**DRIVING SCHOOL MONITORING SYSTEM**

A project report submitted in partial fulfillment of the requirements for the award of the Degree of

**BACHELOR OF SCIENCE IN COMPUTER TECHNOLOGY**

Submitted by

TAMILARASU.B

**(Reg.No:161CT154)**

Under the Guidance of

**Dr.P.Iswarya M.Sc.,Ph.D.,**

(Assistant Professor, Department of COMPUTER Technology)



**DEPARTMENT OF COMPUTER TECHNOLOGY**

Dr. N.G.P. ARTS AND SCIENCE COLLEGE (Autonomous)

(Affiliated to Bharathiar University, Re-Accredited by NAAC with ‘A’ Grade)

Coimbatore– 641 048

March-2019

**DEPARTMENT OF COMPUTER TECHNOLOGY**

**Dr.N.G.P. ARTS AND SCIENCE COLLEGE** (Autonomous)

(Affiliated to Bharathiar University)

(Re-Accredited by NAAC with “A “Grade)

Coimbatore – 641 048

This is to certify that the Project work entitled

**DRIVING SCHOOL MONITORING SYSTEM**

Is bonafide record of work done by

TAMILARASU.B

**(Register No: 161CT154)**

Submitted in partial fulfillment of the requirements for the award of the degree of

**Bachelor of Computer Technology**

Bharathiar University, Coimbatore-641048

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Guide Head of the Department**

**Submitted for Viva-Voce Examination held on\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Internal Examiner** **External Examiner**

**CERTIFICATE**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **CERTIFICATE**

This is to certify that the project work entitled “**DRIVING SCHOOL MONITORING SYSTEM**” is a bonafide work done by TAMILARASU.B **(Reg. No: 161CT154) submitted** in partialrequirements for the award of the degree of **Bachelor of Science Computer Technology** has not formed the basis for the award degree,diplom,associationship,fellowship or any other similar title and I also certify that the project work represents an independent work on the part of the candidate.

**DECLARATION**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**DECLARATION**

I, TAMILARASU.B **(Register No: 161CT154)** do hereby declare that this project titled **“DRIVING SCHOOL MONITORING SYSTEM”** , submitted to Dr.N.G.P. Arts and Science College, affiliated to Bharathiar University, Coimbatore in partial fulfillment for the award of the degree of Computer Technology , is a record of original work done by me under the guidance of **Dr.P.Iswarya M.Sc.,Ph.D.,** Assistant Professor, Department of Computer Technology, Dr.N.G.P Arts and Science College(Autonomous),Coimbatore and this project work has not been submitted for the award of any other Degree/Diploma/Fellowship or similar title any other university.

**PLACE**: Coimbatore

**DATE**: **(TAMILARASU.B)**

**ACKNOWLEDGEMENT**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGEMENT**

My deep sense of gratitude and thanks to our beloved chairman Dr.Nalla **G.Palaniswami M.D.A.B (USA)**.**Dr.ThavamaniD.Palaniswami, Secretary,** Dr.N.G.P Arts and Science College, Coimbatore for giving me an opportunity to undergo my studies in this prestigious institution.

I wish to express my profound thanks to **Prof. Dr.V.Rajendran M.Sc., M.Phil., B.Ed., M.Tech., (Nanotech) Ph.D., (D.Sc) Finst P.(London), Principal** Dr.N.G.P. Arts and Science College for his permission to undertake this study in this esteemed institution.

I would like to thank the Head of the Department **Mrs.D.Maheswari** **M.Sc, M.Phil, Ph.D.,** for his encouragement and motivation in carrying out my work successfully.

My special thanks to **Dr.P.Iswarya M.Sc.,Ph.D.,** Assistant Professor, and **Department of Computer Technology,** Dr.N.G.P Arts and Science College, Coimbatore for the successful completion and submission of my project.

I extend my science thanks to all the staff members of the Department of Computer Technology for their support and guidance.

I wish to thank my parents and friends for their support and encouragement throughout the work.

**CONTENT**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **PARTICULARS** | **Page.No** |
|  | **ABSTRACT** | **1** |
| **1** | **INTRODUCTION** |  |
|  | 1.1 OVERVIEW OF THE PROJECT | **3** |
| **2** | **SYSTEM SPECIFICATION** |  |
|  | 2.1.HARDWARE REQUIREMENTS | **4** |
|  | 2.2.SOFTWARE REQUIREMENTS |  |
| **3** | **SYSTEM ANALYSIS** |  |
|  | 3.1.EXISTING SYSTEM | **6** |
|  | 3.2.PROPOSED SYSTEM |  |
|  | 3.3.SOFTWARE DESCRIPTION | **7** |
| **4** | **SYSTEM DESGIN AND DEVELOPMENT** |  |
|  | 4.1.INPUT DESGIN | **10** |
|  | 4.2.OUTPUT DESGIN |  |
|  | 4.3.DATABASE DESGIN |  |
|  | 4.4.SYSTEM MODULES DESCRIPTION | **11** |
| **5** | **TESTING AND IMPLEMENTATAION** |  |
|  | 5.1.SYSTEM TESTING | **14** |
|  | 5.2.SYSTEM IMPLEMENTATION | **16** |
| **6** | **CONCLUSION & FURTHER ENHANCEMENT** | **19** |
| **7** | **BIBLIOGRAPHY** | **21** |
| **8** | **APPENDICES** | **23** |
|  | A.DATA FLOW DIAGRAM |  |
|  | B.ER DIAGRAM | **28** |
|  | C.SAMPLE CODING |  |
|  | D.SCREENSHOTS |  |

**ABSTRACT**

The aim of this project is to develop an application, which is called “**DRIVING SCHOOL MONITORING SYSTEM**”. It has developed using to PHP as Front-End and MY SQL as Back-End. The aim of this project is develop a “**DRIVING SCHOOL MONITORING SYSTEM**” that helps the driving schools to automate the manual tasks of maintaining clients,Masters data in database. The system designed has an admin login from where the admin can administers the system through admin’s dashboard. Students can access the system on admin’s approval by creating an account online. Students can then view all the categories in the system where they can select the vehicle for training, timing slot, fee packages, and sessions. The system calculates the total fees and students can then make payment online via credit card. The receipt is delivered to the student’s email by the system. System also keeps a track of the student’s session and timings and reminds them via email for every session one day prior. The system also has Master login provided by the admin.Master can thus view his assigned student for training and the next session.

As such, The **“** DRIVING SCHOOL MONITORING SYSTEM **“** system is to be developed to replace such COMPUTER system in order to provide a better controlled and efficient environment which will meet the needs of the day’s services.

**1**

**INTRODUCTION**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2**

# 1. INTRODUCTION

**1.1 Overview of the Project**

In today’s world, driving has become one of the most important human need rather than a luxury. With such kind of thinking driving schools have emerged all over the country to train professional and nonprofessional drivers to meet the ever growing demand for more drivers.

While there exists such kind of demand, observation has shown that most of these schools don’t have proper systems in place to manage such kind of training schools as most of these schools usually use manual systems which have several problems as opposed to today’s world which needs well managed COMPUTER systems to help in such environments.

As such, The **“ DRIVING SCHOOL MONITORING SYSTEM “** system is to be developed to replace such COMPUTER system in order to provide a better controlled and efficient environment which will meet the needs of the day’s services.

The Driving school has following functions:

**Student Registration:**

It is responsible for registering the students and keeping the student personal files. The file is called Student Personal File.

**Instructors:**

The function is responsible for scheduling classes known as Instructors File.

**Accounts:**

This function is responsible for the collection and monitoring of tuition fees payments, it is known as Fees Filed.

**Vehicle Fleet Control:**

This function is responsible for managing to fleet of vehicles at the school, t is called vehicle file.

**3**

**2. SYSTEM SPECIFICATION**

# 2.1 HARDWARE SPECIFICATION

**COMPONENT REQUIREMENT**

|  |  |  |
| --- | --- | --- |
| Processor | : | Intel(R) Pentium(R) Dual CPU E2200 @ 2.20GHz |
| RAM | : | 2 GB |
| Hard disk | : | 80GB or more |
| Monitor | : | Generic PnP Monitor |
| Keyboard | : | Standard PS/2 Keyboard |
| Mouse | : | Microsoft PS/2 Mouse |

# 2.2 SOFTWARE SPECIFICATION

**COMPONENT REQUIREMENT**

|  |  |  |
| --- | --- | --- |
| Front End | : | PHP |
| Back End | : | MY SQL |
| Operation System | : | Windows XP or Latest Versions |
|  |  |  |
|  |  |  |

4

**SYSTEM ANALYSIS**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 5

**3. SYSTEM ANALYSIS**

# 3.1 EXISTING SYSTEM:

 The Driving School Monitoring System works are done in manual.

 The Admin have to maintain the driving student details properly. Otherwise they cannot allot timings for each learner.

 And the Driving school have to preserve whether the learner need license. If they need license the admin should apply for license.

 These processes are also made by manually. So the manual work is very difficult to run the driving classes.

3.2 PROPOSED SYSTEM:

 Driving School Monitoring System contains numerous features that can help driving schools run their business more efficiently.

 One great advantage this software offers is its scalability and flexibility for data management. Using this software admin will be able to manage the most important aspects of business using single software.

 As such, The Driving School Monitoring system is to be developed to replace such COMPUTER system in order to provide a better controlled and efficient environment which will meet the needs of the day’s services.

**6**

**3.3. SOFTWARE DESCRIPTION**

**FRONT – END (PHP)**

**DEFINITION**

The Web server accepts the request and sends the HTML to the Client browser that requests it. Web browser and web server communicate through a common protocol (HTTP). The examples for web server are XAMPP(any of four different operating systems,Apache,MySQL,Php,Perl),WAMP(Windows,Apache,MySQL,Php),MAMP(Macintosh, Apache, MYSQL, PHP).

**PHP:**

PHP stands for Hypertext Pre-processor. PHP scripts run inside Apache server or Microsoft IIS. PHP and Apache server are free. PHP code is very easy. PHP is the most used server side scripting language. PHP files contain PHP scripts and HTML. PHP files have the extension “php”, “php3”, “php4”, or “html”.

PHP code may be executed with a [command line interface](https://en.wikipedia.org/wiki/Command-line_interface) (CLI), embedded into [HTML](https://en.wikipedia.org/wiki/HTML) code, or it can be used in combination with various [web tDrilate systems](https://en.wikipedia.org/wiki/Web_template_system), web content management systems, and [web frameworks](https://en.wikipedia.org/wiki/Web_framework). PHP code is usually processed by a PHP [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) implemented as a [module](https://en.wikipedia.org/wiki/Plugin_(computing)) in a web server or as a [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP can be used for many programming tasks outside of the web context, such as [standalone](https://en.wikipedia.org/wiki/Computer_software) [graphical applications](https://en.wikipedia.org/wiki/Graphical_user_interface) and robotic [drone](https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle) control.

**Using PHP**

Generate dynamic web pages. PHP can display different content to different user or display different content at different times of the day Process the contents of HTML forms. We can use a PHP to retrieve and respond to the data entered into an HTML form.

Unqualified name, or an unprefixed class name like $a = new foo (); or *foo::static method ();* If the current namespace is *current namespace*, this resolves to *current namespace\foo*. If the code is global, non-namespaced code, this resolves to *foo*. One caveat: unqualified names functions. 7

**BACK – END (MYSQL)**

A database is simply a collection of used data just like phone book. MYSQL database include such objects as tables, queries, forms, and more.

In MYSQL tables are collection of similar data. With all tables can be organized differently, and contain mostly different information- but they should all be in the same database file. For instance we may have a database file called video store. Containing tables named members, tapes, reservations and so on. These tables are stored in the same database file because they are often used together to create reports to help to fill out on screen forms.

Relational database:

MYSQL is a relational database. Relational databases tools like access can help us manage COMPUTER in three important ways.

 Reduce redundancy

 Facilitate the sharing of information

 Keep data accurate

**ADVANTAGES OF MYSQL DATABASE SERVER**

The ‘MYSQL’ database server is very fast, reliable and easy to use. If that is what is what you are looking for, you should try it. ‘MYSQL’ server also has a particle set features developed in close cooperation with users.

‘MYSQL’ was originally developed to handle large database much faster than existing solution and has been successfully use in highly demanding production environments for several years.

MySQL is globally renowned for being the most secure and reliable database management system used in popular web applications including WordPress, Drupal, Joomla, Facebook and Twitter. The data security and support for transactional processing that accompany the recent version of MySQL can greatly benefit any business, especially if it is an ecommerce business that involves frequent money transfers.

8

**SYSTEM DESIGN AND DEVELOPMENT**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9

**4.1 INPUT DESGIN:**

Input design is the process of connecting the user-originated inputs into a computer to used format. The goal of the input design is to make the data entry logical & free from errors. Errors in the input database controlled by input design.

The goal of designing input data is to make data entry as easy, logical and error free from errors as possible. In entering data, operators need to know the following:

Admin Module, Candidate Module, License Module, Driver Module, Payment Module, Expense Module

**4.2 OUTPUT DESGIN:**

The output from the system is either by screen or by hard copies. Output design aims at communicating the results of the processing of the users. The reports are generated to suit the needs of the users. The reports have to be generated with appropriate levels. Presenting the data processed by a computer-based COMPUTER system in an attractive and usable form has become very essential these days’ success and acceptance of a system to some extent depends on good presentation. Therefore, system analyst must know fully how to design output report in an attractive way.

# 4.3 DATABASE DESGIN:

The database design involves creation of tables. Tables are represented in physical database as stored files. They have their own independent existence. A table consists of rows and columns. Each column corresponds to a piece of information, specific to a particular item.

This activity deals with the design of the physical database. A key is to determine how the access paths art to be implemented.

10

**4.4 SYSTEM MODULE DESCRIPTION :**

The **“Driving School Monitoring System”** contains following modules.

 Admin Module

 Candidate Module

 License Module

 Driver Module

 Payment Module

 Expense Module

**DESCRIPTION OF MODULES**

**Admin Module**

The module enables the admin people to create and modify the login id and password. This login creation is useful to access candidate, Driver, payment and expense details. When authorized person gives valid login id and password, he can access those details.

**Candidate Details**

In the candidate module, the admin people can enter the new candidate details (name of the candidate, address, contact number, mail id, date of birth, gender, and date of joining). Here the Candidates details are stored by the category wise. And all authorized people can view the candidate details also.

**License Module**

This module stores the license candidate details. Admin people can update license details of students. It also maintains received license list, applied license list, and issued license list. It will be helpful to view whose license is applied and issued.

**11**

**Driver Module**

This module has the Driver details such as driver id, name, address, contact details, Department, qualification, salary, license no. and etc… Admin people can add, update and remove the driver details.

Here the attendance of the candidate will be updated. Using this module, the admin dept. may know the attendance and who are taking leave by date wise. This also helps to make examine of the attendance calculations.

**Payment Module**

This module maintains payment details of all students. Using this module, admin people can view payment details in daily basis, monthly basis. They can also modify these details.

**Expense Module**

This module maintains expense details. Using this module, admin people can view expense details in daily basis, monthly basis. They can also modify these details.

12

**SYSTEM TESTING AND IMPLEMENTATION**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

13

**5. SYSTEM TESTING AND IMPLEMENTATION:**

**5.1 SYSTEM TESTING**

System testing is process of exercising software with the intent of finding and ultimately correcting errors.

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

**The objective of testing is as follows:**

* Testing is the process of executing a program with the intent of finding an error.
* A successful test is that one of the cover of undiscovered error.

**TESTING ISSUSE:**

* Client GUI considerations.
* Target environment and platform diversity considerations.

**TESTING METHODOLOGIES:**

System testing is the state of implementation, which is aimed at ensuring that the system works accurately and efficiently as expect before live operation commences. It certifies that the whole set of programs hang together. System testing requires a test plan that consists of several key activities and steps for run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system.

14

Testing is an important stage in software development. The system test is implementation stage in software development. The system test in implementation should be confirmation that all is correct and an opportunity to show the users that the system works as expected. It accounts the largest percentage of technical effort in the software development process.

Testing phase in the development cycle validates the code against the functional specification. Testing is vital to the achievement of the system goals. The objective of testing is to discover errors. To fulfill this objective a series of test step unit, integration, validations and system tests were planned and expected. The test steps are,

**(i) UNIT TESTING:**

Unit testing focuses verification effort on the smallest unit of software each of the modules was verified individually for errors. This is known as module testing. The testing was carried out during programming stage itself, sample data is given for unit testing. The unit results are recorded for further reference.

**(ii) INTEGRATION TESTING:**

Software validation is achieved through a series of tests that demonstrates conformity with requirements. Thus the proposed system under consideration has been tested by validation and found to be working satisfactorily

**(iii) SYSTEM TESTING:**

This is to verify that all the system elements have been properly integrated and perform allocated functions. Testing executes a program to test the logic changes made in it and with intention of finding errors. Tests are also conducted to find discrepancies between system and its original objective, current specification and documents.

15

**5.2 SYSTEM IMPLEMENTATION**

Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

**The stage consists of**

* Testing the developed program with simple data
* Detection’s and correction of error
* Creating whether the system meets user requirements
* Testing whether the system
* Making necessary changes as desired by the user
* Training user personnel

**IMPLEMENTATION PROCESS:**

The implementation phase is less creative than system design. A system project may be dropped at any time prior to implementation, although it becomes more difficult when it goes to the design phase.

The final report to the implementation phase includes procedural flowcharts, record layouts, report layouts, and a workable plan for implementing the candidate system design into an operational one. Conversion is one aspect of implementation. Several procedures of documents are unique to the conversion phase. They include the following,

16

* The conversion portion of the implementation plan is finalized and approved.
* Files are converted.
* Parallel processing between the existing and the new system are logged on a special form.
* Assuming no problems, parallel processing is discontinued. Implementation results are documented for reference.
* Conversion is completed. Plans for the post-implementation review are prepared. Following the review, the new system is officially operational.

17

**CONCLUSION**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

18

**6. CONCLUSION:**

The **“ DRIVING SCHOOL MONITORING SYSTEM “** system is to be developed to replace such COMPUTER system in order to provide a better controlled and efficient environment which will meet the needs of the day’s services.

The candidate details are fully systemized so the updating and insertion process become easy .The processing time also minimized. License details received from the candidate to maintain the COMPUTER of course completion. Attendances of Driver were entered day by day it leads the process of calculating salary to the Driver.

The reports can be generated based on the requirement given by the admin and retrieve the same easily such as expenses, payments, Driver attendance status, candidate fees status, license issued details etc., the system helps the user to overcome the difficult of maintaining the ledger and notebooks.

19

**BIBLIOGRAPHY**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

20

**7. BIBILIOGRAPHY:**

**BOOK REFERENCES**

* Professional PHP6, By Ed Lecky-Thompson, Steven D. Nowak, and Thomas Myer
* Learning PHP, MYSQL, and JavaScript: A Step-by-Step Guide to Creating Dynamic Websites, By Robin Nixon
* PHP Solutions: Dynamic Web Design Made Easy, By David Powers
* Beginning PHP and MYSQL: From Novice To Professional, By W. Jason Gilmore
* Head First PHP & MYSQL, By Lynn Burghley and Michael Morrison
* Core PHP programming By Leon Atkinson, Zeev Suraski.
* PHP Object-Oriented Solutions By David Powers

**WEB REFERENCE**

* [www.w3schools.com](http://www.w3schools.com/)
* [www.php.net/manual/en/tutorial.php](http://www.php.net/manual/en/tutorial.php)
* [www.tutorialspoint.com](http://www.tutorialspoint.com/)
* [www.tizag.com/php](http://www.tizag.com/php)
* [www.codecademy.com/tracks/php](http://www.codecademy.com/tracks/php)

21

**APPENDICES**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

22

**8. APPENDICES**

**A. DATA FOLW DIAGRAM**

**LEVEL 0:**

Driving School

Admin

Report

**LEVEL 1:**

**LOGIN FORM:**

Admin

Create new login

Change password

Admin table

**23**

**CANDIDATE DETAIS:**

Candidate details

Delete

Update

Add

View

Candidate Table

24

**LEVEL 2:**

**LICENSE DETAILS:**

License details

Apply

License

Issue

License

View

License Table

25

**LEVEL 3:**

**PAYMENT DETAILS:**

Payment details

Update

Add

View

Delete

Payment Table

26

**LEVEL 4:**

**EXPENSE DETAILS:**

Expense details

Add

Update

View

Delete

Expense Table

27

**B.ER DIAGRAM**

**Table Name:** Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **CONSTRAINT** | **DESCRIPION** |
| Username | Varchar | 20 | Not null | Username |
| Password | Varchar | 20 | Not null | Password |

**Table Name:** Candidate

**Primary Key:** Can\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **CONSTRAINT** | **DESCRIPION** |
| Can\_id | Int | 20 | Primary key | Candidate Id |
| Can\_name | Varchar | 20 | Not null | Candidate name |
| Address | Varchar | 50 | Not null | Address |
| Age | Int | 2 | Not null | Age |
| Contact No. | Int | 10 | Not null | Contact Number |
| Mail\_Id | Char | 30 | Not null | Mail Id |
| Gender | Varchar | 10 | Not null | Gender |
| DOJ | Time\_Stamp | 19 | Not null | Date Of Joining |
| License | Varchar | 30 | Not null | Apply for license or not |

28

**Table Name:** License Apply

**Foreign Name:** Can\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **CONSTRAINT** | **DESCRIPION** |
| Can\_id | Int | 20 | Foreign Name | Candidate Id |
| AppDate | Time\_Stamp | 19 | Not null | Applied Date |

**Table Name:** License Issue

**Foreign Name:** Can\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **CONSTRAINT** | **DESCRIPION** |
| Can\_id | Int | 20 | Foreign Name | Candidate Id |
| IssDate | Time\_Stamp | 19 | Not null | Issue Date |

**Table Name:** Payment

**Foreign Name:** Can\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **CONSTRAINT** | **DESCRIPION** |
| Can\_id | Int | 20 | Foreign Name | Candidate Id |
| Pay Date | Time\_Stamp | 19 | Not null | Payment Date |
| Pay Amt | Int | 10 | Not null | Amount |
| Bill no. | Int | 10 | Not null | Bill Number |

**29**

**Table Name:** Driver

**Primary Key:Dri\_**id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **CONSTRAINT** | **DESCRIPION** |
| Dri\_id | Int | 20 | Primary key | Driver Id |
| Dri\_name | Varchar | 20 | Not null | Driver name |
| Address | Varchar | 50 | Not null | Address |
| Contact No. | Int | 10 | Not null | Contact Number |
| Mail\_Id | Char | 30 | Not null | Mail Id |
| Gender | Varchar | 10 | Not null | Gender |
| Designation | Varchar | 10 | Not null | Designation |
| DOJ | Time\_Stamp | 19 | Not null | Date Of Joining |
| Salary | Int | 10 | Not null | Salary |

**30**

**Table Name:** Attendance

**Foreign Name:** Can\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **CONSTRAINT** | **DESCRIPION** |
| Can\_id | Int | 20 | Foreign Name | Candidate Id |
| Add Date | Time\_Stamp | - | Not null | Date |
| Attendance | Varchar | 10 | Not null | Attendance |

**Table Name:** Expense

**Foreign Name:** Can\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **SIZE** | **CONSTRAINT** | **DESCRIPION** |
| Can\_id | Int | 20 | Foreign Name | Candidate Id |
| Date | Time\_Stamp | 19 | Foreign Name | Date |
| Particulars | Varchar | 30 | Not null | Particulars |
| Amount | Int | 10 | Not null | Amount |

31

1. **SAMPLE CODING:**

<?php include('loginserver.php'); ?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Login</title>

<link rel="stylesheet" type="text/css" href="StyleSheets/style.css">

</head>

<body>

<center>

<img class="logo" src="img/php.png" width="275px" height="125px">

</center>

<div class="header">

<h2>DRIVING SCHOOL MANAGEMENT SYSTEM</h2>

<h2>Login</h2>

</div>

<form method="post" action="login.php">

<?php include('errors.php'); ?>

<div class="input-group">

<label>Username</label>

<input type="text" name="uname" value="<?php echo($username); ?>">

</div>

<div class="input-group">

<label>Password</label>

<input type="Password" name="pword1">

</div>

<div class="input-group">

<button type="submit" name="login" class="btnlogin">Login</button>

32

</div>

<div class="input-group">

<p>Forgot your password?

<a href="fgpass.php">Click Here!!</a></p>

</div>

</form>

</body>

</html>

<?php include('loginserver.php'); ?>

<?php

$name="";

$errors = array();

$con=mysqli\_connect('localhost','root','','dsms');

if (isset($\_POST['submitu'])) {

$name=$\_POST['namesearch'];

if (Drity($name)) {

array\_push($errors, "Enter name to filter");

}

if (count($errors)==0) {

$resu=mysqli\_query($con,"SELECT \* FROM candidate WHERE name='$name'");

}

else {

$resu=mysqli\_query($con,"SELECT \* FROM candidate");

}

}

else {

33

$resu=mysqli\_query($con,"SELECT \* FROM candidate");

}

if (isset($\_POST['clearu'])) {

$resu=mysqli\_query($con,"SELECT \* FROM candidate");

}

if (isset($\_GET['deluser'])) {

$id=$\_GET['deluser'];

mysqli\_query($con,"DELETE FROM candidate WHERE id=$id");

header("Location: candidate.php");

}

if (isset($\_GET['exit'])) {

if (isset($\_SESSION['username'])) {

$username=$\_SESSION['username'];

$query="select \* from login where username='$username'";

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

while ($row=mysqli\_fetch\_array($result)) {

if ($row['username']==$username && $row['type']=='Admin') {

header("Location: admin.php");

}

}

}

}

?>

34

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Candidate List</title>

<link rel="stylesheet" type="text/css" href="StyleSheets/stylecandidate.css">

</head>

<body>

<form method="POST" action="candidate.php" class="candi">

<?php if (isset($\_POST['submitu'])): ?>

<?php include('errors.php'); ?>

<?php endif ?>

<br>

<label>Name:</label>

<input type="text" name="namesearch" class="namesearch" value="<?php echo $name; ?>">

<button type="submit" class="date\_btn" name="submitu">Submit</button>

<button type="submit" class="date\_btn" name="clearu">Clear</button>

</form>

<table>

<thead>

<tr>

<th>S.No</th>

<th>Name</th>

<th>Address</th>

<th>Age</th>

<th>Contact No</th>

<th>Email</th>

<th>Gender</th>

<th>DOJ</th>

<th colspan="2">Action</th>

</tr>

35

</thead>

<tbody>

<?php $i=1; while ($row=mysqli\_fetch\_array($resu)) {

$date=date\_create($row['doj']); ?>

<tr>

<td><?php echo($i); ?></td>

<td><?php echo $row['name']; ?></td>

<td><?php echo $row['address']; ?></td>

<td><?php echo $row['age']; ?></td>

<td><?php echo $row['phone']; ?></td>

<td><?php echo $row['email']; ?></td>

<td><?php echo $row['gender']; ?></td>

<td><?php echo date\_format($date,"d-M-y"); ?></td>

<td><a href="candidateedit.php?id=<?php echo($row['id']); ?>">Update</a></td>

<td><a href="candidate.php?deluser=<?php echo($row['id']); ?>">Delete</a></td>

</tr>

<?php $i++; } ?>

</tbody>

</table>

<button type="submit" class="btnclose" name="exit"><a href="candidate.php?exit='1'">Exit</a></button>

</body>

</html>

<?php include('loginserver.php'); ?>

<?php

$name="";

$errors = array();

$con=mysqli\_connect('localhost','root','','dsms');

36

if (isset($\_POST['submitu'])) {

$name=$\_POST['namesearch'];

if (Drity($name)) {

array\_push($errors, "Enter name to filter");

}

if (count($errors)==0) {

$resu=mysqli\_query($con,"SELECT \* FROM payment WHERE name='$name'");

}

else {

$resu=mysqli\_query($con,"SELECT \* FROM payment");

}

}

else {

$resu=mysqli\_query($con,"SELECT \* FROM payment");

}

if (isset($\_POST['clearu'])) {

$resu=mysqli\_query($con,"SELECT \* FROM payment");

}

if (isset($\_GET['exit'])) {

if (isset($\_SESSION['username'])) {

$username=$\_SESSION['username'];

$query="select \* from login where username='$username'";

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

while ($row=mysqli\_fetch\_array($result)) {

if ($row['username']==$username &&

37

$row['type']=='Admin') {

header("Location: admin.php");

}

}

}

}

?>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Payment Details</title>

<link rel="stylesheet" type="text/css" href="StyleSheets/stylecandidate.css">

</head>

<body>

<form method="POST" action="paymentdetail.php" class="candi">

<?php if (isset($\_POST['submitu'])): ?>

<?php include('errors.php'); ?>

<?php endif ?>

<br>

<label>Name:</label>

<input type="text" name="namesearch" class="namesearch" value="<?php echo $name; ?>">

<button type="submit" class="date\_btn" name="submitu">Submit</button>

<button type="submit" class="date\_btn" name="clearu">Clear</button>

</form>

<table>

<thead>

<tr>

38

<th>S.No</th>

<th>Candidate Name</th>

<th>Bill No</th>

<th>Bill Date</th>

<th>Amount</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

<?php $i=1; while ($row=mysqli\_fetch\_array($resu)) {

$billdate=date\_create($row['billdate']); ?>

<tr>

<td><?php echo($i); ?></td>

<td><?php echo $row['name']; ?></td>

<td><?php echo $row['bill']; ?></td>

<td><?php echo date\_format($billdate,"d-M-y"); ?></td>

<td><?php echo $row['amount']; ?></td>

<td><a href="paymentedit.php?id=<?php echo($row['id']); ?>">Update</a></td>

</tr>

<?php $i++; } ?>

</tbody>

</table>

<button type="submit" class="btnclose" name="exit"><a href="paymentdetail.php?exit='1'">Exit</a></button>

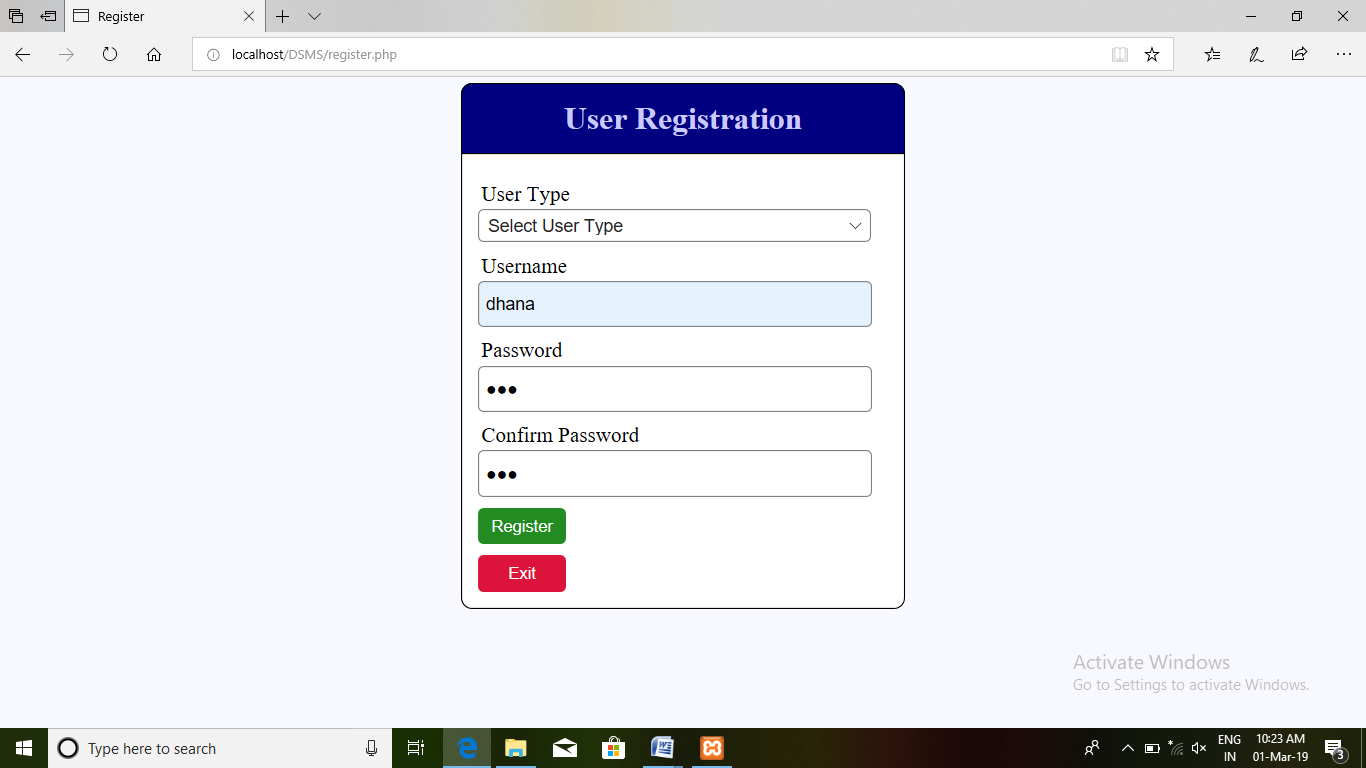
</body>

</html>

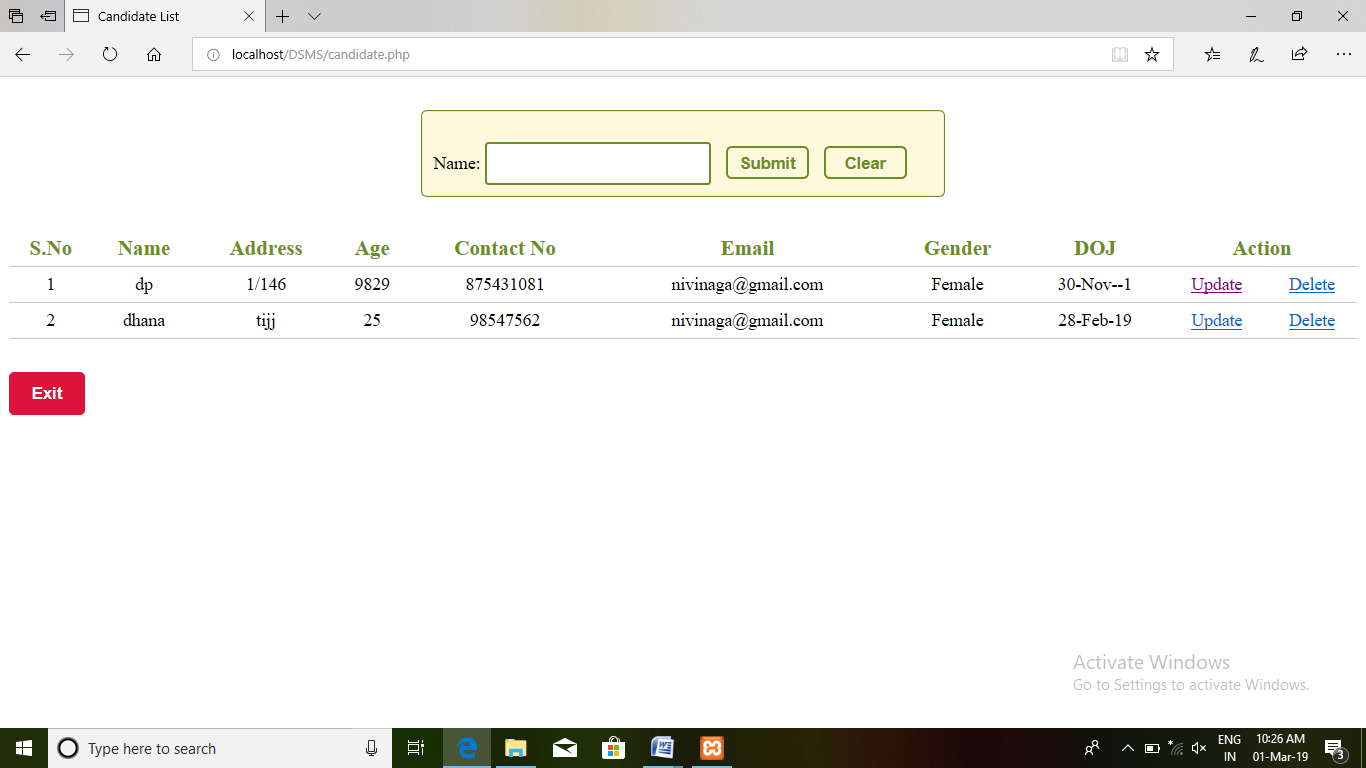
**39**

1. **SAMPLE OUTPUT:**

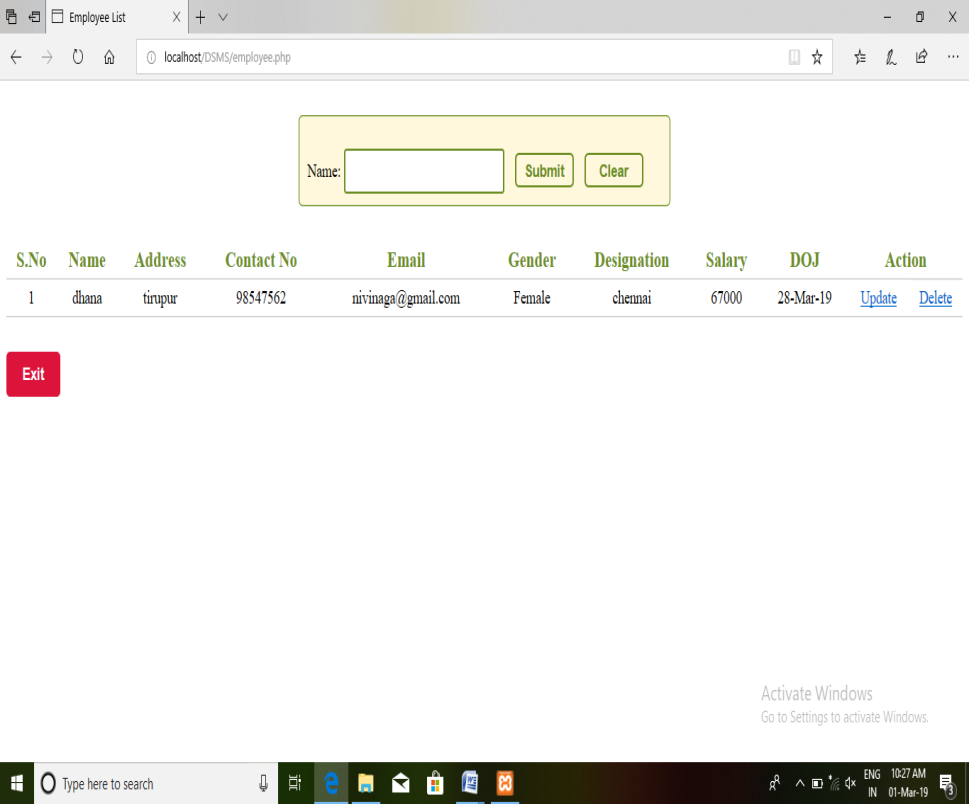
**Login From:**



40

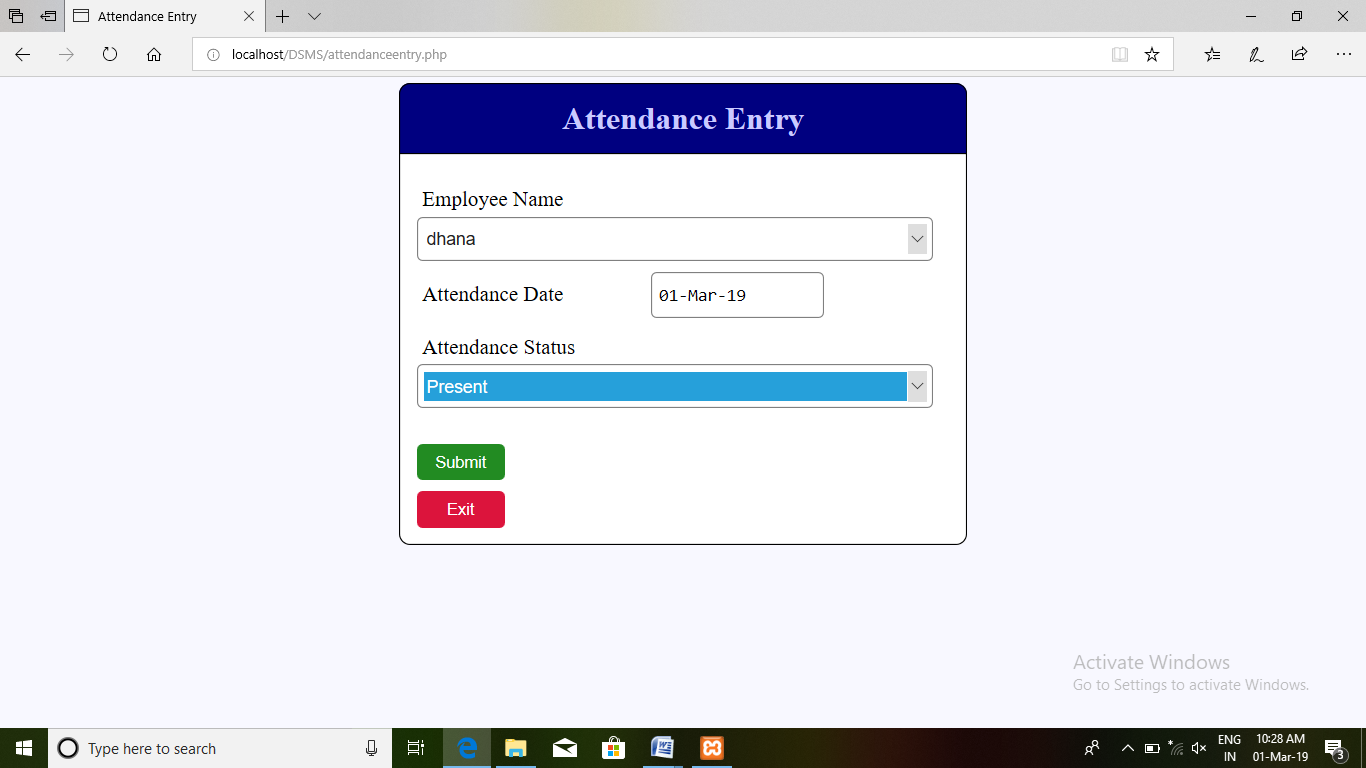
**Candidate Entry:**

**Driver Entry:**

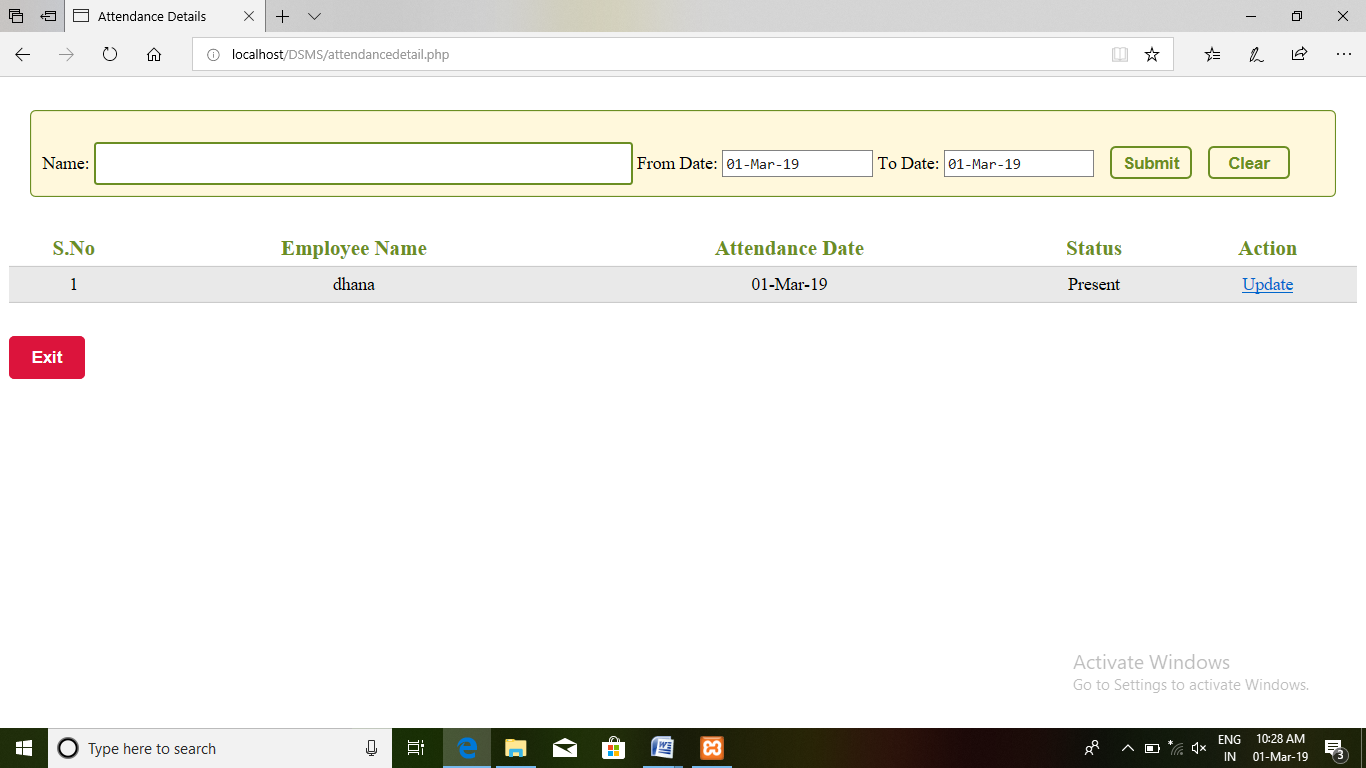


41

**Attendance Entry:**

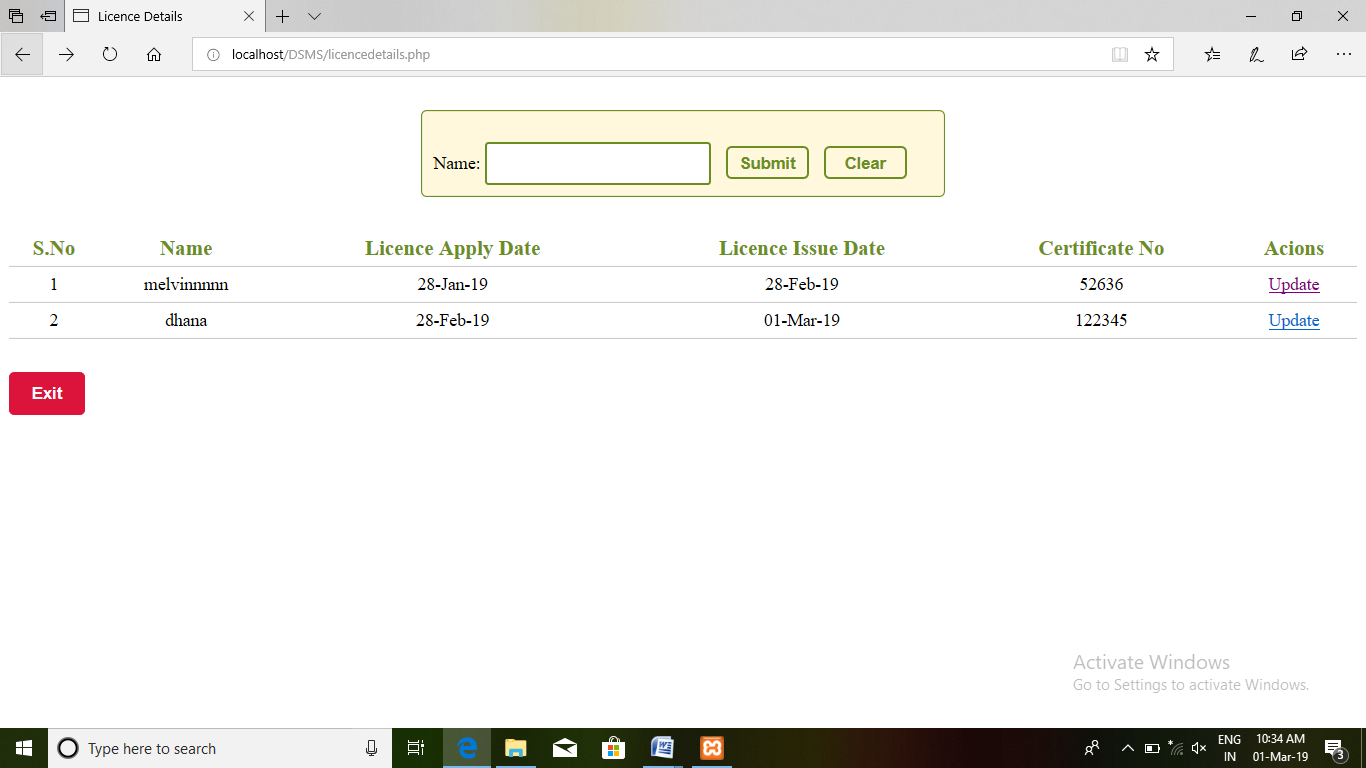


**Attendance Details:**

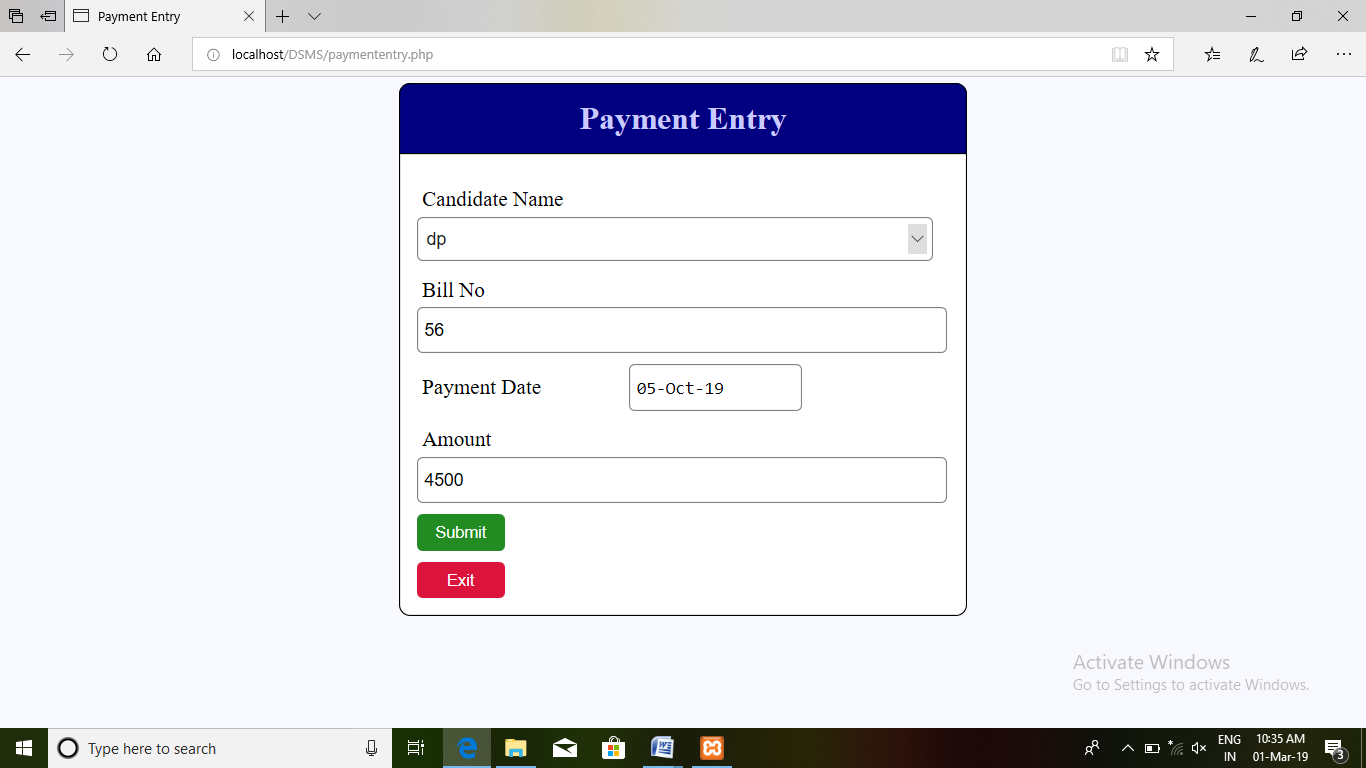


42

**License Details:**

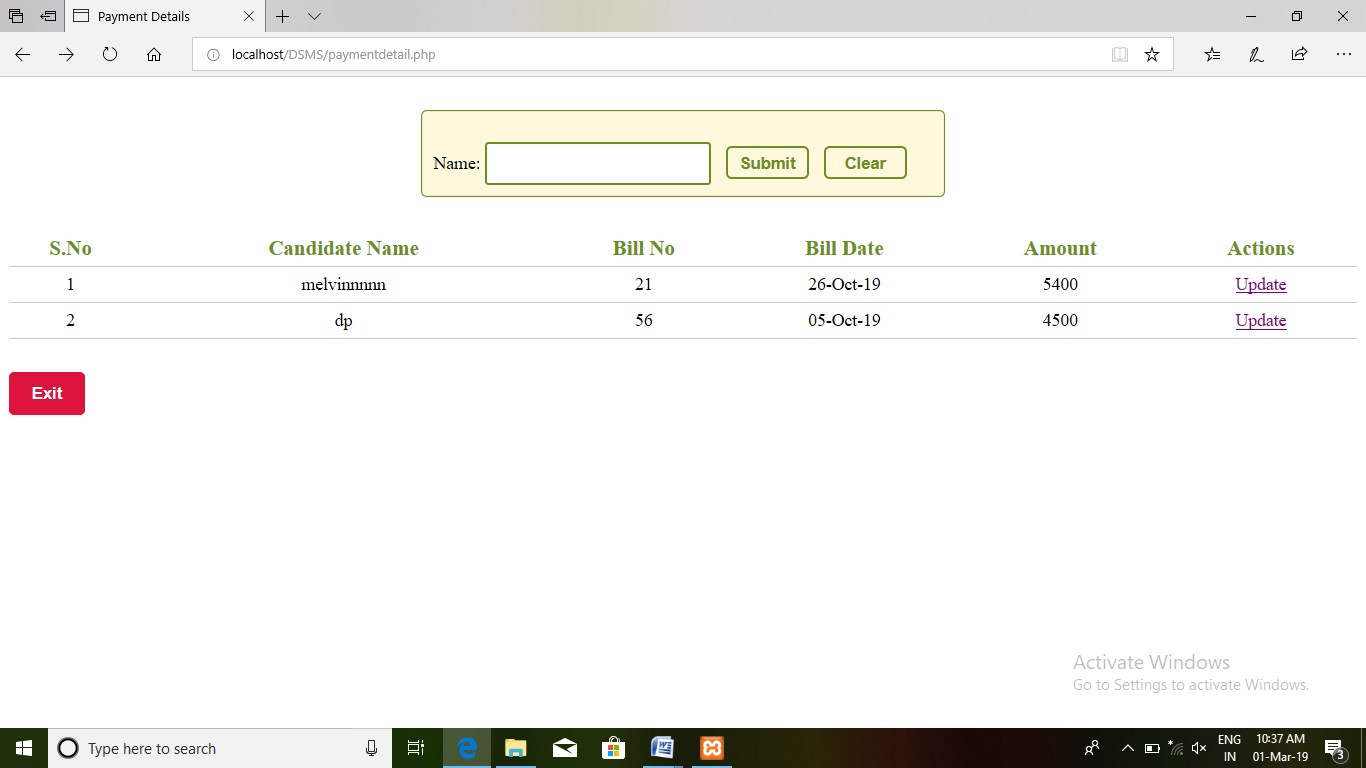


**Payment Entry:**

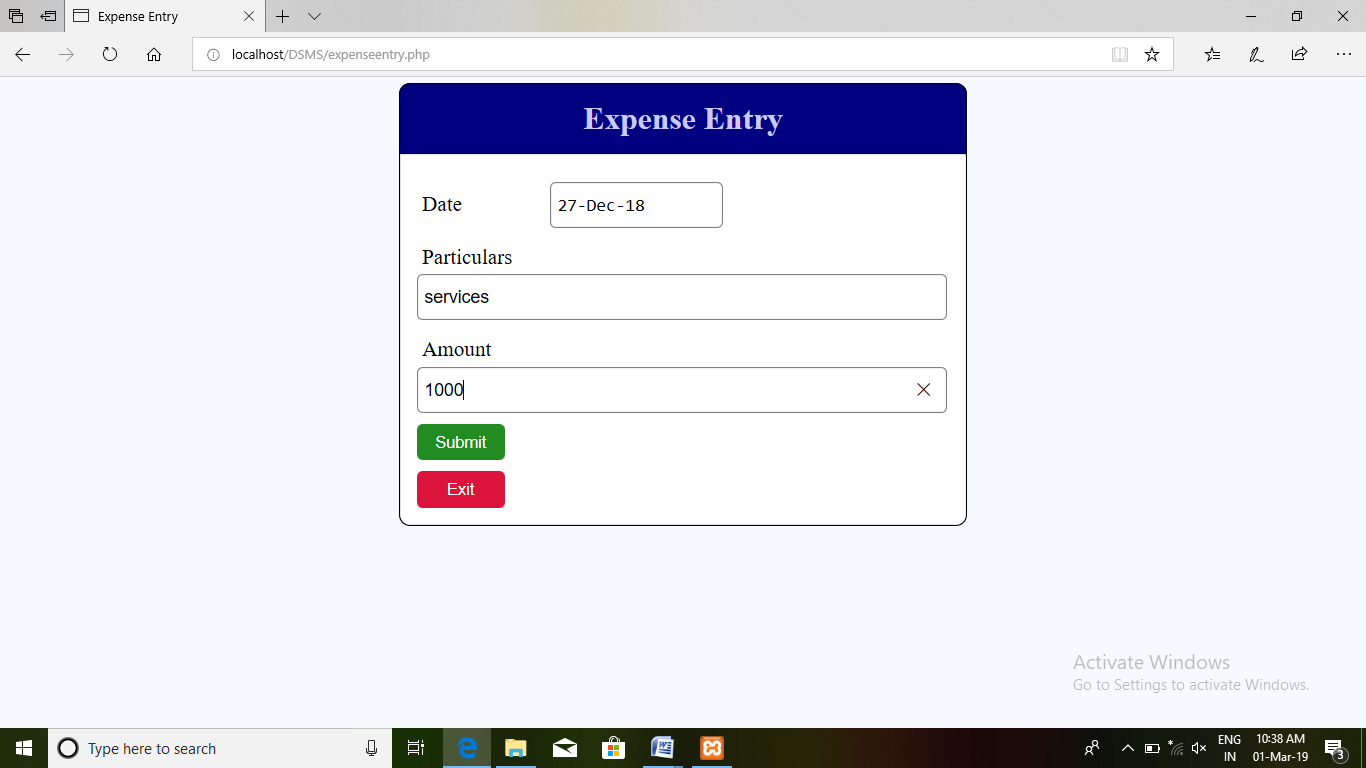


43

**Payment Details:**



**Expense Entry:**



44