MODERN ART GALLERY

Mini Project Report

Submitted by

Anagha P

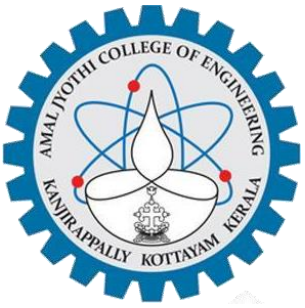
Reg. No.: AJC22MCA-2016

In Partial fulfillment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS

(MCA TWO YEAR)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2023-2024

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**MODERN ART GALLERY"** is the bona fide work of **ANAGHA P (Regno: AJC22MCA-2016)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2023-24.

Dr. Paulin Paul

Internal Guide

Ms. Meera Rose Mathew

Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I hereby declare that the project report **"MODERN ART GALLERY"** is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2023-2024.

Date: **ANAGHA P**

KANJIRAPPALLY **Reg: AJC22MCA-2016**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for his valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Dr. Paulin Paul** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

ANAGHA P

ABSTRACT

.

The art gallery is an e-Platform meant to be a collaborative platform to facilitate artists from diverse domain. Irrespective of the geographical location, exchange ideas, views and to promote their art pieces over the internet for their financial purposes and promoting cross cultural art in a more effective way. The proposed system has three modules, admin, artist and customer. The admin can manage artist and customer, initiate auction schedule and also can manage events. An artist is a registered user with provision to promote his/her art work in a secure way, buy art work with payment, can blog to participate and dominate discussions Pertaining to interested art fields can propose and initiate art exhibitions facilitating for interested parties to make secure transaction of art work with payment. The art uploads can be liked and shared within and outside. The customer can view the uploaded art work, make online secure purchase, enroll in exhibition and also can like and share. Out of this overall proposed system, except auction, payment and Copyright the remaining functionalities will be completed in the mini project.

CONTENT

List of Abbreviation

IDE	Integrated Development Environment
HTML	Hyper Text Markup Language.
CSS	Cascading Style Sheet
SQLite	Relational Database Management System
UML	Unified Modeling Language
AJAX	Asynchronous JavaScript and XML
JS	Java Script

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The proposed Modern Art Gallery e-Platform is designed as a collaborative space for artists worldwide. With three key roles – Admin, Artist, and Customer – the platform aims to facilitate the sharing of ideas and artworks. Admins manage user profiles and events, while artists showcase their work and engage in discussions. Customers can explore artworks, join exhibitions, and interact by liking and sharing. The platform seeks to create a global community for artists and art enthusiasts, fostering cross-cultural exchange and providing a secure space for artistic expression.

1.2 PROJECT SPECIFICATION

Project specifications are the foundational blueprint for the online art gallery project. They succinctly outline key elements, such as system architecture, module functionalities, user interactions, security measures, and future enhancements. These specifications act as a clear roadmap, ensuring a structured approach to development by defining goals, technical aspects, and the framework for successful execution.

- ✧ **User Management**
  Create and delete artist and customer profiles also can Manage user access
- ✧ **User Authentication:**

  Admin enjoys secure access through a login system.
- ✧ **Artwork Management:**
  Review artwork submissions from artists and Remove artwork listings.
- ✧ **Event Management:**
  Create and manage art-related events.
- ✧ **Customer Interaction:**

  Track user interactions, and enhance the overall customer experience this platform by likes and shares
- ✧ **Technologies Used:**

  Frontend: HTML, CSS, JavaScript, AJAX.

  Backend: Django.

  Database: SQLite.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

In the realm of art, the proposed Online Art Gallery is a collaborative e-platform connecting artists worldwide. The system comprises three key modules: Admin, Artist, and Customer.

Artists can showcase their work, participate in events, and share insights through personalized blogs. Customers can explore diverse artworks, connect with artists, and enrich their art appreciation experience.

The system is designed for simplicity and user-friendliness, focusing on creating an intuitive space that encourages artistic expression and appreciation. The goal is to provide a straightforward platform for artists to showcase their talent and for art enthusiasts to engage with diverse artworks, fostering a dynamic and inclusive art culture.

2.2 EXISTING SYSTEM

2.2.1 NATURAL SYSTEM STUDIED

In the traditional art exhibition system, artists rely on physical galleries to showcase their work, limiting their reach to a local or regional audience. The geographical constraints of this system can impede artists from gaining exposure on a global scale. Additionally, the traditional approach may not effectively foster direct interactions between artists and customers, as the audience is restricted to those who can attend the physical exhibition. The absence of a digital platform also means limited opportunities for artists to engage with a broader community, share insights about their work, or receive immediate feedback. In contrast, an online art gallery transcends these limitations, offering artists a borderless canvas to present their creations and enabling direct and instantaneous connections with a diverse and global audience.

2.2.2 DESIGNED SYSTEM STUDIED

The designed system, envisioned as a sophisticated online art platform, aims to seamlessly integrate user management, artwork management, event management, and artist management for a comprehensive art-sharing experience. By providing a centralized hub for artists and art enthusiasts, this system ensures a user-friendly interface that simplifies the process of uploading, showcasing, and exploring artworks. With robust user management features, artists can efficiently maintain their profiles and engage with the community, while event management functionalities empower users to participate in or organize various art-related events. Additionally, artist management tools facilitate collaboration, support, and networking opportunities within the artistic community. This system is designed to be versatile and adaptable to diverse artistic needs, fostering a vibrant online ecosystem for the exchange of ideas and creative expressions.

2.3 DRAWBACKS OF EXISTING SYSTEM

- **Lack of User Management:** The existing system may not have a structured user management system, making it challenging to handle different roles and responsibilities.
- **Limited Artwork Management:** Without a proper artwork management system, the existing platform may not provide artists with a secure way to promote their work or customers with an organized way to explore and interact with artworks.
- **Inefficient Event Management:** The current system may lack features for organizing and managing art-related events, limiting opportunities for artists and customers to engage.

2.4 PROPOSED SYSTEM

The proposed system envisions a well-structured online art platform that focuses on key functionalities like user management, artwork management, event management, and artist management. By emphasizing simplicity and user-friendliness, the system aims to create an inclusive environment for artists and art enthusiasts. The platform becomes more accessible, encouraging a seamless exchange of artistic ideas. This streamlined approach fosters a vibrant community where artists can showcase their work, engage with audiences, and participate in events without the additional layers of financial transactions and legal intricacies. The system's design prioritizes an intuitive user experience, allowing users to concentrate on the essence of art appreciation and collaboration within a supportive online space.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- **Enhanced User Management:** The proposed system allows for the creation and deletion of artist and customer profiles, with improved user access management, providing a more organized platform.
- **Streamlined Artwork Management:** The proposed system facilitates artists in uploading and managing their artwork listings securely, enhancing the overall presentation of artworks for customers.
- **Improved Event Management**: The proposed system introduces a more efficient way to create, manage, and promote art-related events, creating more opportunities for artists and customers to engage.
- **Structured Artist Management:** The proposed system offers better support and assistance to artists, allowing them to manage their profiles and blogs more effectively.
- **Enhanced Customer Interaction:** With features like likes and shares, the proposed system encourages customer engagement and interaction with artworks, creating a more dynamic and interactive art platform.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility is the degree to which a project can be carried out successfully. A feasibility study is conducted to assess the solution's viability, which establishes whether it is viable and implementable in the program. The feasibility study considers details like the availability of resources, software development costs, the advantages of the software to the business once it is built, and the costs associated with maintaining it. The outcome of the feasibility study should be a report recommending whether the requirements engineering, and system development process should be continued. A system is of no real value to a corporation if it does not serve its goals. Even though this may seem obvious, many organizations create systems that do not support their goals, either because they lack a clear statement of these goals, because they fail to specify the system's business requirements, or because other organizational or political factors have an impact on the procurement of the system.

3.1.1 Economical Feasibility

The economic feasibility of your online bookstore project is critical to assess its financial viability and potential profitability. Conducting a thorough evaluation of various financial aspects will help determine if the project is economically sound. Begin with a comprehensive cost-benefit analysis, comparing the total expenses of development, operation, and maintenance against the expected benefits and revenue generation from book sales. Estimate the initial development costs, ongoing operational expenses, and potential revenue from book sales to calculate the projected return on investment (ROI) and the break-even point. Analyze the profit margin per book sale to ensure sustainability and competitiveness. Additionally, conduct market research to understand the demand, competition, and growth potential in the book industry. By conducting a comprehensive economic feasibility analysis, you can make informed decisions and assess the project's financial potential for long-term success.

3.1.2 Technical Feasibility

The technical feasibility of your online bookstore project is crucial to determine whether it can be practically implemented from a technological standpoint. Evaluating various aspects is essential for its success. Firstly, ensure that your organization has the necessary technology infrastructure or can acquire it to support the platform, including web hosting, databases, and server capacity. Check for skilled software developers who can build and maintain the system effectively. Integration with third-party services like payment gateways and shipping providers must be seamless. The architecture should be scalable to accommodate increasing traffic and data as the

platform grows. Security measures to protect user data and transactions must be robust. Ensure the platform is mobile-friendly for users on various devices, and its performance is fast and responsive. Compliance with data protection and privacy regulations is essential. Lastly, evaluate the financial resources required for development, hosting, maintenance, and support. A comprehensive assessment of technical feasibility will help identify potential challenges and enable you to make informed decisions to create a successful online bookstore platform.

3.1.3 Behavioral Feasibility

The Behavioral feasibility of your online bookstore project is essential to assess whether the proposed system can be effectively integrated into the organization's existing operations. Evaluating various aspects will help determine the project's viability and success. Firstly, ensure that the necessary resources, including human capital, technology, and infrastructure, are available or can be obtained within the project's time frame and budget. Consider the skill set of the organization's staff and identify the need for additional training or hiring. Evaluate how well the new system will integrate with existing processes, such as inventory management and order processing. Plan for change management to address potential resistance to organizational changes resulting from the system implementation. Ensure that third-party services or suppliers can support operational requirements. Verify compliance with legal and regulatory aspects related to online sales and data protection. Gather user feedback to gauge acceptance and identify areas for improvement. Lastly, analyse operational costs to ensure alignment with the organization's budget. A thorough assessment of operational feasibility will help ensure the successful integration of the online bookstore platform into the organization's existing operations.

3.1.4 Feasibility Study Questionnaire

1.Project Overview?

The Modern Art Gallery e-Platform is an innovative online space envisioned as a collaborative hub for artists across diverse domains. Its primary goal is to transcend geographical barriers, fostering the exchange of ideas, views, and artworks within the global artistic community. This digital platform provides artists with the means to showcase their creations securely and promotes cross-cultural artistic interactions. The system is structured around three key modules: Admin, Artist, and Customer. The Admin module empowers administrators with comprehensive control over user management, artwork submissions, event organization, and artist support. Artists, as registered users, can leverage the platform to manage their profiles, upload and promote artwork, participate in events, and propose exhibitions. Customers, on the other hand, can explore and engage with artworks while also participating in artist-organized exhibitions. Overall, the Art

Gallery e-Platform stands as a dynamic online ecosystem aimed at bridging artistic communities and enhancing the accessibility of art on a global scale.

2.To what extend the system is proposed for?

The proposed Modern Art Gallery e-Platform is designed as a comprehensive and inclusive digital space for artists and art enthusiasts, transcending geographical constraints. The system is envisioned to facilitate a wide range of activities and interactions among its three key modules: Admin, Artist, and Customer. The Admin module provides overarching control, overseeing user management, artwork submissions, event organization, and artist support. Artists, as registered users, are empowered to securely showcase their creations, engage with the community through events and exhibitions, and manage their profiles. The overarching goal of the platform is to foster a collaborative environment, allowing artists to share their work, ideas, and experiences across cultural boundaries. While the current scope of the project acknowledges certain limitations, the system lays the groundwork for future expansions and enhancements. Overall, the Modern Art Gallery e-Platform is proposed as a dynamic and evolving digital ecosystem that aims to connect, promote, and enhance the accessibility of art on a global scale.

5. Identify the users in your project?

The Modern Art Gallery involves two types of users: Artist, customers and admins.

6. Who owns the system?

Administrator

8.Details of person that you have contacted for data collection.

Name: Dr. Paulin Paul

Position: Asst. Professor(Department of Computer Application)

Amal Jyothi College of Engineering

9.Questionnaire to collect details about the project?

- **Are online transactions available for payments?**

-Yes, online transactions are available for payments.

- **How sales details are stored and maintained?**

-Sales details are accounted in a sales book manually and from the billing system.

- **What are the main objectives you hope to achieve with the online art gallery?**

-Primary objectives include creating a profitable platform for artists to showcase their work and providing art enthusiasts with a curated collection of unique pieces.

- **How the price of the Art is Determined?**

-The price of the artwork varies according to the artworks of the artist.

- **How sales details are stored and maintained?**

-Sales details are accounted in a sales book manually and from the billing system.

- **How do you maintain the artworks?**

-The artworks of each artist is displayed according to their prices and the customer canpurchase the artworks according to their wish.

- **Does the system ensure high security for storing user information?**

-Yes, the system ensures high security for storing user information.

- **Which all artworks are mostly preferred?**

-Mostly preferred artworks are paintings, sculptures, photography etc.

- **Can users view available items in different categories?**

-Yes, users can view all items available in specific categorie

3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Ryzen 7

RAM - 8GB

Hard disk - 512 SSD

3.2.2 Software Specification

Front End	HTML5, CSS, Bootstrap
Back End	Django, SQLite
Database	SQLite
Client on PC	Windows 7 and above.
Technologies used	Django, HTML5, AJAX, Bootstrap, JS, jQuery

3.3 SOFTWARE DESCRIPTION

3.3.1 DJANGO

Django, a leading open-source web framework, epitomizes the pinnacle of modern web development with its efficiency and versatility. Built on Python, Django follows the Model-View-

Controller architecture, prioritizing simplicity and flexibility. Noteworthy features include its Object-Relational Mapping system, streamlined URL routing, and a user-friendly template engine. The built-in admin interface simplifies data management, while robust security measures and middleware support enhance application integrity. Django scales effortlessly, accommodating growing datasets and traffic demands. Its REST Framework extension further extends its utility to API development. Supported by an active community and extensive documentation, Django stands as a powerful toolkit, seamlessly shaping the development of dynamic and secure web applications for diverse purposes, from content management systems to Restful API. In essence, Django empowers developers with a comprehensive and adaptable framework for crafting sophisticated and salable web solutions**.**

3.3.2 SQLite

SQLite, a lightweight and server less relational database management system, is celebrated for its simplicity, portability, and efficiency. Operating as a self-contained, single-file database, SQLite requires minimal setup and eliminates the need for a separate server process. This design makes it a preferred choice for embedded systems, mobile applications, and small to medium-scale projects. With zero configuration and a server less architecture, SQLite streamlines the database management process. It supports ACID transactions, ensuring data integrity and reliability, even in the face of system interruptions. The dynamic typing feature allows flexible data storage, accommodating various data types within the same column.

Noteworthy is SQLite's cross-platform compatibility, making it versatile across different operating systems and environments. Its wide adoption is evident in applications ranging from mobile apps to web browsers, where it is utilized for local data storage. SQLite stands out as a go-to solution for projects requiring a lightweight, self-contained, and easily deployable database system, embodying simplicity without compromising on functionality and reliability.

CHAPTER 4
SYSTEM DESIGN

4.1 INTRODUCTION

Any designed system or product's development process begins with the design phase. An efficient system depends on a well-executed design, which is a creative process. It entails utilizing a variety of techniques and concepts to completely specify a procedure or system for it to be implemented. Regardless of the development paradigm used in software engineering, the design phase is essential. It seeks to produce the architectural detail needed to design a system or product and acts as the process's technical backbone.

To maximize every aspect of efficiency, performance, and accuracy, this program underwent a comprehensive design phase. A user-oriented document is converted into a document for programmers or database workers throughout the design phase.

4.2 UML DIAGRAM

Software engineering uses the Unified Modeling Language (UML), a standardized visual language, to model, develop, and document software systems. UML diagrams provide a concise and organized approach to represent different facets of a software system, serving as a common communication tool for developers, stakeholders, and designers. There are many different varieties of UML diagrams, including class diagrams, sequence diagrams, use case diagrams, and more. Each is designed to communicate a particular piece of knowledge about the design, operation, and interactions of the system. UML diagrams are essential to the software development process because they help with the visualization, analysis, and design of complex systems, which in turn makes the process more effective and efficient.

- Use case diagram
- Sequence diagram
- State diagram
- Activity diagram
- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a visual representation of the interactions between system components. An approach for identifying, outlining, and organizing system requirements is called a use case. The word "system" here refers to a thing that is being created or run, like a website for mail- order

product sales and services. UML (Unified Modeling Language), a standard language for the modelling of real-world objects and systems, uses use case diagrams. The planning of general requirements, the validation of a hardware design, the testing and debugging of a software product in development, the creation of an online help reference, or the completion of a job focused on customer support are all examples of system objectives. For instance, use cases in a product sales context can involve customer service, item ordering, catalogue updating, and payment processing. There are four elements in a use case diagram.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases. Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we must use the following guidelines to draw an efficient use case diagram.
- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Figure 1: Use Case diagram

4.2.1 SEQUENCE DIAGRAM

A sort of Unified Modeling Language (UML) diagram called a sequence diagram is used in software engineering to depict the communications and interactions that take place across time between various system components or objects. It gives a live view of the interactions between objects as they work together to complete a certain job or situation.

Objects are shown as lifelines in a sequence diagram, and vertical lines indicate their existence over time. The flow of calls or messages between objects is shown by arrows and lines between lifelines. These diagrams are very helpful for demonstrating the interactions within a particular use case, assisting in determining the sequence of operations and the functions that each object performs.

Sequence diagrams are essential for understanding the behavior of a system and for ensuring that the software functions as intended, as they offer a clear and detailed depiction of how different components or objects collaborate to achieve a particular functionality.

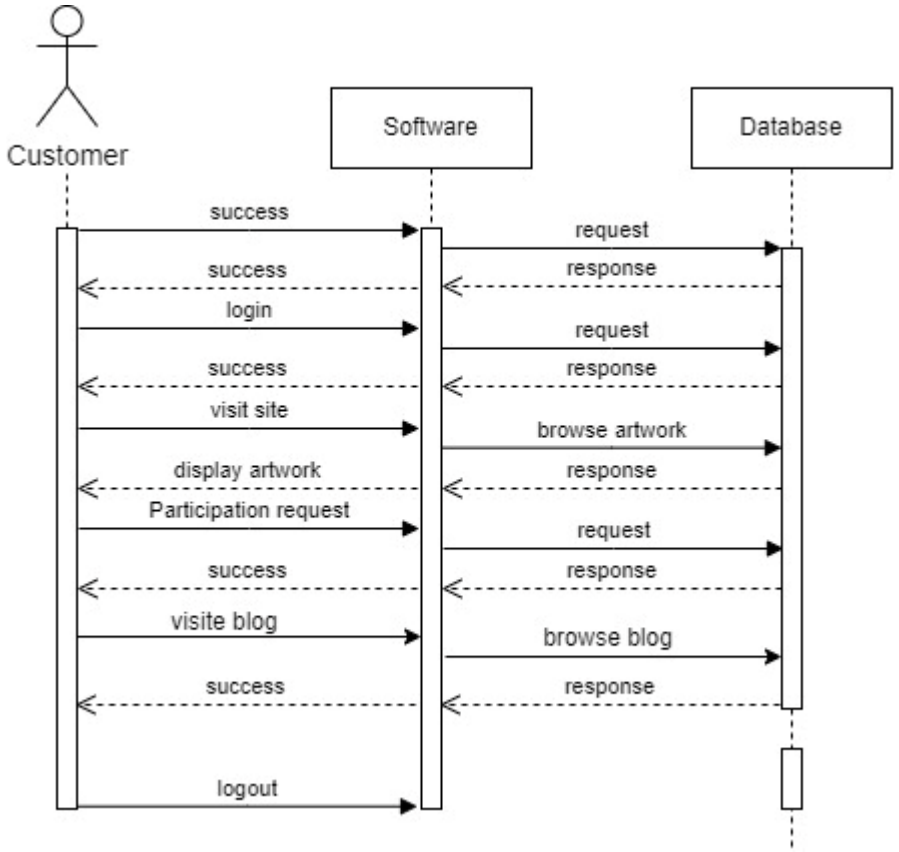


Figure 2: Sequence diagram

4.2.2 State Chart Diagram

A state diagram, also known as a state machine diagram or state chart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modeling Language (UML). In this context, a state defines a stage in the evolution or behavior of an object, which is a specific entity in a program or the unit of code representing that entity. A state diagram resembles a flowchart in nature; however, a flowchart shows the processes within a system that alters the state of an object rather than the actual state changes themselves. The first step to creating a state chart diagram is identifying the initial and final states of a system. Then, all the possible existing states are placed in relation to the beginning and the end. Lastly, all of the events that trigger state changes are labeled as transition elements.

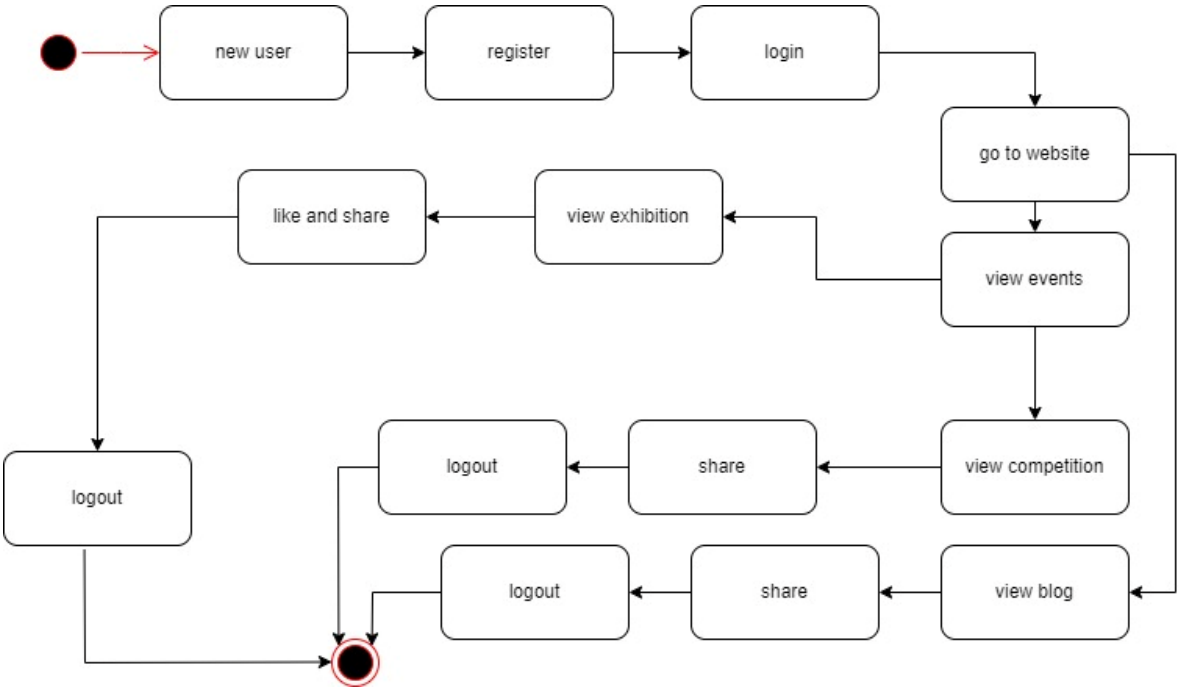


Figure 3: State Chart diagram

4.2.2 Activity Diagram

An activity diagram is essentially a flowchart that shows how one activity leads to another. The action might be referred to as a system operation. One operation leads to the next in the control flow. This flow may be parallel, contemporaneous, or branched. Activity diagrams use many features, such as fork, join, etc., to cope with all types of flow control. An activity diagram is a behavioral diagram i.e., it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

Activity diagram is another important behavioral diagram in UML diagram to describe dynamic

aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

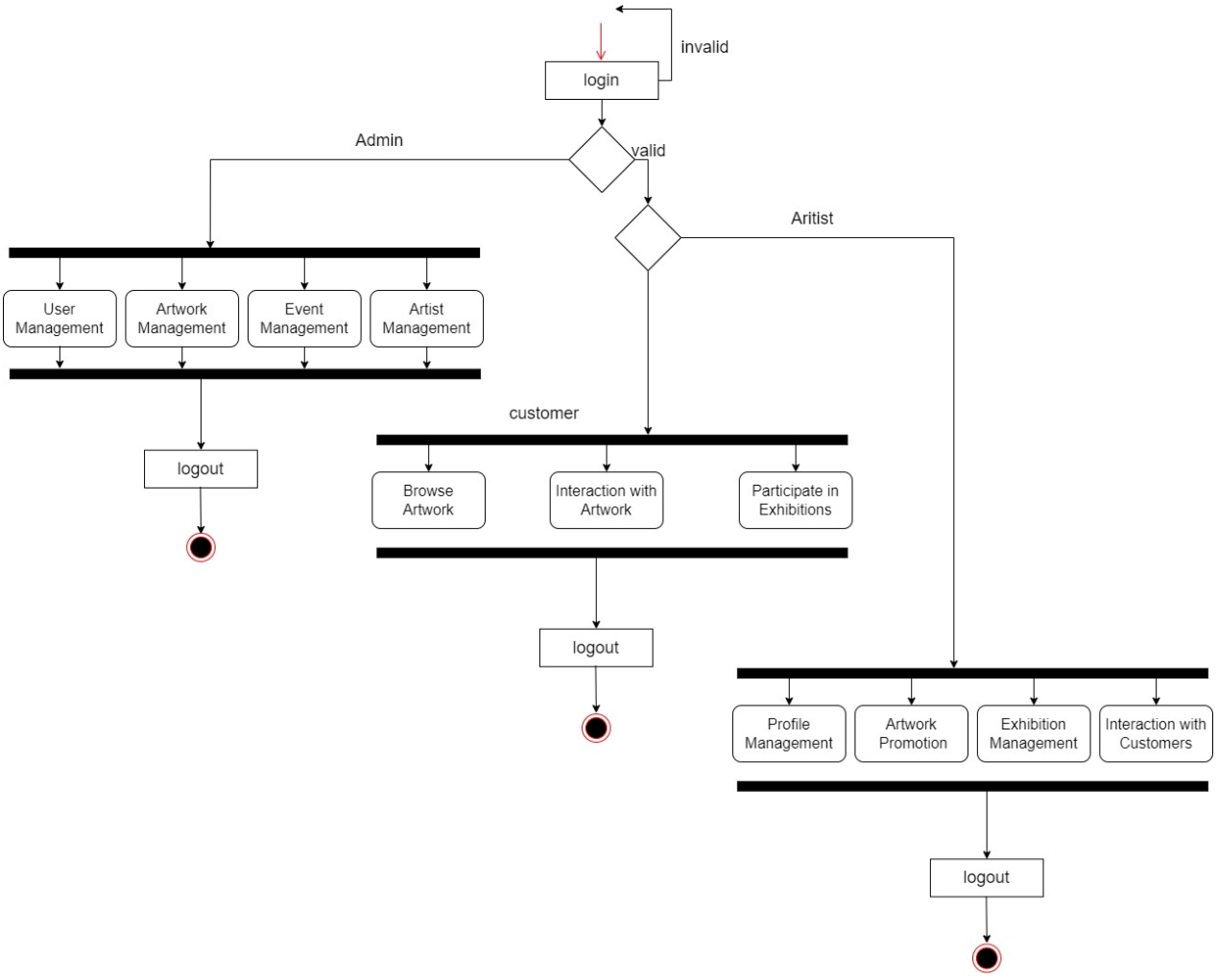


Figure 4: Activity diagram

4.2.3 Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling. Class diagrams are the blueprints of your system or subsystem. You can use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide. Class diagrams are useful in many stages of system design. In the analysis stage, a class diagram can help you to understand the requirements of your problem domain and to identify its components. In an object-oriented software project, the class diagrams that you create during the early stages of the project contain classes that often translate into actual software classes and objects when you write code. Later, you can refine your earlier analysis and conceptual models into class diagrams that show the specific parts of your system, user interfaces, logical implementations, and so on. Your class diagrams then become a snapshot that describes exactly how your system works, the relationships between system components at many levels, and how you plan to implement those components.

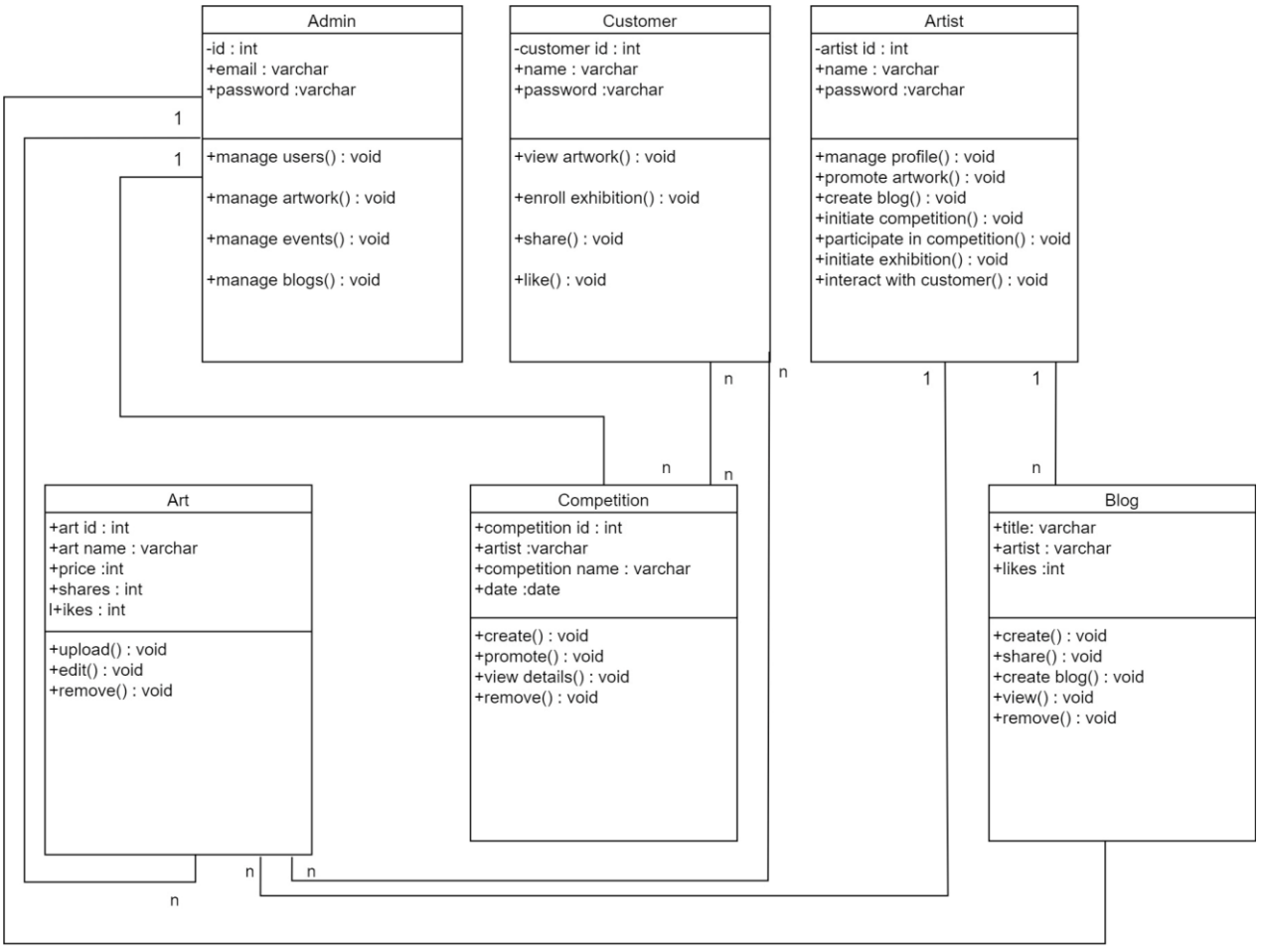


Figure 5: Class diagram

4.2.4 Object Diagram

Since class diagrams are the source of object diagrams, class diagrams are a prerequisite for object diagrams. An instance of a class diagram is represented by an object diagram. Class and object diagrams both use the same fundamental ideas. The static view of a system is also represented by object diagrams, but this static view represents a momentary snapshot of the system. To represent a group of items and their connections as an instance, object diagrams are employed

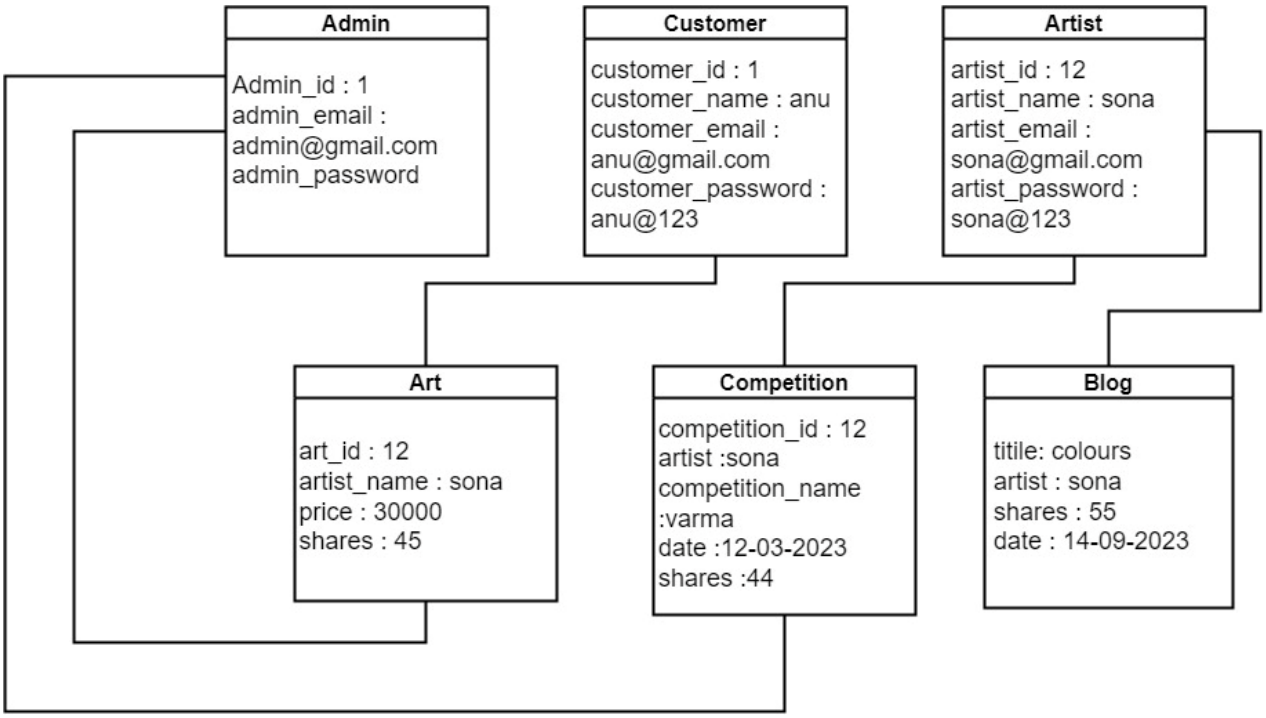


Figure 6: Object diagram

4.2.5 Component Diagram

UML diagrams are used to represent and describe the behavior of object-oriented systems, assisting in visualization, explanation, and thorough documentation. Particularly in class diagrams, where they record the static behavior of a system, these diagrams play a vital role. On the other hand, graphics can impair the efficiency of real multitasking systems. To maintain coordination, each part within the broader process has a distinct duty and only communicates with other essential components. A component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.

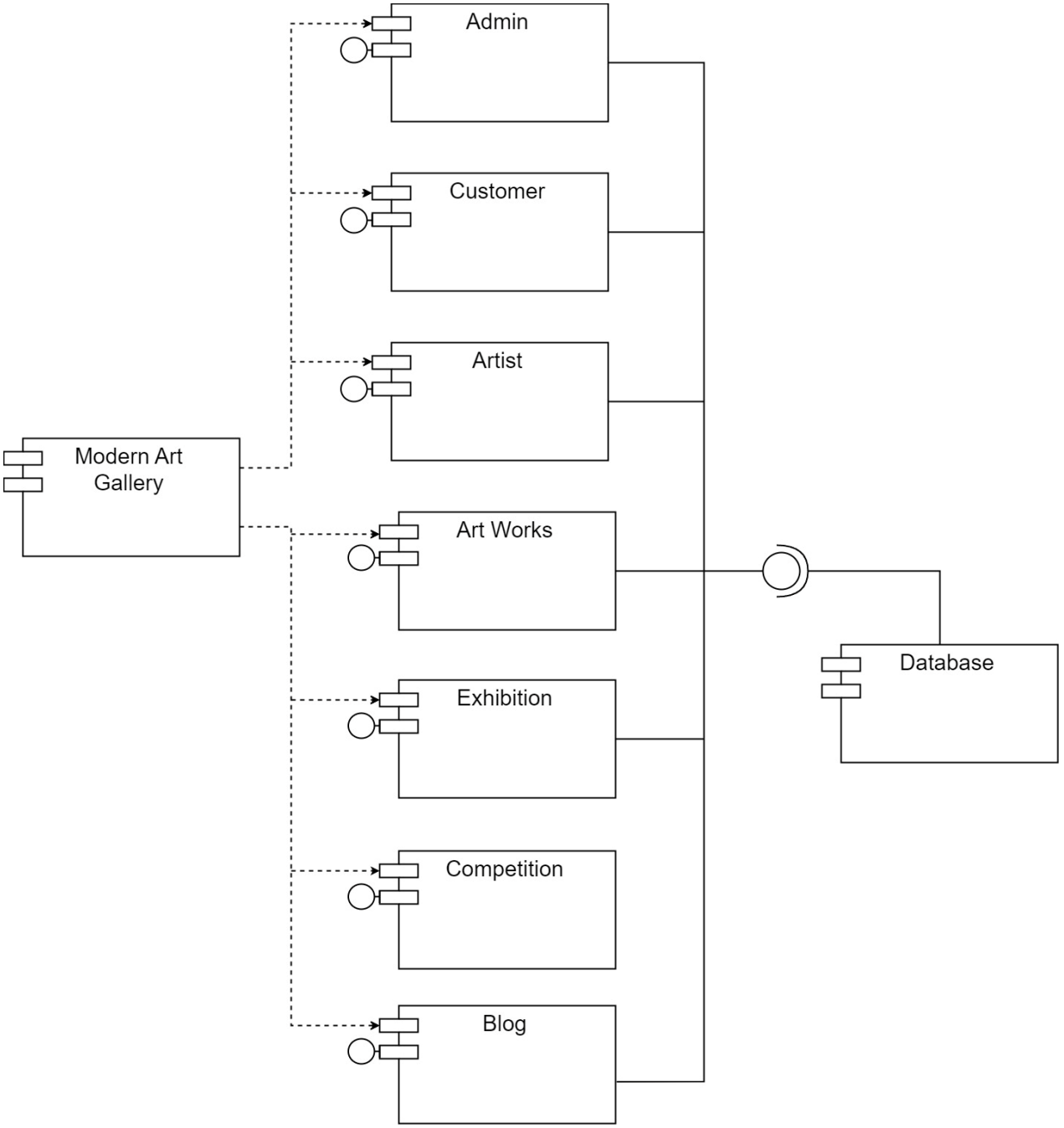


Figure 7: Component diagram

4.2.8 Deployment Diagram

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships. It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths. In contrast to other UML diagram types, which primarily depict the logical components of a system. The

deployment diagram does not focus on the logical components of the system, but it put its attention on the hardware topology.

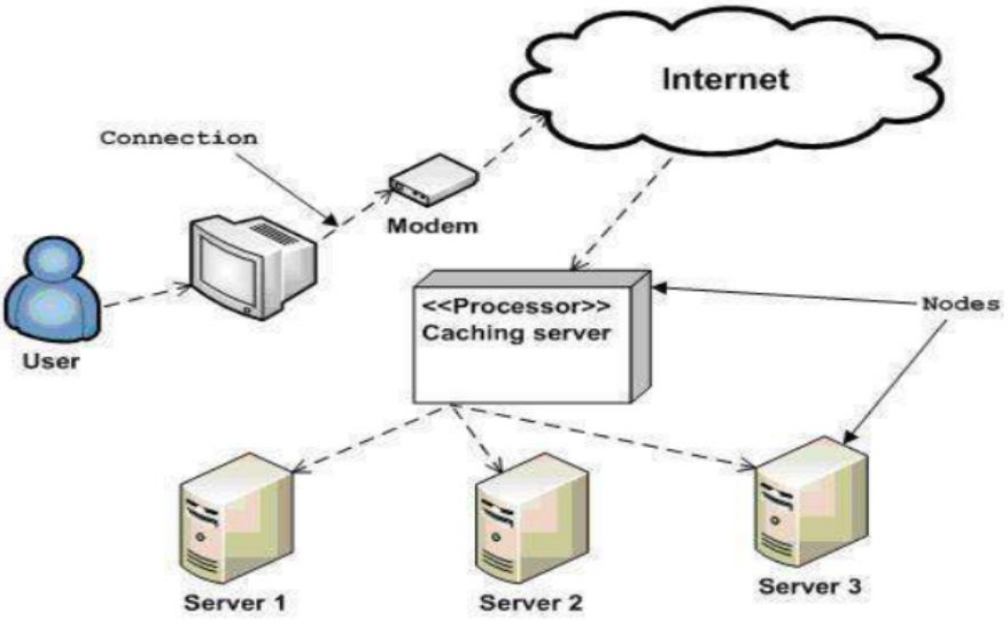
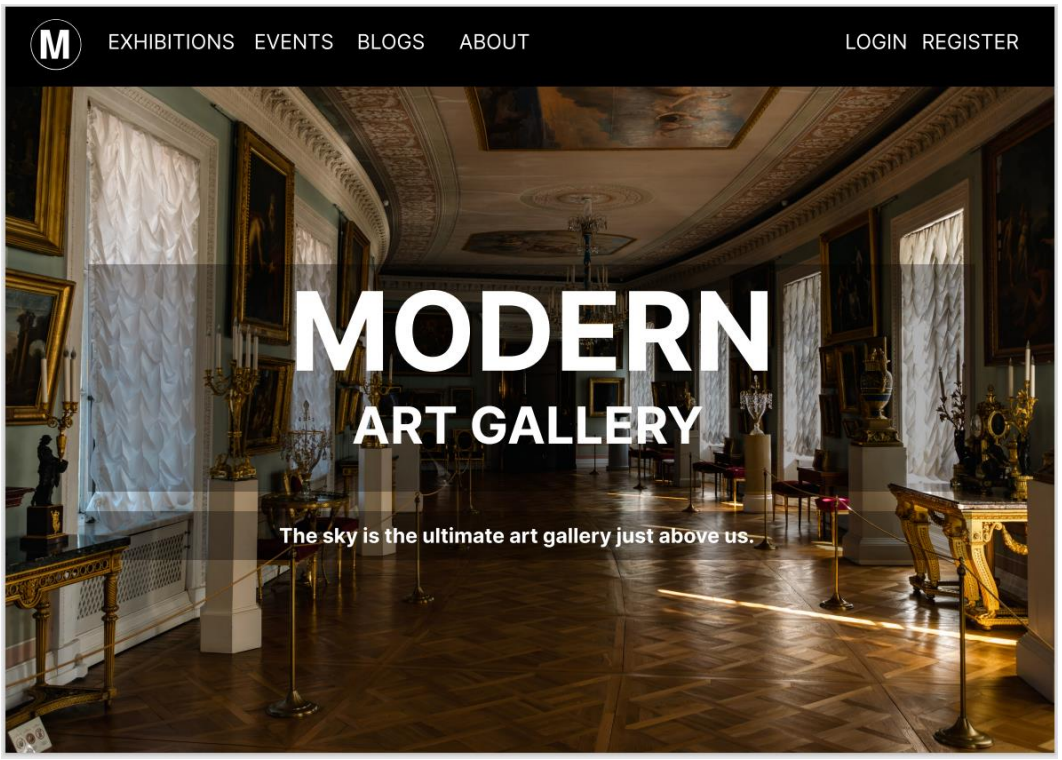


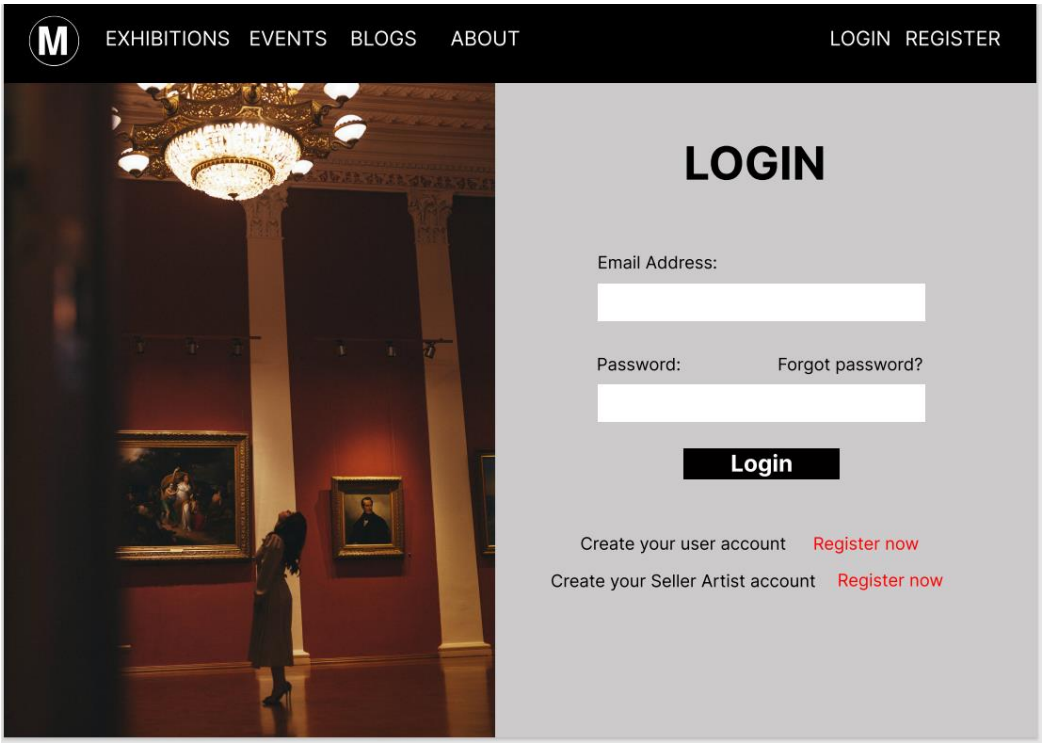
Figure 8: Deployment diagram

4.3 USER INTERFACE DESIGN USING FIGMA

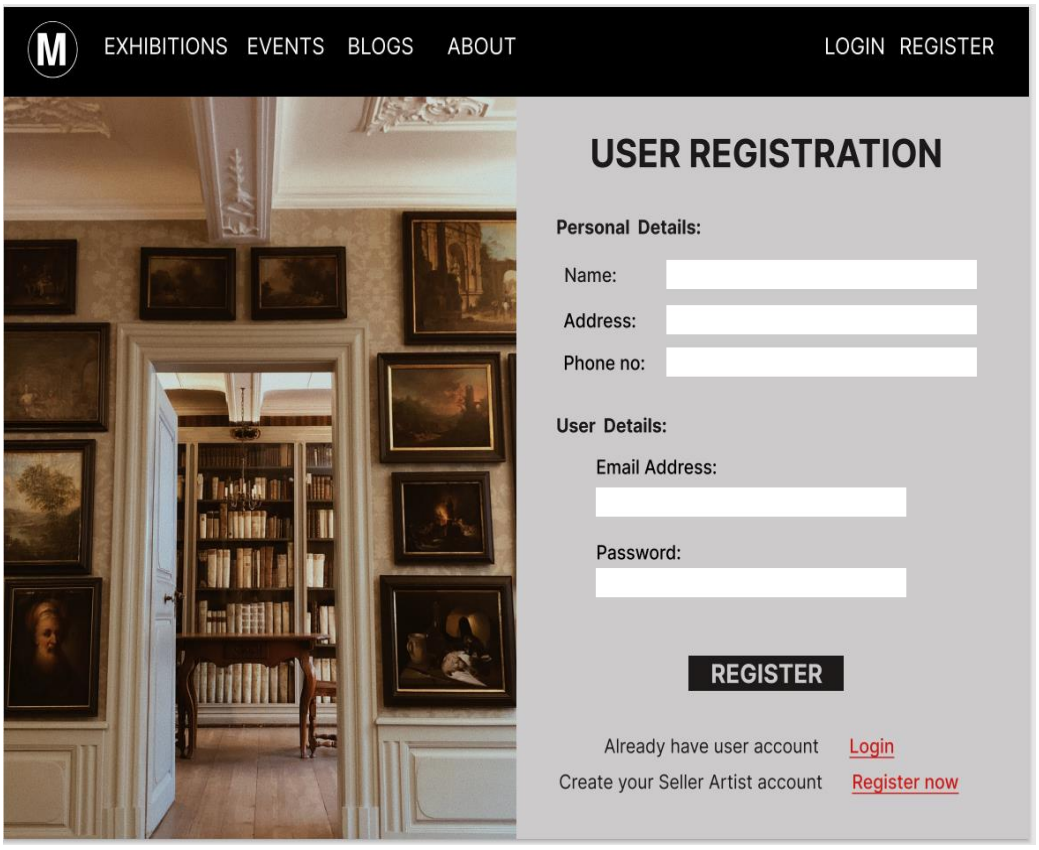
Form Name: Home page



Form Name: Login page



Form Name: Registration page



4.4 DATABASE DESIGN

A database is a collection of data that has been organized to make it simple to manage and update. Information security could be one of the main aims of any database. The database design process is divided into two steps. The goal of the first step is to gather user requirements to create a database that as clearly as possible satisfies user needs. It is known as information-level design and is carried out without the aid of any DBMS. An information-level design is changed to a specific DBMS design that will be used to develop the system in the following stage. The physical-level design phase is where the characteristics of the specific DBMS are considered.

4.4.1 Relational Database Management System (RDBMS)

A Relational Database Management System (RDBMS) is a critical software application for organizing and managing data in a structured manner. It stores data in tables with rows and columns, where each row represents a record, and each column is a specific attribute or field. RDBMS systems like MySQL, Oracle, or Microsoft SQL Server ensure data integrity, enforce relationships between tables, and allow for efficient data retrieval and manipulation through Structured Query Language (SQL). These systems are widely used in various applications, such as e-commerce websites, financial systems, and inventory management, due to their robustness, scalability, and ability to handle complex data structures, making them essential for modern data-driven environments.

A Relational Database Management System (RDBMS) is a cornerstone of modern data management. It structures data into tables, where each row represents a unique entry, and each column corresponds to a specific attribute, ensuring a logical and organized storage system. RDBMS systems employ complex algorithms for data retrieval and manipulation, guaranteeing data consistency and adherence to predefined relationships between tables. Popular RDBMS software, including PostgreSQL, MySQL, and Microsoft SQL Server, provides robust features for transactions, data security, and scalability. These systems are integral to a myriad of applications, from healthcare records and customer databases to inventory control and financial platforms, supporting the efficient storage, retrieval, and analysis of vast datasets, making them essential tools in today's data-driven world.

4.4.2 Normalization

Normalization is a database design technique used in relational database management systems (RDBMS) to eliminate data redundancy and improve data integrity. It involves organizing data in a way that reduces data duplication and ensures that relationships between tables are defined and

maintained.

The primary goals of normalization are:

Minimizing Data Redundancy: By breaking down data into separate tables and ensuring that each piece of data is stored only once, normalization helps reduce the chances of data inconsistencies or errors.Ensuring Data Integrity: Normalization enforces the rules of referential integrity, which means that relationships between tables are well-defined and maintained. This ensures that data remains accurate and consistent.Normalization typically involves dividing a database into multiple related tables and using primary keys and foreign keys to establish relationships between these tables. There are several normal forms, from First Normal Form (1NF) to Fifth Normal Form (5NF), each with specific rules and requirements. The level of normalization achieved depends on the specific needs of the database and the trade-off between data redundancy and query performance.

By applying normalization principles, designers can create efficient and reliable database structures that support data integrity and ease data maintenance and manipulation.

There are several normal forms (NF) that define specific rules and requirements for achieving progressively higher levels of normalization. Here are the most common normal forms, from First Normal Form (1NF) to Fifth Normal Form (5NF):

First Normal Form (1NF):

- Each table has a primary key.
- All columns contain atomic (indivisible) values.
- There are no repeating groups or arrays in columns.

Second Normal Form (2NF):

- The table is in 1NF.
- All non-key attributes are fully functionally dependent on the entire primary key. In other words, all non-key attributes must be dependent on the entire primary key, not just part of it.

Third Normal Form (3NF):

- The table is in 2NF.
- There is no transitive dependency, meaning that non-key attributes are not dependent on other non-key attributes.

Boyce-Codd Normal Form (BCNF):

- A stricter version of 3NF.
- It enforces that every non-trivial functional dependency involves a super key.

Fourth Normal Form (4NF):

- Addresses multi-valued dependencies.

- Eliminates any multi-valued dependencies within the data.

Fifth Normal Form (5NF):

- Also known as Project-Join Normal Form (PJ/NF).
- Handles cases where data can be derived by joining multiple tables.

4.4.3 Sanitization

In Python Django, sanitization refers to the process of cleaning and validating data to ensure that it is safe and free from malicious content before it is used or stored in a database. Sanitization is a crucial security practice to prevent various forms of attacks, such as cross-site scripting (XSS) and SQL injection, which can compromise the security and integrity of a web application.

4.4.4 Indexing

Indexing in Django is a fundamental database optimization technique. It involves creating data structures, or indexes, on specific fields to expedite data retrieval. These indexes act as signposts that enable the database to swiftly locate and fetch relevant data, especially in tables with substantial amounts of information. Django offers automatic index creation for primary keys, unique fields, and foreign keys. Additionally, developers can define custom indexes for fields frequently used in filtering, sorting, or searching. The choice of proper indexing plays a pivotal role in improving query performance, resulting in more responsive and scalable web applications. It's essential to consider application-specific query patterns and create indexes accordingly, as well as to understand the capabilities and limitations of the chosen database backend. Effective indexing is a cornerstone of efficient database operations in Django, contributing to enhanced application performance.

4.5 TABLE DESIGN

1.User table

Primary key: **userid**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	User_id	IntegerField(20)	PK	User type id
2	password	Varchar(30)	Not Null	Password of user
3	username	Charfield(20)	Not Null	Username of user
4	email	EmailField(20)	Unique	Email_id of user
5	User_type	CharField(20)	Not Null	Type of user

2. Artwork table

Primary key: **art_id**

Foreign key: **userid** references table **Tbl_user**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	art_id	IntegerField(20)	PK	Id of art
2	User_id	IntegerField(20)	FK	Id of user
3	Art_name	Varchar(30)	Not Null	Name of art
4	theme	Charfield(20)	Not Null	Theme of the art
5	Art_type	Charfield(20)	Not Null	Type of art
6	description	CharField(20)	Not Null	Description of art
7	price	IntegerField	Not Null	Price of art
8	status	Charfield(20)	Not Null	Status of art
9	Product image	ImageField	Not Null	Image of product

3. Blog table

Primary key: **blog_id**

Foreign key: **userid** references table **Tbl_user**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	blog_id	IntegerField(20)	PK	Id of blog
2	User_id	IntegerField(20)	FK	Id of user
3	title	Charfield(20)	Not Null	title of blog
3	PublishingDate	Datefield(20)	Not Null	Date of publishing
4	description	Charfield(20)	Not Null	Description of blog
8	image	ImageField	Not Null	Image of product

4. Competition table

Primary key: **competition_id**

Foreign key: **userid** references table **Tbl_user**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	competition_id	IntegerField(20)	PK	Id of blog
2	User_id	IntegerField(20)	FK	Id of user
3	competition_name	Charfield(20)	Not Null	Name of competition
4	Date	Datefield(20)	Not Null	Date of competition
5	time	Timefield(20)	Not Null	Time of competition
6	location	Charfield(20)	Not Null	Location of competition
7	image	ImageField	Not Null	Image of competition
8	Am_pm	Charfield(20)	Not Null	Am or pm
9	fees	IntegerField(100)	Not Null	Fees of competition

CHAPTER 5
SYSTEM TESTING

5.1 INTRODUCTION

Software testing is a way to check if the computer program works like it's supposed to. We use testing to make sure the software does what it is supposed to do. Validation means checking or testing things like software to make sure they meet the requirements and standards they are supposed to follow. Software testing is a way to check if a program works well. It goes along with other methods like checking and walking through the program. Validation means making sure that what the user wanted is what they got. There are several rules that can serve as testing objectives.

They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a test works well and follows its goals, it can find mistakes in the software. The test showed that the computer program is working like it's supposed to and is doing well.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

5.2 TEST PLAN

A test plan suggests several required steps that need be taken to complete various testing methodologies. The activity that is to be taken is outlined in the test plan. A computer program, its documentation, and associated data structures are all created by software developers. It is always the responsibility of the software developers to test each of the program's separate components to make sure it fulfills the purpose for which it was intended. To solve the inherent issues with allowing the builder evaluate what they have developed, there is an independent test group (ITG). Testing's precise goals should be laid forth in quantifiable language. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test.

The levels of testing include:

• Unit testing

• Integration Testing

• Data validation Testing & Output Testing

5.2.1 Unit Testing

Unit testing concentrates verification efforts on the software component or module, which is the smallest unit of software design. The component level design description is used as a guide when testing crucial control paths to find faults inside the module's perimeter. The level of test complexity and the untested area determined for unit testing. Unit testing is white-box focused, and numerous components may be tested simultaneously. To guarantee that data enters and exits the software unit under test properly, the modular interface is tested. To make sure that data temporarily stored retains its integrity during each step of an algorithm's execution, the local data structure is inspected. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Before starting any other test, tests of data flow over a module interface are necessary. All other tests are irrelevant if data cannot enter and depart the system properly. An important duty during the unit test is the selective examination of execution pathways. Error circumstances must be foreseen in good design, and error handling paths must be put up to cleanly reroute or halt work when an error does arise. The final step of unit testing is boundary testing. Software frequently fails at its limits.

In the Sell-Soft System, unit testing was carried out by treating each module as a distinct entity and subjecting them to a variety of test inputs. The internal logic of the modules had some issues, which were fixed. Each module is tested and run separately after coding. All unused code was eliminated, and it was confirmed that every module was functional and produced the desired outcome

5.2.2 Integration Testing

Integration testing is a methodical approach for creating the program's structure while also carrying out tests to find interface issues. The goal is to construct a program structure that has been determined by design using unit tested components. The program is tested. Correction is challenging since the size of the overall program makes it challenging to isolate the causes. As soon as these mistakes are fixed, new ones arise, and the process repeats itself in an apparently unending cycle. All the modules were integrated after unit testing was completed in the system to check for an interface inconsistency. A distinctive program structure also developed when discrepancies in program structures.

5.2.3 Validation Testing or System Testing

The testing process comes to an end here. This involved testing the entire system in its entirety, including all forms, code, modules, and class modules. Popular names for this type of testing include system tests and black box testing. The functional requirements of the software are the main

emphasis of the black box testing approach. That example, using Black Box testing, a software engineer can create sets of input conditions that will fully test every program requirement. The following sorts of problems are targeted by black box testing: erroneous or missing functions, interface errors, data structure or external data access errors, performance errors, initialization errors, and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required.

This done with respect to the following points:

• Input Screen Designs,

• Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future.

5.2.5 Automation Testing

Automation Testing is a software testing technique that performs using special automated testing software tools to execute a test case suite. Essentially, it's a test to double-check that the equipment or software does exactly what it was designed to do. It tests for bugs, defects, and any other issues that can arise with product development. Although some types of testing, such as regression or functional testing can be done manually, there are greater benefits of doing it automatically. Automation testing can be run at any time of the day. It uses scripted sequences to examine the software. It then reports on what has been found, and this information can be compared with earlier test runs. Automation developers generally write in the following programming languages: C#, JavaScript, and Ruby.

5.2.6 Selenium Testing

Selenium is an open-source automated testing framework used to verify web applications across different browsers and platforms. Selenium allows for the creation of test scripts in various programming languages such as Java, C#, and Python. Jason Huggins, an engineer at Thought Works, developed Selenium in 2004 while working on a web application that required frequent testing. He created a JavaScript program called "JavaScriptTestRunner" to

automate browser actions and improve testing efficiency. Selenium has since evolved and continues to be developed by a team of contributors. In addition to Selenium, another popular tool used for automated testing is Cucumber. Cucumber is an open-source software testing framework that supports behavior-driven development (BDD). It allows for the creation of executable specifications in a human-readable format called Gherkin. One of the advantages of using Cucumber is its ability to bridge the gap between business stakeholders and technical teams. By using a common language, Cucumber facilitates effective communication and collaboration during the testing process. It promotes a shared understanding of the requirements and helps ensure that the developed software meets the intended business goals. Cucumber can be integrated with Selenium to combine the benefits of both tools. Selenium is used for interacting with web browsers and automating browser actions, while Cucumber provides a structured framework for organizing and executing tests. This combination allows for the creation of end-to-end tests that verify the behavior of web applications across different browsers and platforms, using a business-readable and maintainable format.

Test Case 1

Code

```
from django.test import TestCase
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
class LoginTest(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.base_url = "http://127.0.0.1:8000"  # Replace this with your base URL
        self.driver.maximize_window()
        self.driver.implicitly_wait(10)
    def test_successful_login(self):
        self.driver.get(self.base_url + '/accounts/login/')
        # Check if the login page is accessible by checking the page URL
        self.assertIn("/accounts/login/", self.driver.current_url)
        username = self.driver.find_element(By.NAME, 'username')
        password = self.driver.find_element(By.NAME, 'password')
        username.send_keys("anuanagha")  # Replace with a valid username
        password.send_keys("anuanagha@123")  # Replace with a valid password
```

```
        login_button = self.driver.find_element(By.XPATH, "//button[text()='Login']")
        login_button.click()
        # Assuming a successful login redirects to the home page
        # Check for the home page URL, handling the case where "http://127.0.0.1:8000/" might not
be present
        self.assertIn("http://127.0.0.1:8000/", self.driver.current_url)
    def tearDown(self):
        self.driver.quit()
if __name__ == "__main__":
    unittest.main()
```

Screenshot

Test Report

Test Case 1	
Project Name: Modern Art gallery	
Login Test Case	
Test Case ID: Test_1	**Test Designed By:** Anagha P
Test Priority(Low/Medium/High): High	**Test Designed Date:** 3-12-2023
Module Name: Login Screenshot	**Test Executed By :** Dr. Paulin Paul
Test Title : User login	**Test Execution Date:** 2-12-2023
Description: Verify login with valid username and password	
Pre-Condition : User has valid username and password	

Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)
1	Navigation to Login Page		Login Page should be displayed	Login Page be displayed	Pass
2	Provide Valid username	username: anuanagha	User should be able to login	User is logged in and navigated to corresponding Home Page	Pass
3	Provide Valid Password	Password: anuanagha@123			
4	Click on Login button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Test Case 2:

Code

```
import time
from django.test import TestCase
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
class AddCompetitionTest(TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.base_url = "http://127.0.0.1:8000"
        self.driver.maximize_window()
        self.driver.implicitly_wait(10)
    def test_add_competition_workflow(self):
        # Login
        self.login()
        # Navigate to artistprofile.html
        self.driver.get(self.base_url + '/artistprofile.html')
        # Wait for the "Add Competition" button to be present and clickable
        add_competition_button = WebDriverWait(self.driver, 10).until(
```

```
        EC.element_to_be_clickable((By.XPATH, "//a[contains(text(), 'Add Competition')]"))
    )
    # Scroll to the element and click it using JavaScript
    self.scroll_to_element(add_competition_button)
    add_competition_button.click()
    # Verify if redirected to http://127.0.0.1:8000/addcompetition.html
    self.assertIn("http://127.0.0.1:8000/addcompetition.html", self.driver.current_url)
    # Add a competition
    self.add_competition()
    # Wait for the redirect to competition.html
    try:
        WebDriverWait(self.driver, 10).until(
            EC.url_to_be(self.base_url + "/competitions/")
        )
    except Exception as e:
        print("Error waiting for URL:", e)
    # Verify if redirected to http://127.0.0.1:8000/competition.html after adding the competition
    self.assertIn(self.base_url + "/competitions/", self.driver.current_url)
  def login(self):
    self.driver.get(self.base_url + '/accounts/login/')
    username = self.driver.find_element(By.NAME, 'username')
    password = self.driver.find_element(By.NAME, 'password')
    username.send_keys("anuanagha")  # Replace with a valid username
    password.send_keys("anuanagha@123")  # Replace with a valid password
    login_button = self.driver.find_element(By.XPATH, "//button[text()='Login']")
    login_button.click()
    # Assuming a successful login redirects to the home page
    # Check for the home page URL, handling the case where "http://127.0.0.1:8000/" might not be present
    self.assertIn(self.base_url + "/", self.driver.current_url)
  def add_competition(self):
```

```
        competition_name_input = self.driver.find_element(By.NAME,
'competition_name')
        competition_name_input.send_keys("Sample Competition")
        description_input = self.driver.find_element(By.NAME, 'description')
        description_input.send_keys("A wonderful competition about art.")
        date_input = self.driver.find_element(By.NAME, 'date')
        date_input.send_keys("07-12-2023") # Use the correct format, e.g., "YYYY-MM-
DD"
        time_input = self.driver.find_element(By.NAME, 'time')
        time_input.send_keys("14:30")  # Use the correct format, e.g., "HH:MM"
        # Find the 'ampm' select element
        am_pm_select = self.driver.find_element(By.ID, 'ampm')
        # Select the option by value ("AM" or "PM")
        am_pm_select.send_keys("AM")  # or "PM", depending on your test case
        location_input = self.driver.find_element(By.NAME, 'location')
        location_input.send_keys("Art Gallery")
        fee_input = self.driver.find_element(By.NAME, 'fee')
        fee_input.send_keys("500")  # Replace with the desired fee
        # Upload a sample image (replace the file path with the path to your image)
        image_input = self.driver.find_element(By.NAME, 'productImage')
image_input.send_keys(r"D:\djangoproject\art_p\media\competition_images\76s.jpg")
        # Wait for the overlay to disappear
        self.wait_for_overlay_to_disappear()
        # Find the form element and submit it
        form = image_input.find_element(By.XPATH, "./ancestor::form")
        form.submit()
    def wait_for_overlay_to_disappear(self):
        try:
            WebDriverWait(self.driver, 10).until(
                EC.invisibility_of_element_located((By.ID, "loading-spinner"))
            )
        except Exception as e:
            print("Overlay did not disappear:", e)
```

```
    def scroll_to_element(self, element):
        self.driver.execute_script("arguments[0].scrollIntoView();", element)
    def tearDown(self):
        self.driver.quit()
if __name__ == "__main__":
    unittest.main()
```

Screenshot

Test report

Test Case 2

Project Name: Modern Art Gallery					
Add Competition Test Case					
Test Case ID: Test_2			**Test Designed By:** Anagha P		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 3-12-2023		
Module Name: Competition			**Test Executed By :** Dr. Paulin Paul		
Test Title : Adding Competition			**Test Execution Date:** 3-12-2023		
Description: Adding Competition					
Pre-Condition : User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigation to Login Page		Login Page should be displayed	Login Page be displayed	Pass
2	Provide valid username	Username : anuanagha	User should be able to	User is logged in and navigated to	Pass

3	Provide Valid Password	Password: anuanagha@123	login	corresponding Home Page	
4	Click on Login button				
5	Click on Add competition button	Let it snow		Competition form will displayed	Pass
6	Click on Add competition	Let it snow		Competition will display	Pass
Post-Condition: Competition is added successfully.					

Test Case 3:

Code

```
from django.test import TestCase
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
class AddProductTest(TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.base_url = "http://127.0.0.1:8000"  # Replace this with your base URL
        self.driver.maximize_window()
        self.driver.implicitly_wait(10)
    def test_add_product_workflow(self):
        # Login
        self.login()
        # Navigate to artistprofile.html
        self.driver.get(self.base_url + '/artistprofile.html')
        # Click the "Add Product" button
        add_product_button = self.driver.find_element(By.XPATH, "//a[contains(text(),
'Add Product')]")
        self.scroll_to_element(add_product_button)
```

```
        add_product_button.click()
        # Verify if redirected to http://127.0.0.1:8000/addproduct.html
        self.assertIn("http://127.0.0.1:8000/addproduct.html", self.driver.current_url)
        # Add a product
        self.add_product()
        # Verify if still on http://127.0.0.1:8000/addproduct.html after adding the product
        self.assertIn("http://127.0.0.1:8000/addproduct.html", self.driver.current_url)
    def login(self):
        # Your login code here
        self.driver.get(self.base_url + '/accounts/login/')
        username = self.driver.find_element(By.NAME, 'username')
        password = self.driver.find_element(By.NAME, 'password')
        username.send_keys("anuanagha")  # Replace with a valid username
        password.send_keys("anuanagha@123")  # Replace with a valid password
        login_button = self.driver.find_element(By.XPATH, "//button[text()='Login']")
        login_button.click()
        # Assuming a successful login redirects to the home page
        # Check for the home page URL, handling the case where "http://127.0.0.1:8000/"
might not be present
        self.assertIn("http://127.0.0.1:8000/", self.driver.current_url)
    def add_product(self):
        # Fill in product details
        product_name_input = self.driver.find_element(By.NAME, 'productName')
        product_name_input.send_keys("Sample Product")
        theme_select = self.driver.find_element(By.NAME, 'theme')
        theme_select.send_keys("History")
        art_type_select = self.driver.find_element(By.NAME, 'artType')
        art_type_select.send_keys("Pencil Drawing")
        description_input = self.driver.find_element(By.NAME, 'description')
        description_input.send_keys("A beautiful pencil drawing depicting historical
events.")
        price_input = self.driver.find_element(By.NAME, 'price')
        price_input.send_keys("100")
```

```
        status_select = self.driver.find_element(By.NAME, 'status')
        status_select.send_keys("available")
        # Upload a sample image (replace the file path with the path to your image)
        image_input = self.driver.find_element(By.NAME, 'productImage')
    image_input.send_keys(r"D:\djangoproject\art_p\media\product_images\37s_1.jpg")
        # Wait for the overlay to disappear
        WebDriverWait(self.driver, 10).until(EC.invisibility_of_element_located((By.ID,
"loading-spinner")))
        # Wait for the submit button to be clickable
        submit_button = WebDriverWait(self.driver, 10).until(
            EC.element_to_be_clickable((By.XPATH, "//button[@type='submit']"))
        )
        self.scroll_to_element(submit_button)
        # Submit the form
        submit_button.click()
    def scroll_to_element(self, element):
        self.driver.execute_script("arguments[0].scrollIntoView();", element)
    def tearDown(self):
        self.driver.quit()
if __name__ == "__main__":
    unittest.main()
```

Screenshot

```
(env2) D:\djangoproject\art_p>py manage.py test store
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
There was an error managing chrome (error sending request for url (https://googlechromelabs.github.io/chrome-for-testing/known-good-versions-with-downloads.json
): error trying to connect: dns error: No such host is known. (os error 11001)); using driver found in the cache

DevTools listening on ws://127.0.0.1:61161/devtools/browser/439eb727-818c-4fc8-ab81-519584d0d9a3
.
----------------------------------------------------------------------
Ran 1 test in 22.198s

OK
Destroying test database for alias 'default'...

(env2) D:\djangoproject\art_p>
```

Test report

Test Case 3

Project Name: Modern Art Gallery					
Add Product Test Case					
Test Case ID: Test_3			**Test Designed By:** Anagha P		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 3-12-2023		
Module Name: Add Product			**Test Executed By :** Dr. Paulin Paul		
Test Title : Adding Products			**Test Execution Date:** 3-12-2023		
Description: Adding Products					
Pre-Condition : User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigation to Login Page		Login Page should be displayed	Login Page be displayed	Pass
2	Provide valid username	Username : anuanagha	User should be able to login	User is logged in and navigated to corresponding Home Page	Pass
3	Provide Valid Password	Password: anuanagha@123			
4	Click on Login button				
5	Click on Add product button	Let it snow		Product form will displayed	Pass
6	Click on Add Product	Let it snow		Product will display	Pass
Post-Condition: Product is added successfully.					

Test Case 4:

Code

```
import time
from django.test import TestCase
from selenium import webdriver
```

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
class AddBlogTest(TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.base_url = "http://127.0.0.1:8000"  # Replace this with your base URL
        self.driver.maximize_window()
        self.driver.implicitly_wait(10)
    def test_add_blog_workflow(self):
        # Login
        self.login()
        # Navigate to artistprofile.html
        self.driver.get(self.base_url + '/artistprofile.html')
        # Wait for the "Add Blog" button to be present and clickable
        add_blog_button = WebDriverWait(self.driver, 10).until(
            EC.element_to_be_clickable((By.XPATH, "//a[contains(text(), 'Add Blog')]"))
        )
        # Scroll to the element and click it using JavaScript
        self.scroll_to_element(add_blog_button)
        self.driver.execute_script("arguments[0].click();", add_blog_button)
        # Verify if redirected to http://127.0.0.1:8000/blogupload.html
        self.assertIn("http://127.0.0.1:8000/blogupload.html", self.driver.current_url)
        # Add a blog
        self.add_blog()
        # Verify if redirected to http://127.0.0.1:8000/blog.html after adding the blog
        self.assertIn("http://127.0.0.1:8000/blog.html", self.driver.current_url)
    def login(self):
        self.driver.get(self.base_url + '/accounts/login/')
        username = self.driver.find_element(By.NAME, 'username')
        password = self.driver.find_element(By.NAME, 'password')
        username.send_keys("anuanagha")  # Replace with a valid username
        password.send_keys("anuanagha@123")  # Replace with a valid password
        login_button = self.driver.find_element(By.XPATH, "//button[text()='Login']")
```

```
        login_button.click()
        # Assuming a successful login redirects to the home page
        # Check for the home page URL, handling the case where "http://127.0.0.1:8000/"
might not be present
        self.assertIn("http://127.0.0.1:8000/", self.driver.current_url)
    def add_blog(self):
        # Fill in blog details
        title_input = self.driver.find_element(By.NAME, 'title')
        title_input.send_keys("Sample Blog Title")
        publishing_date_input = self.driver.find_element(By.NAME, 'publishingDate')
        publishing_date_input.send_keys("07-12-2023")  # Use the correct format, e.g.,
"YYYY-MM-DD"
        description_input = self.driver.find_element(By.NAME, 'description')
        description_input.send_keys("A wonderful blog about art.")
        # Upload a sample image (replace the file path with the path to your image)
        image_input = self.driver.find_element(By.NAME, 'image')
        image_input.send_keys(r"D:\djangoproject\art_p\media\blog_images\80.jpg")
        # Wait for the overlay to disappear
        WebDriverWait(self.driver, 10).until(EC.invisibility_of_element_located((By.ID,
"loading-spinner")))
        # Wait for the submit button to be clickable
        submit_button = WebDriverWait(self.driver, 10).until(
            EC.element_to_be_clickable((By.XPATH, "//button[@type='submit']"))
        )
        # Scroll to the element and click it using JavaScript
        self.scroll_to_element(submit_button)
        self.driver.execute_script("arguments[0].click();", submit_button)
        # Wait for the redirect to blog.html
        WebDriverWait(self.driver, 10).until(
            EC.url_to_be("http://127.0.0.1:8000/blog.html")
        )
    def scroll_to_element(self, element):
        self.driver.execute_script("arguments[0].scrollIntoView();", element)
    def tearDown(self):
```

```
        self.driver.quit()
if __name__ == "__main__":
    unittest.main()
```

Screenshot

```
(env2) D:\djangoproject\art_p>py manage.py test store
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
There was an error managing chrome (error sending request for url (https://googlechromelabs.github.io/chrome-for-testing/known-good-versions-with-downloads.json
): error trying to connect: dns error: No such host is known. (os error 11001)); using driver found in the cache

DevTools listening on ws://127.0.0.1:61203/devtools/browser/ee501267-245d-4169-89a0-0d89f1ef5908
.
----------------------------------------------------------------------
Ran 1 test in 20.556s

OK
Destroying test database for alias 'default'...

(env2) D:\djangoproject\art_p>
```

Test report

Test Case 4

Project Name: Modern Art Gallery					
Add Blog Test Case					
Test Case ID: Test_4			**Test Designed By:** Anagha P		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 3-12-2023		
Module Name: Blog			**Test Executed By :** Dr. Paulin Paul		
Test Title : Adding Blog			**Test Execution Date:** 3-12-2023		
Description: Adding Blog					
Pre-Condition : User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigation to Login Page		Login Page should be displayed	Login Page be displayed	Pass
2	Provide valid username	Username : anuanagha	User should be able to login	User is logged in and navigated to corresponding Home Page	Pass
3	Provide Valid Password	Password: anuanagha@123			
4	Click on Login button				

5	Click on Add Blog button	Let it snow		Blog form will displayed	Pass
6	Click on Add Blog	Let it snow		Blog will display	Pass
Post-Condition: Blog is added successfully.					

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion. Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover. The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

The implementation procedures for this project involve a systematic and phased approach to bring the online crime reporting portal to fruition. This complex system, designed to cater to the needs of various stakeholders, requires careful planning and execution.

To commence the implementation, the project team will initiate a detailed requirements analysis phase. This stage involves comprehensive discussions with law enforcement officials, control room staff, prison wardens, and potential end-users to determine their specific needs and expectations. The results of this analysis will inform the development of a detailed system specification and

design.

The development phase will follow, involving the creation of the portal's architecture, databases, and user interfaces. It's during this stage that the functionalities outlined in the project's scope, including user management, crime reporting, and communication features, will be integrated into the system. The portal will be designed with a user-friendly interface to ensure ease of use for all categories of users.

Simultaneously, a dedicated team will work on the incorporation of advanced technologies, such as machine learning capabilities for crime trend analysis and predictive features. These technologies will be implemented carefully to ensure the portal's robustness and security.

Once the development phase is complete, rigorous testing will be conducted to identify and rectify any bugs or issues. The portal will undergo extensive quality assurance and user acceptance testing to ensure that it meets all the necessary requirements and is free of vulnerabilities.

Deployment of the system will be carried out in a controlled manner, ensuring minimal disruption to the existing processes, and allowing for gradual adaptation by the involved stakeholders. Comprehensive training sessions will be conducted to familiarize law enforcement personnel, control room staff, prison wardens, and registered users with the portal's functionality and features. Post-deployment, the project team will continue to provide support and maintenance, addressing any issues that may arise and making necessary improvements as the system matures. Regular updates and security measures will be implemented to safeguard user data and maintain the portal's efficiency.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer-based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database, and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or

part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than “Finding Mistakes”.

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In conclusion, the proposed Modern art gallery project presents a well-structured and versatile platform that caters to the dynamic needs of artists, customers, and administrators. With distinct modules for user management, artwork management, event management, and artist management, the system fosters a collaborative space for the global art community. The envisioned platform encourages artists to showcase their work securely, engage with a broader audience, and participate in art-related events. By emphasizing user-friendly interfaces and streamlined functionalities, the project envisions a vibrant and interactive online ecosystem. While recognizing the limitations of traditional art exhibition systems, the designed platform leverages the advantages of the digital realm to transcend geographical boundaries and foster direct interactions between artists and art enthusiasts. Overall, the project strives to contribute to the promotion of cross-cultural art, providing a valuable space for the exchange of ideas, creative expressions, and artistic endeavors.

7.2 FUTURE SCOPE

The future scope of the Modern art gallery project holds significant potential for growth and improvement. One key area for development is the introduction of online auctions, providing artists a platform to auction their artwork and offering art enthusiasts an engaging way to acquire unique pieces. Furthermore, incorporating robust copyright mechanisms will ensure the protection of artists' intellectual property rights, fostering trust and accountability within the community. The addition of online payment and purchase functionalities is another promising avenue, streamlining transactions and providing customers with a secure and convenient method to support artists. As technology continues to advance, exploring new features and trends in the digital art space will be crucial for staying at the forefront of the industry and providing an enriched experience for users.

.

CHAPTER 8
BIBLIOGRAPHY

REFERENCES:

- Google Support - How to search on Google: Tips on effective Google searches, including finding quick answers
- MyGreatLearning - 180+ SQL Interview Questions and Answers in 2023: A compilation of SQL interview questions for job seekers
- Andrew Hunt, David Thomas, The Pragmatic Programmer: From Journeyman to Master, Pearson India, 1st Edition (2008).
- Qualtrics - Multiple Choice Question: Information about the multiple-choice question type for surveys.

WEBSITES:

- https://www.mojarto.com/
- https://www.jacksonsart.com/blog/
- www.geeksforgeeks.org
- https://www.w3schools.com/django/

CHAPTER 9

APPENDIX

9.1 Sample Code

Index.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<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>Art Web</title>
        <link href="{% static '/css/bootstrap.min.css' %}" rel="stylesheet" >
        <link href="{% static '/css/font-awesome.min.css' %}" rel="stylesheet" >
        <link href="{% static '/css/global.css' %}" rel="stylesheet">
        <link href="{% static '/css/index.css' %}" rel="stylesheet">
        <link href="{% static
'/https://fonts.googleapis.com/css2?family=Fraunces:opsz@9..144&display=swap' %}"
rel="stylesheet">
        <script src="{% static '/js/bootstrap.bundle.min.js' %}"></script>
</head>
<body>
<div class="main clearfix position-relative">
 <div class="main_1 clearfix position-absolute top-0 w-100">
  <section id="header" style="background-color: rgb(0, 0, 0);">
<nav class="navbar navbar-expand-md navbar-light" id="navbar_sticky">
  <div class="container-xl">
   <a class="navbar-brand fs-2 p-0 fw-bold text-white" href="{% static '/index.html' %}"><a
href="#"><img src="{% static '/img/logo.png' %}" width="50" height="50" alt="abc"></a><!--<i
class="fa fa-pencil col_pink me-1 align-middle"></i> art --><span class="col_pink span_1"
>MODERN</span><br><span class="font_12 span_2">ART GALLERY</span></a>
   <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
     <span class="navbar-toggler-icon"></span>
   </button>
```

```
   <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mb-0 ms-auto">
     <li class="nav-item">
      <a class="nav-link active" aria-current="page" href="index.html">Home</a>
     </li>
                <li class="nav-item">
      <a class="nav-link" href="about.html">About </a>
     </li>
                <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="{% static '/#' %}" id="navbarDropdown"
role="button" data-bs-toggle="dropdown" aria-expanded="false">
        Events
      </a>
      <ul class="dropdown-menu drop_1" aria-labelledby="navbarDropdown">
       <li><a class="dropdown-item" href="exhibition.html">Exhibitions</a></li>
       <li><a class="dropdown-item border-0" href="competition.html">competitions</a></li>
      </ul>
     </li>
                <li class="nav-item">
                    <a class="nav-link" href="blog.html">Blog </a>
                 </li>
                 <li class="nav-item">
                    <a class="nav-link" href="contact.html">Contact </a>
                 </li>
                <li class="nav-item dropdown">
   {% if user.is_authenticated %}
      <a class="nav-link dropdown-toggle btn btn-primary d-flex align-items-center justify-
content-center" href="#" id="userDropdown" role="button" data-bs-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
        <i class="fa fa-user"></i>
        <p class="m-0">{{ user.username }}</p>
      </a>
      <div class="dropdown-menu" aria-labelledby="userDropdown">
        {% if user.user_type == 'user' %}
```

```
            <a class="dropdown-item" href="{% url 'userprofile' %}">View Profile</a>
          {% elif user.user_type == 'artist' %}
            <a class="dropdown-item" href="{% url 'artistprofile' %}">View Profile</a>
          {% endif %}
          <a class="dropdown-item" href="{% url 'logout' %}">Logout</a>
        </div>
    {% else %}
      <a href="{% url 'login' %}" class="btn btn-primary ml-lg-3">Login/Register</a>
    {% endif %}
</li>
                    <!--<li class="nav-item">
                            <a class="nav-link" href="register.html">Register</a>
                    </li>-->
      </ul>
    </div>
  </div>
</nav>
</section>
</div>
 </div>
 <div class="main_2 clearfix">
   <section id="center" class="center_home">
 <div id="carouselExampleCaptions" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-indicators">
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="0"
class="active" aria-label="Slide 1"></button>
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="1" aria-
label="Slide 2" class="" aria-current="true"></button>
    <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="2" aria-
label="Slide 3"></button>
  </div>
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="{% static '/img/7s.jpg' %}" class="d-block w-100" alt="...">
```

```
    <div class="carousel-caption d-md-block">
      <br><center><h1 class="text-white font_60" style="font-size:
150px;">MODERN</h1></center>
                <center><h2 class="text-white mt-3" style="font-size: 55px;">ART
GALLERY</h2></center>
                <center><h4 class="text-white mt-3">The sky is the ultimate art gallery just above
us.</h4></center>


    </div>
  </div>
  <div class="carousel-item">
    <img src="{% static '/img/2s.jpg' %}" class="d-block w-100" alt="...">
    <div class="carousel-caption d-md-block">
      <h1 class="text-white font_60">Other Type Painting</h1>
                <h4 class="text-white mt-3">Georgia O'Keeffe</h4>
                <p class="text-white mt-4">I have been absolutely terrified every moment of my
life—and I have never let it keep me from doing a single thing I wanted to do.</p>
                <!--<h6 class="mt-4 mb-0"><a class="button" href="{% static '/#' %}"><i
class="fa fa-bullhorn bg-white col_pink p-3"> </i> <span class="ps-3 pe-3">Back to
overview</span></a></h6>-->
    </div>
  </div>
  <div class="carousel-item">
    <img src="{% static '/img/3s.jpg' %}" class="d-block w-100" alt="...">
    <div class="carousel-caption d-md-block">
      <h1 class="text-white font_60">Trending Art Picture</h1>
                <h4 class="text-white mt-3">Roger Scruton</h4>
                <p class="text-white mt-4">The great artists of the past were aware that human
life is full of chaos and suffering. But they had a remedy for this. And the name of that remedy
was “beauty.” The beautiful work of art brings consolation in sorrow and affirmation in joy. It
shows human life to be worthwhile.</p>
                <!--<h6 class="mt-4 mb-0"><a class="button" href="{% static '/#' %}"><i
class="fa fa-bullhorn bg-white col_pink p-3"> </i> <span class="ps-3 pe-3">Back to
overview</span></a></h6>-->
```

```
    </div>
   </div>
  </div>
  <button class="carousel-control-prev" type="button" data-bs-
target="#carouselExampleCaptions" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-
target="#carouselExampleCaptions" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </button>
</div>
</section>
 </div>
</div>
<section id="about_h" class="p_4 pt-0">
 <div class="container-xl">
  <div class="row port_1 text-center mb-4">
    <div class="col-md-12">
          <h1 class="font_60">ABOUT US</h1>
          <p>The sky is the ultimate art gallery just above us.</p>
          <span class="icon_line col_pink"><i class="fa fa-square-o"></i></span>
         </div>
  </div>
  <div class="about_h1 row">
   <div class="col-md-6">
    <div class="about_h1l">
          <div class="grid clearfix">
                              <figure class="effect-jazz mb-0">
                                    <a href="#"><img src="{% static '/img/6s.jpg' %}"
class="w-100" alt="abc"></a>
                              </figure>
```

```
                    </div>
          </div>
  </div>
  <div class="col-md-6">
    <div class="about_h1r">
          <br><br><h1>ART GALLERY</h1>
          <p>An art gallery is an institution characterized by the sale of artwork in a formal environment organized by a group of people who run the gallery. An art gallery is a place where artwork is also exhibited and open to the public for viewing. </p>
          <br><br><h1 class="mt-4">EXHIBITIONS</h1>
          <p>An art exhibition is any space where art is displayed to be viewed by an audience. Like most things in the world of art, this term can be tough to pin down because it means different things to different folks. </p>
          </div>
  </div>
 </div>
</div>
</section>
<section id="footer" class="pt-3 pb-3 ">
 <div class="container-fluid ">
  <div class="row footer_1">
   <div class="col-md-3">
          <div class="footer_1i">
           <hr class="line_1">
           <h5 class="mb-3">MODERN ART GALLERY</h5>
           <p>2nd Floor, Phoenix,<BR>
                Near Vijay Cross Roads, Navrangpura,<BR>
                Ahmedabad - 380 009. (INDIA)<BR>
                Tel: +91-9712979034</p>
          </div>
         </div>
             <div class="col-md-3">
          <div class="footer_1i">
           <hr class="line_1">
```

```
          <h5 class="mb-3">MORE</h5>
           <ul class="mb-0">
            <li class="d-inline-block"><a class="d-block" href="#">About us</a></li>
                  <li class="d-inline-block"><a class="d-block" href="#">Blog</a></li>
                  <li class="d-inline-block"><a class="d-block" href="#">Events</a></li>
                  <li class="d-inline-block"><a class="d-block" href="#">Contact Us</a></li>
           </ul>
          </div>
         </div>
  </div>
  <div class="row footer_2 mt-4 text-center">
   <div class="col-md-12">
         <ul>
          <li class="d-inline-block me-3 font_14"><a href="#">CONTACT</a></li>
          <li class="d-inline-block me-3 font_14"><a href="#">PRIVACY POLICY</a></li>
          <li class="d-inline-block me-3 font_14"><a href="#">TERMS OF USE</a></li>
          <li class="d-inline-block font_14"><a href="#">FAQ</a></li>
         </ul>
         <p class="mb-0">© 2013 Modern Art Gallery. All Rights Reserved | Design by <a
class="col_pink" href="#">Art gallery</a></p>
         </div>
  </div>
 </div>
</section>
<script src="{% static 'js/jquery-3.6.0.min.js' %}"></script>
<script src="{% static 'js/bootstrap.min.js' %}"></script>
<script>
         window.onscroll = function() {myFunction()};
         var navbar_sticky = document.getElementById("navbar_sticky");
         var sticky = navbar_sticky.offsetTop;
         var navbar_height = document.querySelector('.navbar').offsetHeight;
         function myFunction() {
          if (window.pageYOffset >= sticky + navbar_height) {
                navbar_sticky.classList.add("sticky")
```

```
            document.body.style.paddingTop = navbar_height + 'px';
        } else {
            navbar_sticky.classList.remove("sticky");
            document.body.style.paddingTop = '0'
        }
    }
    </script>
</body>
</html>
```

login.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<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>Art Web</title>
    <link href="{% static '/css/bootstrap.min.css' %}" rel="stylesheet" >
    <link href="{% static '/css/font-awesome.min.css' %}" rel="stylesheet" >
    <link href="{% static '/css/global.css' %}" rel="stylesheet">
    <link href="{% static '/css/index.css' %}" rel="stylesheet">
    <link href="{% static
'/https://fonts.googleapis.com/css2?family=Fraunces:opsz@9..144&display=swap' %}"
rel="stylesheet">
    <script src="{% static '/js/bootstrap.bundle.min.js' %}"></script>

  <style>
.login-container
 {
   max-width: 400px; /* Adjust the maximum width as needed */
   margin: 0 auto; /* Center the container horizontally */
   padding: 20px; /* Add padding to the container for spacing */
   background-color: #000000cf; /* Optional background color for the container */
   border: 1px solid #ccc; /* Optional border for the container */
   border-radius: 5px; /* Optional border-radius for rounded corners */
   box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1); /* Optional box shadow */
}
.error-box
{
   background-color: #f8d7da;
   border: 1px solid #f5c6cb;
   color: #721c24;
   padding: 10px;
```

```
   margin-bottom: 10px;
}
body {
        background-image: url("{% static '/img/22s.jpg' %}");
        background-size: cover; /* Adjust this to control the image size */
        /* Additional background properties can be added here */
      }

 </style>

</head>
<body>
{% include "messages.html" %}
<div class="main clearfix position-relative">
 <div class="main_1 clearfix position-absolute top-0 w-100">
  <section id="header" style="background-color: rgba(0, 0, 0, 0.496);">
<nav class="navbar navbar-expand-md navbar-light" id="navbar_sticky">
 <div class="container-xl">
  <a class="navbar-brand fs-2 p-0 fw-bold text-white" href="{% static '/index.html' %}"><a
href="#"><img src="{% static '/img/logo.png' %}" width="50" height="50" alt="abc"></a><!--<i
class="fa fa-pencil col_pink me-1 align-middle"></i> art --><span class="col_pink span_1"
>MODERN</span><br><span class="font_12 span_2">ART GALLERY</span></a>
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
   <ul class="navbar-nav mb-0 ms-auto">

     <li class="nav-item">
      <a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>
     </li>
                <li class="nav-item">
      <a class="nav-link" href="{% url 'about' %}">About </a>
     </li>
                <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="{% static '/#' %}" id="navbarDropdown"
role="button" data-bs-toggle="dropdown" aria-expanded="false">
        Events
      </a>
      <ul class="dropdown-menu drop_1" aria-labelledby="navbarDropdown">
       <li><a class="dropdown-item" href="{% url 'exhibition' %}">Exhibitions</a></li>
       <li><a class="dropdown-item border-0" href="{% url 'competition'
%}">competitions</a></li>
      </ul>
     </li>
                <li class="nav-item">
                        <a class="nav-link" href="{% url 'blog' %}">Blog </a>
                 </li>
```

```
                <li class="nav-item">
        <a class="nav-link" href="{% url 'contact' %}">Contact</a>
       </li>
                <li class="nav-item">
    {% if user.is_authenticated %}
    <a href="{% url 'logout' %}" class="btn btn-primary ml-lg-3">Logout</a>
    {% else %}
    <a href="{% url 'login' %}" class="btn btn-primary ml-lg-3">Login/Register</a>
    {% endif %}
                        <!--<a class="nav-link btn btn-info" style="font-weight: bold;"
href="login.html">Login/Register</a>-->
                </li>
                <!--<li class="nav-item">
                        <a class="nav-link" href="register.html">Register</a>
                </li>-->
     </ul>
    </div>
  </div>
</nav>
</section>
</div>
 </div>
 <br><br><br><br><br><br><br><br><br><!--<center><h2 style="font-weight:
bold">LOGIN</h2></center>-->
 <div id="home" class="banner-container no-padding">
    <div class="container">
          <div class="col-md-6 mx-auto">
             <div class="row mx-auto"></div>
                <div class="slider-det">
                   <div id="carouselExampleControls" class="carousel slide" data-
ride="carousel">
                     <div class="login-container">
               <form action="" method="POST" id="login-form">
                  {% csrf_token %}
                  {% if messages %}
                   <div class="error-box">
                   {% for message in messages %}
                     {{ message }}
                   {% endfor %}
                   </div>
                  {% endif %}
                  <div>
                   <center><h2 style="font-weight: bold">LOGIN</h2></center>
                   <div class="form-group">
                     <label for="username">Username:</label>
                     <input type="text" id="username" name="username" class="form-control"
required>
                     <span id="username-error" style="color:red;" class="error"></span>
                  </div>
                  <div class="form-group">
```

```
                    <label for="password">Password:</label>
                    <input type="password" id="password" name="password" class="form-control"
required>
                    <span id="password-error" style="color:red;" class="error"></span>
                  </div><br>
                  <center><button type="submit" class="btn btn-info">Login</button></center>
                </form>
                <center>

                  <p>Don't have an account?<a href="{% url 'register' %}" style="color:
#a81c51;">  Create One!</a> 
                    <p><a href="{% url 'password_reset' %}" class="link" style="color: rgb(61, 88,
147);">Forgot Password</a></p>
                </center>
              </div>
            </div>
          </div>
      </div>
 </div>
</div>
</div>
<div class="w-10 col-12 d-flex justify-content-center">
<form method="POST" autocomplete="off">
<div class="form-group" >

</div>
</form>
</div><br><br><br><br><br><br><br><br><br><br>
 <section id="footer" class="pt-3 pb-3 " style="background-color:black;">
  <div class="container-fluid ">
   <div class="row footer_1">
    <div class="col-md-3">
   <div class="footer_1i">
    <hr class="line_1">
    <h5 class="mb-3">MODERN ART GALLERY</h5>
    <p>2nd Floor, Phoenix,<BR>
    Near Vijay Cross Roads, Navrangpura,<BR>
    Ahmedabad - 380 009. (INDIA)<BR>
    Tel: +91-9712979034</p>
   </div>
  </div>
      <div class="col-md-3">
   <div class="footer_1i">
    <hr class="line_1">
    <h5 class="mb-3">MORE</h5>
     <ul class="mb-0">
      <li class="d-inline-block"><a class="d-block" href="#">About us</a></li>
    <li class="d-inline-block"><a class="d-block" href="#">Blog</a></li>
    <li class="d-inline-block"><a class="d-block" href="#">Events</a></li>
    <li class="d-inline-block"><a class="d-block" href="#">Contact Us</a></li>
```

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

    </div>
    <div class="row footer_2 mt-4 text-center">
     <div class="col-md-12">
    <ul>
     <li class="d-inline-block me-3 font_14"><a href="#">CONTACT</a></li>
     <li class="d-inline-block me-3 font_14"><a href="#">PRIVACY POLICY</a></li>
     <li class="d-inline-block me-3 font_14"><a href="#">TERMS OF USE</a></li>
     <li class="d-inline-block font_14"><a href="#">FAQ</a></li>
    </ul>
    <p class="mb-0">© 2013 Modern Art Gallery. All Rights Reserved | Design by <a
class="col_pink" href="#">Art gallery</a></p>
   </div>
    </div>
  </div>
 </section>

 <script>
   window.onscroll = function() {myFunction()};

   var navbar_sticky = document.getElementById("navbar_sticky");
   var sticky = navbar_sticky.offsetTop;
   var navbar_height = document.querySelector('.navbar').offsetHeight;

   function myFunction() {
     if (window.pageYOffset >= sticky + navbar_height) {
     navbar_sticky.classList.add("sticky")
     document.body.style.paddingTop = navbar_height + 'px';
     } else {
     navbar_sticky.classList.remove("sticky");
     document.body.style.paddingTop = '0'
     }
   }
   </script>

<script>
  // Get references to the form and input fields
  const form = document.getElementById("login-form");
  const usernameInput = document.getElementById("username");
  const passwordInput = document.getElementById("password");
  const usernameError = document.getElementById("username-error");
  const passwordError = document.getElementById("password-error");

  // Function to validate the form
  function validateForm() {
      // Reset previous error messages
      usernameError.textContent = "";
      passwordError.textContent = "";
```

```
    // Get the values of the input fields
    const username = usernameInput.value.trim();
    const password = passwordInput.value.trim();

    // Validate username (for example, require a minimum length and only letters)
    if (username.length < 3) {
      usernameError.textContent = "Username must be at least 3 characters.";
      return false; // Prevent form submission
    } else if (!/^[a-zA-Z]+$/.test(username)) {
      usernameError.textContent = "Username can only contain letters.";
      return false; // Prevent form submission
    }

    // Validate password (for example, require a minimum length)
    if (password.length < 6) {
      passwordError.textContent = "Password must be at least 6 characters.";
      return false; // Prevent form submission
    }

    // If all validations pass, allow form submission
    return true;
  }

  // Attach the validateForm function to the form's submit event
  form.addEventListener("submit", function (event) {
    if (!validateForm()) {
      event.preventDefault(); // Prevent form submission if validation fails
    }
  });

  // Real-time validation for the username field
  usernameInput.addEventListener("input", function () {
    const username = usernameInput.value.trim();

    // Validate username as the user types (for example, check if it's too short and contains only
letters)
    if (username.length < 3) {
      usernameError.textContent = "Username must be at least 3 characters.";
    } else if (!/^[a-zA-Z]+$/.test(username)) {
      usernameError.textContent = "Username can only contain letters.";
    } else {
      usernameError.textContent = ""; // Clear the error message if valid
    }
  });
</script>

 </body>
 </html>
```

Exhibition.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<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>Art Web</title>
        <link href="{% static '/css/bootstrap.min.css' %}" rel="stylesheet" >
        <link href="{% static '/css/font-awesome.min.css' %}" rel="stylesheet" >
        <link href="{% static '/css/global.css' %}" rel="stylesheet">
        <link href="{% static '/css/product.css' %}" rel="stylesheet">
        <link href="{% static
'/https://fonts.googleapis.com/css2?family=Fraunces:opsz@9..144&display=swap' %}"
rel="stylesheet">
        <script src="{% static '/js/bootstrap.bundle.min.js' %}"></script>

  <!-- code for share -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" integrity="sha512-" crossorigin="anonymous" referrerpolicy="no-
referrer" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" integrity="sha512-" crossorigin="anonymous" referrerpolicy="no-
referrer" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" integrity="sha512-" crossorigin="anonymous" referrerpolicy="no-
referrer" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" integrity="sha512-" crossorigin="anonymous" referrerpolicy="no-
referrer" />

</head>
<body>
<section id="header">
<nav class="navbar navbar-expand-md navbar-light" id="navbar_sticky">
  <div class="container-xl">
    <a class="navbar-brand fs-2 p-0 fw-bold text-white" href="{% static '/index.html' %}"><a
href="#"><img src="{% static '/img/logo.png' %}" width="50" height="50" alt="abc"></a><!--<i
class="fa fa-pencil col_pink me-1 align-middle"></i> art --><span class="col_pink span_1"
>MODERN</span><br><span class="font_12 span_2">ART GALLERY</span></a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav mb-0 ms-auto">
```

```
      <li class="nav-item">
        <a class="nav-link" aria-current="page" href="index.html">Home</a>
      </li>
                <li class="nav-item">
        <a class="nav-link" href="about.html">About </a>
      </li>

                <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle active" href="#" id="navbarDropdown" role="button"
data-bs-toggle="dropdown" aria-expanded="false">
          Events
        </a>
        <ul class="dropdown-menu drop_1" aria-labelledby="navbarDropdown">
          <li><a class="dropdown-item" href="exhibition.html"> Exhibitions</a></li>
          <li><a class="dropdown-item border-0" href="competition.html">competitions</a></li>
        </ul>
      </li>
                <li class="nav-item">
                        <a class="nav-link" href="blog.html">Blog </a>
                  </li>
                  <li class="nav-item">
                        <a class="nav-link" href="contact.html">Contact </a>
                  </li>

                <li class="nav-item dropdown">
   {% if user.is_authenticated %}
      <a class="nav-link dropdown-toggle btn btn-primary d-flex align-items-center justify-
content-center" href="#" id="userDropdown" role="button" data-bs-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
         <i class="fa fa-user"></i>
         <p class="m-0">{{ user.username }}</p>
      </a>

      <div class="dropdown-menu" aria-labelledby="userDropdown">
         {% if user.user_type == 'user' %}
            <a class="dropdown-item" href="{% url 'userprofile' %}">View Profile</a>
         {% elif user.user_type == 'artist' %}
            <a class="dropdown-item" href="{% url 'artistprofile' %}">View Profile</a>
         {% endif %}
         <a class="dropdown-item" href="{% url 'logout' %}">Logout</a>
      </div>
   {% else %}
      <a href="{% url 'login' %}" class="btn btn-primary ml-lg-3">Login/Register</a>
   {% endif %}
</li>
                  <!--<li class="nav-item">
                        <a class="nav-link" href="register.html">Register</a>
                  </li>-->
      </li>

     </ul>
```

```
   </div>
  </div>
</nav>
</section>

<section id="center" class="center_o bg_gray pt-2 pb-2">
 <div class="container-xl">
  <div class="row center_o1">
   <div class="col-md-5">
    <div class="center_o1l">
          <h2 class="mb-0">EXHIBITION</h2>
         </div>
   </div>
   <div class="col-md-7">
    <div class="center_o1r text-end">
          <h6 class="mb-0"><a href="#">Home</a> <span class="me-2 ms-2"><i class="fa fa-
caret-right"></i></span> Product</h6>
         </div>
   </div>
  </div>
 </div>
</section>

<section id="product" class="p_4">
 <div class="container-xl">
  <div class="row product_1">
   <div class="col-md-9">
    <!--<div class="product_1l">
     <p class="mb-0 mt-2">Showing 1-12 of 23 results</p>
         </div>-->
   </div>
   </div>
  </div>

<div class="row product-container">
        {% for product in products %}
          <div class="col-md-3">
            <div class="product-card">
              <a href="{% url 'product_details' product.id %}">
                <img src="{{ product.product_image.url }}" alt="{{ product.product_name }}"
class="img-fluid" oncontextmenu="return false;">
              </a>
              <h4>{{ product.product_name }}</h4>
              <p>Price: Rs.{{ product.price }}</p>
              <p>{{ product.get_status_display }}</p>
              <p>Artist: {{ product.author.username }}</p>

              <a href="#" class="btn-like">
   <i class="fas fa-heart"></i> Like
</a>  
```

```
        <!-- Share Button -->
      <a href="https://www.facebook.com/sharer/sharer.php?u={{ request.build_absolute_uri }}"
target="_blank" class="btn-share">
    <i class="fab fa-facebook-f"></i>
</a>
<a href="https://twitter.com/intent/tweet?url={{ request.build_absolute_uri }}&text={{
product.product_name }}" target="_blank" class="btn-share">
    <i class="fab fa-twitter"></i>
</a>
<a href="mailto:?subject={{ product.product_name }}&body=Check out this product: {{
request.build_absolute_uri }}" class="btn-share">
    <i class="fas fa-envelope"></i>
</a>
<a href="https://www.instagram.com/?url={{ request.build_absolute_uri }}" target="_blank"
class="btn-share">
    <i class="fab fa-instagram"></i>
</a>

                </div>
              </div>
            {% endfor %}
          </div>
    </div>
</section>

  </div>
  <div class="row footer_2 mt-4 text-center">
   <div class="col-md-12">
          <ul>
           <li class="d-inline-block me-3 font_14"><a href="#">CONTACT</a></li>
           <li class="d-inline-block me-3 font_14"><a href="#">PRIVACY POLICY</a></li>
           <li class="d-inline-block me-3 font_14"><a href="#">TERMS OF USE</a></li>
           <li class="d-inline-block font_14"><a href="#">FAQ</a></li>
          </ul>
          <p class="mb-0">© 2013 Your Website Name. All Rights Reserved | Design by <a
class="col_pink" href="http://www.templateonweb.com">TemplateOnWeb</a></p>
         </div>
  </div>
 </div>
</section>
<script src="{% static 'js/jquery-3.6.0.min.js' %}"></script>
<script src="{% static 'js/bootstrap.min.js' %}"></script>

<script>
window.onscroll = function() {myFunction()};

var navbar_sticky = document.getElementById("navbar_sticky");
var sticky = navbar_sticky.offsetTop;
var navbar_height = document.querySelector('.navbar').offsetHeight;

function myFunction() {
```

```
  if (window.pageYOffset >= sticky + navbar_height) {
    navbar_sticky.classList.add("sticky")
            document.body.style.paddingTop = navbar_height + 'px';
  } else {
    navbar_sticky.classList.remove("sticky");
            document.body.style.paddingTop = '0'
  }
}
</script>

</body>

</html>
```

blog.html

```
{% load static %}
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="styles.css">
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Art Web</title>
    <link href="{% static '/css/bootstrap.min.css' %}" rel="stylesheet">
    <link href="{% static '/css/font-awesome.min.css' %}" rel="stylesheet">
    <link href="{% static '/css/global.css' %}" rel="stylesheet">
    <link href="{% static '/css/about.css' %}" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css2?family=Fraunces:opsz@9..144&display=swap"
rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta3/css/all.min.css">
    <script src="{% static '/js/jquery-3.6.0.min.js' %}"></script>
    <script src="{% static '/js/bootstrap.bundle.min.js' %}"></script>

    <!-- code for share -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" integrity="sha512-" crossorigin="anonymous" referrerpolicy="no-
referrer" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" integrity="sha512-" crossorigin="anonymous" referrerpolicy="no-
referrer" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" integrity="sha512-" crossorigin="anonymous" referrerpolicy="no-
referrer" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" integrity="sha512-" crossorigin="anonymous" referrerpolicy="no-
referrer" />
```

```
   <style>
      .product-table {
         width: 100%;
      }

      .product-table th {
         text-align: left;
      }

      .product-table td {
         padding: 5px;
      }

      .product-table img {
         max-width: 100%;
         height: auto;
      }
   </style>
</head>
<body>

{% include "messages.html" %}
<section id="header">
   <nav class="navbar navbar-expand-md navbar-light" id="navbar_sticky">
      <div class="container-xl">
         <a class="navbar-brand fs-2 p-0 fw-bold text-white" href="{% static '/index.html' %}">
            <a href="#"><img src="{% static '/img/logo.png' %}" width="50" height="50"
alt="abc"></a>
            <span class="col_pink span_1">MODERN</span><br>
            <span class="font_12 span_2">ART GALLERY</span>
         </a>
         <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent"
            aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
         </button>
         <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav mb-0 ms-auto">
               <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="index.html">Home</a>
               </li>
               <li class="nav-item">
                  <a class="nav-link active" href="about.html">About </a>
               </li>
               <li class="nav-item dropdown">
                  <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button"
                     data-bs-toggle="dropdown" aria-expanded="false">
                     Events
                  </a>
```

```
                    <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
                        <li><a class="dropdown-item" href="exhibition.html">Exhibitions</a></li>
                        <li><a class="dropdown-item border-0"
href="competition.html">Competitions</a></li>
                    </ul>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="blog.html">Blog </a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="contact.html">Contact</a>
                </li>
                <li class="nav-item dropdown">
    {% if user.is_authenticated %}
        <a class="nav-link dropdown-toggle btn btn-primary d-flex align-items-center justify-
content-center" href="#" id="userDropdown" role="button" data-bs-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
            <i class="fa fa-user"></i>
            <p class="m-0">{{ user.username }}</p>
        </a>

        <div class="dropdown-menu" aria-labelledby="userDropdown">
            {% if user.user_type == 'user' %}
                <a class="dropdown-item" href="{% url 'userprofile' %}">View Profile</a>
            {% elif user.user_type == 'artist' %}
                <a class="dropdown-item" href="{% url 'artistprofile' %}">View Profile</a>
            {% endif %}
            <a class="dropdown-item" href="{% url 'logout' %}">Logout</a>
        </div>
    {% else %}
        <a href="{% url 'login' %}" class="btn btn-primary ml-lg-3">Login/Register</a>
    {% endif %}
</li>
            </ul>
        </div>
    </div>
    </nav>
</section>

<!-- Your existing HTML code for the navigation section -->

<div class="container">
    <center><h2>BLOGS</h2><br><br><br></center>

    {% if blogs %}
        {% for blog in blogs %}
            <div class="blog-item">
                <h3>{{ blog.title }}</h3>
                <p>{{ blog.publishingDate }}</p>
                <p>{{ blog.description }}</p>
```

```
                        <p>Author: {{ blog.author.username }}</p>

            {% if blog.image %}
               <img src="{{ blog.image.url }}" alt="{{ blog.title }}"><br><br>
            {% endif %}

             <!-- Share Button -->
          <a href="https://www.facebook.com/sharer/sharer.php?u={{ request.build_absolute_uri
}}" target="_blank" class="btn-share">
   <i class="fab fa-facebook-f"></i>
</a>
<a href="https://twitter.com/intent/tweet?url={{ request.build_absolute_uri }}&text={{
product.product_name }}" target="_blank" class="btn-share">
   <i class="fab fa-twitter"></i>
</a>
<a href="mailto:?subject={{ product.product_name }}&body=Check out this product: {{
request.build_absolute_uri }}" class="btn-share">
   <i class="fas fa-envelope"></i>
</a>
<a href="https://www.instagram.com/?url={{ request.build_absolute_uri }}" target="_blank"
class="btn-share">
   <i class="fab fa-instagram"></i>
</a>
          </div>
       {% endfor %}
   {% else %}
       <p>No blogs available.</p>
   {% endif %}

</div>

<div class="row footer_2 mt-4 text-center">
   <div class="col-md-12">
          <ul>
           <li class="d-inline-block me-3 font_14"><a href="#">CONTACT</a></li>
           <li class="d-inline-block me-3 font_14"><a href="#">PRIVACY POLICY</a></li>
           <li class="d-inline-block me-3 font_14"><a href="#">TERMS OF USE</a></li>
           <li class="d-inline-block font_14"><a href="#">FAQ</a></li>
          </ul>
          <p class="mb-0">© 2013 Modern Art Gallery. All Rights Reserved | Design by <a
class="col_pink" href="http://www.templateonweb.com">Art Gallery</a></p>
         </div>
  </div>
 </div>
</section>
<script src="{% static 'js/jquery-3.6.0.min.js' %}"></script>
<script src="{% static 'js/bootstrap.min.js' %}"></script>



<script>
```

```
window.onscroll = function() {myFunction()};

var navbar_sticky = document.getElementById("navbar_sticky");
var sticky = navbar_sticky.offsetTop;
var navbar_height = document.querySelector('.navbar').offsetHeight;

function myFunction() {
  if (window.pageYOffset >= sticky + navbar_height) {
    navbar_sticky.classList.add("sticky")
        document.body.style.paddingTop = navbar_height + 'px';
  } else {
    navbar_sticky.classList.remove("sticky");
        document.body.style.paddingTop = '0'
  }
}
</script>
</body>
</html>
```

Competition.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<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>Art Web</title>
        <link href="{% static '/css/bootstrap.min.css' %}" rel="stylesheet" >
        <link href="{% static '/css/font-awesome.min.css' %}" rel="stylesheet" >
        <link href="{% static '/css/global.css' %}" rel="stylesheet">
        <link href="{% static '/css/product.css' %}" rel="stylesheet">
        <link href="{% static
'/https://fonts.googleapis.com/css2?family=Fraunces:opsz@9..144&display=swap' %}"
rel="stylesheet">
        <script src="{% static '/js/bootstrap.bundle.min.js' %}"></script>
        <!-- Include Font Awesome CSS -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta3/css/all.min.css">

<!-- code for share -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" integrity="sha512-" crossorigin="anonymous" referrerpolicy="no-
referrer" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" integrity="sha512-" crossorigin="anonymous" referrerpolicy="no-
referrer" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" integrity="sha512-" crossorigin="anonymous" referrerpolicy="no-
referrer" />
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" integrity="sha512-" crossorigin="anonymous" referrerpolicy="no-referrer" />

</head>
<body>
<section id="header">
<nav class="navbar navbar-expand-md navbar-light" id="navbar_sticky">
  <div class="container-xl">
    <a class="navbar-brand fs-2 p-0 fw-bold text-white" href="{% static '/index.html' %}"><a href="#"><img src="{% static '/img/logo.png' %}" width="50" height="50" alt="abc"></a><!--<i class="fa fa-pencil col_pink me-1 align-middle"></i> art --><span class="col_pink span_1" >MODERN</span><br><span class="font_12 span_2">ART GALLERY</span></a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav mb-0 ms-auto">

        <li class="nav-item">
          <a class="nav-link" aria-current="page" href="{% url 'index' %}">Home</a>
        </li>
                    <li class="nav-item">
          <a class="nav-link" href="{% url 'about' %}">About </a>
        </li>

                    <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle active" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            Events
          </a>
          <ul class="dropdown-menu drop_1" aria-labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="{% url 'exhibition' %}"> Exhibitions</a></li>
            <li><a class="dropdown-item border-0" href="{% url 'competition' %}"> competitions</a></li>
          </ul>
        </li>
                    <!--<li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            Blog
          </a>
          <ul class="dropdown-menu drop_1" aria-labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="blog.html"> Blog</a></li>
            <li><a class="dropdown-item border-0" href="blog_detail.html"> Blog Detail</a></li>
          </ul>
        </li>-->

                    <li class="nav-item">
```

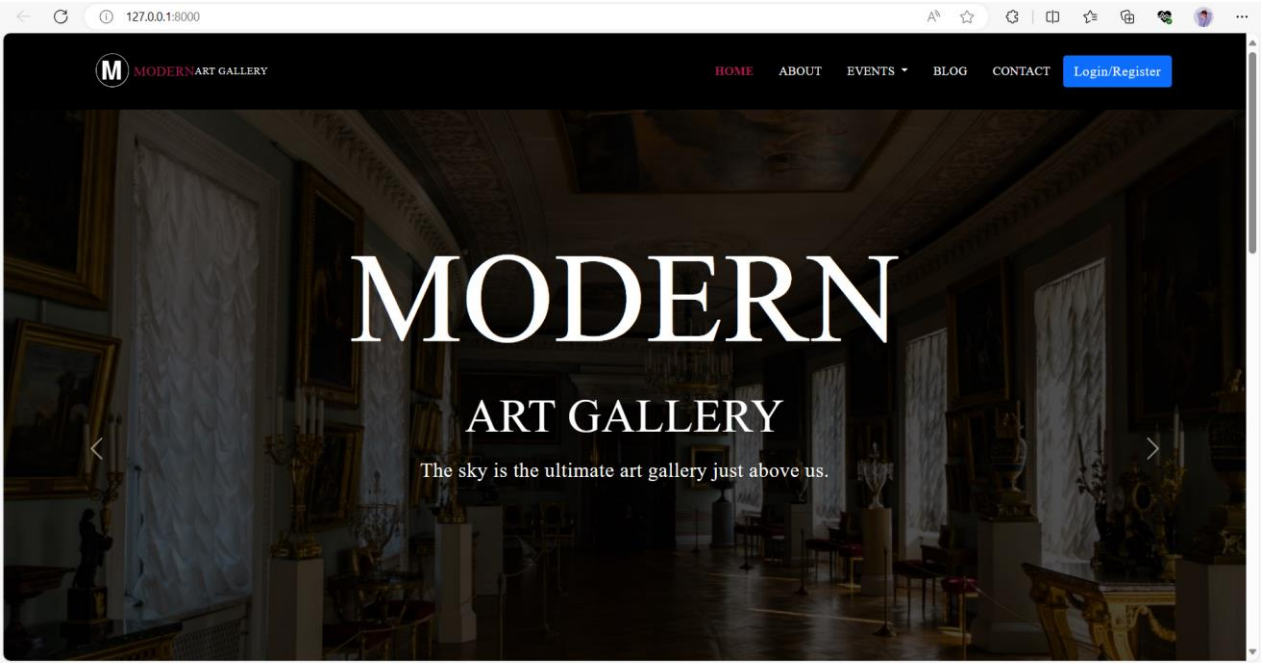
```
                    <a class="nav-link" href="{% url 'blog' %}">Blog </a>
                  </li>
                   <li class="nav-item">
                    <a class="nav-link" href="{% url 'contact' %}">Contact </a>
                  </li>

                 <!--<li class="nav-item">
                    <a class="nav-link" href="Events.html">Events </a>
                  </li>-->

                  <!--<li class="nav-item">
                    <a class="nav-link" href="artstore.html">Art Store </a>
                  </li>-->

                 <li class="nav-item dropdown">
   {% if user.is_authenticated %}
      <a class="nav-link dropdown-toggle btn btn-primary d-flex align-items-center justify-
content-center" href="#" id="userDropdown" role="button" data-bs-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
         <i class="fa fa-user"></i>
         <p class="m-0">{{ user.username }}</p>
      </a>

      <div class="dropdown-menu" aria-labelledby="userDropdown">
         {% if user.user_type == 'user' %}
            <a class="dropdown-item" href="{% url 'userprofile' %}">View Profile</a>
         {% elif user.user_type == 'artist' %}
            <a class="dropdown-item" href="{% url 'artistprofile' %}">View Profile</a>
         {% endif %}
         <a class="dropdown-item" href="{% url 'logout' %}">Logout</a>
      </div>
   {% else %}
      <a href="{% url 'login' %}" class="btn btn-primary ml-lg-3">Login/Register</a>
   {% endif %}
</li>

     </ul>
   </div>
  </div>
</nav>
</section>
<section>
<center><h1>COMPETITIONS</h1></center>
<div class="row competition-container">
   {% for competition in competitions %}
   <div class="col-md-3">
      <div class="competition-card">
         <a href="#">
            <img src="{{ competition.image.url }}" alt="{{ competition.competition_name }}"
class="img-fluid" oncontextmenu="return false;">
         </a>
```

```
        <h4>{{ competition.competition_name }}</h4>
        <p>Artist: {{ competition.author.username }}</p>
        <p>Description: {{ competition.description }}</p>
        <p>Date: {{ competition.date }}</p>
        <p>Time: {{ competition.time }}</p>
        <p>Location: {{ competition.location }}</p>
        <p>Fee: {{ competition.fee }}</p>

        <!--<a href="#" class="btn-like">
           <i class="fa fa-thumbs-up"></i> Like
        </a>  -->

        <!-- Share Button -->
         <a href="https://www.facebook.com/sharer/sharer.php?u={{ request.build_absolute_uri
}}" target="_blank" class="btn-share">
   <i class="fab fa-facebook-f"></i>
</a>
<a href="https://twitter.com/intent/tweet?url={{ request.build_absolute_uri }}&text={{
product.product_name }}" target="_blank" class="btn-share">
   <i class="fab fa-twitter"></i>
</a>
<a href="mailto:?subject={{ product.product_name }}&body=Check out this product: {{
request.build_absolute_uri }}" class="btn-share">
   <i class="fas fa-envelope"></i>
</a>
<a href="https://www.instagram.com/?url={{ request.build_absolute_uri }}" target="_blank"
class="btn-share">
   <i class="fab fa-instagram"></i>
</a>
        <h6 class="font_14 mb-0 mt-4"><a class="button p-3 pt-2 pb-2"
href="#">APPLY</a></h6><br>
      </div>
   </div>
   {% endfor %}
</div>
   </div>
</section>
<script src="{% static 'js/jquery-3.6.0.min.js' %}"></script>
<script src="{% static 'js/bootstrap.min.js' %}"></script>
<script>
window.onscroll = function() {myFunction()};

var navbar_sticky = document.getElementById("navbar_sticky");
var sticky = navbar_sticky.offsetTop;
var navbar_height = document.querySelector('.navbar').offsetHeight;

function myFunction() {
  if (window.pageYOffset >= sticky + navbar_height) {
    navbar_sticky.classList.add("sticky")
        document.body.style.paddingTop = navbar_height + 'px';
  } else {
```

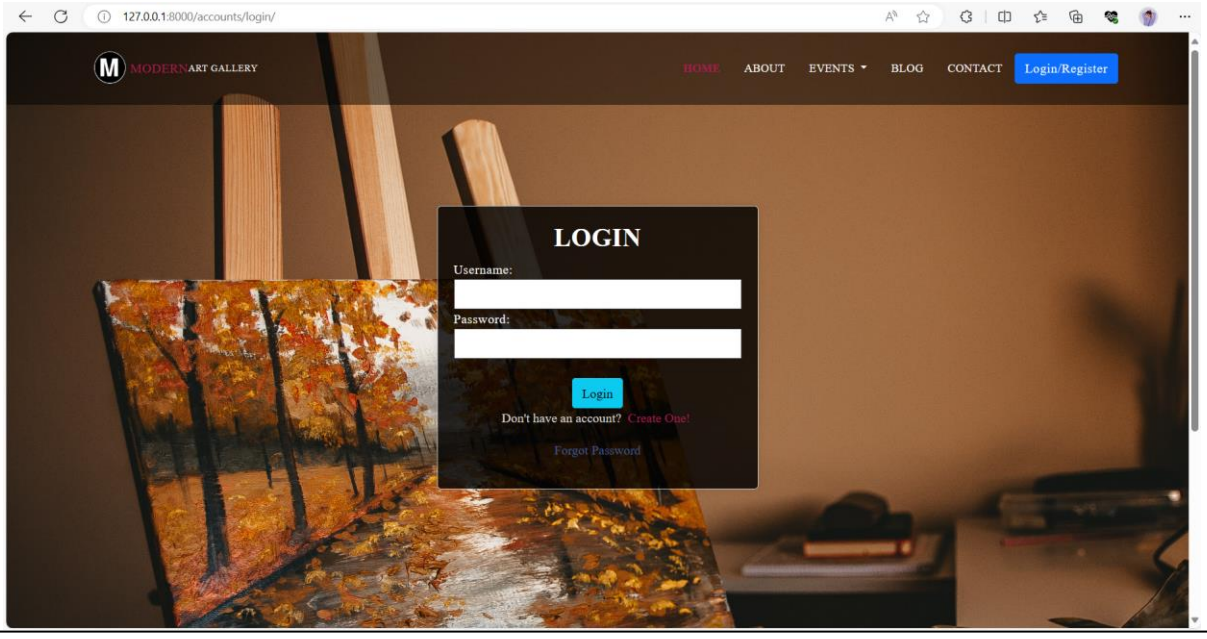
```
    navbar_sticky.classList.remove("sticky");
          document.body.style.paddingTop = '0'
  }
}
</script>

</body>

</html>
```

9.1 Screen Shots

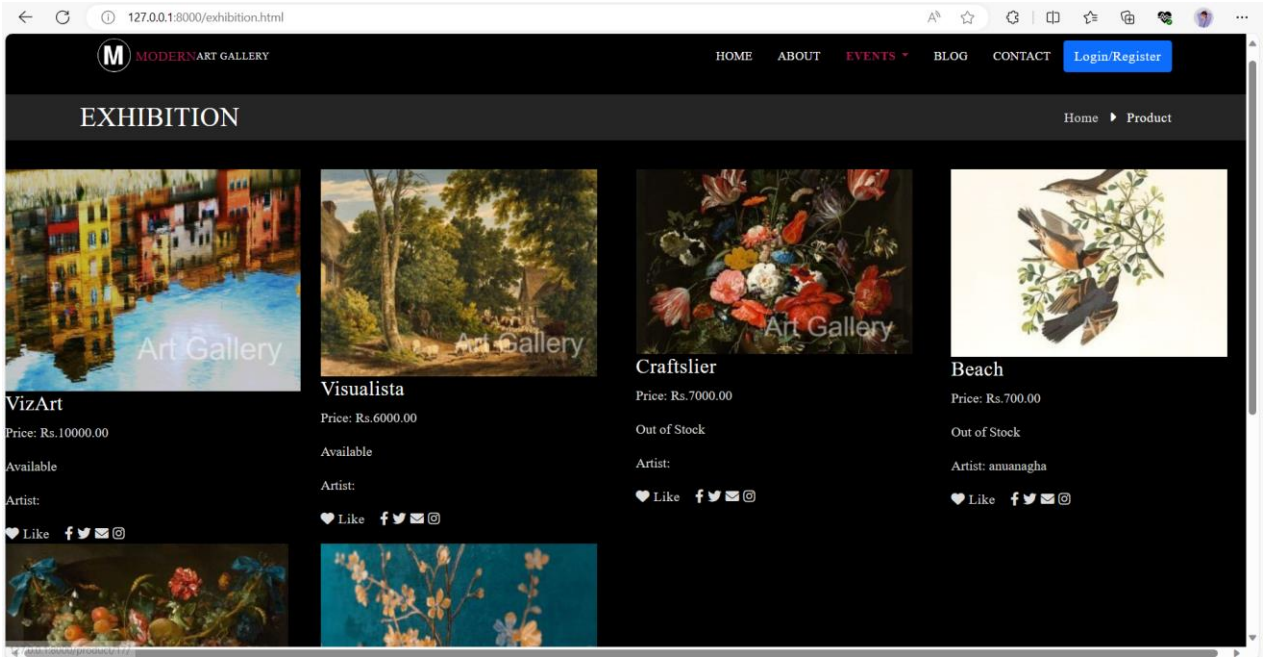
Indexpage



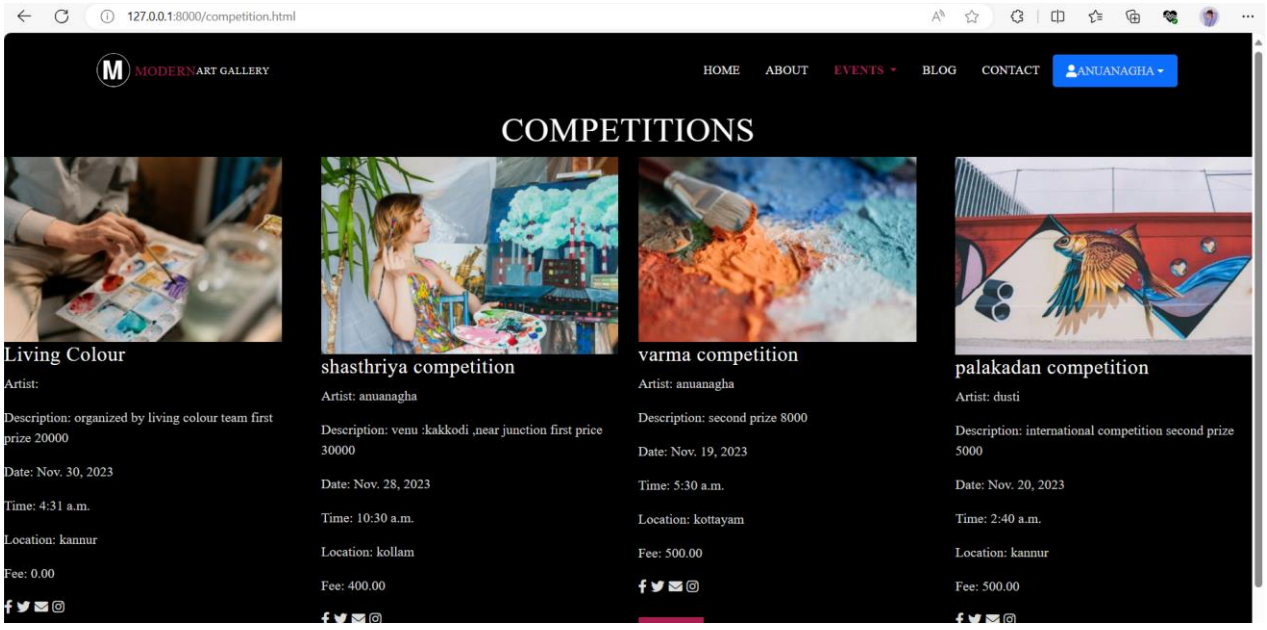
LoginPage



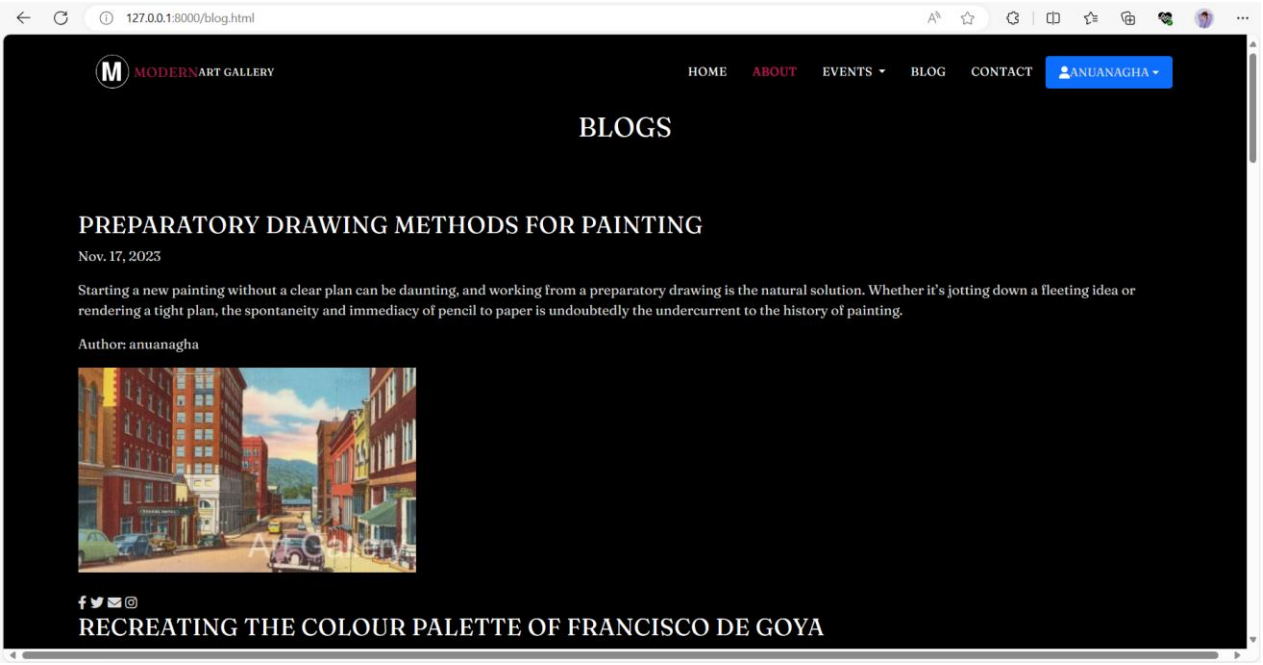
Exhibition Page



Competition Page



Blog Page


127.0.0.1:8000/blog.html
MODERNART GALLERY
HOME
ABOUT
EVENTS
BLOG
CONTACT
ANUANAGHA
BLOGS
PREPARATORY DRAWING METHODS FOR PAINTING
Nov. 17, 2023
Starting a new painting without a clear plan can be daunting, and working from a preparatory drawing is the natural solution. Whether it's jotting down a fleeting idea or rendering a tight plan, the spontaneity and immediacy of pencil to paper is undoubtedly the undercurrent to the history of painting.
Author: anuanagha
RECREATING THE COLOUR PALETTE OF FRANCISCO DE GOYA