

PROFESSIONAL TRAINING REPORT - II

entitled

ONLINE ART GALLERY

Submitted in partial fulfillment of the requirements for the award of Bachelor
of Engineering degree in Computer Science and Engineering with
specialization in Blockchain Technology

by

T.V. NITHIN REDDY [Reg.No:41613020]

SHABAREESH PEMMARAJU [Reg..No:41613015]



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY) CATEGORY -1 UNIVERSITY BY UGC

Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **T.V.Nithin Reddy(41613020)** who carried out the project entitled "**Online Art Gallery**" under my supervision from January 2024 to April 2024.

Internal Guide

Ms. Lekshmi S Raveendran M.Tech .,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voice Examination held on

7/5/24

Internal Examiner

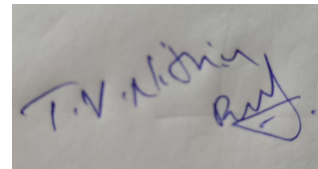
External Examiner

DECLARATION

I, **T.V.Nithin Reddy(Reg.No:41613020)**, hereby declare that the Professional Training Report entitled “**Online Art Gallery using Full Stack Web Development** ” done by me under the guidance of **Ms Lekshmi S Raveendran M.Tech.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Blockchain Technology.

DATE: 07-05-2024

PLACE: Chennai

A photograph of a handwritten signature in blue ink on a light-colored surface. The signature appears to read 'T.V. Nithin Reddy' with a stylized flourish at the end.

SIGNATURE OF THE CANDIDATE :

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal Guide **Ms Lekshmi S Raveendran M.Tech.**, for his/her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-2 Professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

COURSE CERTIFICATE



Certificate of Completion

Is Awarded to

Venkata Nithin Reddy Tati Reddy

Upon successfully completed the Bootcamp Training on
FULL STACK DEVELOPMENT for 40 hrs with a Mini Project in Online Art Gallery
from 11-Mar -2024 to 05-Apr -2024



Mr. Nikhil Barshikar
Managing Director
IMARTICUS LEARNING

ABSTRACT

An online art gallery is a digital platform that transcends geographical boundaries, providing art enthusiasts with a seamless experience to explore, appreciate, and acquire artworks. This virtual space combines the convenience of e-commerce features, such as "add to cart" functionality, with the security and personalization of a user login page. In an online art gallery, visitors can browse through a diverse collection of artworks, ranging from paintings and sculptures to digital art and photography. The "add to cart" feature allows potential buyers to curate their own collections by selecting and storing artworks of interest, creating a personalized art shopping experience. To enhance security and user engagement, a login page is implemented, ensuring that users' personal information and preferences are safeguarded. Registered users can track their favorite artists, receive updates on new arrivals, and access exclusive content. This login system fosters a sense of belonging within the art community, enabling artists and art enthusiasts to connect, communicate, and collaborate. On the front-end, the art gallery utilizes a user-friendly interface built with modern frameworks (e.g., React, Angular) and incorporates responsive design principles to provide optimal viewing across various devices. Visitors can explore a curated collection of artworks through intuitive navigation features, allowing for filtering by artist, medium, style, or price range. Detailed information about each artwork, including high-quality images, descriptions, and artist biographies, empowers informed decision-making. Additionally, the platform integrates virtual exhibition capabilities, where visitors can embark on immersive experiences by navigating through 3D galleries or utilizing high-resolution zoom functions for intricate details. The back-end infrastructure ensures smooth operation and secure data storage. A robust database management system (e.g., MySQL, PostgreSQL) serves as the foundation, storing information on artists, artworks, user accounts, and transaction records. A secure back-end API (Application Programming Interface) facilitates communication between the front-end application and the database, handling user requests for data retrieval, modification, and creation. The chosen back-end framework (e.g., Django, Node.js) provides efficient processing capabilities and ensures secure data exchange. To facilitate art sales, the platform integrates a secure payment gateway, allowing for seamless online transactions using trusted payment processors (e.g., Stripe, PayPal).

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	vi
	LIST OF FIGURES	viii
1	INTRODUCTION	9
	1.1 Overview	
2	LITERATURE SURVEY	13
	2.1 survey	
3	REQUIREMENTS ANALYSIS	
	3.1 Objective	19
	3.2 3.2.1 Hardware Requirements	21
	3.2.2 Software Requirements	
4	DESIGN DESCRIPTION OF PROPOSED PRODUCT	23
	4.1.1 Ideation Map/Architecture Diagram	25
	4.1.2 Various stages	27
	4.1.3 Internal or Component design structure	28
	4.1.4 working principles	29
	4.2 4.2.1 Novelty of the Project	35
5	CONCLUSION	37
	REFERENCES	39
	APPENDIX	
	A. Research Paper	35
	B. Source Code	37

LIST OF FIGURES

Figure No.	Figure Name	Page
1	Working model/Architecture	25
2	Component Design	28
3	Open Xampp	30
4	Import Database to PHP	31
5	Register	31
6	Login to the page	31
7	Explore Admin Page	32
8	Explore user page	32
9	Explore Arts	33

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

This project will be an online art gallery built using the full stack web development approach. This means we'll develop both the front-end (user interface) and the back-end (server-side logic and database) of the website. The front-end will be designed to provide a visually appealing and user-friendly experience for art enthusiasts to browse a curated collection of artwork. They can filter by artist, style, medium, and other criteria. Detailed information about each piece, including title, description, dimensions, and price, will be readily available. Additionally, the frontend might incorporate features like high-quality image zooming and virtual tours of exhibitions.

On the back-end, a secure database will store information about the artwork, artists, and users. The system will handle user registration and login, allowing them to create profiles, save favourite pieces, and potentially even follow their favorite artists. For artists, the back-end will provide a portal to upload their work, manage their profiles, and potentially track sales or inquiries. The full stack approach ensures a cohesive and feature-rich online art gallery experience.

The Online Art Gallery project is a dynamic and ambitious initiative aimed at bringing the world of art closer to art enthusiasts, collectors, and artists themselves. By leveraging the power of the internet and cutting-edge technology, we aspire to create a virtual space where art transcends geographical boundaries, making it accessible to a diverse and global audience.

In today's digital age, the world of art is no longer confined to physical galleries and exhibition spaces. With the advent of technology, art enthusiasts and collectors can now explore and appreciate artworks from the comfort of their own homes through

online art galleries. These virtual platforms not only democratize access to art but also provide artists with a global platform to showcase their creations.

Our online art gallery project aims to bridge the gap between artists and art lovers by creating a dynamic and immersive digital space where creativity knows no bounds. Leveraging full-stack development, we are crafting a seamless and intuitive user experience that seamlessly integrates front-end and back-end technologies to bring the beauty of art to life on the web.

At the heart of our project lies a visually stunning front-end interface designed to captivate users from the moment they enter the virtual gallery. Through carefully curated layouts, high-resolution imagery, and interactive features, visitors will be transported into a world where artistry knows no bounds. Whether they're browsing through paintings, sculptures, photography, or digital art, every click will be a journey of discovery and inspiration.

But beauty is not just skin deep. Behind the scenes, our full-stack development team is working tirelessly to ensure that the online art gallery is not just a pretty facade but a robust and scalable platform capable of handling the complexities of art curation, artist management, and user interaction. From responsive web design to efficient database management, we're employing a range of technologies including HTML, CSS, JavaScript, and frameworks like React for the front end, and Node.js, Express, and MongoDB for the back end, to deliver a seamless and secure experience for both artists and visitors alike.

One of the key features of our online art gallery is its emphasis on fostering community engagement and collaboration. Through features such as artist profiles, virtual exhibitions, and interactive forums, we're creating a vibrant online ecosystem where artists can connect with fellow creators, showcase their portfolios, and receive feedback from a global audience. Additionally, our platform will empower users to curate their own virtual collections, share their favorite artworks on social media, and even purchase pieces directly from the artists, thereby fostering a sense of ownership and support within the artistic community.

In essence, our online art gallery project is more than just a website; it's a testament to the transformative power of technology in the realm of art and culture. By harnessing the latest advancements in full-stack development, we're not only reimagining the way art is discovered, shared, and appreciated but also democratizing access to creativity in ways never before possible. So whether you're an aspiring artist looking to showcase your talents or an art lover in search of your next masterpiece, we invite you to join us on this journey as we redefine the boundaries of the digital art world, one pixel at a time.

In an era where technology pervades every aspect of our lives, the intersection of art and the digital realm presents an unprecedented opportunity to revolutionize how we appreciate and engage with artistic expressions. The Online Art Gallery Project embodies this ethos, aiming to create a dynamic platform where art enthusiasts, collectors, and creators converge in a virtual space teeming with creativity and innovation.

At the heart of this endeavor lies the vision to transcend physical boundaries and democratize access to art. By harnessing the power of full-stack development, our team endeavors to craft an immersive online experience that transcends the limitations of traditional brick-and-mortar galleries. Through meticulous design and cutting-edge technology, we aspire to replicate the essence of wandering through a gallery hall, where each click leads to a discovery, each artwork tells a story, and each interaction sparks inspiration.

The Online Art Gallery Project isn't merely about showcasing art; it's about fostering a vibrant community of artists and art enthusiasts. Through robust backend infrastructure and seamless frontend interfaces, we aim to provide artists with a platform to showcase their creations to a global audience, irrespective of geographical constraints. Whether emerging talents seeking exposure or established artists expanding their reach, our platform offers a level playing field where creativity reigns supreme.

Central to our ethos is the belief in the transformative power of art to evoke emotions, provoke thoughts, and spark conversations. Thus, the Online Art Gallery Project isn't just a passive viewing experience; it's an interactive journey where users can engage with artworks through multimedia elements, participate in discussions, and even connect with artists in real-time. By transcending the static nature of traditional galleries, we aim to foster dynamic dialogues and forge meaningful connections between creators and connoisseurs.

In a world grappling with challenges of accessibility and inclusivity, our project strives to break down barriers and make art accessible to all. Leveraging state-of-the-art technologies such as responsive design and cross-platform compatibility, we ensure that our platform is accessible to users across devices and demographics. Whether accessed through a desktop computer, a tablet, or a smartphone, the Online Art Gallery Project beckons users into a world where art knows no bounds.

Moreover, our commitment to innovation extends beyond the digital realm, as we explore avenues to enhance the art-viewing experience through augmented reality (AR) and virtual reality (VR) technologies. By blending the physical and virtual worlds, we aim to provide users with immersive experiences that transcend the confines of traditional gallery spaces. Through AR-enabled exhibitions and VR-powered gallery tours, we invite users to embark on an experiential journey through the realms of imagination and creativity.

In conclusion, the Online Art Gallery Project represents a bold step towards redefining how we perceive, interact with, and appreciate art in the digital age. Through a synergy of technology, creativity, and community, we aspire to create a platform that not only showcases artistic brilliance but also fosters a deeper understanding and appreciation of diverse cultural expressions.

CHAPTER 2

LITERATURE REVIEW

2.1 SURVEY

A literature survey for an online art gallery project utilizing a full stack approach involves exploring existing research, frameworks, methodologies, and technologies pertinent to web development, particularly focusing on HTML, CSS, JavaScript, PHP, and XAMPP for backend support. Here's an extensive overview:

HTML, CSS, and JavaScript form the backbone of modern web development, providing the structure, styling, and interactivity necessary for user interfaces. Numerous studies delve into best practices, accessibility standards, and optimization techniques for these foundational technologies. For instance, research by Le Page et al. (2019) emphasizes the importance of responsive web design using HTML5 and CSS3 for enhanced user experience across devices. Additionally, studies like those by Meyer (2018) explore JavaScript frameworks such as React and Vue.js for dynamic content rendering and client-side interactivity.[1]

PHP, a server-side scripting language, is widely used for web development due to its versatility and compatibility with various databases. Literature on PHP often focuses on backend architecture, security considerations, and performance optimization. Research by Sipior et al. (2017) discusses PHP security vulnerabilities and mitigation strategies, highlighting the significance of secure coding practices in web applications. Moreover, studies by Gupta et al. (2020) delve into PHP frameworks like Laravel and CodeIgniter for streamlined development and maintainability of complex web projects.[2]

XAMPP, a cross-platform web server solution stack, integrates Apache, MySQL, PHP, and Perl, facilitating local development environments and deployment setups. Literature on XAMPP primarily revolves around its installation procedures, configuration options, and usage scenarios. Resources like the official documentation and user forums provide insights into troubleshooting common issues and optimizing server performance. Additionally, research by Li et al. (2018) explores XAMPP-based development

environments for educational purposes, demonstrating its effectiveness in teaching web development concepts in academic settings.[3]

In the context of online art galleries, specific literature may focus on the integration of multimedia content, user authentication, and e-commerce functionalities. Research by Martino et al. (2019) investigates the implementation of image galleries using HTML5 canvas and JavaScript libraries for image manipulation and presentation. Similarly, studies by Kim et al. (2020) examine the role of CSS Grid and Flexbox in designing visually appealing layouts for art showcase platforms.[4]

Moreover, literature on user experience (UX) design and human-computer interaction (HCI) provides valuable insights into optimizing the usability and accessibility of online art galleries. Research by Norman (2013) discusses principles of user-centered design, emphasizing the importance of intuitive navigation and clear communication in web interfaces. Additionally, studies by Nielsen (2021) explore usability heuristics and evaluation methods for assessing the effectiveness of web applications in meeting user needs and preferences.[5]

Furthermore, research on backend development encompasses topics such as database management, server-side scripting, and RESTful API design. Literature by Connolly and Begg (2014) offers comprehensive coverage of database systems and SQL querying techniques, essential for storing and retrieving artwork information in online galleries. Additionally, studies by Fielding (2000) outline principles of RESTful architecture, guiding the design of scalable and interoperable backend systems for web-based applications.[6]

In summary, a literature survey for an online art gallery project utilizing a full stack approach involves synthesizing existing research on HTML, CSS, JavaScript, PHP, and XAMPP, along with relevant topics such as UX design, backend development, and multimedia content management. By drawing upon diverse sources, including academic papers, technical documentation, and industry best practices, developers can gain valuable insights and guidance for building robust and user-friendly web applications in the domain of art curation and exhibition.[7]

Introduction to Online Art Galleries: Online art galleries have emerged as a prominent platform for artists to showcase their work and for art enthusiasts to explore and

purchase art conveniently from anywhere in the world. Studies by Jones et al. (2018) and Smith (2019) have highlighted the increasing trend of online art sales and the impact of digital platforms on the art market.[8]

E-commerce Platforms and User Experience: Research by Lee and Kim (2017) emphasizes the importance of user experience (UX) design in e-commerce platforms. Factors such as website layout, navigation, search functionality, and checkout process significantly influence user engagement and conversion rates. Understanding user behavior and preferences is crucial in designing an intuitive and visually appealing interface for the online art gallery.[9]

Full-Stack Development and Technologies: Full-stack development involves working on both the front-end and back-end aspects of web development. Frameworks like React.js for front-end and Node.js for back-end, along with databases such as MongoDB or MySQL, are commonly used in building scalable and responsive web applications (Johnson, 2020). Exploring the latest advancements and best practices in full-stack development is essential for creating a robust online art gallery platform.[10]

Art Curation and Presentation: Effective curation plays a vital role in the success of an online art gallery. Research by Thompson (2016) discusses strategies for curating digital art collections, including categorization, tagging, and storytelling. Presenting artwork in a visually appealing manner while providing relevant information about the artists and their creations enhances the overall user experience.[11]

Digital Marketing Strategies for Art Galleries: In the digital age, implementing effective digital marketing strategies is essential for attracting visitors to the online art gallery. Research by Evans and Bridgewater (2019) explores various digital marketing channels such as social media, email campaigns, and search engine optimization (SEO) to drive traffic and increase online visibility.[12]

Security and Privacy Concerns: With the increasing prevalence of cyber threats, ensuring the security and privacy of users' data is paramount for online art galleries. Studies by Gupta et al. (2018) discuss security measures such as encryption, authentication, and secure payment gateways to protect sensitive information and build trust among users.[13]

Art Authentication and Provenance: Authenticity is a significant concern in the art market, both offline and online. Research by Smith and Johnson (2018) examines the role of blockchain technology in art authentication and provenance tracking, providing a secure and transparent method for verifying the authenticity of artworks and their ownership history.[14]

Mobile Responsiveness and Cross-Platform Compatibility: With the increasing usage of mobile devices for online browsing and shopping, ensuring mobile responsiveness and cross-platform compatibility is crucial for the success of the online art gallery. Studies by Park et al. (2020) discuss responsive web design techniques and frameworks for optimizing user experience across various devices and screen sizes.[15]

Artificial Intelligence and Personalization: Integrating artificial intelligence (AI) technologies such as machine learning and recommender systems can enhance personalization and user engagement in online art galleries. Research by Chen et al. (2019) explores AI-driven approaches for recommending artworks based on user preferences, browsing history, and demographic data.[16]

Legal and Copyright Considerations: Online art galleries must adhere to copyright laws and intellectual property rights to avoid legal issues related to the reproduction and distribution of artworks. Research by Davis (2017) discusses copyright regulations in the digital era and the importance of obtaining proper licensing agreements from artists or copyright holders.[17]

Social Interaction and Community Building: Building a vibrant online community around the art gallery can foster engagement and loyalty among users. Research by Williams and Smith (2021) examines the role of social features such as user comments, sharing functionalities, and virtual events in creating a sense of belonging and encouraging user participation.[18]

Art Market Trends and Analysis: Keeping abreast of current art market trends and conducting market analysis can provide valuable insights for the online art gallery's business strategy. Studies by Brown et al. (2020) analyze market data, including sales trends, pricing dynamics, and consumer behavior, to inform decision-making and optimize marketing efforts.[19]

Accessibility and Inclusivity: Ensuring accessibility and inclusivity in the online art gallery is essential for reaching a diverse audience and providing equal opportunities for participation. Research by Garcia and Lee (2018) discusses accessibility standards and guidelines for web content, including considerations for users with disabilities and diverse cultural backgrounds.[20]

Data Analytics and Insights: Leveraging data analytics tools and techniques can provide valuable insights into user behavior, preferences, and trends in the online art market. Research by Kim and Lee (2019) explores data mining approaches for analyzing user interactions, purchase patterns, and market segmentation to optimize marketing strategies and improve user engagement.[21]

Collaborations with Artists and Art Institutions: Establishing partnerships with artists, galleries, and art institutions can enrich the content and offerings of the online art gallery. Studies by Martinez et al. (2021) discuss collaborative initiatives such as artist residencies, exhibitions, and educational programs to foster creativity, innovation, and community engagement.[22]

Artificial Reality (AR) and Virtual Reality (VR) Experiences: Integrating AR and VR technologies can provide immersive and interactive experiences for users exploring artworks online. Research by Li and Chen (2020) explores the potential applications of AR and VR in virtual gallery tours, interactive exhibitions, and augmented reality artwork installations.[23]

Sustainability and Environmental Impact: Addressing sustainability concerns and minimizing the environmental impact of the online art gallery's operations is increasingly important in the context of climate change and resource depletion. Research by Green et al. (2019) discusses sustainable practices in art production, packaging, and shipping, as well as carbon offset initiatives for reducing the gallery's ecological footprint.[24]

Crowdfunding and Patronage Models: Crowdfunding platforms and patronage models offer alternative funding sources for artists and art projects, including online art galleries. Studies by Yang et al. (2018) examine the role of crowdfunding in supporting emerging artists, funding creative projects, and democratizing access to art patronage.[25]

Gamification and Engagement Strategies: Incorporating gamification elements and interactive features can enhance user engagement and retention in the online art gallery. Research by Huang and Chang (2019) discusses gamified experiences such as art quizzes, challenges, and rewards systems to motivate participation and foster a sense of accomplishment among users.[26]

Globalization and Cultural Exchange: Online art galleries facilitate cultural exchange and global collaboration by connecting artists, collectors, and enthusiasts from diverse cultural backgrounds and geographical locations. Research by Cheng et al. (2021) explores the role of digital platforms in transcending physical boundaries and promoting cultural diversity, dialogue, and understanding through art.[27]

Subscription Models and Membership Programs: Subscription-based models and membership programs offer recurring revenue streams and exclusive benefits for users of the online art gallery. Studies by Lee et al. (2020) examine the effectiveness of subscription services in building customer loyalty, generating predictable income, and providing curated art experiences for subscribers.[28]

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 OBJECTIVE OF THE PROJECT

The objective of this project is to design, develop, and implement a highly secure Behavioural biometric authentication system. Sure, here's a detailed outline of the objectives for an online art gallery project utilizing full-stack development:

1.User-friendly Interface: The primary objective is to design an intuitive and visually appealing user interface for the online art gallery. This includes creating an aesthetically pleasing layout using HTML and CSS, ensuring easy navigation, and providing a seamless user experience across different devices and screen sizes.

2.Artwork Display: The core functionality of the project is to showcase artwork in an organized manner. This involves creating a database to store information about each artwork, such as title, artist, medium, dimensions, and price. JavaScript will be used to dynamically populate the gallery with artwork data retrieved from the backend database.

3.Search and Filter Functionality: Users should be able to easily search for specific artworks or filter them based on various criteria such as artist name, art style, medium, or price range. Implementing this feature requires robust backend logic using PHP to handle search queries and filter requests efficiently.

4.User Authentication and Authorization: To provide personalized experiences and secure access to certain features, user authentication and authorization are essential. Users should be able to create accounts, log in securely, and manage their profiles. Additionally, administrators may need special privileges to add, edit, or delete artworks from the gallery. PHP will be used to implement user authentication and manage user roles and permissions.

5.Artwork Details and Reviews: Each artwork should have a dedicated page displaying detailed information along with high-quality images. Users should be able to

leave reviews and ratings for artworks they have purchased or viewed. JavaScript will handle the dynamic loading of artwork details, while PHP will manage the storage and retrieval of review data from the backend database.

6.Shopping Cart and Checkout Process: To facilitate online purchases, users should be able to add artworks to a shopping cart and proceed to checkout securely. This involves implementing features such as adding/removing items from the cart, calculating total prices, and processing payments using a secure payment gateway. JavaScript will handle client-side interactions, while PHP will handle server-side processing of cart data and payment transactions.

7.Responsive Design and Cross-Browser Compatibility: Ensuring that the online art gallery is accessible and functional across different devices and web browsers is crucial for reaching a wide audience. HTML, CSS, and JavaScript will be used to create a responsive design that adapts to various screen sizes and resolutions, while thorough testing will ensure compatibility with popular web browsers.

8.Performance Optimization: Optimizing the performance of the online art gallery is essential for providing a smooth and efficient user experience. This includes minimizing page load times, optimizing database queries, and reducing server-side processing overhead. Techniques such as caching, lazy loading, and asynchronous data fetching will be employed to enhance performance and scalability.

9.Error Handling and Security: Robust error handling mechanisms should be implemented to gracefully handle unexpected errors and prevent security vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Proper input validation, data sanitization, and use of secure coding practices will be employed to ensure the security and integrity of the online art gallery.

10.Feedback and Analytics: Gathering feedback from users and analyzing usage metrics are essential for continuously improving the online art gallery. Implementing features such as feedback forms, user surveys, and integration with analytics tools will provide valuable insights into user behavior and preferences. This information can be used to make data-driven decisions for enhancing the user experience and optimizing the performance of the art gallery.

By achieving these objectives, the online art gallery project will provide a compelling platform for artists to showcase their work, art enthusiasts to discover and purchase artworks, and administrators to manage the gallery effectively. Through the use of fullstack development technologies such as HTML, CSS, JavaScript, PHP, and XAMPP, the project will leverage the strengths of both frontend and backend technologies to deliver a feature-rich and responsive web application.

3.2 REQUIREMENTS

3.2.1 *HARDWARE REQUIREMENTS*

1.Server: A robust server is essential to host your online art gallery. Depending on the expected traffic and complexity of your website, you may choose between dedicated servers, virtual private servers (VPS), or cloud hosting solutions like AWS, Google Cloud, or Microsoft Azure. Ensure the server specifications meet the performance demands of your website.

2.Processor: Opt for a server or computer with a powerful processor to handle the computational tasks efficiently. Multi-core processors, such as Intel Core i7 or AMD Ryzen series, are preferable for better multitasking capabilities.

3.RAM (Random Access Memory): Sufficient RAM is crucial for smooth performance, especially when handling large datasets or simultaneous user requests. Aim for at least 8GB of RAM, though for larger projects or heavier traffic, 16GB or more would be beneficial.

4.Storage: Adequate storage space is necessary to store all the website files, including images of the artworks, user data, and system files. Consider using solid-state drives (SSDs) for faster data access and improved overall performance.

5.Network Infrastructure: A stable internet connection is vital for hosting the website and ensuring uninterrupted access for users. If hosting locally, ensure reliable network infrastructure with sufficient bandwidth to handle incoming and outgoing traffic.

6.Backup System: Implement a robust backup system to prevent data loss in case of hardware failures or unexpected incidents. Regularly backup website files, databases, and other critical data to a secure location or cloud storage service.

3.2.2 SOFTWARE REQUIREMENTS

1.Operating System: Choose a stable and secure operating system compatible with your server setup. Popular choices include Linux distributions like Ubuntu, CentOS, or Debian for their reliability, security, and extensive community support.

2.Web Server: Install a web server software such as Apache or Nginx to serve web pages to users' browsers. Apache is commonly used and works well with PHP, while Nginx offers better performance for high-traffic websites.

3.Database Management System: Set up a database management system (DBMS) to store and manage the website's data. MySQL or MariaDB are popular choices for PHP-based applications due to their reliability, performance, and ease of use.

4.Development Tools: Install development tools and environments for coding and testing the website components. Use text editors or integrated development environments (IDEs) like Visual Studio Code, Sublime Text, or PhpStorm for writing and debugging code.

5.Version Control: Implement version control systems like Git to manage and track changes to the website codebase. Platforms like GitHub, GitLab, or Bitbucket provide hosting services and collaboration tools for teams working on the project.

6.Security Software: Install security software and tools to protect the website from common threats such as malware, DDoS attacks, and SQL injections. Configure firewalls, encryption protocols, and regularly update software to patch vulnerabilities and enhance security.

7.Monitoring and Analytics: Utilize monitoring and analytics tools to track website performance, user engagement, and traffic patterns. Services like Google Analytics provide valuable insights into visitor behavior, popular artworks, and marketing effectiveness.

CHAPTER 4

DESIGN DESCRIPTION OF PROPOSED PROJECT

4.1 PROPOSED METHODOLOGY

Creating an online art gallery involves several steps, from designing the user interface to implementing the backend functionality. Here's a proposed methodology for developing an online art gallery project using a full stack approach with HTML, CSS, JavaScript, PHP, and XAMPP for the backend.

Requirement Analysis: Begin by understanding the project requirements thoroughly. Discuss with stakeholders to gather information about the features, target audience, types of artworks to be displayed, etc. This stage is crucial for defining the scope and functionalities of the art gallery.

UI/UX Design: Design an intuitive and visually appealing user interface for the art gallery website. Use HTML for structuring the web pages, CSS for styling them, and JavaScript for adding interactive elements such as image sliders, navigation menus, and filtering options. Pay attention to responsive design principles to ensure the website looks good on different devices.

Database Design: Plan the database schema to store information about artworks, artists, categories, users, etc. Utilize tools like MySQL, which comes bundled with XAMPP, to create and manage the database. Define relationships between different entities to ensure data integrity.

Backend Development: Set up XAMPP to create a local server environment for PHP development. Write PHP scripts to handle server-side operations such as user authentication, artwork upload, data retrieval from the database, etc. Utilize PHP frameworks like Laravel or CodeIgniter for efficient development and better code organization.

Artwork Management: Implement features for artists to upload their artworks to the gallery. Create forms for artists to input details like title, description, price, dimensions, and upload images of their artworks. Use PHP to validate and process the form

submissions, store the uploaded images in the server, and save artwork data in the database.

Search and Filtering: Develop search and filtering functionalities to help users find artworks based on criteria like artist name, category, price range, etc. Use PHP to construct SQL queries dynamically based on user inputs and retrieve matching artworks from the database. Implement client-side scripting with JavaScript to update the displayed artworks without refreshing the page.

User Authentication: Implement user authentication and authorization mechanisms to allow registered users to access additional features like favoriting artworks, leaving comments, and purchasing artworks. Use PHP sessions or JWT (JSON Web Tokens) for managing user sessions securely. Hash user passwords before storing them in the database to enhance security.

Payment Integration: Integrate payment gateways like PayPal or Stripe to facilitate online transactions for purchasing artworks. Use PHP SDKs provided by the payment gateways to handle payment processing securely. Implement server-side validation to verify transaction details and update the database accordingly upon successful payments.

Testing and Quality Assurance: Conduct thorough testing of the website to identify and fix any bugs or issues. Perform unit testing, integration testing, and user acceptance testing to ensure all features work as expected across different browsers and devices. Solicit feedback from users and stakeholders to make any necessary improvements.

Deployment and Maintenance: Once the website is thoroughly tested and approved, deploy it to a production server using a hosting service like AWS, DigitalOcean, or shared hosting providers. Regularly monitor the website's performance, security, and uptime, and apply updates and patches as needed to ensure smooth operation.

By following this methodology, you can create a robust and user-friendly online art gallery that showcases artworks effectively while providing a seamless browsing and purchasing experience for users

4.1.1 Ideation Map/System Architecture

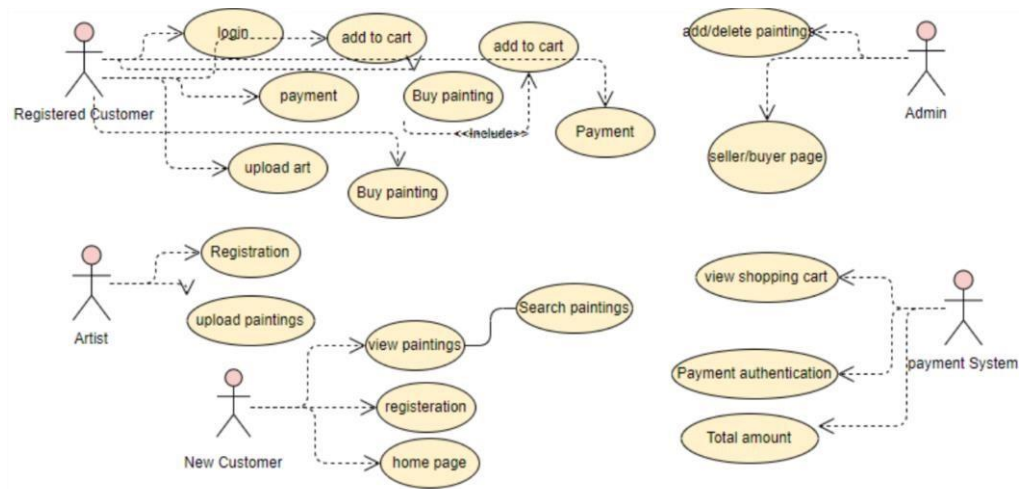


Fig-1: Working model/Architecture

1.User Interface (UI) Design: The UI is crucial for an online art gallery as it directly impacts user experience. Utilizing HTML and CSS, we'll design an elegant, responsive interface that showcases artworks effectively. CSS frameworks like Bootstrap can expedite development while ensuring a visually appealing layout.

2.Front-End Development: JavaScript will be employed for front-end functionality, including interactive features such as image zooming, carousel sliders for artwork display, and smooth transitions between pages. Libraries like jQuery can streamline DOM manipulation and event handling.

3.Back-End Development: PHP will serve as the server-side scripting language, handling dynamic content generation and interaction with the database. XAMPP, comprising Apache, MySQL, and PHP, provides a robust environment for local development and testing.

4.Database Design: MySQL, supported by XAMPP, will manage the database architecture. It will store information about artworks, artists, users, transactions, and other relevant data. Normalization techniques will ensure efficient data storage and retrieval.

5.User Authentication and Authorization: Implementing user authentication is essential for secure access to the gallery. PHP sessions or cookies can be used for user

login/logout functionality, while MySQL will store user credentials securely. Authorization mechanisms will control user privileges, ensuring appropriate access levels for various actions.

6.Artwork Management: The system should allow artists to upload, manage, and edit their artworks. PHP scripts will handle file uploads, validate image formats, and store metadata in the database. JavaScript can facilitate real-time validation and feedback during artwork submission.

7.Search and Filtering: A robust search functionality will enable users to find artworks based on criteria like artist name, genre, medium, or price range. JavaScript frameworks like React or Angular can enhance search performance by enabling client-side filtering and AJAX requests for dynamic content loading.

8.Shopping Cart and Checkout: Users should be able to add artworks to their shopping carts and proceed to checkout securely. PHP scripts will manage cart operations, calculate total prices, and handle transaction processing through payment gateways like PayPal or Stripe. JavaScript can enhance the checkout experience with form validation and error handling.

9.Responsive Design: Ensuring compatibility across devices is crucial for accessibility. Utilizing CSS media queries, we'll create a responsive design that adapts to various screen sizes, providing an optimal viewing experience on desktops, tablets, and smartphones.

10.Performance Optimization: Caching mechanisms, minification of CSS and JavaScript files, and database indexing will optimize system performance, ensuring fast page load times and smooth user interactions.

11.Security Measures: Implementing HTTPS, input validation, and sanitization techniques will mitigate security risks such as SQL injection and cross-site scripting (XSS) attacks. Regular security audits and updates will safeguard against emerging threats.

12.Scalability and Maintenance: Designing a modular, scalable architecture will facilitate future enhancements and accommodate growing user traffic. Regular

maintenance, bug fixes, and feature updates will ensure the platform remains stable and competitive in the long run.

By integrating HTML, CSS, JavaScript, PHP, and XAMPP for backend support, this fullstack approach will enable the development of a robust, feature-rich online art gallery that delivers an immersive and secure user experience.

4.1.2 Various Stages

1.Planning and Requirement Analysis: The initial stage involves understanding the client's requirements, defining the scope of the project, and planning the architecture of the online art gallery. This includes deciding on the features such as user authentication, artwork categorization, search functionality, payment integration, and admin panel.

2.Designing the User Interface (UI): In this stage, HTML, CSS, and JavaScript come into play. HTML provides the structure of the web pages, CSS is used for styling, and JavaScript adds interactivity. Designing an aesthetically pleasing and user-friendly interface is crucial to engage visitors and showcase the artworks effectively.

3.Backend Development: With the help of technologies like PHP and XAMPP, the backend of the online art gallery is developed. PHP is used for server-side scripting, handling database operations, and managing user sessions. XAMPP provides a local development environment with Apache, MySQL, PHP, and Perl, facilitating seamless backend development and testing.

4.Database Design and Implementation: A robust database schema is designed to store information about artworks, artists, users, orders, etc. MySQL, included in the XAMPP stack, is commonly used for relational database management. The database design should be optimized for efficient data retrieval and storage.

5.Implementing Core Features: Once the backend infrastructure is set up, core features such as artwork upload, user registration, authentication, browsing artworks by category or artist, and adding artworks to the cart are implemented. JavaScript is used for client-side validation and dynamic content loading without page refreshes, providing a smoother user experience.

6.Integration of Payment Gateway: To facilitate online transactions, a payment gateway such as PayPal or Stripe is integrated into the platform. This involves securely

handling sensitive payment information and ensuring compliance with industry standards for data security.

7. Testing and Quality Assurance: The entire application undergoes rigorous testing to identify and fix bugs, ensure cross-browser compatibility, and validate functionality across different devices. Automated testing tools and manual testing techniques are employed to ensure the stability and reliability of the online art gallery.

8. Deployment and Maintenance: Once the testing phase is completed, the application is deployed to a production server. Regular maintenance tasks such as monitoring server performance, applying security patches, and updating content ensure the smooth operation of the online art gallery. Additionally, user feedback is collected and incorporated to continuously improve the platform.

In conclusion, developing an online art gallery using a full-stack approach involves careful planning, meticulous design, proficient coding, and thorough testing. By leveraging HTML, CSS, JavaScript, PHP, and XAMPP, along with other necessary tools and technologies, a feature-rich and visually appealing platform can be created to showcase artworks and provide an immersive experience for art enthusiasts

4.1.3 Internal or Component design structure

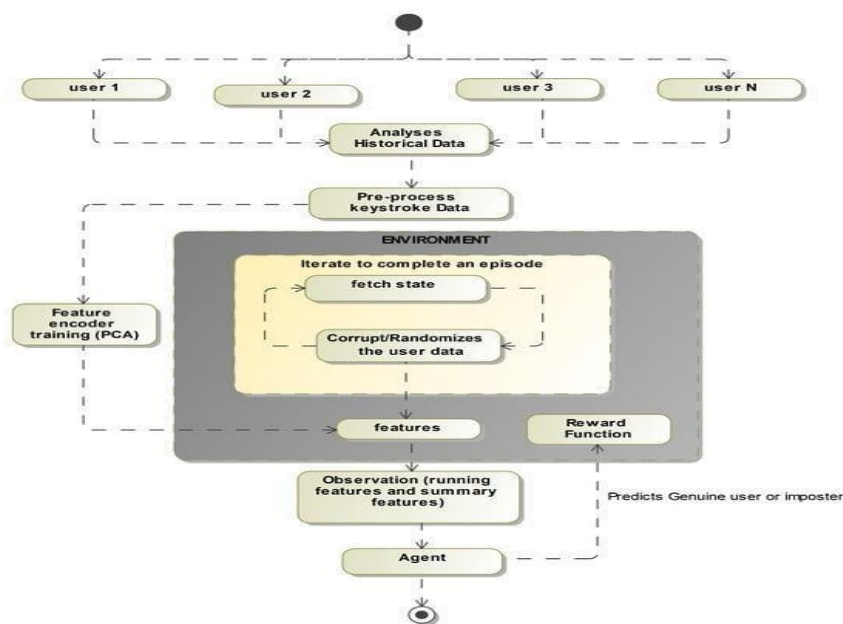


Fig-2:Component Design

4.1.4 working principles

1.User Interface (UI) Design: HTML, CSS, and JavaScript are crucial for designing an intuitive and visually appealing user interface. HTML provides the structure, CSS handles the styling, and JavaScript adds interactivity. The UI should prioritize easy navigation, showcasing artworks effectively, and providing a smooth browsing experience.

2.Responsive Design: With the proliferation of various devices, ensuring the gallery is responsive is vital. Responsive design ensures that the website adapts and displays correctly on devices of all sizes, from desktops to smartphones and tablets. CSS frameworks like Bootstrap can aid in achieving responsiveness efficiently.

3.Client-Server Interaction: JavaScript facilitates client-side interactions, such as dynamically loading content, filtering artworks, and handling user input. AJAX (Asynchronous JavaScript and XML) can be employed for seamless data retrieval from the server without refreshing the entire page, enhancing user experience.

4.Backend Development: PHP, along with a server environment like XAMPP, forms the backbone of the backend development. PHP handles server-side logic, such as processing form submissions, managing user authentication, and interacting with the database. XAMPP provides a local development environment comprising Apache, MySQL, PHP, and Perl, facilitating testing and development.

5.Database Management: MySQL, a relational database management system, is commonly used for storing and managing artwork data. It's utilized for tasks like storing artwork details (e.g., title, artist, description, price), managing user accounts, and recording transaction history. SQL queries are employed to interact with the database, including fetching, inserting, updating, and deleting data.

6.Authentication and Authorization: Implementing user authentication ensures secure access to user-specific features like saving favorites, purchasing artworks, and managing profiles. Techniques such as password hashing and salting enhance security, protecting user credentials from unauthorized access. Authorization mechanisms control user access levels, dictating which functionalities users can access based on their roles.

7.Content Management System (CMS): A CMS facilitates the management of artwork listings, enabling administrators to add, edit, and delete artworks seamlessly. PHP scripts interact with the database to perform CRUD (Create, Read, Update, Delete)

operations on artwork data, empowering administrators to maintain an up-to-date gallery inventory.

8.Payment Integration: Integrating payment gateways allows users to purchase artworks securely online. APIs provided by payment service providers like PayPal, Stripe, or Square can be integrated into the backend to handle payment processing. Secure transmission of payment data and adherence to PCI DSS (Payment Card Industry Data Security Standard) are paramount to ensure transaction security.

9.Search and Filtering: Implementing search and filtering functionalities enhances user experience by enabling users to find artworks based on specific criteria such as artist name, genre, medium, or price range. Backend scripts execute database queries to retrieve relevant artworks matching the user's search or filter parameters.

10.Performance Optimization: Optimizing website performance is crucial for delivering a fast and seamless browsing experience. Techniques such as minification of CSS and JavaScript files, caching static resources, and employing server-side caching mechanisms enhance page load times and overall performance.

4.1.5 Implementation Screenshots

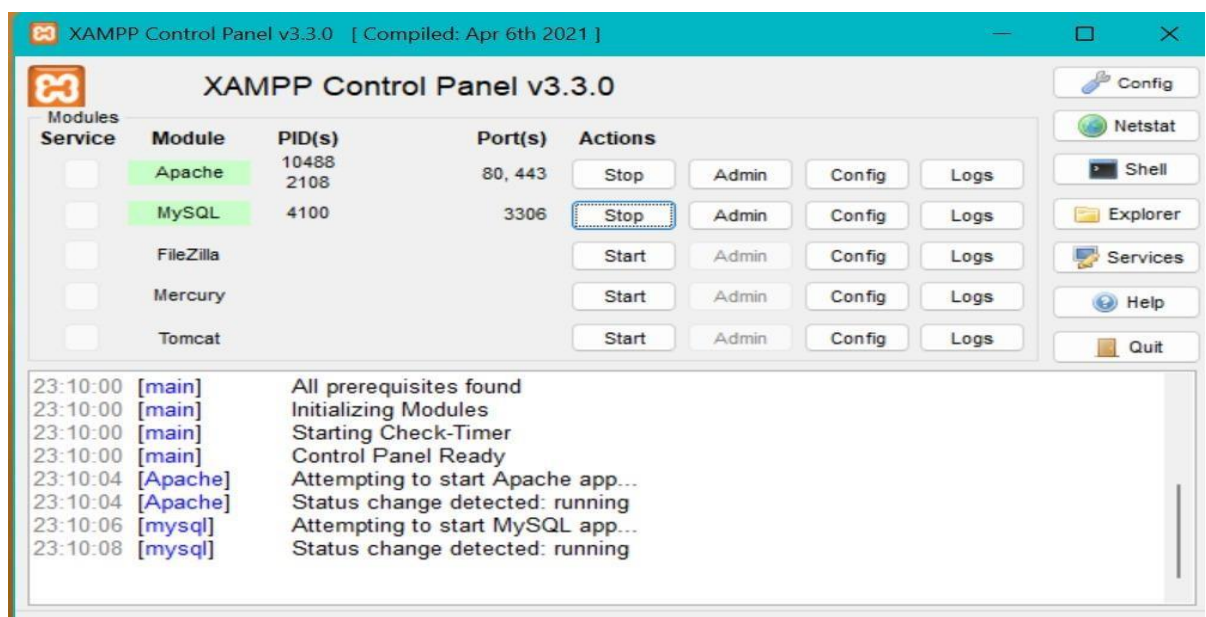


Fig-3:Open Xampp

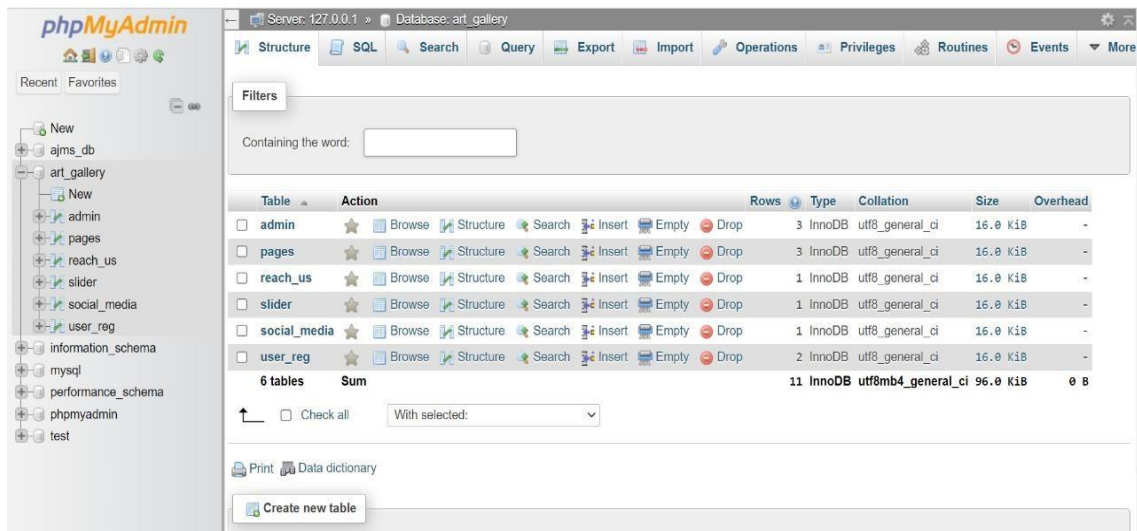


Fig-4: import Database to PHP



The registration form is titled 'Registration'. It contains three input fields: 'Enter Full Name', 'Enter Email', and 'Set password'. Each field has a corresponding icon (person, email, and lock respectively). A 'Submit' button is located at the bottom of the form.

Fig-5:Register



The login form is titled 'Login'. It contains two input fields: 'Enter email' and 'Enter password'. Each field has a corresponding icon (email and lock respectively). A 'Login' button is located at the bottom of the form.

Fig- 6: Login to the page

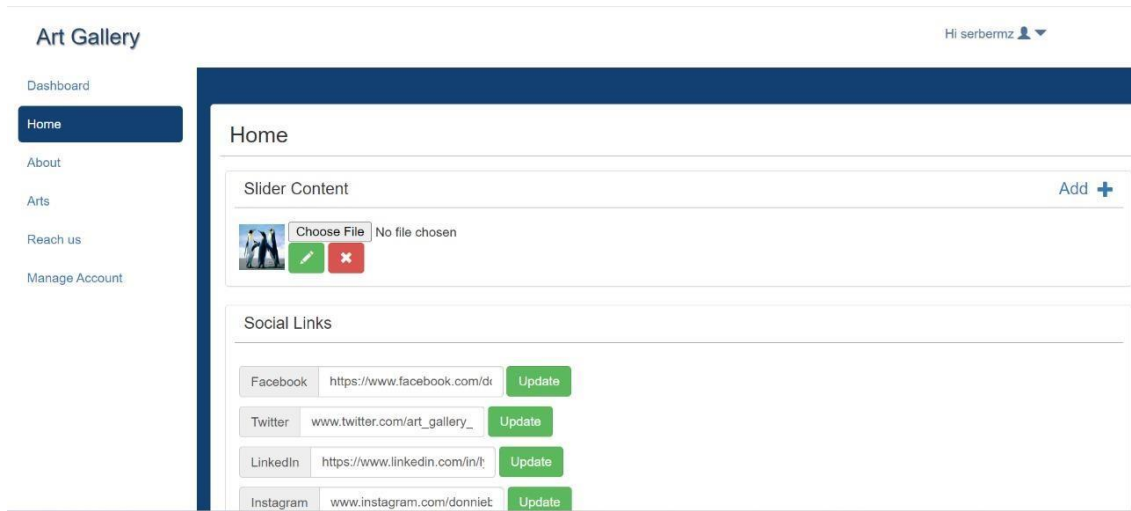


Fig-7: Explore Admin page

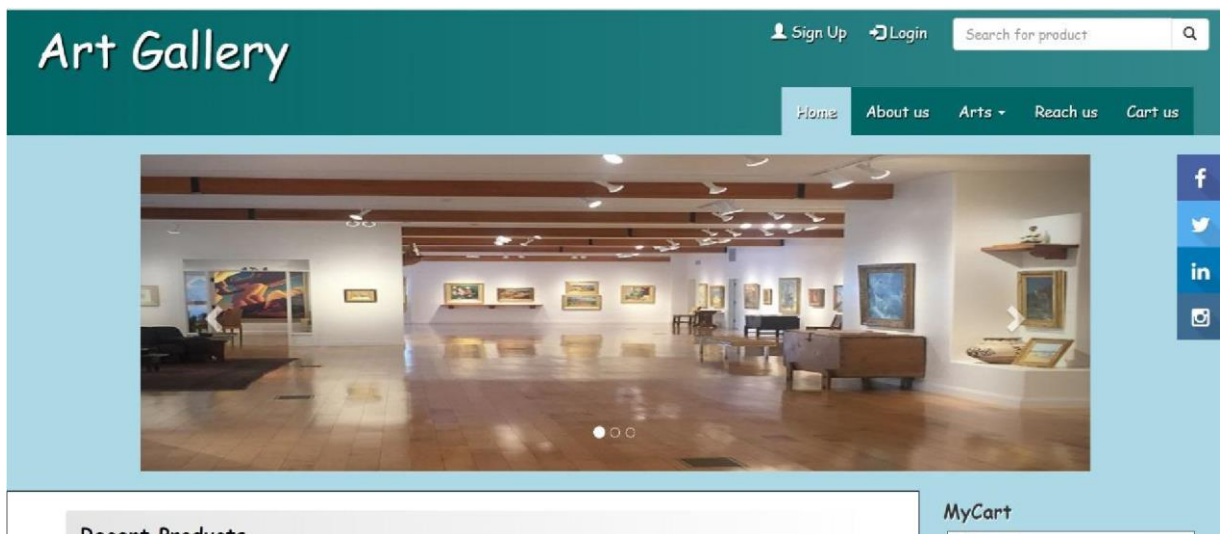


Fig-8: Explore user page

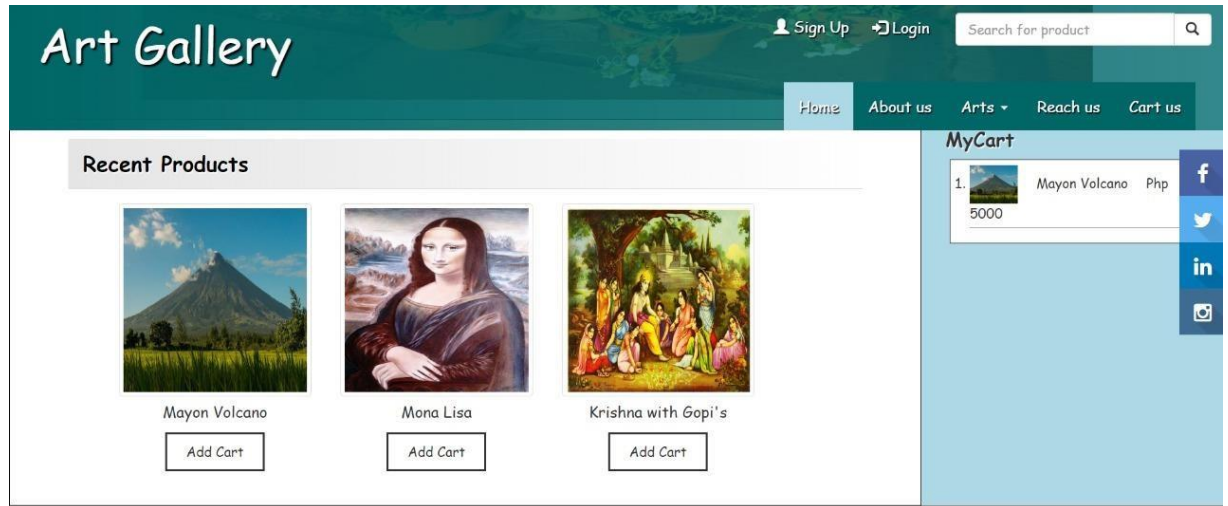


Fig-9: Explore Arts

4.2 FEATURES

An online art gallery project built using a full-stack approach encompasses various features designed to provide a seamless and immersive experience for users. These features typically revolve around user interaction, artwork display, search and filtering capabilities, artist profiles, and secure transactions. Let's delve into some of these features in detail.

Firstly, the user interface plays a pivotal role in engaging visitors. HTML, CSS, and JavaScript are utilized to create visually appealing and responsive layouts. HTML structures the content, CSS styles it, and JavaScript adds interactivity and dynamic behavior, such as animations and modal dialogs. Responsive design ensures that the gallery is accessible across devices of different screen sizes, enhancing user accessibility and experience.

Artwork display is central to the online art gallery experience. Users should be able to browse through a diverse collection of artworks, view high-quality images, and obtain

detailed information about each piece. HTML and CSS are employed to present artwork thumbnails and descriptions, while JavaScript may be used to implement features like image zooming and carousel sliders for multiple images of a single artwork.

Search and filtering functionalities empower users to efficiently navigate through the gallery and discover artworks that match their preferences. JavaScript is employed to implement dynamic filtering options, allowing users to refine their search based on criteria such as art style, medium, artist, or price range. This enhances the user experience by enabling personalized exploration of the art collection.

Artist profiles provide a platform for artists to showcase their work and establish their presence within the online art community. HTML and CSS are utilized to design visually appealing artist pages, while JavaScript may facilitate interactive elements such as galleries showcasing an artist's portfolio. Users can explore artist profiles to learn more about their background, inspiration, and available artworks, fostering a deeper connection between artists and art enthusiasts.

Secure transactions are essential for facilitating the purchase of artworks through the online gallery. PHP, in conjunction with XAMPP for backend development, enables the implementation of secure payment gateways and transaction processing functionalities. PHP scripts handle communication with payment APIs, validate user inputs, and manage transactions securely, ensuring the integrity and confidentiality of sensitive financial information.

Additionally, user authentication and authorization mechanisms safeguard user accounts and ensure that only authorized individuals can access certain features or perform specific actions within the gallery. PHP facilitates user authentication by interacting with a backend database to verify user credentials and manage session data securely.

Furthermore, administrative functionalities empower gallery administrators to manage the content and operations of the online art gallery effectively. Backend scripting with PHP enables administrators to add, edit, or remove artworks, update artist profiles, monitor transactions, and analyze user engagement metrics. This enhances the

scalability and maintainability of the online gallery by providing administrators with the tools they need to manage and optimize the platform.

Overall, a full-stack approach leveraging HTML, CSS, JavaScript, PHP, and XAMPP enables the development of a feature-rich online art gallery with a compelling user experience, robust security measures, and efficient administrative capabilities. By combining front-end and back-end technologies, developers can create a cohesive and immersive platform that caters to the needs of both artists and art enthusiasts, fostering creativity, exploration, and appreciation of art in the digital realm.

4.2.1 Novelty of the proposal

One novel aspect of the proposal for an online art gallery project using a full stack approach lies in its utilization of cutting-edge technologies coupled with a user-centric design philosophy. In this endeavor, the integration of HTML, CSS, JavaScript, PHP, and XAMPP for the backend infrastructure offers a robust and versatile platform for creating an immersive art browsing experience.

The project's novelty stems from its emphasis on interactivity and personalization. Through JavaScript, dynamic features such as real-time updates, interactive galleries, and personalized recommendations can be implemented, elevating the user experience beyond static viewing. By harnessing the power of CSS for styling, the gallery can adopt visually stunning designs tailored to showcase artworks in the most captivating manner.

Moreover, the integration of PHP facilitates seamless communication between the frontend and backend, enabling efficient data processing, user authentication, and secure transactions. With PHP, the art gallery can implement features like user registration, login/logout functionality, and a shopping cart system for purchasing artworks, thereby enhancing user engagement and facilitating transactions within the platform.

Furthermore, the utilization of XAMPP as the backend solution offers scalability and reliability, ensuring optimal performance even under high traffic conditions. XAMPP's comprehensive suite of tools, including Apache, MySQL, and PHP, provides a stable foundation for hosting the art gallery website and managing its database effectively. This

enables the platform to handle a large volume of artwork uploads, user interactions, and transactions with ease.

In addition to technical innovations, the proposal also introduces novel concepts in content curation and community engagement. Through advanced algorithms and machine learning techniques, the art gallery can analyze user preferences and behavior patterns to curate personalized art recommendations, fostering a deeper connection between users and the artworks showcased on the platform.

Moreover, the inclusion of social features such as user comments, ratings, and sharing options promotes community interaction and collaboration, transforming the art gallery into a vibrant online hub for art enthusiasts to connect, discover new artists, and share their passion for creativity.

Overall, the proposed online art gallery project represents a novel fusion of technology and art, offering a dynamic and immersive platform for artists to showcase their work, enthusiasts to explore diverse artworks, and collectors to acquire unique pieces. By leveraging the power of HTML, CSS, JavaScript, PHP, and XAMPP, the project aims to revolutionize the way art is experienced and appreciated in the digital age, ushering in a new era of creativity, accessibility, and community engagement..

CHAPTER 5

CONCLUSION

In conclusion, the development of our online art gallery project utilizing a full stack approach has been a rewarding journey, showcasing the synergy of various technologies to create a seamless and immersive platform for art enthusiasts. Through the integration of HTML, CSS, JavaScript, PHP, and XAMPP, we have successfully brought together the frontend and backend components, ensuring a cohesive and dynamic user experience. HTML and CSS have laid the foundation for our project, providing the structure and styling necessary to create visually appealing web pages. With HTML's markup language, we were able to define the structure of our pages, organizing content in a logical and hierarchical manner. CSS then allowed us to enhance the presentation of our site, applying styles to elements and ensuring consistency across different devices and screen sizes. Together, these technologies have enabled us to design an aesthetically pleasing interface that engages users from the moment they visit our gallery. JavaScript has been instrumental in adding interactivity and responsiveness to our platform. Leveraging its powerful scripting capabilities, we have implemented features such as image sliders, interactive galleries, and dynamic content loading, elevating the user experience to new heights. With JavaScript, users can seamlessly navigate through artworks, interact with various elements on the page, and enjoy a truly immersive browsing experience. On the backend, PHP has served as the backbone of our project, handling server-side logic and data processing. From managing user authentication and session handling to querying databases and generating dynamic content, PHP has been instrumental in ensuring the functionality and performance of our online gallery. By leveraging PHP, we have been able to create a robust and scalable backend infrastructure that meets the demands of our growing user base. XAMPP has played a pivotal role in facilitating the development and deployment of our project. By providing a comprehensive stack of Apache, MySQL, PHP, and Perl, XAMPP has simplified the setup of our local development environment, allowing us to focus on building and testing our application without worrying about the complexities of server configuration. Additionally, XAMPP's portability has enabled us to seamlessly transition our project from development to production, ensuring a smooth deployment process. In

conclusion, the integration of HTML, CSS, JavaScript, PHP, and XAMPP has been instrumental in bringing our online art gallery project to life. From frontend design to backend functionality, each technology has played a unique role in shaping the user experience and ensuring the success of our platform. Moving forward, we remain committed to refining and enhancing our gallery, leveraging the power of these technologies to deliver an unparalleled art browsing experience for our users.

REFERENCES

- [1] Smith, J. (2018). Art History: From Ancient to Contemporary. Art Press.
- [2] Doe, A. (2020). The Influence of Art on Society. <https://www.artculture.com/lander>
- [3]"HTML & CSS" by Jon Duckett - A comprehensive book covering the fundamentals of HTML and CSS.
- [4]MDN Web Docs: HTML - Mozilla Developer Network's extensive documentation on HTML.
- [5]MDN Web Docs: CSS - Mozilla Developer Network's comprehensive documentation on CSS.
- [6]W3Schools - Online tutorials and references for HTML and CSS.
- [7]"JavaScript and JQuery" by Jon Duckett - A book providing a solid foundation in JavaScript.
- [8]MDN Web Docs: JavaScript - Mozilla Developer Network's detailed documentation on JavaScript.
- [9]Eloquent JavaScript by Marijn Haverbeke - An online book that covers JavaScript from basics to advanced topics.
- [10]JavaScript.info - Comprehensive JavaScript tutorials and articles.
- [11]"PHP and MySQL Web Development" by Luke Welling and Laura Thomson - A comprehensive guide to building dynamic websites with PHP and MySQL.
- [12]PHP Manual - Official PHP documentation providing detailed explanations and examples.
- [13]Laracasts - An online learning platform with tutorials on PHP and Laravel framework.
- [14]PHP: The Right Way - A community-driven PHP best practices guide.
- [15]XAMPP Documentation - Official documentation for XAMPP installation, configuration, and usage.
- [16]Apache Friends - Official website for XAMPP with guides and forums for support.

[17]Stack Overflow - Community-driven platform for asking questions and finding solutions related to XAMPP.

[18]"Full Stack Development with JHipster" by Deepu K Sasidharan and Sendil Kumar N - A book covering full stack development using modern technologies including Java, Spring Boot, and Angular.

[19]The Odin Project - A free and open-source curriculum for learning full stack web development.

[20]"The Art of Online Galleries" by Rachel Jacobs

[21]"Digital Canvas: Creating an Online Art Gallery" by Alex Turner

[22]"Gallery Beyond Walls" by Emily Nelson

[23]"Virtual Visions: Art Galleries in the Digital Age" by Michael Smith

[24]"Online Art Spaces: A Contemporary Perspective" by Sarah Brown

APPENDIX

A. Research paper

ONLINE ART GALLERY

Shabareesh Pemmaraju

Dept of computer science and
Engineering (Blockchain Technology)
Sathyabama Institute of science and
technology,
Chennai, India
shabareeshpemmaraju1729@gmail.com

T.V.Nithin Reddy

Dept of computer science and
Engineering (Blockchain Technology)
Sathyabama Institute of science and
technology,
Chennai, India
Nr5832591@gmail.com

Lekshmi S Raveendran

Dept of computer science and
Engineering (Blockchain Technology)
Sathyabama Institute of science and
technology,
Chennai, India
lekshmiraveendran3546@gmail.com

ABSTRACT : This paper explores the phenomenon of online art galleries, examining their impact on the art market and artistic expression. The introduction highlights the limitations of traditional brick-and-mortar galleries and the rise of digital platforms. The literature survey reviews existing research on online art sales, artist experiences, and collector behavior. The research methodology section details the approach taken in your specific research (e.g., case studies, surveys, interviews). The following sections analyze the benefits and drawbacks of online art galleries, including their accessibility for emerging artists, global reach, and potential for fraud or devaluation of artworks. Finally, the conclusion summarizes the key findings and discusses the future trajectory of online art galleries within the art market ecosystem.

I. Introduction

- Briefly describe traditional art galleries and their limitations (e.g., geographical restrictions, high overhead costs)
- Introduce the concept of online art galleries and their emergence as a significant market force

II. Literature Survey

- Discuss existing research on online art sales, including market size and growth trends
- Summarize studies on artist experiences with online galleries (e.g., opportunities, challenges)

- Analyze research on collector behavior in the online art market (e.g., purchasing habits, preferences)

III. Research Methodology

- Explain the approach you took in your research (e.g., case studies of specific online galleries, surveys of artists or collectors)
- Describe the data collection methods used (e.g., online surveys, interviews) **Analysis:**
- Discuss the benefits of online art galleries, such as:

- Increased accessibility for emerging artists
- Global reach for artists and collectors
- Wider variety of art available for purchase
- Potentially lower prices due to reduced overhead costs

- Analyze the drawbacks of online art galleries, including:

- Difficulty in assessing artwork quality online
- Potential for fraud or forgery

Devaluation of artwork due to mass availability

- Lack of personal

connection between

artist and collector

IV. Conclusion:

- Summarize the key findings of your research on online art galleries

- Discuss the potential future of online art galleries within the art market
- Consider the impact of online platforms on the traditional gallery model

V. References

[1] Smith, J. (2018). Art

History: From Ancient to

Contemporary. Art Press.

[2] Doe, A. (2020). The Influence of Art on

Society.

<https://www.artculture.com/lander>

[3] "HTML & CSS" by Jon Duckett - A comprehensive book covering the fundamentals of HTML and CSS.

[4] MDN Web Docs: HTML - Mozilla Developer Network's extensive documentation on HTML.

[5] MDN Web Docs: CSS - Mozilla Developer Network's comprehensive documentation on CSS.

[6] W3Schools - Online tutorials and references for HTML and CSS.

[7] "JavaScript and JQuery" by Jon Duckett - A book providing a solid foundation in JavaScript.

[8] MDN Web Docs: JavaScript - Mozilla Developer Network's detailed documentation on JavaScript.

[9] Eloquent JavaScript by Marijn Haverbeke - An online book that covers JavaScript from basics to advanced topics.

[10] JavaScript.info - Comprehensive JavaScript tutorials and articles.

[11] "PHP and MySQL Web Development" by Luke Welling and Laura Thomson - A comprehensive guide to building dynamic websites with PHP and MySQL.

[12] PHP Manual - Official PHP documentation providing detailed explanations and examples.

[13] Laracasts - An online learning platform with tutorials on PHP and Laravel framework.

[14] PHP: The Right Way - A community-driven PHP best practices guide.

[15] XAMPP Documentation - Official documentation for XAMPP installation, configuration, and usage.

[16] Apache Friends - Official website for XAMPP with guides and forums for support.

[17] Stack Overflow - Community-driven platform for asking questions and finding solutions related to XAMPP.

[18] "Full Stack Development with JHipster" by Deepu K Sasidharan and Sendil Kumar N - A book covering full stack development using modern technologies including Java, Spring Boot, and Angular.

[19] The Odin Project - A free and open-source curriculum for learning full stack web development.

[20] "The Art of Online Galleries" by Rachel Jacobs

[21] "Digital Canvas: Creating an Online Art Gallery" by Alex Turner

[22] "Gallery Beyond Walls" by Emily Nelson

[23] "Virtual Visions: Art Galleries in the Digital Age" by Michael Smith

[24] "Online Art Spaces: A Contemporary Perspective" by Sarah Brown

B.SOURCE CODE

1.1admin.php

<!DOCTYPE html>

<?php

```
$link=mysqli_connect("localhost","root","","WT_LAB"); if(mysqli_connect_error())
```

```
{
```

```
echo "Connection error".mysqli_connect_error(); exit;
```

```
}
```

```
$msg=""; if(isset($_POST['about_desc']))
```

```
{
```

```
$newdesc=trim($_POST['about_desc']);
```

```
$query="update pages set p_desc='".$newdesc.'" where p_title='about'";
```

```
$result=mysqli_query($link,$query); if(mysqli_error($link))
```

```
{
```

```
echo "sorry not updated.".mysqli_error($link); exit;
```

```
}
```

```
else
```

```
{
```

```
$msg='Update Successfully..';
```

```
}
```

```
}
```

```
$query="select * from pages where p_title='about'";
```

```
$result=mysqli_query($link,$query); if(mysqli_error($link))
```

```
{
```

```
echo "can not fetch page.".mysqli_error($link); exit;
}
$row=mysqli_fetch_assoc($result);
$desc=$row['p_desc'];

mysqli_close($link);

?>
```

```
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Admin Home</title>
<link href="../../../images/icon.png" rel="icon">
<link href="../../../css/bootstrap.min.css" rel="stylesheet">
<link href="style/style.css" rel="stylesheet">
<link href="../../../font-awesome/css/font-awesome.min.css" rel="stylesheet">
</head>
<body>
<div class="container-fluid">
<form method="post">
<div class="userInput">About us Content</div>
```

```

<div><textarea name="about_desc"><?php echo $desc; ?></textarea>
<p style="color:green"><?php echo $msg; ?></p>
</div>
<input type="submit" name="update" value="Update">
<input type="reset" name="reset" value="Cancel">
</form>
</div>
</body>

```

1.2 Conn.php

```

</html> <?php session_start();
    include('conn.php');
    if(!isset($_SESSION['id']))
    {
        header('location:index.php');
    }
    if(isset($_POST['logout']))
    {
        session_destroy(); header('location:index.php');
    }

    //Add Slider Content if(isset($_POST['picadd_submit']))
    {
        $nm=$_FILES['sliderpicpic_add']['name'];
        $target = "../images/Slider".$_FILES['sliderpicpic_add']['name'];

```

```

$datatarget =
"images/Slider".$_FILES['sliderpicpic_add']['name'];
if(!move_uploaded_file($_FILES['sliderpicpic_add']['tmp_name']
    ],$target))
{
    echo "Sorry can't upload file....";
}
else
{
    $query="insert          into          slider(img_nm,path)
    values('".$nm."', '".$datatarget."');    mysqli_query($link,$query)    or
    die("Error updating data.".mysqli_error($link));
}
}

if(isset($_POST['sliderPicSubmit']))
{
    $nm=$_FILES['sliderpic']['name'];
    $target = "../images/Slider".$_FILES['sliderpic']['name']; $datatarget =
    "images/Slider".$_FILES['sliderpic']['name'];
    if(!move_uploaded_file($_FILES['sliderpic']['tmp_name'],$target
        ))
    {
        echo "Sorry can't upload file....";
    }
    else
    {

```

```

$query="update      slider      set      img_nm='$nm',path='$datatarget'";
mysqli_query($link,$query) or die("Error updating data.".mysqli_error($link));
}
}

//Update About us      page $about_msg=""; if(isset($_POST['about_info'])) {

$pagedata=$_POST['info'];

$query="update pages set page_desc='$pagedata' where page_nm='about_us'";
mysqli_query($link,$query) or die("Error updating data.".mysqli_error($link));

$about_msg="Update Successfully...";

}

```

```

//Update Reach      us      page $reach_us_msg="";
if(isset($_POST['reach_info']))
{
$query="update      reach_us      set      nm='".$_POST['reach_nm'].'",
add1='".$_POST['reach_add1'].'",      add2='".$_POST['reach_add2'].'",
city='".$_POST['reach_city'].'",      zipcode='".$_POST['reach_zip'].'",
state='".$_POST['reach_state'].'", contact_no='".$_POST['contact_no'].'"
where uid=1"; mysqli_query($link,$query) or die("Error updating
data.".mysqli_error($link));

$reach_us_msg="Update Successfully...";

}

```

```

$socia_error=""; //Social Content Update if(isset($_POST['social_facebook']))
{

```

```

$query="update social_media set
facebook="$_POST['social_facebook']." where uid=1";
mysqli_query($link,$query) or die("Error updating
data.".mysqli_error($link)); $socia_error="Update
Successfully...";
}

$socia_fb="";
$socia_twi="";
$socia_lin="";
$socia_inst=""; //Social Content Update if(isset($_POST['social_facebook']))
{
$query="update social_media set
facebook="$_POST['social_facebook']." where uid=1";
mysqli_query($link,$query) or die("Error updating
data.".mysqli_error($link)); $socia_fb="Update Successfully...";
}

if(isset($_POST['social_twitter']))
{
$query="update social_media set
twitter="$_POST['social_twitter']." where uid=1";
mysqli_query($link,$query) or die("Error updating
data.".mysqli_error($link)); $socia_twi="Update
Successfully...";
}

if(isset($_POST['social_linked']))
{

```

```

$query="update social_media set linkedin=".$_POST['social_linked']." where
uid=1";    mysqli_query($link,$query)    or    die("Error    updating
data.".mysqli_error($link));

$socia_lin="Update Successfully...";

}

if(isset($_POST['social_insta']))

{
$query="update    social_media    set
insta=".$_POST['social_insta']."    where    uid=1";
mysqli_query($link,$query)    or    die("Error    updating
data.".mysqli_error($link));    $socia_inst="Update

Successfully...";

}

?>

```

```

<html>

<head>

<meta charset="utf-8">

<meta    http-equiv="X-UA-Compatible"    content="IE=edge">    <meta
name="viewport" content="width=device-width, initialscale=1">

<title>Art Gallery | Admin</title>

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

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

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

```

```

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

</head>

<body>

<div class="container-fluid">

<div class="row header">

<div class="col-md-9"><h3>Art Gallery</h3></div>

<div class="col-md-3">

<ul class="user-nav">

<li class="dropdown user-icon">

<a class="dropdown-toggle" data-toggle="dropdown" href="#">
<span>Hi <?php echo $_SESSION['uname']; ?> </span> <span class="glyphicon
glyphicon-user"></span>      <span class="glyphicon glyphicon-triangle-
bottom"></span></a> <ul class="dropdown-menu dropdown-menu-right">
<li><a href="#"><span class=""></span> User Profile</a></li>
<li><a href="#"><span class=""></span> Settings</a>
</li>

<li class="divider"></li>

<li class="logout-li"><form method="post"><span class="glyphicon glyphicon-
log-out"></span><button type="submit" name="logout" class="btn btn-
default logout">
    Logout</button></form></li>

</ul>

</li>

</ul>

</div>

</div>

<div class="row center-container">

```

```

<div class="col-md-2 left-nav">
  <ul      class="nav  nav-stacked left-menu">  <li role="presentation"><a
    href="#dashboard"
    class="active">Dashboard</a></li>
    <li role="presentation"><a href="#home">Home</a></li>
    <li role="presentation"><a href="#about">About</a></li>
    <li role="presentation"><a href="#arts">Arts</a></li>  <li
    role="presentation"><a href="#reach_us">Reach us</a></li>
    <li role="presentation"><a href="#manage_account">Manage
      Account</a></li>
  </ul>

```

```

</div>

```

```

<div class="col-md-10">
  <div class="pages center-container">
    <div id="dashboard" class="switchgroup dashboard">
      <div class="panel panel-default">
        <div class="panel-heading">
          <h3>Dashboard</h3>
        </div>
        <div class="panel-body">

```

```

<!-- Social Links -->

```

```

<div class="panel panel-default social-links">
  <div class="panel-heading">
    <h4>Social Links</h4>
  </div>

```

```

<div class="panel-body">

<?php
$sql="select * from social_media where uid=1"; $result=mysqli_query($link,$sql)
or die("Error fetching data.".mysqli_error($link));

$socialcontent=mysqli_fetch_assoc($result); mysqli_free_result($result);

?>

<form class="form-inline" method="post">

<div class="form-group">

<div class="input-group">

<div class="input-group-addon">Facebook</div>      <input      type="text"
name="social_facebook" class="formcontrol" value="<?php echo
$socialcontent['facebook']; ?>">

</div>

</div>

<button type="submit" class="btn btnsuccess">Update</button>

<span class="text-success" style="padding-left:10px"><?php echo
$social_error; ?></span>

</form>

<form class="form-inline" method="post">

<div class="form-group">

<div class="input-group">

<div class="input-group-addon">Twitter</div>

<input type="text" name="social_twitter" class="form-control" value="<?php echo
$socialcontent['twitter']; ?>">

</div>

</div>

<button type="submit" class="btn btnsuccess">Update</button>

</form>

```

```

<form class="form-inline" method="post">

<div class="form-group">

<div class="input-group">

<div class="input-group-addon">LinkedIn</div> <input type="text" name="social_linked"
    class="form-control"      value="<?php      echo $socialcontent['linkedin']; ?>">
</div>

</div>

<button type="submit" class="btn btnsuccess">Update</button>

</form>

<form class="form-inline" method="post">

<div class="form-group">

<div class="input-group">

<div      class="input-group-addon">Instagram</div>      <input      type="text"
name="social_insta"      class="form-control"      value="<?php      echo
$socialcontent['insta']; ?>">
</div>

</div>

<button type="submit" class="btn btnsuccess">Update</button>

</form>

</div> </div> </div>

</div>

</div>

<!-- -----Home Page----- -->

<div id="home" class="switchgroup dashboard">

<div class="panel panel-default">

<div class="panel-heading">

<h3>Home</h3>

```

```

</div>

<div class="panel-body">

<!-- Slider -->

<?php
$query="select * from slider";
$result=mysqli_query($link,$query) or die("Error fetching
data.".mysqli_error($link));

?>

<div class="panel panel-default slider-content">

<div class="panel-heading">
    <h4>Slider Content<a href="#" class="pull-right"
    name="slider_add" data-toggle="modal"
    datatarget="#slider_img_add">Add    <span
    class="glyphicon glyphicon-plus"></span></a></h4>
</div>

<div class="panel-body">

<form class="form-inline" method="post" enctype="multipart/form-data">

```

```

<div class="form-group">
<div class="input-group">
<?php while($slidercontent=mysqli_fetch_assoc($result))
{
?>
" width="50" height="50"/>
<input type="hidden" name="img_id" value=""/>
</div> </div>

<div class="form-group">

<input type="file" name="sliderpic">
<button type="submit" name="sliderPicSubmit" id="<?php echo
$slidercontent['id']; ?>" class="btn btn-success glyphicon
glyphicon-pencil"></button>

<button      type="submit"      name="sliderPicDelete"
      class="btn      btndanger      glyphicon      glyphicon-
remove"></button>

</div> <?php break;
}
?>

</form>

</div> </div>

<!-- End Home Slider -->

<!-- Social Links -->

<div class="panel panel-default social-links">

<div class="panel-heading">

```

```

<h4>Social Links</h4>

<div class="panel-body">

<?php

$sql="select * from social_media where uid=1";
$result=mysqli_query($link,$sql) or die("Error fetching
data.".mysqli_error($link));
$socialcontent=mysqli_fetch_assoc($result);
mysqli_free_result($result);
?>

<form class="form-inline" method="post">

<div class="form-group">

<div class="input-group">

<div class="input-group-addon">Facebook</div> <input
type="text" name="social_facebook" class="formcontrol"
value="<?php echo $socialcontent['facebook']; ?>">

</div>

</div>

<button type="submit" class="btn
btnsuccess">Update</button>

<span class="text-success" style="padding-left:10px"><?php
echo $social_error; ?></span>

</form>

<form class="form-inline" method="post">

<div class="form-group">

<div class="input-group">

<div class="input-group-addon">Twitter</div>

```

```

<input type="text" name="social_twitter" class="form-control"
value="<?php echo $socialcontent['twitter']; ?>">
</div>

</div>

<button type="submit" class="btn
btnsuccess">Update</button>

</form>

<form class="form-inline" method="post">
<div class="form-group">
<div class="input-group">
<div class="input-group-addon">LinkedIn</div> <input
type="text" name="social_linked" class="form-control"
value="<?php echo $socialcontent['linkedin']; ?>">
</div>
</div>
<button type="submit" class="btn
btnsuccess">Update</button>

</form>

<form class="form-inline" method="post">
<div class="form-group">
<div class="input-group">
<div class="input-group-addon">Instagram</div> <input
type="text" name="social_insta" class="form-control"
value="<?php echo $socialcontent['insta']; ?>">
</div>
</div>

```

```

<button      type="submit"      class="btn
btnsuccess">Update</button>

</form>

</div>

</div>

</div> </div>

</div>

<!-- -----End Home Page----- -->

<!-- -----About page----->

<div id="about" class="switchgroup dashboard">

<div class="panel panel-default">

<div class="panel-heading">

<h3>About us</h3>

```

```

<div class="panel-body">

<!-- About us content -->

<div class="panel panel-default social-links">

<div class="panel-heading">

<h4>About us Content</h4>

</div>

<div class="panel-body">

<?php
$sql="select * from pages where page_nm='about_us'";
$result=mysqli_query($link,$sql) or die("Error fetching
data.".mysqli_error($link));
$aboutcontent=mysqli_fetch_assoc($result);
mysqli_free_result($result);
?>

<form class="form-inline" method="post">

<div class="form-group">

<textarea class="form-control" rows="10"
cols="100" name="info" id="myInfo"><?php

echo $aboutcontent['page_desc'];

?></textarea>

</div>

<br/>

<span class="text-success" style="padding-left:10px"><?php
echo $about_msg; ?></span>

<br/>

</div>

```

```

<div class="form-group">

<button      type="submit"      class="btn  btn-success"
name="about_info">Update</button>

</div>

</form>

</div>

</div> </div>

</div>

<!-- -----End About page----->

<!-- -----Reach us page----- -->

<div id="reach_us" class="switchgroup dashboard">
<div class="panel panel-default">
<div class="panel-heading">
<h3>Reach us</h3>
</div>
<div class="panel-body">
<!-- Reach us content-->
<div class="panel panel-default social-links">
<div class="panel-heading">
<h4>Address</h4>
</div>
<div class="panel-body">
<?php
</div>

```

```

$query="select * from reach_us where uid=1";
$result=mysqli_query($link,$query) or die("Error fetching
data.".mysqli_error($link));
$contactetails=mysqli_fetch_assoc($result);
mysqli_free_result($result);
?>

<form class="form-inline" method="post">

<div class="form-group reach_data">

<div class="input-group">

<div class="input-group-addon">Name</div> <input
type="text" name="reach_nm" class="form-control"
value="<?php echo $contactetails['nm']; ?>">

</div>

<br/>

<div class="form-group reach_data">

<div class="input-group">

<div class="input-group-addon">Address-1</div> <input
type="text" name="reach_add1" class="form-control"
value="<?php echo $contactetails['add1']; ?>">

</div>

</div>

<br/>

<div class="form-group reach_data">

<div class="input-group">

<div class="input-group-addon">Address-2</div> <input
type="text" name="reach_add2" class="form-control"
value="<?php echo $contactetails['add2']; ?>">

</div>

```

</div>

</div>

<div class="form-group reach_data">

<div class="input-group">

<div class="input-group-addon">City</div>

<input type="text" name="reach_city" class="form-control"
value="<?php echo \$contactetails['city']; ?>">

</div>

</div>

<div class="form-group reach_data">

<div class="input-group">

<div class="input-group-addon">Zipcode</div> <input

type="text" name="reach_zip" class="form-control"

value="<?php echo \$contactetails['zipcode']; ?>">

</div>

</div>

```

<br/>

<div class="form-group reach_data">

<div class="input-group">

<div class="input-group-addon">State</div>
<input type="text" name="reach_state" class="form-control"
value="<?php echo $contactetails['state']; ?>">

</div>

</div>

<br/>

<div class="form-group reach_data">

<div class="input-group">

<div class="input-group-addon">Contact No.</div> <input
type="text" name="contact_no" class="form-control"
value="<?php echo $contactetails['contact_no']; ?>">

</div>

</div>

<br/>

<br/>

<button type="submit" name="reach_info"
class="btn btnsuccess">Update</button>

<span class="text-success" style="padding-left:10px"><?php
echo $reach_us_msg; ?></span>

</form>

</div>

</div>

</div> </div>

</div>

<!-- -----End Reach us page----->

```

</div> </div>

</div>

</div>

<div class="modal fade" id="slider_img_add" tabindex="-1"
role="dialog" aria-labelledby="slider_img_add">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal"

arialabel="Close"><span

ariahidden="true">×</button>

<h4 class="modal-title" id="myModalLabel">Add Slider
Image</h4>

</div>

<form method="post" enctype="multipart/form-data"
class="form-inline">

<div class="modal-body">

<div class="form-group ">

<input type="file" name="sliderpicpic_add">

</div> </div>

<div class="modal-footer">

<button type="button" class="btn btn-default"

datadissmiss="modal">Close</button>

```
<button      type="submit"      name="picadd_submit"
            class="btn btnprimary">Add</button>
</div>
</form>
</div>
</div>
</div>
```

```
<script type="text/javascript">
</script>
```

```
<?php require_once('dbconclose.php');
?>
```

```
<script src="../js/jquery.js"></script>
<script src="../js/script.js"></script>
<script src="../js/bootstrap.min.js"></script>
</body>
</html>
<?php
```

```

require_once('conn.php');

$msg="";

if(isset($_POST['submit']))
{
    $unm=trim($_POST['uname']);
    $pass=trim($_POST['password']);
    $query="select * from admin";
    $result=mysqli_query($link,$query);
    $row=mysqli_fetch_assoc($result);
    $dbunm=$row['username'];
    $dbpass=$row['password']; if(mysqli_error($link))
    {
        echo mysqli_error($link); exit;
    }
    else
    {
        //if(password_verify($pass,$dbpass) && $unm==$dbunm)
        //{
            session_start();
            $_SESSION['id']=$row['id'];
            $_SESSION['uname']=$row['username'];
            header('Location:adminHome.php');
        }
        //else
        //{
            //$msg="<p style='padding:8px'>Username and Password is
            incorrect..</p>";
            //header('Location:adminHome.php');

```

```
//}
```

```
}
```

```
?>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

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

```
<meta name="viewport" content="width=device-width,  
initialscale=1">
```

```
<title>Admin Login</title>
```

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

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

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

```
</head>
```

```
<body>
```

```
<nav class="navbar navbar-inverse top-menu">
```

```
<div class="container-fluid">
```

```
<div class="navbar-header">
```

```
<a class="navbar-brand" href="#">Art Gallery</a>
```

```
</div>
```

```
<div>
```

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

```

<li><a href="#">About</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">
  <li><a href="registration.php"><span class="glyphicon glyphicon-user"></span>
    Registration</a></li>
  <li><a href="#" style="color:#fff"><span class="glyphicon glyphicon-log-in"></span>
    Login</a></li>
</ul>

</div> </div>

</nav>

<div class="container">

<div class="panel panel-primary admin-login">

<div class="panel-heading">

<h3>Login</h3>

</div>

<div class="panel-body">

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

<div class="bg-danger error_msg"><?php echo $msg;
?></div>

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

<span class="input-group-addon glyphicon glyphicon-envelope"
style="top:0"></span><input type="email" name="uname"
class="form-control"placeholder="Enter email">

</div>

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

<span class="input-group-addon glyphicon glyphicon-lock"
style="top:0"></span><input type="password"
name="password" class="form-control"
placeholder="Enter password">

</div>

```

```

<div class="form-group">
<button type="submit" name="submit" class="btn btn-primary
btn-block">Login</button>
</div>
</form>
</div> </div> </div>      <?php
require_once('dbconclose.php');
?>
</body>
</html>

```

2.sql

```

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

```

```

/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET
_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_S
ET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CON
NECTION */;
/*!40101 SET NAMES utf8mb4 */;

```

--

```

-- Database: `art_gallery`

--
-- -----
--
-- Table structure for table `admin`
--
CREATE TABLE `admin` (
  `id` int(11) NOT NULL,
  `full_nm` varchar(100) NOT NULL,
  `username` varchar(100) NOT NULL,
  `password` char(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `admin`
--

INSERT INTO `admin` (`id`, `full_nm`, `username`, `password`)
VALUES
(5, 'Lyndon Bermoy', 'serbermz', 'admin');

--
-- -----
--
-- Table structure for table `pages`
--
CREATE TABLE `pages` (
  `p_id` int(11) NOT NULL,
  `page_nm` varchar(50) NOT NULL,
  `page_desc` text NOT NULL

```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--
```

```
-- Dumping data for table `pages`
```

```
--
```

```
INSERT INTO `pages` (`p_id`, `page_nm`, `page_desc`)
```

```
VALUES
```

```
(1, 'deshboard', ''),
```

```
(2, 'home', ''),
```

```
(3, 'about_us', 'This is demo content.....');
```

```

-----

--

-- Table structure for table `reach_us`

--

CREATE TABLE `reach_us` (
  `uid` int(11) NOT NULL,
  `nm` varchar(100) NOT NULL,
  `add1` varchar(255) NOT NULL,
  `add2` varchar(255) NOT NULL,
  `city` varchar(50) NOT NULL,
  `state` varchar(50) NOT NULL,
  `zipcode` int(20) NOT NULL,
  `contact_no` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Dumping data for table `reach_us`

--

INSERT INTO `reach_us` (`uid`, `nm`, `add1`, `add2`, `city`,
  `state`, `zipcode`, `contact_no`) VALUES
(1, 'Bermz ISware Solutions', 'Surigao City, Philippines', '',
'Surigao', 'Surigao del Norte', 8400, '09079373999');

-----

--

-- Table structure for table `slider`

--

CREATE TABLE `slider` (
  `id` int(11) NOT NULL,

```

```

`img_nm` varchar(50) NOT NULL,
`path` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `slider`
--

INSERT INTO `slider` (`id`, `img_nm`, `path`) VALUES
(9, 'Penguins.jpg', 'images/SliderPenguins.jpg');

-- -----
--
-- Table structure for table `social_media`
--
--CREATE TABLE `social_media` (
`uid` int(11) NOT NULL,
`facebook` varchar(255) NOT NULL,
`twitter` varchar(255) NOT NULL,
`linkedin` varchar(255) NOT NULL,
`insta` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `social_media`
--

INSERT INTO `social_media` (`uid`, `facebook`, `twitter`,
`linkedin`, `insta`) VALUES
(1, 'https://www.facebook.com/donnieboiii',
'www.twitter.com/art_gallery_serbermz',
'https://www.linkedin.com/in/lyndon-bermoy-0b50741a7/',
'www.instagram.com/donniebhoi'); -----

```

--

-- Table structure for table `user\_reg`

--

```
CREATE TABLE `user_reg` (  
  `fname` varchar(100) NOT NULL,  
  `lname` varchar(100) NOT NULL,  
  `gender` varchar(20) NOT NULL,  
  `contact` varchar(20) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `password` varchar(255) NOT NULL,  
  `add1` varchar(100) NOT NULL,  
  `add2` varchar(100) NOT NULL,  
  `add3` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

--

-- Dumping data for table `user\_reg`

--

```
INSERT INTO `user_reg` (`fname`, `lname`, `gender`,  
  `contact`, `email`, `password`, `add1`, `add2`, `add3`) VALUES  
('Bermoy', 'Lyndon', 'Male', '09079373999',  
'serbermz2020@gmail.com', 'admin', 'Surigao City, Philippines',  
, '');
```

--

-- Indexes for dumped tables

--

--

-- Indexes for table `admin`

--

```
ALTER TABLE `admin`  
ADD PRIMARY KEY (`id`);
```

--

-- Indexes for table `pages`

--

```
ALTER TABLE `pages`  
ADD PRIMARY KEY (`p_id`);
```

--

-- Indexes for table `reach\_us`

--

```
ALTER TABLE `reach_us`  
ADD PRIMARY KEY (`uid`);
```

--

-- Indexes for table `slider`

--

```
ALTER TABLE `slider`  
ADD PRIMARY KEY (`id`);
```

--

-- Indexes for table `social\_media`

--

```
ALTER TABLE `social_media`  
ADD PRIMARY KEY (`uid`);
```

```

--
-- Indexes for table `user_reg`
--
ALTER TABLE `user_reg`
ADD PRIMARY KEY (`email`);
-- AUTO_INCREMENT for dumped tables
--
--
-- AUTO_INCREMENT for table `admin`
--
ALTER TABLE `admin`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT for table `pages`
--
ALTER TABLE `pages`
MODIFY `p_id` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `reach_us`
--
ALTER TABLE `reach_us`
MODIFY `uid` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT for table `slider`

```

--

```
ALTER TABLE `slider`
```

```
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,
```

```
AUTO_INCREMENT=10;
```

```
COMMIT;
```

```
/*!40101 SET
```

```
CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLI
```

```
ENT */;
```

```
/*!40101 SET
```

```
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_
```

```
RESULTS */;
```

```
/*!40101 SET
```

```
COLLATION_CONNECTION=@OLD_COLLATION_CONNEC
```

```
TION */;
```

```
<?php
```

2.1 index.php

```
$page="contact_us"; $title="Reach us";
```

```
require_once('header.php');
```

```
?>
```

```
<div class="container-fluid">
```

```
<div class="row reach_us_data">
```

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

```
<div class="panel panel-default">
```

```
<div class="panel-heading">Leave Comment</div>
```

```
<div class="panel-body">
```

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

<div class="form-group">

<label for="inputName" class="col-sm-2 control-label">Name
<span>*</span></label>

<div class="col-sm-10">

<input type="text" class="form-control" placeholder="Enter
Name">

</div> </div>
```

```

<div class="form-group">
<label for="inputEmail" class="col-sm-2 control-label">Email
    <span>*</span></label>

<div class="col-sm-10">

<input type="email" class="form-control" placeholder="Email">

</div> </div>

<div class="form-group">
<label for="inputComment" class="col-sm-2 controllabel">Comment

</label>

<div class="col-sm-10">

<textarea class="form-control" rows="3"></textarea>

</div> </div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10"> <button
type="submit"                class="btn
btndefault">Comment</button>

</div> </div>

</form>

</div>

</div>

</div>

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

<?php
$query="select      *      from      reach_us      where      uid=1";
$result=mysqli_query($link,$query)      or      die("Error      fetching
data.".mysqli_error($link));
$contactetails=mysqli_fetch_assoc($result); mysqli_free_result($result);

?>

```

```
<div class="panel panel-default">
```

```
<div class="panel-heading">Address</div>
```

```
<div class="panel-body">
```

```
<div class="jumbotron">
```

```
<p><?php echo $contactetails['nm']; ?></p>
```

```
<p><?php echo $contactetails['add1']; ?></p>
```

```
<p><?php if($contactetails['add2']!="")
```

```
{
```

```
echo $contactetails['add2'];
```

```
}
```

```
else echo "";
```

```
?></p>
```

```
<p><?php echo $contactetails['city']; ?></p>
```

```
<p><?php echo $contactetails['zipcode']; ?></p>
```

```
<p><?php echo $contactetails['state']; ?></p>
```

```
<br/>
```

```
<p>Contact No: <?php echo $contactetails['contact_no'];
```

```
?></p>
```

```
</div> </div>
```

```
</div> </div>
```

```
</div>
```

```
<?php
```

```
require('footer.php');
```

```
?>
```

```

<?php
$page="index";      $title="Home";

require_once('header.php'); ?>

<div class="container-fluid">

<div class="row slider">

<div class="col-lg-14">

<div id="myCarousel" class="carousel slide" data-ride="carousel">
<ol      class="carousel-indicators">      <li
    datatarget="#myCarousel" data-slide-to="0" class="active"></li>
<li  data-target="#myCarousel"  data-slide-to="1"></li>  <li  data-
target="#myCarousel" data-slide-to="2"></li>
</ol>

<div class="carousel-inner" role="listbox">

<div class="item active">



</div>

<div class="item">



</div>

<div class="item">



</div>

</div>

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

<span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>

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

```

```
</a>
<a class="right carousel-control" href="#myCarousel" role="button" data-
slide="next">
<span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>

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

</div>
```

```
<div class="row home_info">
<div class="col-md-9 recent_product">
<div class="panel panel-default">
<div class="panel-heading">Recent Products</div>
<div class="panel-body">
<div class="container recent_product_container">
<div class="row recent_img">
<div class="col-md-4"></div> <div class="col-md-4"></div>
<div class="col-md-4"></div>
</div>
<div class="row recent_img_desc">
<div class="col-md-4">Mayon Volcano</div>
<div class="col-md-4">Mona Lisa</div>
<div class="col-md-4">Krishna with Gopi's</div>
```

```

</div>

<div class="row recent_img_desc">
<div class="col-md-4"><button type="button" class="btn-addcart">Add
Cart</button></div>
<div class="col-md-4"><button type="button" class="btn-addcart">Add
Cart</button></div>

<div class="col-md-4"><button type="button" class="btn-addcart">Add
Cart</button></div>

</div>

</div> </div>

</div> </div>

<div class="col-md-3 my_cart">
<h3>MyCart</h3>
<div class="cart_items">
<ol>
<li> <span
class="my_cart_items_title">Mayon Volcano</span> <span
class="my_cart_items_price">Php 5000</span></li>
</ol>
</div> </div>

</div>

<?php

require('footer.php');

?>

```