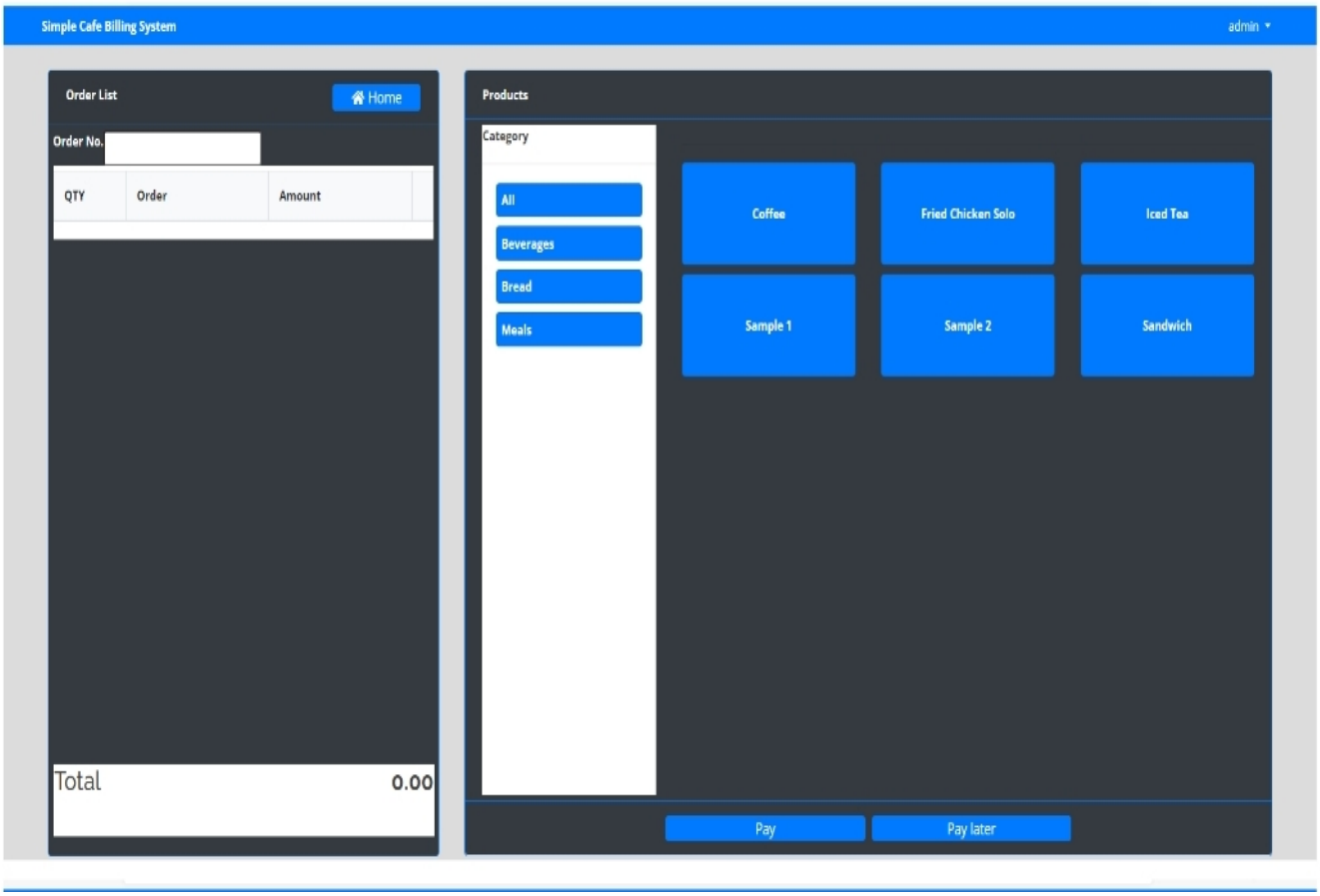


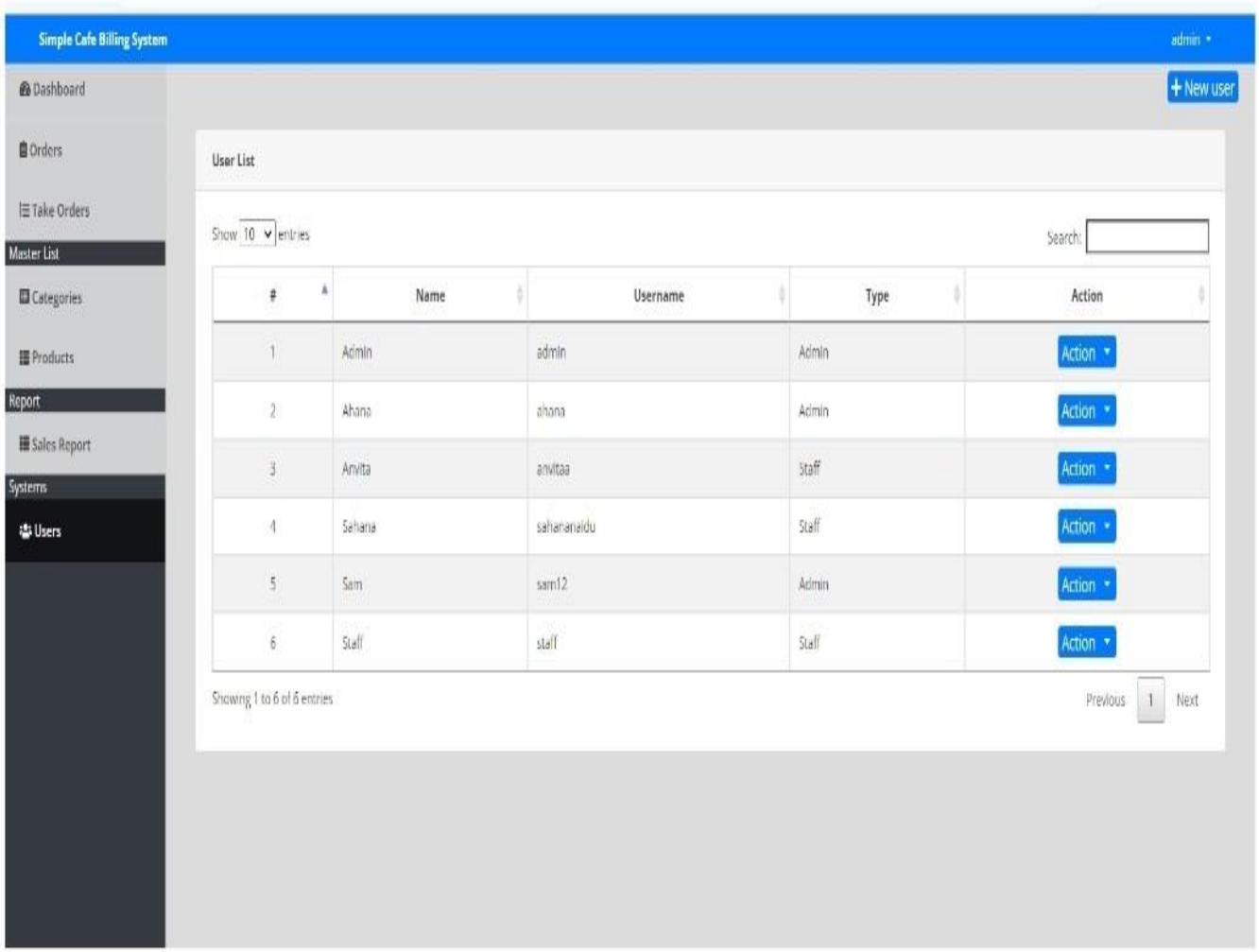

Simple Cafe Billing System
admin
Dashboard
Orders
Take Orders
Master List
Categories
Products
Report
Sales Report
Systems
Users
Welcome back admin!


Simple Cafe Billing System
admin
Dashboard
Orders
Take Orders
Master List
Categories
Products
Report
Sales Report
Systems
Users
List of Orders
+New
Show 10 entries
Search:
Date Invoice Order Number Amount Status Action
1 Jan 17,2022 N/A 1 0.00 Unpaid Edit View Delete
2 Nov 02,2020 989742510629 1 170.00 Paid Edit View Delete
Showing 1 to 2 of 2 entries
Previous 1 Next


Simple Cafe Billing System
admin
Dashboard
Orders
Take Orders
Master List
Categories
Products
Report
Sales Report
Systems
Users
Product Form
Category
Please select here
Name
Description
Price
Available
Save
Cancel
Product List
Show 10 entries
Search:
Category Product Info. Action
1 Meals Name: Sample 1 Price: 500.00 Status: Available Description: Sample 1 Edit Delete
2 Beverages Name: Iced Tea Price: 15.00 Status: Available Description: Iced Tea Edit Delete
3 Meals Name: Sample 2 Price: 100.00 Status: Available Description: Sample 2 Edit Delete
4 Meals Name: Fried Chicken Solo Price: 70.00 Status: Available Description: Fried Chicken Solo w/ rice Edit Delete
5 Beverages Name: Coffee Price: 70.00 Status: Available Description: Cold Edit Delete
6 Bread Name: Sandwich Price: 50.00 Status: Available Description: Cheese Edit Delete
Showing 1 to 6 of 6 entries
Previous 1 Next


Simple Cafe Billing System
admin
Dashboard
Orders
Take Orders
Master List
Categories
Products
Report
Sales Report
Systems
Users
List of Orders
+ New
Show 10 entries
Search:
Date Invoice Order Number Amount Status Action
1 Jan 17,2022 N/A 1 0.00 Unpaid Edit View Delete
2 Nov 02,2020 989742510629 1 170.00 Paid Edit View Delete
Showing 1 to 2 of 2 entries
Previous 1 Next


Simple Cafe Billing System
admin
Order List
Home
Order No.
QTY Order Amount
Total 0.00
Products
Category
All
Beverages
Bread
Meals
Coffee
Fried Chicken Solo
Iced Tea
Sample 1
Sample 2
Sandwich
Pay
Pay later


Simple Cafe Billing System
admin
Dashboard
Orders
Take Orders
Master List
Categories
Products
Report
Sales Report
Systems
Users
+ New user
User List
Show 10 entries
Search:
Name Username Type Action
1 Admin admin Admin Action
2 Ahana ahana Admin Action
3 Anvita anvitaa Staff Action
4 Sahana sahanaraidu Staff Action
5 Sam sam12 Admin Action
6 Staff staff Staff Action
Showing 1 to 6 of 6 entries
Previous 1 Next

CHAPTER 8:

CONCLUSION

The operation points at the support and administration of the various accommodations are possible in the different parts of the system. It mainly takes care of the resort management at the core area of the database. The method provides information about the varied options that are accessible and their situation-specific to availability. The database also maintains the atomic information regarding the different units that are available under one resort and the compositional details of the Unit facilities that are available. Each unit is well provided and is completely planned for the basic adaptability of the visitors who are expected to stay.