**Art Gallery Management System**

**Abstract:**

The Art Gallery Management System with Simple CMS website using PHP/MySQL is a simple project that will help a certain gallery to manage their paintings or sculptures in their business. The main purpose of this project is to help the art gallery business market their arts online using their own website. This system includes upcoming events advertisement of an artist that rented the gallery to hep also their client, and this feature of the system may also use by the art gallery business to announce their own events. The system will store the information of each painting or sculpture including the artist who created and editable content. Each art that is stored in the system, the management can upload multiple views of a certain artwork depending to the views they wanted to publish to their website.

**1.INTRODUCTION:**

An online art gallery stores information about the “Artists”, their names, birthplace, age and style of art about “art work”, artist , unique title , type of art and prices of art works . The pieces of artwork is classified into various kinds like medieval art , nature , animal, landscapes , etc . The gallery keeps information about customers as well as the artists as their unique name, total amount spent on gallery and the likings of the customer. The main aim of the project is the management of the database of ART GALLERY. This project is insight into the design and implementation of a Art Gallery Management. This is done by creating a database of the available details in Art Gallery. The primary aim of this Art Gallery Management System is to improve accuracy and enhance safety and efficiency of tracking and keeping details of art and paintings in art gallery. I have developed this software for ensuring effective policing by providing statistics of the Members.

The MYSQL database is used as a platform along with PHP and WAMP Server support. Application and the GUI are developed in HTML5, CSS3 using PHP and WAMP Server.

Overall this Art Gallery Management System is used to manage most art related activities like exhibitions, gallery management, art stocks etc. in gallery.

* 1. **SYSTEM SPECIFICATION:**

**1.2.1 HARDWARE SPECIFICATION**

* System : Inteli3
* Hard Disk : 500 GB.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 4 GB.
* Keyboard : 101 Keyboard

**1.2.2 SOFTWARE SPECIFICATION**:

* Operating System : Windows 10
* Front End : PHP version 7
* Back End : MySQL version 7
* Server : XAMPP server

**CHAPTER-2**

**SYSTEM STUDY**

System study is used in every field where there is a work of developing something. Analysis can also be defined as a series of components that perform organic function together. In computer world, in this stage a statement of the problem is formulated and a model is built by the analyst in encouraging real-world situation. This phase show the important properties associated with the situation. Actually, the analysis model is a concise, precise abstraction and agreement on how the desired system must be developed.

**2.1 Existing System:**

Art gallery is the place where the arts or art craft of the artists can be displayed for the visitors view. People also comment on the arts that artist’s exhibit so that there will be some scope of improvement of the arts that is displayed. Sometimes it will be difficult to maintain the details of the artist and their paintings details through the pen paper method. This is main drawbacks in our existing system. So our proposed system is that allows the art gallery owners to maintain the details of the artists and the details of their paintings with great ease. This will be one of the useful application for the owners of the art gallery which will reduce most of their work.

**2.2 Proposed System :**

ONLINE ART GALLERY is a website and it is very helpful for the art lovers and others who want to buy an art work. This website helps the end users to search their arts and paintings and they can enquiry for the selected pieces. It facilitates the artists to create their profile on this system and keep track of their art works and also upload their new art forms in the digital form . They will be able to see which of their paintings have been sold and number of likes each of their paintings have got . The paintings would be categorized into different types such as nature, animal, landscapes, etc. Customers would be able to access any particular category according to their interests or any particular artist of their liking after they create an account on this website. They can like any painting and if they are interested in buying it then they can add it to their cart and proceed for checkout . User and artists can also edit their profile info and photos.

**2.3 Modules:**

* + - Admin Module
    - Customer Module

**Admin Module:**

**Login Page**

The page where the system users will submit their system credentials to access and manage the data of the Art Gallery Management System.

**Home Page**

The page where the system user will be redirected by default after logging into the system.

**Artist List**

The page where all the artists who created the artworks in the gallery are listed and managed.

**Arts**

The page in the system that lists all the artworks in the gallery and where can system users managed the information of the artworks.

**Events Page**

The page where all scheduled events of the gallery are listed and managed.

**Enquiry Page**

The page where all enquiry requests of customers are listed and managed.

**Users Page**

The page where can system users managed the list of users in the system.

**Customer Module:**

**Home Page**

The page where the customers or visitors will be redirected by default when browsing the art gallery's website. This the where the upcoming events list of the art gallery is listed.

**Explore Page**

The page where all artworks in the art gallery are listed including the artworks that are for sale. This page displays the content of each artwork that the management published.

**About Page**

The page where the Art Gallery about content displays.

**Enquiry Module:**

Here user can select the art and enquiry about that art.

**3. SOFTWARE SPECIFICATION**

**SOFTWARE DESCRIPTION**

**XAMPP:**

XAMPP is a [free and open source](https://en.wikipedia.org/wiki/Free_software) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [web server](https://en.wikipedia.org/wiki/Web_server) [solution stack](https://en.wikipedia.org/wiki/Solution_stack) package developed by Apache Friends, consisting mainly of the [Apache HTTP Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server), [Maria DB](https://en.wikipedia.org/wiki/MariaDB) [database](https://en.wikipedia.org/wiki/Database), and [interpreters](https://en.wikipedia.org/wiki/Interpreter_%28computing%29) for scripts written in the [PHP](https://en.wikipedia.org/wiki/PHP) and [Perl](https://en.wikipedia.org/wiki/Perl) [programming languages](https://en.wikipedia.org/wiki/Programming_language).

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.

XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet.

**CROSS PLATFORM**

Cross-platform software is a type of software application that which works on multiple operating systems or devices, which are often referred to as platforms. A platform means an operating system such as Windows, Mac OS, Android or iOS. When a software application works on more than one platform, the user can utilize the software on a wider choice of devices and computers.

**BENEFITS OF CROSS PLATFORM**

The benefit of a cross-platform software app or program is that you can use the same program whether you’re on a Windows PC or whether you’re logging in from your laptop or smartphone. The Microsoft Office suite of applications, which includes Word, Excel, and PowerPoint, are available on Windows, Mac OS, iOS (iPhone/iPad) and Android. While there are differences based on how the platforms work, you’ll have a similar experience within the application between all of your devices.

Having a similar experience across any platform means there’s a much smaller learning curve, if one even exists at all, so you’ll be more productive and be able to use a software product you’re familiar with regardless of the operating system or device you choose. In addition, your files can be moved much more easily between your devices so you can use the software with whatever device you have with you at the time. And there’s a way to keep all of your work in sync across all of your devices, by using the cloud.

**EXAMPLES OF CROSS PLATFORM**

# Unity3D

First, let’s talk about Unity3D. I think the game engine should be preferred by people who want to write mobile games.  
You can develop games on 17 platforms using multiple languages, including Linux. Of course, ios, android and windows phone is also the most ideal game engine to develop games.

You can develop your application using C #, JS, C ++.

Link to: [https://unity3d.com](https://unity3d.com/)

# Xamarin

Xamarin Some time ago, it was purchased by Microsoft and is a perfect fit for developers using C #.

Because it is a C # language, it has a lot of documentation, and because of Microsoft support, Xamarin is the choice for C # developers.

In addition, you can do everything you can do in Objective-C, Swift and Javada with the Xamarian library.

Link to: [https://xamarin.com](https://xamarin.com/)

# React Native

React Native is an open-source JavaScript library developed by the new generation of React — Facebook, which was open to Github in 2013. Native application creation means writing applications only for a specific operating system. React Native helps developers reuse their code over the web and on the mobile. Developers will not have to create the same app from scratch for iOS and Android. They will be able to reuse the code in each operating system. The great thing about React Native is that there is little difference between a finished application in Objective-C or Java and an application built using React Native. Android and iOS code development environments are very different. So it takes time to remove the application to two different platforms. However, with React Native, only one developer can write on different mobile operating systems.

**APACHE:**

The Apache HTTP Server, colloquially called Apache is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [web server](https://en.wikipedia.org/wiki/Web_server) software, released under the terms of [Apache License](https://en.wikipedia.org/wiki/Apache_License) 2.0. Apache is developed and maintained by an open community of developers under the auspices of the [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation).

The vast majority of Apache HTTP Server instances run on a [Linux distribution](https://en.wikipedia.org/wiki/Linux_distribution), but current versions also run on [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [OpenVMS](https://en.wikipedia.org/wiki/OpenVMS),  and a wide variety of [Unix-like](https://en.wikipedia.org/wiki/Unix-like) systems. Past versions also ran on [NetWare](https://en.wikipedia.org/wiki/NetWare), [OS/2](https://en.wikipedia.org/wiki/OS/2) and other operating systems,  including ports to mainframes.

Originally based on the [NCSA HTTPd](https://en.wikipedia.org/wiki/NCSA_HTTPd) server, development of Apache began in early 1995 after work on the NCSA code stalled. Apache played a key role in the initial growth of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), quickly overtaking NCSA HTTPd as the dominant [HTTP](https://en.wikipedia.org/wiki/HTTP) server. In 2009, it became the first web server software to serve more than 100 million [websites](https://en.wikipedia.org/wiki/Website). As of January 2021, [Netcraft](https://en.wikipedia.org/wiki/Netcraft) estimated that Apache served 24.63% of the million busiest websites, while [Nginx](https://en.wikipedia.org/wiki/Nginx) served 23.21% and Microsoft is in third place at 6.85% (for some of Netcraft's other stats Nginx is ahead of Apache), while according to W3Techs, Apache is ranked first at 35.0% and Nginx second at 33.0% and Cloudflare Server third at 17.3%.

**LANGUAGE SPECIFICATION**

**PHP**

**INTRODUCTION OF PHP**

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
* PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
* PHP is forgiving: PHP language tries to be as forgiving as possible.
* PHP Syntax is same as C language.

**What is a PHP File?**

* PHP files can contain text, HTML, CSS, JavaScript, and PHP code.
* PHP code are executed on the server, and the result is returned to the browser as plain HTML.
* PHP files have extension ".php".

**What Can PHP Do?**

* PHP can generate dynamic page content and it can create, open, read, write, delete, and close files on the server and it can collect form data.
* PHP can send and receive cookie and it can add, delete, modify data in your database and it can be used to control user-access and encrypt data.

**Why PHP?**

* PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.).
* PHP is compatible with almost all servers used today (Apache, IIS, etc.).
* PHP supports a wide range of databases.
* PHP is free.
* PHP is easy to learn and runs efficiently on the server side.

## What is Database?

* A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.
* Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.
* So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

**MySQL Database**

* MySQL is released under an open-source license. So you have nothing to pay to use it. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table.
* The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).MySQL is customizable.
* The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments

**TABLE CREATION**

* Name of the table
* Names of fields
* Definitions for each field
* Field Attribute **NOT NULL** is being used because we do not want this field to be NULL. So if user will try to create a record with NULL value, then MySQL will raise an error.
* Field Attribute **AUTO\_INCREMENT** tells MySQL to go ahead and add the next available number to the id field.
* Keyword **PRIMARY KEY** is used to define a column as primary key. You can use multiple columns separated by comma to define a primary key.

## ADMINISTRATIVE MYSQL COMMAND

* **USE DATABASE NAME**: This will be used to select a particular database in MySQL work area.
* **SHOW DATABASES:** Lists the databases that are accessible by the MySQL DBMS.
* **SHOW TABLES:** Shows the tables in the database once a database has been selected with the use command.
* **SHOW COLUMNS FROM Table name:** Shows the attributes, types of attributes, key information, whether NULL is permitted, defaults, and other information for a table.
* **SHOW INDEX FROM Table name:** Presents the details of all indexes on the table, including the PRIMARY KEY

## CREATING TABLES USING PHP SCRIPT:

Create new table in any existing database you would need to use PHP function **mysqli\_query()**.

## Dropping Tables Using PHP Script:

Drop an existing table in any database, you would need to use PHP function **mysqli\_query()**.

## INSERTING DATA USING PHP SCRIPT:

**CREATE**

Create table statement is used to create a table in MySQL.

**SELECT**

The SELECT statement is used to select data from one or more tables.

**UPDATE**

The UPDATE statement is used to update existing records in a table:

## DELETE

The DELETE statement is used to delete records from a table:

**DATABASE DESIGN:**

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage.

They are structured and put together to design the data storage and retrieval system. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently.

The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed.

Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates.

**INPUT DESIGN**

The Input design is the main feature of the system. Input design determines the format and validations criteria for data entering the system. Inputs originate with end-users; human factors play a significant role in input design. The input design is designed to control the input, to avoid delay, errors in data, to avoid extra steps, to keep the process simple. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

The following are the general principles, which are considered in designing inputs are,

* + - Enter only variable data
    - Do not input data that can be calculated
    - List of values
    - Sequence entry

**OUTPUT DESIGN**

Designing the output is more important than working up with few layout charts and reports. The outputs are designed based on the issue encountered. It will also take care of who will receive the output, what for it is produced how much details are needed, when it is needed and by what method.

The outputs designed in this system are easy to use and useful for their jobs. The outputs are simple to read interpret. The outputs obtained from this system are designed by using a few guidelines, which are given below. The information should be clear and accurate, yet concise and restricted to relevant data. Reports should have titles, the data and descriptive heading for columns of data, numbered pages and so on.

**4. SYSTEM TESTING**

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client/server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

Testing issues

* Client GUI considerations
* Target environment and platform diversity considerations
* Distributed database considerations
* Distributed processing considerations

**TYPES OF TESTING**

1. Unit Testing

2. Integration Testing

3. Validation Testing

4. User acceptance Testing

5. System Testing

**Unit Testing**

All modules were tested and individually as soon as they were completed were checked for their correct functionality. Unit testing is carried out by verify and recover errors within the boundary of the smallest unit or a module. In this testing step, each module was found to be working satisfactory per the expected output of the module. In the package development, each module is tested separately after it has been completed and checked with valid data.

**Integration Testing**

The entire project was split into small programs; each of these single programs gives a frame as an output. These programs were tested individually; at last all these programs where combined together by creating another program where all these constructions were used. It gives a lot of problem by not functioning in an integrated manner.

The user interface testing is important since the user has to declare that the arrangements made in the frames are convenient and it is satisfied. When the frames are the test, the end user gave suggestion. Since they were much exposed to do the work manually.

**Validation Testing**

At the culmination of the black box testing software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of test i.e., validation succeeds when the software functions in a manner that can be reasonably accepted by the customer.

**User Acceptance Testing**

User acceptance testing of the system is the key factor the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of development and making change whenever required. This is done with regard to the input screen design and output screen design.

**System Testing**

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

**5. SYTEM IMPLEMENTATION**

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

The stage consists of

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

**Implementation Procedures**

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

The final report to the implementation phase includes procedural flowcharts, record layouts, report layouts, and a workable plan for implementing the candidate system design into an operational one. Conversion is one aspect of implementation.

**System Maintenance**

Maintenance is actually the implementation of the review plan. As important as it is, many programmers and analysts are to perform or identify themselves with the maintenance effort. There are psychological, personality and professional reasons for this. Analysts and programmers spend far more time maintaining programs than they do writing them. Maintenance accounts for 50-80 percent of total system development.

Maintenance is expensive. One way to reduce the maintenance costs are through maintenance management and software modification audits.

* Maintenance is not as rewarding as exciting as developing systems. It is perceived as requiring neither skill not experience.
* Users are not fully cognizant of the maintenance problem or its high cost.
* Few tools and techniques are available for maintenance.
* A good test plan is lacking.
* Standards, procedures, and guidelines are poorly defined and enforced.
* Programs are often maintained without care for structure and documentation.
* There are minimal standards for maintenance.
* Programmers expect that they will not be in their current commitment by time their programs go into the maintenance cycle.

**6. SYSTEM DESIGN**

System design is "the process of studying a procedure or business in order to identify its goals, purposes and create systems and procedures that will achieve them in an efficient way". Another view sees system analysis as a problem-solving technique that breaks down a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose.

The field of system analysis relates closely to requirements analysis or to operations research. It is also "an explicit formal inquiry carried out to help a decision maker identify a better course of action and make a better decision than they might otherwise have made."

* **DESIGN NOTATION**

Design notations are used when planning and should be able to communicate the purpose of a program without the need for formal code. Commonly used design notations are:

* DFD
* ERD
* **DFD (DATA FLOW DIAGRAM):**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an [information system](https://en.wikipedia.org/wiki/Information_system), modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the [visualization](https://en.wikipedia.org/wiki/Data_visualization) of [data processing](https://en.wikipedia.org/wiki/Data_processing) (structured design). A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel unlike a [flowchart](https://en.wikipedia.org/wiki/Flowchart) which also shows this information.

Data flow diagrams were popularized in the late 1970s, arising from the book Structured Design, by computing pioneers Ed Yourdon and Larry Constantine. They based it on the “data flow graph” computation models by David Martin and Gerald Estrin. The structured design concept took off in the software engineering field, and the DFD method took off with it. It became more popular in business circles, as it was applied to business analysis, than in academic circles

.

**DFD SYMBOLS:**

Process that transforms data flow w

Source or Destination of data

Data flow

Data store

* **ENTITY RELATIONSHIP DIAGRAM**

The relation upon the system is structure through a conceptual ER-Diagram, which not only specifies the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue. The Entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the data modeling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.

The set of primary components that are identified by the ERD are

* + Data object
  + Relationships
  + Attributes
  + Various types of indicators

The primary purpose of the ERD is to represent data objects and their relationships.

**ER-DIAGRAM SYMBOL**

Entity

Relationship

Flow

* **INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system. It is achieved by creating user-friendly screens for the data entry to handle large volume of data.

The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities. When the data is entered it will check for its validity. Data can be entered with the help of screens.

* **DATABASE DESIGN**

Data base is designed to manage large bodies of information. The management of data involves both the definitions of structures for the storage of information. In addition the data base system must provide for the safety of the information solved, despite system crashes or due to attempts at unauthorized access. For developing an efficient database user have to fulfill certain conditions such as controlled redundancy.

* Defining the data
* Inputting the data
* Locating the data
* Accessing the data
* Communicating the data

Revising the data

**Objectives of Database design**

For designing data base design several objectives have to be met as follows:

* Ease of use
* Control of data integrity
* Control of redundancy
* Control of security
* Data independence (logical & physical)
* Data storage protection
* System performance
* **OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

In this Online Repository System project output is to view customer details, employee list, order tracking details and attendance percentage result.

**SYSTEM FLOW DIAGRAM**

Data Flow Diagram:

Level 0:

User

Artist

Search art

Add artist details

Get response

Level 1:

User

Table\_user

Login

Add user details

Table\_Art

Table\_Enquiry

Search art works

enquiry items

View Art

Enquiry paints

Level 2:

Add artist details

Login

Table\_artist

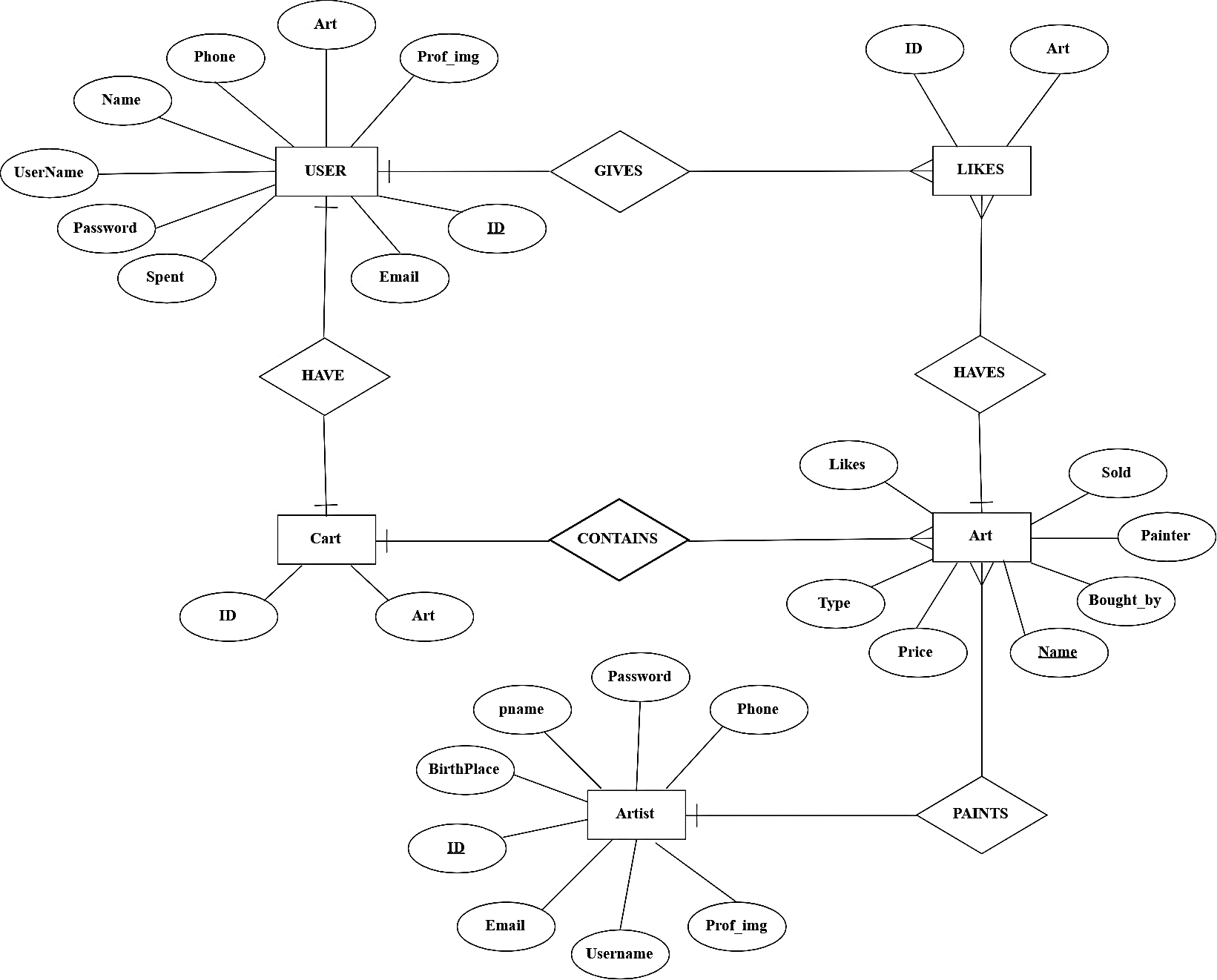
Artist

View enquiry

View enquiry Details

Table Enquiry

ER Diagram:



**7. DATABASE DESIGN:**

**Table Name: admin**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| id | int |  | unique id |
| Username | varchar | 50 | admin login name |
| Password | Text | 50 | password |

**Table Name: Art**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| Art\_id | Int | 11 | Art’s unique id |
| Description | Text | 50 | Art’s Description |
| Artist Name | Varchar | 50 | Artist Name |
| Price | int | 20 | Art Price |
| Discount | int | 20 | Price discount |
| Category | Varchar | 20 | Art Category |
|  |  |  |  |

**Table Name: Customer\_details**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| Customer\_id | Int | 11 | Customer’s unique id |
| Customer\_name | Varchar | 50 | Customer name |
| Mobile | Bigint | 50 | Customer mobile |
| Email | Varchar | 50 | Customer Email |
| Address | Text | 50 | Customer Address |
| Gender | Varchar | 20 | Customer gender |

**Table Name: Enquiry\_details**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| id | Int | 11 | Enquiry unique id |
| Customer\_name | Varchar | 50 | Customer name |
| Mobile | Bigint | 50 | Customer mobile |
| Email | Varchar | 50 | Customer Email |
| Address | Text | 50 | Delivery Address |
| Art name | Varchar | 20 | Artname |

**8. CONCLUSION:**

This project is developed by using html,css,bootstrap as front end and php, mysql as back end. By use of this project provides an efficient method for storing the art details of the artist. The target of this project is to make a web application that helps its artist to keep track of his/her art. The website has been successfully implemented and has been found to replace the existing system effectively. It provides an easier and efficient way to manage art details. Finally, this project came out successfully from those tests for the actual implementation.

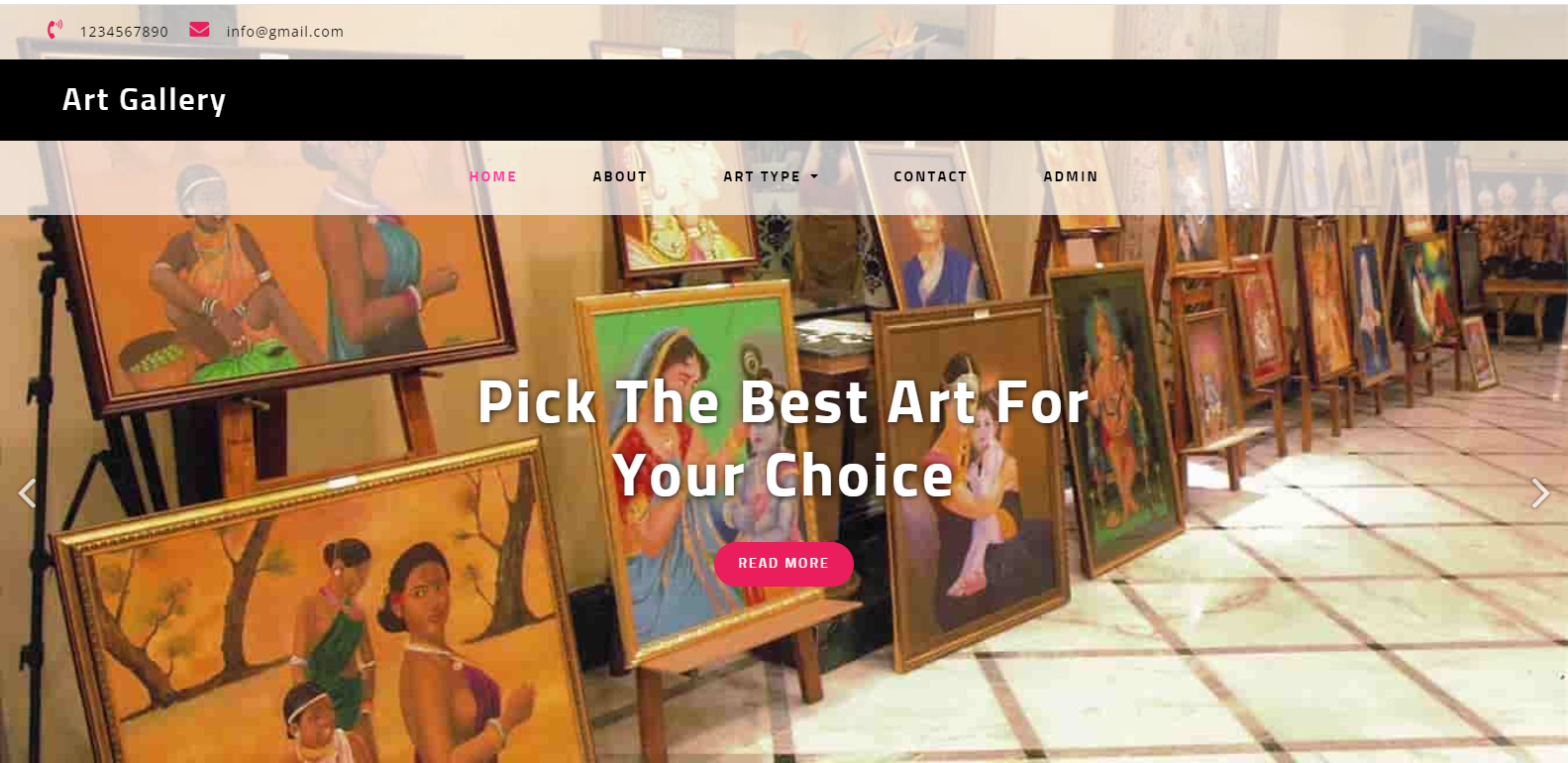
**FUTURE ENHANCEMENT:**

There is a wide scope for the future development of the software, the world of computer fields is not static it is always subject to change .The technology which is to Famous today will become outdated very next day. To keep abstract of technical improvements, the system may be refinement. So it is not concluded. Yet it will improve with further enhancements. If there is an arrival of updated software advanced than this, must be change it for further development. Further enhancement can be done with efficient manner with the disruption to the system.

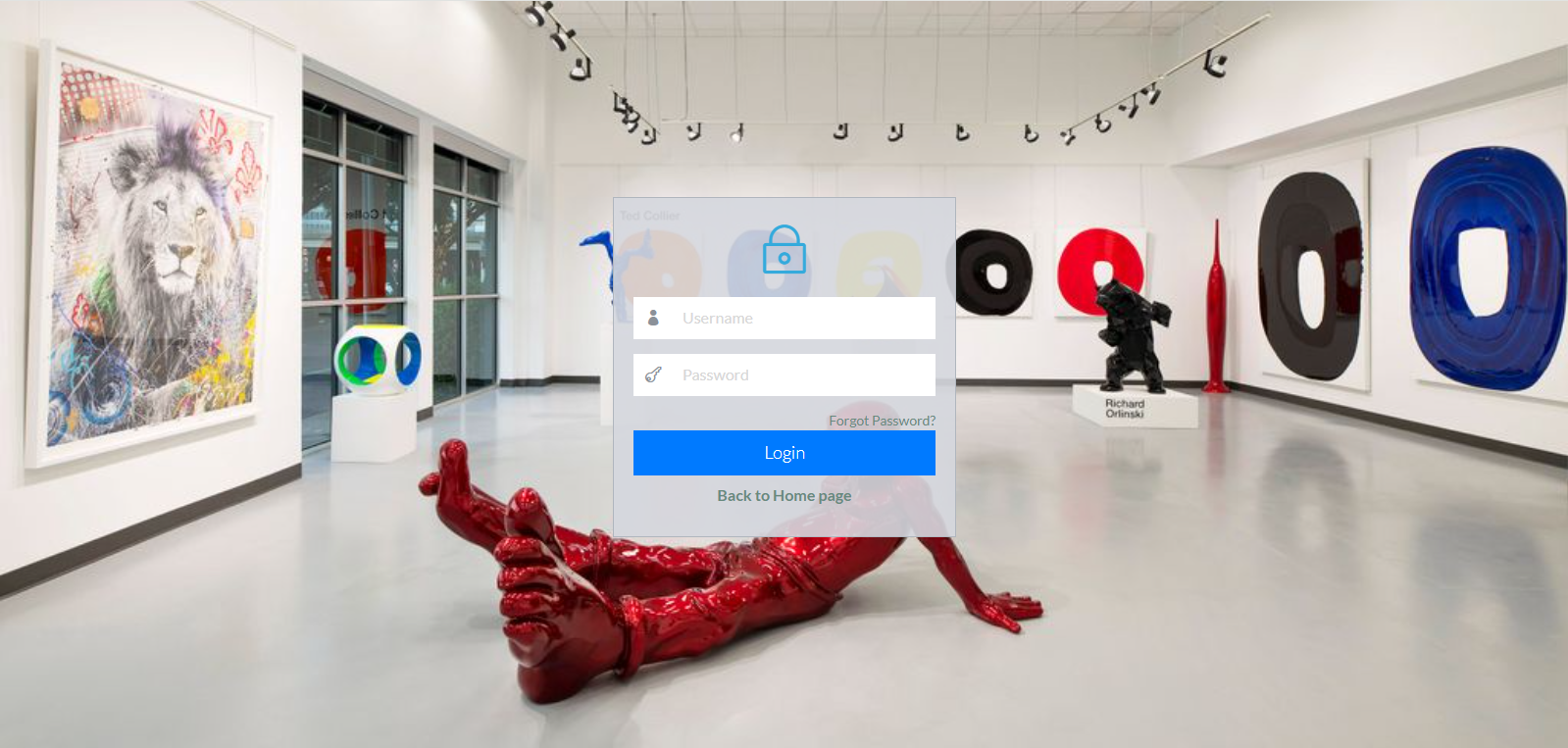
**9. Appendix**

**Screenshot:**

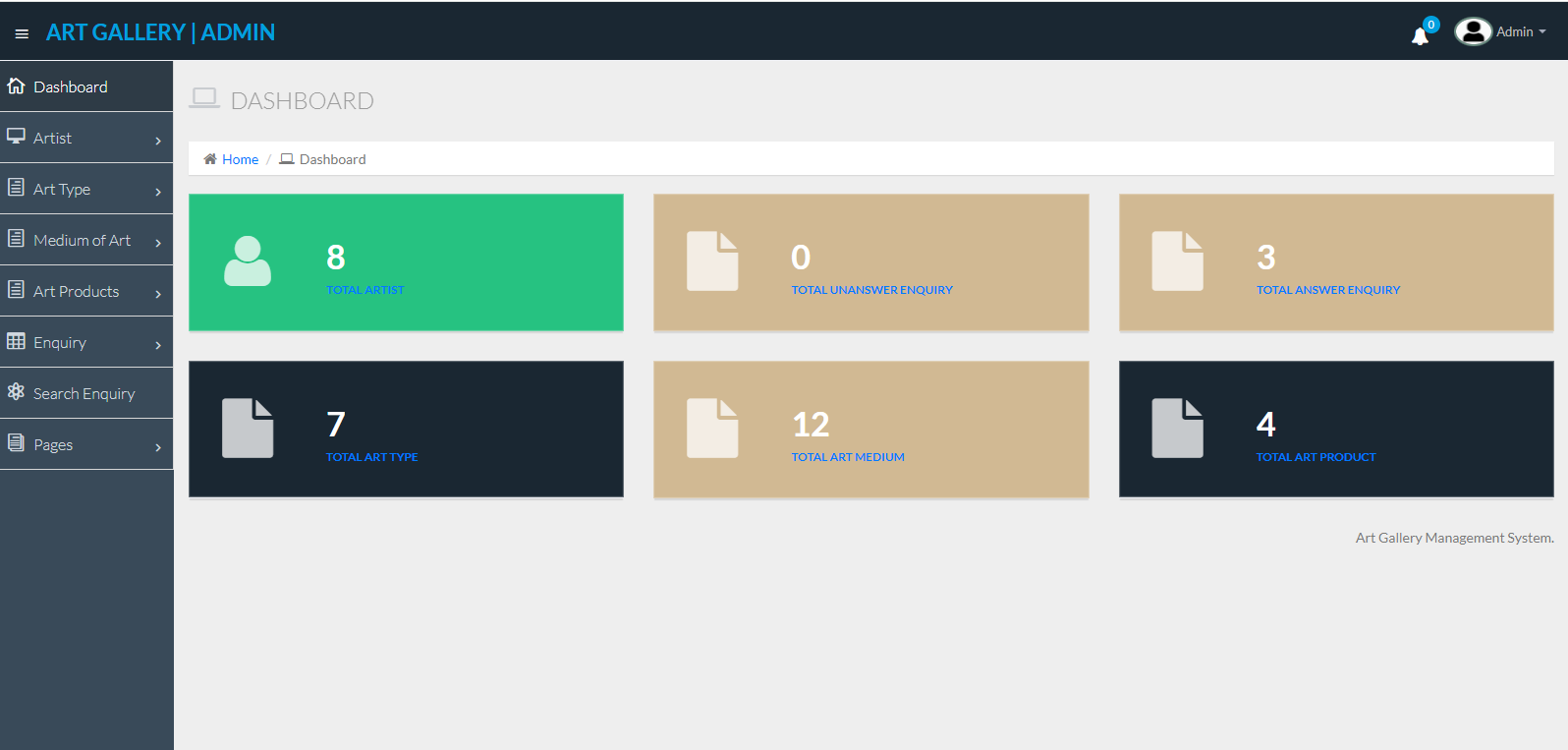
**Welcome page:**

****

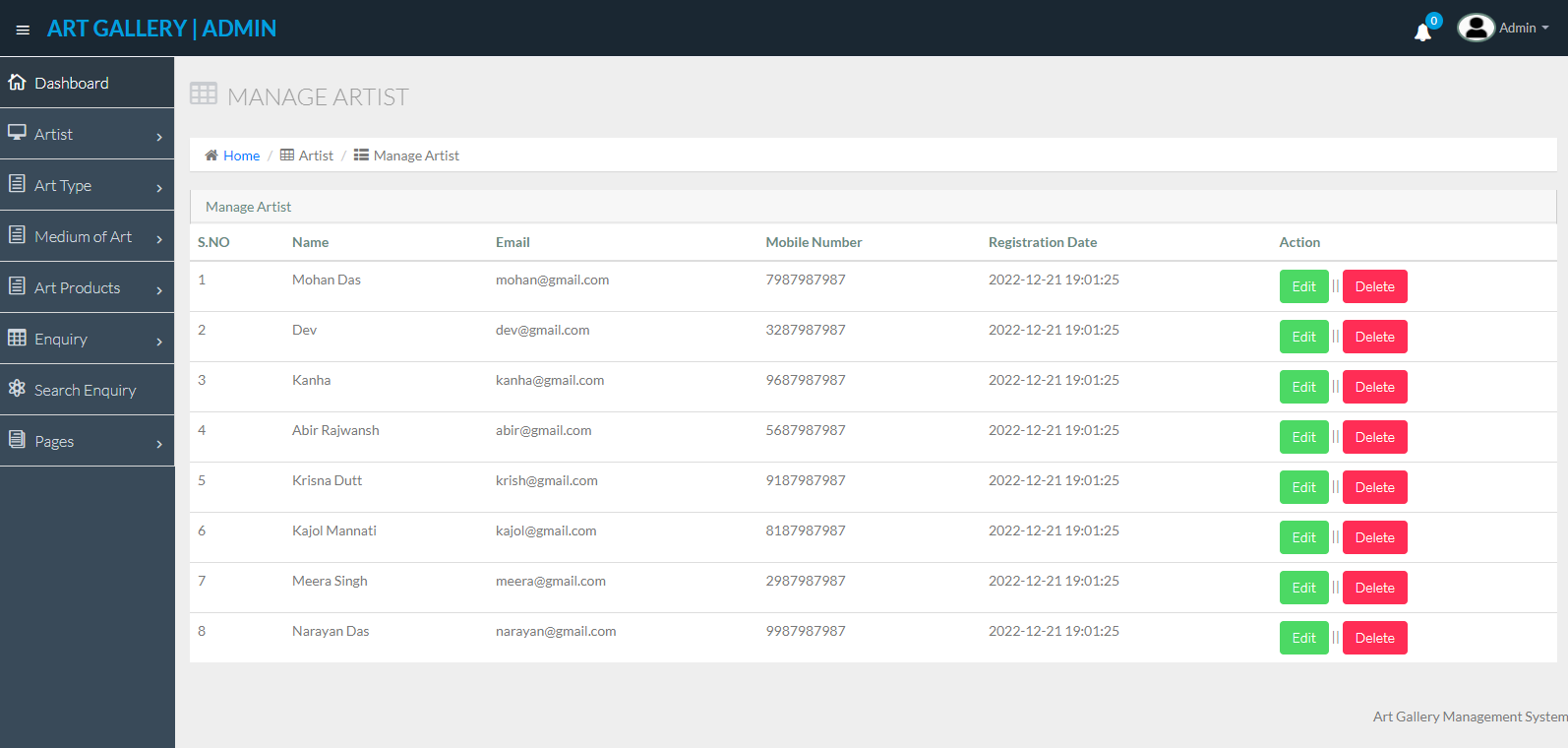
**Admin Login page:**

****

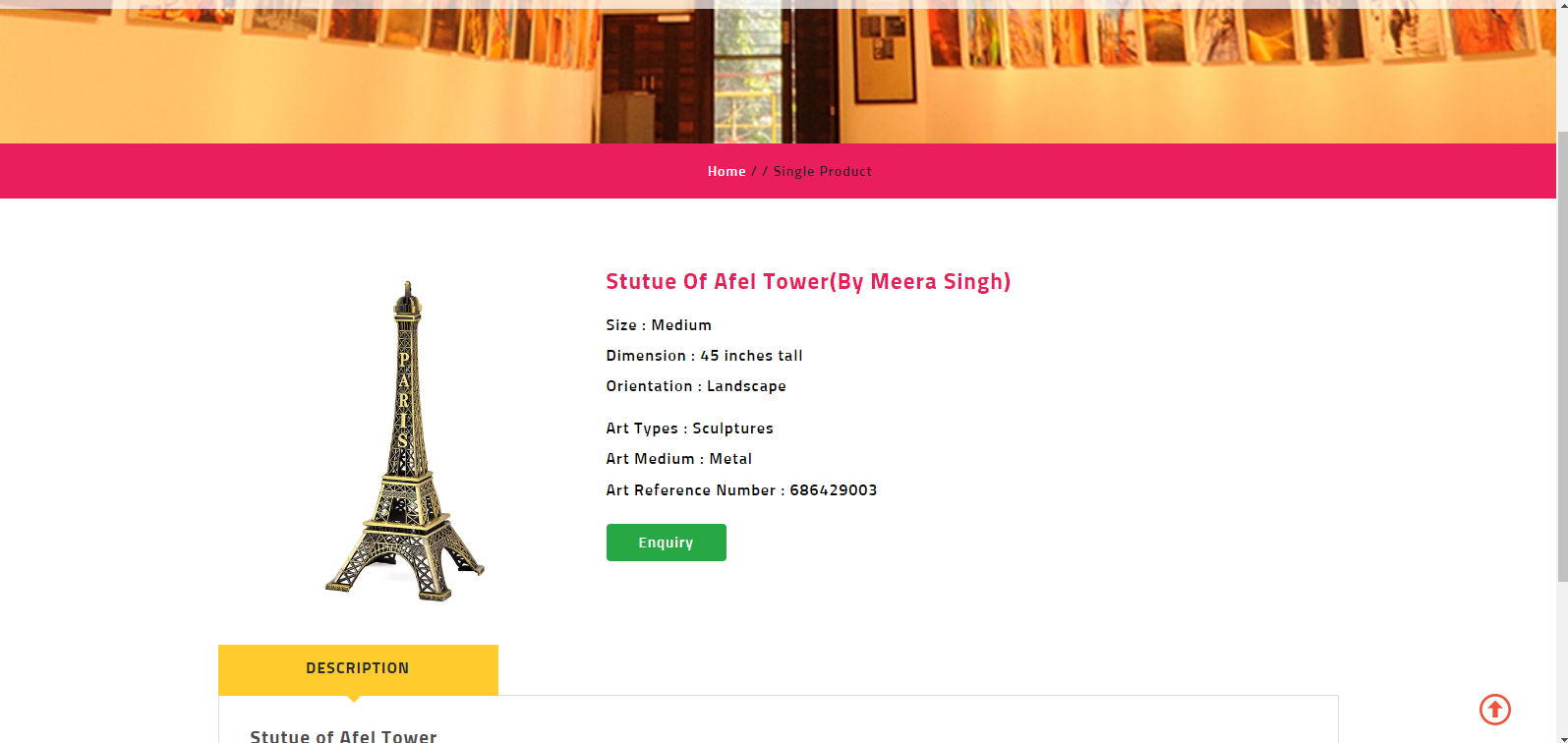
**Dashboard:**

****

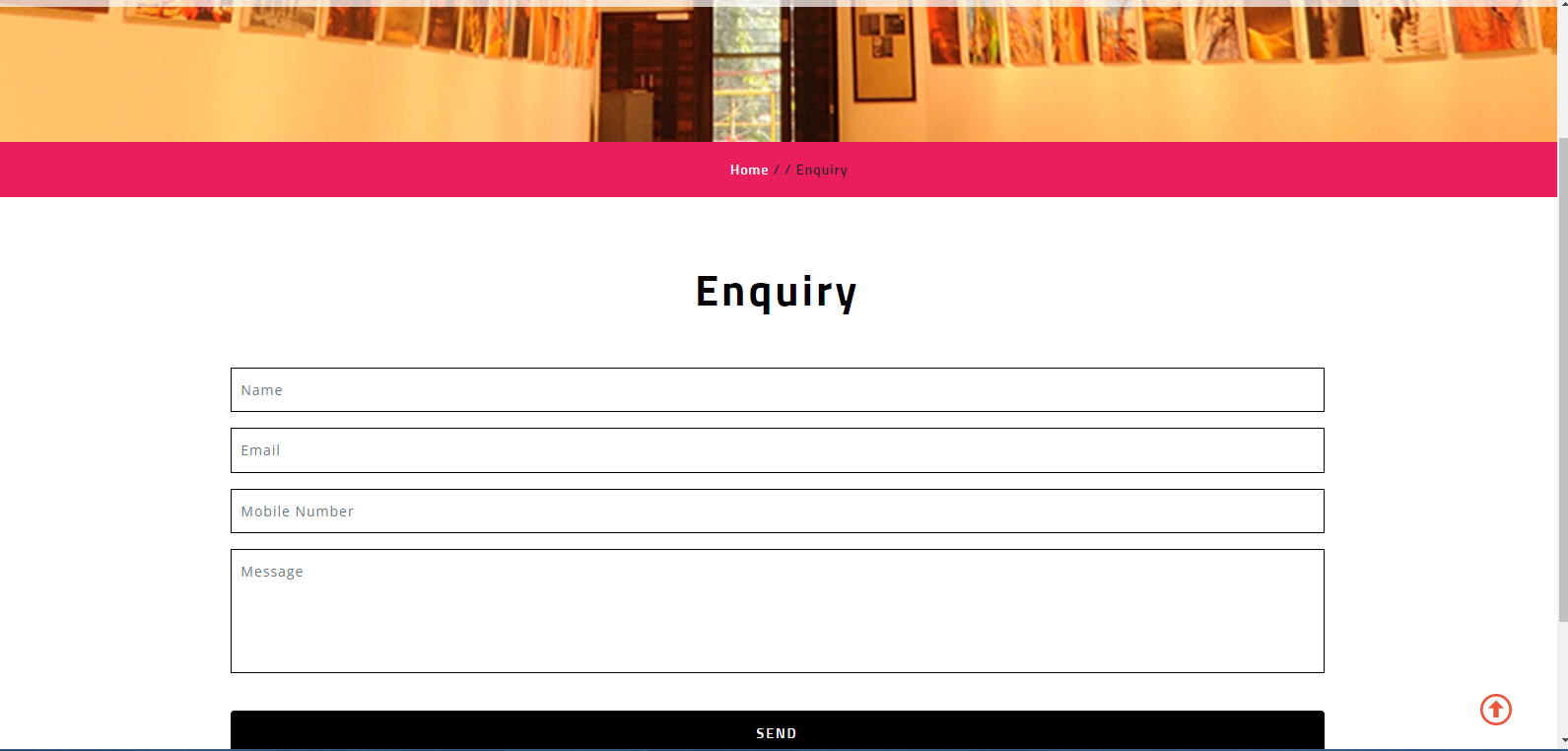
**Artist list:**

****

**Artist details:**

****

**Enquiry Form:**

****

**Samplecode:**

**Login.php:**

<!DOCTYPE html>

<html>

<head>

<title>Log In</title>

<style type="text/css">

.error{

position: absolute;

top: 100px;

left: 470px;

font-family: "Yu Gothic UI Light";

color: white;

text-align: center;

font-size: 14px;

border-radius: 8px;

border: 1px solid none;

padding: 10px 10px 10px 10px;

background-color: #d61717;

opacity: 0.7;

z-index: 50;

}

</style>

</head>

<body>

<?php

include("session.php");

$servername = "localhost";

$uname = "root";

$password1 = "";

$dbname = "online\_art\_gallery\_database\_final";

// Create connection

$conn = mysqli\_connect($servername, $uname, $password1, $dbname);

// Check connection

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

require\_once("form\_functions.php");

require\_once("functions.php");

// START FORM PROCESSING

if (isset($\_POST['submit'])) { // Form has been submitted.

$errors = array();

// perform validations on the form data

$required\_fields = array('username', 'password');

$errors = array\_merge($errors, check\_required\_fields($required\_fields, $\_POST));

$fields\_with\_lengths = array('username' => 50, 'password' => 50);

$errors = array\_merge($errors, check\_max\_field\_lengths($fields\_with\_lengths, $\_POST));

$username = trim(mysql\_prep($\_POST['username']));

$password = trim(mysql\_prep($\_POST['password']));

$hashed\_password = sha1($password);

if ( empty($errors) ) {

// Check database to see if username and the hashed password exist there.

$sql = "SELECT username ,password, USER\_ID,user\_type FROM user WHERE username = '$username' AND password = '$password'";

$result\_set = mysqli\_query($conn,$sql);

confirm\_query($result\_set);

if (mysqli\_num\_rows($result\_set) == 1) {

// username/password authenticated

// and only 1 match

$found\_user = mysqli\_fetch\_array($result\_set);

$\_SESSION['USER\_ID'] = $found\_user['USER\_ID'];

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

$\_SESSION['user\_type'] = $found\_user['user\_type']; // redirect\_to("home-2.php");

if($\_SESSION['user\_type'] == "Admin")

{

redirect\_to("account\_home.php");

}

else if($\_SESSION['user\_type'] == "Customer" | $\_SESSION['user\_type'] == "Artist" )

{

redirect\_to("home.php");

}

} else {

include("login\_form.php");

// username/password combo was not found in the database

echo '<p class = "error">Username/password combination incorrect.<br />

Please make sure your caps lock key is off and try again.</p>';

}

} else {

if (count($errors) == 1) {

Echo "There was 1 error in the form.";

} else {

echo "There were " . count($errors) . " errors in the form.";

}

}

} else { // Form has not been submitted.

if (isset($\_GET['logout']) && $\_GET['logout'] == 1) {

$message = "You are now logged out.";

}

$username = "";

$password = "";

}

?>

</body>

</html>

**Admin\_artwork.php:**

<?php session\_start();

error\_reporting(0);

include("includes/connection.php");

include("admin.php");

?>

<!DOCTYPE html>

<html>

<head>

<style>

.head-table {

font-size: 40px;

font-family: "Yu Gothic UI Light";

color: rgb( 45, 112, 213 );

position: absolute;

top: 80px;

left: 45px;

}

.head-user{

font-variant: small-caps;

font-size: 50px;

font-family: "Yu Gothic UI Light";

color: rgb( 00, 00, 00 );

position: absolute;

top: 63px;

left: 170px;

z-index: 19;

}

.hr{

position: absolute;

border: 1px solid #2d70d5;

top: 145px;

margin-left: -26px;

width: 1150px;

left: 73px;

}

.photo {

position: relative ;

width: 300px;

height : 250px;

}

.desc-title{

color:#2d70d5;

font-variant: small-caps;

font-family: "Yu Gothic UI Light";

font-size: 29px;

position: relative;

top: 0px;

left: 5px;

text-decoration: none;

}

.desc-content{

position: relative;

font-size: 18px;

font-family: "Yu Gothic UI Light";

top: 0px;

}

.desc-content2{

position: relative;

font-size: 18px;

font-family: "Yu Gothic UI Light";

top: -20px;

}

.pic-table{

border: 3px solid white;

box-shadow: 0px 6px 20px 0px rgba(0, 0, 0, 0.2);

background-color: #fafafa;

border-collapse: collapse;

float: left;

overflow: auto;

margin: 0px 50px 100px 0px;

}

.space{

margin-top: 120px;

position: relative;

left:130px;

}

.deletebutton{

border-radius: 8px;

font-family: "Yu Gothic UI Light";

font-weight: bold;

color: white;

font-size: 25px;

background-color: #FF0017;

position: relative;

top: -20px;

padding: 10px;

left: 10px;

width: 100px;

height: 35px;

text-decoration: none;

}

</style>

</head>

<body>

<script>

function YNconfirm() {

if (window.confirm('Are you sure you want to delete this artwork?'))

{

return true;

}

else

return false;

};

</script>

<p><h1 class="head-table" >TABLE: </h1>

<h1 class="head-user" >Artworks</h1><br> <hr class="hr" style="border-bottom: 2px solid #2d70d5;">

</p>

<form action="delete\_artwork.php" method="POST">

<?php

$query\_category="SELECT art\_work.art\_id, art\_work.art\_title,art\_work.art\_price, user.user\_fname, user.user\_mname,user.user\_lname,art\_work.art\_description,art\_work.art\_imagepath, art\_work.art\_category

FROM art\_work,user

where art\_work.user\_id = user.user\_id AND art\_work.art\_status = 'Available' ORDER BY ART\_WORK.ART\_DATE ASC";

$result\_category = mysqli\_query($conn,$query\_category);

if(mysqli\_num\_rows($result\_category) <=0)

{

echo '<h1 align="Center"><br><br><br><br><br><br><br>No Available Artworks </h1>';

}

else{

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

echo ' <div class="space" >

<table class="pic-table">

<tr>

<td>

<img class="photo" src="pictures/arts/'.$row['art\_imagepath'].'" > <br>'.

'<a href=info\_art.php?id='.$row['art\_id'].' class="desc-title"> '.$row['art\_title'].' </a>

<p class="desc-content">'.$row['art\_category'].'</p>

<p class="desc-content" style="float: right;">P'.$row['art\_price'].'.00</p> <br>

<p class="desc-content2">'.$row['user\_fname'].' '.$row['user\_mname'].' '.$row['user\_lname'].'</p>

<p><a class="deletebutton" href =admin\_artworks\_action.php?delete='.$row['art\_id'].'&pic='.$row['art\_imagepath'].' onclick="return(YNconfirm());" > Delete </a></p>

</td>

</tr>

</table>

</div>';

}

}

echo "<br><br>";

?>

</div>

</form>

</body>

</html>

**Artist\_details.php:**

<?php include("includes/connection.php");

include("includes/head.php"); ?>

<!DOCTYPE html>

<html>

<head>

<style>

.photo1 {

position: relative;

left: 80px;

top: 45px;

width: 207px;

height : 270px;

box-shadow: 1px 1px 5px 0px rgb(0, 0, 1);

}

.backgd{

border-radius: 8px;

border: 1px solid none;

background-color: white;

padding: 0px 10px 10px 0px;

margin-left: 120px;

position: relative;

top: 175px;

width:1010px;

height: 350px;

}

.info-title{

color:#2d70d5;

font-variant: small-caps;

font-family: "Yu Gothic UI Light";

font-size: 25px;

position: relative;

top: -205px;

left: 420px;

text-decoration: none;

}

.info-content{

position: relative;

font-size: 25px;

font-family: "Yu Gothic UI Light";

top: -115px;

left: 785px;

}

.headspace{

position: absolute;

font-variant: small-caps;

color: rgb(0, 0, 3);

font-family: "Yu Gothic UI Light";

font-size: 30px;

left: 80px;

top: 110px;

}

.headspace2{

position: absolute;

font-variant: small-caps;

color: rgb(0, 0, 3);

font-family: "Yu Gothic UI Light";

font-size: 30px;

margin-left: 80px;

margin-top: 100px;

}

.hr{

position: relative;

border: 1px solid #2d70d5;

top: -15px;

width: 1100px;

left: 0px;

}

.photo {

position: relative ;

width: 270px;

height : 260px;

margin: 0 0px 0 0px;

}

.desc-title{

color:#2d70d5;

font-variant: small-caps;

font-family: "Yu Gothic UI Light";

font-size: 29px;

position: inherit;

top: 0px;

left: 5px;

text-decoration: none;

}

.desc-content{

position: relative;

font-size: 18px;

font-family: "Yu Gothic UI Light";

top: 0px;

}

.desc-content2{

position: relative;

font-size: 18px;

font-family: "Yu Gothic UI Light";

top: -20px;

}

.pic-table{

border: 3px solid white;

box-shadow: 0px 6px 20px 0px rgba(0, 0, 0, 0.2);

background-color: #fafafa;

border-collapse: collapse;

float: left;

margin: 0px 50px 50px 0px;

}

.space{

margin-top: 200px;

position: relative;

left:130px;

}

</style>

</head>

<body>

<?php

$user\_id = $\_GET['id'];

$query\_category="SELECT user\_imagepath,user\_fname, user\_lname,user\_mname, user\_gender, user\_house\_number, user\_street, user\_brgy, user\_province, user\_contact FROM user WHERE user\_id = '$user\_id'";

$result\_category = mysqli\_query($conn,$query\_category);

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

echo ' <h2 class="headspace"> '.$row['user\_fname'].'\'s Info <hr class="hr" style="border-bottom: 2px solid #2d70d5;"></h2>

<p></p>

<div class="backgd"><a href= "pictures/profile/'.$row['user\_imagepath'].'"> <img class="photo1" src="pictures/profile/'.$row['user\_imagepath'].'"></a>

<p class = "info-title">Full Name: <br>Gender: <br>City: <br>Country: <br>Contact Info: </p></div>

<p class = "info-content">'.$row['user\_fname'].' '.$row['user\_mname'].'. '.$row['user\_lname'].'<br>'.$row['user\_gender']. '<br> '.$row['user\_province'].'<br>Philippines<br>0'.$row['user\_contact'].'</p></div> ';

}

$query\_category2="SELECT user\_fname FROM user WHERE user\_id = '$user\_id'";

$result\_category2 = mysqli\_query($conn, $query\_category2);

while ($row2=mysqli\_fetch\_array($result\_category2)) {

echo '<h2 class="headspace2"> '.$row2['user\_fname'].'\'s Artworks <hr class="hr" style="border-bottom: 2px solid #2d70d5;"></h2>';

}

$query\_category1="SELECT art\_work.art\_imagepath,art\_work.art\_id, art\_work.art\_title,art\_work.art\_price, user.user\_fname, user.user\_mname,user.user\_lname,art\_work.art\_description,art\_work.art\_imagepath,art\_work.art\_status, art\_work.art\_category FROM art\_work,user WHERE

art\_work.user\_id = user.user\_id AND

art\_work.user\_id = '$user\_id' ORDER BY art\_work.art\_title ASC";

$result\_category1 = mysqli\_query($conn,$query\_category1);

if(mysqli\_num\_rows($result\_category1) <=0)

{

echo '<br><br><br><br><br><br><br><h1 align="Center">No Artworks Available </h1>';

}

else{

while($row1=mysqli\_fetch\_array($result\_category1)){

echo ' <div class="space" >

<table class="pic-table">

<tr>

<td>

<img class="photo" src="pictures/arts/'.$row1['art\_imagepath'].'" > <br>'.

'<a href=info\_art.php?id='.$row1['art\_id'].' class="desc-title"> '.$row1['art\_title'].' </a>

<p class="desc-content">'.$row1['art\_category'].'</p>

<p class="desc-content" style="float: right;">P'.$row1['art\_price'].'.00</p> <br>

<p class="desc-content2">'.$row1['user\_fname'].' '.$row1['user\_mname'].' '.$row1['user\_lname'].'</p>

</td>

</tr>

</table>

</div>';

}

}

?>

</div>

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

</body>

</html>

**10.BIBLIOGRAPHY**

**BOOKS:**

* Luke Welling, Laura Thomson, **“PHP and MYSQL Web Development”,** Fourth Edition, Publisher: Pearson Education Inc, Year: 2008.
* RasmusLerdof, Kevin Tatroe, **“PROGRAMMING PHP”** Second Edition, and Publisher: O’Reilly Media Inc, Year: 2006.
* W.Jason Gilmore, **“BEGINNING PHP and MYSQL: From Novice to Professional”** Third Edition, Publisher: Apress, year: 2008.
* Brad Bulger, Jay Greenspan, David Wall, **“MySQL and PHP Database Applications”,** Second Edition, publisher: Wiley, Year: 2003.
* Matt Zandstra, **“Sams Teach Yourself PHP in 24 Hours”,** Third Edition, Publisher: Sams Publishing, year: 2003.
* Michael Kofler, **“The Definitive Guide to MySQL 5”,** Third Edition, Publisher: Apress, Year: 2005.

**WEB SITES:**

* www.php.net
* www.mysql.com
* www.planet-php.net
* www.w3schools.com