# *DECLARATION*

I hereby declare that the Project report titled ArtStore: An E-Commerce Platform for Art Reproductions is my original work and has not been published or submitted for any degree, diploma or other similar titles elsewhere. This has been undertaken for the purpose of partial fulfilment of B-Tech (Computer Science Engineer)

NAME: Anurag kohli

ROLL NO: 2202862

|  |
| --- |
|  |
|  |

**PREFACE**

This project report attempts to bring under one cover the entire hard work and dedication put in by me in the completion of the project work on “**ArtStore: An E-Commerce Platform for Art Reproductions “**.

I have expressed my experiences in my own simple way. I hope who goes through it will find it interesting and worth reading. All constructive feedback is cordially invited.

**ACKNOWLEDGMENT**

It is really a matter of pleasure for me to get an opportunity to thank all the persons who contributed directly or indirectly for the successful completion of the project report, “**ArtStore: An E-Commerce Platform for Art Reproductions**.”.

First of all I am extremely thankful to my university **Doaba Institute of Engineering & Technology** for providing me with this opportunity and for all its cooperation and contribution. I am highly thankful to our respected project guide for giving me the encouragement and freedom to conduct my project.

I am also grateful to all my faculty members for their valuable guidance and suggestions for my entire study.

I would also like to thank the Computer Park team for extending their valuable time and cooperation.

Anurag kohli

Roll No: 2202862

Contents

[CHAPTER 1: INTRODUCTION 6](#_Toc186909750)

[1.1General Introduction 6](#_Toc186909751)

[1.2Project Objectives 7](#_Toc186909752)

[1.3 Problem Statement 8](#_Toc186909753)

[CHAPTER 2 9](#_Toc186909754)

[SYSTEM PROPOSAL 9](#_Toc186909755)

[2.1 Existing System 9](#_Toc186909756)

[2.1.1 Disadvantages 9](#_Toc186909757)

[2.2 Proposed System 10](#_Toc186909758)

[2.3 Advantages 10](#_Toc186909759)

[CHAPTER 3 11](#_Toc186909760)

[SYSTEM DIAGRAM 11](#_Toc186909761)

[3.1 Architecture Diagram 11](#_Toc186909762)

[3.2 Flow Diagram 12](#_Toc186909763)

[3.4 Class Diagram 13](#_Toc186909764)

[Chapter 4: Implementation 15](#_Toc186909765)

[4.1 Modules 15](#_Toc186909766)

[4.2 Module Descriptions 16](#_Toc186909767)

[Chapter 5: System Requirements 19](#_Toc186909768)

[5.0 Hardware Requirements 19](#_Toc186909769)

[5.1 Software Requirements 20](#_Toc186909770)

[CHAPTER 6 21](#_Toc186909771)

[Software Description 21](#_Toc186909772)

[6.0 INTRODUCTION TO PHP 21](#_Toc186909773)

[Cross platform: 21](#_Toc186909774)

[CHAPTER 7 23](#_Toc186909775)

[INTRODUCTION TO JAVASCRIPT 23](#_Toc186909776)

[7.0 The Advantages of JavaScript 24](#_Toc186909777)

[CHAPTER 8 27](#_Toc186909778)

[INTRODUCTION TO MYSQL 27](#_Toc186909779)

[Chapter 9: Conclusion and Future Enhancements 29](#_Toc186909780)

[9.1 Conclusion 29](#_Toc186909781)

[9.2 Future Enhancements 29](#_Toc186909782)

[6.0 TABLE DESIGN 30](#_Toc186909783)

[7.0 SAMPLE COADINGS AND SAMPLE SCREEN SHOTS 31](#_Toc186909784)

[8.0 REFERENCES 53](#_Toc186909785)

**CHAPTER 1: INTRODUCTION**

**1.1General Introduction**

Digital transformation has been an important tool for companies to attract customers directly, as they continue to train various industries. In the art industry, online sales are increasing regularly, providing new ways to access artists, collectors and enthusiasts. Despite this growth, many existing platforms lack a specialized focus on high-quality art reproductions, leaving art lovers to choose from limited and often generic options that don't match their own tastes.

ArtStore is a PHP-based web application designed to meet the needs of art lovers looking for a curated collection of fine art prints. ArtStore differs from traditional e-commerce sites by focusing specifically on art: the platform offers high-quality reproductions of famous paintings, sculptures, and other art forms, allowing customers to experience the beauty of fine art in their own space. Built with PHP, ArtStore combines functionality, efficiency, and ease of use to deliver a smooth browsing and buying experience. It is designed to allow users to explore a wide range of artworks, view detailed information about each piece, choose frames and sizes, and purchase custom prints.

The system also includes safe payment processing to ensure safe transactions. With friendly designs, ArtsTore aims to fill the gap between art producers and consumers, creating a dynamic digital market to celebrate art. This platform simplifies the process of finding, purchasing and customizing artworks, making them accessible to a wider audience. Using the power of a PHP backend, ArtStore provides efficient data processing, user account management and smooth order processing – all essential features for a robust e-commerce experience.

**1.2Project Objectives**

The main objective of the ArtStore project is to develop a comprehensive e-commerce platform for selling fine art reproductions. The system is designed to manage all aspects of the online art buying process, from displaying detailed descriptions of the artworks to facilitating secure transactions and managing customer orders. ArtStore is built in PHP and focuses on providing an efficient and seamless user experience for both clients and admins.

Key objectives of the ArtStore project include:

* Creating a digital marketplace exclusively for art reproductions, making high-quality artworks accessible to a broader audience.
* Allowing customers to explore, view, and purchase art in various customizable formats, including framing options and different sizes.
* Providing secure, streamlined payment processing and user account management
* Enabling the administrator to manage artwork listings, track orders, and handle inventory effectively within a centralized system.
* Reducing manual processes related to art sales, thereby saving time and increasing operational efficiency.

**1.3 Problem Statement**

As the demand for selling art online, especially fine art prints, grows, existing e-commerce platforms often fail to meet the specific needs of art lovers and collectors. Traditional online stores are not focused on the art sector and offer limited information, customization, and overall browsing experience. As a result, art enthusiasts have a fragmented experience that does not reflect the unique charm of art as a product.

ArtStore aims to bridge this gap by creating a dedicated platform that offers a seamless and immersive shopping experience, especially for fine art prints. Challenges in managing and selling art online include accurately representing the details of each artwork, offering customization options, and ensuring a safe and secure transaction process. In addition, the administrator needs an effective tool to maintain a reserve, process order and website function without a wealth of technical knowledge.

**CHAPTER 2**

**SYSTEM PROPOSAL**

**2.1 Existing System**

In the existing system, art reproduction sales are generally conducted through either physical galleries or generic e-commerce platforms. These traditional methods lack specialized features for showcasing artwork effectively and do not cater to the specific needs of art buyers, such as customizable options, detailed views, and background information on the artworks and artists. Additionally, physical galleries have limited reach, and general e-commerce sites may not convey the uniqueness and quality of art reproductions adequately. This can create a disconnected experience for art enthusiasts looking to purchase art online.

**2.1.1 Disadvantages**

Limited Accessibility: Art reproductions are less accessible due to the limitations of physical galleries or the lack of specialization on general e-commerce platforms.

Lack of Customization: Buyers often cannot choose customized options such as framing or sizing easily.

Inconsistent Experience: Art sales on generic platforms may lack engaging, immersive experiences tailored to the aesthetics and detail expected in the art industry.

Manual Inventory and Order Management: Galleries and online stores often require significant manual effort to manage stock and orders, leading to inefficiencies.

**2.2 Proposed System**

The proposed ArtStore system aims to address the limitations of the existing system by creating a dedicated online platform for art reproductions. This platform will provide a structured, user-friendly experience that allows customers to explore and purchase art reproductions with ease. Through a web-based application, ArtStore will offer features specific to art sales, such as customizable framing and size options, detailed artwork descriptions, and secure online transactions. The platform will also include an administrative dashboard for efficient management of artwork listings, orders, and customer accounts.

**2.3 Advantages**

Enhanced Accessibility: ArtStore allows customers to browse and purchase art reproductions from anywhere, increasing accessibility and reach.

Simplified Customization: The platform provides convenient customization options, allowing customers to personalize their art purchases with framing and size choices.

Efficient Management: Administrators can easily manage the inventory, orders, and customer data within a centralized system, reducing manual work and improving operational efficiency.

Mobile Compatibility: With a responsive design, ArtStore enables users to access the platform from their mobile devices, ensuring convenience and flexibility.

**CHAPTER 3**

**SYSTEM DIAGRAM**

**3.1 Architecture Diagram**

Vvbvbv

ffgf

ADD PRODUCT

ARTSTORE MANAGEMENT SYSTEM



LOGOUT

ADD CATEOGRY

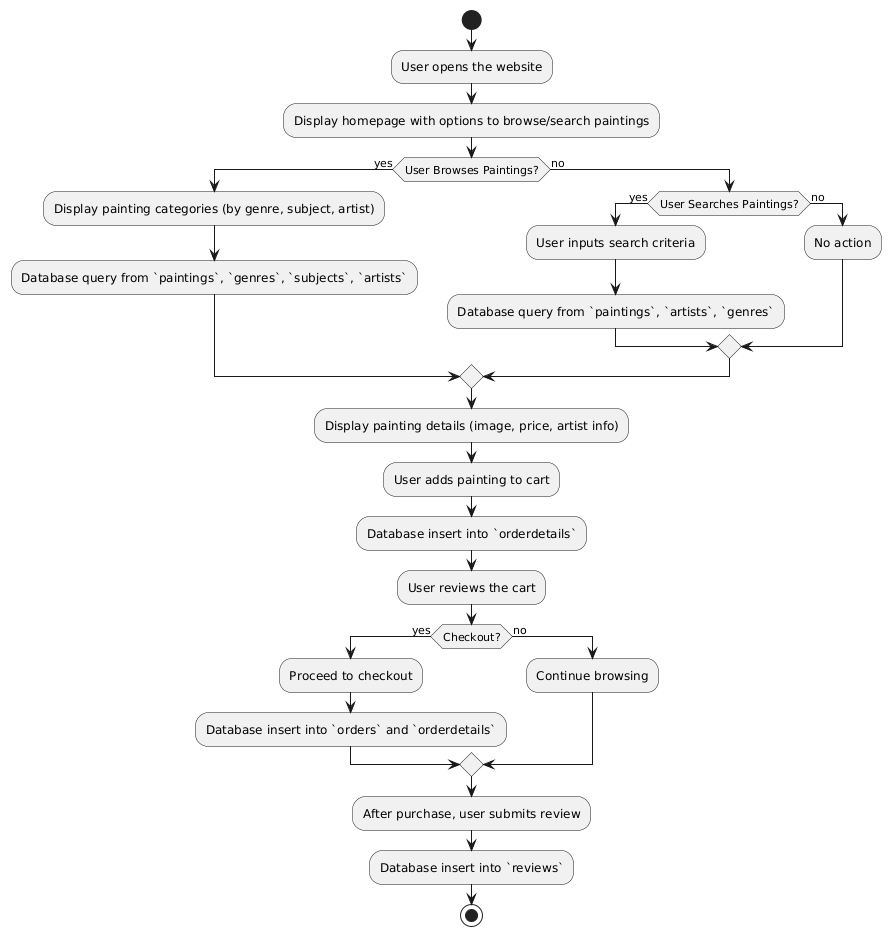
LOGIN

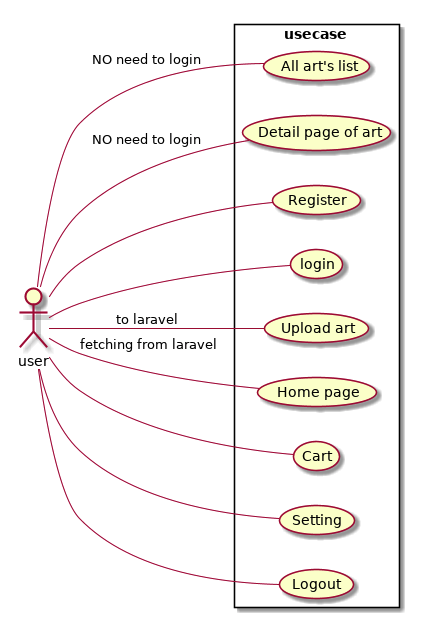
ORDER

**ADMIN**

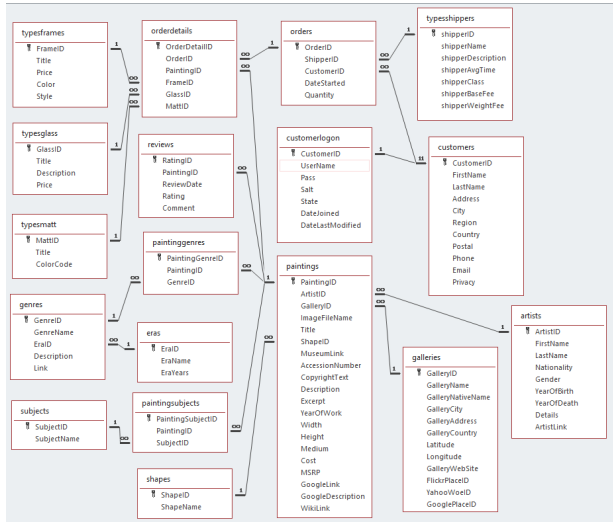
PAYMENT PROCESS

**3.2 Flow Diagram**



**3.3 USE CASE**

**3.4 Class Diagram**



**Chapter 4: Implementation**

**4.1 Modules**

For an ArtStore application, here are the key modules:

* Admin
* Customer
* Artist

**Admin Module:**

* User Management: Manage user accounts (admins, artists, customers).
* Artpiece Management: Add, edit, and delete art pieces.
* Category Management: Create and manage categories for art pieces.
* Order Management: Process and track orders.
* Payment Management: Integrate with payment gateways.
* Report Generation: Generate sales reports, inventory reports, etc.

**Customer Module:**

* User Registration/Login: Create and manage customer accounts.
* Browse Artworks: Search and filter artworks by category, artist, or keyword.
* Product Details: View detailed information about each artwork, including images, description, and pricing.
* Shopping Cart: Add and remove items from the shopping cart.
* Checkout: Process orders, including shipping address, payment details, and order confirmation.
* Order History: View past orders and order status.

**Artist Module:**

* Artist Registration/Login: Create and manage artist accounts.
* Artwork Submission: Submit new artworks for approval.
* Artwork Management: Edit and update existing artworks.
* Earnings and Royalties: View and manage earnings and royalty payments.

**4.2 Module Descriptions**

**Admin Module:**

* **User Management:**
* Create and manage different user roles (admin, artist, customer).
* Assign permissions to different user roles.
* **Artpiece Management:**
  + Add new art pieces to the store, including details like title, artist, description, price, and images.
  + Edit existing art pieces to update information or change availability.
  + Delete art pieces that are no longer available or relevant.
* **Category Management:**
  + Create new categories to organize art pieces.
  + Edit existing categories to update names or descriptions.
  + Delete unnecessary categories.
* **Order Management:**
  + Process orders, including payment verification and order fulfillment.
  + Track order status and provide shipping updates to customers.
  + Generate invoices and shipping labels.
* **Payment Management:**
  + Integrate with payment gateways to process online payments.
  + Handle refunds and chargebacks.
* **Report Generation:**
  + Generate sales reports, inventory reports, and customer reports.
  + Analyze sales data to identify trends and make informed business decisions.

**Customer Module:**

* **User Registration/Login:**
  + Create a new customer account with email and password.
  + Log in to the account to access personalized features.
* **Browse Artworks:**
  + Use search and filter options to find specific artworks.
  + View detailed information about each artwork, including artist bio, artwork description, and high-resolution images.
* **Shopping Cart:**
  + Add desired artworks to the shopping cart.
  + Modify quantities and remove items from the cart.
  + View the total cost of items in the cart.
* **Checkout:**
  + Provide shipping address and payment information.
  + Choose a shipping method and payment option.
  + Review order details and confirm the purchase.
* **Order History:**
  + View past orders, including order status, shipping information, and purchase details.
  + Track the shipment of orders.

**Artist Module:**

* **Artist Registration/Login:**
  + Create an artist account to submit artworks for sale.
  + Log in to the account to manage artwork listings and view earnings.
* **Artwork Submission:**
  + Submit new artworks, including titles, descriptions, images, and pricing.
  + Provide information about the artwork's medium, size, and availability.
* **Artwork Management:**
  + Edit existing artworks to update information or change availability.
  + Remove artworks that are no longer available for sale.
* **Earnings and Royalties:**
  + View earnings from sold artworks.
  + Track royalty payments and other financial transactions.

**Chapter 5: System Requirements**

**5.0 Hardware Requirements**

* Processor: Intel Xeon or AMD Ryzen with multiple cores
* RAM: 16GB or more
* Storage: SSD storage (500GB or more)
* Network: High-speed internet connection
* Operating System: Linux-based (Ubuntu, CentOS) or Windows Server
* Client:

**OR**

* Processor: Intel Core i3 or AMD Ryzen 3 or equivalent
* RAM: 4GB or more
* Storage: SSD or HDD (256GB or more)
* Display: 13-inch or larger
* Operating System: Windows 10, macOS, or Linux-based
* Browser: Chrome, Firefox, Safari, Edge

**5.1 Software Requirements**

* Web Server: Apache HTTP Server
* Programming Language: PHP 5.2
* Database: MySQL 5.2
* IDE: Dreamweaver

**Client-Side:**

* Web Browser: Chrome, IE8
* Operating System:
* Server: Windows 7
* Client: Windows 7, Windows 10, macOS, Linux

**CHAPTER 6**

**Software Description**

**6.0 INTRODUCTION TO PHP**

PHP is the latest incarnation of PHP (PHP: Hypertext Pre-processor)-a programming, language

devised by Rasmus Lerdorf in 1994 for building dynamic, interactive Websites. Since then, it’s

been evolving into a full-fledged language in its own right, thanks to the hard work of all the

people who contribute to its development.

A sure sign that PHP is maturing (OOP) principles and improved support for XML the send

engine (the part that interprets and executes PHP code) now enables PHP5 developers to

implement, among a host of other things, graceful application-wide error handling.

With all the new features and functionality that PHP5 provides, it’s important for programmers

to “upgrade” their understanding in order to best make use of this powerful Web scripting tool.

And that’s why it is important for you, the reader to invest your time learning about the latest

and greatest that the people developing PHP5 have to offer.

You know it’s a language for writing computer programs, so the real questions is “what sort of

programs can you write with it?” in technical terms, PHP’s main use is as a cross-platform,

html embedded, server-side web scripting language. Let’s take a moment to examine these terms.

**Cross platform**: most PHP code can be processed without alternation on computers running

many different operating systems. For Example, a PHP script that runs on Linux generally also

runs well on windows.

**HTML-embedded:** PHP code can be written in files containing a mixture of PHP

instruction and HTML code.

**Server-side:** The PHP programs are run on server-specially a web server.

**Web scripting language:** PHP programs run via a web browser.

This means you will write programs that mix PHP code and HTML, run them on a web server,

and access them from a web browser that displays the result of your PHP processing by

showing you the HTML returned by the web server. In other words, you can make your

programs available for other people to access across the web, simply by placing them on a

public web server. You are probably already familiar with HTML (hypertext mark-up

language)-it’s the main language used to create web pages, combining plain text with special

tags that tell browsers how to treat that text. HTML is used to describe how different elements

in a web page should be displayed, how pages should be linked, where to put image, and so on.

Pure HTML documents, for all their versatility, are little more than static arrangements of text

and pictures, albeit nicely presented ones. However, most of the sites you find on the web aren’t

static but dynamic even interactive. They can show you a list of articles containing a particular

word, in which you are interested, show you the latest news, even greet you by name when you

log on. They enable you to interact, and present you with different information according to

the choice you make. You can’t build a web site like that using raw HTML, and that’s where

PHP comes in. what sort of things can you do with it? Well, you can program sites that Present

data from a wide variety of sources, such as databases, files, or even other Web pages.

Incorporate interactive elements, such as search facilities, message boards, and straw polls.

Enable the user to perform actions, such as sending e-mail or buying something. In other words,

PHP can be used to write the sort of sites that those who regularly use the web are likely to

encounter every day. From search engines to information portals to e-commerce sites, most

major web sites incorporate some or all of these sorts of programming. Among other things in the course of this book, you will use PHP to build

**CHAPTER 7**

**INTRODUCTION TO JAVASCRIPT**

An explanation of exactly what JavaScript is has to begin with Java. Java is a new kind of Web

programming language developed by Sun Microsystems. A Java program, or applet, can be

loaded by an HTML page and executed by the Java Interpreter, which is embedded into the

browser. Java is a complex language, similar to C++. Java is object-oriented and has a wide

variety of capabilities; it's also a bit confusing and requires an extensive development cycle.

That's where JavaScript comes in. JavaScript is one of a new breed of Web languages called

scripting languages. These are simple languages that can be used to add extra features to an

otherwise dull and dreary Web page. While Java is intended for programmers, scripting

languages make it easy for nonprogrammers to improve a Web page. JavaScript was originally

developed by Netscape Corporation for use in its browser, Netscape Navigator. It includes a

convenient syntax, flexible variable types, and easy access to the browser's features. It can run

on the browser without being compiled; the source code can be placed directly into a Web

page. You can program in JavaScript easily; no development tools or compilers are required.

You can use the same editor you use to create HTML documents to create JavaScript, and it

executes directly on the browser (currently, Netscape or Microsoft Internet Explorer).

JavaScript was originally called Live Script, and was a proprietary feature of the Netscape

browser. JavaScript has now been approved by Sun, the developer of Java, as a scripting

language to complement Java. Support has also been announced by several other companies.

Although useful in working with Java, you'll find that JavaScript can be quite useful in its own

right. It can work directly with HTML elements in a Web page, something Java can't handle.

It is also simple to use, and you can do quite a bit with just a few JavaScript statements.

**7.0 The Advantages of JavaScript**

**An Interpreted Language:** JavaScript is an interpreted language, which requires no

compilation steps. This provides an easy development process. The syntax is completely

interpreted by the browser just as it interpreted HTML tags.

**Embedded Within HTML:** JavaScript does not requires any special or separate editor

for programs to be written edited or compiled. It can be written in any text editor like Notepad,

along with appropriate HTML tags, and saved as filename. Html. HTML files with embedded

JavaScript commands can then be read and interpreted by any browser that is JavaScript

enabled.

**Minimal Syntax-Easy to Learn:** By learning just a few commands and simple rules of

syntax, complete applications can be built using JavaScript.

**Quick Development**: Because JavaScript does not require time-consuming compilations,

scripts can be developed in a short period of time. This is enhanced by the fact many GUI

interface features, such as alerts, prompts, confirm boxes, and other GUI elements, are handle

by client side JavaScript, the browser and HTML code.

**Design for Simple, Small Programs:** It is well suited to implement simple, small

programs (for example, a unit conversion calculator between miles and kilometres or pounds

and kilograms).Such programs can be easily written and executed at an acceptable speed using

JavaScript. In addition, they can be easily interpreted into a web page.

**Performance:** JavaScript can be written such that the HTML files are fairly compact and

quite small. This minimizes storage requirements on the web server and download time for the

client. Additionally, because JavaScript are usually include in the same file as the HTML code

for a web page, they require fewer separate network accesses.

**Procedural Capabilities**: Every programming language needs to support facilities such

as Condition checking, Looping and Branching .JavaScript provides syntax, which can be used

to add such procedural capabilities to web page (filename.html) coding.

**Designed for Programming User Events**: JavaScript supports Object/Events based

programming JavaScript recognizes when a form Button is pressed. This event can have

suitable JavaScript code attached, which will executed when the Button Pressed event occurs.

JavaScript can be used to implement context sensitive help. Whenever an HTML form’s

Mouse cursor Mouse Over a button or a link on the page a helpful and informative massage

can be displayed in the status bar at the button of the browser window.

**Easy Debugging and Testing** : Being an interprets language ,scripts in JavaScript are

tested line by line, and the errors are also listed as they are encountered ,i.e. an appropriate

error message along with the line number is listed for every error that is encountered. It is

thus easy to locate errors, make changes, and test it again without the overhead and delay of

compiling.

**Platform Independence / Architecture Neutral:** JavaScript is a programming

language that is completely independent of the hardware on which it works. It is a language

that is understood by any JavaScript enabled browser .Thus ,JavaScript application work on

any machine that has an appropriate JavaScript enabled browser can be anywhere on the

network. Since each browser is for a specific platform, JavaScript interpretation will be with

respect to the specific platform. The browser will add whatever platform specific Information

is required to the JavaScript while it interprets the code. Thus, JavaScript is truly platform

independent. A JavaScript programmer developed on a UNIX machine will work perfectly

well on a Windows machine. The fact that a platform specific browser, maintained at the

client end, does the interpretation of JavaScript, relieves the developer of the responsibility of

maintaining multiple source code files for multiple platform Information is required to the

JavaScript while it interprets the code. Thus, JavaScript is truly platform independent. A

JavaScript programmer developed on a UNIX machine will work perfectly well on a

Windows machine. The fact that a platform specific browser , maintained at the client end,

does the interpretation of JavaScript , relieves the developer of the responsibility of

maintaining multiple source code files for multiple platform.

**CHAPTER 8**

**INTRODUCTION TO MYSQL**

MySQL is a fast, easy-to-use RDBMS used for databases on many Web sites. Speed was the developers’ main focus from the beginning. In the interest of speed, they made the decision to offer fewer features than their major competitors (for instance, Oracle and Sybase). However, even though MySQL isles full featured than its commercial competitors, it has all the features needed by the large majority of database developers. It’s easier to install and use than its commercial competitors, and the difference in price is strongly in MySQL’s favour. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. The company licenses its two ways:

* **Open source software**: MySQL is available via the GNU GPL (General Public License) for no charge. Anyone who can meet the requirements of the GPL can use the software for free. If you’re using MySQL as a database on a Web site (the subject of this book), you can use MySQL for free, even if you’re making money with your Web site.
* **Commercial license:** MySQL is available with a commercial license for those who prefer it to the GPL. If a developer wants to use MySQL as part of a new software product and wants to sell the new product, rather than release it under the GPL, thdeveloper needs to purchase a commercial license. The fee is very reasonable.

Finding technical support for MySQL is not a problem. You can join one of several e-maildiscussion lists offered on the MySQL Web site at <www.mysql.com>. You can even search thee-mail list archives, which contain a large knowledge base of MySQL questions and answers.If you’re more comfortable getting commercial support, MySQL AB offers technical supportcontracts — five support levels, ranging from direct e-mail support to phone support, at fiveprice levels.

**Advantages of MySQL**

MySQL is a popular database with Web developers. Its speed and small size make it ideal fora Web site. Add to that the fact that its open source, which means free, and you have thefoundation of its popularity. Here is a rundown of some of its advantages:

* **It’s fast.** The main goal of the folks who developed MySQL was speed.Consequently, the software was designed from the beginning with speed in mind.
* **It’s inexpensive.** MySQL is free under the open source GPL license, and the fee fora commercial license is very reasonable.
* **It’s easy to use.** You can build and interact with a MySQL database by using a few simple statements in the SQL language, which is the standard language for communicating with RDBMSs.
* **It can run on many operating systems.** MySQL runs on a wide variety of operating systems — Windows, Linux, Mac OS, most varieties of UNIX (including Solaris, AIX, and DEC UNIX), FreeBSD, OS/2, Irix, and others.
* **Technical support is widely available.** A large bbase of users provides free support via mailing lists. The MySQL developers also participate in the e-mail lists.You can also purchase technical support from MySQL AB for a very small fee.
* **It’s secure.** MySQL’s flexible system of authorization allows some or all database privileges (for example, the privilege to create a database or delete data) to specific users or groups of users. Passwords are encrypted.
* **It supports large databases.** MySQL handles databases up to 50 million rows or more. 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).
* **It’s customizable.** The open source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

**Chapter 9: Conclusion and Future Enhancements**

**9.1 Conclusion**

The ArtStore web application, developed using PHP and MySQL, provides a comprehensive platform for buying and selling art online. It offers a user-friendly interface, secure payment processing, and efficient order management. The platform caters to both artists and art enthusiasts, enabling them to connect and transact seamlessly.

By leveraging PHP's powerful capabilities, ArtStore delivers a dynamic and interactive user experience. The system's robust database management ensures efficient data storage and retrieval, while the integration of secure payment gateways safeguards financial transactions.

**9.2 Future Enhancements**

**Mobile App:** Develop a mobile app to provide a seamless shopping experience on smartphones and tablets.

Augmented Reality (AR): Integrate AR technology to allow users to visualize artworks in their own spaces before purchasing.

**Virtual Reality (VR):** Create immersive VR experiences to showcase artworks in virtual galleries.

**AI-Powered Recommendations:** Utilize AI algorithms to recommend personalized artwork based on user preferences and purchase history.

**Blockchain Integration:** Implement blockchain technology to ensure transparency and security in art provenance and ownership.

**Community Features:** Foster a community of artists and art enthusiasts through social features like forums, blogs, and live chat.

**Enhanced Search and Filtering:** Improve search and filtering capabilities to help users find specific artworks more efficiently.

**Multilingual Support:** Expand the platform to support multiple languages to reach a wider global audience.

**6.0 TABLE DESIGN**

**7.0 SAMPLE CODINGS**

**CREATING FRONT END PAGE TO LINK ALL TABLES**

<!DOCTYPE html>

<html>

<head>

<title>ART GALLERY DATABASE</title>

</head>

<body><h1>

<link rel="stylesheet" href='https://fonts.googleapis.com/css?family=Aclonica'>

<center>

<h1><font style="border: 13px groove powderblue; border-radius: 7px" color=floralwhite face="Aclonica"; >ART GALLERY MANAGEMENT SYSTEM</font></h1>

</center>

<style> h1{

padding-top: 0px;

border-radius: 5px;

margin-bottom: -1px;

}

.column{}

body{background: url("https://images.pexels.com/photos/277054/pexels-photo-277054.jpeg?cs=srgb&dl=adult-architecture-art-277054.jpg&fm=jpg") no-repeat;

background-size:contain;}

ul{

margin:3pc;

padding:0.1px;

list-style:none;}

ul li{

float:left;

width:265px;

height:49px;

background-color:mediumorchid;

opacity:.8;

line-height:40px;

text-align:center;

font-size:25px;

margin-right:7px;

margin-left: 5px;

font-family: verdana;

font-weight: bold;

font-size: 35px;

}

ul li a{

text-decoration:underline;

text-align: center;

color:white; display:block;

padding-bottom: 7.5px; }

ul li a:hover{

background-color:mediumseagreen;

color: yellow;

font-family: sans-serif;}

ul li ul li{

display:none; }

ul li:hover ul li {

display:block;}

</style><br>

<ul><center>

<li><a href="gallery.html" >GALLERY</a></li>

<li><a href="exhibition.html">EXHIBITION</a></li>

<li><a href="artwork.html">ARTWORK</a></li>

<li><a href="customer.html">CUSTOMER</a></li>

<li><a href="artist.html">ARTIST</a></li>

<li><a href="contacts.html">CONTACTS</a></li>

</center>

</ul> <br> </h1>

<style>

.example1 { height: 50px;

overflow: hidden;

position: relative;

}

.example1 h3 {

font-size: 3em; color: #F7F9F9;

position: absolute;

width: 100%; height: 100%;

margin-top: 0px;

line-height: 50px;

text-align: center;

-moz-transform:translateX(100%);

-webkit-transform:translateX(100%);

transform:translateX(100%);

-moz-animation: example1 15s linear infinite;

-webkit-animation: example1 15s linear infinite;

animation: example1 11s linear infinite; }

@-moz-keyframes example1 {

0%{ -moz-transform: translateX(100%); }

100%{ -moz-transform: translateX(-100%); }

}

@-webkit-keyframes example1 {

0%{ -webkit-transform: translateX(100%); }

100%{ -webkit-transform: translateX(-100%); }

}

@keyframes example1 {

0%{-moz-transform: translateX(100%);

-webkit-transform: translateX(100%);

transform: translateX(100%);}

100% {-moz-transform: translateX(100%);

-webkit-transform: translateX(-100%);

transform: translateX(-100%);}

}</style>

<div class="example1">

<h3 style="font-family: Verdana">This is an ART GALLERY PROJECT!</h3>

</div>

<style>

\* {box-sizing: border-box;}

.column{ float: left;

width: 33.33%;

padding: 1px; }

.row::after{ content: "";

clear: both; display: table;

}

</style></head>

</html>

**5.2.2 CREATION OF NEXT PAGE AFTER SELECTION**

<!DOCTYPE html>

<html>

<head>

<title>GALLERY</title>

</head>

<body>

<h1>

<link rel="stylesheet" href='https://fonts.googleapis.com/css?family=Aclonica'>

<link rel="stylesheet" href='https://fonts.googleapis.com/css?family=Lemon'>

<link rel="stylesheet" href='https://fonts.googleapis.com/css?family=Jockey One'>

<center>

<h1 style="font-size: 40"><font style="border: 14px groove floralwhite;" color=greenyellow face="Aclonica" ><u>GALLERY</font></h1><br>

</center>

<style> body{ background: url("https://leeuwinestate.com.au/wp-content/uploads/2015/07/Art-Gallery-2.jpg") no-repeat;

background-size:cover;

font-family:"Verdana";

text-color:white;

}

</style>

<b> <font style="font-family: Lemon" font color="white" size="10">

SELECT ANY ONE OPTION FROM BELOW:</font>

</b>

<style>

ul{

margin:3pc;

padding:0.1px;

list-style:none;}

ul li{

float:initial;

width:400px; height:65px;

background-color:mediumslateblue;

opacity:.8; line-height:30px;

text-align:center; font-size:35px;

margin-bottom: 12px; }

ul li a{

text-decoration:underline;

text-align: center;

font-family: Jockey One;

font-weight: bold; font-size: 6;

color:white; display:block;

}

ul li a:hover{

background-color: white;

color: black;

background-size: contain; }

ul li ul li{

display:none; }

ul li:hover ul li{

display:block; }

</style><br>

<ul> <li><a href="ginsert.html" >ADD VALUES INTO THE GALLERY </a></li>

<li><a href="gsearch.php">SEARCH VALUES FROM THE

GALLERY</a></li>

<li><a href="gdisplay.php">DISPLAY CONTENTS FROM THE

GALLERY TABLE</a></li>

<li><a href="gstored.php">STORED PROCEDURE OF THE GALLERY TABLE</a></li>

<li><a href="gdelete.php">DELETE VALUES FROM THE

GALLERY</a></li> </h1>

<p style="font-family: arial"><a href="FrontEnd.html"><font style="color:gold">GO BACK</font></a></p>

</body>

</html>

**5.2.3.a INSERTION IN FRONT END HTML CODE**

<!DOCTYPE html>

<html>

<head>

<title>Insertion in Gallery</title>

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

<style>

h1{

border: 10px solid grey;

border-radius: 28px;

padding: 19px;

}

body {

font-family: Arial, Helvetica, sans-serif;

background-color: white; }

\* {

box-sizing: border-box;}

.container {

padding: 19px; background-color: snow; }

input[type=text], input[type=text]

{

width: 50%; padding: 19px;

margin: 5px 0 32px 0;

display: inline-block;

border-radius: 8px;

border: 2px solid grey;

background: #f1f1f0;

font-weight: bold; font-size: 19px;

}

input[type=text]:focus, input[type=text]:focus {

background-color: #ffffff; outline: none;}

hr {

border: 1.5px solid #f1f1f1;

margin-bottom: 35px;

}

.registerbtn{

background-color: forestgreen;

padding: 16px 10px;

margin: 8px 20px 20px 50px;

border-radius: 24px; cursor: pointer;

width: 15%; opacity: 0.7;

align-content: center;

font-family: "verdana";

font-weight: bold; color: white;

}

.registerbtn:hover {

opacity: 1;

background-color:forestgreen;}

label{

font-weight: bold;

font-size: 20px;

font-family: 'verdana';

}

</style></head>

<body>

<form ACTION="ginsert.php"METHOD="POST">

<div class="container">

<center><h1 style="color:#3366cc" size="30";><font style="border: 10px solid grey; padding: 19px;">FILL THE FORM WITH PROPER DETAILS</font></h1></center>

<hr><center>

<label for="G\_ID"><b>Gallery ID</b></label><br>

<input type="text" placeholder="Enter G\_ID" name="G\_ID" required><br>

<label for="GNAME"><b>GNAME</b></label><br>

<input type="text" placeholder="Enter Gallery Name" name="GNAME" required><br>

<label for="LOCATION"><b>GLOCATION</b></label><br>

<input type="text" placeholder="Enter GLOCATION" name="LOCATION" required><br>

</hr>

<button type="submit" class="registerbtn">SUBMIT</button>

<button type="reset" class="registerbtn">RESET</button>

</center> </div> </form> </body>

</html>

**5.2.3.b CONNECTIVITY OF THE PHP TO DATABASE TO INSERT**

**A RECORD**

<?php

if(isset($\_POST['G\_ID']) && isset($\_POST['GNAME']) && isset($\_POST['LOCATION'])):

$gid = $\_POST['G\_ID'];

$gname = $\_POST['GNAME'];

$location = $\_POST['LOCATION'];

$link = new mysqli('localhost','root','','art\_gallery');

if($link->connect\_error)

die('connection error: '.$link->connect\_error);

$sql3 = "INSERT INTO GALLERY(gid, gname, location)

VALUES('".$gid."', '".$gname."', '".$location."')";

$result = $link->query($sql3);

if($result > 0):

echo 'Successfully Inserted into GALLERY table.';

else:

echo 'Unable to post';

endif;

$link->close();

die();

endif;

?>

**5.2.4 SEARCHING OF VALUES FROM FRONT END**

<html>

<head>

<title>Search Gallery</title>

</head>

<style> table{

border-collapse: collapse;

width: 60%;

padding: 150px;

margin-left: 280px; }

th, td {

text-align: center;

padding: 8px;

border-radius: 12px;

}

tr:nth-child(even) {

background-color: #f2f2f2;

font-family: "arial";

font-weight: bold; }

th { background-color: mediumslateblue;

color: white; font-family: "verdana";

font-weight: bold; }

input[type=text] { width: 119px;

box-sizing: border-box;

border: 2px solid #ccc;

border-radius: 9px; font-size: 16px;

background-color: white;

background-position: 10px 10px;

background-repeat: no-repeat;

padding: 22px 20px 22px 10px;

-webkit-transition: width 0.4s ease-in-out;

transition: width 0.4s ease-in-out;

font-weight: bold; font-size: 30px; }

input[type=text]:focus {

width: 60%; }

div{ font-family: "verdana";

font-weight: bold;

font-size: 30px;

font-style: bold;

margin-left:25px;

margin-top: 35px; }

.btn{ background-color: forestgreen;

color: white; padding: 16px 10px;

margin: 8px 20px 20px 50px;

border-radius: 24px;

cursor: pointer;

width: 10%; opacity: 0.7;

align-content: center;

font-family: "verdana";

font-weight: bold;

}

.btn:hover {

opacity: 1; background-color:forestgreen; }

b{ font-family: "verdana";

background-color: lightcyan;

color: black;

margin-left:80px;

border-radius: 8px;

text-align: center;

font-size: 30px;

width: 85%; }

span{ font-family: "verdana";

background-color: lightcyan;

color: black; margin-top:4px;

border-radius: 8px; text-align: center;

font-size: 30px; margin-left:0px;

width: 35%; font-weight: bold;}

</style>

<body style="background-color: lavender">

<h1><center><font style="border:9px solid grey" face="arial">SEARCH FROM GALLERY TABLE </font></center></h1>

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

<div>Enter Gallery ID:<input type="text" name="G\_ID" placeholder="G\_ID"><br></div><br><br>

<button type="submit" value ="Find" class="btn">SEARCH</button></form>

<?php

$host="localhost";

$user="root";

$password="";

$con= new mysqli($host,$user,$password,"art\_gallery");

if ($con->connect\_error) {

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

if ($\_SERVER["REQUEST\_METHOD"] == "POST")

{

$n=$\_POST['G\_ID'];

echo "<b><br>Entered Gallery ID is $n<br></b>";

$sql="select \* from gallery where gid='$n'";

$result = $con->query($sql);

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

echo "<b><br>Search Successful<br><br></b>";

echo "<br><br><br><br><table><tr><th>G\_ID</th><th>GNAME</th><th><br>LOCATION<br></br></th></tr>";

while($row = $result->fetch\_assoc())

{

echo "<tr><td>" . $row["gid"]. "</td><td>" .$row["gname"]. "</td><td><br>"

. $row["location"]. "<br></br></td></tr>"; }

echo "</table>";

} else {

echo "<span><br><br>OPPS!!! Search Unsuccessful!<br><br>There is no such Gallery ID exists. Please Enter Correct Gallery ID and Search Again.</span>"; }

}

$con->close();

?>

</body>

</html>

**5.2.5 DISPLAYING VALUES FROM FRONT END**

<!DOCTYPE html>

<html>

<head>

<title>Display Gallery</title>

<style>

table {

border-collapse: collapse;

width: 100%; color: #588c7e;

font-family: monospace;

font-size: 25px; text-align: left;

font-family:"Verdana";

font-weight: bold;

text-align: center;

border-radius: 14px; }

th{

background-color: mediumpurple;

color: white;

border-radius: 14px; }

h1{

font-family: "Arial";

font-size: 50px;

border: 9px solid grey;

border-radius: 17px;

color: black; }

td{

padding: 12px;

border-radius: 14px; }

tr:nth-child(even) { background-color: #f2f2f2;

border-radius: 14px;}

</style>

</head>

<body style="background-color: lavender">

<h1><center><font style="border:9px solid grey"> DISPLAY CONTENTS /\/ GALLERY TABLE </font></center></h1>

<table>

<tr>

<th><br>Gallery ID<br><br></th>

<th><br>GName<br><br></th>

<th><br>Location<br><br></th><br><br>

</tr>

<?php

$con = mysqli\_connect("localhost", "root", "", "art\_gallery");

if ($con->connect\_error) {

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

$sql = "SELECT \* from GALLERY";

mysqli\_query($con,$sql);

if ($result = mysqli\_query($con, $sql)){

while($row = $result->fetch\_assoc())

{

echo "<tr><td>" . $row["gid"]. "</td><td>" . $row["gname"]. "</td><td><br>". $row["location"]. "<br></br></td></tr>";

}

echo "</table>";

}else {

echo "0 results"; }

$con->close();

?>

</table></body></html>

**5.2.6** **DELETION OF VALUES FROM FRONT END**

<html>

<head>

<title>Delete from Gallery</title>

</head>

<style>b{

font-size: 30px;

font-family: 'Arial';

padding: 27px 62px;}

input[type=text] {

width: 120px;

box-sizing: border-box;

border: 2px solid #ccc;

border-radius: 9px;

font-size: 16px;

background-color: white;

background-position: 10px 10px;

background-repeat: no-repeat;

padding: 22px 20px 22px 10px;

-webkit-transition: width 0.4s ease-in-out;

transition: width 0.4s ease-in-out;

font-weight: bold;

font-size: 30px; }

input[type=text]:focus {

width: 50%; }

div{

font-family: 'Verdana';

font-size: 35px;

font-weight:bold;

margin-left:25px;

margin-top: 35px; }

.btn{ background-color: forestgreen;

color: white;

padding: 16px 10px;

margin: 8px 20px 20px 50px;

border-radius: 24px;

cursor: pointer;

width: 10%; opacity: 0.7;

align-content: center;

font-family: "verdana";

font-weight: bold; }

.btn:hover {

opacity: 1;

background-color:forestgreen;

}

</style>

<body style="background-color: lavenderblush">

<h1><center><font style="border:9px solid grey" face="arial">DELETE FROM GALLERY TABLE </font></center></h1>

<form action="gdelete.php" method="POST">

<div>Enter Gallery ID:<input type="text" name="G\_ID" placeholder="G\_ID"><br></div><br><br>

<button type="submit" value ="Delete" class="btn">DELETE</button>

<button type="reset" value ="Reset" class="btn">RESET</button>

</form>

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "art\_gallery";

$con = new mysqli($servername, $username, $password, $dbname);

if ($\_SERVER["REQUEST\_METHOD"] == "POST"){

$a=$\_POST['G\_ID'];

if($a!=""){

$sql1 = "SELECT \* from Gallery where gid='$a'";

$result = mysqli\_query($con,$sql1);

if(mysqli\_num\_rows($result)>0){

$sql3="DELETE from Gallery where gid='$a'";

mysqli\_query($con,$sql3);

echo "<b>Record with G\_ID = $a is deleted successfully.<b>";}

else{

echo "<b>!!!Error in Deleting Record!!!<br> Record '$a' was not found in database.<b>";

} }else{

echo "<b>G\_ID Field is Empty</b>";}

$con->close();}

?>

</body>

</html>

**7.1 SAMPLE OUTPUT**

**RESULTS**

This section describes the screens of “Art Gallery”. The snapshots are shown below for each module.

**SNAPSHOTS**

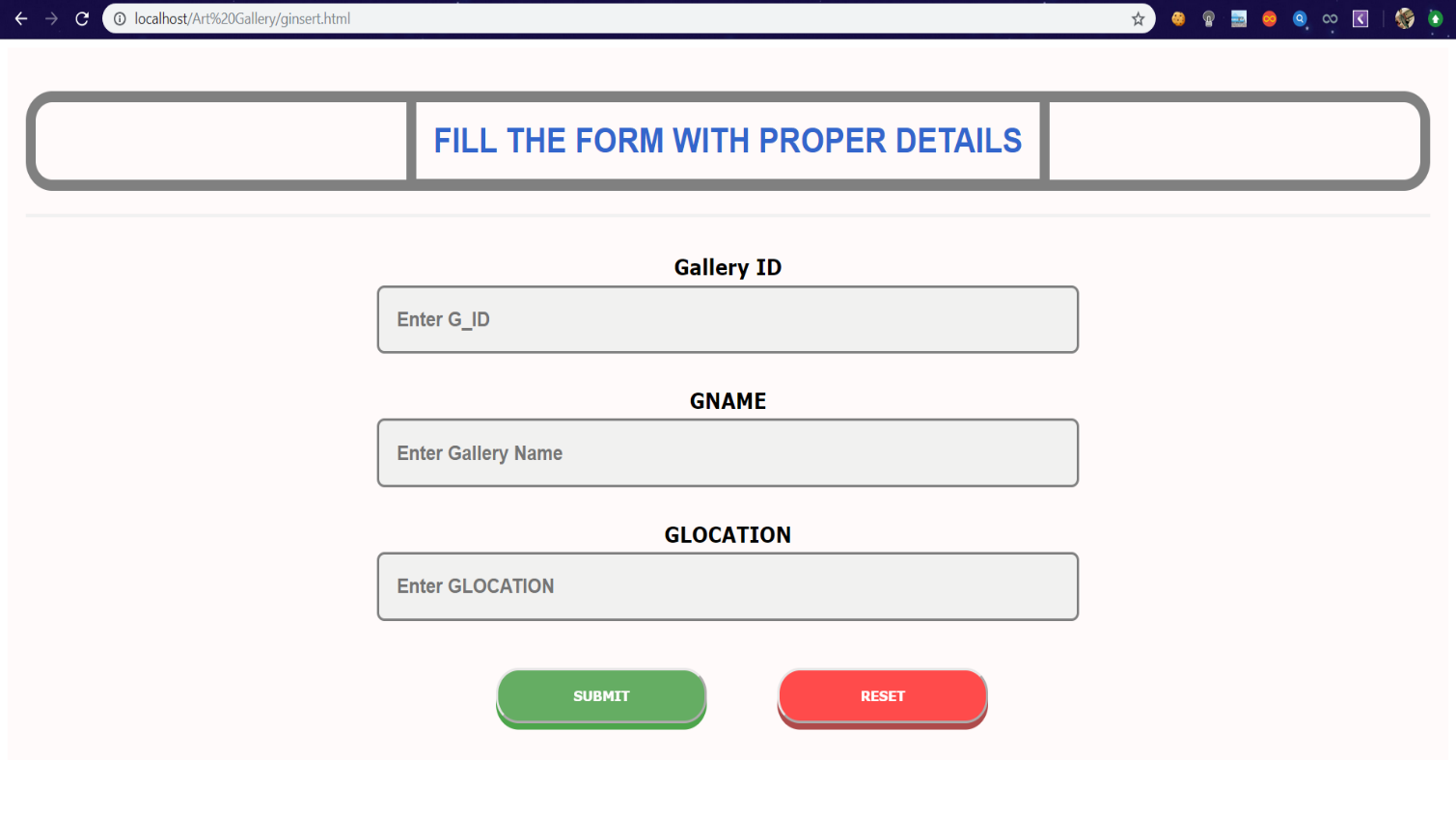
* This is the main page that shows all the operations which are present in Art Gallery Database.

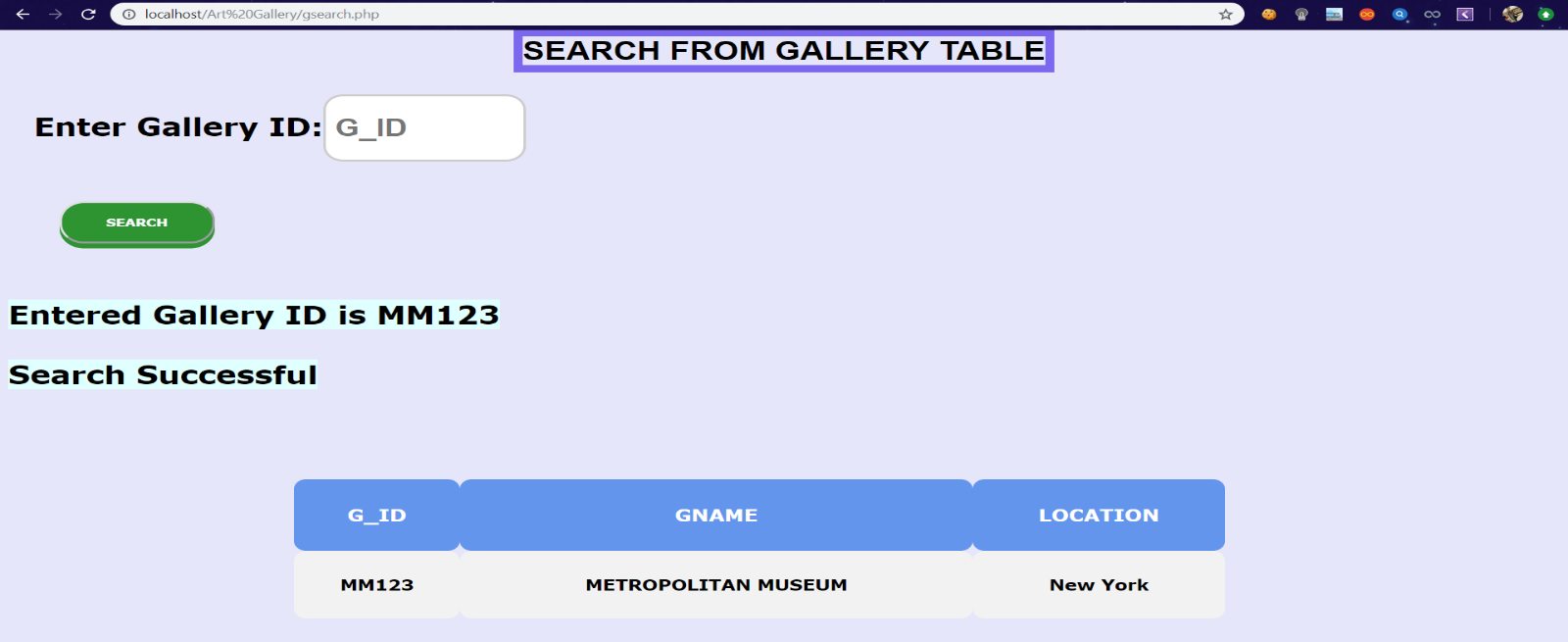
**FIGURE 7.1 ART GALLERY DATABASE MAIN PAGE**

* The selection page is the next displayed as soon as the table is selected in Main Page. Gallery table contains Insert, Search, Display and Delete tables where values can be inserted, deleted, etc.

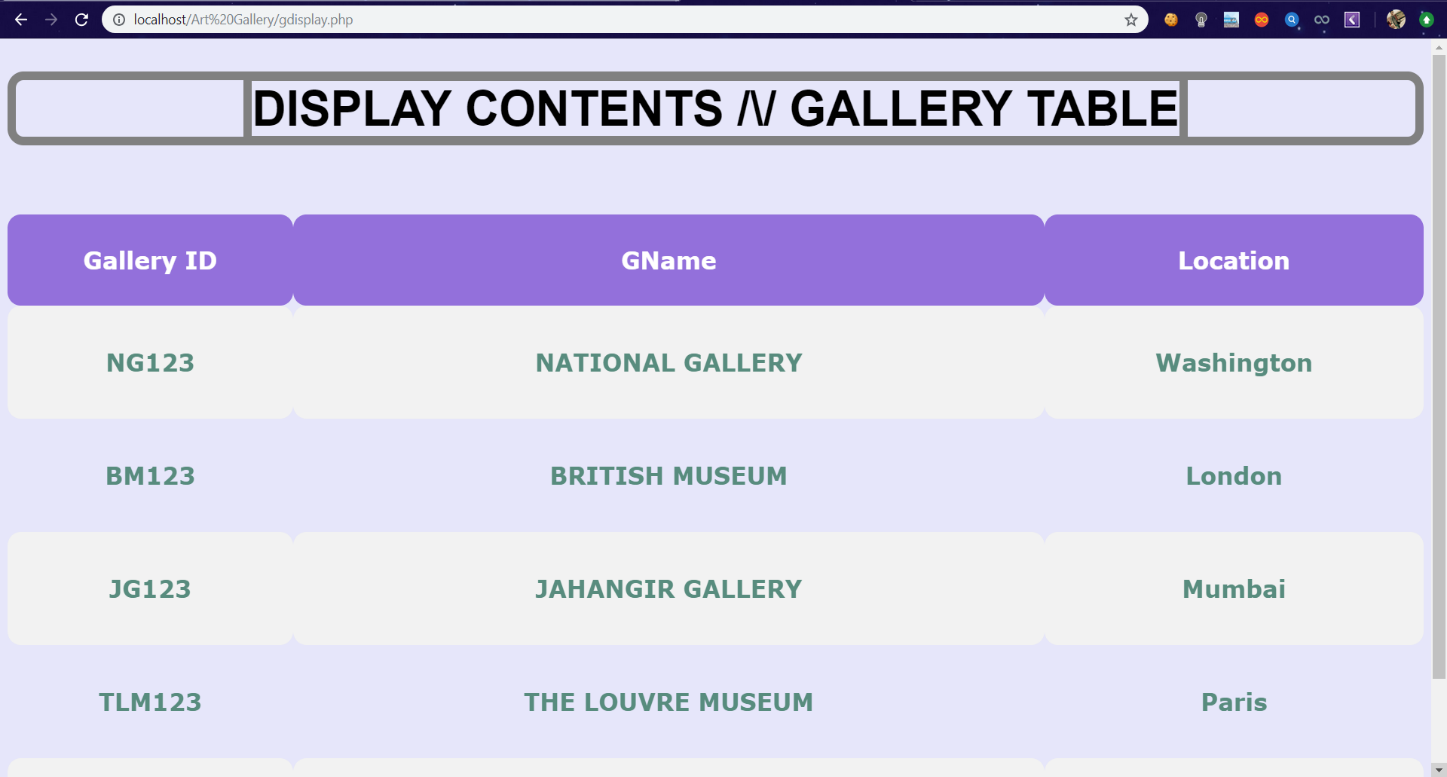
**FIGURE 7.2 GALLERY TABLE SELECTION PAGE**

* This snapshot shows the insertion page of Gallery Table. This front end page supports the insertion of all the attributes in Gallery table like GID, GName and Location.

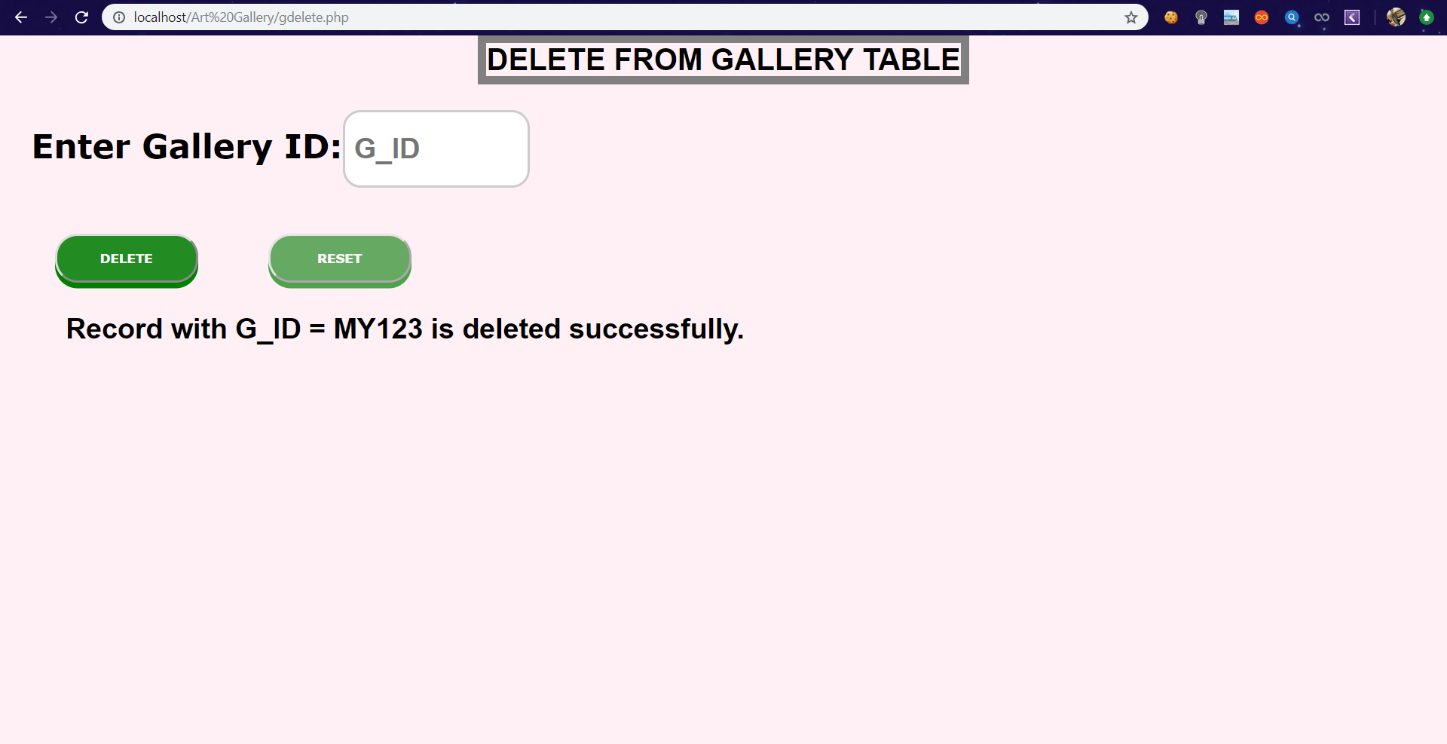
 **FIGURE 7.3: INSERTION PAGE OF GALLERY TABLE**

* This snapshot contains all the details of given GID by using Search method. Search table of Gallery contains the values like GID, GName and Location where it is searched by GID which is a primary key for this table.

**FIGURE 7.4: SEARCH PAGE OF GALLERY TABLE**

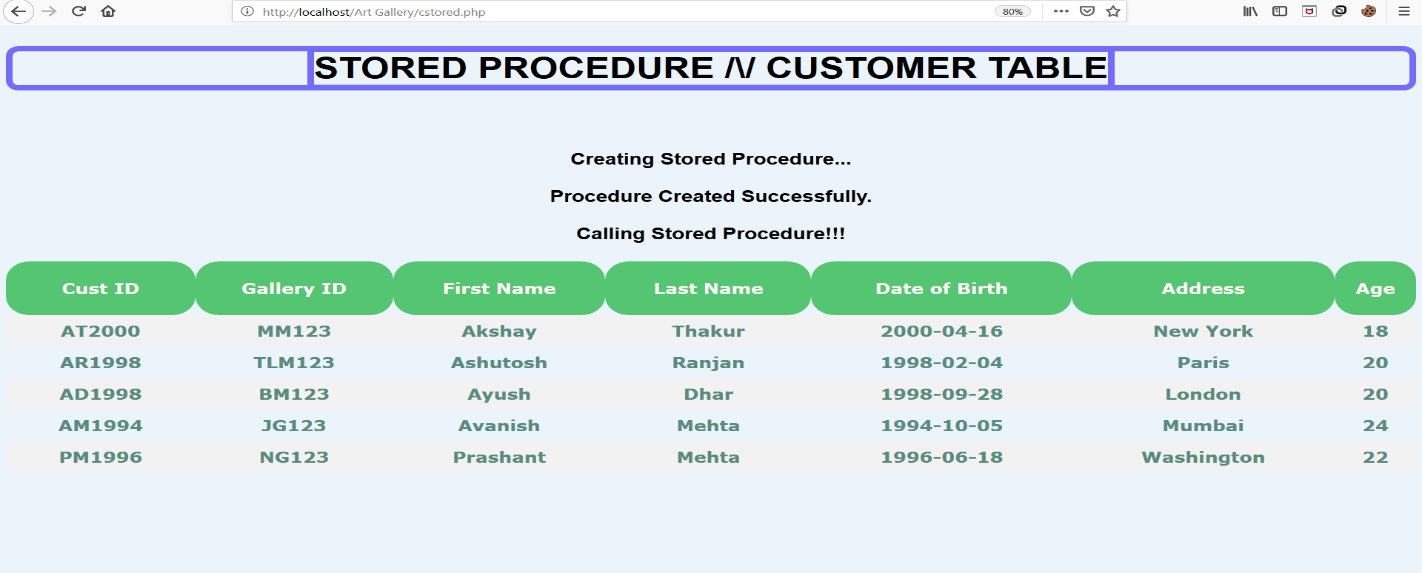
* This snapshots displays all the values entered in that table. Here, Gallery table is displayed on frontend page by using proper display query.

**FIGURE 7.5: DISPLAY PAGE OF GALLERY TABLE**

* This snapshot shows the working status of Delete Page of Gallery Table. In this page, GID is used as a parameter to delete certain record with all of their attributes in that rows. If the entered value is present in database then the value will be deleted.

**FIGURE 7.6: DELETION PAGE OF GALLERY TABLE**

* This snapshot shows the working status of Stored Procedure of Customer table. By using stored procedure, we’re calculating the Age of Customer using Date of Birth as parameter.

****

**FIGURE 7.7: STORED PROCEDURE PAGE OF CUSTOMER TABLE**

**8.0 REFERENCES**

1. <https://www.w3schools.com/php/default.asp>

2. <https://www.sitepoint.com/php/>

3. <https://www.php.net/>

4. <https://www.mysql.com/>