**CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **Contents** | | **Page No.** |
| **1.** | **Synopsis** | | 5 |
| 1.1 | Title of the project | 5 |
| 1.2 | Objectives | 5 |
| 1.3 | Project Category | 5 |
| 1.4 | Structure of the Project | 5 |
| 1.5 | Hardware Interface | 6 |
| 1.6 | Software Interface | 6 |
| 1.7 | Module description | 6 |
| 1.8 | Methodology | 8 |
| 1.9 | Future scope of the project | 8 |
| **2.** | **Literature Survey** | | 9 |
| 2.1 | PHP | 10 |
| 2.2 | My SQL | 10 |
| 2.3 | Java Script | 11 |
| 2.4 | CSS 3 | 11 |
| 2.5 | HTML 5 | 11 |
| 2.6 | XAMPP | 12 |
| **3.** | **Software Requirement Specification** | | 13 |
| 3.1 | Introduction | 14 |
| 3.1.1 | Purpose | 14 |
| 3.1.2 | Scope | 14 |
| 3.1.3 | Definition, Acronyms, Abbreviations | 15 |
| 3.1.4 | Project Overview | 15 |
| 3.2 | Overall Description | 15 |
| 3.2.1 | Product Perspective | 15 |
| 3.2.2 | Product Function | 15 |
| 3.2.3 | User Characteristics | 16 |
| 3.2.4 | General Constraints | 16 |
| 3.2.5 | Assumption and Dependencies | 16 |
| 3.3 | External Interface Requirements | 17 |
| 3.3.1 | User Interface | 17 |
| 3.3.3 | Hardware Interface | 17 |
| 3.3.4 | Software interface | 17 |
| 3.3.5 | Functional Requirements | 18 |
| **4.** | **System Design** | | 21 |
| 4.1 | Introduction | 22 |
| 4.1.1 | Purpose | 22 |
| 4.2 | Description of Program | 23 |
| 4.2.1 | Context Flow Diagram | 23 |
| 4.2.2 | Data Flow Diagram | 24 |
| 4.2.3 | Rules for Constructing DFD | 25 |
| **5.** | **Database Design** | | 30 |
| 5.1 | Introduction | 31 |
| 5.2 | Schema Description | 31 |
| 5.3 | Entity Relation Diagram | 35 |
| **6.** | **Detailed Design** | | 37 |
| 6.1 | Introduction | 38 |
| 6.2 | Structured English | 39 |
| **7.** | **System Coding** | | 44 |
| 7.1 | Introduction | 45 |
| 7.2 | Module Coding | 45 |
| **8.** | **System Testing** | | 88 |
| 8.1 | Introduction | 88 |
| 8.2 | Level of Testing | 88 |
| 8.2.1 | Unit Testing | 88 |
| 8.2.2 | Code walkthrough |  |
| 8.2.3 | Integrated Testing | 89 |
| 8.2.4 | Output Testing | 89 |
| 8.2.5 | System Testing | 89 |
| 8.3 | Test Cases | 89 |
| **9.** | **Snapshots** | | 95 |
| **10.** | **Conclusion and Future Enhancement** | | 101 |
| 10.1 | Conclusion | 102 |
| 10.2 | Future Scope of the project | 102 |
| **11.** | **Limitations** | | 103 |
| **12.** | **Bibliography** | | 105 |
| 12.1 | Book References | 106 |
| 12.2 | Web Reference | 106 |

**SYNOPSIS**

**CHAPTER 1**

**SYNOPSIS**

**1.1 Title of the Project: “PAY&PARK”.**

Now a day in many public places such as malls, hospitals, offices, market areas there is a crucial problem of vehicle parking.

The main purpose of this project is to automate all possible functionalities of the parking management. The development of new system contains advanced features. It enables employees and administrators to log into their account through online. Even the customer can view parking slots through online. In this system employees and administrators can view the details about parking records, branch records vehicle types etc.

**1.2Objectives of the Project:**

* The main objective of this project is to introduce an easy and modern parking system which helps customers as well as operators.
* We make this system for Private space owners, malls, hotels to calculate vehicle parking bill by entering in-time and out-time.
* Even this system helps customer to check nearest parking space and available parking slots.

**1.3Project Category:**

Web Based Application

**1.4 Structure of the project:**

This system can be used to store the parking record of a vehicle parked in a parking location and generate the fees payable by the customer.

**Tools/Platform:**

|  |  |
| --- | --- |
| **Tools** | **Platform** |
| **Front End** | **PHP** |
| **Back End** | **MYSQL Server** |

**1.5 Hardware Interface:**

Processor : Intel Dual Core or above

Operating System : windows 7 or above

Processer Speed : 1.9 GHZ or above

RAM : 512MB RAM

Hard disk : 40GB Hard disk

**1.6 Software Interface:**

Software requirements are used in operating system.

* Front end tool PHP.
* Back end MYSQL Server.
* XAMPP 1.8.2
* Apache server
* Dream viewer

**1.7Module Description:**

Initially user must login by entering the user name and password. The project consists of the following modules.

* **Log in module**:

In this module, the Administrators and the employees can log in to their dashboards and manage their accounts.

* + Log in module.
* **Dashboards:**

In this module, the Administrators and Employees can manage their profiles change their passwords view the records.

* Dashboard
* Employee account
* Change password module
* Add admin
* Add branch
* Add employee
* Add parking cost
* Add parking locations
* Add parking slots
* Add vehicle types
* View admin
* View branch
* View employee
* View parking cost
* View parking locations
* View parking slots
* View vehicle type
* **Add parking records:**

In this module, the employees can add the parking records by selecting vehicle types after selecting vehicle types loading available parking slots then adding the in time of the vehicle and when the vehicle is exiting the are adding the out time

This generates a bill.

* Add parking records
* View parking records
* **Report module:**

This module is for administrator where admin can view various kinds of report. Admin can view:

* Branch wise report
* Vehicle parking report
* Parking income report
* Employee wise report
* **Voucher Module:**

This module, a special voucher is created for the customers to avail discounts for the frequently visited customers.

* + Add customer
  + View customer
  + Add voucher
  + View voucher
  + View voucher

**1.8 Methodology:**

It determines the sequential to the software development that begins at the system level and progress through analysis design, coding, and testing.

**1.9 Future scope of the project:**

We can develop and integrate with Android or iPhone based apps

in the future.

We can integrate a Card payment system in the future.

**LITERATURE SURVEY**

**L**

**CHAPTER-2**

**LITERATURE SURVEY**

* 1. **PHP:**
* PHP stands for Hypertext Pre-Processor.
* PHP is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into html.
* Originally created by Rasmus Lerdorf in 1994.
* PHP code may be embedded into HTML code, or it can be used in combination with various web template systems and web frameworks.
* PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executables.
* The web server combines the results of the interpreted and executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.
  1. **MySQL:**
* Myself is an open-source relational database management system (RDBMS).
* Its name is a combination of “My”, the name co-founder Michael Wideners and “SQL” the abbreviation for structured query language.
* Information in the MySQL database are stored in the form of related tables.
* MySQL database are typically used for web application development.
* A MySQL database can be accessed directly using C, C++, Eiffel, Java, Perl, PHP and Python computer languages.
* MySQL database queried using a subset of the standard structured query language (SQL) commands.
  1. **JavaScript:**

JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and later the document content that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications. JavaScript is a prototype-based language with dynamic typing and has first-class functions.

Its syntax is influenced by C. JavaScript copies many names and naming conversions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

* 1. **Cascading Style Sheet 3 (CSS 3):**

CSS is a style sheet language used to describe the presentation semantics (that is, the look and formatting) of a document written in markup language (Style defines how to display HTML elements). Its most common application is to style web page written in HTML and any kind of XML documents. Cascading Style Sheets (CSS) are widely recognized for their contribution in building fact-loading, standards compliant, easily modifiable web pages. External Style Sheets are stored in CSS file CSS 3 is the latest standards for CSS. CSS 3 is completely backwards-compatible with earlier versions of CSS. The online Blood Management System uses the CSS3 specifications.

* 1. **HTML5:**
* HTML stands for Hyper Text Markup Language.
* A markup language is a set of markup tags.
* HTML markup tags are usually called as HTML tags.
* HTML tags are keywords surrounded by angle brackets like <html>.
* HTML document contain HTML tags and plain text.
* HTML documents are also called as Web Pages.

HTML5 is a core technology markup language of the Internet used for structuring and presenting content for the World Wide Web. As of October 2014, this is the final and complete fifth revision of the HTML standard of the World Wide Web Consortium (W3C). The previous version, HTML 4, was standardized in 1997. Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices (web browsers, parsers, etc.).

**2.6 XAMPP:**

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, Maria DB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), Maria DB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (Maria DB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

SS

**SOFTWARE REQUIREMENT SPECIFICATION**

**CHAPTER 3**

**SOFTWARE REQUIREMENT SPECIFICATION**

**3.1 Introduction:**

Software Requirement Specification (SRS) is the starting point of the software development activity. The main aim of SRS is to bridge the communication gap between the client and developer. SRS is medium through which client and user needs are accurately specified. The client usually does not understand software or software development process and developer doesn’t understand the client problem. Developing an SRS is to fulfill client

Needs. An SRS provides a reference for the validation of the final product that is the SRS helps the client to determine if the system meets the requirement.

**3.1.1 Purpose:**

The purpose of this document is to specify requirement and give guidelines for the development of ONLINE PAY&PARK MANAGEMENT system. This document aims to give brief description about the vehicle parked and it’s owner. It provides the visibility in the design and provides information needed for software support.

**3.1.2 SCOPE:**

The scope of the document is to provide sufficient detail functional and

non- functional requirements of the project to see that all the functionalities are properly implemented. Any changes made to the requirement in the figure will have to go through a formal approval process.

**3.1.3 Definition, Acronyms, Abbreviations:**

|  |  |
| --- | --- |
| Acronym | Definition |
| DFD | Data Flow Diagram |
| GUI | Graphical User Interface |
| PHP | Hypertext Pre-Processor |
| ER | Entity Relation |
| XAMPP | It is a software which install PHP and MySQL |
| Html | Hypertext markup language |
| MySQL | Database server |

**3.1.4 Project Overview:**

This project is related to manage the parking system. Which makes easier to maintain the record of the parked vehicle and his owner. This SRS is organized in a way that any user can easily understand and use the Pay&Park management system.

**3.2 Overall description:**

The overall description of our project can be stated as creating and managing the database, developing friendly user interface to manipulate the database, provide an authentication mechanism to safely accomplish tasks mentioned above. This section of the SRS describes all general factors of the product and its requirement.

**3.2.1 Product Perspective:**

Product of this project software, which replace all the records that are handled manually by Pay&Park management system. There are many records such as parking records, admin, employee, branch records etc. Handling all the records is a difficult task for the user this product provides easy way for maintaining records.

**3.2.2 Product Function:**

The program is based on application meaning it runs on windows environment product can store the data and should be able to generate reports on user request and should be able to print the information.

**3.2.3 User Characteristics:**

* Administrator: has all the privileges like making data entry. Admin is the main user of the website who controls complete website with full authority.
* Employee: adding parking records and viewing other records.
* Customer: only view the records.

**3.2.4 General Constraints**

* The application should be 32-bit/64-bit windows application.
* All definable options shall have default values supplied by the application.
* The system shall detect error in case any occur.
* The alteration to the application shall be done using Microsoft SQL

Server 2008 and later versions.

* The application shall be installable on any windows computer**.**
* No error condition shall cause the application to exit premature.

**3.2.5 Assumption and dependencies:**

* The user should have the basic knowledge of computer.
* Only the administrator will administer the system.
* It can be accessed over the internet
* Employee can only view the records.

**SPECIFIC REQUIREMENTS**

**3.3 External interface requirements:**

**3.3.1User interface:**

User interface is graphical user interface consisting of tabs and forms. The user interface must be simple and easy to use. All interactions of the software with the people. Hardware should be clearly specified. This software provides GUI, there are command button, textboxes, labels and data grids are used.

**3.3.2 Hardware Interface:**

Processor : Intel Dual Core or above

Operating System : windows 7 or above

Processer Speed : 1.9 GHZ or above

RAM : 1GB RAM

Hard disk : 40GB Hard disk

**3.3.4Software Interface:**

Software requirements are used in operating system.

* Front end tool PHP.
* Back end MYSQL Server.
* XAMPP 1.8.2
* Apache server
* Dream viewer
* Design Interface HTML, CSS, AJAX.

**3.3.5 Functional Requirements:**

The functions of the form used in the software are as follows. It specifies which output should be produced from the given inputs, they describe the relationship between the input and output of the system. All operation to be performed on the input data to obtain the output should be specified. Unimportant part of the specifications is the system behavior in the abnormal situation like invalid input or error during the computations. the functional requirements must clearly state what the system should do if such situations occur. This includes specifying the validity checks on the input and output design.

The system has two users. Administrator and donor.

* **Administration has the following privileges**:

Adding another admin

Add branch

Add employee

Add parking cost

Add parking locations

Add parking slots

Add vehicle types

Add customer

Add voucher

**Modules:**

The project consists of the following modules:

* **Log in module.**

In this module, the Administrators and the employees can log in to their dashboards and manage their accounts.

* + Log in module.
* **Dashboards.**

In this module, the Administrators and Employees can manage their profiles change their passwords view the records.

* Dashboard
* Employee account
* Change password module
* Add admin
* Add branch
* Add employee
* Add parking cost
* Add parking locations
* Add parking slots
* Add vehicle types
* View admin
* View branch
* View employee
* View parking cost
* View parking locations
* View parking slots
* View vehicle types
* **Add parking records:**

In this module, the employees can add the parking records by selecting vehicle types after selecting vehicle types loading available parking slots then adding the in time of the vehicle and when the vehicle is exiting the are adding the out time

This generates a bill.

* Add parking records
* View parking records
* **Report module:**

This module, is for administrator where admin can view various kinds of report.

Admin can view:

* Branch wise report
* Vehicle parking report
* Parking income report
* Employee wise report
* **Voucher Module**

This module, a special voucher is created for the customers to avail discounts for the frequently visited customers.

* + Add customer
  + View customer
  + Add voucher
  + View voucher

S

**SYSTEM DESIGN**

**Chapter 4**

**SYSTEM DESING**

**4.1Introduction:**

System design is solution to the creation of a new system. This important phase is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in feasibility study. Emphasis is on translating the performance requirements into design specifications.

Design goes through logical and physical development. Logical designs preview the present physical system input and output specifications and details the implementations plan and internal logic of each of the modules. here it should be ensured that the design is technically sound with minimum of redundancy and maintains. The physical design maps out the details of the physical system, plans the system and specifies any hardware and software.

**4.1.1Purpose:**

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the step 1 moving problem domain to the solution domain. It is also a bridge between requirements specifications and final solution domain. For satisfying requirement, the design of the item is perhaps the most critical factor, which affects the quality of the final product it also has very major effects on the later phases. Such as testing as well as maintenance.

Software design is a multistep process that focuses on 4 distinct attributes of the program.

* Data structure
* Software architecture
* Interface presentations
* Procedural details

The design process translates the requirement into representations of the software that can be assessed for quality, modularity and concept of abstraction enable the designer to simplify and reuse the software components, program and data structure contributes to the overall view of software architecture while procedure provides the details necessary for the algorithm segmentation.

Design describes the final system and process by which it is developed. It refers to the technical Specification that will be applied in implementing the candidate system it also includes the construction of the problem and testing. The various steps in design make the phase more flexible and valuable.

The goal of the design phase is to produce a model or representation of a system which can be used later to build that system. The produced model is called as the design the two most important properties that concern designs are efficiency and simplicity.

**4.2Description of program:**

**4.2.1Context flow diagram(CFD):**

The context flow diagram for basic accounting software system is shown in the figure. The input of this action is shown in this diagram. However, the number of details about the function of the basic accounting software system is given below. Using this as starting point a CFD of the system is given



**4.2.2Data flow diagram:**

The data flow diagram describes the flow of data, with the help of various levels in a crystal-clear way:

The DFD serves two purposes:

* To provide an indication of flow data are transformed as they move through the system.
* To depict the function that transforms the data flow

It provides information that is used during the analysis of the information domain and serves as a basis for the modeling function. The DFD is also known as data flow graph.

The DFD may be used to represent a system or software at any level of abstraction. In fact, DFD may be partitioned into levels that represents increasing information flow and function details. Therefore, the DFD provides a mechanism for functional modeling as well as information flow modeling.

A level 0 DFD also called a fundamental system model or a context model represents the entire software element as a single bubble with input and output data indicated incoming and outgoing arrows, respectively.

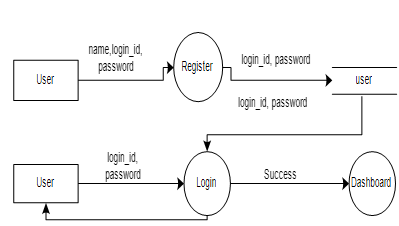
Additional processes (bubbles) and information flow paths are represented as the levels 0.

DFD is partitioned to reveal more details. For e. g: level 1 DFD might contain four or five bubbles with interconnecting arrows. Each of the processes represented a level 1 is sub function of the overall system depicted in the context model. The basic notation used to create a DFD makes it easy to analyze and understand. The DFD is a graphical tool that can be very valuable during software requirement analysis.

|  |  |
| --- | --- |
| NOTATION | DESCRIPTION |
|  | A circle represents a process or transform that is applied to data or control and changes it on some way. |
|  | A rectangle is used to represents an external entity, that is, a system element or another system that produces information for transformation by the software or receives information produced by the software. |
|  | An arrow represents one or more data items or data objects. |

**4.2.3Rules for constructing DFD:**

* Process should be named for easy understanding.
* The direction of flow is from top to bottom and left to right.
* The direction of should not have any kind of loops.

**DFD Level 1:**

**DFD Level 2:**

**DFD Level 3:**

**DFD Level 4:**

**DFD Level5:**

**DFD Level 6:**

**DATA BASE DESIGN**

d

**CHAPTER 5**

**DATABASE DESIGN**

**5.1Introduction:**

A database is an inherent collection of data with some inherent meaning designed, built populated with data for a specific purpose. The following guideline is being following during the database design.

* Descriptive names for the tables, columns and indexes.
* Singular names for tables and columns.
* Proper data type each column.

**5.2 Schema Description:**

**Admin:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Description** | **Constraints** |
| Admin id | Int (10) | Admin id | Primary key |
| Admin name | Varchar (25) | Admin name | Not null |
| Login id | Varchar (25) | Login id | Not null |
| Password | Varchar (100) | Password | Not null |
| Status | Varchar (10) | Active or inactive | Not null |

**Branch:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Description** | **Constraints** |
| Branch id | Int (10) | Branch id | Primary key |
| Branch name | Varchar (25) | Branch name | Not null |
| Branch details | Text | Branch details | Not null |
| address | Text | Address | Not null |
| Contact no | Varchar (15) | Contact no | Not null |
| Status | Varchar (10) | Active or Inactive | Not null |

**Customer:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Description** | **Constraints** |
| Customer id | Int (10) | Customer id | Primary key |
| Customer name | Varchar (25) | Customer name | Not null |
| Mobile no | Varchar (10) | Mobile no | Not null |
| address | Text | Address | Not null |
| Vehicle no | Int (10) | Vehicle no | Foreign key |

**Employee:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Description** | **Constraints** |
| Employee id | Int (10) | Employee id | Primary key |
| Employee name | Varchar (25) | Employee name | Not null |
| Branch id | Int (10) | Branch id | Foreign key |
| Login id | Varchar (20) | Login id | Not null |
| password | Varchar (100) | Password | Not null |
| mob no | Varchar (15) | Mobile no | Not null |
| Status | Varchar (10) | Active or Inactive | Not null |

**Parking cost:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Description** | **Constraints** |
| Parking cost id | Int (10) | Parking cost id | Primary key |
| Parking slot id | Int (10) | Parking slot id | Foreign key |
| No of hours | Int (10) | No of hours | Foreign key |
| Cost | Float (10,2) | Cost | Foreign key |
| status | Varchar (10) | Active or Inactive | Not null |

**Parking location:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Description** | **Constraints** |
| Parking location id | Int (10) | Parking location id | Primary key |
| Parking location | Varchar (10) | Parking location | Not null |
| Branch id | Int (10) | Branch id | Foreign key |
| Location image | Varchar (10) | Location image | Not null |
| Location address | Text | Location address | Not null |
| status | Varchar (10) | Status | Not null |

**Parking record:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Description** | **Constraints** |
| Parking record id | Int (10) | Parking record id | Primary key |
| Receipt no | Int (10) | Receipt no | Foreign key |
| Bill no | Int (10) | Bill no | Foreign key |
| In date time | Date time | In date time | Not null |
| Out date time | Date time | Out date time | Not null |
| Vehicle type id | Int (10) | Vehicle type id | Foreign key |
| Parking location id | Int (10) | Parking location id | Foreign key |
| Parking slot id | Int (10) | Parking slot id | Foreign key |
| Parking cost id | Int (10) | Parking cost id | Foreign key |
| Employee id | Int (10) | Employee id | Foreign key |
| Total cost | Float (10,2) | Total cost | Foreign key |
| Vehicle no | Varchar (25) | Vehicle no | Not null |

**Parking slot:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Description** | **Constraints** |
| Parking slot id | Int (10) | Parking slot id | Primary key |
| Vehicle type id | Int (10) | Vehicle type id | Foreign key |
| Parking location id | Int (10) | Parking location id | Foreign key |
| Number of slots | Int (10) | Number of slot | Foreign key |
| status | Varchar (10) | Active or Inactive | Not null |

**Vehicle type:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Description** | **Constraints** |
| Vehicle type id | Int (10) | Vehicle type id | Primary key |
| Vehicle type | Varchar (25) | Vehicle type | Not null |
| Vehicle icon | Varchar (100) | Vehicle icon | Not null |
| status | Varchar (10) | Active or Inactive | Not null |

**Voucher:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Description** | **Constraints** |
| Voucher id | Int (10) | Voucher id | Primary key |
| Customer id | Varchar (10) | Customer id | Foreign key |
| Voucher no | Varchar (10) | Voucher no | Foreign key |
| status | Varchar (10) | Active or Inactive | Not null |

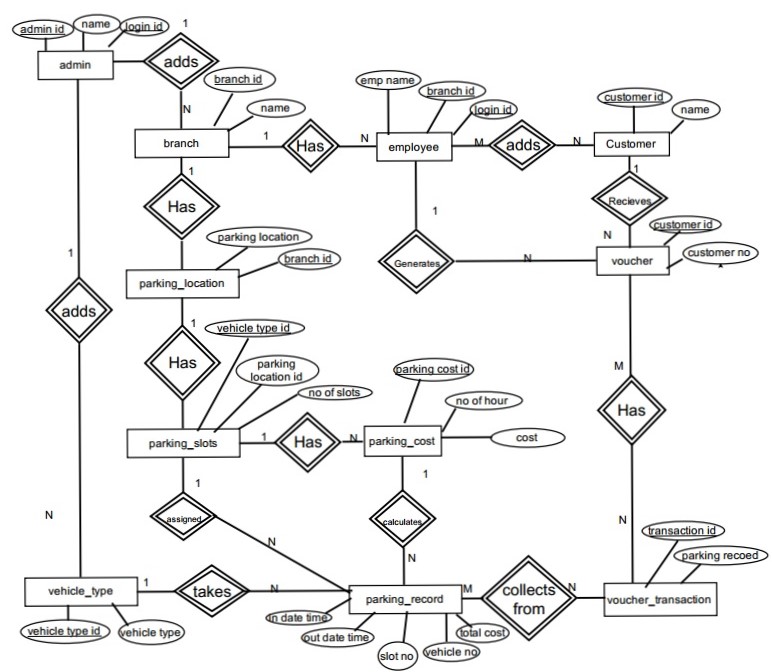
**Voucher transaction:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Type** | **Description** | **Constraints** |
| Voucher transaction id | Int (10) | Voucher transaction id | Primary key |
| Transaction type | Varchar (10) | Transaction type | Not null |
| Transaction amt | Float (10,2) | Transaction amount | Foreign key |
| Trans date | Date | Transaction date | Not null |
| Parking record id | Int (10) | Parking record id | Foreign key |
| Note | Text | Note | Not null |
| Status | Varchar (10) | Active or Inactive | Not null |

**5.3 Entity Relationship (ER) Diagram:**

Entity Relationship model is popular high level conceptual data model and its variations are frequently used for the conceptual design of the database applications and may be used for the conceptual design of the database design tools employ its concepts. We described the basic data structuring concepts and data constraints of the ER model and discuss their use in the design of conceptual schemas for database application. We also represent the diagrammatic notations known as ER diagrams.

The focus of ER modeling is the data items in the system and the relationship between them. It aims to create an ER model for the data from the user perspective. The sentence later can be used during the development of the database and there are methods that use the ER model to design database for the different database modules are frequently represented as ER diagram through the model can also be represented in mathematical forms**.**

****

**DETAILED DESIGN**

**Chapter-6**

**DETAILED DESIGN DOCUMENT**

**6.1Introduction:**

The details we decide the internal logic for the module, which implement the given specification detailed design is one of the design process for software items. Details process for software items. Detailed design, which is also known as logic design, involves the internal sign of the module and how the specification of the module can be satisfied detailed design is an extension of system design.

The first step before the detailed design or code for a module can be developed is that specification of the module be given precisely. once the module is precisely specified. the external logic for the module that will implement the given identification can be decided.

The Basic logic in detailed design is to specify the logic for the different modules that been specified during system design. Specifying the logic will require developing an algorithm

That will be implementing the given specification.

This phase includes the specification of table structures, which makes us understand the internal logic of the system. the tables are described with their field names, data type, table constraints, description of each field etc.

**6.2 Structured English:**

**Register form:**

Begin

If (click on submit) then

Check for the null values and other invalid entries if all are valid then save it to the database

Else

Display message “invalid”

End if

Else

Generate unique user id and store data in database

End if

End

**Login form:**

Begin

If (click on login) then

Check the username and password, if the user exists then redirect into dashboard

Else if (user does not exist) then

Display message “Invalid Username” or “Invalid password”

End if

End

**Add admin:**

Begin

If (click on add record) then

Check for the null values and other invalid entries if all are valid then save it to the database

Else

Display message “invalid “

End if

End

**Add employee:**

Begin

If (click on add record) then

Check for the null values and other invalid entries if all are valid then save it to the database

Else

Display message “invalid “

End if

End

**Add branch:**

Begin

If (click on add record) then

Check for the null values and other invalid entries if all are valid then save it to the database

Else

Display message “invalid “

End if

End

**Add parking location:**

Begin

If (click on add record) then

Check for the null values and other invalid entries if all are valid then save it to the database

Else

Display message “invalid “

End if

End

**Add vehicle type:**

Begin

If (click on add record) then

Check for the null values and other invalid entries if all are valid then save it to the database

Else

Display message “invalid “

End if

End

**Add parking slots:**

Begin

If (click on add record) then

Check for the null values and other invalid entries if all are valid then save it to the database

Else

Display message “invalid “

End if

End

**Add parking record:**

Begin

If (select parking location) then

Show vehicle type

If (select vehicle type) then

Show the parking slots

If (select a slot which is not previously selected) then

Show a pop of window to enter vehicle number & customer name

If (click on submit) then

Check for the null values and other invalid entries if all are valid then save it to the database

Else

Display message “invalid “

End if

End

**Exist vehicle:**

Begin

If (clicked parked slot) then

Calculate number of hours’ total cost and display it to the user

If (clicked confirm exit) then

Print the bill and make the slot available for parking the next vehicle

**Add feedback:**

Begin

If (click on feedback) then

Check for the null values and other invalid entries if all are valid then save it to the database

Else

Display message “invalid “

End if

End

**CODING**

**Chapter-7**

**SYSTEM CODING**

**7.1 Introduction:**

The main goal of the coding or programming phase is to translate the design of the system produced during the design phase into code in a given programming language. It is then executed on the computer to verify whether the design is correct or not. In coding phase the output is code can be extremely useful in enhancing the understandability. Internal documentation of code is done using comments in the program. Comments are textual statements that are meant for the program reader and are not executed.

The coding phase effects both testing and maintenance phases. Well write code can reduce the testing and maintenance effort. The goal of coding should be to reduce the testing and maintenance effort.

**7.2 Module Coding:**

**Admin login module:**

<?php

include("header.php");

if(!isset($\_SESSION["adminid"]))

{

echo "<script>window.location='log.php';</script>";

}

if($\_SESSION[sessionid] == $\_POST[sessionid])

{

if(isset($\_POST['submit']))

{

if(isset($\_GET['editid']))

{

$sql = "UPDATE admin SET admin\_name='$\_POST[adminname]',login\_id='$\_POST[loginid]',status='$\_POST[status]' where admin\_id='$\_GET[editid]'";

$qsql = mysqli\_query($con,$sql);

if(!$qsql)

{

echo mysqli\_error($con);

}

if(mysqli\_affected\_rows($con) ==1)

{

echo "<script>alert('Admin record updated successfully..');</script>";

}

}

else

{

$pwd = md5($\_POST['password']);

$sql = "INSERT INTO admin(admin\_name,login\_id,password,status) VALUES('$\_POST[adminname]','$\_POST[loginid]','$pwd','$\_POST[status]')";

$qsql = mysqli\_query($con,$sql);

if(!$qsql)

{

echo mysqli\_error($con);

}

if(mysqli\_affected\_rows($con) ==1)

{

echo "<script>alert('Admin record inserted successfully..');</script>";

}

}

}

}

$\_SESSION[sessionid] = rand();

if(isset($\_GET['editid']))

{

$sqledit = "SELECT \* FROM admin WHERE admin\_id='$\_GET[editid]'";

$qsqledit = mysqli\_query($con,$sqledit);

$rsedit = mysqli\_fetch\_array($qsqledit);

}

?>

<section id="divider">

<div class="container">

<div class="row">

<div class="col-md-offset-2 col-md-8 col-sm-12">

<h2 class="wow fadeInUp" data-wow-delay="0.4s">Admin</h2>

<form action="" method="post" name="frmadmin" onSubmit="return validateform()">

<input type="hidden" name="sessionid" value="<?php echo $\_SESSION[sessionid]; ?>" >

<div class="col-md-offset-2 col-md-8 col-sm-12 wow fadeInUp" data-wow-delay="0.5s" style="text-align:left;">

<h4>Admin name</h4>

<input name="adminname" value="<?php if(isset($\_GET['editid'])){ echo $rsedit['admin\_name'];} ?>" type="text" class="form-control" id="adminname" placeholder="Enter your name" ><span id="erradminname" style="color:red;font-Weight:bold;" ></span>

</div>

<div class="col-md-offset-2 col-md-8 col-sm-12 wow fadeInUp" data-wow-delay="0.5s" style="text-align:left;">

<h4>Log in ID</h4>

<input name="loginid" type="text" value="<?php if(isset($\_GET['editid'])){ echo $rsedit['login\_id'];} ?>" class="form-control" id="loginid" placeholder="Enter the login ID" ><span id="errloginid" style="color:red;font-Weight:bold;" ></span>

</div>

<div class="col-md-offset-2 col-md-8 col-sm-12 wow fadeInUp" data-wow-delay="0.5s" style="text-align:left;">

<h4>Password</h4>

<input name="password" type="password" class="form-control" id="password" placeholder="Enter the password" ><span id="errpassword" style="color:red;font-Weight:bold;" ></span>

</div>

<div class="col-md-offset-2 col-md-8 col-sm-12 wow fadeInUp" data-wow-delay="0.5s" style="text-align:left;">

<h4>Confirm password</h4>

<input name="cpassword" type="password" class="form-control" id="cpassword" placeholder="confirm your password" ><span id="errcpassword" style="color:red;font-Weight:bold;" ></span>

</div>

<div class="col-md-offset-2 col-md-8 col-sm-12 wow fadeInUp" data-wow-delay="0.5s" style="text-align:left;">

<h4>Status</h4>

<select name="status" class="form-control">

<option value="">Select status</option>

<?php

$arr = array("Active","Inactive");

foreach($arr as $val)

{

echo "<option value='$val'>$val</option>";

}if(isset($\_GET['editid']))

{

echo "<option value=$rsedit[status] selected>$rsedit[status]</option>";

}

?>

</select><span id="errstatus" style="color:red;font-Weight:bold;" ></span>

</div>

<div class="col-md-offset-2 col-md-8 col-sm-12 wow fadeInUp" data-wow-delay="0.5s"><br>

<input name="submit" type="submit" class="section-btn btn btn-success smoothScroll" id="submit"

<?php

if(isset($\_GET['editid']))

{

echo ' value="Click Here to update" ';

}

else

{

echo ' value="Click Here to Add" ';

}

?>>

</div>

</form>

</div>

</div>

</div>

</section>

<?php

include("footer.php");

?>

<script type="application/javascript">

function validateform()

{

var i=0;

var alphaExp = /^[a-zA-Z]+$/; //Variable to validate only alphabets

var alphaspaceExp = /^[a-zA-Z\s]+$/; //Variable to validate only alphabets with space

var alphanumericExp = /^[a-zA-Z0-9]+$/i; //Variable to validate only alphanumerics

var numericExpression = /^[0-9]+$/; //Variable to validate only numbers

var mobileno = /^\d{10}$/;

var mailformat = /^\w+([\.-]?w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/;

document.getElementById("erradminname").innerHTML = "";

document.getElementById("errloginid").innerHTML = "";

document.getElementById("errpassword").innerHTML = "";

document.getElementById("errcpassword").innerHTML = "";

document.getElementById("errstatus").innerHTML = "";

if(!document.frmadmin.adminname.value.match(alphaspaceExp))

{

document.getElementById("erradminname").innerHTML = "Admin name should not contain number and special characters..";

i=1;

}

if(document.frmadmin.adminname.value == "")

{

document.getElementById("erradminname").innerHTML = "Admin name should not be empty..";

i=1;

}

if(!document.frmadmin.loginid.value.match(alphanumericExp))

{

document.getElementById("errloginid").innerHTML = "Login ID should not contain special characters..";

i=1;

}

if(document.frmadmin.loginid.value == "")

{

document.getElementById("errloginid").innerHTML = "Login ID should not be empty..";

i=1;

}

if(document.frmadmin.password.value.length < 6)

{

document.getElementById("errpassword").innerHTML = "Password should contain atleast 6 characters";

i=1;

}

if(document.frmadmin.password.value == "")

{

document.getElementById("errpassword").innerHTML = "Password should not be empty..";

i=1;

}

if(document.frmadmin.password.value != document.frmadmin.cpassword.value)

{

document.getElementById("errcpassword").innerHTML = "Password and Confirm Password not matching..";

i=1;

}

if(document.frmadmin.cpassword.value == "")

{

document.getElementById("errcpassword").innerHTML = "Confirm Password should not be empty..";

i=1;

}

if(document.frmadmin.status.value == "")

{

document.getElementById("errstatus").innerHTML = "Status should not be empty..";

i=1;

}

if(i==0)

{

return true;

}

else

{

return false;

}

}

</script>

**Header module:**

<!DOCTYPE html>

<?php

session\_start();

date\_default\_timezone\_set('Asia/Kolkata');

include("dbconnection.php");

if(isset($\_POST['adminsubmit']))

{

$pwd = md5($\_POST[password]);

$sql = "SELECT \* FROM admin WHERE login\_id ='$\_POST[loginid]' AND password='$pwd' AND status='Active'";

$qsql = mysqli\_query($con,$sql);

$rs = mysqli\_fetch\_array($qsql);

if(mysqli\_num\_rows($qsql) ==1)

{

$\_SESSION["adminid"] = $rs[admin\_id];

echo "<script>window.location='dashboard.php';</script>";

}

else

{

echo "<script>alert('Invalid login credentials entered.');</script>";

}

}

if(isset($\_POST['empsubmit']))

{

$pwd = md5($\_POST[epassword]);

$sql = "SELECT \* FROM employee WHERE login\_id ='$\_POST[eloginid]' AND password='$pwd' AND status='Active'";

$qsql = mysqli\_query($con,$sql);

$rs = mysqli\_fetch\_array($qsql);

if(mysqli\_num\_rows($qsql) ==1)

{

$\_SESSION["employeeid"] = $rs[employee\_id];

$\_SESSION["branch\_id"] = $rs[branch\_id];

echo "<script>window.location='employeeaccount.php';</script>";

}

else

{

echo "<script>alert('Invalid login credentials entered.');</script>";

}

}

if(isset($\_SESSION["employeeid"]))

{

$sqlusremployee = "SELECT \* FROM employee WHERE employee\_id ='$\_SESSION[employeeid]'";

$qsqlusremployee = mysqli\_query($con,$sqlusremployee);

$rsusremployee = mysqli\_fetch\_array($qsqlusremployee);

}

?>

<html lang="en">

<head>

<link rel="icon" type="image/ico" href="images/favicon.ico">

<meta charset="UTF-8">

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

<meta name="description" content="">

<meta name="keywords" content="">

<meta name="author" content="">

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">

<title>Pay&Park</title>

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

<link rel="stylesheet" href="css/animate.css">

<link rel="stylesheet" href="css/font-awesome.min.css">

<link rel="stylesheet" href="css/magnific-popup.css">

<link rel="stylesheet" href="css/owl.theme.css">

<link rel="stylesheet" href="css/owl.carousel.css">

<link href='https://fonts.googleapis.com/css?family=Unica+One' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,300,700' rel='stylesheet' type='text/css'>

<!-- Main css -->

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

<style>

#filein

{

font-size:1.3em;

color:#FFF;

background-color:#000;

border-radius:25px;

width:100px;

text-align:center;

border:1px solid #000;

}

#filein:hover

{

text-align:center;

width:100px;

color:#fff;

border:1px solid #FFF;

background-color:#0CF;

border-left-color:#000;

border-right-color:#000;

}

/\*Dropdown menu css code starts here \*/

#primary\_nav\_wrap

{

margin-top:15px

}

#primary\_nav\_wrap ul

{

list-style:none;

position:relative;

float:left;

margin:0;

padding:0

}

#primary\_nav\_wrap ul a

{

display:block;

}

#primary\_nav\_wrap ul li

{

position:relative;

float:left;

margin:0;

padding:0

}

#primary\_nav\_wrap ul li.current-menu-item

{

background:#ddd

}

#primary\_nav\_wrap ul li:hover

{

background:#f6f6f6

}

#primary\_nav\_wrap ul ul

{

display:none;

position:absolute;

top:100%;

left:0;

background:#fff;

padding:0

}

#primary\_nav\_wrap ul ul li

{

float:none;

width:200px

}

#primary\_nav\_wrap ul ul a

{

line-height:120%;

padding:10px 15px

}

#primary\_nav\_wrap ul ul ul

{

top:0;

left:100%

}

#primary\_nav\_wrap ul li:hover> ul

{

display:block

}

/\*Dropdown menu css code ends here \*/

.ov:hover

{

box-shadow: 2px 2px 2px 2px #000;

}

</style>

</head>

<body data-spy="scroll" data-target=".navbar-collapse" data-offset="50">

<div class="preloader">

<div class="sk-spinner sk-spinner-pulse"></div>

</div>

<!-- Navigation Section -->

<div class="navbar navbar-default navbar-fixed-top">

<div class="container">

<div class="navbar-header">

<button class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">

<span class="icon icon-bar"></span>

<span class="icon icon-bar"></span>

<span class="icon icon-bar"></span>

</button>

<a href="index.php" class="navbar-brand"><span>Pay</span>& Park</a>

</div>

<div class="collapse navbar-collapse">

<nav id="primary\_nav\_wrap">

<ul class="nav navbar-nav navbar-right">

<?php

if(isset($\_SESSION["adminid"]))

{

echo '<li><a href="dashboard.php" class="smoothScroll">Dashboard</a></li>' ;

?>

<li><a href="#" style="cursor:pointer;">User</a>

<ul>

<li><a href="admin.php" style="cursor:pointer;">Add Admin</a></li>

<li><a href="viewadmin.php" style="cursor:pointer;">View Admin</a></li>

<li><a href="employee.php" style="cursor:pointer;">Add employee</a></li>

<li><a href="viewemployee.php" style="cursor:pointer;">View employee</a></li>

</ul>

</li>

<li><a href="#" style="cursor:pointer;">settings</a>

<ul>

<li><a href="branch.php" style="cursor:pointer;">Add Branch</a></li>

<li><a href="viewbranch.php" style="cursor:pointer;">View Branch</a></li>

<li><a href="vehicletype.php" style="cursor:pointer;">Add Vehicle type</a></li>

<li><a href="viewvehicletype.php" style="cursor:pointer;">View Vehicle type</a></li>

</ul>

</li>

<li><a href="#" style="cursor:pointer;">parking settings</a>

<ul>

<li><a href="parkinglocation.php" style="cursor:pointer;">Add parking location</a></li>

<li><a href="viewparkinglocation.php" style="cursor:pointer;">View parkig location</a></li>

</ul>

</li>

<li><a href="#" style="cursor:pointer;">voucher</a>

<ul>

<li><a href="customer.php" style="cursor:pointer;">Add customer</a></li>

<li><a href="#" style="cursor:pointer;">View custemor</a></li>

<li><a href="Voucher.php" style="cursor:pointer;">Add voucher</a></li>

<li><a href="#" style="cursor:pointer;">View voucher</a></li>

<li><a href="Voucher\_transaction.php" style="cursor:pointer;">Deposit Money</a></li>

<li><a href="#" style="cursor:pointer;">View voucher transaction</a></li>

</ul>

</li>

<li><a href="#" style="cursor:pointer;">profile</a>

<ul>

<li><a href="adminprofile.php" style="cursor:pointer;">My profile</a></li>

<li><a href="adminchangepassword.php" style="cursor:pointer;">change password</a></li>

</ul>

</li>

<li><a href="#" style="cursor:pointer;">parking</a>

<ul>

<li><a href="#" style="cursor:pointer;">view parking record</a></li>

</ul>

</li>

<li><a href="#" style="cursor:pointer;">report</a>

<ul>

<li><a href="#" style="cursor:pointer;">Branch wise report</a></li>

<li><a href="#" style="cursor:pointer;">vehicle parking report</a></li>

<li><a href="#" style="cursor:pointer;">parking income report</a></li>

<li><a href="#" style="cursor:pointer;">employee wise report</a></li>

</ul>

</li>

<?php

echo '<li><a href="logout.php" class="smoothScroll">Logout</a></li>' ;

}

else if(isset($\_SESSION["employeeid"]))

{

echo '<li><a href="employeeaccount.php" style="cursor:pointer;" class="smoothScroll">Account</a></li>' ;

?>

<li><a href="#" style="cursor:pointer;">voucher</a>

<ul>

<li><a href="customer.php" style="cursor:pointer;">Add customer</a></li>

<li><a href="viewcustomer.php" style="cursor:pointer;">View custemor</a></li>

<li><a href="Voucher.php" style="cursor:pointer;">Add voucher</a></li>

<li><a href="#" style="cursor:pointer;">View voucher</a></li>

<li><a href="Voucher\_transaction.php" style="cursor:pointer;">Deposit Money</a></li>

<li><a href="#" style="cursor:pointer;">View voucher transaction</a></li>

</ul>

</li>

<li><a href="#" style="cursor:pointer;">profile</a>

<ul>

<li><a href="employeeprofile.php" style="cursor:pointer;">My profile</a></li>

<li><a href="employeechangepassword.php" style="cursor:pointer;">change password</a></li>

</ul>

</li>

<li><a href="#" style="cursor:pointer;">Parking</a>

<ul>

<li><a href="parking\_record.php" style="cursor:pointer;">add parking record</a></li>

<li><a href="#" style="cursor:pointer;">view parking record</a></li>

</ul>

</li>

<li><a href="#" style="cursor:pointer;">Report</a>

<ul>

<li><a href="#" style="cursor:pointer;">Branch wise report</a></li>

<li><a href="#" style="cursor:pointer;">vehivle parking report</a></li>

<li><a href="#" style="cursor:pointer;">parking income report</a></li>

<li><a href="#" style="cursor:pointer;">employee wise report</a></li>

</ul>

</li>

<?php

echo '<li><a href="logout.php" class="smoothScroll">Logout</a></li>' ;

}

else

{

if(basename($\_SERVER['PHP\_SELF']) == "index.php")

{

echo '<li><a href="#home" class="smoothScroll">Home</a></li>';

}

else

{

echo '<li><a href="index.php#home" class="smoothScroll">Home</a></li>';

}

echo '<li><a href="#screenshot" class="smoothScroll">About Us</a></li>' ;

echo '<li><a class="smoothScroll" data-toggle="modal" data-target="#modal2" style="cursor:pointer;">Admin Login</a></li>' ;

echo '<li><a class="smoothScroll" data-toggle="modal" data-target="#modal3" style="cursor:pointer;">Employee Login</a></li>' ;

echo '<li><a class="smoothScroll" data-toggle="modal" data-target="#modalcust" style="cursor:pointer;">Customer Login</a></li>' ;

echo '<li><a class="smoothScroll" data-toggle="modal" data-target="#modal1" style="cursor:pointer;">Contact Us</a></li>' ;

}

?>

</ul>

</nav>

</div>

</div>

</div>

**Footer module:**

<!-- Footer Section -->

<style type="text/css">

#modal3 .modal-dialog .modal-content.modal-popup form a {

color: #FFF;

}

</style>

<footer>

<div class="container">

<div class="row">

<div class="col-md-8 col-sm-6">

<div class="wow fadeInUp footer-copyright" data-wow-delay="0.4s">

<p>Copyright &copy; 2017 Pay & Park

<span>||</span>

Design by: Pearel Nazareth

<!--<span>||</span><a href="https://plus.google.com/+templatemo" title="free css templates" target="\_blank">Admin Login</a> -->

</p>

</div>

</div>

<div class="col-md-4 col-sm-6">

<ul class="wow fadeInUp social-icon" data-wow-delay="0.8s">

<li><a href="#" class="fa fa-facebook"></a></li>

<li><a href="#" class="fa fa-twitter"></a></li>

<li><a href="#" class="fa fa-google-plus"></a></li>

<li><a href="#" class="fa fa-dribbble"></a></li>

<li><a href="#" class="fa fa-linkedin"></a></li>

</ul>

</div>

</div>

</div>

</footer>

<!-- Modal Contact -->

<div class="modal fade" id="modal1" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content modal-popup">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<h2 class="modal-title">Contact Form</h2>

</div>

<form action="#" method="post">

<input name="name" type="text" class="form-control" id="name" placeholder="Your Name" required>

<input name="email" type="email" class="form-control" id="email" placeholder="Email Address" required>

<textarea name="message" rows="3" class="form-control" id="message" placeholder="Message" required></textarea>

<input name="submit" type="submit" class="form-control" id="submit" value="Send Message">

</form>

</div>

</div>

</div>

<!-- Modal Admin log in-->

<div class="modal fade" id="modal2" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content modal-popup">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<h2 class="modal-title">Admin Login Panel</h2>

</div>

<form action="" method="post">

<input name="loginid" type="text" class="form-control" id="loginid" placeholder="Login ID" required>

<input name="password" type="password" class="form-control" id="password" placeholder="Password" required>

<input name="adminsubmit" type="submit" class="form-control" id="adminsubmit" value="Click here to Login">

</form>

</div>

</div>

</div>

<!-- Modal Contact -->

<div class="modal fade" id="modal3" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content modal-popup">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<h2 class="modal-title">Employee Login Panel</h2>

</div>

<form action="" method="post">

<input name="eloginid" type="text" class="form-control" id="eloginid" placeholder="Login ID" required>

<input name="epassword" type="password" class="form-control" id="epassword" placeholder="Password" required>

<input name="empsubmit" type="submit" class="form-control" id="submit" value="Click here to Login">

</form>

</div>

</div>

</div>

<!-- Modal customer login -->

<div class="modal fade" id="modalcust" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content modal-popup">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<h2 class="modal-title">Customer Login Panel</h2>

</div>

<form action="" method="post">

<input name="eloginid" type="text" class="form-control" id="eloginid" placeholder="Login ID" required>

<input name="epassword" type="password" class="form-control" id="epassword" placeholder="Password" required>

<input name="custsubmit" type="submit" class="form-control" id="submit" value="Click here to Login">

</form>

</div>

</div>

</div>

<!-- Back top -->

<a href="#" class="go-top"><i class="fa fa-angle-up"></i></a>

<!-- SCRIPTS -->

<script src="js/jquery.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/jquery.magnific-popup.min.js"></script>

<script src="js/magnific-popup-options.js"></script>

<script src="js/owl.carousel.min.js"></script>

<script src="js/smoothscroll.js"></script>

<script src="js/wow.min.js"></script>

<script src="js/custom.js"></script>

</body>

</html>

**View admin module:**

<?php

include("header.php");

if(!isset($\_SESSION["adminid"]))

{

echo "<script>window.location='log.php';</script>";

}

if(isset($\_GET['delid']))

{

$sql = "DELETE FROM admin where admin\_id='$\_GET[delid]'";

$qsql = mysqli\_query($con,$sql);

if(mysqli\_affected\_rows($con) ==1)

{

echo "<script>alert('Admin record deleted successfully..');</script>";

}

}

?>

<section id="divider" >

<div class="container">

<h2 class="wow fadeInUp" data-wow-delay="0.4s">View Admin records</h2>

<table id="example" class="table table-striped table-bordered" cellspacing="0" width="100%">

<thead>

<tr>

<th scope="col">&nbsp;Admin Name</th>

<th scope="col">&nbsp;Login ID</th>

<th scope="col">&nbsp;Status</th>

<th scope="col">&nbsp;Action</th>

</tr>

</thead>

<tbody>

<?php

$sql= "SELECT \* FROM admin";

$qsql= mysqli\_query($con,$sql);

while($rs = mysqli\_fetch\_array($qsql))

{

echo "<tr>

<td>&nbsp;$rs[admin\_name]</td>

<td>&nbsp;$rs[login\_id]</td>

<td>&nbsp$rs[status]</td>

<td>&nbsp;<a href='admin.php?editid=$rs[admin\_id]'>Edit</a> | <a href='viewadmin.php?delid=$rs[admin\_id]' onclick='return deleterecord()' >Delete</a></td>

</tr>";

}

?>

</tbody>

</table>

</div>

</section>

<?php

include("footer.php");

include("datatables.php");

?>

<script type="application/javascript">

function deleterecord()

{

if(confirm("Are you sure want to delete this record???") == true)

{

return true;

}

else

{

return false;

}

}

</script>

**Parking record module:**

<?php

include("header.php");

if(!isset($\_SESSION["employeeid"]))

{

echo "<script>window.location='log.php';</script>";

}

?>

<section id="divider">

<div class="container">

<div class="row">

<div class="col-md-offset-2 col-md-8 col-sm-12">

<h2 class="wow fadeInUp" data-wow-delay="0.4s">Branch</h2>

<form action="" method="post" >

<div class="col-md-offset-2 col-md-8 col-sm-12 wow fadeInUp" data-wow-delay="0.5s" style="text-align:left;">

<h4>Location</h4>

<?php

$sqlloc = "SELECT \* FROM parking\_location INNER JOIN branch ON parking\_location.branch\_id=branch.branch\_id WHERE parking\_location.status='Active'";

if(isset($\_SESSION["employeeid"]))

{

$sqlloc = $sqlloc . " AND branch.branch\_id= '$\_SESSION[branch\_id]'";

}

?>

<select name="location" id="location" class="form-control" onchange="loadlocation(this.value)">

<option value="">Select location</option>

<?php

$qsqlloc = mysqli\_query($con,$sqlloc);

while($rsloc = mysqli\_fetch\_array($qsqlloc))

{

echo "<option value='$rsloc[parking\_location\_id]'>$rsloc[parking\_location]</option>";

}

?>

</select>

</div>

<br><br><br><br><br><br><br>

<div id="divparkinglocation"></div>

</form>

</div>

</div>

</div>

</section>

<?php

include("footer.php");

?>

<script type="application/javascript">

function loadlocation(locationid)

{

if (locationid == "")

{

document.getElementById("divparkinglocation").innerHTML = "";

return;

}

else

{

if (window.XMLHttpRequest)

{

// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = new XMLHttpRequest();

}

else

{

// code for IE6, IE5

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.open("GET","ajaxlocation.php?locationid="+locationid,true);

xmlhttp.send();

xmlhttp.onreadystatechange = function()

{

if (this.readyState == 4 && this.status == 200)

{

document.getElementById("divparkinglocation").innerHTML = this.responseText;

}

};

}

}

function loadslots(slotid)

{

if (slotid == "")

{

document.getElementById("divslotrec").innerHTML = "";

return;

}

else

{

if (window.XMLHttpRequest)

{

// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = new XMLHttpRequest();

}

else

{

// code for IE6, IE5

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.open("GET","ajaxslots.php?slotid="+slotid,true);

xmlhttp.send();

xmlhttp.onreadystatechange = function()

{

if (this.readyState == 4 && this.status == 200)

{

document.getElementById("divslotrec").innerHTML = this.responseText;

}

};

}

}

function bookslot(loc,vehtyp,parking\_slot\_id,slid)

{

document.getElementById("frmvehadd").style.visibility='visible';

//window.location='parking\_record\_entry.php?loc='+loc+'&vehtyp='+vehtyp+'&parking\_slot\_id='+ parking\_slot\_id +'&slid='+slid;

document.getElementById("loc").value = loc;

document.getElementById("vehtyp").value = vehtyp;

document.getElementById("parking\_slot\_id").value = parking\_slot\_id;

document.getElementById("slid").value = slid;

document.getElementById("vehno").value = "";

document.getElementById("custname").value = "";

document.getElementById("divvehadd").innerHTML = '';

document.getElementById("lblheading").innerHTML ='Add Vehicle record for slot';

}

function funaddveh(loc,vehtyp,slotid,slid,vehno,custname,voucherid)

{

alert(voucherid);

var currentdttime = '';

document.getElementById("frmvehadd").visibility='hidden';

document.getElementById("divvehadd").innerHTML = '<img src="images/loading.gif" width="200" height="200" alt=""/>';

if (window.XMLHttpRequest) {

// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = new XMLHttpRequest();

} else {

// code for IE6, IE5

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("divslotrec").innerHTML = this.responseText;

currentdttime = document.getElementById("insdt").value;

/\*document.getElementById("divvehadd").innerHTML = '<img src="images/right.jpg" width="50" height="50" alt=""/>';\*/

document.getElementById("lblheading").innerHTML ='Print';

document.getElementById("divvehadd").innerHTML = '<div id="idexitprint"><table id="example" class="table table-striped table-bordered" cellspacing="0" width="100%" style="background-color: antiquewhite;"><thead><tr><th scope="col">&nbsp;Receipt number</th><td scope="col" align="left">' + document.getElementById("receiptno").value + '</td></tr><tr><th scope="col">&nbsp;Vehicle number</th><td scope="col" align="left">' + vehno + '</td></tr><tr><th scope="col">&nbsp;Customer name</th><td scope="col" align="left"> ' + custname + ' </td></tr><tr><th scope="col">&nbsp;Entry time</th><td scope="col" align="left">' + currentdttime + '</td></tr><tr><th scope="col">&nbsp;Slot number</th><td scope="col" align="left"><font size="12" ><strong>' + slid + '</strong></td></tr></thead></table></div><hr /><input type="button" name="btnprint" value="Print" onclick="printdiv(`idexitprint`)" >';

document.getElementById("frmvehadd").style.visibility='hidden';

//Code to close modal

/\*

$(function () {

$('#modal4').modal('toggle');

});

\*/

}

};

xmlhttp.open("GET","ajaxslots.php?loc="+loc+"&vehtyp="+vehtyp+"&slotid="+slotid+"&slid="+slid+"&vehno="+vehno+"&custname="+custname+"&currentdttime="+currentdttime+"&voucherid="+voucherid,true);

xmlhttp.send();

}

function exitslot(parking\_record\_id,loc,vehtyp,parking\_slot\_id)

{

document.getElementById("exitfrmvehadd").style.visibility='visible';

document.getElementById("exitloc").value = loc;

document.getElementById("exitvehtyp").value = vehtyp;

document.getElementById("exitparking\_slot\_id").value = parking\_slot\_id;

document.getElementById("exitlblheading").innerHTML ='Exit entry';

document.getElementById("exitdivvehadd").innerHTML = '';

//alert(parking\_record\_id);

if (window.XMLHttpRequest) {

// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = new XMLHttpRequest();

} else {

// code for IE6, IE5

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("exitdivparkingrecid").innerHTML = this.responseText;

}

};

xmlhttp.open("GET","ajaxexitrecord.php?parking\_record\_id="+parking\_record\_id+"&parking\_slot\_id=" + parking\_slot\_id + "&btnexitrecord=set",true);

xmlhttp.send();

}

function funexitveh(parking\_record\_id,exitloc,exitvehtyp,exitparking\_slot\_id,vehicle\_no,customer\_name,entrytime,exittime,hourdiff,totcost,slot\_no)

{

var currentdttime = '';

document.getElementById("exitfrmvehadd").visibility='hidden';

document.getElementById("exitdivvehadd").innerHTML = '<img src="images/loading.gif" width="200" height="200" alt=""/>';

if (window.XMLHttpRequest) {

// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = new XMLHttpRequest();

} else {

// code for IE6, IE5

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("divslotrec").innerHTML = this.responseText;

currentdttime = document.getElementById("insdt").value;

document.getElementById("exitlblheading").innerHTML ='Print';

document.getElementById("exitdivvehadd").innerHTML = '<div id="idprint"><table id="example" class="table table-striped table-bordered" cellspacing="0" width="100%" style="background-color: antiquewhite;"><thead><tr><th scope="col">&nbsp;Bill No.</th><td scope="col" align="left">' + document.getElementById("bill\_no").value + '</td></tr><tr><th scope="col">&nbsp;Slot No.</th><td scope="col" align="left">' + slot\_no + '</td></tr><tr><th scope="col">&nbsp;Vehicle Number.</th><td scope="col" align="left">' + vehicle\_no + '</td></tr><tr><th scope="col">&nbsp;Customer Name.</th><td scope="col" align="left">' + customer\_name + '</td></tr><tr><th scope="col">&nbsp;Entry time.</th><td scope="col" align="left">' + entrytime + '</td></tr><tr><th scope="col">&nbsp;Exit Time.</th><td scope="col" align="left">' + exittime + '</td></tr><tr><th scope="col">&nbsp;No. of hours</th><td scope="col" align="left">' + hourdiff + '</td></tr><tr><th scope="col">&nbsp;Total cost</th><td scope="col" align="left">' + totcost + '</td></tr></thead></table></div><hr /><input type="button" name="btnprint" value="Print" onclick="printdiv(`idprint`)" >';

document.getElementById("exitfrmvehadd").style.visibility='hidden';

//Code to close modal

/\*

$(function () {

$('#modal4').modal('toggle');

});

\*/

}

};

xmlhttp.open("GET","ajaxslots.php?parking\_record\_id="+parking\_record\_id+"loc="+exitloc+"&vehtyp="+exitvehtyp+"&slotid="+exitparking\_slot\_id,true);

xmlhttp.send();

}

function printdiv(elem)

{

var mywindow = window.open('', 'PRINT', 'height=400,width=600');

mywindow.document.write('<html><head><title>' + document.title + '</title>');

mywindow.document.write('</head><body >');

mywindow.document.write('<center><h1>' + document.title + '</h1></center>');

mywindow.document.write(document.getElementById(elem).innerHTML);

mywindow.document.write('</body></html>');

mywindow.document.close(); // necessary for IE >= 10

mywindow.focus(); // necessary for IE >= 10\*/

mywindow.print();

mywindow.close();

return true;

}

function loadvoucherrecord(voucherno)

{

if (window.XMLHttpRequest)

{

// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = new XMLHttpRequest();

}

else

{

// code for IE6, IE5

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange = function()

{

if (this.readyState == 4 && this.status == 200)

{

//document.getElementById("voucherid").value;

document.getElementById("divvoucher").innerHTML = this.responseText;

document.getElementById("custname").value = document.getElementById("vouchercustname").value;

document.getElementById("vehno").value = document.getElementById("vouchervehicle\_no").value;

}

};

xmlhttp.open("GET","ajaxloadvoucher.php?voucherno="+voucherno,true);

xmlhttp.send();

}

</script>

<div class="modal fade" id="modal4" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content modal-popup">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<h2 class="modal-title" id="lblheading">Add Vehicle record for slot </h2>

</div>

<div id="divvehadd"></div>

<div id="frmvehadd"><form action="" method="post" onSubmit="return false;">

<input name="loc" id="loc" type="hidden" class="form-control" required>

<input name="vehtyp" id="vehtyp" type="hidden" class="form-control" required>

<input name="parking\_slot\_id" id="parking\_slot\_id" type="hidden" class="form-control" required>

<input name="slid" id="slid" type="hidden" class="form-control" required>

<input name="voucherno" id="voucherno" type="text" onKeyUp="loadvoucherrecord(this.value)" autocomplete="off" class="form-control" placeholder="Voucher number">

<div id="divvoucher"><input type="hidden" name="voucherid" id="voucherid" value="0" ></div>

<input name="vehno" id="vehno" type="text" autocomplete="off" class="form-control" placeholder="Vehicle Number" required>

<input name="custname" id="custname" type="text" autocomplete="off" class="form-control" placeholder="Customer Name" autocomplete="off" required>

<input onClick="funaddveh(loc.value,vehtyp.value,parking\_slot\_id.value,slid.value,vehno.value,custname.value,voucherid.value)" name="btnaddparking" type="submit" class="form-control" id="submit" value="Submit" >

</form></div>

</div>

</div>

</div>

<div class="modal fade" id="modal5" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content modal-popup">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<h2 class="modal-title" id="exitlblheading">Exit Entry </h2>

</div>

<div id="exitdivvehadd"></div>

<div id="exitfrmvehadd">

<form action="" method="post" onSubmit="return false;">

<input name="exitloc" id="exitloc" type="hidden" class="form-control" required>

<input name="exitvehtyp" id="exitvehtyp" type="hidden" class="form-control" required>

<input name="exitparking\_slot\_id" id="exitparking\_slot\_id" type="hidden" class="form-control" required>

<div id="exitdivparkingrecid"></div>

</form>

</div>

</div>

</div>

</div>

**Ajax slots module:**

<?php

session\_start();

date\_default\_timezone\_set('Asia/Kolkata');

include("dbconnection.php");

$dt =date("Y-m-d H:i:s");

//Coding to insert Vehicle entry record

if(isset($\_GET[vehno]))

{

$sqlreceipt = "SELECT MAX(receipt\_no) FROM parking\_record";

$qsqlreceipt = mysqli\_query($con,$sqlreceipt);

$rsreceipt = mysqli\_fetch\_array($qsqlreceipt);

$receiptno= $rsreceipt[0];

if($receiptno == 0)

{

$receiptno= "101";

}

else

{

$receiptno = $receiptno + 1;

}

//loc='+loc+'&vehtyp='+vehtyp+'&parking\_slot\_id='+ parking\_slot\_id +'&slid='+slid;

$sql = "INSERT INTO parking\_record( receipt\_no, in\_date\_time, vehicle\_type\_id, parking\_location\_id, parking\_slot\_id, slot\_no, employee\_id, vehicle\_no, customer\_name, customer\_id) VALUES ('$receiptno','$dt','$\_GET[vehtyp]','$\_GET[loc]','$\_GET[slotid]','$\_GET[slid]','$\_SESSION[employeeid]','$\_GET[vehno]','$\_GET[custname]','')";

$qsql = mysqli\_query($con,$sql);

if(!$qsql)

{

echo mysqli\_error($con);

}

$insid = mysqli\_insert\_id($con);

if($\_GET[voucherid] != 0)

{

$sqlvt = "INSERT INTO voucher\_transaction( voucher\_id, transaction\_type, trans\_date, parking\_record\_id, status) VALUES ('$\_GET[voucherid]','Debit','$dt','$insid','Inactive')";

$qsqlvt = mysqli\_query($con,$sqlvt);

if(!$qsqlvt)

{

echo mysqli\_error($con);

}

}

if(mysqli\_affected\_rows($con) ==1)

{

echo "<script>alert('Record inserted successfully..');</script>";

}

}

//Coding to insert vehicle exit record

if(isset($\_GET[parking\_record\_id]))

{

$sqlbill\_no= "SELECT MAX(bill\_no) FROM parking\_record";

$qsqlbill\_no = mysqli\_query($con,$sqlbill\_no);

$rsbill\_no = mysqli\_fetch\_array($qsqlbill\_no);

$bill\_no= $rsbill\_no[0];

if($bill\_no == 0)

{

$bill\_no= "101";

}

else

{

$bill\_no= $bill\_no + 1;

}

$sqlparking\_record = "UPDATE parking\_record SET out\_date\_time='$dt',bill\_no='$bill\_no' where parking\_record\_id='$\_GET[parking\_record\_id]'";

$qsqlparking\_record = mysqli\_query($con,$sqlparking\_record);

}

?>

<input type="hidden" name="insdt" id="insdt" value="<?php echo $dt; ?>" >

<input type="hidden" name="receiptno" id="receiptno" value="<?php echo $receiptno; ?>" >

<input type="hidden" name="bill\_no" id="bill\_no" value="<?php echo $bill\_no; ?>" >

<?php

include("dbconnection.php");

$sql= "SELECT \* FROM parking\_slots WHERE parking\_slot\_id ='$\_GET[slotid]'";

$qsql= mysqli\_query($con,$sql);

$rs = mysqli\_fetch\_array($qsql);

$sqlvehicle\_type= "SELECT \* FROM vehicle\_type WHERE vehicle\_type\_id ='$rs[vehicle\_type\_id]'";

$qsqlvehicle\_type= mysqli\_query($con,$sqlvehicle\_type);

$rsvehicle\_type = mysqli\_fetch\_array($qsqlvehicle\_type);

?>

<table id="example" class="table table-striped table-bordered wow fadeInUp" cellspacing="0" width="100%" style="background-color:#FFC" >

<thead>

<?php

for($i=1;$i<=$rs[number\_of\_slots];)

{

echo "<tr>";

for($j=0;$j<5;$j++)

{

$sqlslotchk = "SELECT \* FROM parking\_record WHERE vehicle\_type\_id='$rs[vehicle\_type\_id]' AND parking\_location\_id='$rs[parking\_location\_id]' AND parking\_slot\_id='$rs[parking\_slot\_id]' AND slot\_no='$i' AND bill\_no='0'";

$qsqlslotchk = mysqli\_query($con,$sqlslotchk);

$rsslotchk = mysqli\_fetch\_array($qsqlslotchk);

if(mysqli\_num\_rows($qsqlslotchk)>=1)

{

?>

<th class='ov' scope='col' style='cursor:pointer;background-color:red;vertical-align: middle;' data-toggle="modal" data-target="#modal5" onClick="exitslot(`<?php echo $rsslotchk[0]; ?>`,`<?php echo $rs[parking\_location\_id]; ?>`,`<?php echo $rs[vehicle\_type\_id]; ?>`,`<?php echo $rs[parking\_slot\_id]; ?>`)" >

<?php

echo "<center>";

echo $i;

?>

<br><img src="vehicleicons/<?php echo $rsvehicle\_type[vehicle\_icon]; ?>" width="75px" height="75px" >

<?php

echo "</center></th>";

}

else

{

//onclick='bookslot(`$rs[parking\_location\_id]`,`$rs[vehicle\_type\_id]`,`$rs[parking\_slot\_id]`,$i)'

?>

<th class='ov' scope='col' style='width:125px;height:125px;cursor:pointer;' onClick="bookslot(`<?php echo $rs[parking\_location\_id]; ?>`,`<?php echo $rs[vehicle\_type\_id]; ?>`,`<?php echo $rs[parking\_slot\_id]; ?>`,`<?php echo $i; ?>`)" data-toggle="modal" data-target="#modal4" >&nbsp;

<?php echo $i; ?>

</th>

<?php

}

$i++;

}

echo "</tr>";

}

?>

</thead>

</table>

**Ajax location module:**

<?php

include("dbconnection.php");

?>

<table id="example" class="table table-striped table-bordered wow fadeInUp" cellspacing="0" width="100%">

<thead>

<tr>

<th scope="col">&nbsp;Image</th>

<th scope="col">&nbsp;Location</th>

<th scope="col">&nbsp;Address</th>

</tr>

</thead>

<tbody>

<?php

$sql= "SELECT \* FROM parking\_location WHERE parking\_location\_id='$\_GET[locationid]'";

$qsql= mysqli\_query($con,$sql);

while($rs = mysqli\_fetch\_array($qsql))

{

echo "<tr>

<td>&nbsp;<img src='locationimage/$rs[location\_img]' width='50' height='50'></td>

<td>&nbsp;$rs[parking\_location]</td>

<td>&nbsp$rs[location\_address]</td>

</tr>";

}

?>

</tbody>

</table>

<div class="col-md-offset-2 col-md-8 col-sm-12 wow fadeInUp" style="text-align:left;">

<h4>Vehicle type</h4>

<select name="location" class="form-control" onChange="loadslots(this.value)" >

<option value="">Select Vehicle type</option>

<?php

$sql= "SELECT \* FROM parking\_slots WHERE parking\_location\_id='$\_GET[locationid]' AND status='Active'";

$qsql= mysqli\_query($con,$sql);

while($rs = mysqli\_fetch\_array($qsql))

{

$sqlvtype = "SELECT \* FROM vehicle\_type WHERE status='Active' AND vehicle\_type\_id ='$rs[vehicle\_type\_id]'";

$qsqlvtype = mysqli\_query($con,$sqlvtype);

$rsvtype = mysqli\_fetch\_array($qsqlvtype);

echo "<option value='$rs[parking\_slot\_id]'>$rsvtype[vehicle\_type] ( $rs[number\_of\_slots] ) </option>";

}

?>

</select>

</div>

<br><br><br><br><br><br><br>

<div id="divslotrec"></div>

**Ajax entry record module:**

<?php

include("dbconnection.php");

$dt =date("Y-m-d h:i:s");

if(isset($\_GET[vehno]))

{

$sqlreceipt = "SELECT MAX(receipt\_no) FROM parking\_record";

$qsqlreceipt = mysqli\_query($con,$sqlreceipt);

$rsreceipt = mysqli\_fetch\_array($qsqlreceipt);

$receiptno= $rsreceipt[0];

if($receiptno == 0)

{

$receiptno= "101";

}

else

{

$receiptno = $receiptno + 1;

}

//loc='+loc+'&vehtyp='+vehtyp+'&parking\_slot\_id='+ parking\_slot\_id +'&slid='+slid;

$sql = "INSERT INTO parking\_record( receipt\_no, in\_date\_time, vehicle\_type\_id, parking\_location\_id, parking\_slot\_id, slot\_no, employee\_id, vehicle\_no, customer\_name, customer\_id) VALUES ('$receiptno','$dt','$\_GET[vehtyp]','$\_GET[loc]','$\_GET[slotid]','$\_GET[slid]','$\_SESSION[employeeid]','$\_GET[vehno]','$\_GET[custname]','')";

$qsql = mysqli\_query($con,$sql);

if(!$qsql)

{

echo mysqli\_error($con);

}

if(mysqli\_affected\_rows($con) ==1)

{

echo "<script>alert('Record inserted successfully..');</script>";

}

}

?>

<table id="example" class="table table-striped table-bordered" cellspacing="0" width="100%" style="background-color: antiquewhite;">

<thead>

<tr>

<th scope="col">&nbsp;Vehicle number</th>

<td scope="col" align="left"><?php echo $rs[vehicle\_no]; ?></td>

</tr>

<tr>

<th scope="col">&nbsp;Customer name</th>

<td scope="col" align="left"><?php echo $rs[customer\_name]; ?></td>

</tr>

<tr>

<th scope="col">&nbsp;Entry time</th>

<td scope="col" align="left"><?php echo $rs[in\_date\_time]; ?></td>

</tr>

<tr>

<th scope="col">&nbsp;Slot number</th>

<td scope="col" align="left"><?php echo $rs[slot\_no]; ?></td>

</tr>

</thead>

</table>

<input onClick="funexitveh(<?php echo $\_GET[parking\_record\_id]; ?>)" name="btnaddparking" type="submit" class="form-control" id="submit" value="Exit" >

**Ajax exit record module:**

<?php

session\_start();

date\_default\_timezone\_set('Asia/Kolkata');

include("dbconnection.php");

$dt =date("Y-m-d H:i:s");

if(isset($\_GET["btnexitrecord"]))

{

$sql = "SELECT \* FROM parking\_record WHERE parking\_record\_id='$\_GET[parking\_record\_id]'";

$qsql= mysqli\_query($con,$sql);

$rs = mysqli\_fetch\_array($qsql);

//echo mysqli\_num\_rows($qsql);

//echo $rs[0];

}

?>

<table id="example" class="table table-striped table-bordered" cellspacing="0" width="100%" style="background-color: antiquewhite;">

<thead>

<tr>

<th scope="col">&nbsp;Vehicle number</th>

<td scope="col" align="left"><?php echo $rs[vehicle\_no]; ?></td>

</tr>

<tr>

<th scope="col">&nbsp;Customer name</th>

<td scope="col" align="left"><?php echo $rs[customer\_name]; ?></td>

</tr>

<tr>

<th scope="col">&nbsp;Entry time</th>

<td scope="col" align="left"><?php $time2 = $rs[in\_date\_time];

echo $entrytime = date("d - M - Y h:i A", strtotime($time2))

?></td>

</tr>

<tr>

<th scope="col">&nbsp;Exit time</th>

<td scope="col" align="left"><?php echo $exittime = date("d - M - Y H:i A", strtotime($dt)); ?></td>

</tr>

<tr>

<th scope="col">&nbsp;Number of hours</th>

<td scope="col" align="left"><?php $hourdiff = round((strtotime($dt) - strtotime($time2))/3600, 1);

if($hourdiff <1)

{

echo $hourdiff=1;

}

else

{

echo $hourdiff;

}

?> Hrs</td>

</tr>

<tr>

<th scope="col">&nbsp;Total cost</th>

<td scope="col" align="left">₹

<?php

$totcost = 0;

if($hourdiff >= 24)

{

$remhr = ceil($hourdiff - 24);

$sqlhr = "SELECT sum(cost) FROM parking\_cost WHERE (no\_of\_hours>='1' AND no\_of\_hours<='24') AND parking\_slot\_id='$\_GET[parking\_slot\_id]'";

$qsqlparking\_cost= mysqli\_query($con,$sqlhr);

$rsparking\_cost = mysqli\_fetch\_array($qsqlparking\_cost);

$a24hrcost = $rsparking\_cost[0];

$sqlparking\_cost = "SELECT \* FROM parking\_cost WHERE no\_of\_hours = '24' AND parking\_slot\_id='$\_GET[parking\_slot\_id]'";

$qsqlparking\_cost= mysqli\_query($con,$sqlparking\_cost);

$rsparking\_cost = mysqli\_fetch\_array($qsqlparking\_cost);

$cost24hrs = $rsparking\_cost[cost];

$totalextracost = $remhr \* $cost24hrs ;

$totcost = $a24hrcost + $totalextracost;

}

else

{

$tothr = ceil($hourdiff);

$sqlhr = "SELECT sum(cost) FROM parking\_cost WHERE (no\_of\_hours>='1' AND no\_of\_hours<='$tothr') AND parking\_slot\_id='$\_GET[parking\_slot\_id]'";

$qsqlparking\_cost= mysqli\_query($con,$sqlhr);

$rsparking\_cost = mysqli\_fetch\_array($qsqlparking\_cost);

$totcost= $rsparking\_cost[0];

}

echo $totcost;

?>

</td>

</tr>

<tr>

<th scope="col">&nbsp;Slot number</th>

<td scope="col" align="left"><?php echo $rs[slot\_no]; ?></td>

</tr>

</thead>

</table>

<input onClick="funexitveh('<?php echo $\_GET[parking\_record\_id]; ?>',exitloc.value,exitvehtyp.value,exitparking\_slot\_id.value,'<?php echo $rs[vehicle\_no]; ?>','<?php echo $rs[customer\_name]; ?>','<?php echo $entrytime; ?>','<?php echo $exittime; ?>','<?php echo $hourdiff; ?>','<?php echo $totcost; ?>','<?php echo $rs[slot\_no]; ?>')" name="btnaddparking" type="submit" class="form-control" id="submit" value="Confirm Exit" >

**TESTING**

**CHAPTER 8**

**TESTING**

**8.1 Introduction:**

Testing is the process of detecting errors. Testing performs a very special role for quality assurance and for ensuring the reliability of software. The results of testing are used later during maintenance also.

**8.2 Levels of testing:**

* Unit testing
* Integration testing
* Validation testing
* Output testing
* User acceptance testing

**8.2.1UNIT TESTING:**

Unit testing focuses verification effort on the smallest unit of software unit of software i.e. the module. Using detailed design and the process specification testing is done to uncover errors within the boundary of the module all the modules must be successful in the unit test before the start of integration testing begins. In this project unit testing is very much essential because the error is accurate. Unit testing is testing of the different units or modules of a system in isolation. testing is done to check whether each module in the software works properly so that it gives desired output in each module in the software works properly

**8.2.2Code walkthrough:**

In this phase of testing the code is thoroughly checked for discrepancies that may occur in the code such as redundancy of code, deviations in naming conventions etc.

**8.2.3Integration testing:**

In integration testing all the code modules are put together and tested for desired outputs. The modules unit tested is integration and tested. All the modules are combined in this resting step. Then the entire program is tested as whole. The integration testing is done

**8.2.4Output testing:**

Feeding sample valid input image and then comparing the ratio obtained in the compressed output image with expected ratio conduct it. Correctness of the output depends on the inputted image.

**8.2.5System testing:**

In system testing entire system is tested as whole with all forms, code modules and class modules. After the integration testing the whole of the system is tested in different environments and it is found that the system well without giving any runtime error. hence after the testing it is concluded that the system will work fine in all environments.

**8.3 Test cases:**

**Test case 1:**

**Objective:** test for login form

**Test data: Valid:** valid username and password to enter the program.

**Invalid:** when entered invalid username and password

**Outpu**t: **Valid:** logged in successfully

**Invalid:** show the error message

**Result: Valid:** the admin/employee can enter the program.

**Invalid:** the user is prompt with an error message and restricted to enter the system

**Conclusion:** both valid and invalid results are tested. output tally with the required result hence the test is successful.

**Test case 2:**

**Objective:** test for add admin, parking cost form

**Test data: Valid**: all the required fields are entered.

**Invalid:** some fields are blank or contain invalid data

**Output: Valid:** Allows to insert the record

**Invalid:** displays an error message to help the user understand the mistake

**Result: Valid:** record will be added to database.

**Invalid:** record will be not added to database.

**Conclusion:** both valid and invalid results are tested. output tally with the required result hence the test is successful.

**Test case 3:**

**Objective:** test for add customer

**Test data: Valid**: all the required fields are entered.

**Invalid:** some fields are blank or contain invalid data

**Output: Valid:** Allows to insert the record

**Invalid:** displays an error message to help the user understand the mistake

**Result: Valid:** record will be added to database.

**Invalid:** record will be not added to database.

**Conclusion:** both valid and invalid results are tested. output tally with the required result hence the test is successful.

**Test case 4:**

**Objective:** test for add branch

**Test data: Valid**: all the required fields are entered.

**Invalid:** some fields are blank or contain invalid data

**Output: Valid:** Allows to insert the record

**Invalid:** displays an error message to help the user understand the mistake

**Result: Valid:** record will be added to database.

**Invalid:** record will be not added to database.

**Conclusion:** both valid and invalid results are tested. output tally with the required result hence the test is successful.

**Test case 5:**

**Objective:** test for add employee

**Test data: Valid**: all the required fields are entered.

**Invalid:** some fields are blank or contain invalid data

**Output: Valid:** Allows to insert the record

**Invalid:** displays an error message to help the user understand the mistake

**Result: Valid:** record will be added to database.

**Invalid:** record will be not added to database.

**Conclusion:** both valid and invalid results are tested. output tally with the required result hence the test is successful.

**Test case 6:**

**Objective:** test for entering vehicle record form

**Test data: Valid:** empty slot is selected

**Invalid:** selected occupied slot

**Output: Valid:** Allows to insert the record

**Invalid:** displays an error message to help the user understand the mistake

**Result: Valid:** record will be added to database.

**Invalid:** record will be not added to database.

**Conclusion:** both valid and invalid results are tested. output tally with the required result hence the test is successful.

**Test case 7:**

**Objective:** test for add vehicle type

**Test data: Valid**: all the required fields are entered.

**Invalid:** some fields are blank or contain invalid data

**Output: Valid:** Allows to insert the record

**Invalid:** displays an error message to help the user understand the mistake

**Result: Valid:** record will be added to database.

**Invalid:** record will be not added to database.

**Conclusion:** both valid and invalid results are tested. output tally with the required result hence the test is successful.

**Test case 8:**

**Objective:** test for add parking location

**Test data: Valid**: all the required fields are entered.

**Invalid:** some fields are blank or contain invalid data

**Output: Valid:** Allows to insert the record

**Invalid:** displays an error message to help the user understand the mistake

**Result: Valid:** record will be added to database.

**Invalid:** record will be not added to database.

**Conclusion:** both valid and invalid results are tested. output tally with the required result hence the test is successful.

**Test case 9:**

**Objective:** test for add parking slots

**Test data: Valid**: all the required fields are entered.

**Invalid:** some fields are blank or contain invalid data

**Output: Valid:** Allows to insert the record

**Invalid:** displays an error message to help the user understand the mistake

**Result: Valid:** record will be added to database.

**Invalid:** record will be not added to database.

**Conclusion:** both valid and invalid results are tested. output tally with the required result hence the test is successful.

**Test case 10:**

**Objective:** test for add employee

**Test data: Valid**: all the required fields are entered.

**Invalid:** some fields are blank or contain invalid data

**Output: Valid:** Allows to insert the record

**Invalid:** displays an error message to help the user understand the mistake

**Result: Valid:** record will be added to database.

**Invalid:** record will be not added to database.

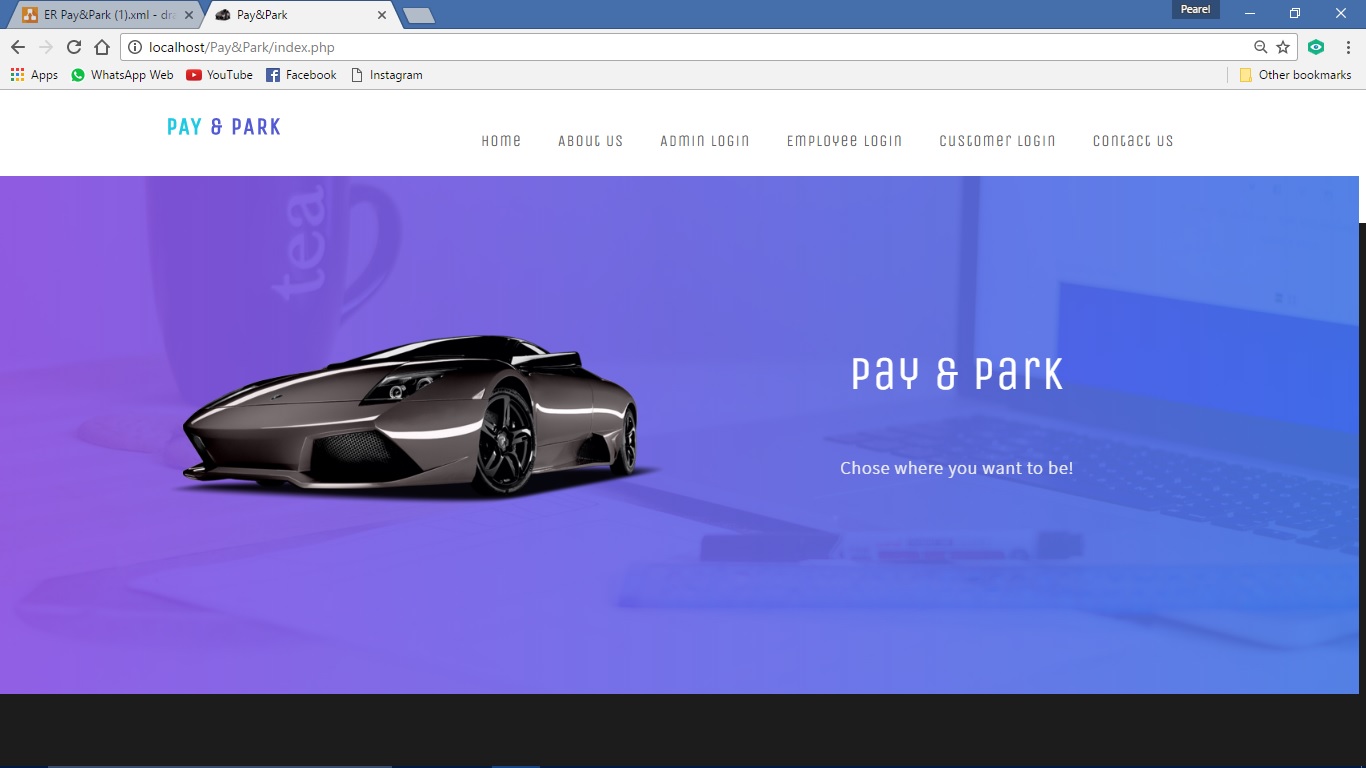
**Conclusion:** both valid and invalid results are tested. output tally with the required result hence the test is successful.

**SNAPSHOTS**

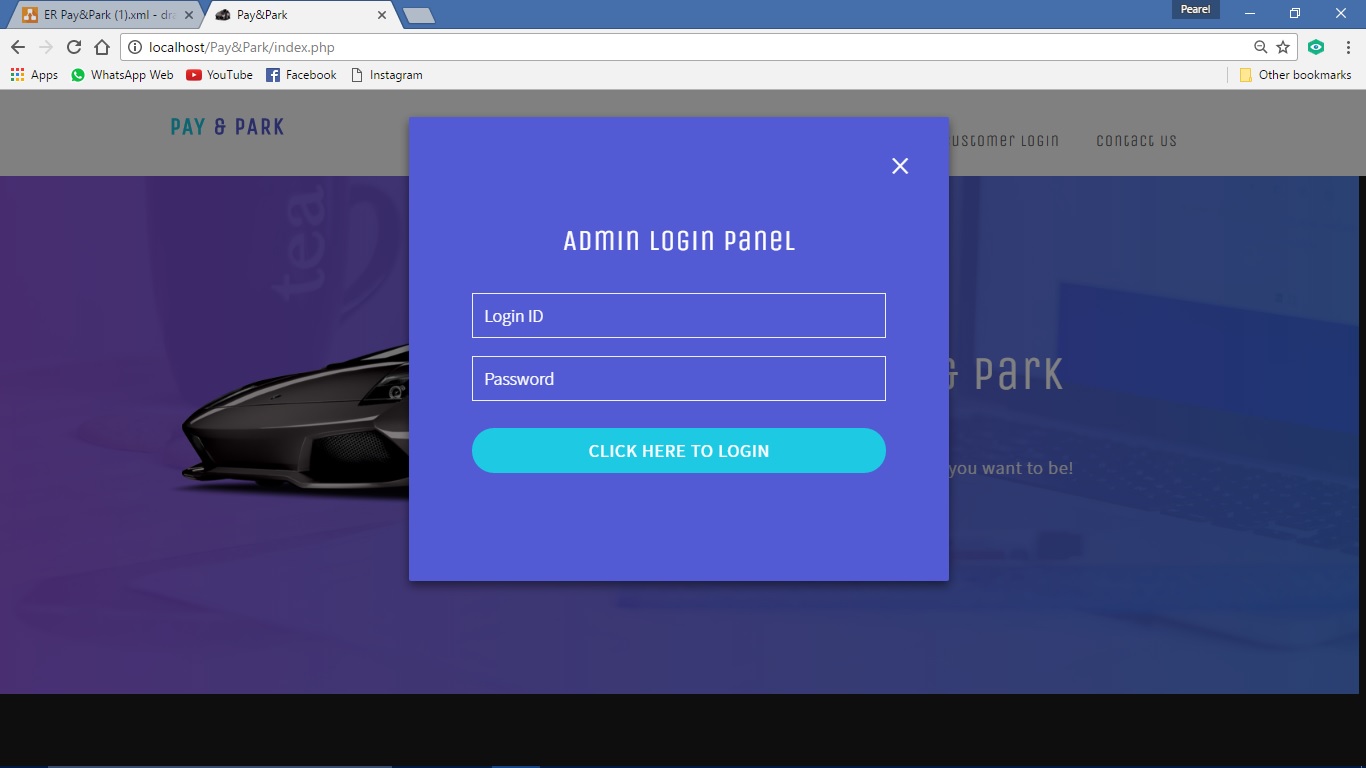
**Chapter-9**

**Snapshots**

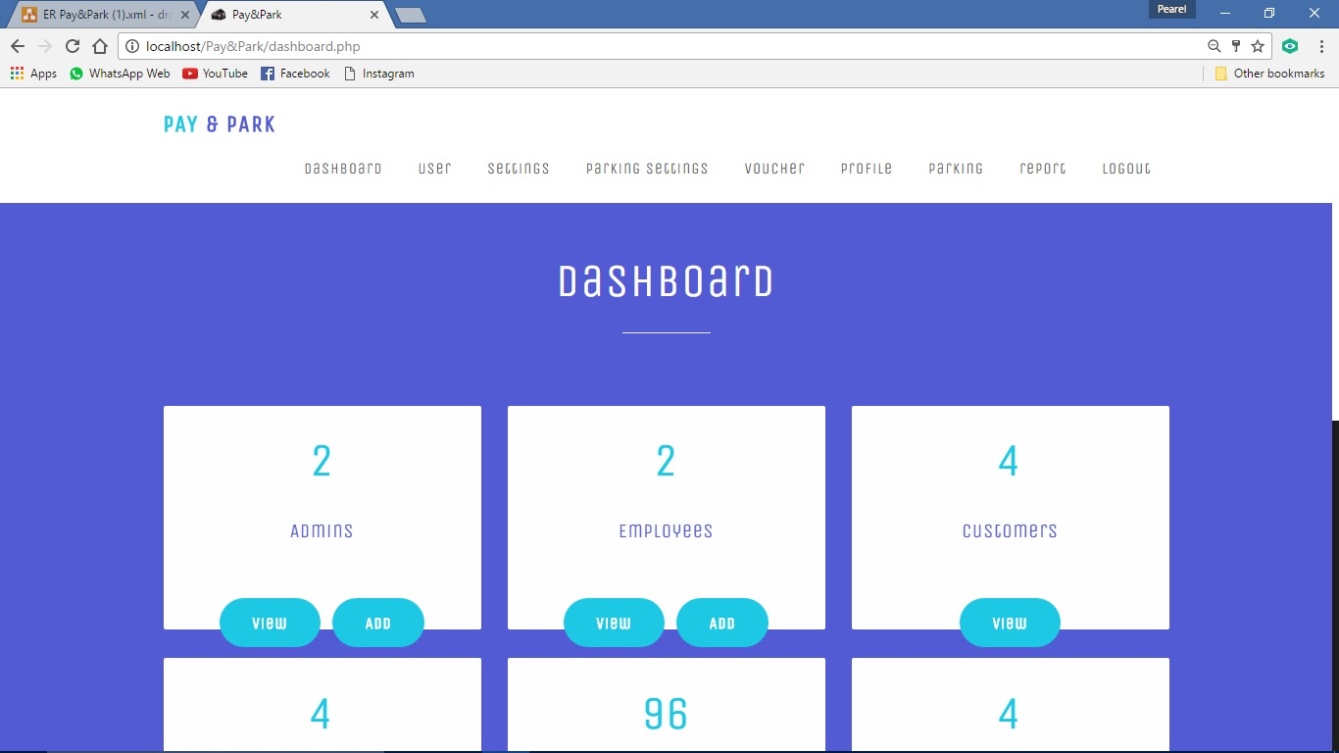
**Index Page:**

****

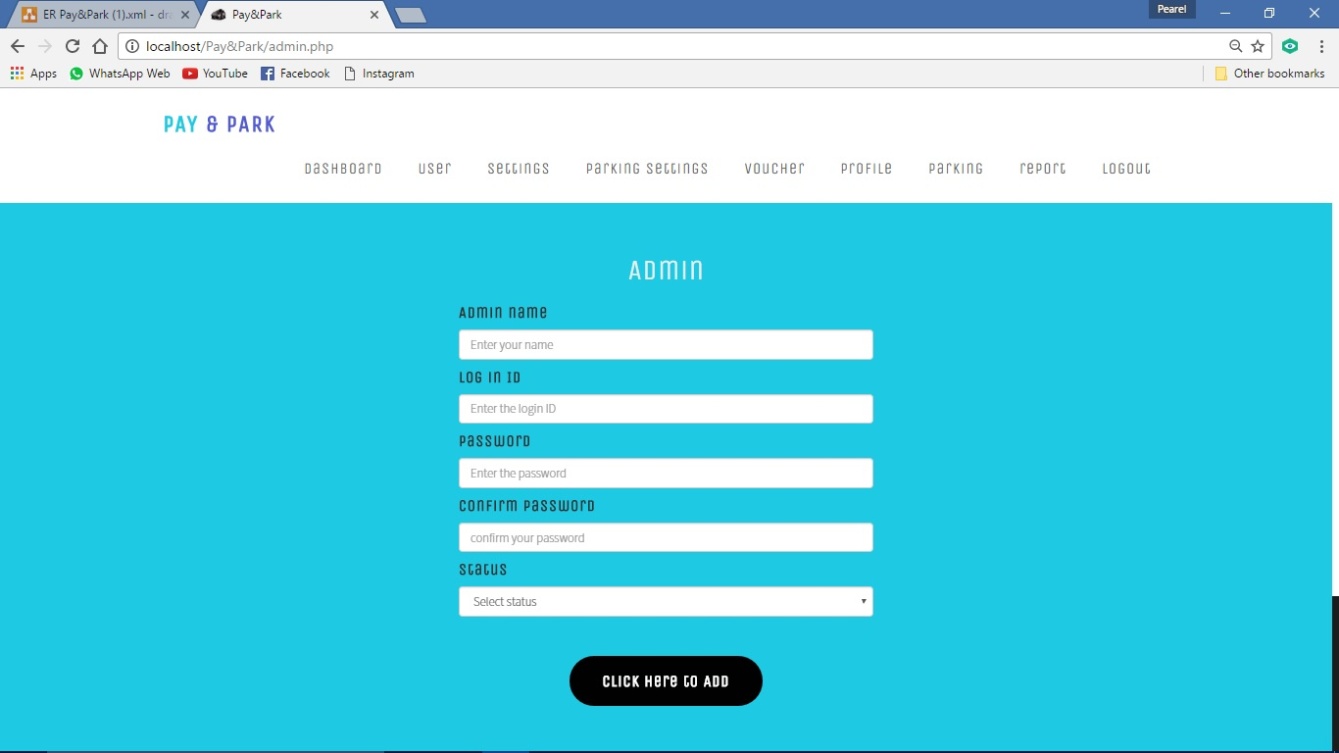
**Admin login:**

****

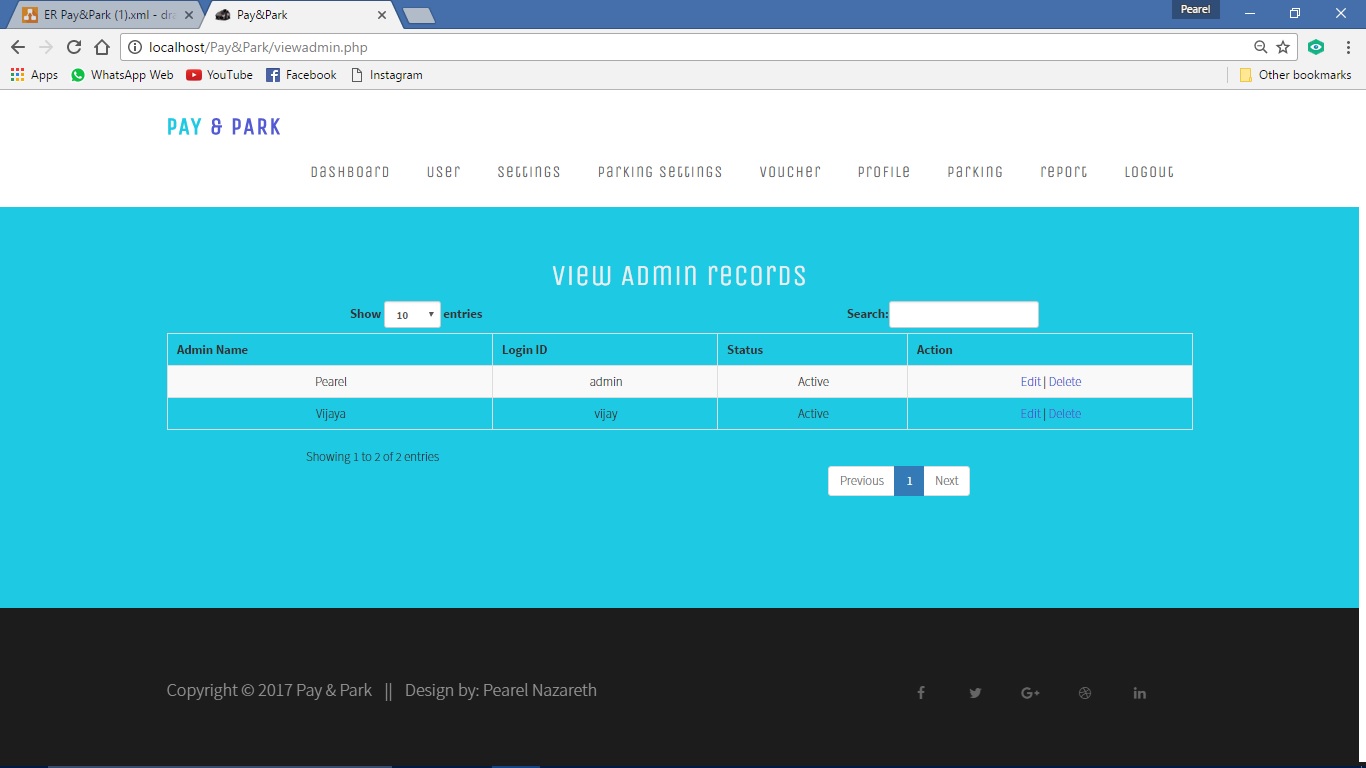
**Dash board:**

****

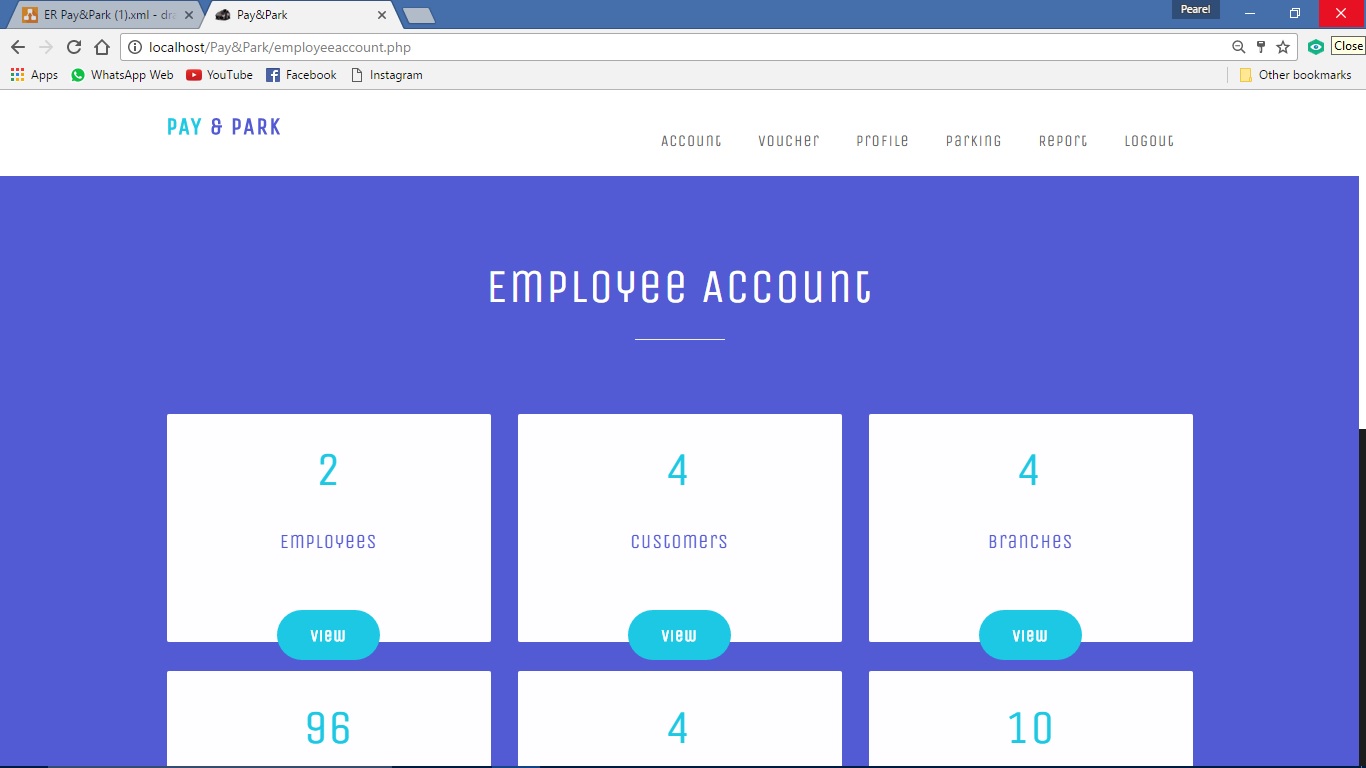
**Add admin:**

****

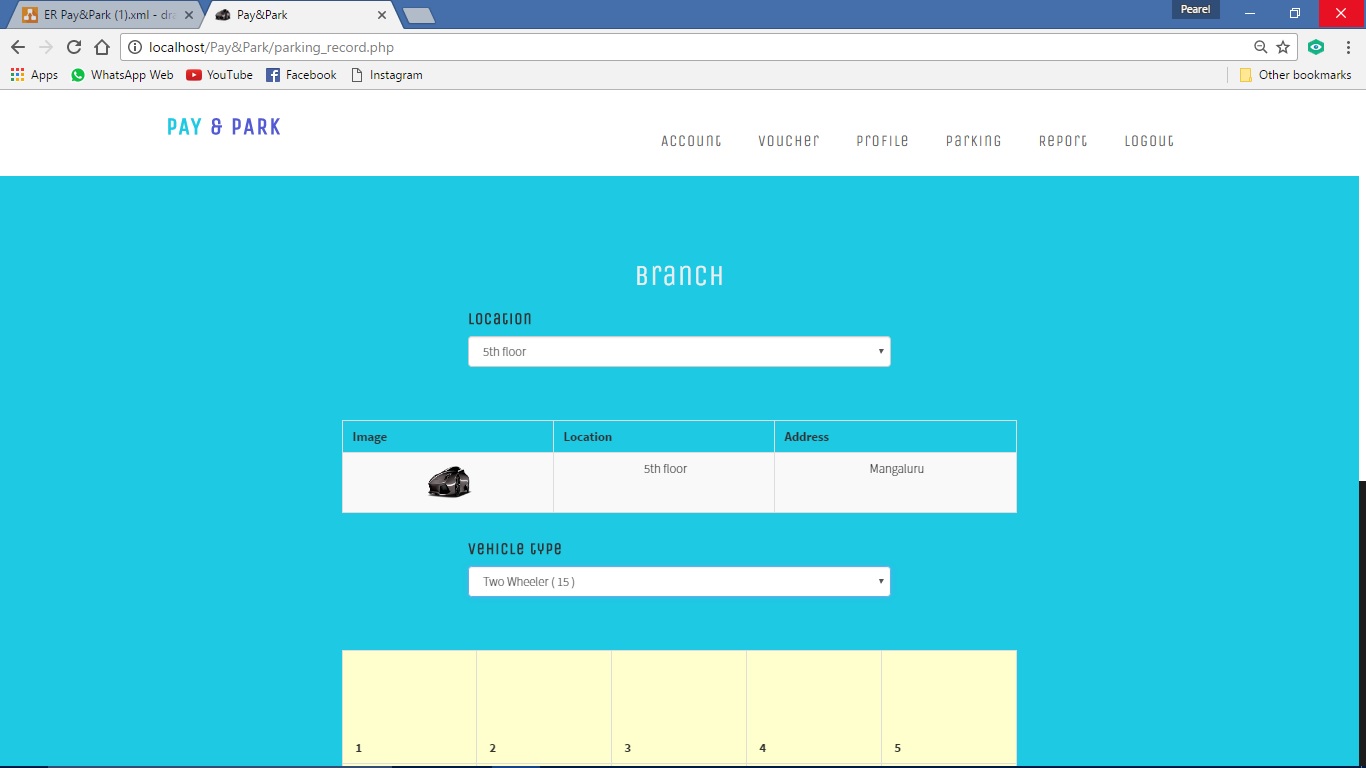
**View admin:**

****

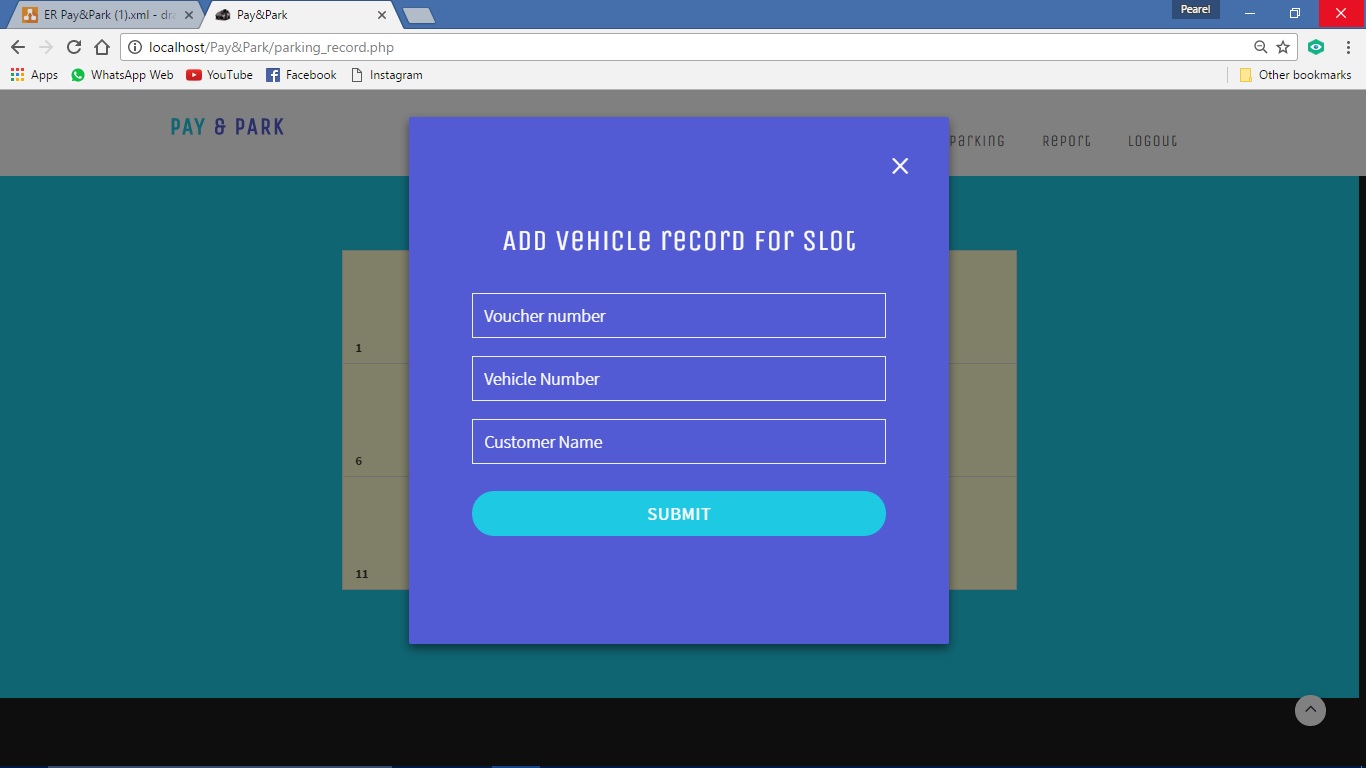
**Employee account:**

****

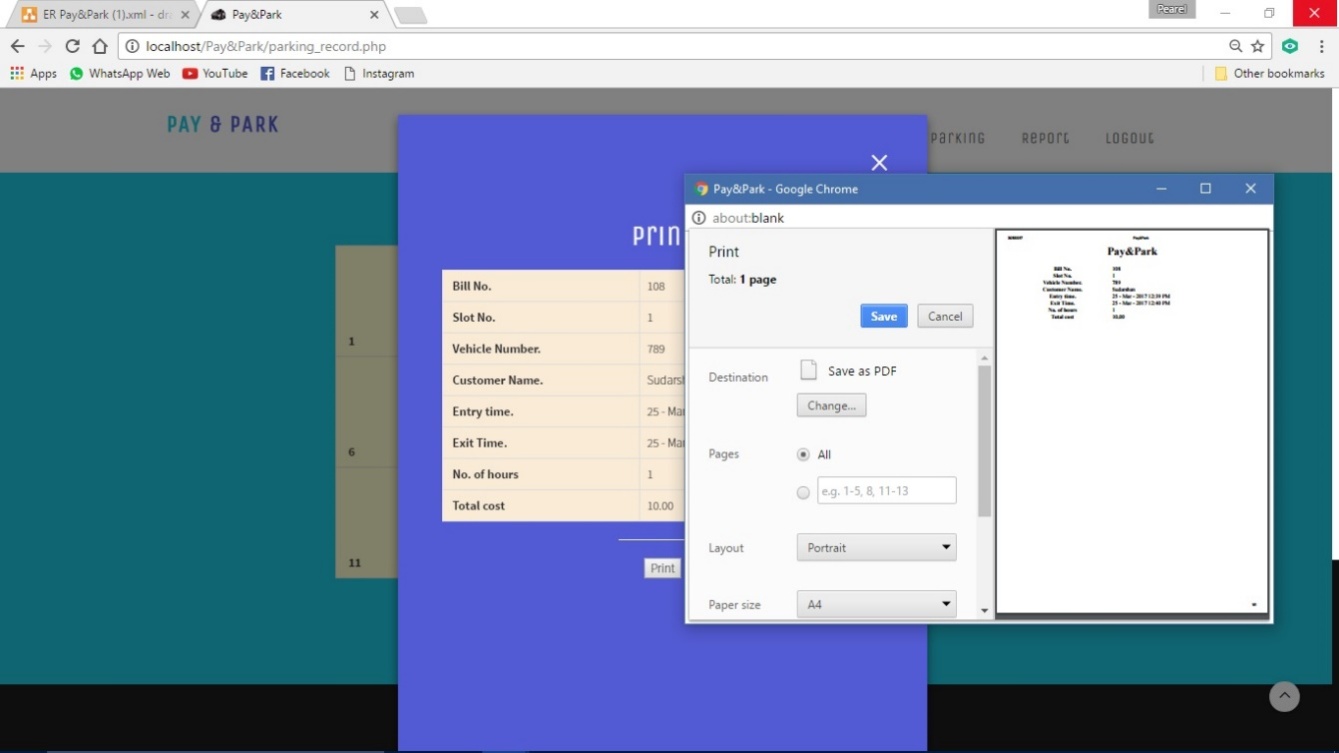
**Add parking record:**

****

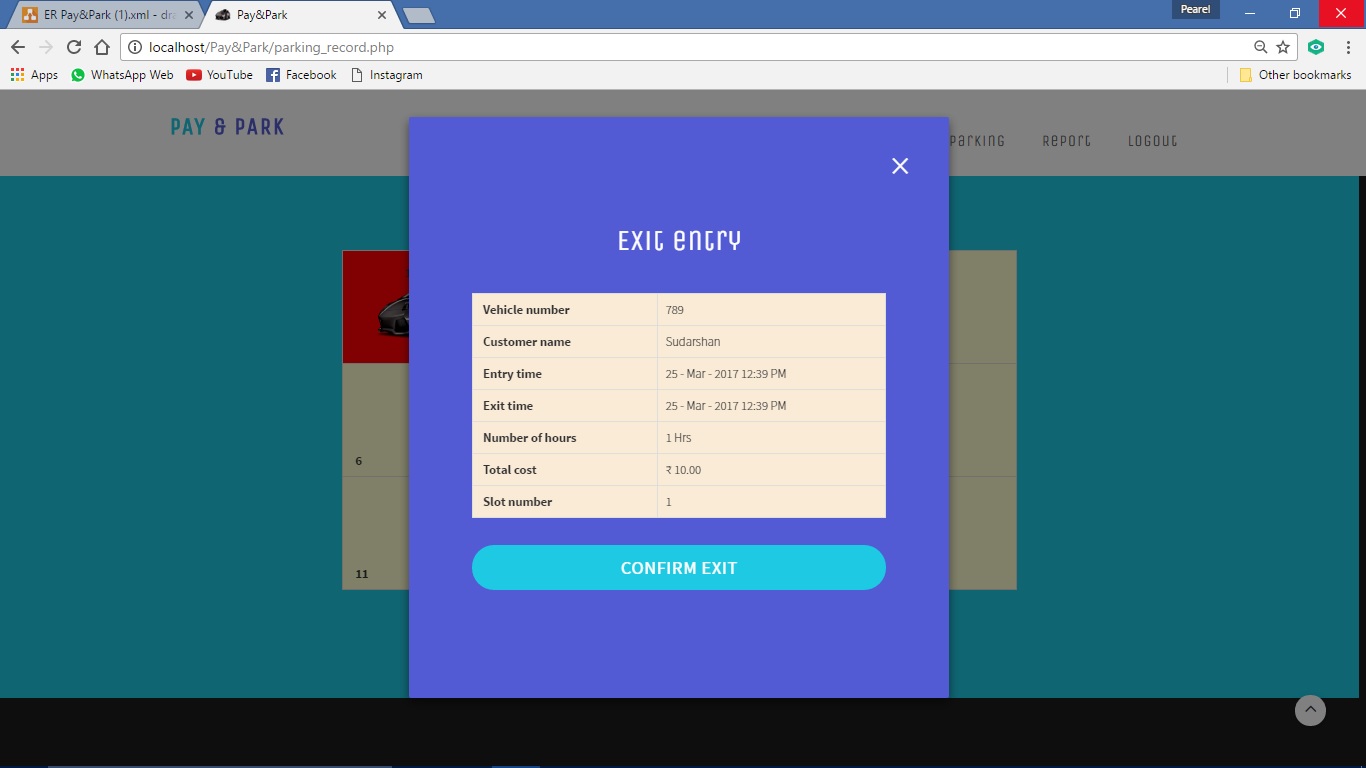
**Adding parking record:**

****

**Printing bill:**

****

**Exit vehicle:**

****

**CONCLUSION AND FUTURE ENHANCEMENT**

**CHAPTER 10**

**CONCLUSION AND FUTURE ENHANCEMENT**

**10.1Conclusion:**

“PAY&PARK” is developed for the purpose of computerizing and maintaining the parking system. Before the records were inserted manually but in the pay& park management system all the work is automated, where employee enters a couple of things and other things like entry time is inserted automatically. When the vehicle exits employee only enters slot number, other things like exit time and parking fees generated automatically and the slot is freed. This helps to maintain the records efficiently.

**10.2 Future Enhancement:**

We can consider much future scope to this application. The following are some of these.

* In future, we can further enhance this software for tablets and mobile phones.
* Later basis software is developed as friendly and attractive
* Add more security features.
* We can develop and integrate with Android or iPhone based apps in the future.
* We can integrate a Card payment system in the future.

**LIMITATION**

**Chapter-11**

**Limitations**

**11.1 Limitations:**

* The user should have the basic knowledge of computer.
* Only the administrator will administer the system.
* It can be accessed over the internet
* Employee can only view the records.

**Bibliography**

**CHAPTER 12**

**BIBLIOGRAPHY**

**12.1 Book references:**

* Software engineering by Pankaj jalote
* HTML and CSS made simple by Ivan Bayross

**12.2 Web reference:**

* [**www.w3schools.com**](http://www.w3schools.com)**.**
* [**www.phptutorials.com**](http://www.phptutorials.com)
* [**www.stackoverflow**](http://www.stackoverflow).