**1. INTRODUCTION**

An online tool called Online Art Gallery is used to display and sell works of art by artists through auction, regardless of their nationality, gender, or other specific criteria. Each artist can upload a digital copy of their artwork under the appropriate categories after registering online to become a member of the art gallery. They have the option of hosting their artwork for fixed price. Art enthusiasts must visit the exhibition to purchase their favourite works of art or paintings. However, they do not have enough time these days to visit art galleries and gather the arts and paintings.

Art is one of the most fulfilling pursuits in life. It provides an avenue for self-expression, creative fulfillment, and enjoyment. It also happens to be a viable career path. A lot of artists struggle with the idea that art is only something to be pursued for pleasure, or that art is a profession with limited financial prospects. We’re passionate about combating this idea and are committed to helping emerging artists get the tools they need to get started or advance in their artistic careers.

**Existing System:**

Customer can also register online and they can browse art works that are arranged in different categories scientifically. Each customer can easily build their own profile to view their preferred artwork. Additionally, any user has the option to buy a piece of art, add their own piece, and view all the details of other well-known pieces by purchasing a membership using the built-in payment mechanism and downloading the invoice.

**Proposed System:**

ART GALLERY is application software and it is very helpful for the art lovers and others

who want to know the details about art and add their own art.

This application helps the end-users to search their arts and paintings and they can place order for the selected pieces. The end-user can also get the information about the art work

and the respective price, so that they can so that they can contact the seller.

Art Gallery brings you an opportunity to view online art exhibitions at our Art Gallery the Online Art Gallery is updated daily, so the user can view and buy the latest collection of contemporary art online from anywhere in the world.

**2. SOFTWARE REQUIREMENT SPECIFICATION**

**2.1 FUNCTIONAL REQUIREMENTS**

Functional requirements define a function of a system or its components. A function is described as a set of inputs, their processing and outputs. The functional requirements of a system, deals with the functionality or services that the system is expected to provide and these requirements depend on the software being developed. Therefore, functional requirements describe the system functions, inputs, outputs, exceptions and so on. The functional requirements must clearly state what the system should do in abnormal situations like invalid output of error during computation specifically it should specify the behaviors of the system for invalid inputs and invalid outputs. It should also specify the action when the input is valid but normal operation cannot be performed. The functional requirements specification must be both:

1) Complete and

2) Consistent

Completeness is defining all services required by the user. Consistency means that requirements should not have conflicting definitions

**2.2 NON-FUNCTIONAL REQUIREMENTS**

Non-functional requirements requirements define constraints on the functions or services offered by the system. They include reliability, security, response time capabilities of I/O devices and data representations used in the system interfaces. They are often more critical than individual requirement since, they relate to the system as a whole rather than individual system components. They also arise due to user needs, organizational policies and due to external factors, such as safety regulations and privacy legislations.

There are 3 types of Non-functional requirements which evolve from required characteristics of the software, the organization developing the software or from external sources:

**Product requirements**

Product requirements specify the behaviors or the characteristics of the software.

These include:

Efficiency requirements: How fast the system should execute and memory requirements.

Reliability requirements set the limit of failure rate.

Probability requirements deal inter-operability with other software and hardware system.

Usability requirements

**Organizational requirements**

The organizational requirements come from the policies and procedures of the organization developing the software product such as:

Implementation of programming language or designs method to be used.

Delivery requirements specifying,

When the software product along with documentation needs to be delivered.

Process standards to be used.

**External requirements**

An external requirement is a broad area covering all the requirements desired from the external factors of the system and the development process such as:

Legislative requirements defining the safety and privacy regulations to be followed to ensure that the system operates within the law.

Inter-operability requirements define how the system interacts with system in other organization.

**2.3 SYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS**

* Processor: Intel Core i3 and above
* Input device: Keyboard and mouse
* Output device: Monitor
* RAM: 8.00GB
* System type: 64 bit- Operating system
* Hard disk space: 8GB (usable)

**SOFTWARE REOUIREMENTS**

* Front End: Macromedia dream weaver 8
* Back End: MySQL Xampp
* Operating system: Windows 10

**2.4 Technology Used**

**MACROMEDIA DREAMWEAVER 8**

Dreamweaver is a program for producing and managing Web sites. it lets you build Web pages and sites quickly and maintain them with ease. It also lets you add interactive behavior and advanced technologies to.

Dreamweaver subsequent to version 8.0 have been more compliant with [W3C](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) standards. Recent versions have improved support for [Web](https://en.wikipedia.org/wiki/World_Wide_Web) technologies such as [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets), [JavaScript](https://en.wikipedia.org/wiki/JavaScript), and various [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting) [languages](https://en.wikipedia.org/wiki/Programming_language) and [frameworks](https://en.wikipedia.org/wiki/Software_framework) including [ASP](https://en.wikipedia.org/wiki/Active_Server_Pages) (ASP JavaScript, ASP VBScript, ASP.NET C#, ASP.NET VB), [ColdFusion](https://en.wikipedia.org/wiki/ColdFusion), [Scripted](https://en.wikipedia.org/wiki/Scriptlet), and [PHP](https://en.wikipedia.org/wiki/PHP)

Dreamweaver 8 is a web design and an [Integrated Development Environment](https://en.wikipedia.org/wiki/Integrated_Development_Environment) (IDE) application that is used to develop and design websites. Dreamweaver includes a code editor that supports [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), [code completion](https://en.wikipedia.org/wiki/Code_completion), real-time [syntax checking](https://en.wikipedia.org/wiki/Syntax_analysis#Programming_languages), and code introspection for generating code hints to assist the user in writing code.

Dreamweaver, like [other HTML editors](https://en.wikipedia.org/wiki/Comparison_of_WYSIWYG_HTML_editors#Editor_features), edits [files](https://en.wikipedia.org/wiki/Computer_file) locally then uploads them to the remote web server using [FTP](https://en.wikipedia.org/wiki/File_Transfer_Protocol), [SFTP](https://en.wikipedia.org/wiki/SSH_file_transfer_protocol), or [WebDAV](https://en.wikipedia.org/wiki/WebDAV).

Some features of Dreamweaver 8 are:

* **Code:** It is a window use to write code for each functionality that is added to perform some particular tasks.
* **Design:** It is window used to customize the design for each single page in the project. That enhances look and feel.
* **Split:** It is window use to design and code simultaneously within the same window.

**XAMPP SERVER**

XAMPP 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) [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), [MariaDB](https://en.wikipedia.org/wiki/MariaDB) [database](https://en.wikipedia.org/wiki/Database), and [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)) 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). Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

XAMPP's ease of deployment means a [WAMP](https://en.wikipedia.org/wiki/WAMP) or [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) stack can be installed quickly and simply on an operating system by a developer, with the advantage that common add-in applications such as [WordPress](https://en.wikipedia.org/wiki/WordPress) and [Joomla!](https://en.wikipedia.org/wiki/Joomla!) can also be installed with similar ease.

XAMPP is regularly update to the latest releases of [Apache](https://en.wikipedia.org/wiki/Apache_HTTP_Server), [MariaDB](https://en.wikipedia.org/wiki/MariaDB), [PHP](https://en.wikipedia.org/wiki/PHP) and [Perl](https://en.wikipedia.org/wiki/Perl). It also comes with a number of other modules including [OpenSSL](https://en.wikipedia.org/wiki/OpenSSL), [phpMyAdmin](https://en.wikipedia.org/wiki/PhpMyAdmin), [Media Wiki](https://en.wikipedia.org/wiki/MediaWiki), [Joomla](https://en.wikipedia.org/wiki/Joomla), [WordPress](https://en.wikipedia.org/wiki/WordPress) and more. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another. XAMPP is offered in both a full and a standard version (Smaller version).

**Apache**

It is an HTTP a cross-platform web server. It is used worldwide for delivering web content. The server application has made free for installation and used for the community of developers under the aegis of Apache Software Foundation. The remote server of Apache delivers the requested files, images, and other documents to the user.

**phpMyAdmin**

It is a tool used for dealing with MariaDB. Its version 4.0.4 is currently being used in XAMPP. Administration of DBMS is its main role.

**XAMPP Control Panel**

It is a panel that helps to operate and regulate upon other components of the XAMPP. Version 3.3.0 is the most recent update. A detailed description of the control panel will be done in the next section of the tutorial

**2.5 Languages Used**

**PHP**

PHP is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language) [scripting language](https://en.wikipedia.org/wiki/Scripting_language) geared toward [web development](https://en.wikipedia.org/wiki/Web_development). It was originally created by Danish-Canadian [programmer](https://en.wikipedia.org/wiki/Programmer) [Rasmus Lerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf) in 1994. The PHP [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the [recursive initialism](https://en.wikipedia.org/wiki/Recursive_initialism) PHP: HypertextPreprocessor.

PHP code is usually processed on a [web server](https://en.wikipedia.org/wiki/Web_server) by a PHP [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) implemented as a [module](https://en.wikipedia.org/wiki/Plugin_(computing)), a [daemon](https://en.wikipedia.org/wiki/Daemon_(computing)) or as a [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) executable. On a web server, the result of the [interpreted](https://en.wikipedia.org/wiki/Interpreter_(computing)) and executed PHP code – which may be any type of data, such as generated [HTML](https://en.wikipedia.org/wiki/HTML) or [binary](https://en.wikipedia.org/wiki/Binary_number) image data – would form the whole or part of an [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) response. Various [web template systems](https://en.wikipedia.org/wiki/Web_template_system), web [content management systems](https://en.wikipedia.org/wiki/Content_management_system), and [web frameworks](https://en.wikipedia.org/wiki/Web_framework) exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone [graphical applications](https://en.wikipedia.org/wiki/Graphical_user_interface) and [robotic](https://en.wikipedia.org/wiki/Robotics) [drone](https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle) control. PHP code can also be directly executed from the [command line](https://en.wikipedia.org/wiki/Command-line_interface).

The standard PHP interpreter, powered by the [Zend Engine](https://en.wikipedia.org/wiki/Zend_Engine), is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [PHP License](https://en.wikipedia.org/wiki/PHP_License). PHP has been widely ported and can be deployed on most web servers on a variety of [operating systems](https://en.wikipedia.org/wiki/Operating_system) and [platforms](https://en.wikipedia.org/wiki/Computing_platform).

**JAVASCRIPT**

**JavaScript**  often abbreviated **JS**, is a [programming language](https://en.wikipedia.org/wiki/Programming_language) that is one of the core technologies of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/CSS). As of 2022, 98% of [websites](https://en.wikipedia.org/wiki/Website) use JavaScript on the [client](https://en.wikipedia.org/wiki/Client_(computing)) side for [webpage](https://en.wikipedia.org/wiki/Web_page) behavior, often incorporating third-party [libraries](https://en.wikipedia.org/wiki/Library_(computing)).

All major [web browsers](https://en.wikipedia.org/wiki/Web_browser) have a dedicated [JavaScript engine](https://en.wikipedia.org/wiki/JavaScript_engine) to execute the [code](https://en.wikipedia.org/wiki/Source_code) on [users](https://en.wikipedia.org/wiki/User_(computing))' devices.

JavaScript is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), often [just-in-time compiled](https://en.wikipedia.org/wiki/Just-in-time_compilation) language that conforms to the [ECMAScript](https://en.wikipedia.org/wiki/ECMAScript) standard. It has [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing), [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) [object-orientation](https://en.wikipedia.org/wiki/Object-oriented_programming), and [first-class functions](https://en.wikipedia.org/wiki/First-class_function). It is [multi-paradigm](https://en.wikipedia.org/wiki/Programming_paradigm), supporting [event-driven](https://en.wikipedia.org/wiki/Event-driven_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming), and [imperative](https://en.wikipedia.org/wiki/Imperative_programming) [programming styles](https://en.wikipedia.org/wiki/Programming_paradigm). It has [application programming interfaces](https://en.wikipedia.org/wiki/Application_programming_interface) (APIs) for working with text, dates, [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), standard [data structures](https://en.wikipedia.org/wiki/Data_structure), and the [Document Object Model](https://en.wikipedia.org/wiki/Document_Object_Model) (DOM).

The ECMAScript standard does not include any [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O), such as [networking](https://en.wikipedia.org/wiki/Computer_network), [storage](https://en.wikipedia.org/wiki/Data_storage), or [graphics](https://en.wikipedia.org/wiki/Computer_graphics) facilities. In practice, the web browser or other [runtime system](https://en.wikipedia.org/wiki/Runtime_system) provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but are now core components of some [servers](https://en.wikipedia.org/wiki/Server_(computing)) and a variety of [applications](https://en.wikipedia.org/wiki/Application_software). The most popular runtime system for this usage is [Node.js](https://en.wikipedia.org/wiki/Node.js).

**HTML**

**Hypertext Markup Language** (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags are used to directly introduce content into the page. Some tags are used to surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

**Features of HTML:**

* It is easy to learn and easy to use.
* It is platform independent.
* Images, video and audio can be added to a web page.
* Hypertext can be added to text.
* It is a markup language

**CSS**

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the colour of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colours are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

* **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
* **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.
* **Easy maintenance** − To make a global change, simply change the style, and al

elements in all the web pages will be updated automatically.

* **Superior styles to HTML** − CSS have a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So, it’s a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

**INTRODUCTION TO DATABASE MANAGEMENT SYSTEM**

Database Management System (DBMS) is a general-purpose software system that facilitates the process of defining, constructing, manipulating, and sharing databases among various users and applications. It is a software for storing and retrieving user data while considering appropriate security measures.

It consists of a group of programs which manipulate the database. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data.

DBMS allows users to create their own databases as per their requirement. The term “DBMS” includes the user of the database and other application programs. It provides an interface between the data and the software application

**MySQL**

**MySQL**  is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). Its name is a combination of "My", the name of co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter My ,and "SQL", the abbreviation for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). A [relational database](https://en.wikipedia.org/wiki/Relational_database) organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an [operating system](https://en.wikipedia.org/wiki/Operating_system) to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), and is also available under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) licenses. MySQL was owned and sponsored by the [Swedish](https://en.wikipedia.org/wiki/Sweden) company [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB), which was bought by [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) (now [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation)). In 2010, when Oracle acquired Sun, Widenius [forked](https://en.wikipedia.org/wiki/Fork_(software_development)) the [open-source](https://en.wikipedia.org/wiki/Open-source) MySQL project to create [MariaDB](https://en.wikipedia.org/wiki/MariaDB)

**3. SYSTEM DESIGN**

**3.1 SYSTEM DESCRIPTION**

**“ART GALLERY MANAGEMENT SYSTEM”** has 11 modules:

* Administration Module
* User Module
* Database Module
* Registration/Login Module
* Subscription Module
* Payment Module
* Update/Delete Module
* Data Viewing Module
* Interested Module
* Favorites Module
* Article Module

**Administration Module:** In this module the admin can view, edit, add the product ,article and can also have a look on the details of subscribed users .

**User Module:** In this module user can register, login, subscribe, add product, give feedback, view list of property and can add article.

**Database Module:** In this module when a new user subscribes the plan and will enter all the details. this information will be use full for admin when need in time.

**Registration/Login Module:** In this module both user and admin can login to the application to gain access to further services provided.

**Subscription Module:** In this model the user would be able to subscribe to a plan as per need .

**Payment Module:** In this module the user would have do a payment for the plan he needs and to continue the plan by paying for it.

**Update/Delete Module:** in this module both the user and admin will have access to do changes if needed to resolve the issue.

**Data viewing Module:** In this Module the admin will have access to all data of user and data related to user.

**Interested Module:** In this module the user would able to mark the product as interested and what to add to their interested list.

**Favourites Module:** In this module the admin will able to view the properties which are under user interested list.

**Article Module:** In this module admin can write and post article about their experience, offers etc.

**Receipt Module:** In this module the user can take a printout of the payment that they have made for the plan

**3.2** **FLOWCHART**

Login

Register

Admin User

Article

LOGOUT

Interested

Product Listing

Update/Delete property

Add product

Feedback

Change password

View/update profile

Article

Feedback view

Favorites

Add Admin

Update/Delete property

Add product

Dashboard

Users

Home

**3.3 DATA FLOW DIAGRAM (DFD)**

Data Flow diagram (DFD) is also called as data flow graph and commonly used during problem analysis. It is useful in understanding a system it is effectively used for portioning during problem analysis. A DFD shows the flow of data through system. It shows the movement of data through the different transformation or process in the system. It processes are shown by named circle or bubbles. An open-ended rectangle represents the data storage. Most of the cases the DFD are very large; in such a case the DFD must be developed by step wise refinement. Level DFD set has a starting DFD, which is very abstract representation of the system, identifying the major inputs and outputs, major process in the system. Then each bubble/circle in the DFD is expanded into a DFD during refinement. It is important that the net inputs and outputs for a DFD for a process are the same as the input of the process in the higher level DFD. During the refinement, the data may be broken into its component for processing.

**DATA FLOW DIAGRAM SYMBOLS**

|  |  |
| --- | --- |
| **SYMBOL** | **SIGN** |
| ARROW |  |
| CIRCLE |  |
| OPEN ENDED BOX |  |
| SQUARE |  |

#### **Description of DFD Levels**

**Level 0:**

DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modelled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

**Level 1:**

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its subprocesses.

**Level 2:**

DFD Level 2 then goes one step deeper into parts of Level 1. It may require more text to reach the necessary level of detail about the system’s functioning.

Progression to Levels 3, 4 and beyond is possible, but going beyond Level 3 is uncommon. Doing so can create complexity that makes it difficult to communicate, compare or model effectively.

**DFD LEVEL 0**

**Admin**

Admin

Verification process

**DFD LEVEL 1**

Admin

Verification process

login

validation

validation

**DFD LEVEL 2**

Login

Admin

process

validation

verification

validation

Add Add Add View View

Feedback

Favourites

News

Product

Admin

Edit/Delete

Edit/Delete

**DFD LEVEL 0**

**User**

User

Verification process

**DFD LEVEL 1**

User

Verification process

validation

Active plan

Login

validation

**DFD LEVEL 2**

Home Page

Validation

Login

User

process

verification

Home page

View View View Add View

Subscribed user

Product listing

Feedback

Product

Article

Shows Adds

View details

Edit/Delete

Interested

**3.4 ER DIAGRAM**

The ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

An ER diagram looks very similar to the flowchart. ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

**Entity:**

Entity is the thing which we want to store information. It is an elementary basic building block of storing information about business process. An entity represents an object defined within the information system about which you want to store the information. Entities are distinct things in the enterprise.

**Relationships:**

A relationship is named collection or association between entities or used to relate two or more entities with some common attributes or meaningful interaction between the objects.

**Attributes:**

Attributes are the properties of the entities and relationship. Descriptor of the entity. Attributes are elementary pieces of information attached to an entity.

|  |  |
| --- | --- |
| **DESCRIPTION** | **SYMBOL** |
| Entity |  |
| Relationship |  |
| Attribute |  |

**NORMALIZATION**

In relational database, the process of organizing data to minimize redundancy is called as Normalization. It is usually involves dividing a database into two or more tables and defining relationships between the tables. The objectives is to isolate data so that additions, deletions and modification of the fields can be made in just one then propagated through the rest of the database vis defined relationships. There are three normal forms, each with increasing levels of normalization:

**First Normal Form (1 NF):**

The relation is in first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if every attribute in that relation is singled valued attribute

**ER DIAGRAM**

User

plan

with

User plan

makes

Views/add

is

product

manages

admin

agms

gives

article

manages

view

gives

feedback

interested

**3.5 TABLE DESCRIPTION**

**ADMIN REGISTRATION TABLE (admin\_reg):**

This is admin registration table , in this admin can register himself by entering the below details. And the login credentials are also stored in this table.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Datatype** | **Description** |
| Adname | varchar(20) | Admin name |
| admin\_id | varchar(10) | Id of admin |
| admin\_email\_id | varchar(25) | Email of the admin |
| admin\_mobile\_no | varchar(10) | Mobile number of admin |
| admin\_password | varchar(20) | Admin password |

**FEEDBACK TABLE (feedback):**

This is a feedback table , in which all user’s feedback are stored in this table. Users can give their feedback regarding the website and all the user’s suggestion are stored in this table.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Datatype** | **Description** |
| User\_id | varchar(20) | Id of user |
| Fname | text | Name of user |
| Femail | varchar(35) | Email of user |
| Phone | varchar(10) | Phone number of user |
| Feedback | tinytext | Feedback given by user |
| Suggestions | text | Suggestion given by user |
| reg\_date | varchar(100) | Date when the feedback is given |

**INTRESTED TABLE (interested):**

This is a table in which admin can view user’s interested property. The user’s interested property are recorded in this table.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Datatype** | **Description** |
| user\_id | varchar(15) | Id of user |
| Pid | Int(50) | Automatic generated property id |
| Logintime | varchar(100) | Time when the property was selected as interested |

**PLAN TABLE (plan):**

In this table, the user gets the details about the subscription for the plan. Accordingly, the users can subscribe any one of the plans given.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Datatype** | **Description** |
| Plan\_id | varchar(100) | Indicates no of plans present |
| Plan\_Description | varchar(100) | Describes the plan |
| Plan\_Duration | varchar(100) | Basic duration for the plan |
| Plan\_Amount | varchar(100) | Amount charged for the plan |

**REGISTRATION TABLE (user\_reg):**

In this table, it stores the details who have registered for the website. User can register into the website by giving the following details. And user’s login details are also stored here.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Datatype** | **Description** |
| Name | varchar(25) | User name |
| User\_id | varchar(20) | Id chosen by user |
| Email\_id | varchar(40) | User’s email id |
| Mobile\_no | varchar(10) | User’s mobile no |
| Password | varchar(15) | Password set by user |

**USER PLAN TABLE (userplan):**

In this table, it stores the details about the user’s who have subscribed for the particular plan. The user’s plan detail are stored in this table.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Datatype** | **Description** |
| Plan\_id | varchar(100) | Id of the plan |
| Plan\_Description | varchar(100) | Describes the plan |
| Plan\_Duration | varchar(100) | Duration the plan |
| Plan\_Amount | varchar(100) | Amount for the plan |
| Plan\_Start\_Date | varchar(100) | Start of the plan |
| Plan\_End\_Date | varchar(100) | End of the plan |
| Email\_id | varchar(100) | User Email id |

**PRODUCT TABLE (product):**

This is a table in which admin/user can adds the property. And all the details regarding adding of the property are stored in this table.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Datatype** | **Description** |
| Pid | int(255) | Automated id generated |
| user\_id | varchar(255) | ID of user who added the property |
| admin\_id | varchar(255) | ID of admin who added the property |
| Pname | varchar(255) | Product name |
| artist\_name | varchar(255) | Artist name |
| Pyear | int(255) | Product year |
| Style | varchar(255) | Product style |
| Medium | varchar(255) | Product medium |
| Subject | varchar(255) | Product subject |
| Price | varchar(255) | Product price |
| Height | int(50) | Product height |
| Width | int(50) | Product width |
| Image | varchar(500) | Product image |
| short\_desc | varchar(500) | Product short description |
| long\_desc | varchar(500) | Product long description |
| Speciality | varchar(500) | Product speciality description |

**ARTICLE TABLE (article):**

In this, the admin can add article related to Art gallery to the user. So that user will get the inform is added by the admin.

|  |  |  |
| --- | --- | --- |
| **Column name** | **Datatype** | **Description** |
| Id | int(50) | Id of article |
| Title | varchar(100) | Title give by user/admin for article |
| Content | longtext | The article they would write |
| image\_file | varchar(500) | They can also post a picture along article |
| article\_date | Date | The date when article is posted |

**4.TESTING**

**4.1 SYSTEM TESTING**

**TESTING: -** Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

**Stages:**

**Unit Testing:** is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected.

**Integration Testing**: is defined as a type of testing where software modules are integrated logically and tested as a group.

**External Interfaces Testing:** Interface Testing is defined as a software testing type which verifies whether the communication between two different software systems is done correctly.

**Security Testing:** is a type of Software Testing that uncovers vulnerabilities, threats, risks in a software application and prevents malicious attacks from intruders

**System Testing:** System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.

**Black Box Testing**: Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths.

**Testing Process**

The testing process starts with a test plan. The test plan specifies all the activities needed with the schedule and the guidelines for testing. Based on the plan, testing begins for each unit the test cases are selected and specified. The test unit is executed with the test cases. Reports are produced and analyzed. When testing of some of the units is complete, these tested units can be combined with other untested modules to form new units. Testing begins with test plans and terminates with acceptance testing. A number of units are formed during testing, the final unit being the entire system There are three basic activities involved in testing the units:

• Select the test case

• Execute the test case

• Evaluate the results of the testing

**Test Plan**

It is the document for the entire project. It defines the scope, approach, and schedule for testing. It also specifies the persons responsible for the different activities of testing. The inputs for forming the test plan are:

• The project plans

• Requirement document

• System design document

**Test Case Specification**

This is a major activity in the testing process. Test cases have to be specified in each unit.

The test case specification gives the details of all the cases, inputs to be used and the expected outputs for these test cases.

**Approaches to testing**

There are two approaches to testing:

• Functional testing

• structural testing

**Functional testing**

As the name suggests, in function testing, the structure of the program is not considered. The test cases are decided on the basis of the requirement of the program modules and the intervals of the nodules are not considered for the selection of the test cases. Functional testing also called Black Box Testing. There are no practice criteria for selecting the test cases but the number of techniques being used for selecting the test cases is found to be successful in detecting errors. Some of the techniques are:

• Equivalence class partitioning

• Boundary value analysis

• Cause-effect graphing

• Special causes

**Structural Testing**

The main objectives of structural testing are not to exercise all the different input and output conditions but to test different programming structure and data structures used in the program. The testing aims to achieve test cases that will force the desired coverage of the different structures. To determine if the coverage is sufficient, some coverage criteria are used.

**4.2** **VALIDATION**

The goals of verification and validation activities are to access and improve the quality of the work products generated during the development and modification of the software. Quality attributes of interest include correctness, completeness, consistence, reality, usefulness, efficiency, and conformance of standards, and overall cost effectiveness.

Verification and validation involve assessment of work products to determine conformance to specification. Specification includes the requirements specification, the design documentation, and various stylistic guidelines, implementation languages standards, project standards, organization standards, and user expectation, as well as the meta-specifications for formats and notations used in the various product specifications. The requirements must be examined for conformance to user needs, environmental constraints and notational standards. The documentation must be verified with respect to the requirements and notational convections, and the source code must be examined for the conformance to the requirements, the design documentation standards. In addition, the supporting documents (user’s manual, test plans, principle of operation etc.) must be examined for correctness, completeness, consistency, adherence to standards Validation is the process of evaluating the software at the end of software development testing.

**4.3 Test cases:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST**  **CASE**  **NO.** | **TEST**  **SCENARIO** | **TEST**  **CASE** | **TEST**  **STEPS** | **INPUT**  **DATA** | **EXPECTED**  **RESULT** | **ACTUAL**  **RESULT** | **STATUS** |
| 1. | Registration  Page | When user  enters a new login id, email\_id,  password then  registration  successfully | Enter new  Login id  Enter new email\_id  Enter password | New  Login\_id  And valid  Email  address | Registration  successfully | Same as expected | PASS |
| 2. | Registration page | When user enters already registered emailid and password the registration unsuccessful | Enter preexisting  Loginid  Enter password | Pre-existing  Loginid  And valid  password | Loginid  Already  exist | Same as  Expected  result | PASS |
| 3. | Login page | When user  Enters registered userid and password then login successful | Enter userid  Enter password | Valid userid and password | Login successful | Same as expected | PASS |
| 4. | Login page | When user enters unregistered  Userid and password | Enter unregistered  Userid and password | Invalid userid and password | Login unsuccessful | Same as expected | PASS |
| 5. | Property listing page | When the user is subscribed | Views the property details | subscribed | Subscription successful | Same as expected | PASS |
| 6. | Property  Listing page | When the user is not subscribed | Unable to view the property details | Not subscribed | Subscription unsuccessful | Same as expected | PASS |
| 7. | Change password page | When user does not enter any credentials  In change password | Does not enter userid  password and confirm password | Empty credentials | Please fill out the fields | Same as expected | PASS |
| 8. | Change password page | when user enters correct password and confirm password | Enter password and confirm password | Both password and confirm password password are same | Password changed successfully | Same as expected | PASS |

# 5. SAMPLE CODING

**User Registration:**

<!DOCTYPE html>

<html>

<head>

<title>Registerform</title>

<style>

\* {

margin: 0px;

padding: 0px;

}

body {

font-size: 80%;

background-image: url('images/c1.png');

font-family: montserrat;

background-size: cover;

background-repeat: no-repeat;

}

.header {

width: 26.5%;

margin: 45px auto 0px;

color: white;

background: #107279;

text-align: center;

border: 1px solid #B0C4DE;

border-bottom: none;

border-radius: 10px 10px 0px 0px;

padding: 10px;

font-size: 16px;

}

form,

.content {

width: 25%;

margin: 0px auto;

padding: 20px;

border: 1px solid #80C4DE;

background: white;

border-radius: 0px 0px 10px 10px;

}

.input-group {

margin: 10px 0px 10px 0px;

}

.input-group label {

display: block;

text-align: left;

margin: 3px;

font-size: 16px;

}

.input-group input {

height: 30px;

width: 90%;

padding: 5px 10px;

font-size: 16px;

border-radius: 5px;

border: 1px solid gray;

}

.button1 {

padding: 15px 30px;

font-size: 15px;

color: white;

background: #107279;

;

border: none;

border-radius: 4px;

text-decoration: none;

display: inline block;

margin: 3px 1px;

transition-duration: 0.4s;

}

.success {

color: #3C763D;

background: #dff0d8;

border: 1px solid #3c763d;

margin-bottom: 20px;

}

</style>

</head>

<script>

function checkpassword(form) {

password = form.password.value;

confirmpassword = form.confirmpassword.value;

if (password != confirmpassword) {

alert("Passwords did not match!\nPlease enter same password in both the fields.")

return false;

} else {

return true;

}

}

</script>

<body>

<div class="header">

<h2>Register</h2>

</div>

<form action="user\_reg.php" method="POST" onsubmit="return submitUserForm();" style="text\_align:centre;">

<div class="input-group">

<label><b>User Name</b></label>

<div class="input-group">

<input type="text" name="name" id="name" placeholder="Full Name" required>

</div>

<div class="input-group">

<label><b>User ID</b></label>

<input type="text" name="user\_id" id="user\_id" pattern="[A-Za-z0-9]{8}" title="Can contain letters and numbers which is of 8 characters" placeholder="User Id" required>

</div>

<div class="input-group">

<label><b>Email Id</b></label>

<input type="text" name="email\_id" id="email\_id" pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$" placeholder="Email Id" required>

</div>

<div class="input-group">

<label><b>Mobile No</b></label>

<input type="text" name="mobile\_no" id="mobile\_no" pattern="[0-9]\*" maxlength="10" minlength="10" placeholder="Mobile No" required>

</div>

<div class="input-group">

<label><b>Password</b></label>

<input type="password" name="password" id="password" placeholder="password" pattern="(?=.\*\d)(?=.\*[A-Z]).{6,}" title="Must contain atleast one number and one uppercase letter and have atleast 6 or more characters" required>

</div>

<label><b> Confirm Password</b></label>

<input type="password" name="confirmpassword" id="confirmpassword" placeholder=" Confirm Password" required>

<div class="input-group">

<div class="g-recaptcha" data-callback="verifyCaptcha" data-sitekey="6LcoA8QbAAAAALWlCxUEnz3T8n6HXkNAk7h7v5Ck"></div>

<div id="g-recaptcha-error"></div>

</div>

<input class="button1" type="submit" value="Register" id="Register" onclick="return checkpassword(form)"><br>

<p><b>Already have an account??<a href="login.html">Login</a></b></p>

</div>

</form>

<script src="https://www.google.com/recaptcha/api.js" async defer></script>

<script>

function submitUserForm() {

var response = grecaptcha.getResponse();

console.log(response.length);

if (response.length == 0) {

document.getElementById('g-recaptcha-error').innerHTML =

'<span style="color:red;">This field is required.</span>';

return false;

}

return true;

}

function verifyCaptcha() {

console.log('verified');

document.getElementById('g-recaptcha-error').innerHTML = '';

}

</script>

</body>

</html>

**Add Property:**

<html>

<head>

<meta charset="utf-8">

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

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- ===== BOX ICONS ===== -->

<link href='https://cdn.jsdelivr.net/npm/boxicons@2.0.5/css/boxicons.min.css' rel='stylesheet'>

<!-- ===== CSS ===== -->

<link rel="stylesheet" href="assets/css/styles.css">

<link rel="stylesheet" href=

"https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

<script src=

"https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">

</script>

<script src=

"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js">

</script>

<script src=

"https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js">

</script>

<style>

div.one{

margin-top:10px;

margin-left:210px;

text-align:left;

}

</style>

</head>

<body id="body-pd">

<header class="header" id="header">

<div class="header\_\_toggle">

<i class='bx bx-menu' id="header-toggle"></i>

</div>

<div class="header\_\_img">

<img src="images/icon1.png" alt="">

</div>

</header>

<div class="l-navbar" id="nav-bar">

<nav class="nav">

<div>

<a href="#" class="nav\_\_logo">

<i class='bx bxs-user nav\_\_icon'></i>

<span class="nav\_\_logo-name">Admin</span>

</a>

<div class="nav\_\_list">

<a href="admindashboard.php" class="nav\_\_link active">

<i class='bx bx-grid-alt nav\_\_icon' ></i>

<span class="nav\_\_name">Dashboard</span>

</a>

<a href="adminay.php" class="nav\_\_link">

<i class='bx bx-box nav\_\_icon' ></i>

<span class="nav\_\_name">Overview</span>

</a>

<a href="adminviewprofile.php" class="nav\_\_link">

<i class='bx bx-user nav\_\_icon' ></i>

<span class="nav\_\_name">Users</span>

</a>

<a href="propadd.php" class="nav\_\_link">

<i class='bx bx-home nav\_\_icon' ></i>

<span class="nav\_\_name"> Add Property </span>

</a>

<a href="up2.php" class="nav\_\_link">

<i class='bx bx-pencil nav\_\_icon' ></i>

<span class="nav\_\_name">Update/Delete</span></a>

<a href="newadreg.html" class="nav\_\_link">

<i class='bx bx-user-plus nav\_\_icon'></i>

<span class="nav\_\_name">Add Admin</span>

</a>

<a href="fetch.php" class="nav\_\_link">

<i class='bx bx-book-heart nav\_\_icon'></i>

<span class="nav\_\_name">Favorites</span>

</a>

<a href="viewfeedback.php" class="nav\_\_link">

<i class='bx bx-box nav\_\_icon' ></i>

<span class="nav\_\_name">Feedback View</span>

</a>

<a href="blog.php" class="nav\_\_link">

<i class='bx bx-bookmark nav\_\_icon' ></i>

<span class="nav\_\_name">Article </span>

</a>

<a href="index.html" class="nav\_\_link">

<i class='bx bx-log-out nav\_\_icon'></i>

<span class="nav\_\_name">Logout</span>

</a>

</div>

</div>

</nav>

</div>

<!--===== MAIN JS =====-->

<form name="myform" onsubmit="return validation()" action="pinsert.php" method="POST" enctype="multipart/form-data">

<div class="container">

<form class="form-horizontal" role="form">

<br><br><br><h2>Add Form</h2><br>

<div class="form-group">

<label for="pname" class="col-sm-3 control-label">PAINTING NAME</label>

<div class="col-sm-9">

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

<span class="help-block">eg.: Mona Lisa </span>

</div>

</div>

<div class="form-group">

<label for="artist\_name" class="col-sm-3 control-label">ARTIST NAME</label>

<div class="col-sm-9">

<input type="text" id="artist\_name" name="artist\_name" placeholder="Artist Name" class="form-control" autofocus required="required" >

<span class="help-block">eg.: Leonardo Da Vinci</span>

</div>

</div>

<div class="form-group">

<label for="pyear" class="col-sm-3 control-label">Year</label>

<div class="col-sm-9">

<input type="text" id="pyear" name="pyear" placeholder="Painting Year" class="form-control" autofocus required="required" >

<span class="help-block">eg.: 1990</span>

</div>

</div>

<div class="form-group">

<label for="style" class="col-sm-3 control-label">TYPES OF PAINTING STYLES </label>

<div class="col-sm-9">

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

<option>select option</option>

<option value="Realism">Realism</option>

<option value="Photorealism">Photorealism</option>

<option value="Expressionism">Expressionism</option>

<option value="Impressionism">Impressionism</option>

<option value="Abstract">Abstract</option>

<option value="Surrealism">Surrealism</option>

<option value="Pop Art">Pop Art</option>

<option value="Other">Other</option>

</select>

</div>

</div>

<div class="form-group">

<label for="medium" class="col-sm-3 control-label">TYPES OF PAINTING

MEDIUMS</label>

<div class="col-sm-9">

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

<option>select option</option>

<option value="Oil">Oil </option>

<option value="Watercolour">Watercolour</option>

<option value="Acrylic">Acrylic</option>

<option value="Gouache">Gouache</option>

<option value="Pastel">Pastel</option>

<option value="Encaustic">Encaustic</option>

<option value="Fresco">Fresco</option>

<option value="Spray Paint">Spray Paint</option>

<option value="Digital">Digital</option>

<option value="Others">Others</option>

</select>

</div>

</div>

<div class="form-group">

<label for="subject" class="col-sm-3 control-label">TYPES OF PAINTING

SUBJECTS</label>

<div class="col-sm-9">

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

<option>select option</option>

<option value="History Painting">History Painting</option>

<option value="Portrait Art">Portrait Art</option>

<option value="Genre Painting">Genre Painting</option>

<option value="Landscape Painting">Landscape Painting</option>

<option value="Still Life Painting">Still Life Painting</option>

<option value="Others">Others</option>

</select>

</div>

</div>

<div class="form-group">

<label for="price" class="col-sm-3 control-label">Price</label>

<div class="col-sm-9">

<input type="text" id="price" name="price" placeholder="Enter the price" class="form-control" autofocus required="required" >

<span class="help-block">eg.: $71</span>

</div>

</div>

<div class="form-group">

<label for="height" class="col-sm-3 control-label">PAINTING HEIGHT</label>

<div class="col-sm-9">

<input type="text" id="height" name="height" placeholder="height" class="form-control" autofocus required="required" >

<span class="help-block">In Centimeter</span>

</div>

</div>

<div class="form-group">

<label for="width" class="col-sm-3 control-label">PAINTING WIDTH</label>

<div class="col-sm-9">

<input type="text" id="width" name="width" placeholder="width" class="form-control" autofocus required="required" >

<span class="help-block">In Centimeter</span>

</div>

</div>

<br><div class="form-group">

<label for="short\_desc" class="col-sm-3">SHORT DISCRIPTION</label>

<textarea class="form-control" name="short\_desc" rows="3" cols="6" style="height:auto"></textarea>

</div>

<div class="form-group">

<label for="long\_desc" class="col-sm-3">LONG DISCRIPTION</label>

<textarea class="form-control" name="long\_desc" rows="3" cols="6" style="height:auto"></textarea>

</div>

<div class="form-group">

<label for="speciality" class="col-sm-3">Speciality of your painting</label>

<textarea class="form-control" name="speciality" rows="3" cols="6" style="height:auto"></textarea>

</div>

<div class="form-group " >

<label for="images" class="col-sm-3">Upload Painting Picture</label>

<input type="file" id="images" name="images" multiple/>

</div>

<div class="one">

<button type="submit" name="submit" value="submit" class="btn btn-success active">Submit</button>

</div>

</div>

</form>

<script>

var img=document.forms['myform']['images'];

var validExt=["jpeg","png","jpg"];

function validation(){

if(img.value!=''){

var img\_ext=img.value.substring(img.value.lastIndexOf('.')+1);

var result=validExt.includes(img\_ext);

if(result==false){

alert("selected files is not an image...");

return false;

}

else{

if(parseFloat(img.files[0].size/(1024\*1024))>=3)

{

alert("Files size must be smaller than 3 mb");

}

}

}

else{

alert("no image is selected...");

return false;

}

}

</script>

</div>

</form>

<script src="assets/js/main.js"></script>

</body>

</head>

## **Property Details After Subscription**

<?php

session\_start();

$pid=$\_GET['pid'];

$conn=new mysqli("localhost","root","","art\_gallery");

if($conn->connect\_error)die("connection failed".$conn->connect\_error);

else

echo" ";

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

{

$sql = "SELECT \* from userplan WHERE current\_date < plant\_End\_Date and email\_id='{$em}' ";

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

$found=0;

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

{

while($row = mysqli\_fetch\_assoc($result))

{

$found++;

}

////////////////////////////////////////////////////////

if ($found==1)

{

$sql = "SELECT \* from product WHERE pid='$pid'";

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

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

{

while($row = mysqli\_fetch\_assoc($result))

{

$user\_id=$row["user\_id"];

$admin\_id=$row["admin\_id"];

$pname = $row["pname"];

$artist\_name = $row["artist\_name"];

$pyear = $row["pyear"];

$style = $row["style"];

$medium = $row["medium"];

$subject = $row["subject"];

$price = $row["price"];

$height = $row["height"];

$width = $row["width"];

$short\_desc = $row["short\_desc"];

$long\_desc = $row["long\_desc"];

$speciality = $row["speciality"];

$images = $row['image'];

$check=($row["user\_id"] != NULL);

if($check){

$seller=$row["user\_id"];

}

else {

$seller=$row["admin\_id"];

}

}

}

}

?>

<html>

<head>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!--========== BOX ICONS ==========-->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/boxicons@latest/css/boxicons.min.css">

<!--========== CSS ==========-->

<link rel="stylesheet" href="assets1/css/styles.css">

<title>Userdashboard</title>

<style>

/\* Basic Styling \*/

html, body {

height: 100%;

width: 100%;

margin: 0;

font-family: 'Roboto', sans-serif;

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 15px;

display: flex;

}

/\* Basic Styling \*/

.left-column {

width: 65%;

position: relative;

}

.right-column {

width: 40%;

margin-top: 60px;

}

.left-column img {

width: 85%;

position: absolute;

left: 0;

top: 0;

opacity: 0;

transition: all 0.3s ease;

}

.left-column img.active {

opacity: 1;

}

.product-description {

border-bottom: 1px solid #E1E8EE;

margin-bottom: 20px;

}

.product-description span {

font-size: 12px;

color: #358ED7;

letter-spacing: 1px;

text-transform: uppercase;

text-decoration: none;

}

.product-description h1 {

font-weight: 300;

font-size: 52px;

color: #43484D;

letter-spacing: -2px;

}

.product-description h1 {

font-weight: 200;

font-size: 45px;

color: #43484D;

letter-spacing: -2px;

}

.product-description p {

font-size: 16px;

font-weight: 300;

color: #86939E;

line-height: 24px;

}

.product-color {

margin-bottom: 30px;

}

.color-choose div {

display: inline-block;

}

.color-choose input[type=&amp;quot;radio&amp;quot;] {

display: none;

}

.color-choose input[type=&amp;quot;radio&amp;quot;] + label span {

display: inline-block;

width: 40px;

height: 40px;

margin: -1px 4px 0 0;

vertical-align: middle;

cursor: pointer;

border-radius: 50%;

}

.color-choose input[type=&amp;quot;radio&amp;quot;] + label span {

border: 2px solid #FFFFFF;

box-shadow: 0 1px 3px 0 rgba(0,0,0,0.33);

}

.color-choose input[type=&amp;quot;radio&amp;quot;]#red + label span {

background-color: #C91524;

}

.color-choose input[type=&amp;quot;radio&amp;quot;]#blue + label span {

background-color: #314780;

}

.color-choose input[type=&amp;quot;radio&amp;quot;]#black + label span {

background-color: #323232;

}

.color-choose input[type=&amp;quot;radio&amp;quot;]:checked + label span {

background-image: url(images/check-icn.svg);

background-repeat: no-repeat;

background-position: center;

}

.cable-choose {

margin-bottom: 20px;

}

.cable-choose button {

border: 2px solid #E1E8EE;

border-radius: 6px;

padding: 13px 20px;

font-size: 14px;

color: #5E6977;

background-color: #fff;

cursor: pointer;

transition: all .5s;

}

button{

border: 2px solid #E1E8EE;

border-radius: 6px;

padding: 13px 20px;

font-size: 14px;

color:white;

background-color:#CCEEFF;

cursor: pointer;

transition: all .5s;

}

.cable-choose button:hover,

.cable-choose button:active,

.cable-choose button:focus {

border: 2px solid #86939E;

outline: none;

}

.cable-config {

border-bottom: 1px solid #E1E8EE;

margin-bottom: 20px;

}

.cable-config a {

color: #358ED7;

text-decoration: none;

font-size: 12px;

position: relative;

margin: 10px 0;

display: inline-block;

}

.cable-config a:before {

content: &amp;quot;?&amp;quot;;

height: 15px;

width: 15px;

border-radius: 50%;

border: 2px solid rgba(53, 142, 215, 0.5);

display: inline-block;

text-align: center;

line-height: 16px;

opacity: 0.5;

margin-right: 5px;

.product-price {

display: flex;

align-items: center;

}

.product-price span {

font-size: 26px;

font-weight: 300;

color: #43474D;

margin-right: 20px;

}

.cart-btn {

display: inline-block;

background-color:white;

border-radius: 6px;

font-size: 16px;

color:white ;

text-decoration: none;

padding: 12px 30px;

transition: all .5s;

}

.cart-btn:hover {

background-color:blue;

}

</style>

</head>

<body>

<div class="nav" id="navbar">

<nav class="nav\_\_container">

<div>

<a href="userdashboard.php" class="nav\_\_link nav\_\_logo">

<i class='bx bxs-home-smile'></i>

<span class="nav\_\_logo-name">Home Genie</span>

</a>

<div class="nav\_\_list">

<div class="nav\_\_items">

<h3 class="nav\_\_subtitle">Profile</h3>

<a href="userdashboard.php" class="nav\_\_link active">

<i class='bx bx-home nav\_\_icon' ></i>

<span class="nav\_\_name">Home</span>

</a>

<div class="nav\_\_dropdown">

<a href="userviewprofile.php" class="nav\_\_link">

<i class='bx bx-user nav\_\_icon' ></i>

<span class="nav\_\_name">Profile</span>

<i class='bx bx-chevron-down nav\_\_icon nav\_\_dropdown-icon'></i>

</a>

<div class="nav\_\_dropdown-collapse">

<div class="nav\_\_dropdown-content">

<a href="userviewprofile.php" class="nav\_\_dropdown-item">View

Profile</a>

<a href="userupdateprofile.php" class="nav\_\_dropdown-item">Update

Profile</a>

</div>

</div>

</div>

<a href="feed.php" class="nav\_\_link">

<i class='bx bx-message-rounded nav\_\_icon' ></i>

<span class="nav\_\_name">Feedback</span>

</a>

</div>

<div class="nav\_\_items">

<h3 class="nav\_\_subtitle">Menu</h3>

<div class="nav\_\_dropdown">

<a href="#" class="nav\_\_link">

<i class='bx bx-bell nav\_\_icon' ></i>

<span class="nav\_\_name">Seller</span>

<i class='bx bx-chevron-down nav\_\_icon nav\_\_dropdown-icon'></i>

</a>

<div class="nav\_\_dropdown-collapse">

<div class="nav\_\_dropdown-content">

<a href="sellerprop.php" class="nav\_\_dropdown-item">Add

Property</a>

<a href="upsellertb.php" class="nav\_\_dropdown-

item">Update/Delete</a>

</div>

</div>

</div>

<a href="cart.php" class="nav\_\_link">

<i class='bx bx-library nav\_\_icon' ></i>

<span class="nav\_\_name">Property Listing</span>

</a>

<a href="myaccount.php" class="nav\_\_link">

<i class='bx bx-bookmark nav\_\_icon' ></i>

<span class="nav\_\_name">Interested</span>

</a>

<a href="usersideb1.php" class="nav\_\_link">

<i class='bx bx-bookmark nav\_\_icon' ></i>

<span class="nav\_\_name">News</span>

</a>

</div>

</div>

</div>

<a href="home.html" class="nav\_\_link nav\_\_logout">

<i class='bx bx-log-out nav\_\_icon' ></i>

<span class="nav\_\_name">Log Out</span>

</a>

</nav>

</div>

<main class="container">

<!-- Left Column / Headphones Image -->

<div class="left-column">

<img src="<?php echo "http://localhost/".$images?>"class="active" alt="">

</div>

<!-- Right Column -->

<div class="right-column">

<!-- Product Description -->

<div class="product-description">

<span>Property details</span>

<h1>Name: <?php echo $pname?></h1>

<h2>Artist Name: <?php echo $artist\_name?></h2>

<h3>Seller: <?php echo $seller?></h3>

<p>Year:<?php echo $pyear?></p>

<p>Style:<?php echo $style?></p>

<p>Medium:<?php echo $medium?></p>

<p>Subject:<?php echo $subject?></p>

<!--<p>Price:<?php echo $price?></p>-->

<p>Height<?php echo $height?></p>

<p>Width:<?php echo $width?></p>

<p>Short\_desc:<?php echo $short\_desc?></p>

<p>Long\_desc:<?php echo $long\_desc?></p>

<p>Speciality:<?php echo $speciality?></p>

<h2>Cost : <?php echo $price;

}

else

{

echo '<script>alert("Sorry ,You Dont have subscription");window.location="cart.php";</script>';

}

}

?

</div>

<!-- Product Pricing -->

<div class="product-price">

</div>

</div>

</main>

<script src="assets1/js/main.js"></script>

<script>

$(document).ready(function() {

$('.color-choose input').on('click', function() {

var headphonesColor = $(this).attr('data-image');

$('.active').removeClass('active');

$('.left-column img[data-image = ' + headphonesColor + ']').addClass('active');

$(this).addClass('active');

});

});

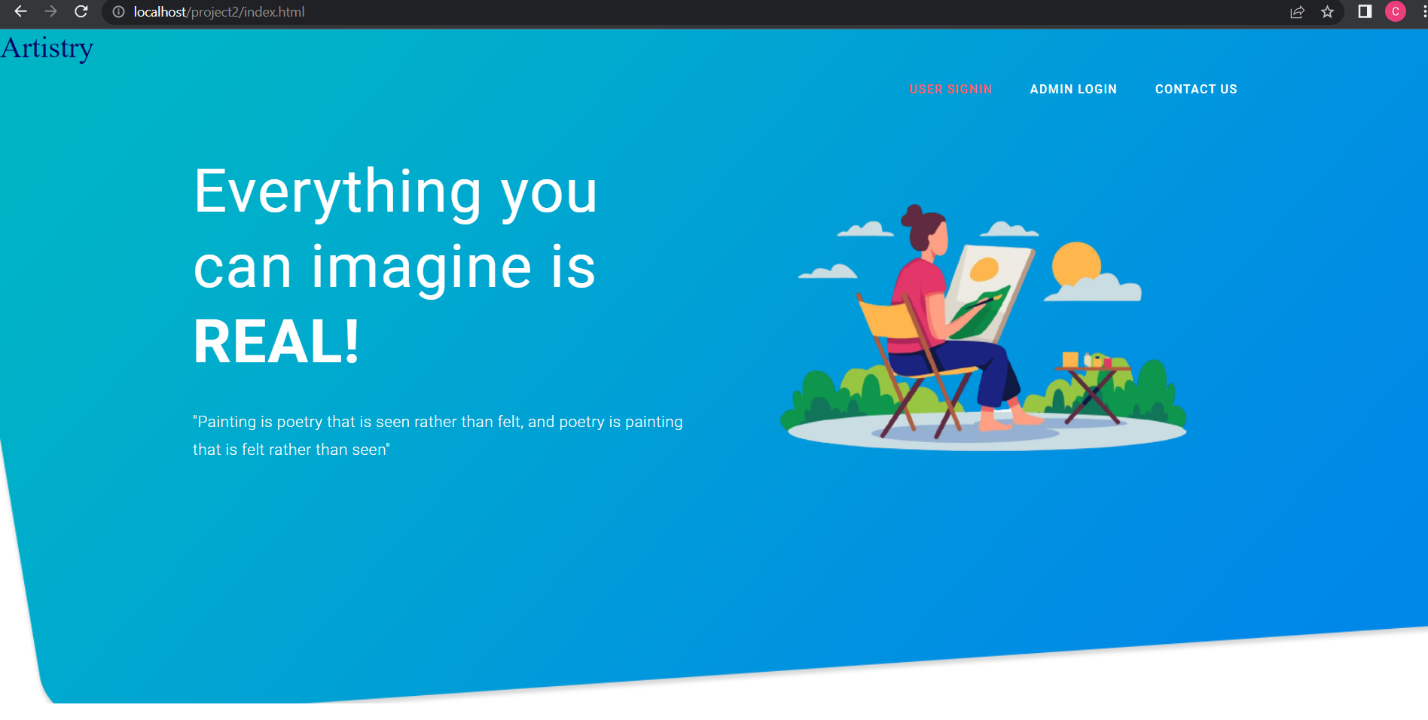
</script>

</body>

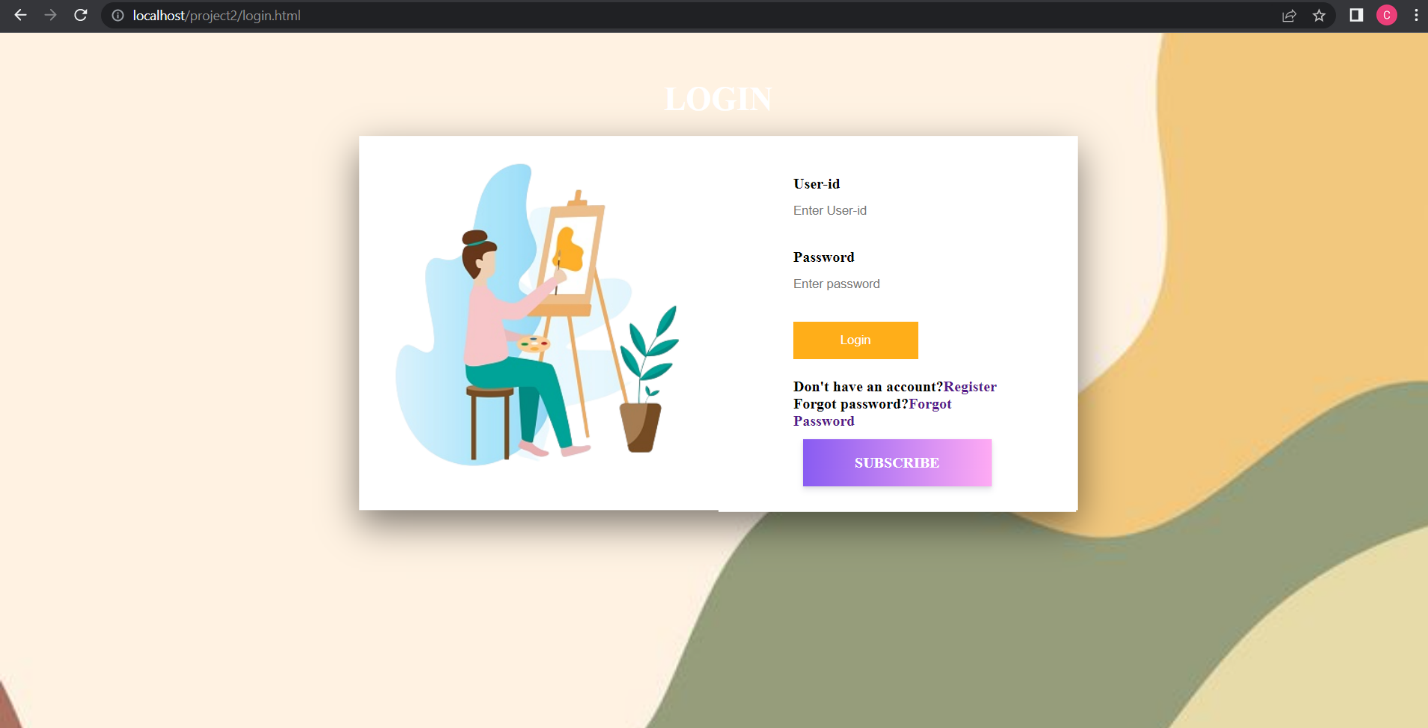
</html>

**6. SCREENSHOTS**

**HOME**

****

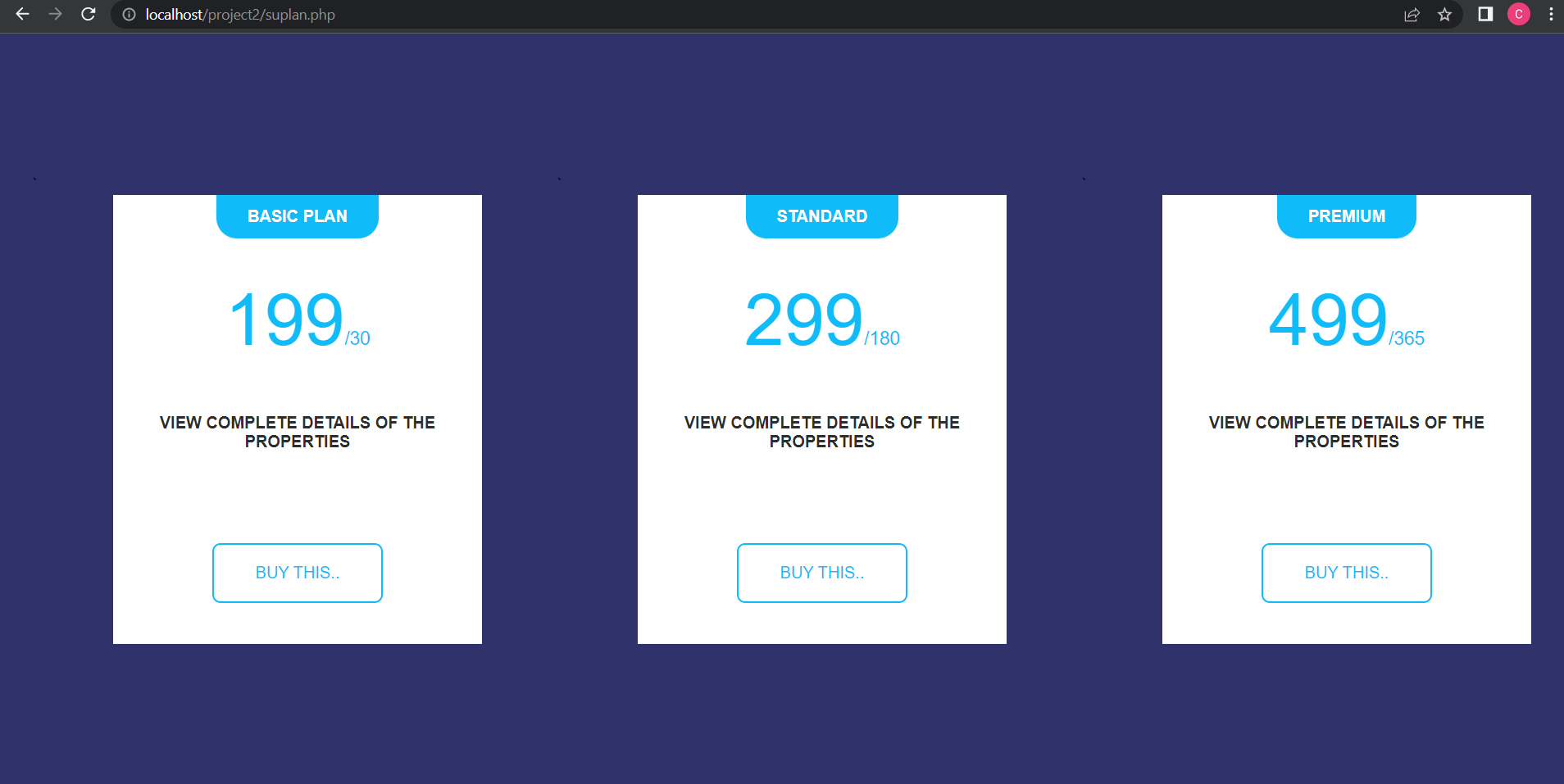
**USER LOGIN**

****

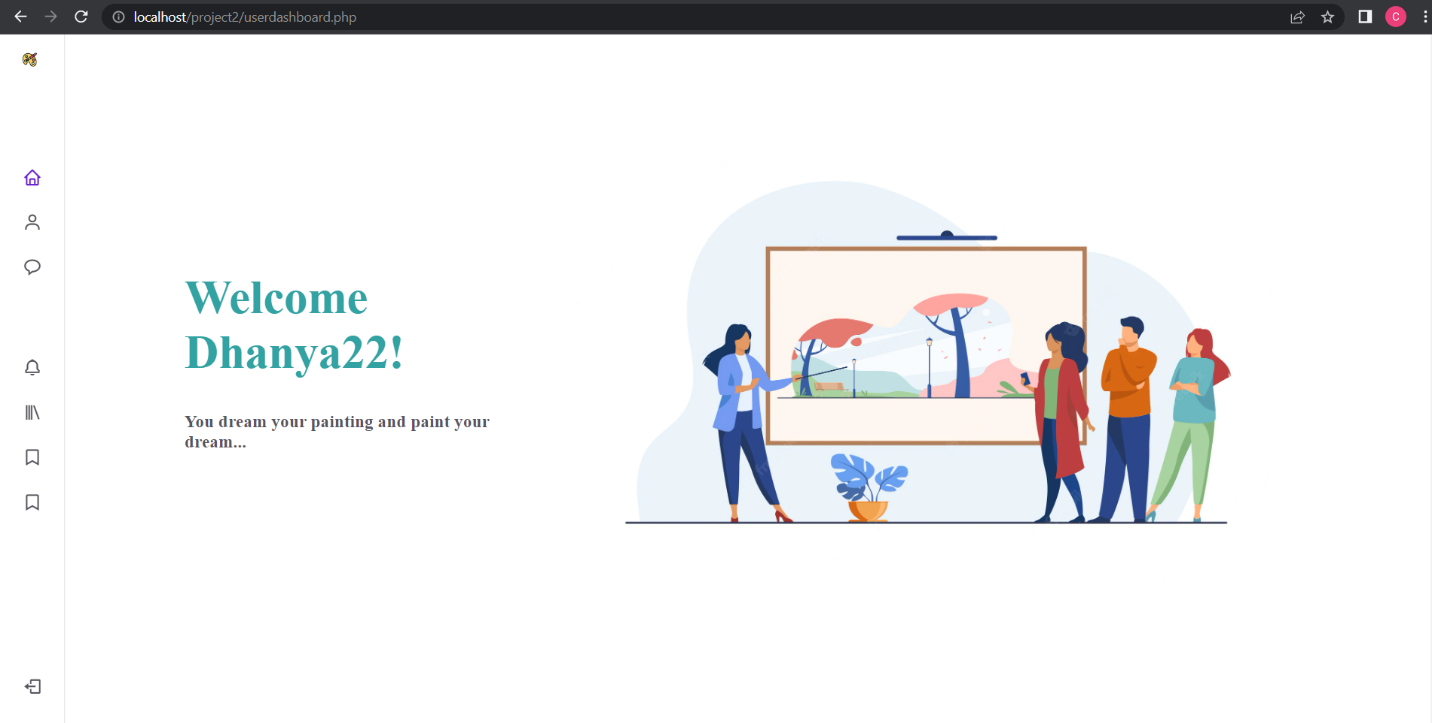
**USER REGISTER**

****

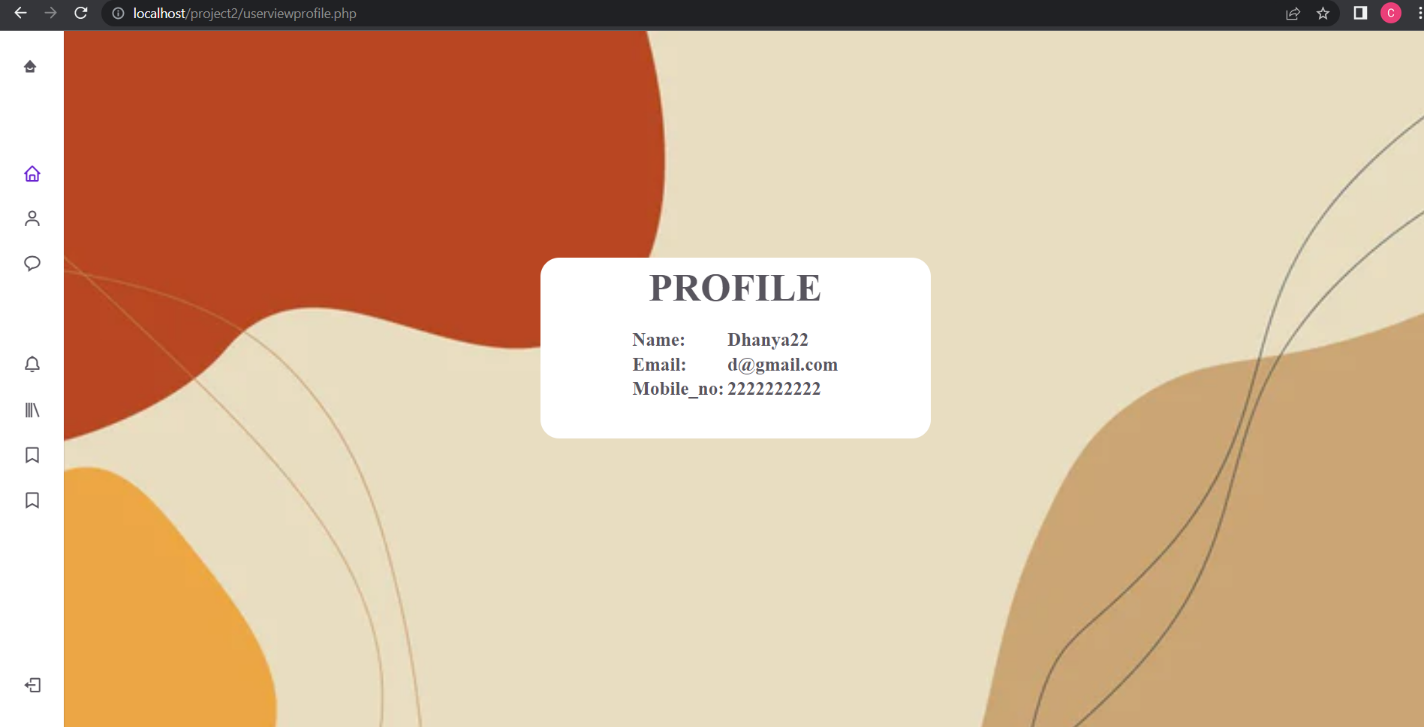
**USER PLAN**

****

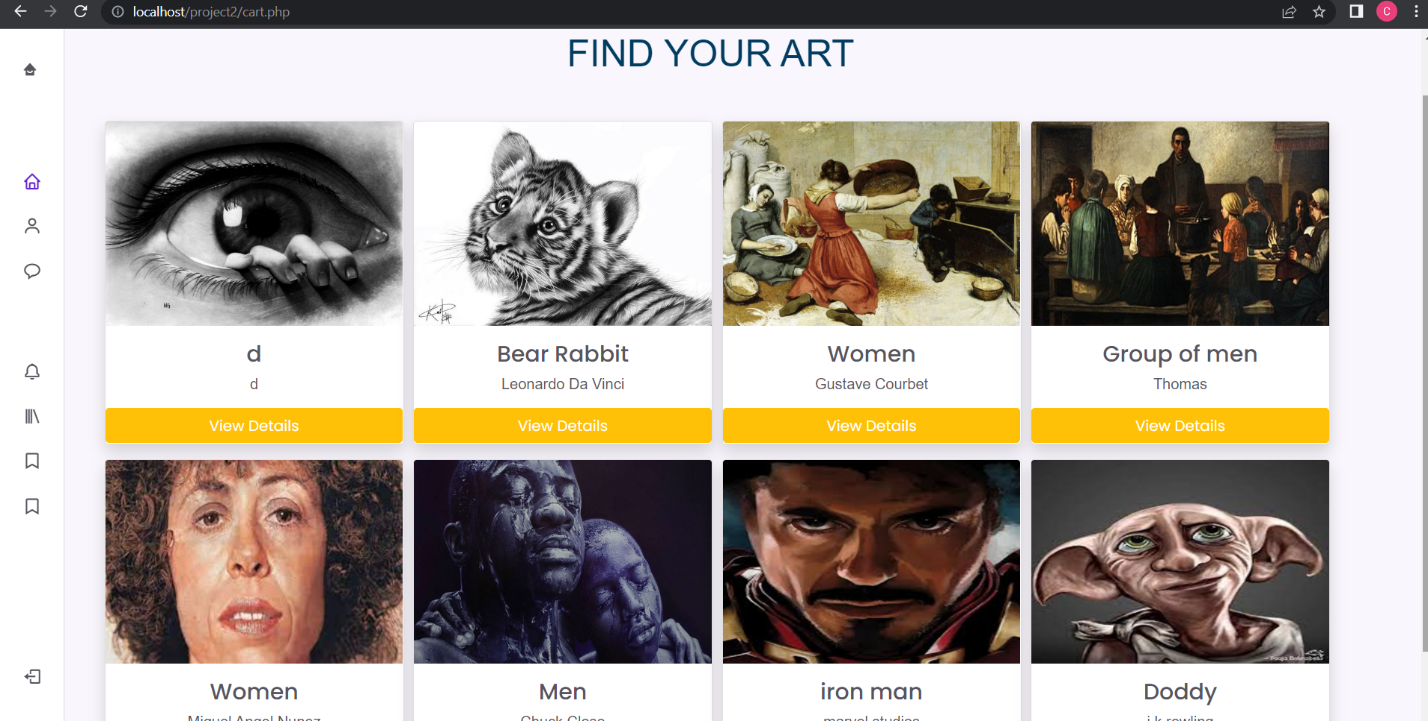
**USER DASHBOARD**

****

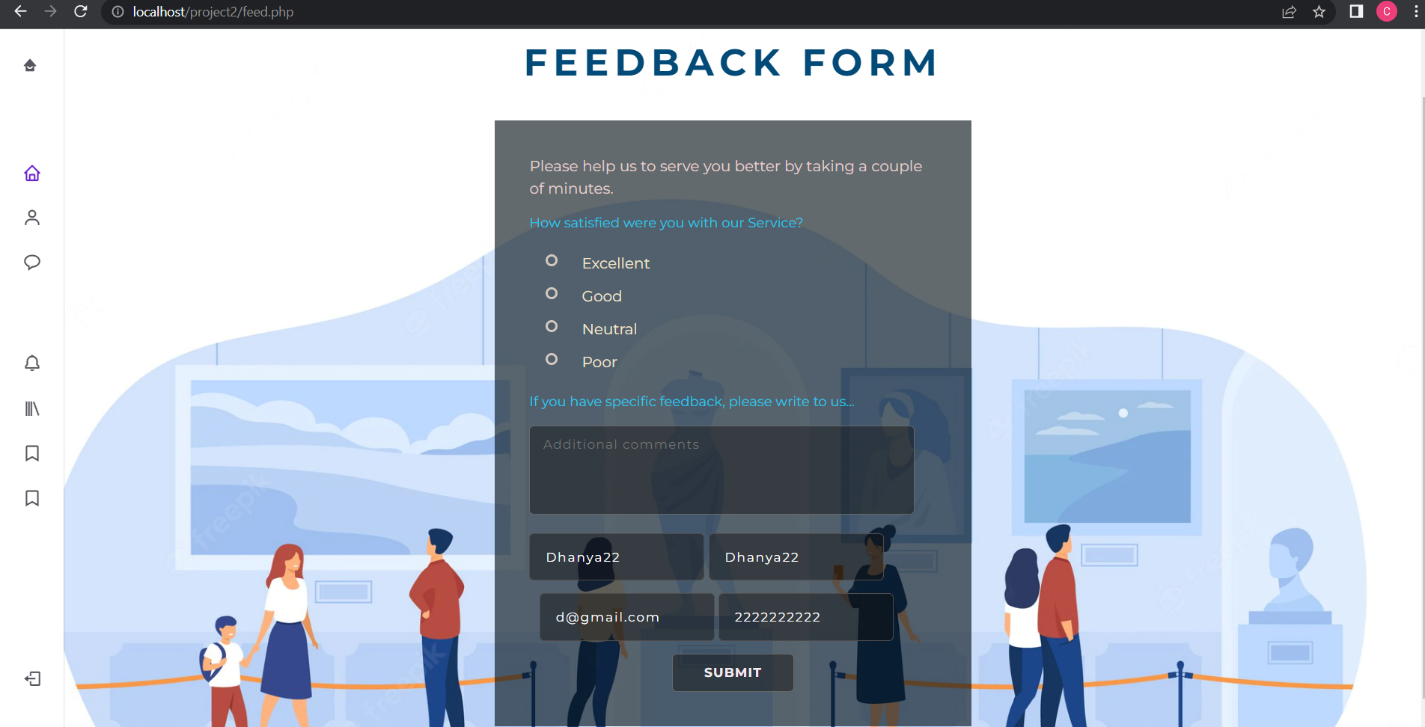
**USER PROFILE**

****

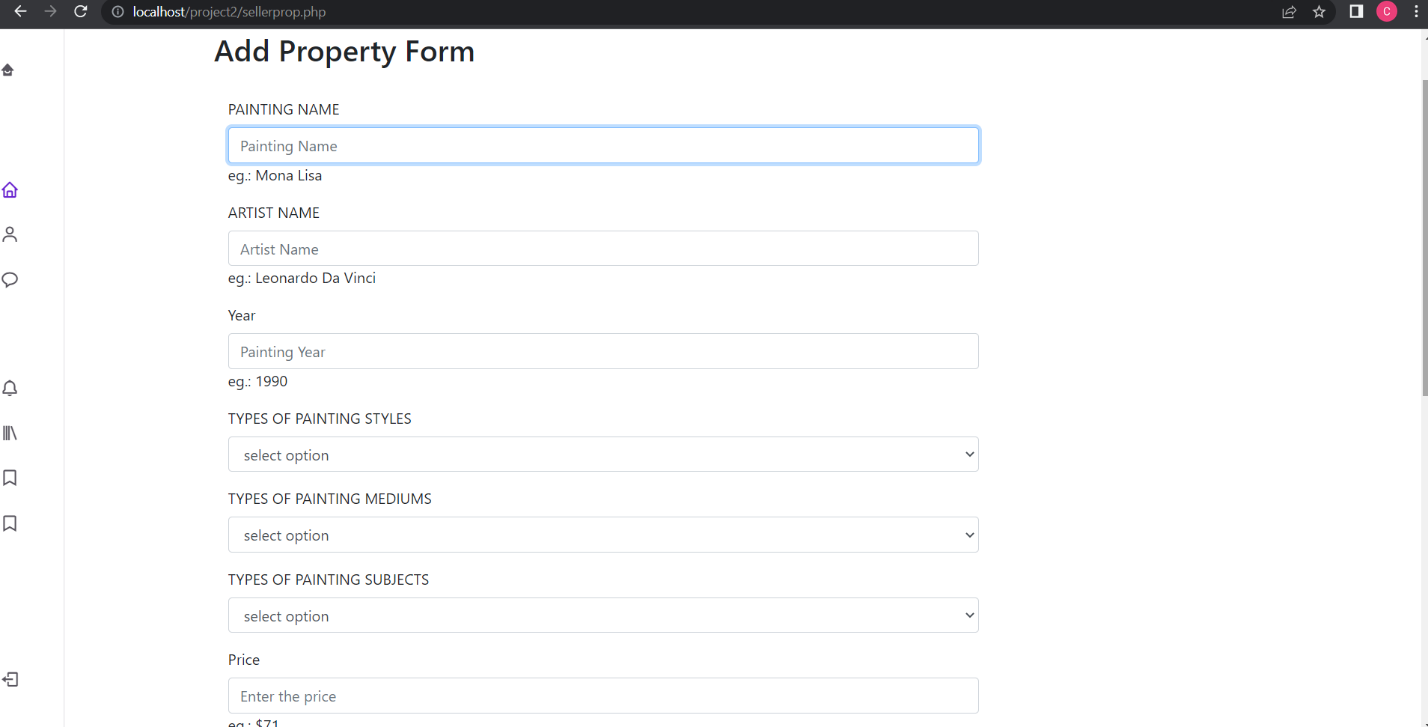
**VIEW PRODUCT**

****

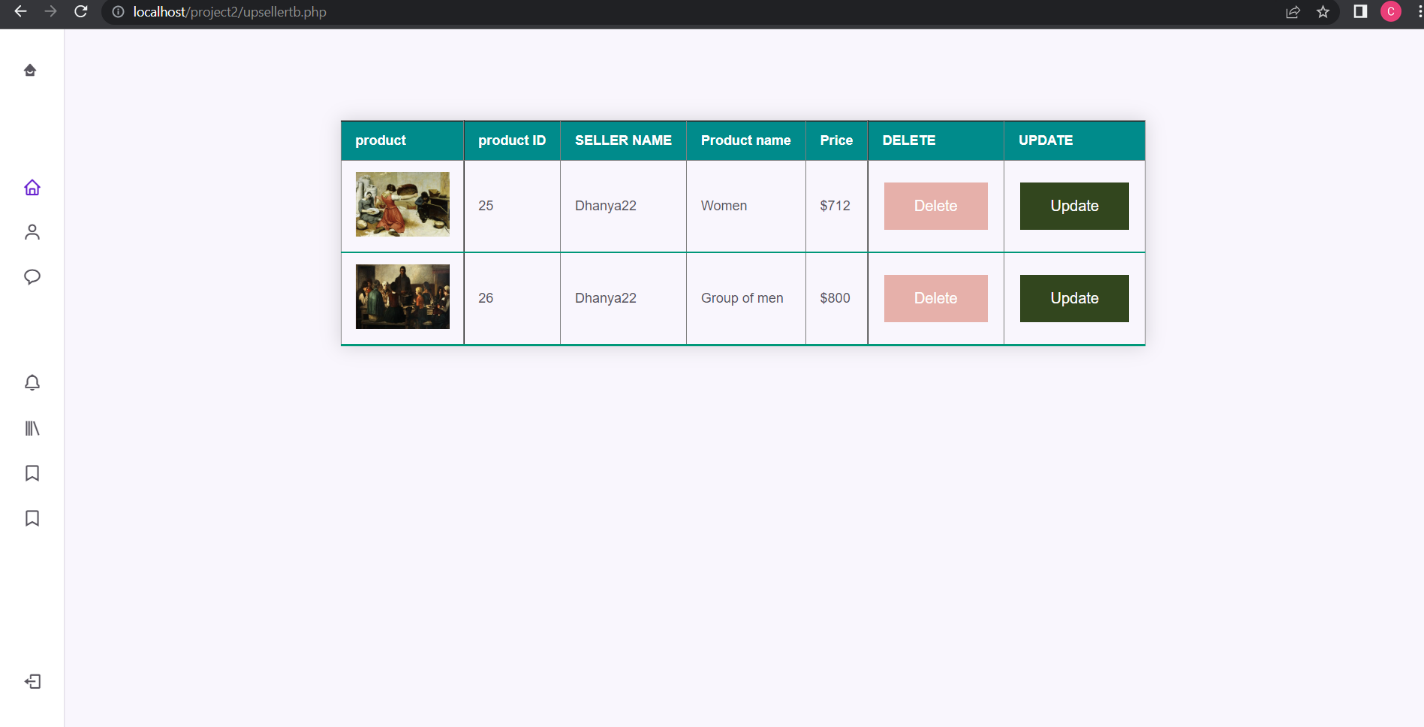
**FEEDBACK**

****

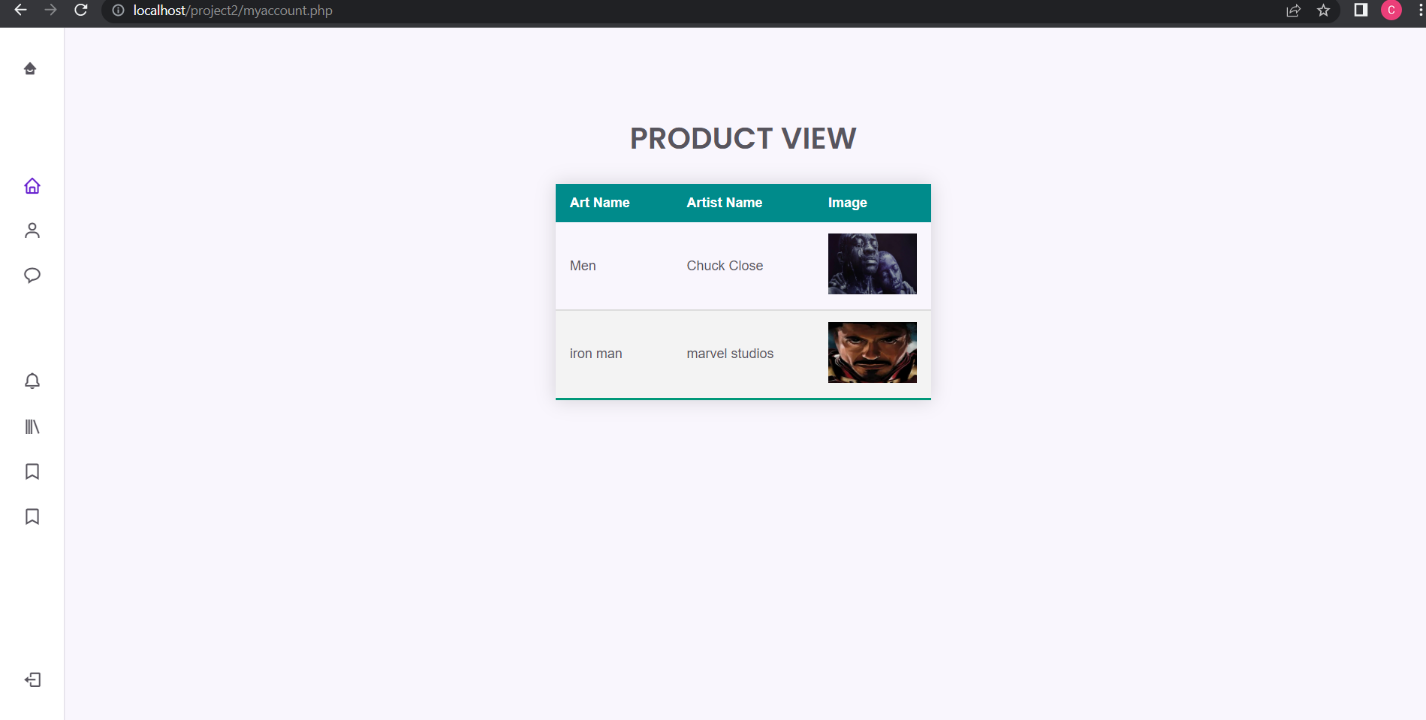
**USER PRODUCT ADD**

****

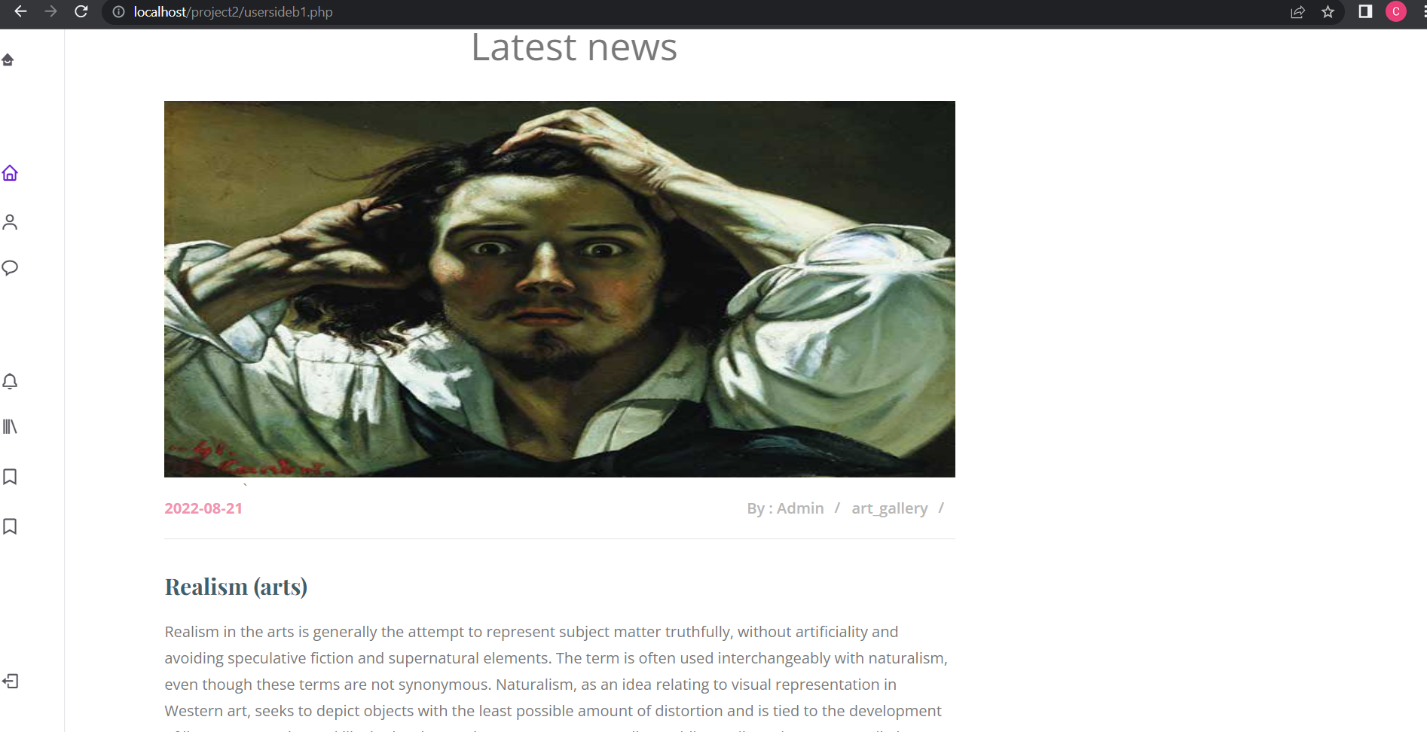
**USER PRODUCT UPDATE**

****

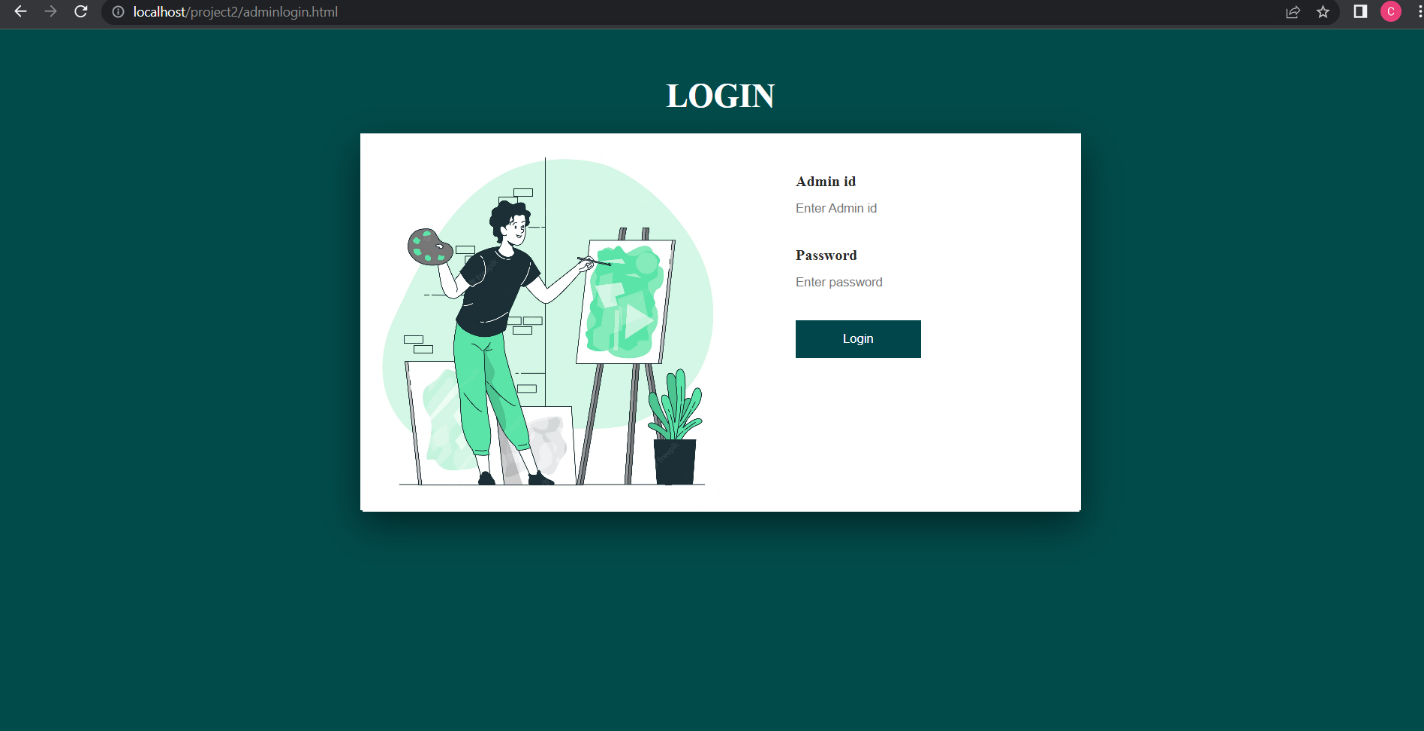
**USER INTERESTED**

****

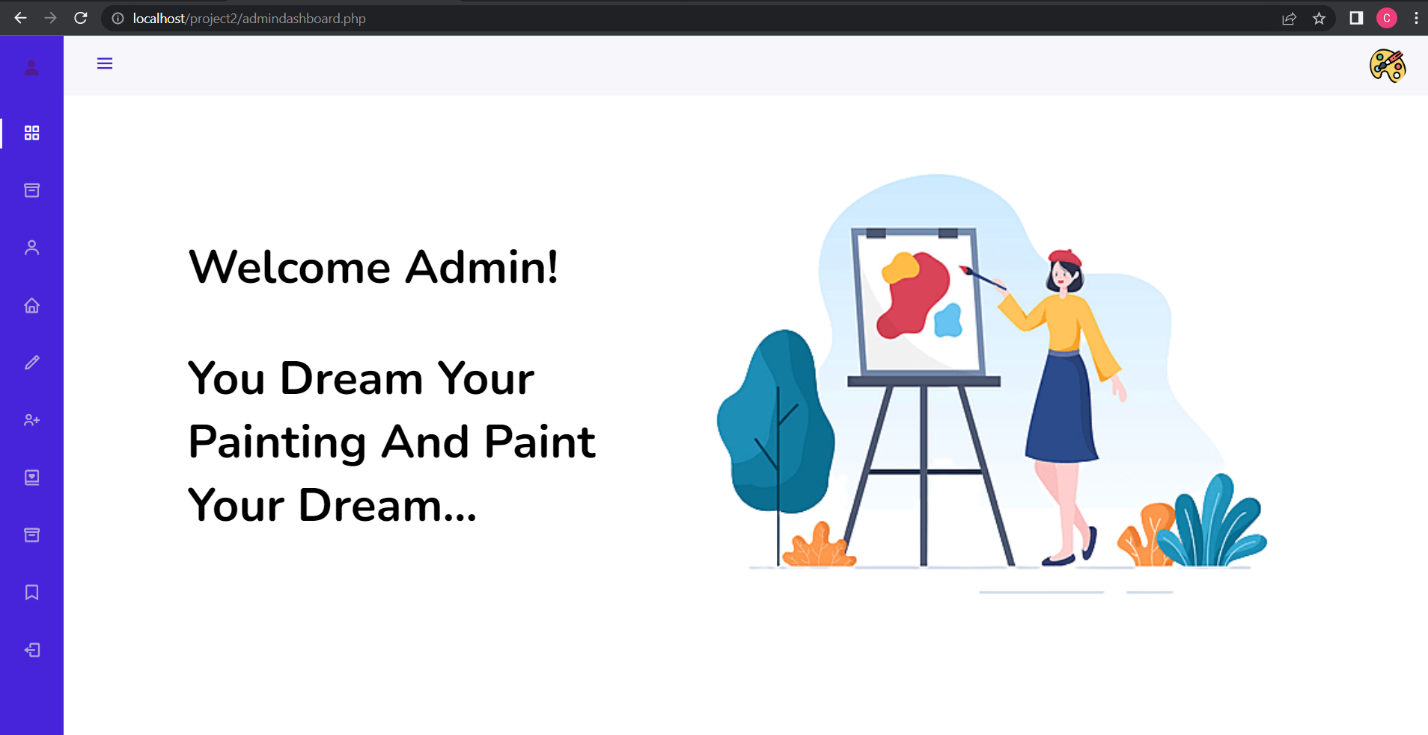
**USER VIEW NEWS**

****

**ADMIN LOGIN**

****

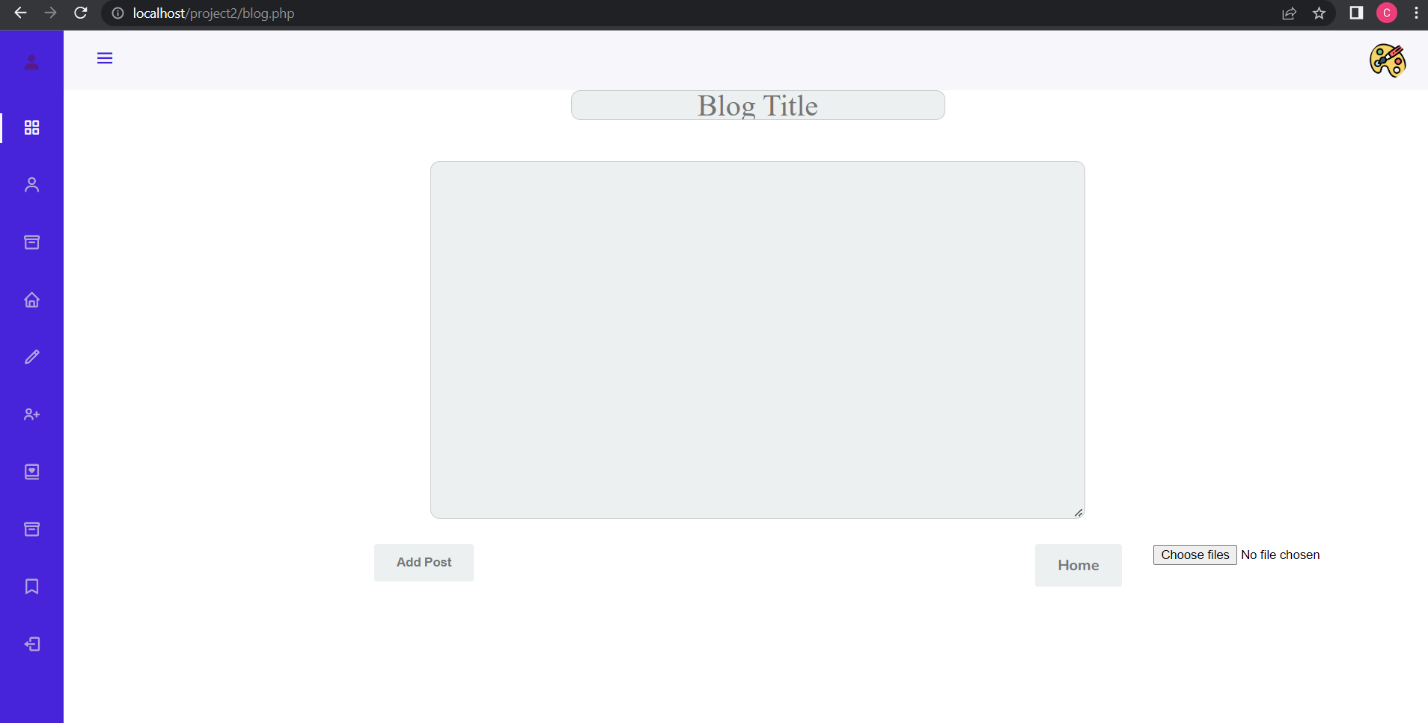
**ADMIN DASHBOARD**

****

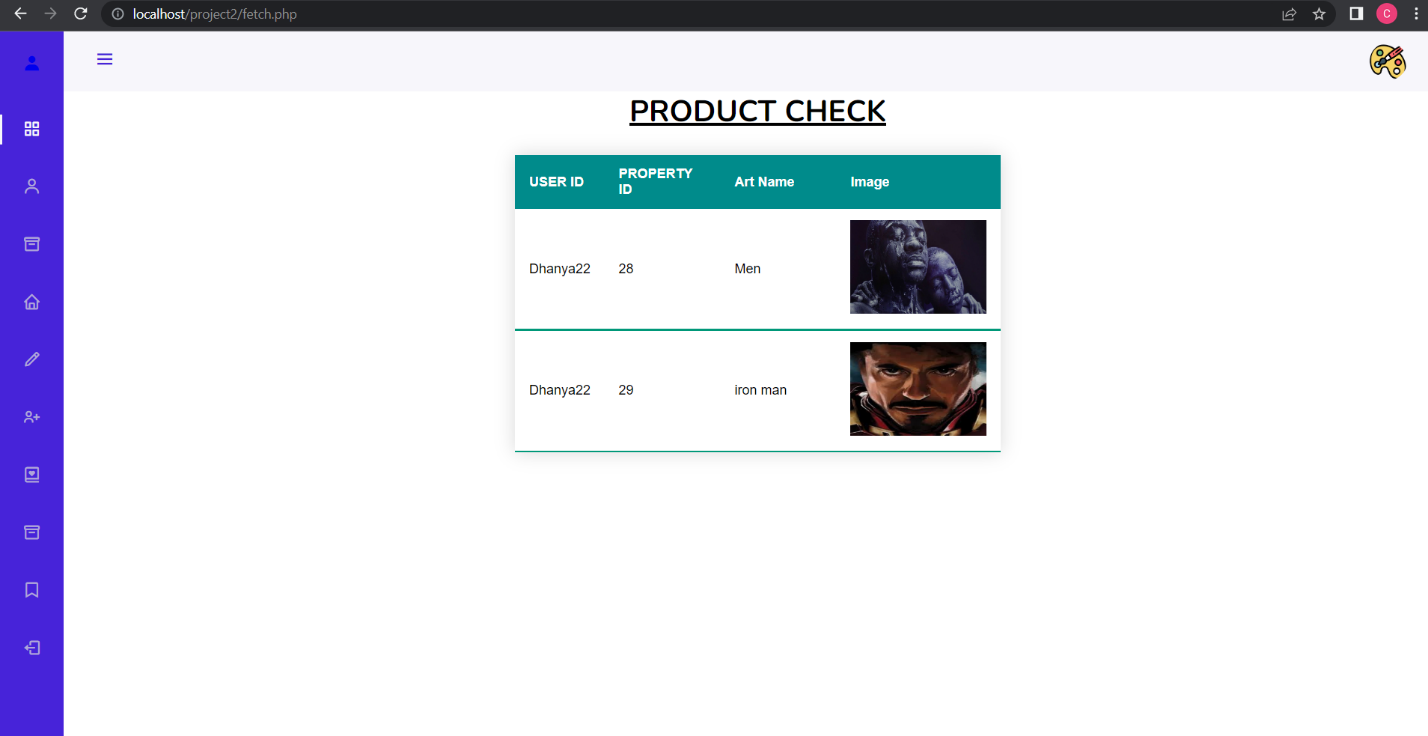
**ADD ADMIN**

****

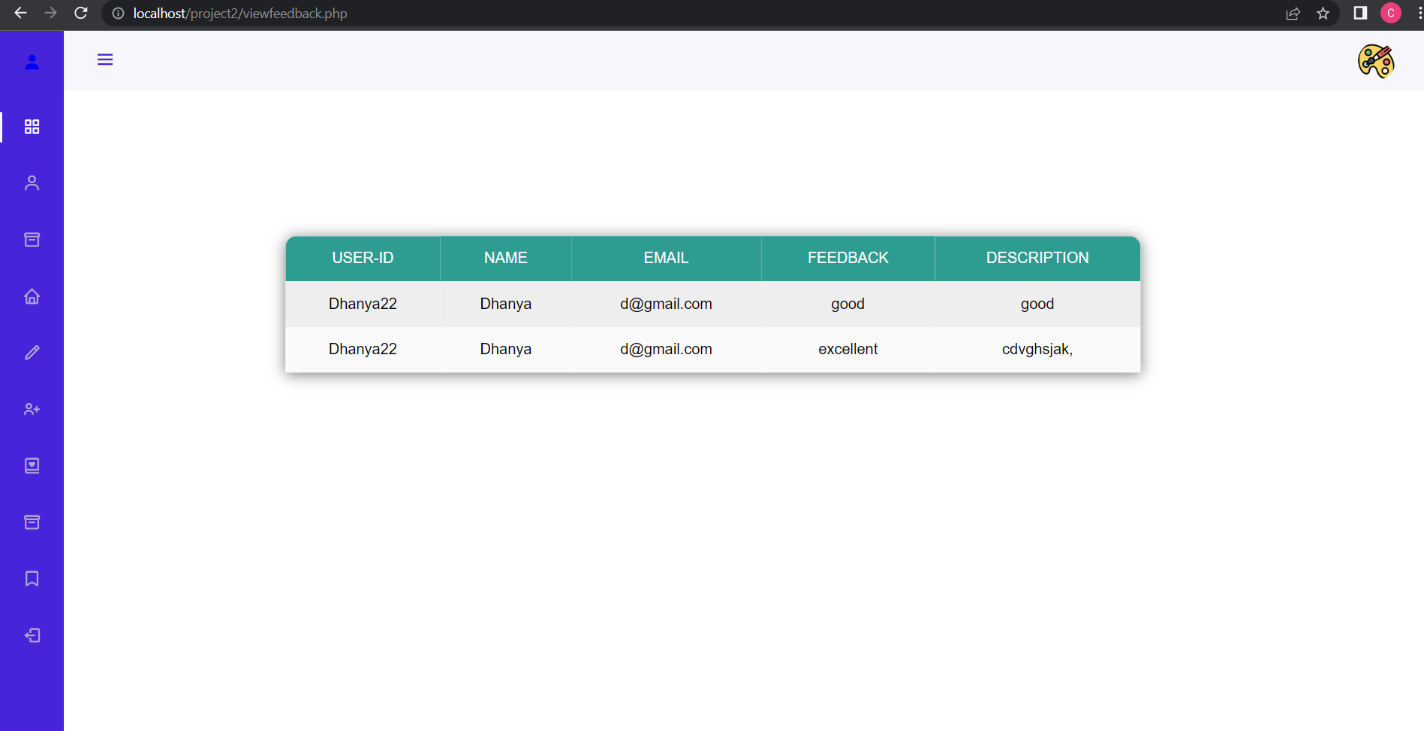
**ADD BLOG**

****

**ADMIN VIEW USER INTEREST**

****

**ADMIN VIEW USER FEEDBACK**

****

**7. CONCLUSION**

Our application Art Gallery Management System helps user to see the Product added by the admin accordingly, which helps the user to know the details of the product. Admin can also view user’s Interested property. The User can also show his/her interest regarding the product. Even the user also given the privilege to add the product which helps other user’s to see the product added by the user. Through Subscription the user knows the details regarding the product like Seller’s Name of the product, Cost of the Product, Size of the Product.

**8. FUTURE ENHANCEMENT**

It is not possible to develop a system that makes requirements of the user. User requirements keeps changing as the system is being used.

Some of the future enhancements that can be done to this system are :

* As the technology emerges, it is possible to upgrade the system and can be adoptable to desired environment.
* Based on the future security issues, security can be improved using emergent

technologies.

* We can add direct buying option to the users.
* Predicting user interest and personalized recommendation can be done.

**9. BIBILOGRAPHY**

**HTML:**

* https://www.w3schools.com/html/html5\_video.asp
* https://www.w3schools.com/html/html\_tables.asp
* https://www.w3schools.com/html/html\_colors.asp
* https://www.w3schools.com/html/html\_images.asp
* https://www.w3schools.com/html/html\_classes.asp

**CSS:**

* https://www.w3schools.com/css/css\_navbar.asp
* https://www.w3schools.com/css/css3\_colors.asp
* <https://www.w3schools.com/css/css_font.asp>
* <https://www.w3schools.com/howto/howto_css_dropdown.asp>
* <https://www.w3schools.com/howto/howto_css_images_side_by_side.asp>

**PHP:**

* https://www.w3schools.com/php/php\_mysql\_connect.asp
* <https://www.w3schools.com/php/php_mysql_create.asp>
* https://www.w3schools.com/php/php\_mysql\_create\_table.asp

**JAVASCRIPT:**

* https://www.w3schools.com/howto/howto\_js\_slideshow.asp
* https://sweetalert.js.org/guides/
* <https://www.w3schools.com/js/js_functions.asp>