

Mangalore University

**A PROJECT REPORT**

**ON**

**(DREAM BOYS DANCE GALLERY)**

Submitted in partial fulfilment of requirement for the award of the degree

**BACHELOR OF COMPUTER APPLICATION**



**BY**

**Neha R Bangera 201961522133**

**Deekshitha 201961522127**

**Swathi 201961522142**

UNDER THE VALUABLE GUIDANCE OF

Internal Guide External Guide

**Mr. Vishwa Ganapathi Bhat Mr. Guruprasad**

**Lecturer in Computer Application (Dream boys dance studio)**

**Sri Ramakrishna College,**

**Bunt’s Hostel**

**Mangaluru-575003**

**August-2023**

**DECLARATION**

I hereby declare that this project report titled as “**DREAM BOYS DANCE GALLERY**” has been prepared during the year 2022-2023 under the valuable guidance and supervision of Ms. Swetha H Parodylecturer in computer application and project guide Sri Ramakrishna college Mangalore, in partial fulfilment for the award of Bachelor’s in Computer Application.

I also declare that this project is the result of the guidance of external, internal and our own effort and has not been submitted to any other university for the award of any degree.

Date:

Place: Mangalore

Neha

Deekshitha

Swathi

ACKNOWLEDGEMENT

I express my sincere gratitude to all the people who have guided me and made this project a successful on with all their possible enthusiastic support and encouragement.

I express my heartily gratitude to Mr. Vishwa Ganapathi Bhat (internal guide) for their guidance and valuable suggestion during project work and all the Teachers of the B.C.A. department, Sri Ramakrishna College.

Finally, I extend thanks to our parents and friends who are directly and indirectly involved in the completion of our project.

|  |
| --- |
| **INDEX** |

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Content** | **Page No.** |
| 1. | **Introduction**   1. Introduction of the system 2. Project title 3. Category 4. Overview 5. Background 6. Introduction of the system 7. Brief note on existing system 8. Objective of the system 9. Scope of the system 10. Structure of the system 11. System architecture 12. System technical architecture 13. System hardware architecture 14. External interface 15. End users 16. Software/hardware used for the development and implementation | 8-12 |
| 2. | **Software Requirement Specification**   1. Introduction 2. Purpose 3. Scope 4. Definition, Acronyms and Abbreviation 5. Reference 6. Overview 7. Overall Description 8. Product Perspective 9. Product function 10. User characteristics 11. General constraints 12. Assumption and dependencies 13. Specific requirements 14. External interface requirements 15. User interface 16. Hardware interface 17. Software interface 18. Communication interface 19. Functional requirements 20. Performance requirements 21. Design constraints 22. System attributes 23. Other requirements     1. Safety requirements     2. Security requirements | 13-18 |
| 3. | **System Design**   1. Introduction 2. Assumption and constraints 3. Functional decomposition 4. Description of program 5. Context flow diagram 6. Data flow diagram | 19-27 |
| 4. | **Database Design**   1. Introduction 2. Purpose and scope 3. Database identification 4. Schema information 5. Table definition 6. ER diagram | 28-31 |
| 5. | **Detailed design**   1. Introduction 2. Pseudo codes 3. Structure of the software package 4. Structure chart of admin 5. Structure chart of student 6. Structure chart of organiser | 32-36 |
| 6. | **Program code listing**   1. Database connection 2. Authorization/Authentication 3. Add payment 4. Delete batch 5. Index | 37-93 |
| 7. | **User Interface (Screens and Reports)**   1. Login 2. Main Screen / Home page 3. Booking screen 4. Student fees payment screen 5. Add invoice screen 6. Invoice print 7. View 8. Error messages | 93-97 |
| 8. | **Testing**   1. Introduction 2. Testing steps 3. Test cases | 98-102 |
| 9. | Conclusion | 102-103 |
| 10. | Bibliography | 104-104 |

1. **INTRODUCTION**

**1.1 Introduction of the system**

This project is an online website which provides a platform for all age groups to learn and Excel in the most amazing field called dance. Candidates can register online for their desired dance forms. This website will be able to present viewer all related information about the academy as a normal website in the internet.

This project keeps track of all Registration information, Staff information, Batch information, Attendance details and day-to-day transactions. Stock performance like costume rents, props etc will be recorded. It reduces the manual work of the admin as well as the users.

**1.1.2** **Project Title**

**“DREAM BOYS DANCE GALLERY”**

**Category**

Web based RDBMS (Relational Database Management System)

**1.1.3 Overview**

Dream boys dance gallery website provides platform for people to book events through online. They can also register for classes. People can contact them from the contact us section. Students are placed in a class appropriate to their age and experience.

* 1. **Background**

**1.2.1 Introduction Of the company**

Name: Guru prasad

Address: Top floor, kavoor tower, near union bank kavoor, Mangalore

Contact no: 9731275291

* + 1. **Brief note on existing system**

In the existing system, user must register and book the services manually. The manual booking of events will take lot of time and hard work, and they will get limited information. Many people’s do not know about the company and their services.

**1.3 Objectives of the System:**

* It provides better and efficient service.
* To reduce paper work and provide data security.
* Faster way to get information about the classes.
* To save both time and effort of the user.
* People can contact for any choreography.
* To provide full details about the student and attendance.

**1.4 Scope of system**

* Here users can directly book them for shows and contact them for any kind of choreography.
* This website is secure because user must enter login details to access images and other contents.
* It will provide functionality to manage the system and the user information.

**1.5 Structure of the system:**

**Organizer**

**Admin**

**Student**

**Dream Boys Dance Gallery**

In Dance gallery admin can view the registered student details in website. He can manage the employee like adding, removing who is no longer in gallery. Admin can manage the order placed by the customer and payment done by the customer. Admin can view all the feedback of customer. Here Admin can check the order and booking details of the customer. Employee can view the bill details. Customer can login to the website if they want to make orders. Customer can pay the total bill/fees amount through bank transfer and UPI method of payment. Customer can give the feedback about the service and products.

**Admin:**

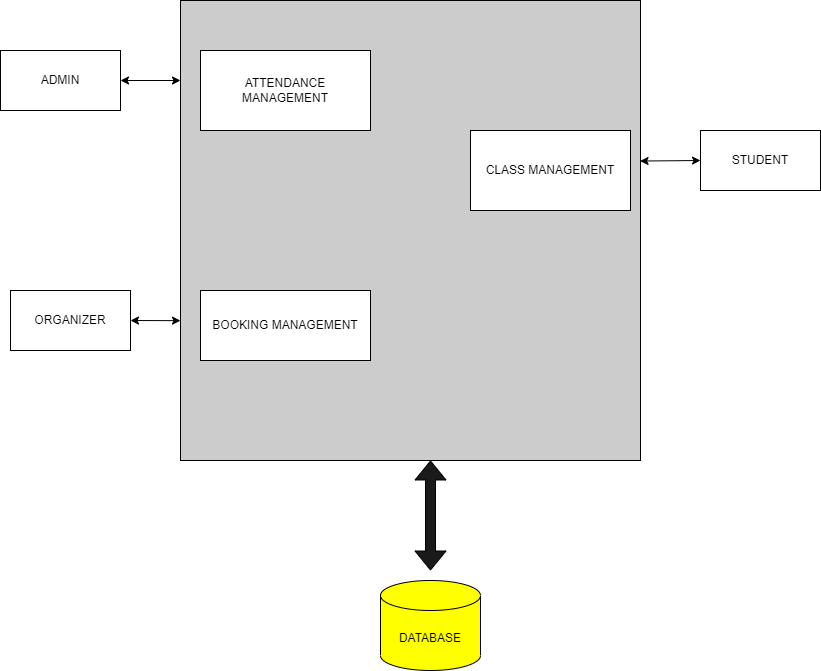
* Login credentials
* Add batch
* View batch
* Add student
* View student
* Add attendance
* View attendance
* Add invoice
* View dance book
* Add student payments
* View student payment
* Logout

**Student:**

* About us
* Registration
* View classes

**Organizer:**

* Book events
* View about us
* View classes
  1. **System Architecture**



**End User**

* **Admin**

Admin can view the registered students to the website. He can add, update and delete the categories of the services. Also he can add, delete and update the students/choreographers. He can confirm and reject the booking. We can say that admin has overall control of the system. He can view the feedback also. Admin can upload the real images of the events.

* **Students**

Students can view the details of both attendance and bookings. He can view the event images and feedback.

* **Organiser**

Organizer can go to the website and book them for events. He can make the payment and give the feedback also.

**1.11** **Software / Hardware used for the development and implementation**

* **Hardware:**
  + - Processor : Intel dual core or above
    - Processor Speed :Minimum 1.8 GHz
    - RAM :2 GB or above
    - Hard Disk :Minimum 40 GB
    - Input Devices :Mouse, Keyboard
    - Output Devices : Monitor, Printer
* **Software:**
* Text Editor : Sublime text 3
* Web server : Apache
* Database Server : MySQL
* Language : PHP
* User Interface Design : HTML, CSS, JavaScript
* Operating System : Windows, Linux
* Browsers : Chrome, Firefox

# 2.SOFTWARE REQUIREMENT SPECIFICATION

**2.1 INTRODUCTION:**

SRS- is a complete description of the behaviour of a system to be developed. It deals with the requirements of the proposed system. It describes what the proposed system should do without describing how the system will do it. The SRS helps the client to determine if the system meets the requirement.

The following section provides an overview of the derived Software Requirements Specification(SRS) for the application “Dream Boys Dance Studio Management System” web application. To begin with, the purpose of the document is presented and its intended audience outlined. Subsequently, the scope of the project specified by the document is given with a particular focus on what the resultant software will do and the relevant benefits associated with it. The nomenclature used throughout the SRS is also offered. To conclude, a complete document overview is provided to facilitate increased reader comprehension and navigation.

**2.1.1 PURPOSE:**

The purpose of this document is to give detailed description of the requirements for the Dream Boys Dance Studio Management System. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to users to registration, event booking and, attendance etc.

**2.1.2 SCOPE:**

The main objective of the project “Dream boys Dance Studio” is to provide an online facility for students to register themselves, users to book events and admin to manage attendance, payment and stock details. Today is the era of computers. Users can browse and look at their profiles and check for required details.

**2.1.3 DEFINITION, ACRONYMS AND ABBREVIATIONS:**

1. PHP: Hypertext Pre-Processor.

2. CSS: Cascading Style Sheet.

3. HTML: Hyper Text Markup Language.

4. OS: Operating System.

5. RDBMS: Relational Data Base Management System.

6. SRS: Software Requirement Specifications.

7. DFD: Data flow Diagram.

8 CFD: Context Flow Diagram.

9. ERD: Entity Relationship Diagram.

10. MYSQL: MY Structured Query Language.

11. GUI: Graphical User Interface.

12. Web-based Application: An application that runs on the internet.

**2.1.4 REFERENCE:**

Website:

[www.C#tutorial.com](http://www.C#tutorial.com)

[www.w3schools.com](http://www.w3schools.com)

**2..1.5 Overview:**

The remainder of this document is divided into two sections. It provides the functional requirements and constraints for the system as well as how it will operate from a user’s perspective. Then it also provides a more detailed system specification including design and structure.

**2.2 Overall Description:**

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts. This project is developed for registration, event booking.

**2.2.1 PRODUCT FUNCTION:**

* Admin
* Student
* Organizer
  + **Admin Module:**

Admin will be provided with the login ID and password in our website. The administrator is the trusted personality who can change users and their information. The admin keeps the information as sensitive possible. The user are validated and approved by the admin after verification.

The Functionality of admin is as follows:

* + Login credentials
  + Add batch
  + View batch
  + Add student
  + View student
  + Add attendance
  + View attendance
  + Add invoice
  + View dance book
  + Add student payments
  + View student payment
  + logout
  + **Student Module:**

The Student module will need to Register themselves. Students can view the classes and attendance.

The Functionality of Student module is as follows:

* + View About us
  + Book class
  + View dance forms
* **Organizers Module:**

Organizers modules is used to book events online. Organizers are those people who organize the event. They can view the schedules.

The Functionality of Organizers module is as follows:

* + Book class
  + View about us
  + View dance forms

**2.1.2 USER CHARCTERSTICS:**

There are 2 types of user that interact with the site admin and organizer. User can login through website. Each of these have different tasks which is performed. Admin is able to modify the details.

**2.1.3 GENERAL CONSTRAINTS:**

No internet connection required for Dream Boys Dance Studio.

The user who accessing the system should be authorized.

**2.1.4 ASSUMPTION AND DEPENDENCES:**

It is assumed that user can view the site and can register if they require the services.

The Administration should be careful in modifying or deleting any data which will cause inconsistency in the database.

**2.3 SPECIFIC REQURIMENTS:**

**2.3.1 External Interfaces Requirements:**

**2.3.2 User Interfaces:**

The user interfaces are user friendly the user will have restricted access and system should not allow access to unauthorized user. Appropriate error messages are generated when a user perform an operation which is invalid.

**2.3.3 Hardware Interfaces:**

* Processor: AMD Ryzen 3 5300U with Radeon Graphics
* RAM: 8.00GB
* Hard Disk: Minimum 150GB

**2.3.4 Software Requirements:**

* Front End: HTML5, CSS3, JavaScript and bootstrap
* Scripting Language: PHP
* Database: MySQL
* Browser: Chrome, Internet Explorer, Firefox.

**2.3.5 Communicational Interfaces:**

Communication is done through internet.

**2.4 Functional Requirements:**

**Registration:**

To enter the page, only user has to register with user name and password. Requirement of registration are full name, username, email id, password.

**Login:**

The System provides facility to login the system. Admin, Vendor and Customer can login to the system by entering valid username and password.

**Logout:**

After using this website Admin, Vendor and Customer can logout.

**2.5 NON-FUNCTIONAL REQUIREMENTS:**

* **Performance requirements:**

The response time should not vary with the increasing the size of the data storage.

* **Security Requirements:**

This purposed website is secure. Administrators can be able to modify and delete the data.

**Software Quality Attributes:**

**Security:**

The security of the system is maintained by giving access to only authentication of user name and password.

**Reliability:**

Good Validation of user input will be done to avoid incorrect storage and records.

**Maintainability:**

During the maintenance stage SRS document can be referenced for the validation.

**Timeless:**

The system carries out all operators will consumption of very less time.

**3.System Design**

**3.1 Introduction**

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of the systems theory to product development. Systems design implies a systematic approach to the design of the system. It may take a bottom-up on top-down approach, but either way the process is systematic where in takes in to account all related variables of the system that needs to be created—from the architecture, to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system.

**3.2 Assumptions and Constraints**

* Itis assumed that all the data entered by the user are correct.

**3.2.1 Constraints**

* This system requires an internet connection.
* Each user must keep their password as confidential.
* No one has rights to change the information of someone else account on this website.
* The end system should also allow for seamless recovery, without data loss, individuals device failure.
* This system is provisioned to be built on the object-oriented PHP which is highly flexible.
* Decision regarding which database to use should be taken considering the fact that data being exchanged or stored is large, and the appropriate data management system will yield efficient performance.

**3.3 Functional Decomposition**

Functional decomposition corresponds to the various functional relationships as how the original complex business function was developed. It mainly focuses on how the overall functionality is developed and its interaction between various components. Large or complex functionalities are more easily understood when broken down into pieces using functional decomposition.

**3.3.1 System Software Architecture**

The XAMPP software has following architecture:

**Client**

**Client**

Web-Server and Business Logic

Data Base

Services

Dynamic HTML-Pages

**Server**

Static HTML- pages

**Figure 3.3.1 System Software Architecture**

**3.4 Description of Programs**

**3.4.1 Context Flow Diagram (CFD)**

Context flow diagram is a top-level data flow diagram. It only contains one process node that generalizes the function of the entire system in relationship to external entities. In context diagram the entire system is treated as a single process and all its inputs, outputs, sinks and sources are identified and shown.

Response

Request

**USER**

**ADMIN**

Request

Response

**3.4.2 Data Flow Diagram (DFD)**

Data Flow Diagram is a graphical representation of a system or a portion of the system. It consists of data flows, process, sources and sink and stores all the description through the use of easily understandable symbols.

DFD is one of the most important modeling tools. It is used to model the system, Components that interact with the system, uses the data and information flows in the system.

DFD shows the information moves through the and how it is modified by a series of transformations. It is a graphical technique that depicts information moves from input or output.

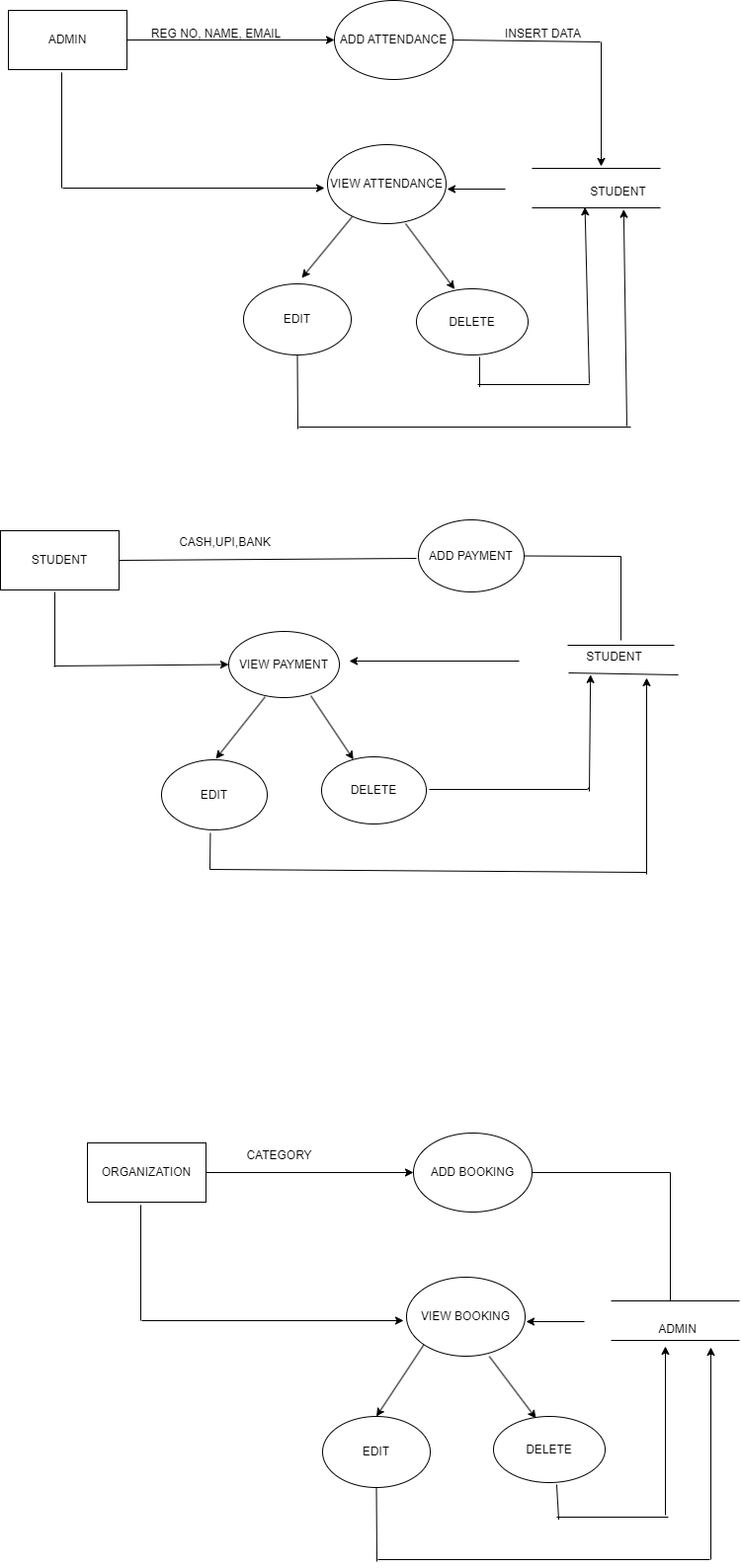
DFD is also knows as bubble chart or Data Flow Graphs. DFD may be used to represent the system at any of abstraction. DFD’s may partition into a level that represents increasing information flows and functional details.

Rules Regarding DFD Construction:

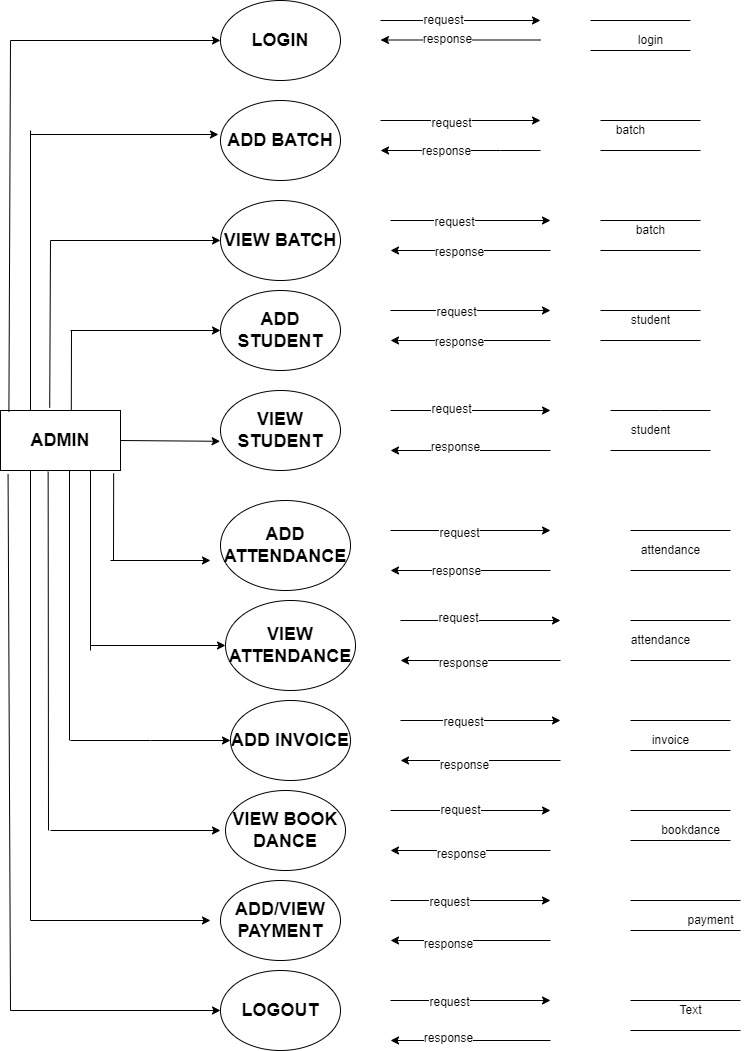
* A process cannot have only outputs.
* A process cannot have only inputs.
* The inputs to a process must be sufficient to produce the outputs from the process.
* All data stores must be connected to at least one process.
* All data stores must be connected to a source or sink.
* A data flow can have only one direction of flow. Multiple data flows to and/or from the same process and data store must be shown by separate arrows.
* If the exact same data flows to two separate arrows, it should be represented by a forked arrow.
* Data cannot flow directly back into the process it has just left. All data flows must be named using a noun phrase

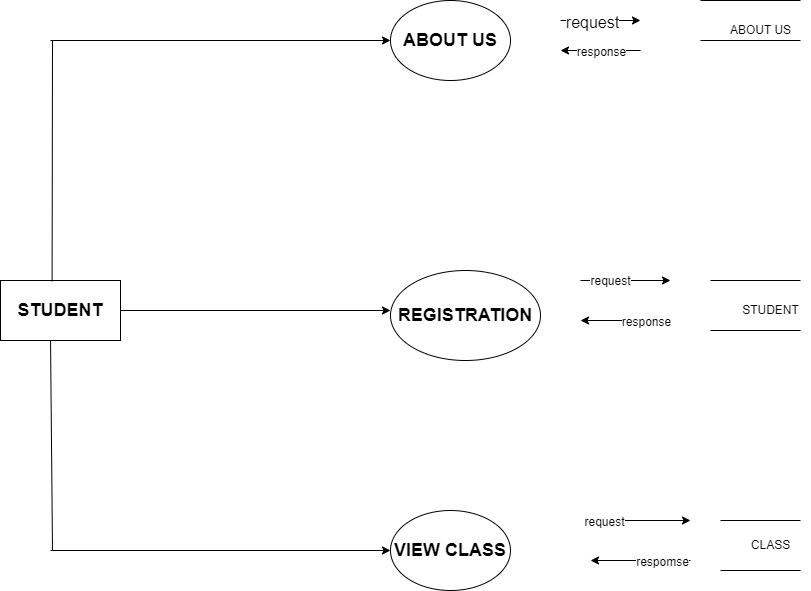
|  |  |  |
| --- | --- | --- |
| **Name** | **Notation** | **Description** |
| Process |  | A process transforms incoming data flow into outgoing data flow. The processes as shown by named circles. |
| Data Store |  | Data stores are repositories of data in the system. They are sometimes also reffered to as files. |
| Data Flow |  | Data flows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it. |
| External Entity |  | External entities are objects outside the system with which the system communicates. External entity are sources and destinations of the system’s inputs and outputs. |

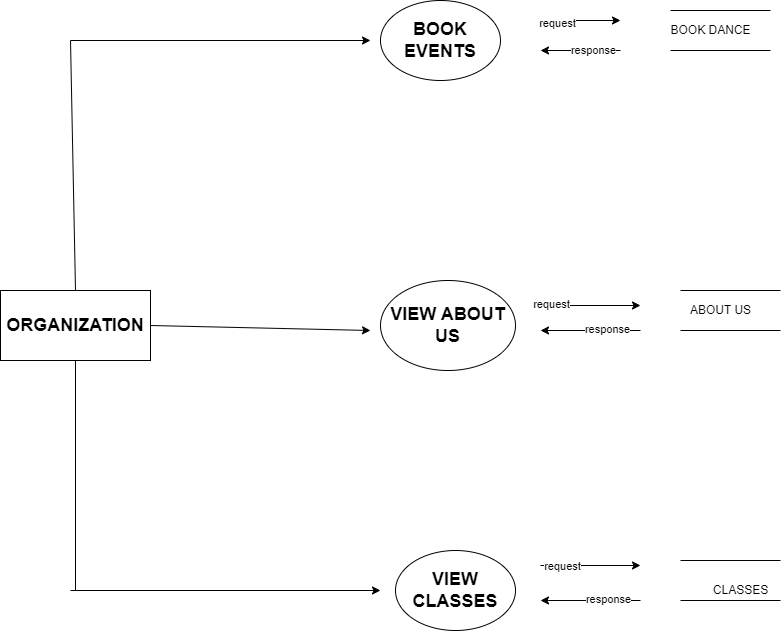
**DFD Level 1:**

****

**DFD level 2:**

****

****

****

**4.Database design**

**4.1 INTRODUCTION**

Database description describes all the databases used in the software to store all the records. The database in turn is further described in detail giving all the fields used with their data type, constraints available to them and descriptions. Constraints include primary key, foreign key, etc., which allow the entities to be uniquely identified.

In this database description we describe all databases which are used to store all the records.

**4.2 Purpose:**

* Database description describes the entire database used in the software to store all the records.
* Database design is the process of producing a detailed data model of a database. Database design is a collection of related data.
* This document describes standards to use when designing and developing the database.

**4.3 Table definition:**

**Admin:**

|  |  |  |  |
| --- | --- | --- | --- |
| ATTRIBUTE | DATA TYPE | LENGTH | DESCRIPTION |
| Id | int | 11 | Primary key |
| Email | varchar | 200 | Not null |
| Password | varchar | 200 | Not null |

**Student:**

|  |  |  |  |
| --- | --- | --- | --- |
| ATTRIBUTE | DATA TYPE | LENGTH | DESCRIPTION |
| Id | Int | 11 | Primary key |
| Sname | Varchar | 200 | Not null |
| Email | Varchar | 200 | Not null |
| Phone | Varchar | 12 | Not null |
| Btype | Int | 12 | Not null |
| Address | Mediumtext | - | Not null |
| image | varchar | 1000 | Not null |

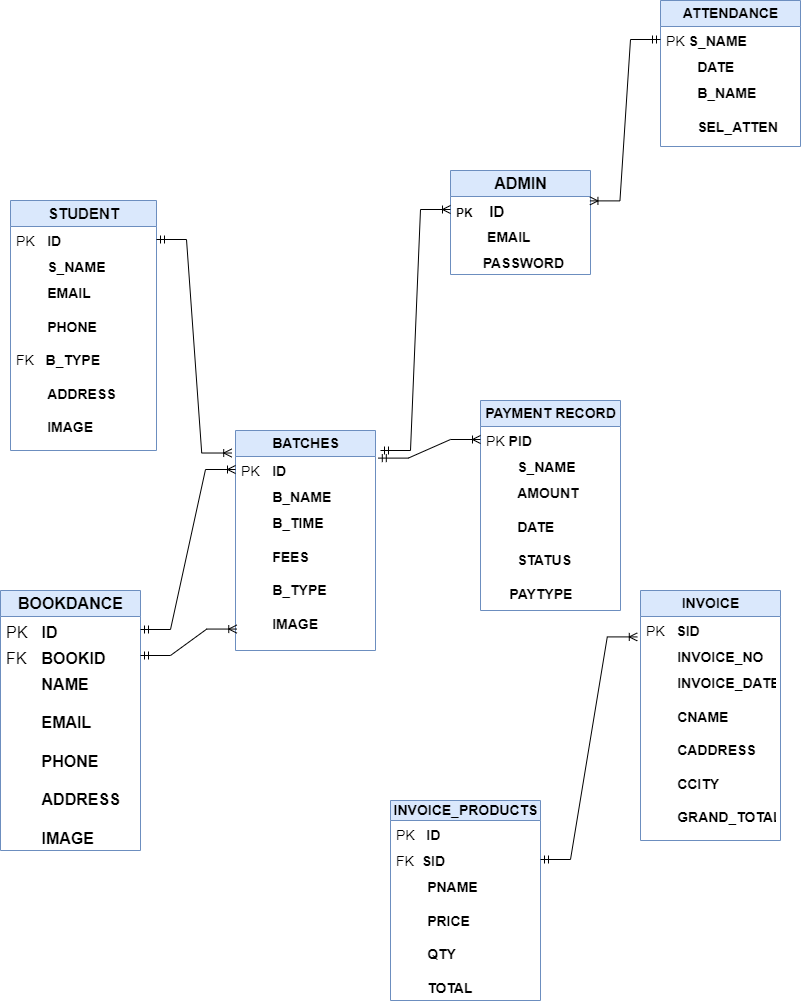
**Batches:**

|  |  |  |  |
| --- | --- | --- | --- |
| ATTRIBUTE | DATA TYPE | LENGTH | DESCRIPTION |
| Id | Int | 11 | Primary key |
| Bname | Varchar | 200 | Not null |
| Btime | Varchar | 200 | Not null |
| Fees | int | 20 | Not null |
| Btype | Varchar | 100 | Not null |
| image | varchar | 200 | Not null |

**Bookdance:**

|  |  |  |  |
| --- | --- | --- | --- |
| ATTRIBUTE | DATA TYPE | LENGTH | DESCRIPTION |
| Id | Int | 11 | Primary key |
| Bookid | Int | 20 | Not null |
| Name | Varchar | 200 | Not null |
| Email | Varchar | 200 | Not null |
| Phone | Varchar | 12 | Not null |
| Address | Mediumtext | - | Not null |
| image | varchar | 500 | Not null |

**4.4 Schema diagram**



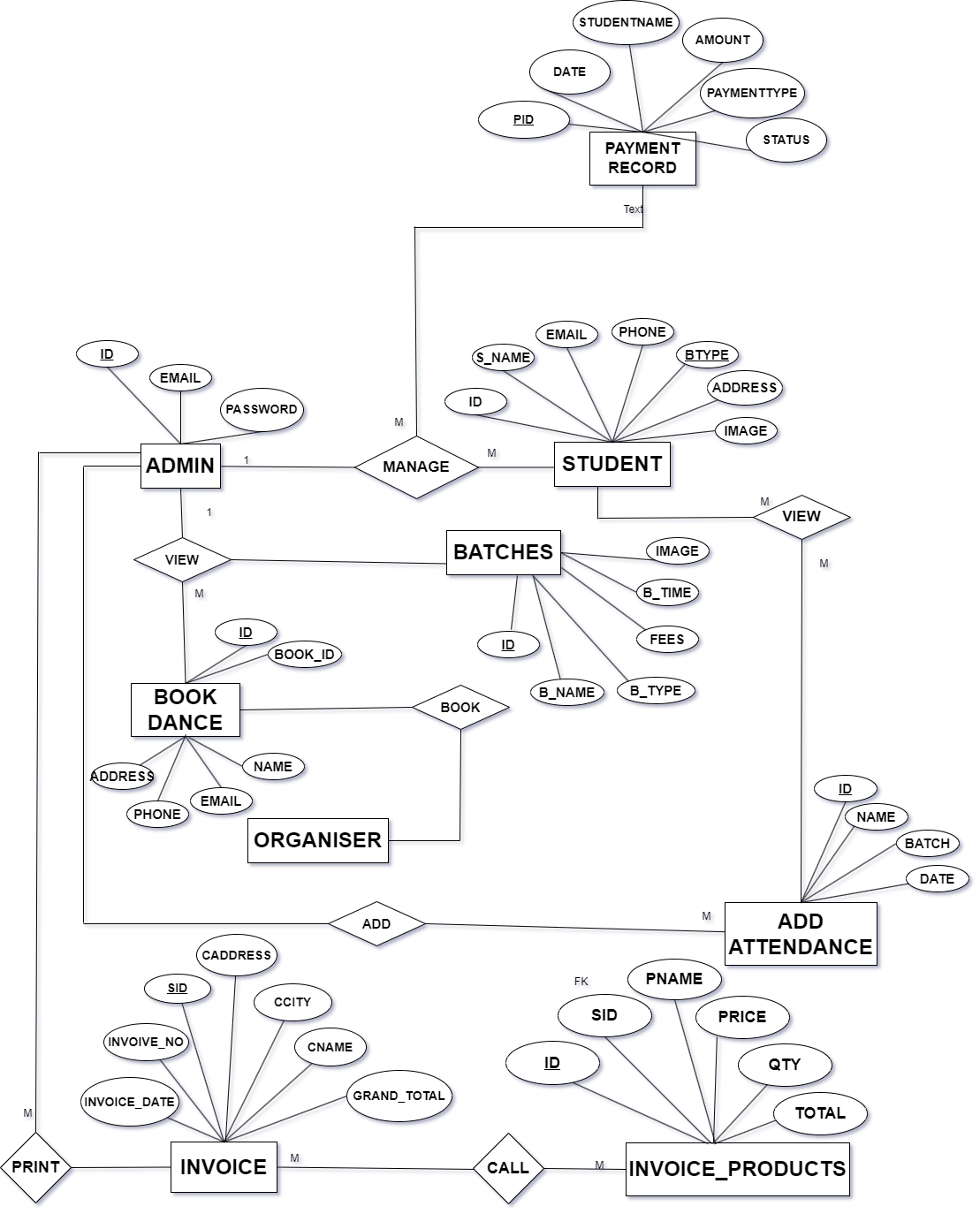
**ER diagram**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagram often use symbols to represent three different types of information. Boxes are commonly used for entities. Diamonds are commonly used to represent relationships and ovals are used to represent attributes.

An entity relationship model (ERM) in software engineering is an abstract and conceptual representation of data. Entity-relationship modeling is a relational schema database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database and its requirements in a top-down fashion.

An entity may be defined as a thing which is recognized as being capable of an independent existence and which can be uniquely identified. An entity is an abstraction from the complexities of some domain. When we speak of an entity we normally speak of some aspect of the real world which can be distinguished from other aspects of the real world.

**5.2 Diagram:**



**5.DETAILED DESIGN**

**5.1 Introduction:**

Detail design starts after the system design phase is completed. The goal of this phase is to develop the internal logic of each of the models identified during the system design.

Before deciding on the logic of the module, formal specification of the module maybe developed. The specification should be such that they are complete, unambiguous and precise and they do not suggest any particular implementations. Two modules units are frequently chosen for formal specifications. Functional modules and data abstraction modules.

Every system requires not only data, but also the structure of the data. A database management system (DBMS) collects and structures related files so that many users can retrieve. manipulate and store data. Here we use MS-SQL as the DBMS. Database is a collection of inter related data stored with minimum redundancy to serve many users quickly and efficiently. Database design are designed to manage large bodies of information and also for easy and flexible retrieval of data.

**5.2 Pseudo code:**

Login page:

BEGIN

Input the username

Input the password

IF the username and password valid

Display successful message

ELSE

Display failure message

Delete:

BEGIN

IF option select is DELETE THEN

DELETE selected data in the database

END IF

END

Update:

BEGIN

IF option selected is UPDATE THEN

UPDATE data in the database

ENDIF

Redirect to main page

END

User registration:

BEGIN

Get user information

Validate user information

IF user information are valid then

Store user information in register file

Print “registration successful”

ELSE

Display error message

END

ADD STUDENT:

BEGIN

Input the student details

IF student information is valid THEN

Store student information in student database

Print “added successfully”

ELSE

Display error message

END

**5.3 Structured chart:**

5.3.1 Structure chart of admin:

**Admin**

**Login**

**Add Batch**

**Add Student**

**Add Attedence**

**Add Payment**

**View**

**Add**

**View**

**View**

U**pdate**

**Delete**

**Delete**

**Delete**

**view**

**Delete**

**5.3.2 Structure chart of student:**

**Student**

**Home**

**About us**

**Classes**

**Book Class**

**View**

**Book**

**View**

**View**

**View**

**5.3.3 Structure chart of organizer:**

**Organiser**

**Home**

**About us**

**Classes**

**Book Class**

**View**

**Book**

**View**

**View**

**View**

**6.CODING**

**Introduction:**

The goal of coding or programming phase is to translate the system design, produced during the designing phase, into code in a given programming language which can be executed by a computer and that performs the computation specified by the design. During the implementation, it should be kept in mind that the programs should not be constructed so that they are easy to write, but that they are easy to read and understand.

Programming style:

Here we will list some general rules that can be applied for writing good code.

1. Names: Selecting module and variable names is often not considered important novice programmers. Most variable sin a program reflect some entity in the problem domain, and the modules reflect some process. Variable names should be closely related to the entity they represent, and module names should reflect their activity.
2. Control constraints: As discussed earlier, it is desirable that as much as possible single entry, single exit constructs be used. It is also desirable to use a few standard control constructs rather than using a wide variety of constructs, just because they are available in the language.
3. Information hiding: As discussed earlier, information hiding should be supported where possible. Only the access functions for the data structures should be made visible while hiding the data structures behind these functions.

IV. User-defined types: Modern languages allow users to define data applicable.

Nesting: The different control constructs, particularly the if-then-else, can be nested. If the nesting becomes too deep, the programs become harder to understand. In case of deeply nested if-then-else’s, it is often difficult to determine if statement to which a particular else clause is associated. If possible, deep nesting should be avoided.

**Program Code Listing**

**6.1 Database connection**

<?php

$severname= "localhost";

$username= "root";

$password= "";

$dbname= "dreamboys";

$conn= new mysqli($severname,$username,$password,$dbname);

if($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

?>

**6.2 Authorization/Authentication**

<?php

session\_start();

if(session\_destroy()) {

header("Location:index.php");

}

?>

**6.3 Data store/retrieval/update**

<?php

session\_start();

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

{

header("Location: index.php");

}

include "connection.php";

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

{

$sname= $\_POST["sname"];

$email= $\_POST["email"];

$phone= $\_POST["email"];

$btype= $\_POST["btype"];

$address= $\_POST["address"];

$totalnoclass= $\_POST["totalnoclass"];

$image= $\_FILES['image'] ['name'];

$imageTemname1= $\_FILES['image'] ['tmp\_name'];

$tarpath= "studentuploads/";

$query = "INSERT INTO student (sname,email,phone,btype,address,image,totalclass) Values('$sname','$email','$phone','$btype','$address','$image','$totalnoclass')";

$run = $conn->query($query) or die("Error in saving Data".$conn->error);

if ($run) {

move\_uploaded\_file($imageTemname1,$tarpath.$image);

echo '<script>alert("Student Added Successfull!")</script>';

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

}

}

?>

<!DOCTYPE html>

<!--

// WEBSITE: https://themefisher.com

// TWITTER: https://twitter.com/themefisher

// FACEBOOK: https://www.facebook.com/themefisher

// GITHUB: https://github.com/themefisher/

-->

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8" />

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

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

<title>Mono - Responsive Admin & Dashboard Template</title>

<!-- theme meta -->

<meta name="theme-name" content="mono" />

<!-- GOOGLE FONTS -->

<link href="https://fonts.googleapis.com/css?family=Karla:400,700|Roboto" rel="stylesheet">

<link href="plugins/material/css/materialdesignicons.min.css" rel="stylesheet" />

<link href="plugins/simplebar/simplebar.css" rel="stylesheet" />

<!-- PLUGINS CSS STYLE -->

<link href="plugins/nprogress/nprogress.css" rel="stylesheet" />

<link href="plugins/DataTables/DataTables-1.10.18/css/jquery.dataTables.min.css" rel="stylesheet" />

<link href="plugins/jvectormap/jquery-jvectormap-2.0.3.css" rel="stylesheet" />

<link href="plugins/daterangepicker/daterangepicker.css" rel="stylesheet" />

<link href="https://cdn.quilljs.com/1.3.6/quill.snow.css" rel="stylesheet">

<link href="plugins/toaster/toastr.min.css" rel="stylesheet" />

<!-- MONO CSS -->

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

<!-- FAVICON -->

<link href="images/favicon.png" rel="shortcut icon" />

<!--

HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries

-->

<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

<script src="plugins/nprogress/nprogress.js"></script>

</head>

<body class="navbar-fixed sidebar-fixed" id="body">

<script>

NProgress.configure({ showSpinner: false });

NProgress.start();

</script>

<div id="toaster"></div>

<!-- ====================================

——— WRAPPER

===================================== -->

<div class="wrapper">

<!-- ====================================

——— LEFT SIDEBAR WITH OUT FOOTER

===================================== -->

<?php include "sidebar.php" ?>

<!-- ====================================

——— PAGE WRAPPER

===================================== -->

<div class="page-wrapper">

<!-- Header -->

<?php include "header.php" ?>

<!-- ====================================

——— CONTENT WRAPPER

===================================== -->

<div class="content-wrapper">

<div class="content">

<!-- Top Statistics -->

<div class="row">

<div class="col-md-12">

<div class="card card-default">

<div class="card-header">

<h2>Add Batches</h2>

<a class="btn mdi mdi-code-tags" data-toggle="collapse" href="#collapse-basic-input" role="button"

aria-expanded="false" aria-controls="collapse-basic-input"> </a>

</div>

<div class="card-body">

<div class="collapse" id="collapse-basic-input">

<pre class="language-html mb-4">

</pre>

</div>

<form action="addstudent.php" method="POST" enctype="multipart/form-data">

<div class="form-group">

<label for="exampleFormControlInput2">Student Name</label>

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

</div>

<div class="form-group">

<label for="exampleFormControlPassword">Email</label>

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

</div>

<div class="form-group">

<label for="exampleFormControlPassword">Phone Number</label>

<input type="text" class="form-control" id="exampleFormControltext" name="phone" placeholder="Phone Number" required>

</div>

<div class="form-group">

<label for="exampleFormControlPassword">Total No of Class</label>

<input type="number" class="form-control" id="exampleFormControltext" name="totalnoclass" placeholder="Total No of Class" required>

</div>

<div class="form-group">

<label for="exampleFormControlPassword">Select Batch</label>

<select class="form-control" id="exampleFormControlSelect12" name="btype" >

<?php

$i=1;

$query="Select \* from batches";

$result = $conn->query($query);

if ($result->num\_rows > 0 ){

while ($row = $result->fetch\_object()){

?>

<option value="<?php echo $row->id ?>"><?php echo $row->bname ?></option>

<?php

$i++;

}

}

?>

</select>

</div>

<div class="form-group">

<label for="exampleFormControlPassword">Address</label>

<textarea class="form-control" id="exampleFormControltext" name="address"></textarea>

</div>

<div class="form-group">

<label for="exampleFormControlPassword">Student Image</label>

<input type="file" class="form-control" id="exampleFormControltext" name="image" />

</div>

<div class="form-footer mt-6">

<button type="submit" class="btn btn-primary btn-pill" name="submit">Submit</button>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Footer -->

<footer class="footer mt-auto">

<div class="copyright bg-white">

<p>

&copy; <span id="copy-year"></span> Copyright by <a class="text-primary" href="http://www.iamabdus.com/" target="\_blank" >Dream Boys</a>.

</p>>

</div>

<script>

var d = new Date();

var year = d.getFullYear();

document.getElementById("copy-year").innerHTML = year;

</script>

</footer>

</div>

</div>

<!-- Card Offcanvas -->

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

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

<script src="plugins/simplebar/simplebar.min.js"></script>

<script src="https://unpkg.com/hotkeys-js/dist/hotkeys.min.js"></script>

<script src="plugins/apexcharts/apexcharts.js"></script>

<script src="plugins/DataTables/DataTables-1.10.18/js/jquery.dataTables.min.js"></script>

<script src="plugins/jvectormap/jquery-jvectormap-2.0.3.min.js"></script>

<script src="plugins/jvectormap/jquery-jvectormap-world-mill.js"></script>

<script src="plugins/jvectormap/jquery-jvectormap-us-aea.js"></script>

<script src="plugins/daterangepicker/moment.min.js"></script>

<script src="plugins/daterangepicker/daterangepicker.js"></script>

<script>

jQuery(document).ready(function() {

jQuery('input[name="dateRange"]').daterangepicker({

autoUpdateInput: false,

singleDatePicker: true,

locale: {

cancelLabel: 'Clear'

}

});

jQuery('input[name="dateRange"]').on('apply.daterangepicker', function (ev, picker) {

jQuery(this).val(picker.startDate.format('MM/DD/YYYY'));

});

jQuery('input[name="dateRange"]').on('cancel.daterangepicker', function (ev, picker) {

jQuery(this).val('');

});

});

</script>

<script src="https://cdn.quilljs.com/1.3.6/quill.js"></script>

<script src="plugins/toaster/toastr.min.js"></script>

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

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

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

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

<!-- -->

</body>

</html>

**Update bookings:**

<?php

session\_start();

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

{

header("Location: index.php");

}

include "connection.php";

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

require 'php\_mail/autoload.php';

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

{

$id= $\_POST["id"];

$status= $\_POST["status"];

$name=$\_POST["name"];

$email=$\_POST["email"];

if($status == ''){

echo '<script>alert("Empty Selection)</script>';

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

}

else{

$query="UPDATE bookdance SET status='$status' WHERE id='$id'";

$run=$conn->query($query)or die("Error in saving data".$conn->error);

if($run)

{

$mail = new PHPMailer(TRUE);

$mail->setFrom('yathirajshtt@gmail.com', 'Dream Dance Booking Confirmation');

$mail->addAddress($email, $name);

$mail->Subject = 'Dream Dance Booking Confirmed';

$message = '<html><body>';

$message .= '<h4> Dream Dance Booking Confirmed!!!</h4>';

$message .="<p>Name:".$name."</p><br />";

$message .= "</body></html>";

$mail->Body = $message;

$mail->SMTPOptions = array(

'ssl'=>array(

'verify\_peer' =>false,

'verify\_peer\_name' =>false,

'allow\_self\_signed'=>true

)

);

$mail->IsHTML(true);

$mail->isSMTP();

$mail->Host = 'smtp.gmail.com';

$mail->SMTPAuth = TRUE;

$mail->SMTPSecure = 'tls';

$mail->Username = 'yathirajshtt@gmail.com';

$mail->Password = 'zawddzqodecnmtcm';

$mail->Port = 587;

$mail->SMTPDebug = 0;

if($mail->send()){

echo '<script>alert("Successfully Booked ")</script>';

}

else{

echo '<script>alert("Successfully Not Booked ")</script>';

}

echo '<script>alert("Successfully Updated)</script>';

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

}

}

}

?>

<!DOCTYPE html>

<!--

// WEBSITE: https://themefisher.com

// TWITTER: https://twitter.com/themefisher

// FACEBOOK: https://www.facebook.com/themefisher

// GITHUB: https://github.com/themefisher/

-->

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8" />

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

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

<title>Mono - Responsive Admin & Dashboard Template</title>

<!-- theme meta -->

<meta name="theme-name" content="mono" />

<!-- GOOGLE FONTS -->

<link href="https://fonts.googleapis.com/css?family=Karla:400,700|Roboto" rel="stylesheet">

<link href="plugins/material/css/materialdesignicons.min.css" rel="stylesheet" />

<link href="plugins/simplebar/simplebar.css" rel="stylesheet" />

<!-- PLUGINS CSS STYLE -->

<link href="plugins/nprogress/nprogress.css" rel="stylesheet" />

<link href="plugins/DataTables/DataTables-1.10.18/css/jquery.dataTables.min.css" rel="stylesheet" />

<link href="plugins/jvectormap/jquery-jvectormap-2.0.3.css" rel="stylesheet" />

<link href="plugins/daterangepicker/daterangepicker.css" rel="stylesheet" />

<link href="https://cdn.quilljs.com/1.3.6/quill.snow.css" rel="stylesheet">

<link href="plugins/toaster/toastr.min.css" rel="stylesheet" />

<!-- MONO CSS -->

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

<!-- FAVICON -->

<link href="images/favicon.png" rel="shortcut icon" />

<!--

HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries

-->

<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

<script src="plugins/nprogress/nprogress.js"></script>

</head>

<body class="navbar-fixed sidebar-fixed" id="body">

<script>

NProgress.configure({ showSpinner: false });

NProgress.start();

</script>

<div id="toaster"></div>

<!-- ====================================

——— WRAPPER

===================================== -->

<div class="wrapper">

<!-- ====================================

——— LEFT SIDEBAR WITH OUT FOOTER

===================================== -->

<?php include "sidebar.php" ?>

<!-- ====================================

——— PAGE WRAPPER

===================================== -->

<div class="page-wrapper">

<!-- Header -->

<?php include "header.php" ?>

<!-- ====================================

——— CONTENT WRAPPER

===================================== -->

<div class="content-wrapper">

<div class="content">

<!-- Top Statistics -->

<div class="row">

<div class="col-md-12">

<div class="card card-default">

<div class="card-header">

<h2>Update Status</h2>

<a class="btn mdi mdi-code-tags" data-toggle="collapse" href="#collapse-basic-input" role="button"

aria-expanded="false" aria-controls="collapse-basic-input"> </a>

</div>

<div class="card-body">

<div class="collapse" id="collapse-basic-input">

<pre class="language-html mb-4">

</pre>

</div>

<form action="updatebookingstatus.php" method="POST" enctype="multipart/form-data">

<?php

$i=0;

include "connection.php";

$id= (isset($\_GET['id']) ? $\_GET['id'] : '');

$query = "SELECT \* FROM bookdance WHERE id='$id' ";

$result = $conn->query($query);

if ($result->num\_rows > 0 ){

while ($row = $result->fetch\_object()){

?>

<div class="form-group">

<label for="exampleFormControlInput2">Batches</label>

<input type="hidden" class="form-control" id="exampleFormControltext" name="id" value="<?php echo $row->id ?>">

<input type="hidden" class="form-control" id="exampleFormControltext" name="name" value="<?php echo $row->name ?>">

<input type="hidden" class="form-control" id="exampleFormControltext" name="email" value="<?php echo $row->email ?>">

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

<option value="pending">Pending</option>

<option value="confirmed">Confirmed</option>

</select>

</div>

<div class="form-footer mt-6">

<button type="submit" class="btn btn-primary btn-pill" name="submit">Submit</button>

</div>

<?php

}

}

?>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Footer -->

<footer class="footer mt-auto">

<div class="copyright bg-white">

<p>

&copy; <span id="copy-year"></span> Copyright by <a class="text-primary" href="http://www.iamabdus.com/" target="\_blank" >Dream Boys</a>.

</p>

</div>

<script>

var d = new Date();

var year = d.getFullYear();

document.getElementById("copy-year").innerHTML = year;

</script>

</footer>

</div>

</div>

<!-- Card Offcanvas -->

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

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

<script src="plugins/simplebar/simplebar.min.js"></script>

<script src="https://unpkg.com/hotkeys-js/dist/hotkeys.min.js"></script>

<script src="plugins/apexcharts/apexcharts.js"></script>

<script src="plugins/DataTables/DataTables-1.10.18/js/jquery.dataTables.min.js"></script>

<script src="plugins/jvectormap/jquery-jvectormap-2.0.3.min.js"></script>

<script src="plugins/jvectormap/jquery-jvectormap-world-mill.js"></script>

<script src="plugins/jvectormap/jquery-jvectormap-us-aea.js"></script>

<script src="plugins/daterangepicker/moment.min.js"></script>

<script src="plugins/daterangepicker/daterangepicker.js"></script>

<script>

jQuery(document).ready(function() {

jQuery('input[name="dateRange"]').daterangepicker({

autoUpdateInput: false,

singleDatePicker: true,

locale: {

cancelLabel: 'Clear'

}

});

jQuery('input[name="dateRange"]').on('apply.daterangepicker', function (ev, picker) {

jQuery(this).val(picker.startDate.format('MM/DD/YYYY'));

});

jQuery('input[name="dateRange"]').on('cancel.daterangepicker', function (ev, picker) {

jQuery(this).val('');

});

});

</script>

<script src="https://cdn.quilljs.com/1.3.6/quill.js"></script>

<script src="plugins/toaster/toastr.min.js"></script>

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

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

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

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

<!-- -->

</body>

</html>

**Delete Batch:**

<?php

include 'connection.php';

$id=(isset($\_GET['id'])?$\_GET['id']:'');

$query = "DELETE FROM batches where id='$id' ";

if($conn->query($query)==TRUE){

echo '<script>alert(Dance Batches successfully deleted")</script>';

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

}

?>

**Add payment:**

<?php

session\_start();

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

{

header("Location: index.php");

}

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

require 'php\_mail/autoload.php';

include "connection.php";

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

{

$studentname= $\_POST["studentname"];

$email= $\_POST["email"];

$date= $\_POST["date"];

$amount= $\_POST["amount"];

$status= 'paid';

$paymenttype= $\_POST['paymenttype'];

$query = "INSERT INTO paymentrecord(studentname,amount,date,status,paymenttype) Values('$studentname','$amount','$date','$status','$paymenttype')";

$run = $conn->query($query) or die("Error in saving Data".$conn->error);

if ($run) {

$mail = new PHPMailer(TRUE);

$mail->setFrom('yathirajshtt@gmail.com', 'Dream Dance Payment');

$mail->addAddress($email, $studentname);

$mail->Subject = 'Dream Dance Payment Confirmed';

$message = '<html><body>';

$message .= '<h4> Dream Dance Payment Confirmed!!!</h4>';

$message .="<p>Name:".$studentname."</p><br />";

$message .="<p>Your Paid Amount:".$amount." on Date ".$date."</p><br />";

$message .= "</body></html>";

$mail->Body = $message;

$mail->SMTPOptions = array(

'ssl'=>array(

'verify\_peer' =>false,

'verify\_peer\_name' =>false,

'allow\_self\_signed'=>true

)

);

$mail->IsHTML(true);

$mail->isSMTP();

$mail->Host = 'smtp.gmail.com';

$mail->SMTPAuth = TRUE;

$mail->SMTPSecure = 'tls';

$mail->Username = 'yathirajshtt@gmail.com';

$mail->Password = 'zawddzqodecnmtcm';

$mail->Port = 587;

$mail->SMTPDebug = 0;

if($mail->send()){

echo '<script>alert("Payment Added Successfull!")</script>';

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

}

else{

echo '<script>alert("Payment Not Booked ")</script>';

}

}

echo '<script>alert("Payment Added Successfull!")</script>';

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

}

?>

<!DOCTYPE html>

<!--

// WEBSITE: https://themefisher.com

// TWITTER: https://twitter.com/themefisher

// FACEBOOK: https://www.facebook.com/themefisher

// GITHUB: https://github.com/themefisher/

-->

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8" />

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

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

<title>Mono - Responsive Admin & Dashboard Template</title>

<!-- theme meta -->

<meta name="theme-name" content="mono" />

<!-- GOOGLE FONTS -->

<link href="https://fonts.googleapis.com/css?family=Karla:400,700|Roboto" rel="stylesheet">

<link href="plugins/material/css/materialdesignicons.min.css" rel="stylesheet" />

<link href="plugins/simplebar/simplebar.css" rel="stylesheet" />

<!-- PLUGINS CSS STYLE -->

<link href="plugins/nprogress/nprogress.css" rel="stylesheet" />

<link href="plugins/DataTables/DataTables-1.10.18/css/jquery.dataTables.min.css" rel="stylesheet" />

<link href="plugins/jvectormap/jquery-jvectormap-2.0.3.css" rel="stylesheet" />

<link href="plugins/daterangepicker/daterangepicker.css" rel="stylesheet" />

<link href="https://cdn.quilljs.com/1.3.6/quill.snow.css" rel="stylesheet">

<link href="plugins/toaster/toastr.min.css" rel="stylesheet" />

<!-- MONO CSS -->

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

<!-- FAVICON -->

<link href="images/favicon.png" rel="shortcut icon" />

<!--

HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries

-->

<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

<script src="plugins/nprogress/nprogress.js"></script>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css">

</head>

<body class="navbar-fixed sidebar-fixed" id="body">

<script>

NProgress.configure({ showSpinner: false });

NProgress.start();

</script>

<div id="toaster"></div>

<!-- ====================================

——— WRAPPER

===================================== -->

<div class="wrapper">

<!-- ====================================

——— LEFT SIDEBAR WITH OUT FOOTER

===================================== -->

<?php include "sidebar.php" ?>

<!-- ====================================

——— PAGE WRAPPER

===================================== -->

<div class="page-wrapper">

<!-- Header -->

<?php include "header.php" ?>

<!-- ====================================

——— CONTENT WRAPPER

===================================== -->

<div class="content-wrapper">

<div class="content">

<!-- Top Statistics -->

<div class="row">

<div class="col-md-12">

<div class="card card-default">

<div class="card-header">

<h2> Student Fees</h2>

<a class="btn mdi mdi-code-tags" data-toggle="collapse" href="#collapse-basic-input" role="button"

aria-expanded="false" aria-controls="collapse-basic-input"> </a>

</div>

<div class="card-body">

<div class="collapse" id="collapse-basic-input">

<pre class="language-html mb-4">

</pre>

</div>

<form action="addstudentpayment.php" method="POST" enctype="multipart/form-data">

<?php

$i=1;

$querys="Select \* from student";

$results = $conn->query($querys);

if ($results->num\_rows > 0 ){

while ($rows = $results->fetch\_object()){ ?>

<input type="hidden" class="form-control" id="exampleFormControlPassword" name="email" value="<?php echo $rows->email ?>" required/>

<?php }} ?>

<div class="form-group">

<label for="exampleFormControlInput2">Student Name</label>

<select class="form-control" id="exampleFormControlPassword" name="studentname" id="studentname">

<?php

$i=1;

$query="Select \* from student";

$result = $conn->query($query);

if ($result->num\_rows > 0 ){

while ($row = $result->fetch\_object()){

?>

<

<option value="<?php echo $row->sname ?>"><?php echo $row->sname ?></option>

<?php

$i++;

}

}

?>

</select>

</div>

<div class="form-group">

<label for="exampleFormControlInput2">amount</label>

<input type="text" class="form-control" id="exampleFormControlPassword" name="amount" required/>

</div>

<div class="form-group">

<label for="exampleFormControlInput2">Date</label>

<input type="date" class="form-control" id="exampleFormControlPassword" name="date" required />

</div>

</div>

<div class="form-group">

<label for="exampleFormControlPassword">Select Payment Type</label>

<select class="form-control" id="exampleFormControlSelect12" name="paymenttype" required >

<option value="cash">cash</option>

<option value="onlinepayment">onlinepayment</option>

</select>

</div>

<div class="form-footer mt-6">

<button type="submit" class="btn btn-primary btn-pill" name="submit">Submit</button>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Footer -->

<footer class="footer mt-auto">

<div class="copyright bg-white">

<p>

&copy; <span id="copy-year"></span> Copyright by <a class="text-primary" href="http://www.iamabdus.com/" target="\_blank" >Dream Boys</a>.

</p>

</div>

<script>

var d = new Date();

var year = d.getFullYear();

document.getElementById("copy-year").innerHTML = year;

</script>

</footer>

</div>

</div>

<!-- Card Offcanvas -->

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

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

<script src="plugins/simplebar/simplebar.min.js"></script>

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

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

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

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

<!-- -->

</body>

</html>

**Index:**

<!DOCTYPE html>

<html>

<head>

<!-- Basic -->

<meta charset="utf-8" />

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

<!-- Mobile Metas -->

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />

<!-- Site Metas -->

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

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

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

<title>Dream Boys Gallery</title>

<!-- slider stylesheet -->

<link rel="stylesheet" type="text/css"

href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.1.3/assets/owl.carousel.min.css" />

<!-- bootstrap core css -->

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

<!-- fonts style -->

<link href="https://fonts.googleapis.com/css?family=Poppins:300,400,600,700|Roboto:300,400,700&display=swap"

rel="stylesheet">

<!-- Custom styles for this template -->

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

<!-- responsive style -->

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

</head>

<body>

<div class="hero\_area">

<!-- header section strats -->

<?php include "navbar.php" ?>

<!-- end header section -->

<!-- slider section -->

<section class=" slider\_section position-relative">

<div class="number-container d-none d-md-block">

<div class="number-box">

<span>

01

</span>

<hr>

<span>

02

</span>

</div>

</div>

<div id="carouselExampleControls" class="carousel slide" data-ride="carousel">

<div class="carousel-inner">

<div class="carousel-item active">

<div class="container slider\_item-box">

<div class="row pt-5">

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

<div class="slider-box-detail">

<h1>

Dance is the hidden language of the soul.

</h1>

<p>

A good dancer is not necessarily defined by great technique, skill, or ability to pick up choreography but my confidence. When you feel the music, it penetrates to your soul.

Until the 20th century, most ballroom dances were sequence dances. The way people moved was planned in set formation. These formations were usually lines or squares. Everyone moved at the same time, and finished at the same time. The music played for a set time, and then stopped. After the invention of the waltz, around 1800, another style of dancing developed.[3] In the waltz, and later dances, people danced in couples, but they did so separately. They did not dance in formation, but moved round the room as they pleased (but anti-clockwise). Often, new dance styles arrive. Some dance as individuals, separately, as they please. Street dance is like that. All these types of dance have music.

At the same time, round the world there are many traditional dances. Some of them have been going for hundreds of years. We call them folkloric dances.

</p>

<div>

<a href="">

<div>

</span>

</div>

</a>

</div>

</div>

</div>

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

<div class="slider\_box-img">

<div class="slider\_item\_img-box">

<img src="images/1.jpg" alt="" class="img-fluid">

</div>

</div>

</div>

</div>

</div>

</div>

<div class="carousel-item">

<div class="container slider\_item-box">

<div class="row pt-5">

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

<div class="slider-box-detail">

<h1>

</h1>

<p>

</p>

<div>

<a href="">

<div>

<span>

Read More

</span>

<span>

<img src="images/right.png" alt="" style="width: 18px;">

</span>

</div>

</a>

</div>

</div>

</div>

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

<div class="slider\_box-img">

<div class="slider\_item\_img-box">

<img src="images/dancer.png" alt="" class="img-fluid">

</div>

</div>

</div>

</div>

</div>

</div>

<div class="carousel-item">

<div class="container slider\_item-box">

<div class="row pt-5">

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

<div class="slider-box-detail">

<h1>

</h1>

<p>

</p>

<div>

<a href="">

<div>

<span>

Read More

</span>

<span>

<img src="images/right.png" alt="" style="width: 18px;">

</span>

</div>

</a>

</div>

</div>

</div>

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

<div class="slider\_box-img">

<div class="slider\_item\_img-box">

<img src="images/dancer.png" alt="" class="img-fluid">

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="carousel-control">

<a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-slide="prev">

<span class="" aria-hidden="true"></span>

<span class="sr-only">Previous</span>

</a>

<a class="carousel-control-next" href="#carouselExampleControls" role="button" data-slide="next">

<span class="" aria-hidden="true"></span>

<span class="sr-only">Next</span>

</a>

</div>

</div>

</section>

<!-- end slider section -->

</div>

<!-- about section -->

<section class="about\_section">

<div class="container">

<h2>

About Us

</h2>

<div class="row">

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

<div class="about-detail">

<h3>

Dream Boys dance gallery is a well known dance studio located in kavoor.

It was started by Master Guru prasad. This was entirely boys class but now it consists of both boys and girls.

This class takes up dance shows and also host a dance reality show in mangalore. This class has won several awards on their choreography

. This class works on students dance perfection and also gives them several opportunities.

</h3>

<p>

</p>

<div>

<a href="">

<div>

<span>

Read More

</span>

<span>

<img src="images/right.png" alt="" style="width: 18px;">

</span>

</div>

</a>

</div>

</div>

</div>

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

<div class="about-imge-box">

<img src="images/group.jpg" alt="" class="img-fluid">

</div>

</div>

</div>

</div>

</section>

<!-- end about section -->

<!-- class section -->

<section class="class\_section py-5 my-5">

<h2>

Our Classes

</h2>

<div class="container pb-5 ">

<div class="d-flex justify-content-between flex-wrap">

<div class="mt-4 mx-auto mx-sm-0 mt-lg-0">

<div class="class\_img-box box-img-1">

<a href="">

Hip Hop Dance

</a>

</div>

</div>

<div class="mt-4 mx-auto mx-sm-0 mt-lg-0">

<div class="class\_img-box box-img-2">

<a href="">

Ballet Dance

</a>

</div>

</div>

<div class="mt-4 mx-auto mx-sm-0 mt-lg-0">

<div class="class\_img-box box-img-3">

<a href="">

Break Dance

</a>

</div>

</div>

<div class="mt-4 mx-auto mx-sm-0 mt-lg-0">

<div class="class\_img-box box-img-4">

<a href="">

Freestyle Dance

</a>

</div>

</div>

</div>

</div>

<div class="class\_section-btn">

<a href="">

<div>

<span>

Read More

</span>

<span>

<img src="images/red-next.png" alt="" style="width: 18px;">

</span>

</div>

</a>

</div>

</section>

<!-- end class section -->

<!-- master section -->

<section class="master\_section mb-5">

<h2>

Learn From Masters

</h2>

<div class="master\_custom-container">

<div class="row">

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

<div class="master\_content-box">

<div class="master\_img-box">

<img src="images/master-1.png" alt="">

</div>

<div class="master\_detail">

<h4>

Dance Masters

</h4>

<p>

</p>

</div>

</div>

</div>

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

<div class="master\_content-box">

<div class="master\_img-box master\_img-box-white">

<img src="images/master-2.png" alt="">

</div>

<div class="master\_detail">

<h4>

Music Lessons

</h4>

<p>

</p>

</div>

</div>

</div>

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

<div class="master\_content-box">

<div class="master\_img-box">

<img src="images/master-3.png" alt="">

</div>

<div class="master\_detail">

<h4>

Skill Based Coaching

</h4>

<p>

</p>

</div>

</div>

</div>

</div>

<div class="master\_section-btn">

<a href="">

<div>

<span>

Read More

</span>

<span>

<img src="images/right.png" alt="" style="width: 18px;">

</span>

</div>

</a>

</div>

</div>

</section>

<!-- end master section -->

<!-- contact section -->

<section class="contact\_section py-5">

<div class="container">

<div class="d-flex justify-content-center d-md-block">

<h2>

Contact Us

</h2>

</div>

<div class="row">

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

<form action="">

<div class="contact\_form-container">

<div>

<h5>

Address

</h5>

<p>

Top floor, kavoor tower, near Union Bank, kavoor,Mangalore.

</p>

<p>

91 97312 75291 </p>

<p>

dreamboys@gmail.com

</p>

</div>

</div>

</form>

</div>

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

<div class="contact\_img-box">

<img src="images/2.jpg" alt="">

</div>

</div>

</div>

</div>

</section>

<!-- end contact section -->

<!-- info section -->

<?php include "footer.php" ?>

<!-- footer section -->

<script type="text/javascript" src="js/jquery-3.4.1.min.js"></script>

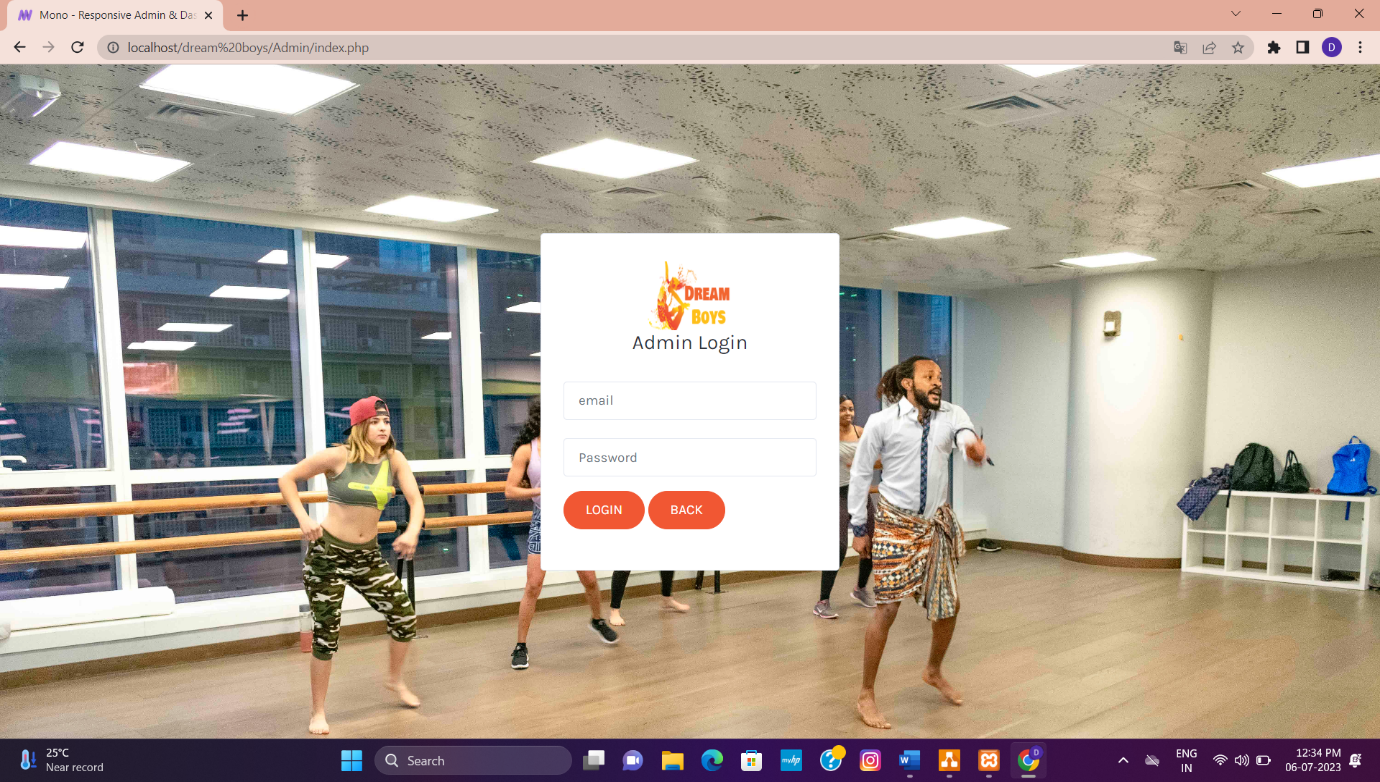
<script type="text/javascript" src="js/bootstrap.js"></script>

</body>

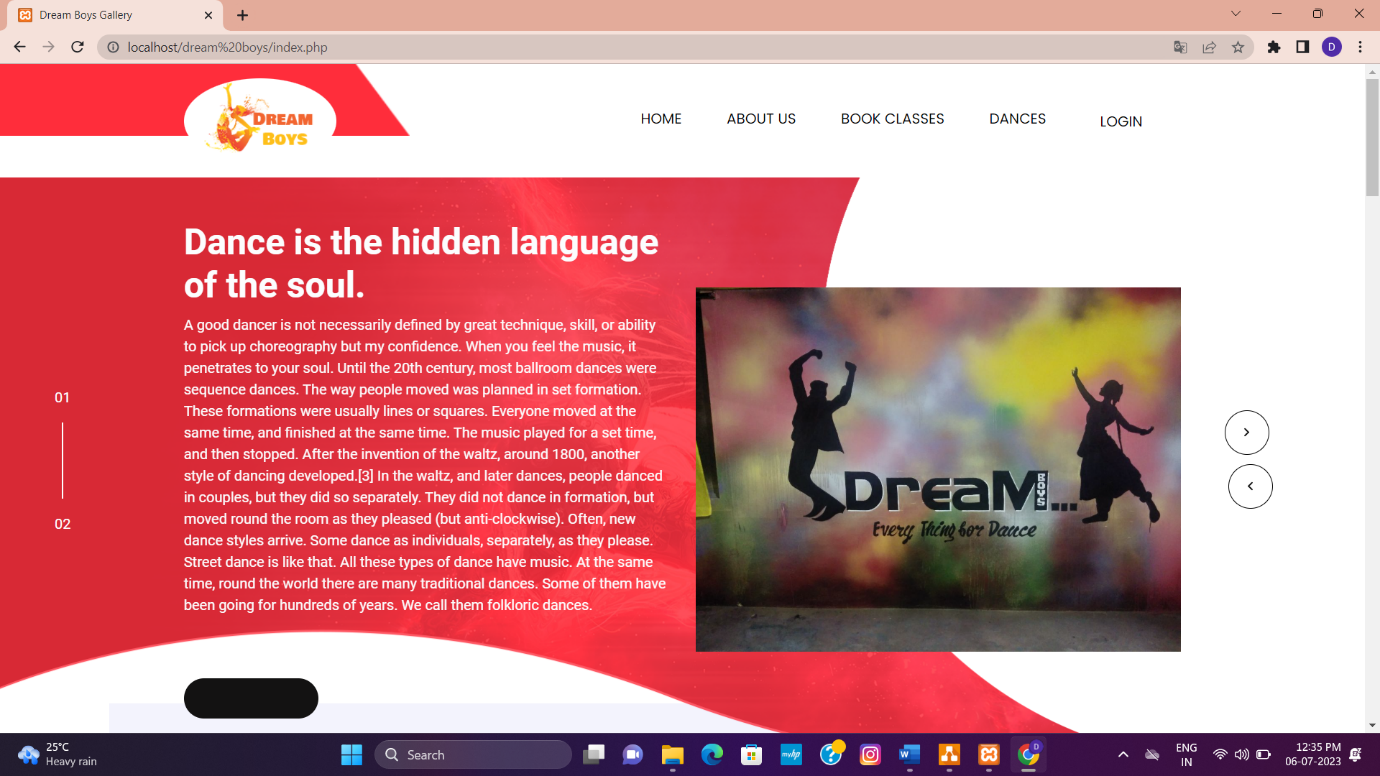
</html>

**7.USER INTERFACE**

**7.1 login:**



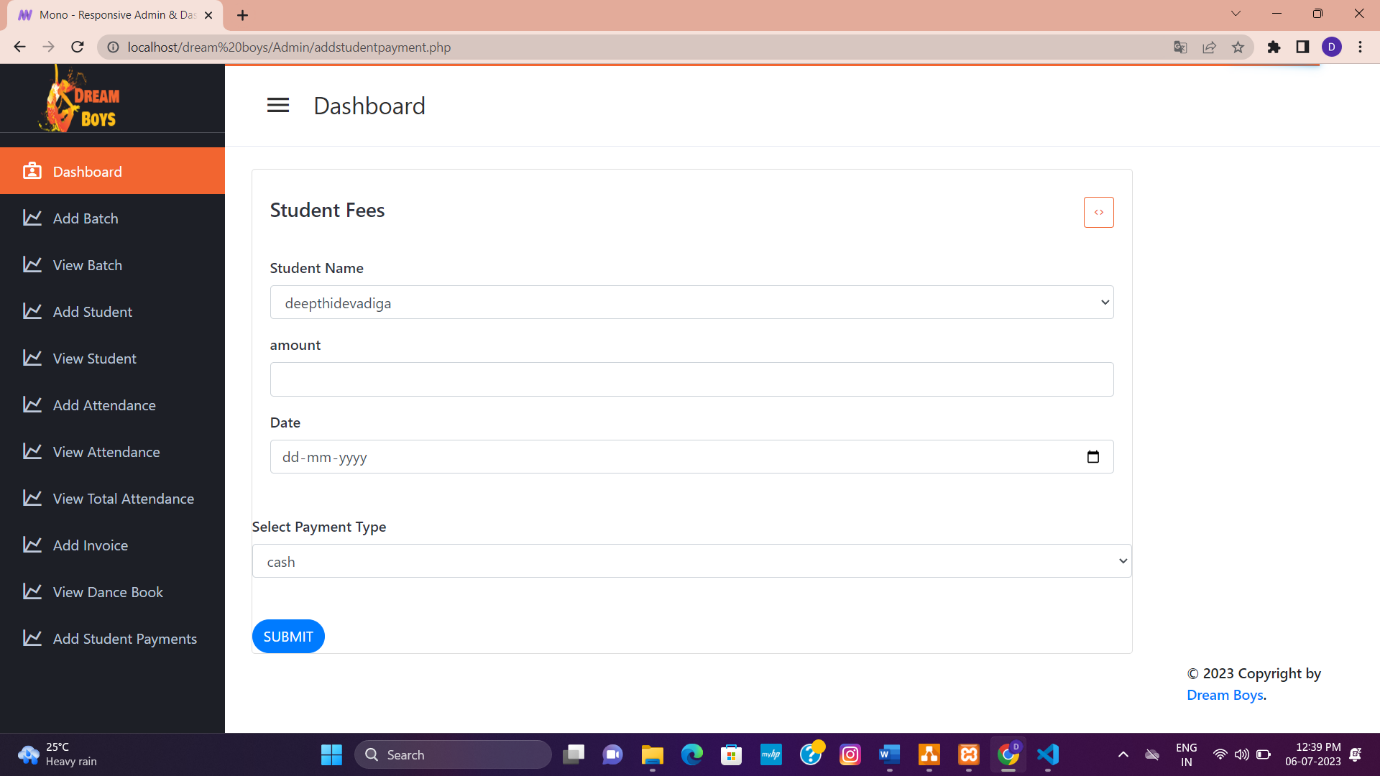
**7.2 Main screen/Home page**



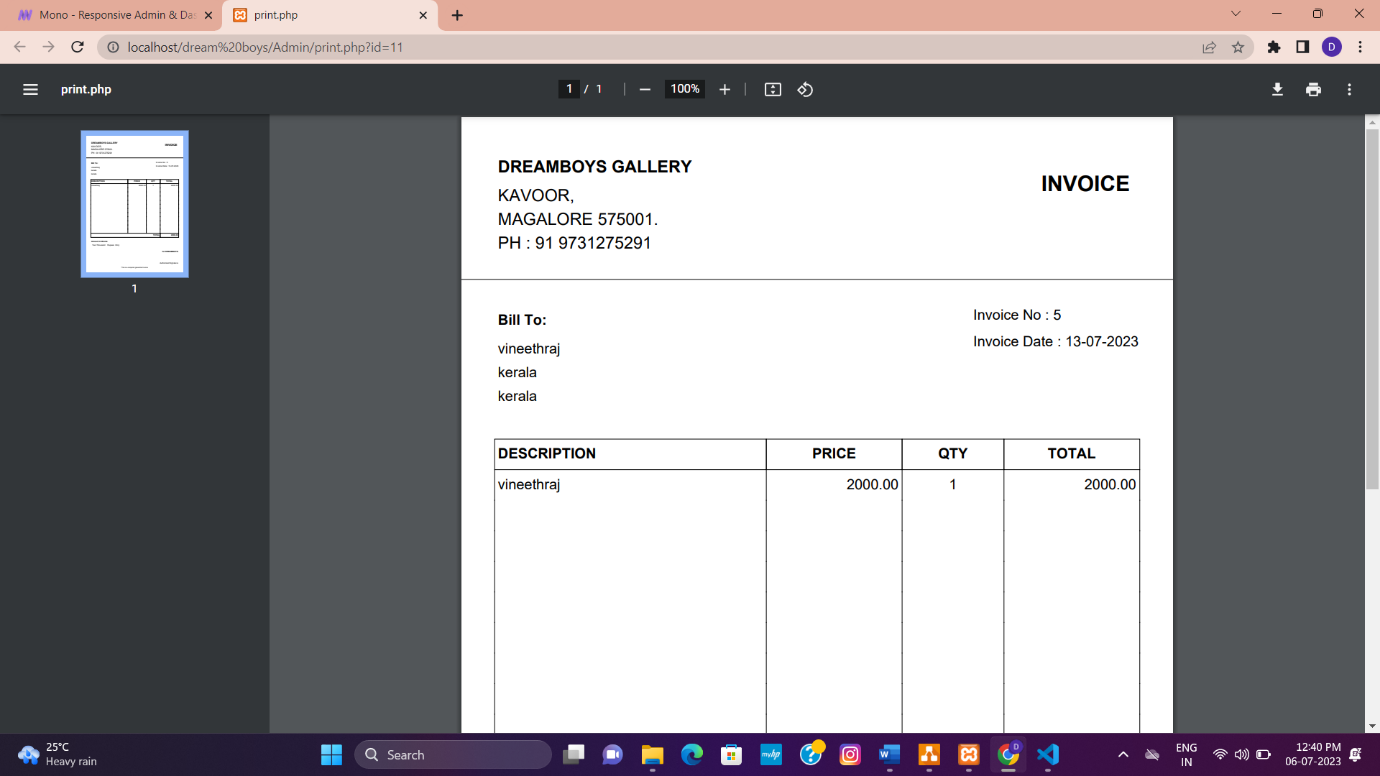
**7.3 Bookings**:



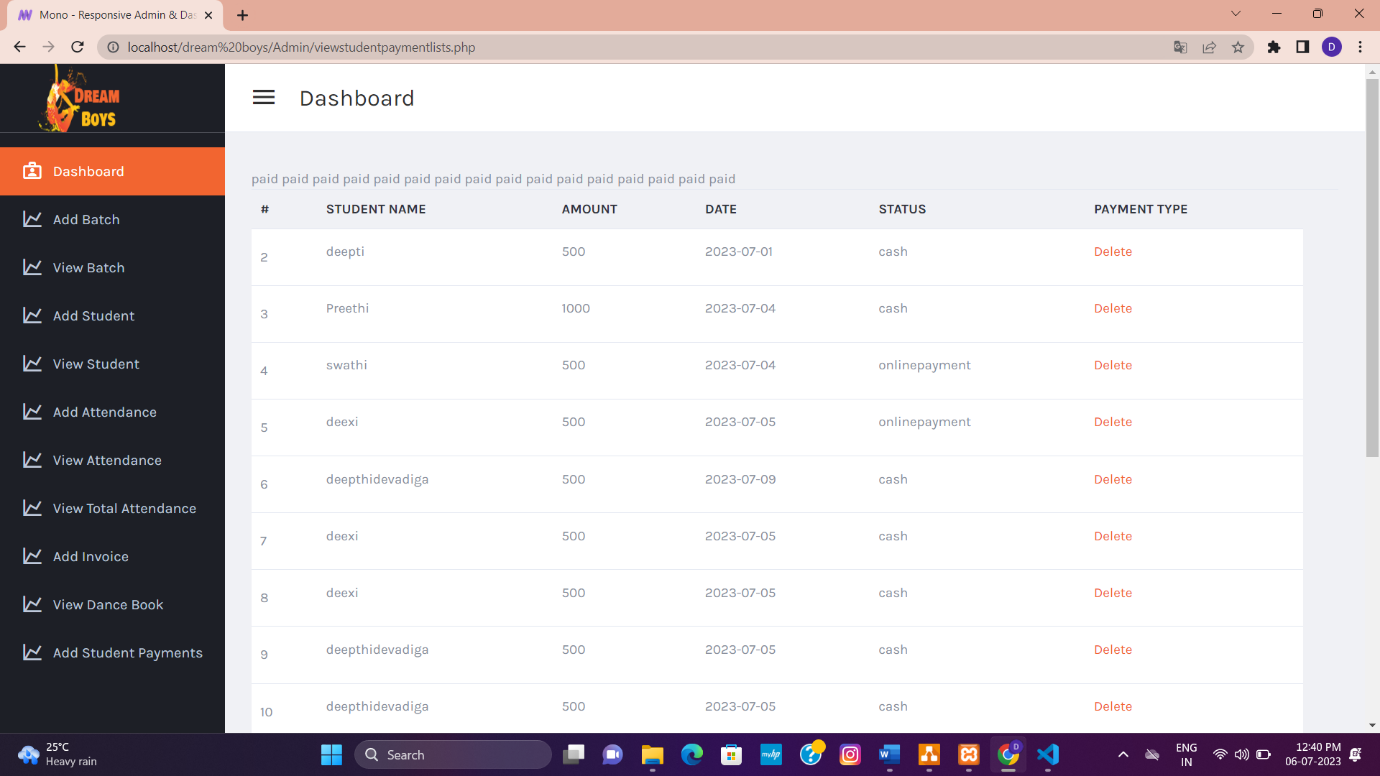
**7.4 Fees payment**:



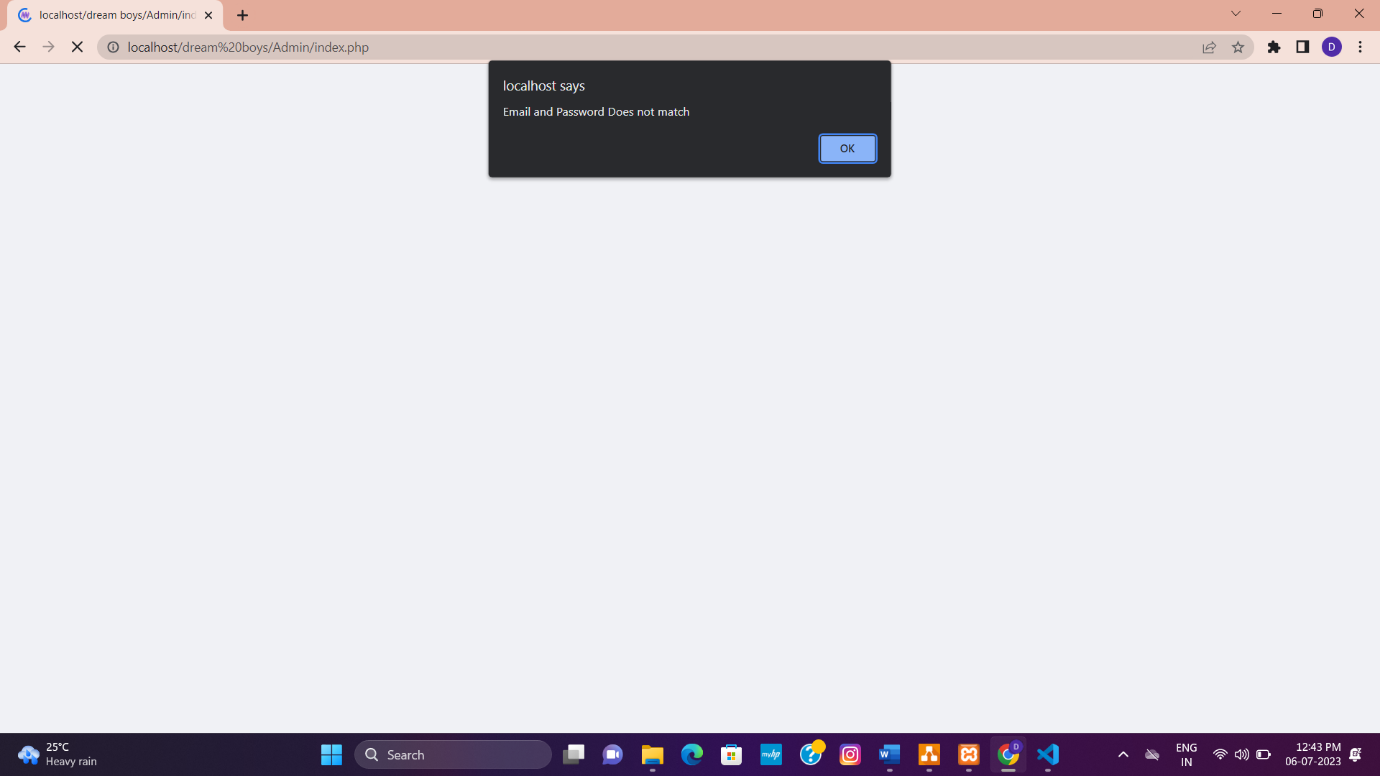
**7.5 Invoice:**



7.6 View payment:



7.7 Error message:



**8. TESTING**

**8.1 Introduction**

Software testing is an investigation conducted to provide stack holders with information about the quality of the product or service under test. Testing has been defined as the process of analysing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. Software testing is the process used to assess the quality of computer software.

It involves operation of a system or application under controlled conditions and evaluating the results. The controlled conditions should include both normal and abnormal conditions. Testing should intentionally attempt to make things go wrong to determine if things happen when they should. It is oriented to ‘detection’.

**Software testing has three main purposes**:

* The verification process confirms that the software meet its technical specifications. A “specification” is a description of a function in terms of a measurable output value given a specific input value under specific preconditions.
* The validation process confirms that the software meets the business requirements.
* A defect is a variance between the expected and actual result. The defect’s ultimate source may be traced to a fault introduced in the specification, design, or development phases. Not all the defects will necessarily result in failures.

**There are two types of software testing:**

* Black box testing-internal system design is not considered in this type of testing. Test are based on requirements and functionality.
* White box testing-this testing is based on knowledge of thaw internal logic of an application’s code. Also known as glass box testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths and conditions.

A test case is a software testing document, which consists of event, action, input, output, expected result and actual result. Clinically defined a test case is an input and an expected result. This can be pragmatic as ‘for condition x your derived result is y’; where as other test cases described in more detail the input scenario and what results might be expected. It can occasionally be a series of steps but one with expected results or expected outcome. A test case should also contain a place for the actual result.

White box testing is applicable at the unit, integration and system levels of the software

testing process.

**Testing objectives**

* Finding defects which may get created by the programmer while developing the software
* Gaining confidence in and providing information about the level of quality
* To prevent defects
* To make sure that the end results meets the business and user requirements.
* To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specification.

**Testing methods**

System testing is the stage of implementation. This is to check whether system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. The candidate system is subject to a variety of tests: online response, volume, stress, recovery, security and usability tests. A series of tests are performed for the proposed system is ready for user acceptance testing.

**8.2 Testing steps**

**Unit testing**

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

**Integration testing**

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface.

The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

**Validation**

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

* The function or performance characteristics confirm to specification and are accepted.
* A deviation from specification is uncovered and a deficiency lists is created.
* Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

**Output testing**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hard copy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

**User acceptance testing**

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

* This is done in regard to the following point:
* Input screen design.
* Output screen design.
* Online message should be guide to the user.
* Format of reports and other outputs.

**8.3 Test cases**

Admin Login form

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **Test Condition** | **Expected Result** | **Result** |
| 1 | If admin clicks on login button without entering username and password. | Please fill out this field | Successful |
| 2 | If username is blank but password is entered. | Please fill out this field | Successful |
| 3 | If password is blank but username is entered. | Please fill out this field | Successful |
| 4 | If the username or password is incorrect. | Invalid username and password | Successful |
| 5 | If the valid username and valid password is entered. | System displays admin’s main home page. | Successful |

User registration

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **Test Condition** | **Expected Result** | **Result** |
| 1 | If user name is empty | Please fill out this filed | Successful |
| 2 | If user password is empty | Please fill out this filed | Successful |
| 3 | If user’s full name is empty | Please fill out this field | Successful |
| 4 | If email is not in proper format | Please include an ‘@’ in email address | Successful |
| 5 | If address is empty | Please fill out this field | Successful |
| 6 | If contact number is not in proper format | Please match the requested | Successful |
| 7 | If no image is selected | Please select a file | Successful |
| 8 | If contact is empty | Please fill out this field | Successful |
| 9 | If email is empty | Please fill out this field | Successful |

**8.4 System Testing:**

System Testing is the testing of a complete and fully integrated software product. Usually, software is only one element of a larger computer-based system. Ultimately, software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

System testing tables:

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Test condition** | **Test report** |
| 1 | System lading | Successful |
| 2 | System run procedure | Successful |
| 3 | File I/O operation | Successful |
| 4 | Database communication | Successful |
| 5 | Server/client interaction | Successful |
| 6 | Memory usage | Normal |
| 7 | System processor usage | Normal |
| 8 | Authentication/Authorization | Successful |

**9.CONCLUSION & FUTURE ENHANCEMENT**

My project Dream Boys gallery guarantees to simplify the process of registration and booking. It promises great benefits to the users as it allows easy interactions, provides a user friendly interface, reduces wastage of time. My system is extremely reliable and completely integrated so that we always know what is in our hand and what is committed to the customers.

This project helped me to improve my knowledge in PHP. The importance of a good software design was learned during the project. I hope people find this software informative, useful, interesting and challenging. Developing this software was a good experience.

In present days, the software has to be simple, user friendly and should be able to incorporate any further improvements. This software facilitates these features. The project was developed after extensive research conducted in a real life scenario which would exactly depict the present working condition.

**Future Enhancement :**

The application developed is designed is such a way that any future enhancement can be done with easy. New module can be added to the portal with less effort. Future portal will be facilitating customer with online transaction through credit and services.

**10.BIBILOGRAPHY**

1. http://searchsoftwarequality.techtarget.com/definition/software-requirementsspecification.

2. [www.w3schools.com/php/](http://www.w3schools.com/php/)

3. Pankaj Jalote, An integrated approach to software engineering,ed. 3 , Publication: Narosa Publishing House.

4. Nixon (Robin), Learning PHP MySQL and JavaScript, Publication: Shroff publishers&Distributors Pvt Pankaj Jalote Ltd.

5. PHP Developer’s Guide By Greg Buezek.

6. MYSQL A Complete Reference By Alex Leon, Mathew Leon