DECLARATION BY THE CANDIDATE

I, the undersigned solemnly declare that the report of the project work entitled ***3D Art Gallery*** is based on my own work carried out during the course of my study under the guidance of ***Prof. Vibhore Jain Sir.***

I assert that the statements made and conclusions drawn are an outcome of the project work. I further declare that to the best of my knowledge and belief that the report does not contain any part of any work which has been submitted for the award of any other degree/diploma/certificate in this University/deemed University of India or any other country. All helps received and citations used for the preparation of the thesis have been duly acknowledged.

Signature of the Coordinator	**Signature of the Guide**	**Signature of the Candidate**
Prof. Dinesh K. Bhawnani	**Prof. Vibhore Jain**	**Pranjal Hinduja**
Assistant Professor	**Designation of the Guide**	**URN:**
Computer Science &	**Computer Science &**	**Enrollment No.:**
Engineering	**Engineering**	**Sem: 5th sem**

CERTIFICATE OF THE SUPERVISOR

This is to certify that the report of the project entitled **3D Art Gallery** is a record of bonafide research work carried out by **Pranjal Hinduja bearing University Roll No.300102220057 & Enrollment No.:** under my guidance and supervision for the award of Degree of Bachelor of Technology in Computer Science and Engineering, Bhilai Institute of Technology, Durg, an autonomous institute affiliated to Chhattisgarh Swami Vivekanand Technical University, Bhilai (CG).

To the best of my knowledge and belief the project

- Embodies the work of the candidate him/herself,
- Has duly been completed,
- Fulfills the requirement of the Ordinance relating to the B.Tech. degree of the University and
- Is up to the desired standard both in respect of contents and language for being referred to examiners.

_____

(Signature of the Guide)

Name: Prof. Vibhore Jain

Designation: Assistant Professor

Department: Computer Science and Engineering

CERTIFICATE BY THE EXAMINERS

The Project entitled 3D Art Gallery submitted by **Pranjal Hinduja (University Roll No.: 300102220057 Enrollment No. BK4145)** has been examined by the undersigned as a part of the examination and is hereby recommended for the award of the degree of Bachelor of Technology in Computer Science Engineering, Bhilai Institute of Technology, Durg, an autonomous institute affiliated to Chhattisgarh Swami Vivekanand Technical University, Bhilai (CG).

Internal Examiner External Examiner

Date: Date:

ACKNOWLEDGEMENT

I have great pleasure in the submission of this project report entitled **3D Art Gallery** in partial fulfillment the degree of Bachelor of Engineering (CSE). While submitting this Project report, I take this opportunity to thank those directly or indirectly related to project work.

I would like to thank my guide Prof. Vibhore Jain who has provided the opportunity and organizing project for me. Without his active co-operation and guidance, it would have become very difficult to complete task in time.

I would like to express sincere thanks and gratitude to Dr. Arun Arora, **Director of the Institution**, Dr. Mohan Gupta, **Principal of the Institution**, Dr. Sunita Soni, **Head of the Department** Computer Science & Engineering for their encouragement and cordial support.

While Submission of the project, I also like to thanks to Prof. Dinesh Kumar Bhawnani and Prof. Vibhore Jain, **Project Coordinator, faculties** and all the staff of department of Computer Science & Engineering, **Bhilai Institute of Technology, Durg** for their continuous help and guidance throughout the course of project.

Acknowledgement is due to our parents, family members, friends and all those persons who have helped us directly or indirectly in the successful completion of the project work.

_____

(Signature of the candidate)

Name: Pranjal Hinduja

Bhilai Institute of Technology,Durg,Chhattisgarh

TABLE OF CONTENTS

1.INTRODUCTION

1.1 OBJECTIVE

Being an artist has always been a struggling occupation. They need a creative approach and a straight commitment to seek peak excellence. The real fight begins when these artists exhibit their work at galleries. They usually reach out to intermediaries that allow them to display their artwork and abilities in public locations. Following are some use cases of blockchain based virtual art galleries:

-Every year, online art sales grow by high numbers, but only a few people buy artwork online. However, tokenizing them with smart contract development makes them available in segments and opens doors of investment for art lovers. Paintings that were difficult to sell can now be accessed by potential buyers through the democratized system.

-The potential of blockchain to sustain valuable digital artwork in the technical era. Optimizing blockchain technology and unique digital codes gives an ideal security basis for increasing the demand for digital art galleries.

We are creating an art gallery where users can move around freely and see various artworks. These artworks will be stored as NFTs on the blockchain. Our application is comprised of two parts, an NFT Marketplace Smart contract that will be present on the blockchain, and a 3D game built using Unity3D game engine that will be used to fetch the NFTs listed in the Marketplace smart contract and displays them inside an interactive 3D art gallery.

This blockchain based project would allow artists to flourish in their work. Autonomous works of art built upon a blockchain-based network and with smart contract development facilitated the sculpture.

Online Art Gallery is an online application, which is used to display and sell art works of artist irrespective of their nationality, gender and other narrow consideration, through auction. Artist can register online for being a member in the art gallery with their respective metamask account and each artist can upload the digital copy of their artwork under the respective categories. They can host their artwork either for auction or for fixed price.

Art lovers must go to the art exhibition to collect their favorite arts or painting. But now-a-days they are not getting enough time to go to the galleries and collect the arts and paintings. In 3D Art Gallery Customer can also register online and they can browse art works that are arranged in different categories scientifically. Each Customer can create their own gallery to see his favorite art works without much difficulty and each user has the right to purchase an artwork using the integrated

payment gateway and participate in auction by submitting their bids. Qualified bidder should remit the amount using payment gateway and after each valid payment they can own their artwork.

Online Art Gallery is an application software, and it is very helpful for the art lovers and others who want to know the addresses where this kind of arts will we sold. This application helps the end-users to search their arts and paintings and they can place order for the selected pieces. The end user can also get the information about the art exhibition and the respective address, so, that they can visit to those exhibitions. Art Gallery brings you an opportunity to view online art exhibitions at our Online Art Gallery, we bring you details of all art exhibitions held in the past and the forthcoming show. The Online Art Gallery is updated daily, so the user can view and buy the latest collection of contemporary art online from any where in the world. You can view and buy the latest Indian contemporary art collection available at their exhibitions and at their online gallery.

The system to be developed is for portraying and selling the pieces of art in an Art Gallery, based on the values depicted in the art and the overall quality work of the different Artists. Different conditions must be satisfied while selling or buying through NFT's.

This document is meant to delineate the features of the art gallery, to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other. This document explains the specifications and requirements of the "Art Gallery" a website. It describes the functional features and non-functional requirements of the site such as interface, design and other details related to Art Gallery. This document is intended for the entrepreneurs who are interested in opening their Art Gallery and students also can read this documentation for learning purposes.

Blockchain based art galleries would allow artists to flourish in their work. Autonomous works of art built upon a blockchain-based network and with smart contract development facilitated the sculpture.

An NFT gallery is not different from more popular traditional places. Most digital exhibition spaces start with curating the artworks or artists representing a specific theme. NFT art galleries usually have websites where people can purchase the art pieces on display. Here, one can easily authenticate artworks and complete art sales themselves. It is a faster process compared to traditional art galleries. NFT galleries play a crucial role in improving the experience of finding and acquiring these digital art pieces.

Blockchain combines the history of owners, transaction dates, fees, and many other insights about the artwork to create a unique digital certificate. This certificate is impossible to forge and solves the issue of authenticity of the NFTs you display in your gallery.

1.2 PROJECT DESCRIPTION

1.2.1 Website Perspective

This web project is totally independent and needs a server i.e., web server or a local server, apache in our case. The client sends a request to the server, the application host on some server will communicate to the application and to the database management system if needed to facilitate the request. The database server will respond to the user according to demands of the user.

1.2.2 Online Art Gallery Functions

This web project is all about an Art gallery, which consists of two views.

1. Admin Panel
2. Client Panel

Where the admin is responsible for the whole database. Admin can perform all the designs, creates and update operations. Admin will be able to

- May add new items in the database using the admin panel
- May update the items in the database
- May delete the items in the database

A client who will use the application from the front end, will be able to visit the website. Clients will be able to select different types of paintings i.e., according to their interest. The client will be able to search different paintings with the name.

Art Gallery embeds the following features and function to its users, i.e., admin and client.

- Should make the admin able to login into the admin panel after authentication.
- Should make the admin able to view list all clients.
- Should make the admin able to view the entire available Gallery in the database.
- Should make the admin able to insert new paintings, new profile and new categories.
- Should make the admin able to make new admin, delete or update the existing user.
- Should make the client able to select more than one Gallery.
- Should provide the security for the system.

1.2.3 User Characteristics

There are two types of system users. The first is user, user can login and upload his information and check another profile. The second user is the administrative who have the capability for maintain the record. Admins manage the whole database. Admin is the person who will control the application by entering new events, by updating or deleting new events. I.e., paintings, and categories. The user should know the details of a transaction.

1.2.4 General Constraints

Web application is based on internet which required both hardware and software, so we will facilitate the requirements about software which are given below. Web applications used are the specific software that are include PHP 6.0, Apache and data Base scheme used MySQL and hardware

requirements are dual core process with at least 1GB RAM for used this software. For better results better hardware will be consider a plus point.

2.SYSTEM STUDY

2.1 EXISTING AND PROPOSED SYSTEM

2.1.1 EXISTING SYSTEM

Blockchain. Art (BCA) has developed a unique blockchain that is dedicated to the art industry and transacting digital & time-based artworks. Like many other industries, the art world is the midst of a digital renaissance. Galleries and museums are urgently searching for more channels to connect with their buyers and audiences, while artists are experimenting more and more with digital mediums. Established galleries are fast-paced, lean businesses that generally do not have the administrative bandwidth to quickly develop and deploy emerging technologies like blockchain; the same goes for many museums and cultural nonprofits. These arts organizations need a platform that's been designed specifically for them — one which can support their pre-existing digital channels and seamlessly integrate with their brand. BCA is an industry-specific framework with a user-friendly interface that's written in proper art world vernacular. BCA is partnering with galleries, sought-after artists and beloved institutions. In doing so, we are enabling a new generation of collector's access to the art market. BCA provides established art industry veterans (and rising Blockchain + The Art World algorithmically, without sacrificing security or confidentiality. Smart contracts and hash algorithms ensure that the chain remains immutable, since every hash is dependent on the hash (or block) that came before it. Thus, blockchain-based documents cannot be tampered with or misappropriated in any way. There are currently only a handful of companies in the blockchain space that cater exclusively to the art world establishment and, among those, none that were developed to integrate with the gallery ecosystem. In general, the art world's blockchain companies are either purely for provenance registration (i.e., Artery) or function as catchall marketplaces where unknown artists can upload and sell their own work directly to collectors. BCA is a necessary upgrade for galleries and other art world participants who are seeking access to the digital art market without sacrificing their brand equity or relying on tech developers who do not understand the art industry. BCA is being built by seasoned art world professionals with input from future gallery partners and artists to ensure that their needs and priorities are accounted for in every stage of BCA's development.

2.1.2 PROPOSED SYSTEM

3D art gallery is web application based on blockchain which connects artists with their customers, this website will help the artists to showcase their art in the form of NFTs in this website and helping them to sell it to prospective customers. It will also help the customers to discover and search the art which they like, and they will have an option to bid for the art and buy the art if they want. There will be three types of user's roles on the website:

-The first one is buyer who can search and discover the art without any need to register or login but if they want to buy any art they have to register and login to buy the art.
-Second one is a seller(artist) who must register and login to post their art works for exhibiting and selling on the website.
-Third user is admin who specifies authority and restrictions on the users. There will be a separate console for artist, buyer and administrator.

NFT galleries play a crucial role in improving the experience of finding and acquiring these digital art pieces. Three dimensional virtual exhibitions provide a compelling environment to recreate the navigation in the physical ambience of a museum. The digital reproductions of the artworks can be seen side by side in a 3D environment making possible to get an idea of their dimensions. Moreover, in the case of a 3D artwork, a virtual navigation allows to view the object and requires modeling the 3D virtual space and digitizing the artworks. After that, it is necessary to integrate the artworks in the virtual space, either as a reproduction of an actual exhibition or as a virtual exhibition conceived with the purpose of promoting the collection of the museum.

An NFT gallery website is simply a place that displays a collection of this unique art.
Although NFTs are digital artworks, not every digital media presentation qualifies as an NFT exhibition. Also, these unique showcases can be virtual or take place at a physical location. To be more precise, NFTs represent the digital contracts that determine who own which music, video, meme, or artwork on the web. This feature is possible through blockchain technology. The system confirms every unique NFT address using an online ledger that keeps track of transactions.

The overall result is a virtual exhibition which can be available on a website. In a virtual visit, the navigation inside a 3D virtual space should be performed interactively and in real time, giving the participant the sensation of being inside the recreated space. In the cultural domain, 3D virtual environments can recreate historical sites. The virtual model of a building is sometimes the only way to view the changes regarding architecture of a building over time or its appearance before damage caused by weathering, natural disasters or wars. These virtual reconstructions have in common with museum virtual exhibition the aim of cultural diffusion and the techniques used both for 3D reconstruction and presentation.

2.2 FEASIBILITY STUDY

In this section we discuss the various aspects of our project, their feasibility and scalability. We describe what features in the scope of the software are feasible and what are not. We have also discussed the various strategies to achieve feasibility in software aspects that have been specified in the requirements of the software to be developed.

An online art gallery is a website that displays artworks. Usually, the online gallery is run as a business, with the purpose of displaying the artwork being to promote it to potential buyers. Other variations include: An online art market for collectors also known as an online secondary market.

There are some number of factors that are fulfilled to make the project work according to the requirements described in document. One such requirement is that the mentioned system will use the latest tool available in the market, so it is assumed that user will use the latest web browser for working that will load the UI properly so that it ensures a smooth experience for each user category.

Also, there is huge involvement of JavaScript code so it is assumed that the user will not disable the JavaScript in the browser to run the application smoothly. Furthermore, Users of the system should be technically skilled persons and they should have a basic understanding of how to handle this system. Here we have categorized the functionalities of the software according to their feasibility and project scope that will be realistic to achieve and implement:

In Scope:

a. Maintain records of art works made by artists as a portfolio.
b. Type of artwork artists want to display.
c. User categorization based on their personalized art choice.
d. Necessary information about buyers like their transaction history, artworks owned.
e. Cryptocurrency is owned by the buyer, and it should be sufficient to buy the desired artwork.

Out of Scope:

a. Internal working details of blockchain used.
b. Any market related prediction.
c. Art piece customization as per user requirement is not feasible to achieve.

This application helps the end-users to search their arts and paintings and they can place order for the selected pieces. Art Gallery brings you an opportunity to view online art exhibitions at our Online Art Gallery, we bring you details of all art exhibitions held in the past and the forthcoming show. The Online Art Gallery is updated daily, so the user can view and buy the latest collection of contemporary art online from anywhere in the world. You can view and buy the latest Indian contemporary art collection available at their exhibition through NFT based buying and bidding system.

2.3 TOOLS AND TECHNOLOGIES USED

2.3.1 Blockchain

Blockchain first appeared via a cryptography email in 2008 when Satoshi Nakamoto (an anonymous person or a group of people) published their infamous "Bitcoin Whitepaper."

A blockchain is a digital database that's decentralized and maintained by numerous users across a network of computers, or "nodes." Every database record is secured through an algorithm that is cryptographically linked to its preceding record, ensuring the immutability of the entire chain of records or "blocks" (hence the name: blockchain). Each block is assigned a unique ID that's known as a "hash." A hash is another word for the algorithm that links the blocks together. It's a kind of cryptographic signature, like a "digital fingerprint." This problem is rooted in game theory and essentially boils down to one question: how does one enable trust in a digital transaction? Satoshi's answer — the "proof-of-work" chain. Proof-of-work (abbreviated to PoW) is a "consensus protocol" initiated by Bitcoin, the cryptocurrency that runs on blockchain and can be traded and spent without intermediary banks or governmental oversight. The "consensus protocol" determines that a block has been appropriately hashed and encrypted and can therefore be added to the blockchain.

2.3.2 Ethereum Test net Goerli

The Ethereum Implementation Proposal also introduced the "smart contract," known as ERC-721.11 A smart contract is defined as a computer protocol that digitally facilitates, verifies, or enforces the negotiation or performance of a contract, independent of a third party. Everything uploaded to the blockchain becomes data and receives a hash, resulting in an encrypted value that does not resemble the original data. This encryption process allows that data to then be verified algorithmically, without sacrificing security or confidentiality. Smart contracts and hash algorithms ensure that the chain remains immutable, since every hash is dependent on the hash (or block) that came before it. Thus, blockchain-based documents cannot be tampered with or misappropriated in any way.

2.3.3 Chainsafe

ChainSafe is a leading blockchain R&D firm, specializing in protocol engineering, cross-chain interoperability & web3 gaming. Chainsafe is a reliable implementer that steadily tries to improve their own performance and is constantly on the lookout for new ways to drive the growth of the web3 ecosystem. As a valuable contributor to the Kusama and Polkadot ecosystem, ChainSafe continues to produce a wide variety of infrastructure for both its clients and the wider community.

2.3.4 Unity 3D

Unity is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005 at Apple Worldwide Developers Conference as a Mac OS X game engine. The

engine has since been gradually extended to support a variety of desktop, mobile, console and virtual reality platforms. It is particularly popular for iOS and Android mobile game development, is considered easy to use for beginner developers, and is popular for indie game development. The engine can be used to create three-dimensional (3D) and two-dimensional (2D) games, as well as interactive simulations and other experiences. The engine has been adopted by industries outside video gaming, such as film, automotive, architecture, engineering, construction, and the United States Armed Forces.

2.3.5 VS Code

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool among 82,000 respondents, with 70% reporting that they use it.

2.3.6 MetaMask

MetaMask allows users to store and manage account keys, broadcast transactions, send and receive Ethereum-based cryptocurrencies and tokens, and securely connect to decentralized applications through a compatible web browser or the mobile app's built-in browser. Websites or other decentralized applications are able to connect, authenticate, and/or integrate other smart contract functionality with a user's MetaMask wallet (and any other similar blockchain wallet browser extensions) via JavaScript code that allows the website to send action prompts, signature requests, or transaction requests to the user through MetaMask as an intermediary.

2.3.7 WebGL

WebGL (Web Graphics Library) is a JavaScript API for rendering interactive 2D and 3D graphics within any compatible web browser without the use of plug-ins. WebGL is fully integrated with other web standards, allowing GPU-accelerated usage of physics and image processing and effects as part of the web page canvas. WebGL elements can be mixed with other HTML elements and composited with other parts of the page or page background. WebGL programs consist of control code written in JavaScript and shader code that is written in OpenGL ES Shading Language (GLSL ES), a language like C or C++, and is executed on a computer's graphics processing unit (GPU). WebGL is designed and maintained by the non-profit Khronos Group. On February 9, 2022, Khronos Group announced WebGL 2.0 support for all major browsers.

2.3.8 C#

C# is a general-purpose, high-level multi-paradigm programming language. C# encompasses static typing, strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented, and component-oriented programming disciplines. The C# programming language was designed by Anders Hejlsberg from Microsoft in 2000 and was later approved as an international standard by Ecma (ECMA-334) in 2002 and ISO/IEC (ISO/IEC 23270) in 2003. Microsoft introduced C# along with .NET Framework and Visual Studio, both of which were closed source. At the time, Microsoft had no open-source products. Four years later, in 2004, a free and opensource project called Mono began, providing a cross-platform compiler and runtime environment for the C# programming

language. As of November 2022, the most recent stable version of the language is C# 11.0, which was released in 2022 in .NET 7.0.

2.4 HARDWARE AND SOFTWARE REQUIREMENTS

- **Browser with blockchain wallet support (chrome, brave, etc.):** The compatible user browser to get in touch with this project should have blockchain wallet support in order to have complete access to the project and a smooth experience.

- **Unity 3D:** The concerned user should have the application software Unity 3D game engine installed in their system.

- **PC with min 4 gb ram:** The user's system should have minimal 4 gb ram storage.

- **Knowledge about NFT exchange in Blockchain:** In order to operate via the project, the user is expected to have prior knowledge of NFTs and the blockchain market so that he can have a wiser buying and selling experience.

3. SOFTWARE REQUIREMENTS SPECIFICATION

3.1 USERS

There are two types of system users. The first is user, user can login and upload his information and check another profile. The second user is the administrative who have the capability for maintain the record. Admins manage the whole database. Admin is the person who will control the application by entering new events, by updating or deleting new events. i.e., paintings, and categories. The user should know the details of a transaction.

3.2 FUNCTIONAL REQUIREMENTS:

3.2.1 INTERFACE REQUIREMENTS

3.2.1.1 User interface

There are two types of user interface.

(1) Client User Interface
(2) Administrator User Interface

3.2.1.2 Client interface

The minimal requirements are that the user would be able to communicate to the system using GUI. GUI stand for graphical User Interface. GUI is bases on HTML5, CSS3, JavaScript and different frameworks and libraries. a JS library and Bootstrap, a CSS framework. Different GUI based component are used for the following actions.

3.2.1.3 Administrator Interface

The minimal requirements required for the administrator interface that the admin will access the control panel through a GUI using some web browser. The GUI again bases on unity. No command line utility will be given to the user to access the system.

3.2.1.4 Hardware Interfaces

Art Gallery is an app that is why all hardware interface of this, will be those of server On top of which it will be running. Hence the app will incorporate the server for all the hardware instances such as CPU, memory and communication.

3.2.1.5 Software Interfaces

As our mentioned system runs on a polygon based blockchain, so a browser that supports blockchain wallet extensions and transaction of NFTs is required.

3.2.2 FUNCTIONAL REQUIREMENTS

This section narrates the features and process performed by the web site. Firstly, the following features of the system and their interaction with administrator are described check paintings, delete artist Update profiles, Update User, and Delete User. Then the following features related to User are described: Search Paintings, Registration, and feedback, they are followed by a detailed specification of the functionality of the art gallery. Data definition will also be discus in this document. Data flow diagram will be provided in these documents.

3.2.2.1 Add Artist profile

- This functional feature deals with the administrator. The admin can add new painting in the database by using the interface of the Art Gallery.
- The admin will enter the Artist name, Artist address, time, date and other details about profile using different UI elements.
- The server with communicate the polygon marketplace to insert the new item in the database.
- A success message will be shown to the user and database will be updated.
- An error message will be shown upon some error while updating the item in the database.

3.2.2.2 Delete Artist profile

- This functional feature also deals with the administrator. The admin can delete Artist in the database by using the interface of the art gallery.
- The admin will delete the item in the database with the help of ID of the artist. He would select the id of the artist from a dropdown menu.
- The request will communicate the distributed nodes to delete the item in the database.
- A success message will be shown to the user and database will be updated.
- An error message will be shown upon some error while deleting the item in the blockchain.

3.2.2.3 Make A New User

- This functional feature too deals with the administrator. The admin can make new admin in the blockchain by using the interface of the art gallery.
- The admin will enter the name, username, email id and password of the new admin as input.
- A request will be sent to the polygon marketplace to add new user's details.
- A success message will be shown.
- An error message will be shown upon some error while inserting the new user in the database

3.2.2.4 Update Existing User

- This functional feature also deals with the administrator. The admin can update an existing user in the database by using the interface of the Art Gallery.
- The admin would edit the username, user address and other details about the user using different UI.
- A request will be sent to the polygon marketplace to update new user's details.
- A success message will be shown to the user and database will be updated.
- An error message will be shown upon some error while updating the admin in the database.

3.2.2.5 New User Registration

- This functional feature deals with the user who is using the application's front-end. He would have to register him before any detail.
- The user will enter his complete name, address, his phone number, email address to register him.
- A request will be sent to the polygon marketplace to add new user's details.
- Database will be updated with a new user and a success message will.

- An error message will be shown upon some error while searching a specific item the database.

3.2.2.7 User Login

- The user may login to artist detail by using the credentials assigned to him upon registration.
- The user will enter his email address and password to login.
- The application will communicate to the blockchain database to check if the user has registered already.
- User will be logged in if the entered credentials meet the one in the database and a success message will be shown otherwise, he will be asked to enter correct data.
- An error message will be shown upon some error while logging into the database

3.3 NON-FUNCTIONAL REQUIREMENTS

3.3.1 Performance

- Average load time of the app should be less than 5 seconds.
- Average response time of the system should be less than 10 seconds.
- Our system should easily be accessible on the system having minimum internet speed of 1MB/s

3.3.2 Reliability: There will be a maximum of 1 bug/KLOC.

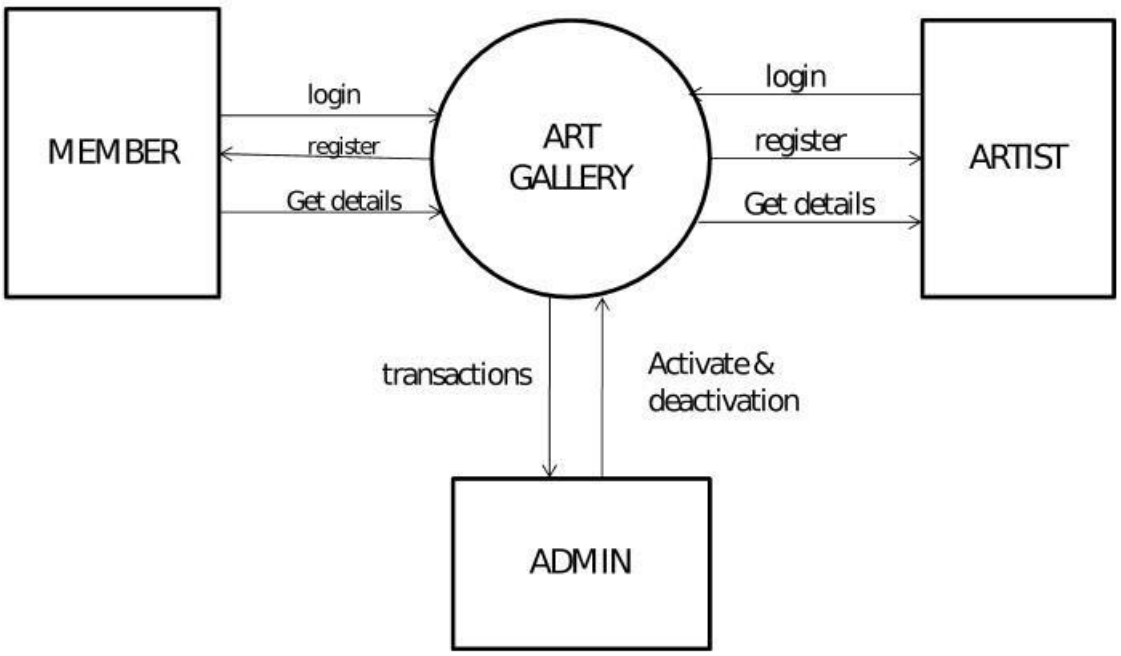
3.3.3 Supportability: All code will be written as specified by the Hungarian Naming Convention.

3.3.4 Availability: Seminar Generator will be available to client 7 days a week and 24 hours a day. In case of any system crash a backup will be available to make the system live.

3.3.5 Security: Users ‘information will be secure, and he will be able to access only their own personal information

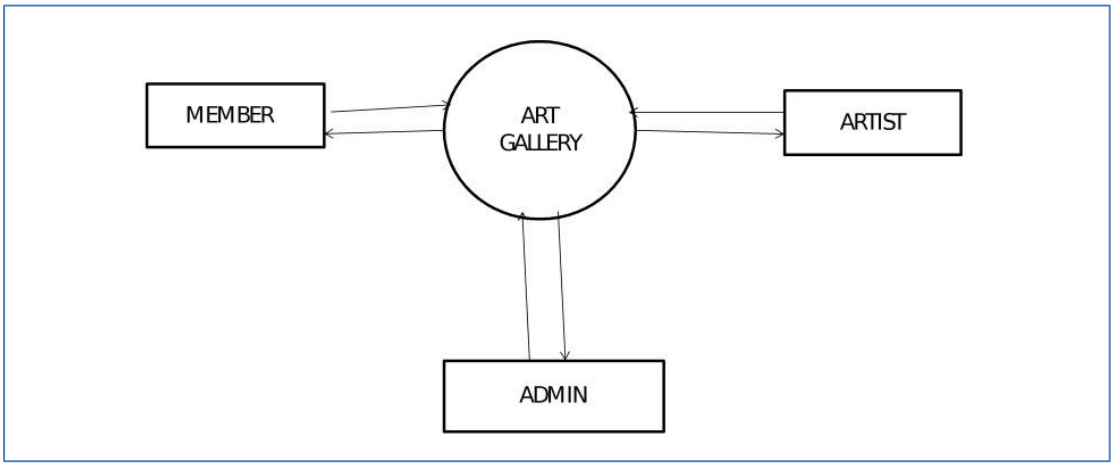
4.SYSTEM ANALYSIS AND DESIGN

4.1 CONTEXT DIAGRAMS:

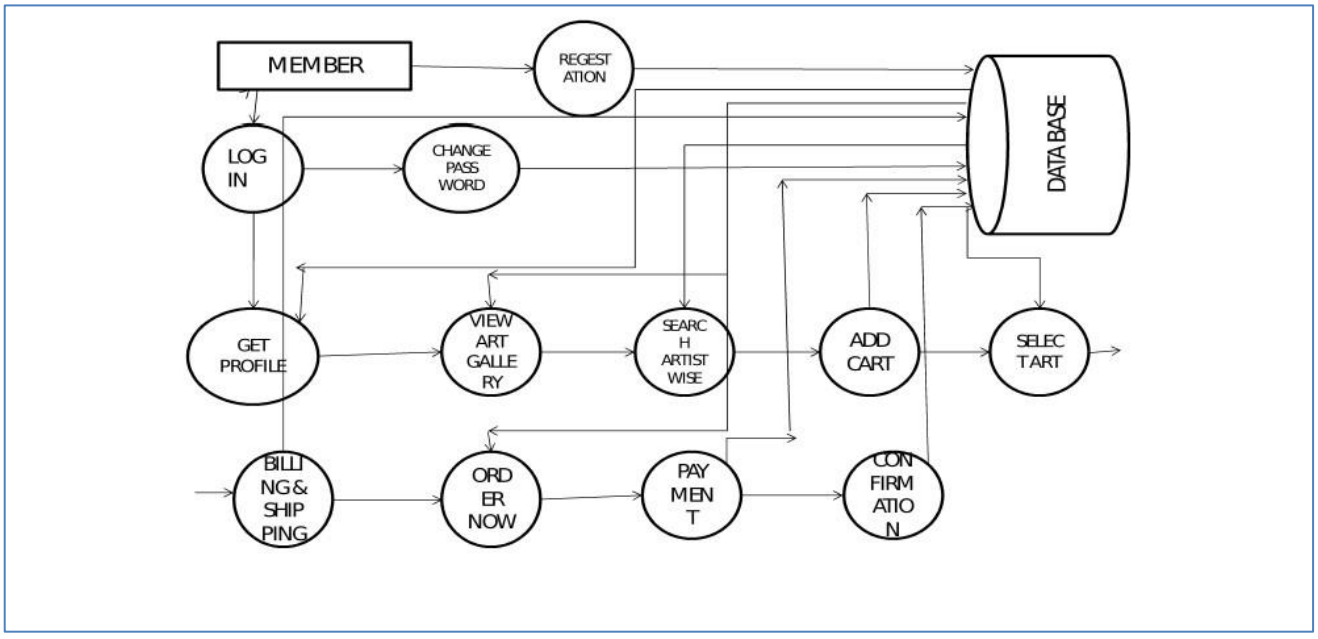


Data flow diagram is used to decrease the movement of data through a system store of data in the system. Data flow diagrams are the central tool based on which components are developed. The transformation of data from input to output, through process may be describe logically and independently of physically components associated with the system. They are called logical data flow diagrams. In contrast physical data flow diagrams show the actual implementation and movement of data between people, Department, and workstation. The data flow diagram shows functional composition of the system. The first level of conceptual level in context diagram is flowed by the description of input and output for each of entities the next level of DFD is level 0, which shows the main functions in the system. Level 0 is followed by the description of the main functions. The main function is further broken into functions and sub function.

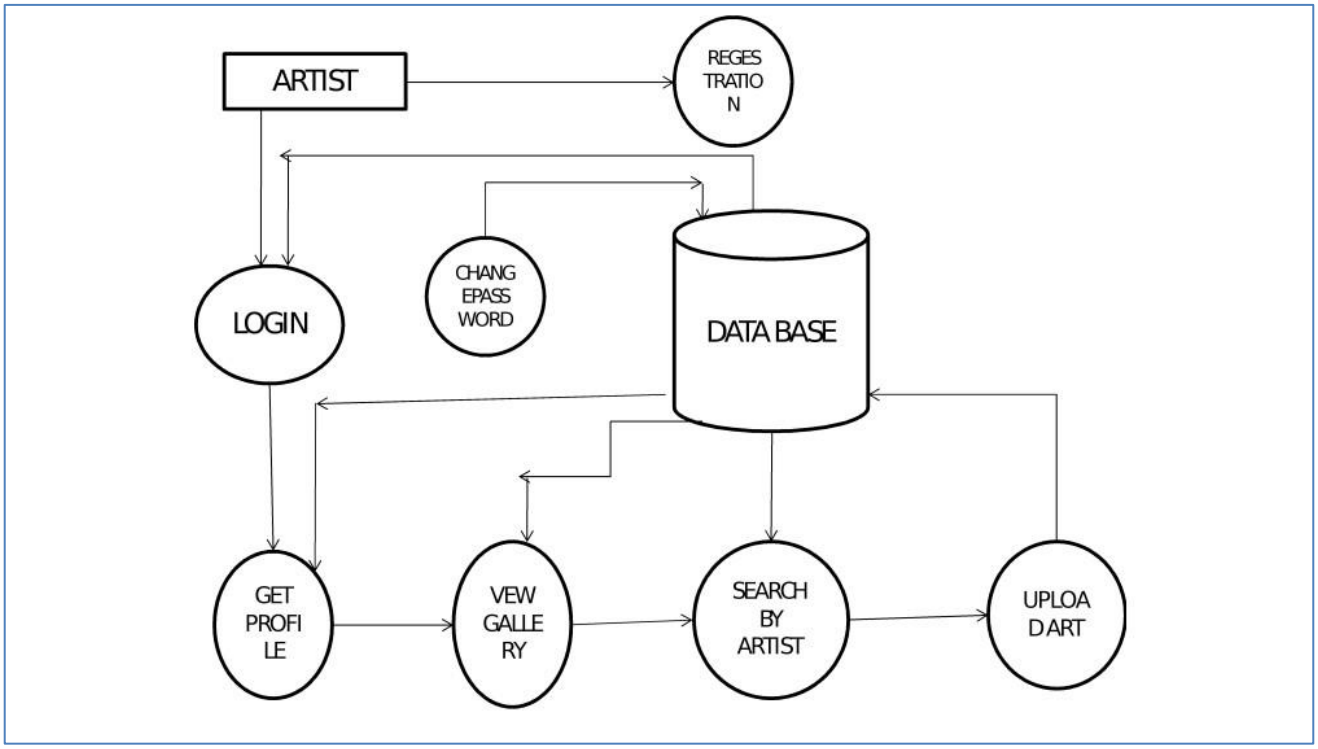
DFD Level 0:



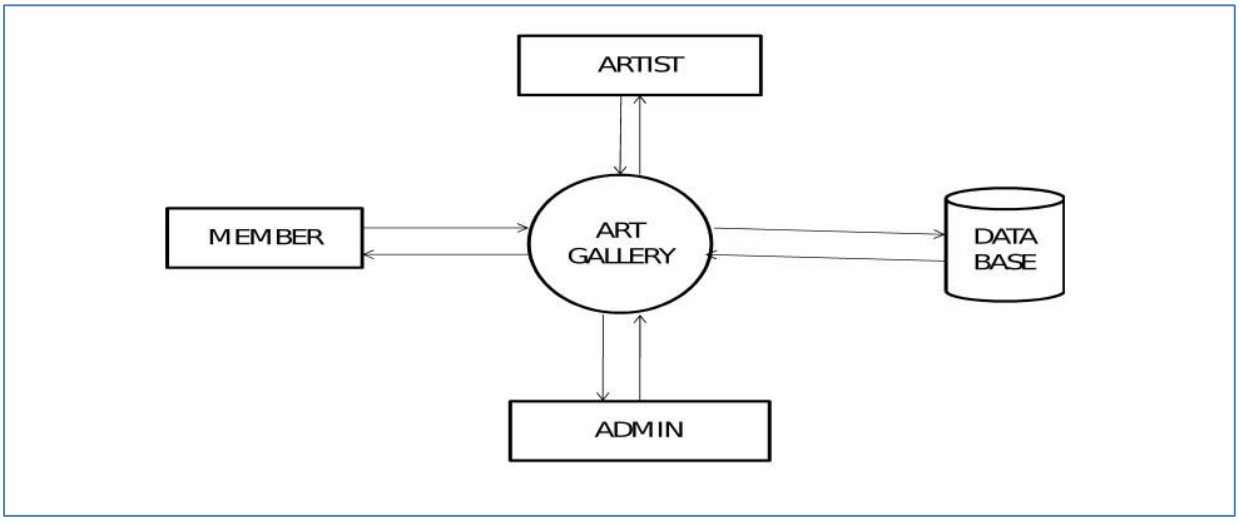
DFD Level 1 member:



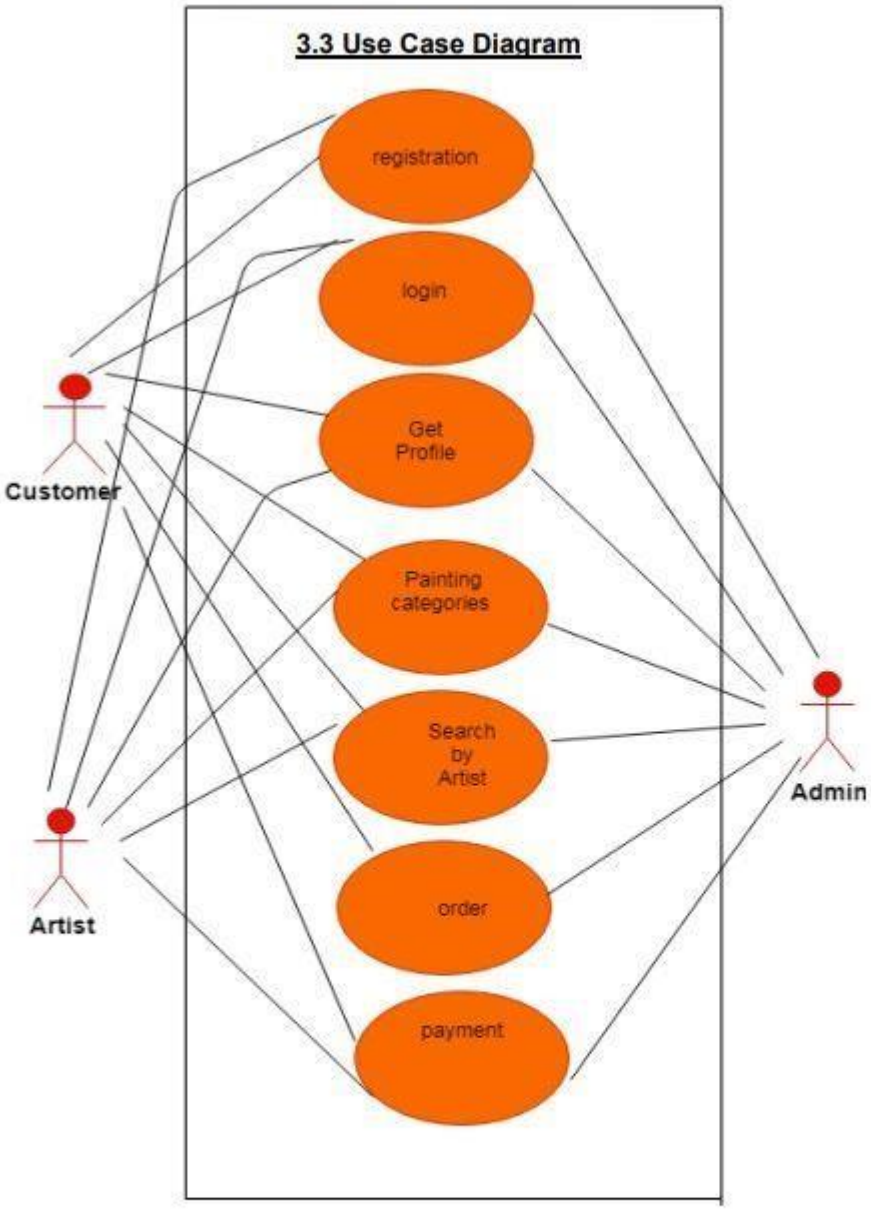
DFD Level 1 Artist:



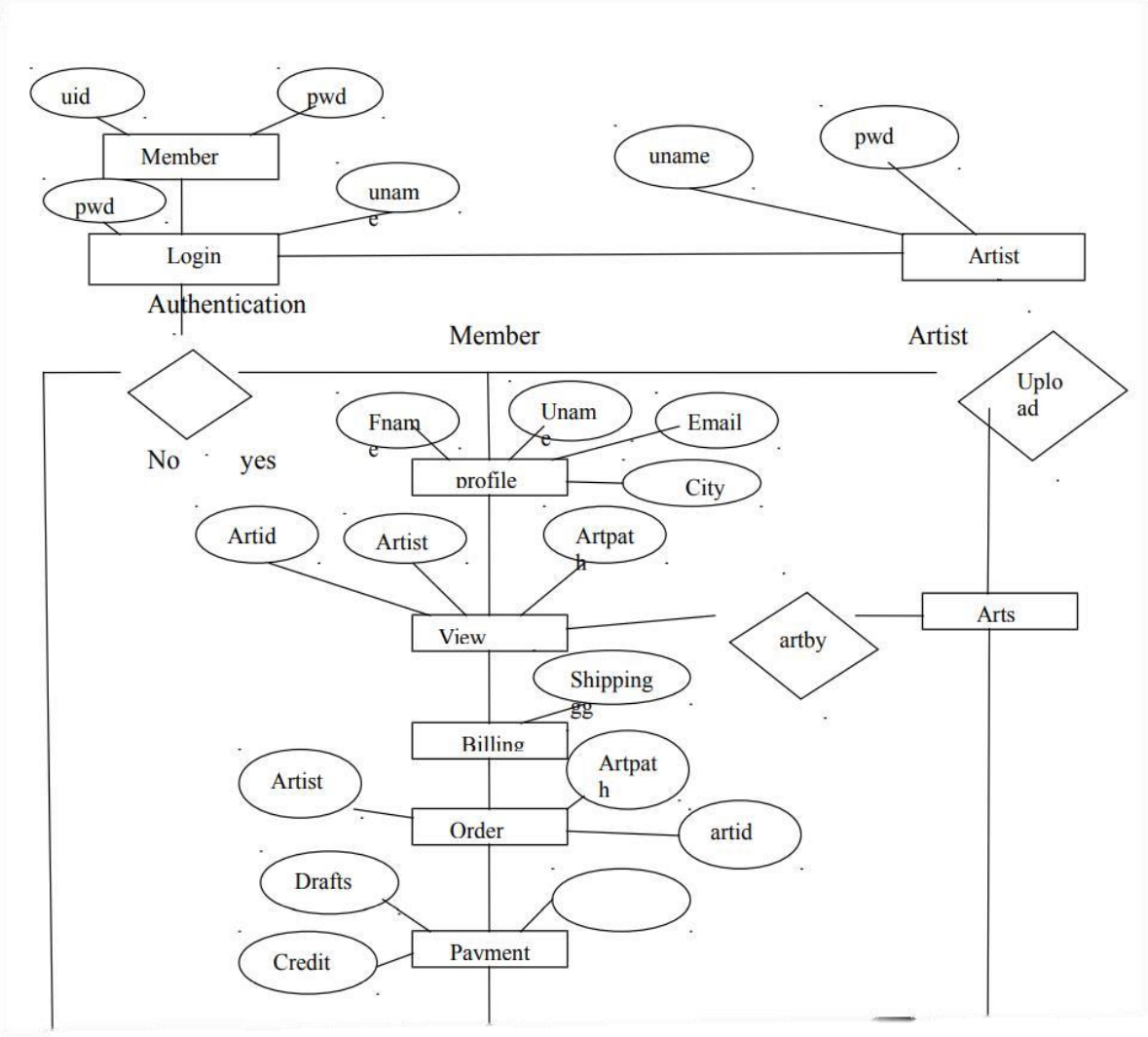
DFD Level 1 Admin:



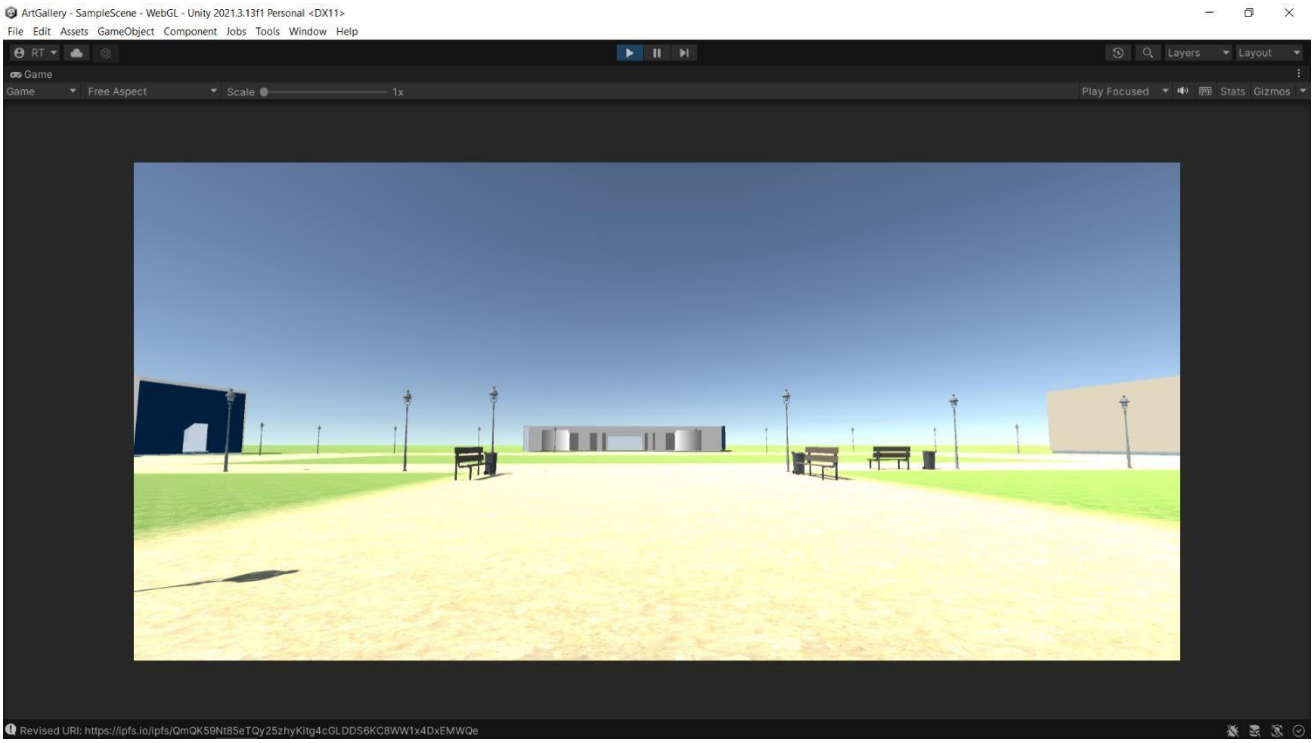
4.2 USE CASE DIAGRAM

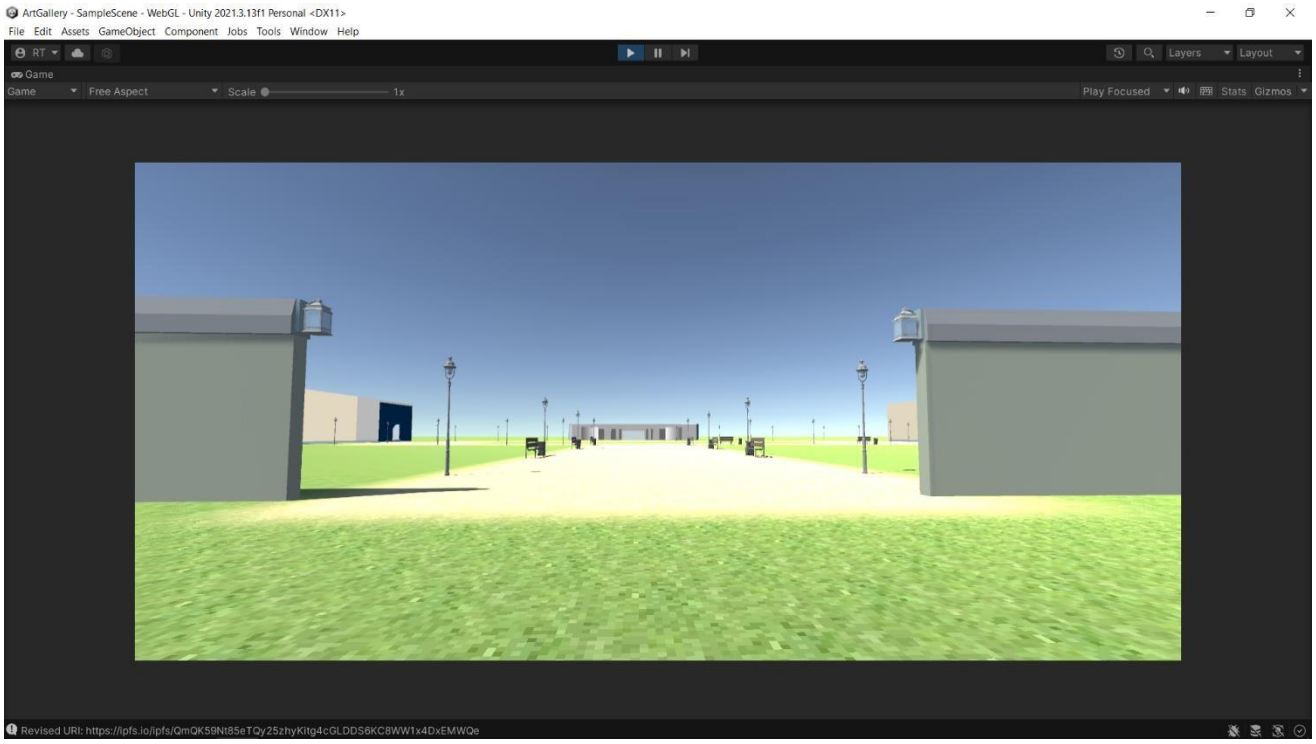


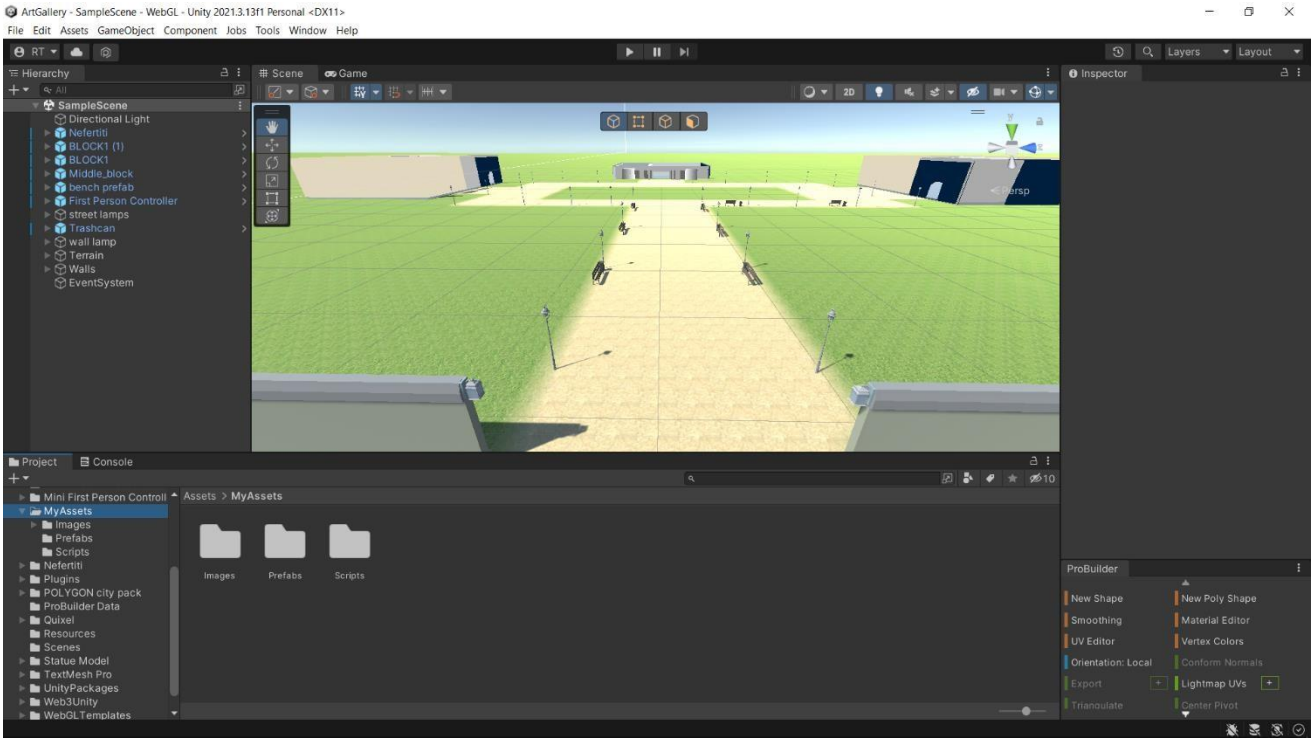
4.3 DATABASE DESIGN (ER Diagram)

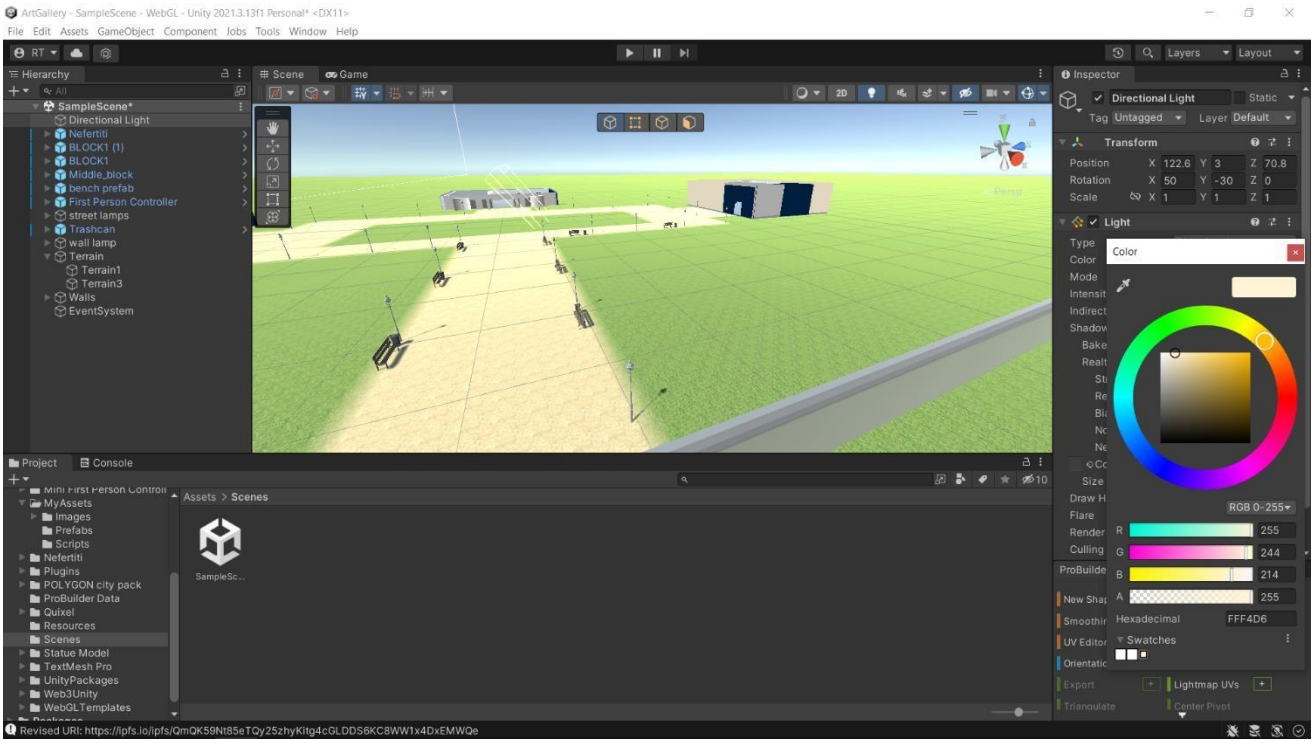


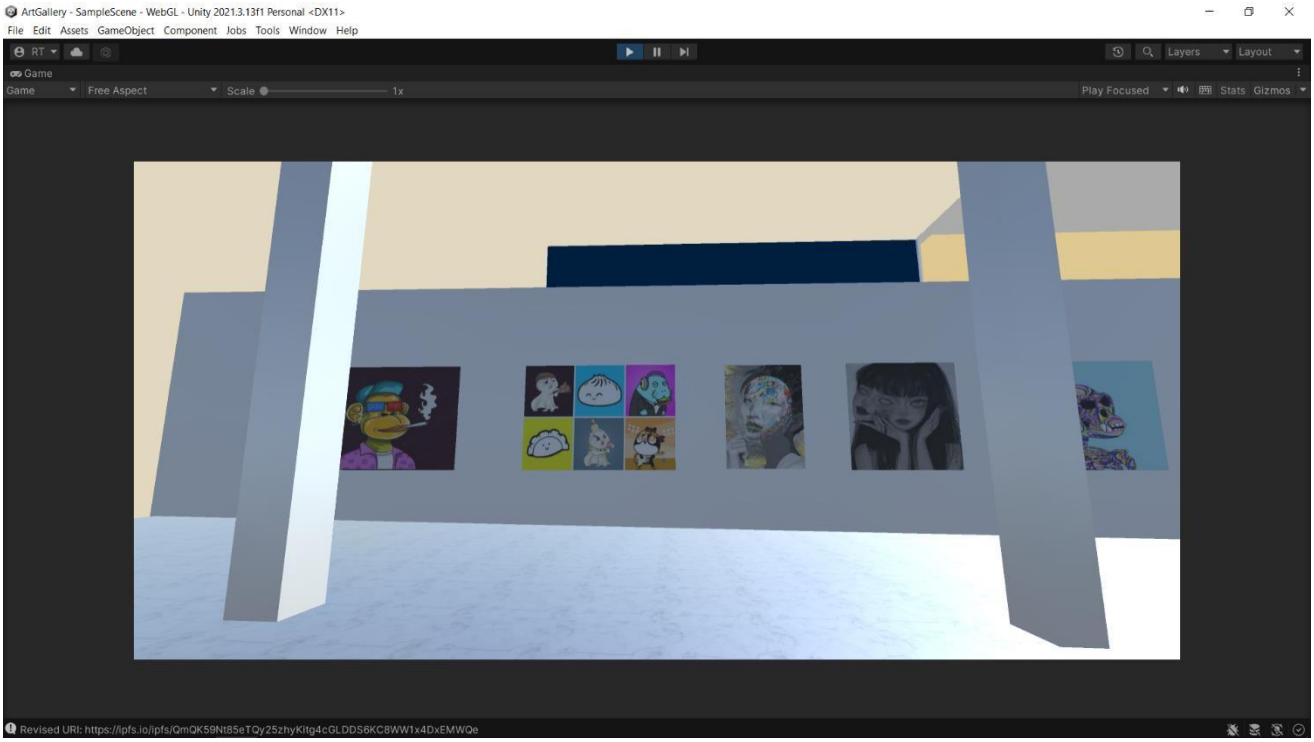
5.IMPLEMENTATION

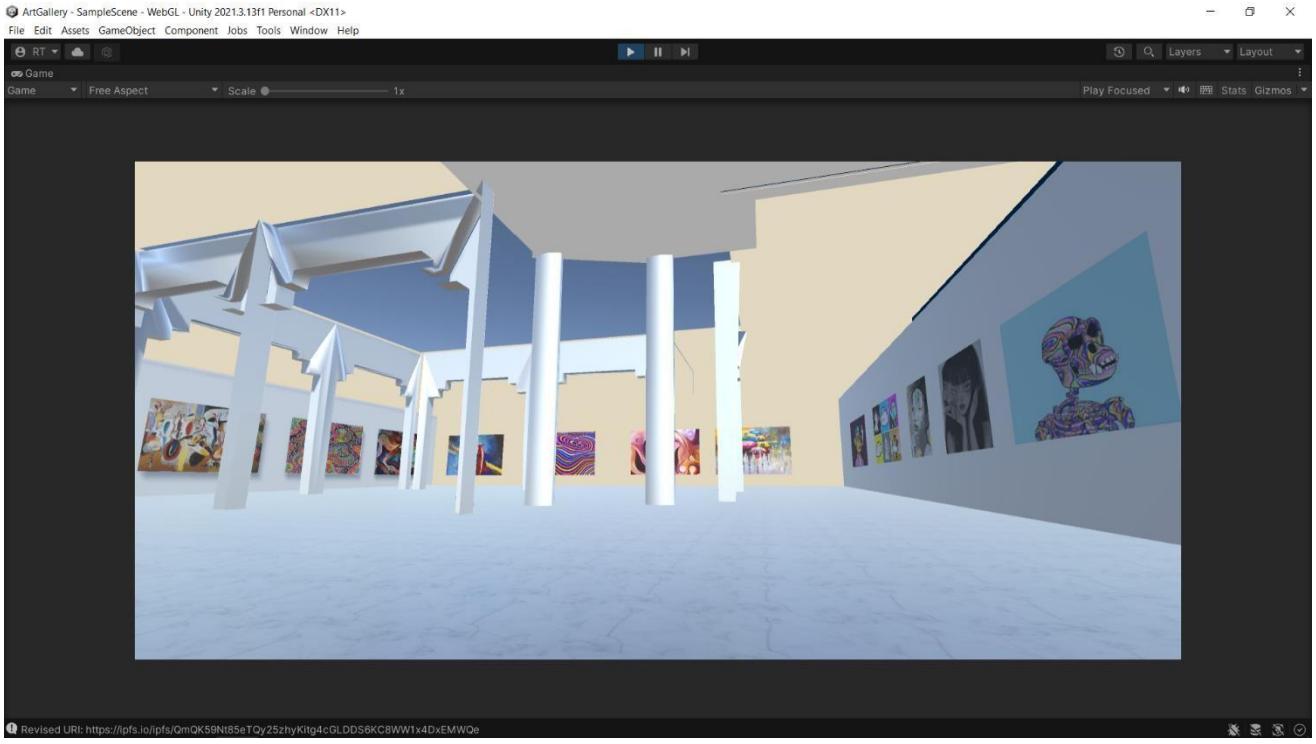


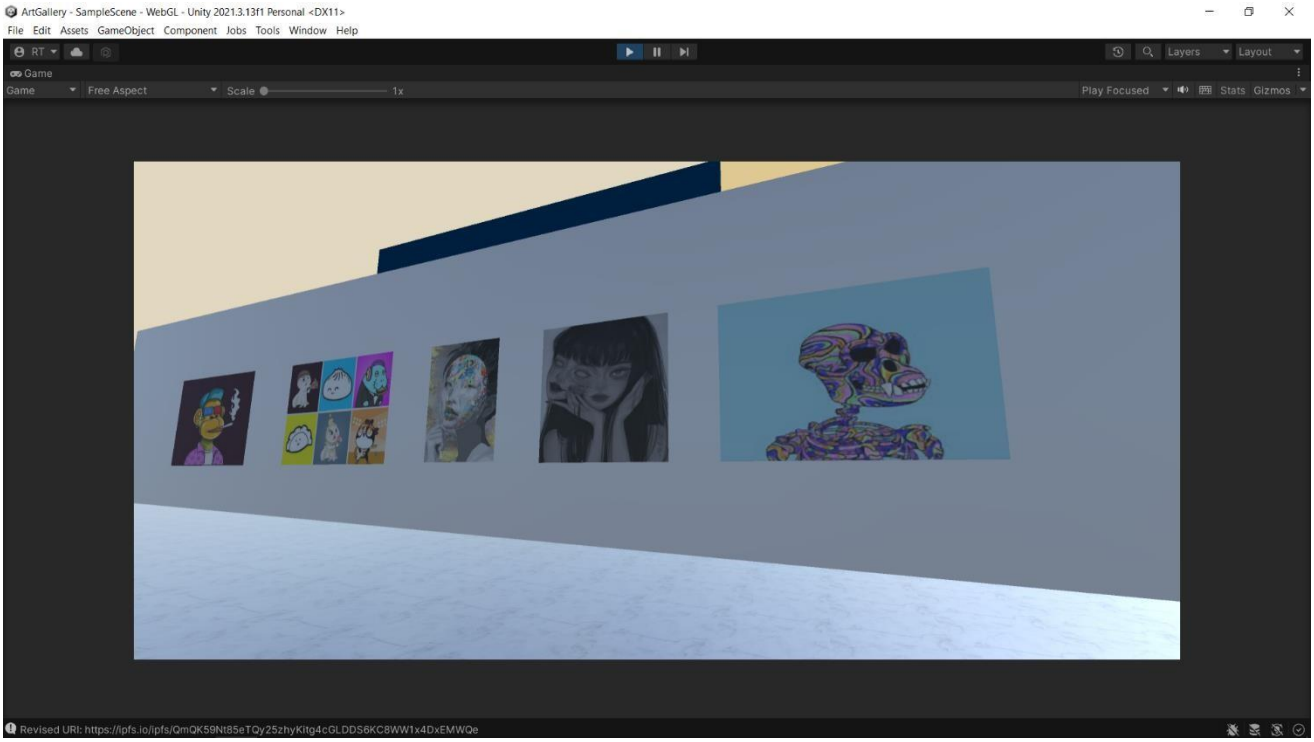


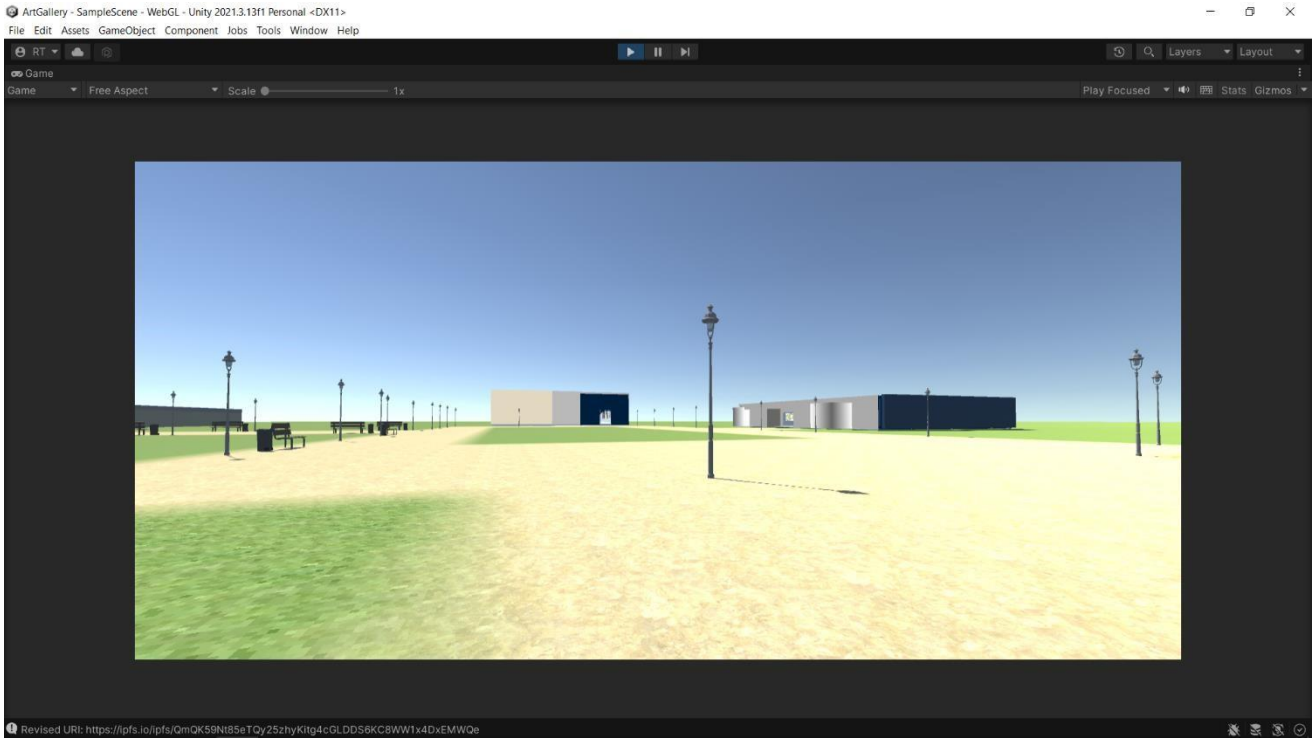

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
Hierarchy
SampleScene
Directional Light
Nefertiti
BLOCK1 (1)
BLOCK1
Middle_block
bench prefab
First Person Controller
street lamps
Trashcan
wall lamp
Terrain
Walls
EventSystem
Scene
Game
Persp
Inspector
Project
Console
Assets > MyAssets
Images
Prefabs
Scripts
ProBuilder
New Shape
New Poly Shape
Smoothing
Material Editor
UV Editor
Vertex Colors
Orientation: Local
Conform Normals
Export
Lightmap UVs
Triangulate
Center Pivot

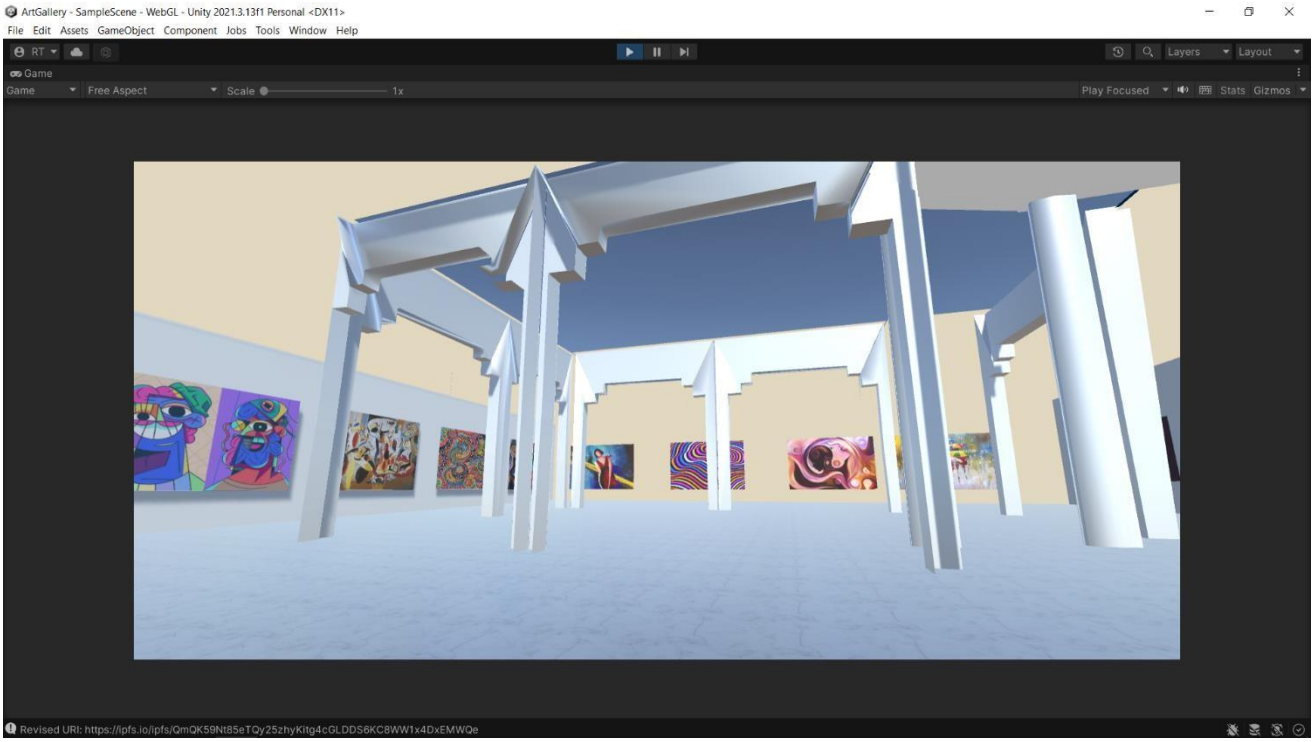

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal* <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
Hierarchy
SampleScene*
Directional Light
Nefertiti
BLOCK1 (1)
BLOCK1
Middle_block
bench prefab
First Person Controller
street lamps
Trashcan
wall lamp
Terrain
Terrain1
Terrain3
Walls
EventSystem
Scene
Game
Persp
Inspector
Directional Light
Static
Tag Untagged
Layer Default
Transform
Position X 122.6 Y 3 Z 70.8
Rotation X 50 Y -30 Z 0
Scale X 1 Y 1 Z 1
Light
Type
Color
Color
Mode
Intensit
Indirect
Shadow
Bake
Realt
Size
Draw H
Flare
Render
Culling
RGB 0-255
R 255
G 244
B 214
A 255
Hexadecimal FFF4D6
Swatches
Project
Console
Assets > Scenes
SampleSc...
Export
Lightmap UVs
Triangulate
Center Pivot
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe

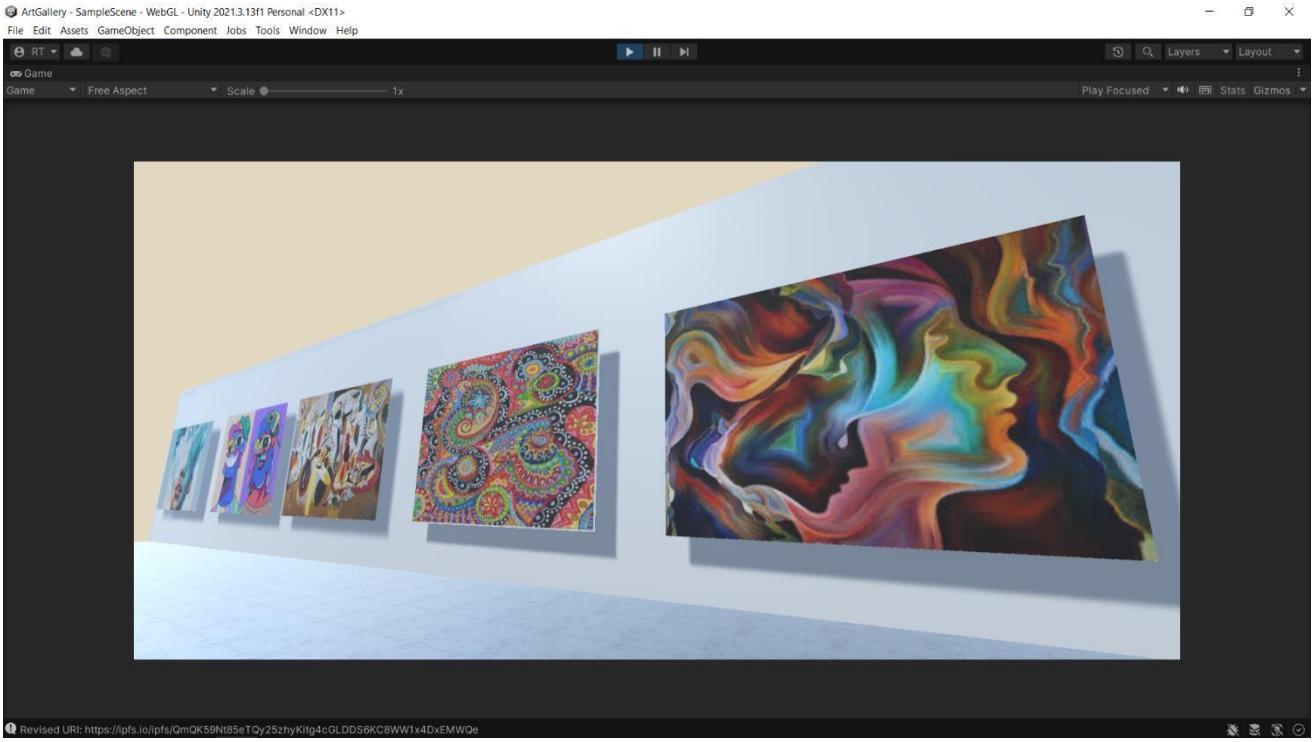

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
RT
Layers
Layout
Game
Game
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe

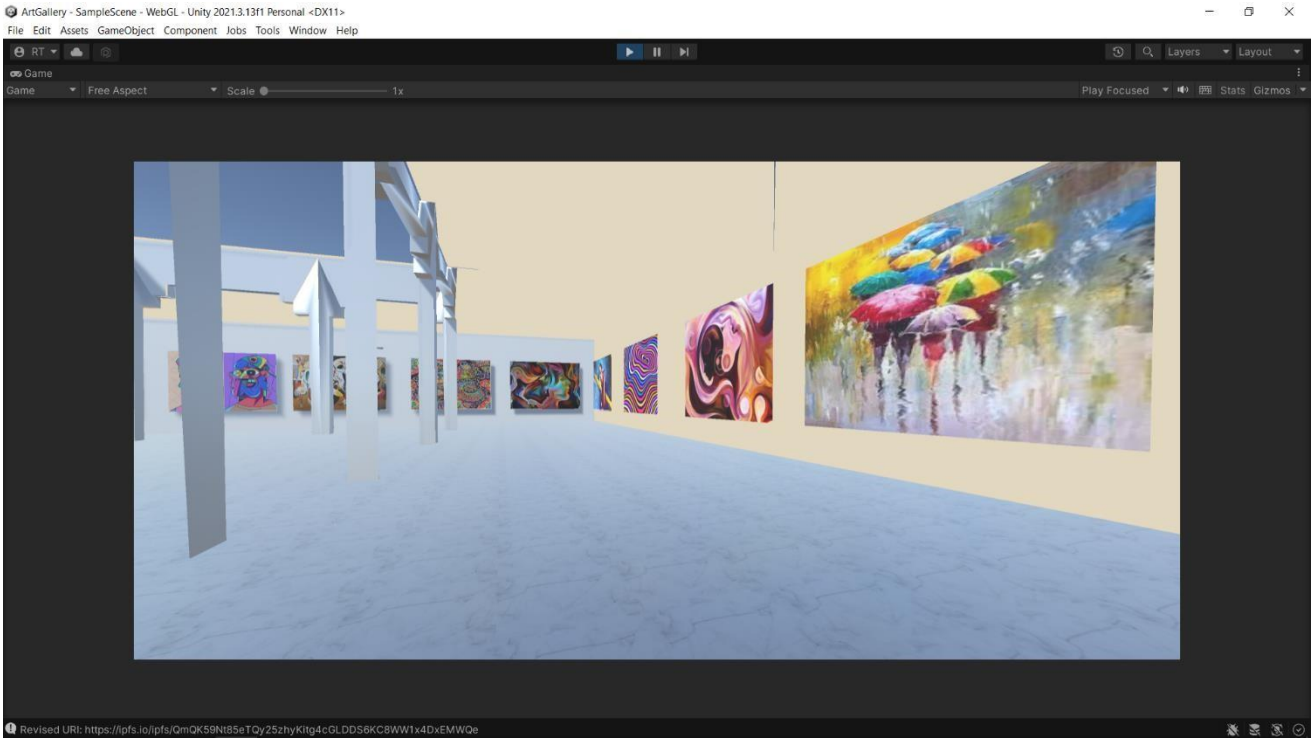

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
RT
Layers
Layout
Game
Game
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe

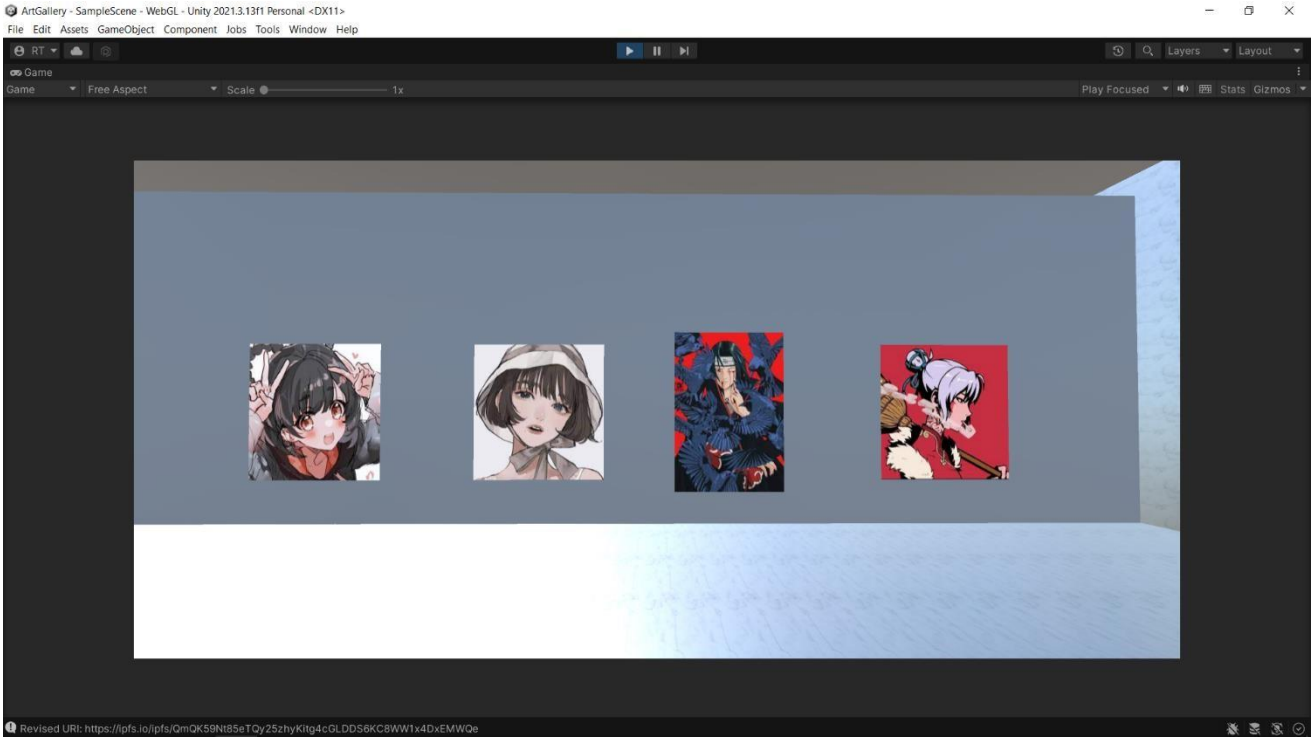

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
RT
Layers
Layout
Game
Game
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe

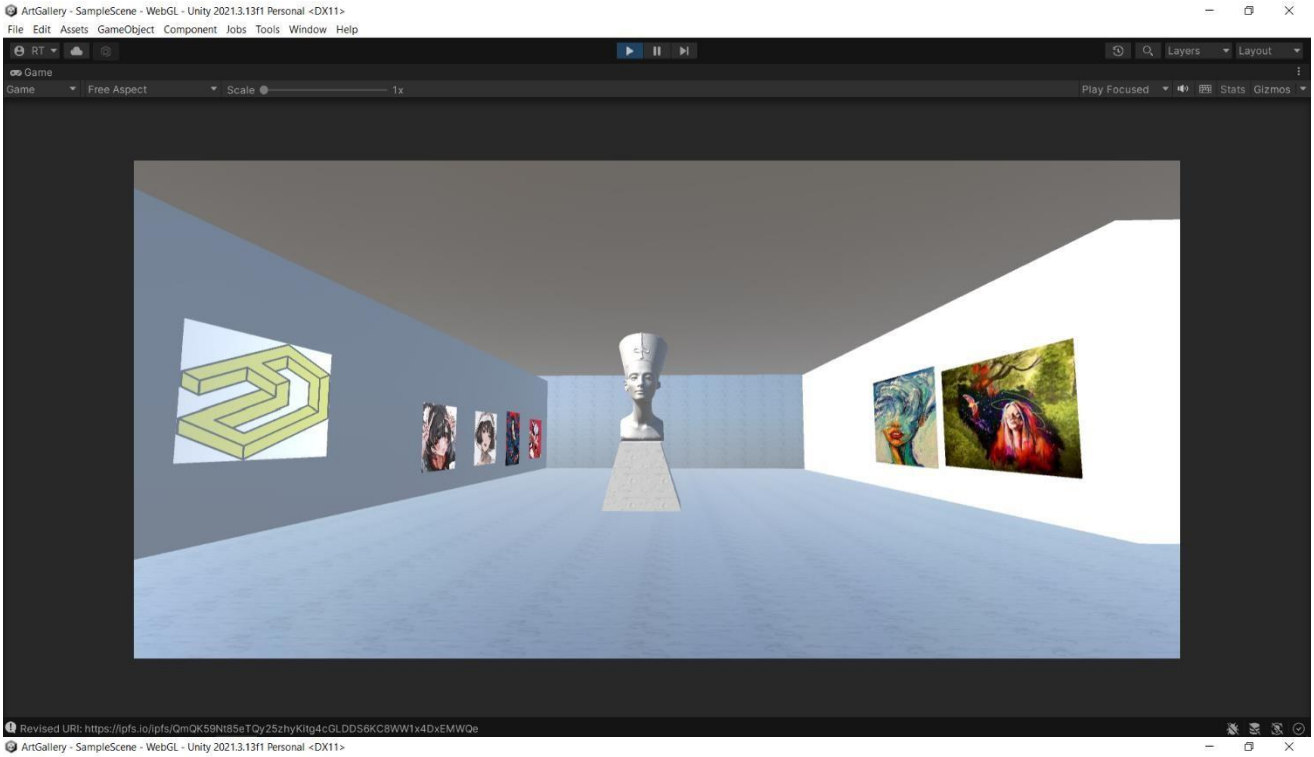

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
RT
Layers
Layout
Game
Game
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe

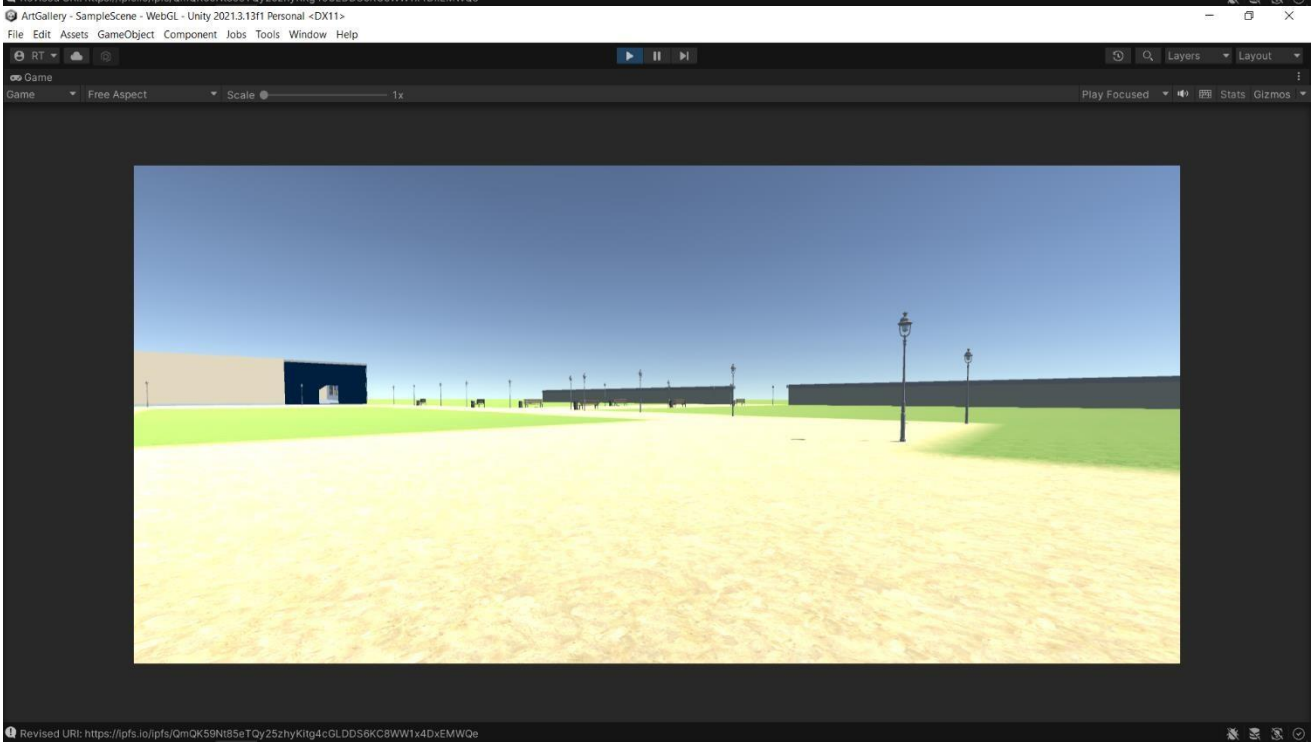

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
RT
Layers
Layout
Game
Game
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe

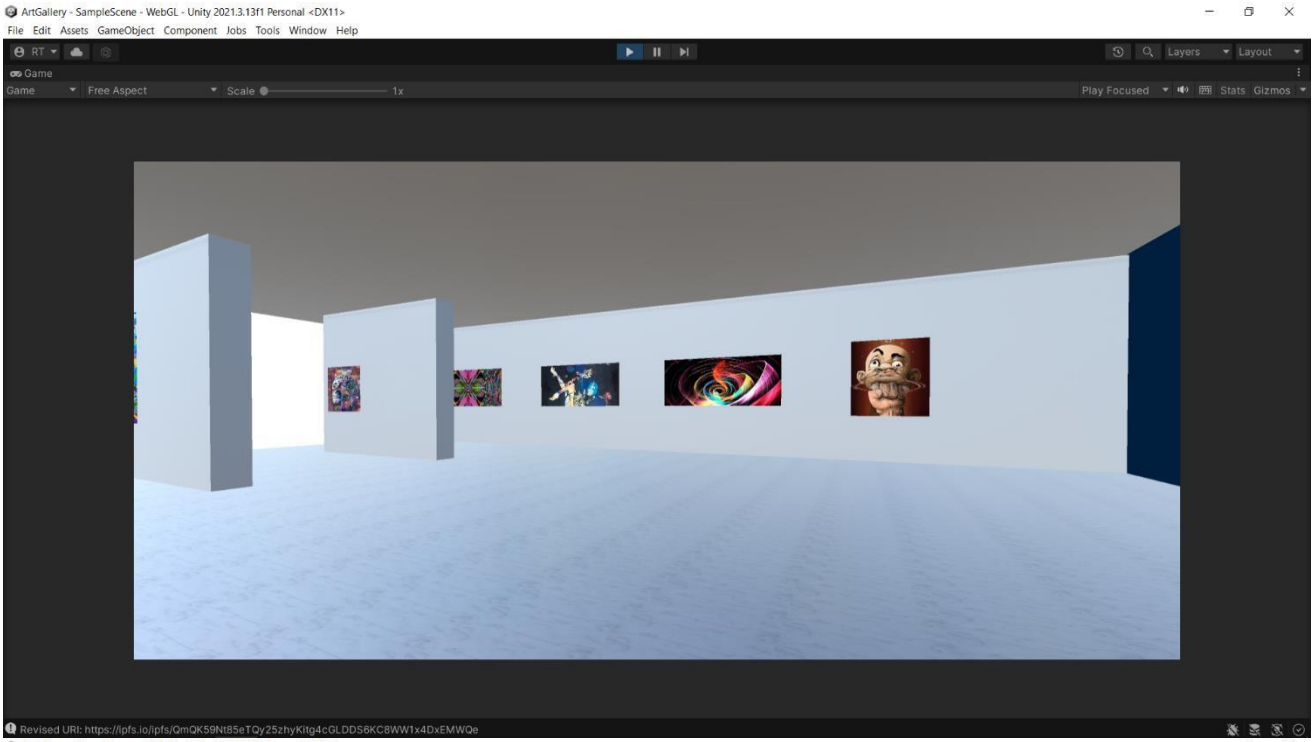

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
RT
Layers
Layout
Game
Game
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe

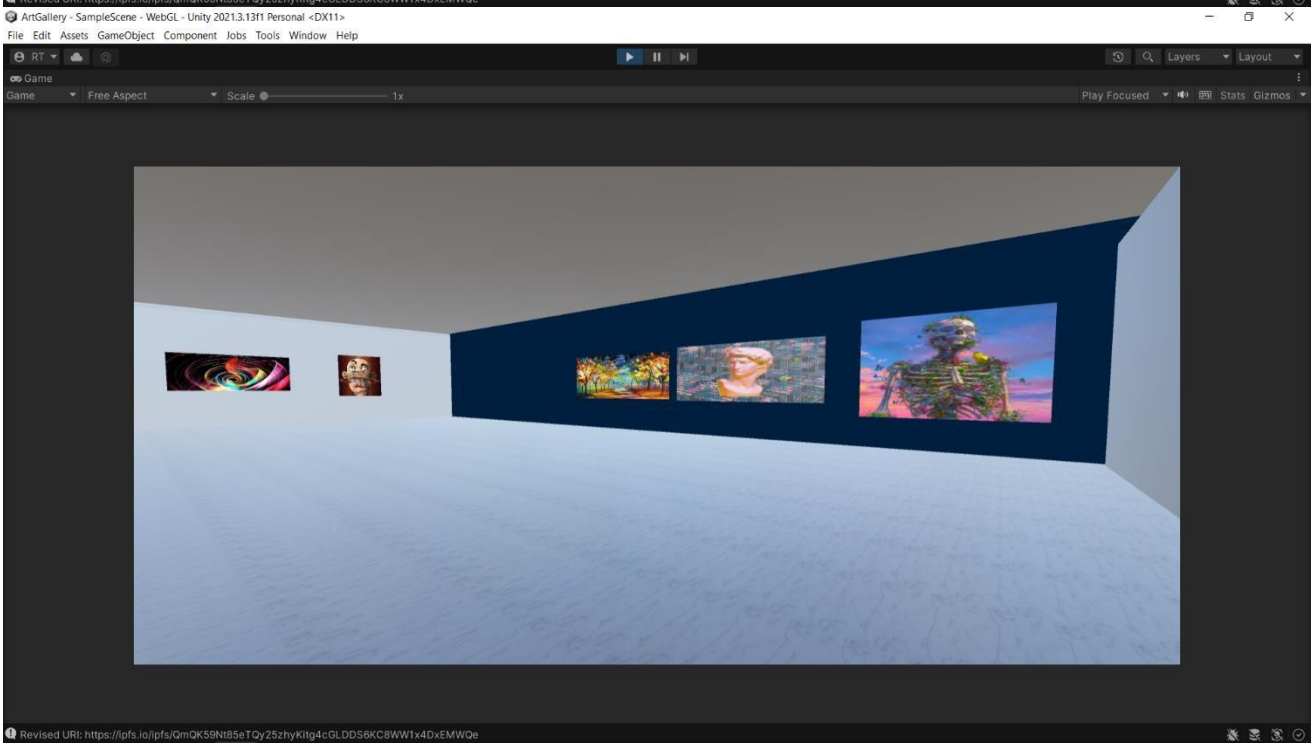

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
Game
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Layers
Layout
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe

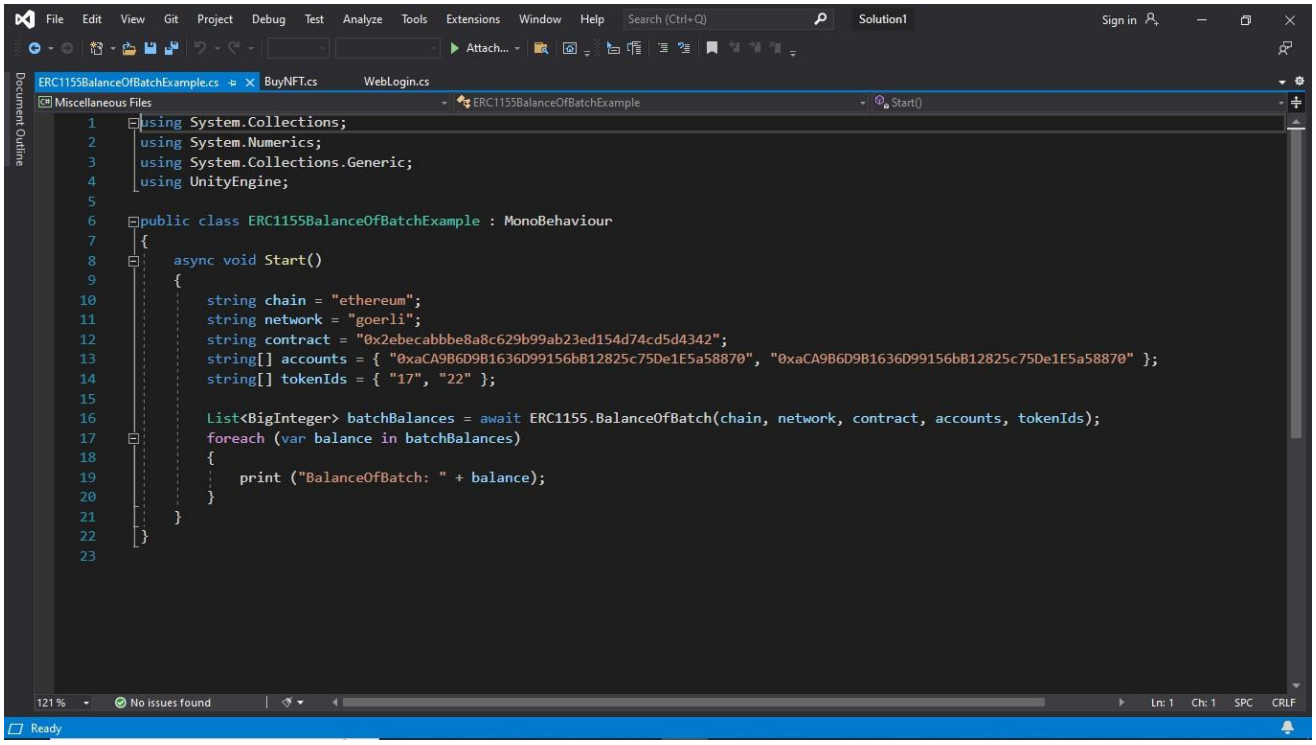

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
Game
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Layers
Layout
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe

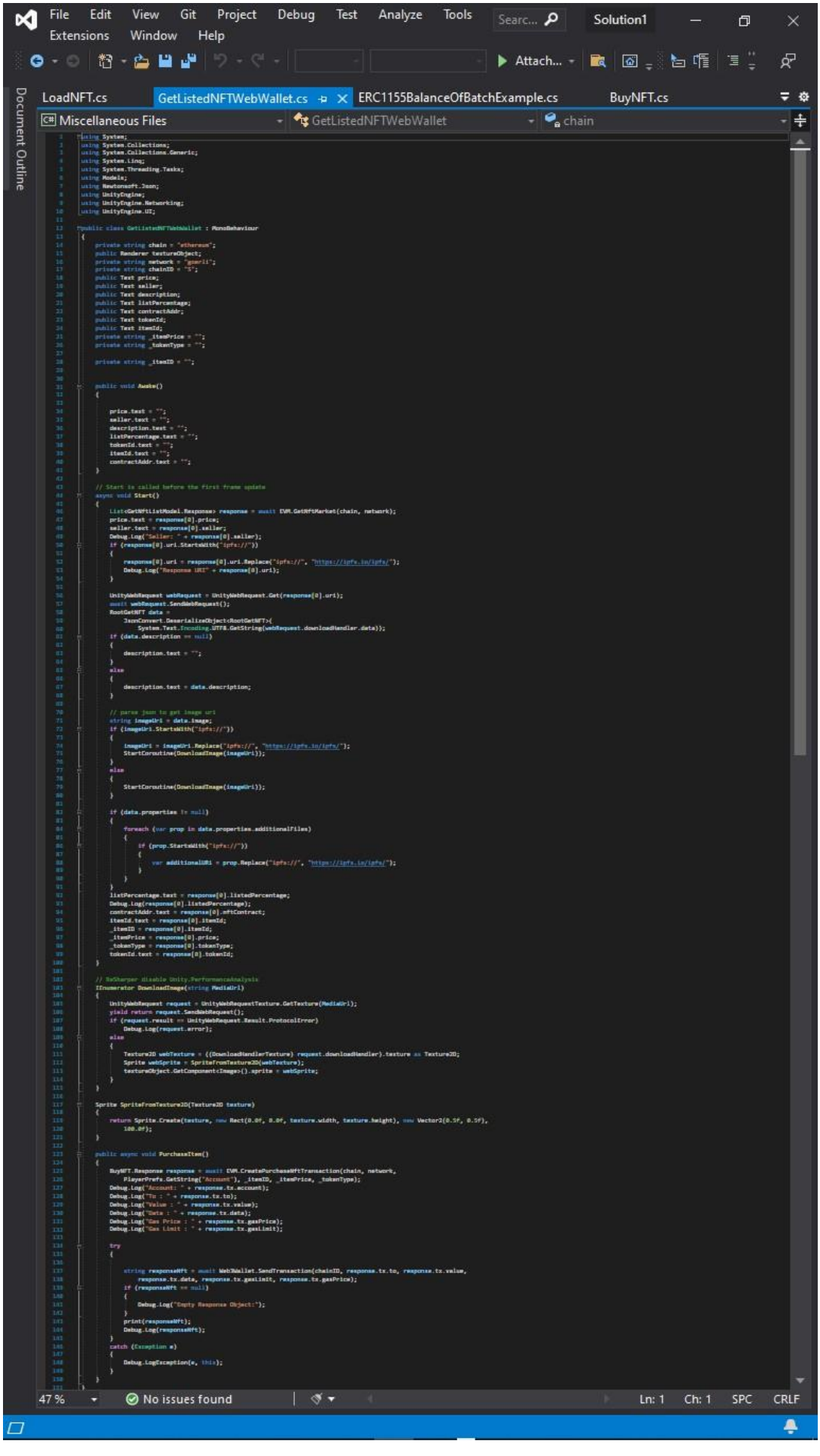

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
Game
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Layers
Layout
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe

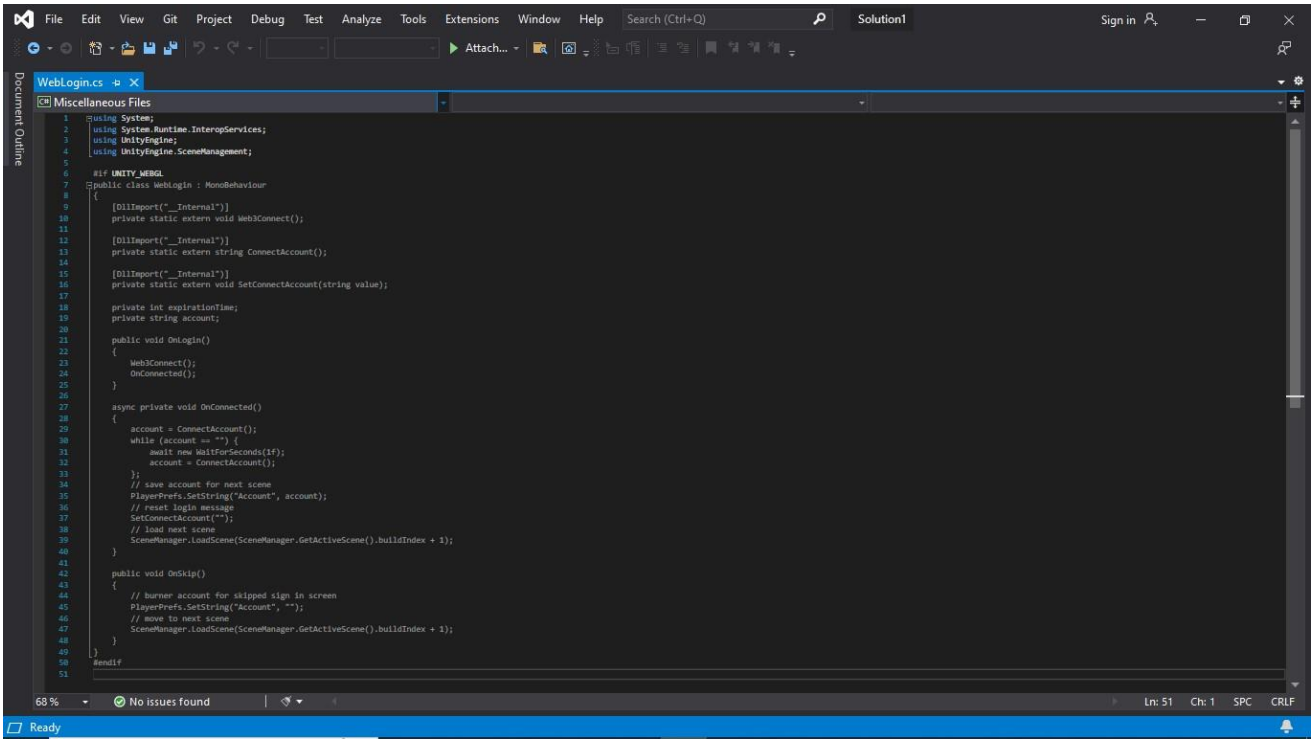

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
Game
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Layers
Layout
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe

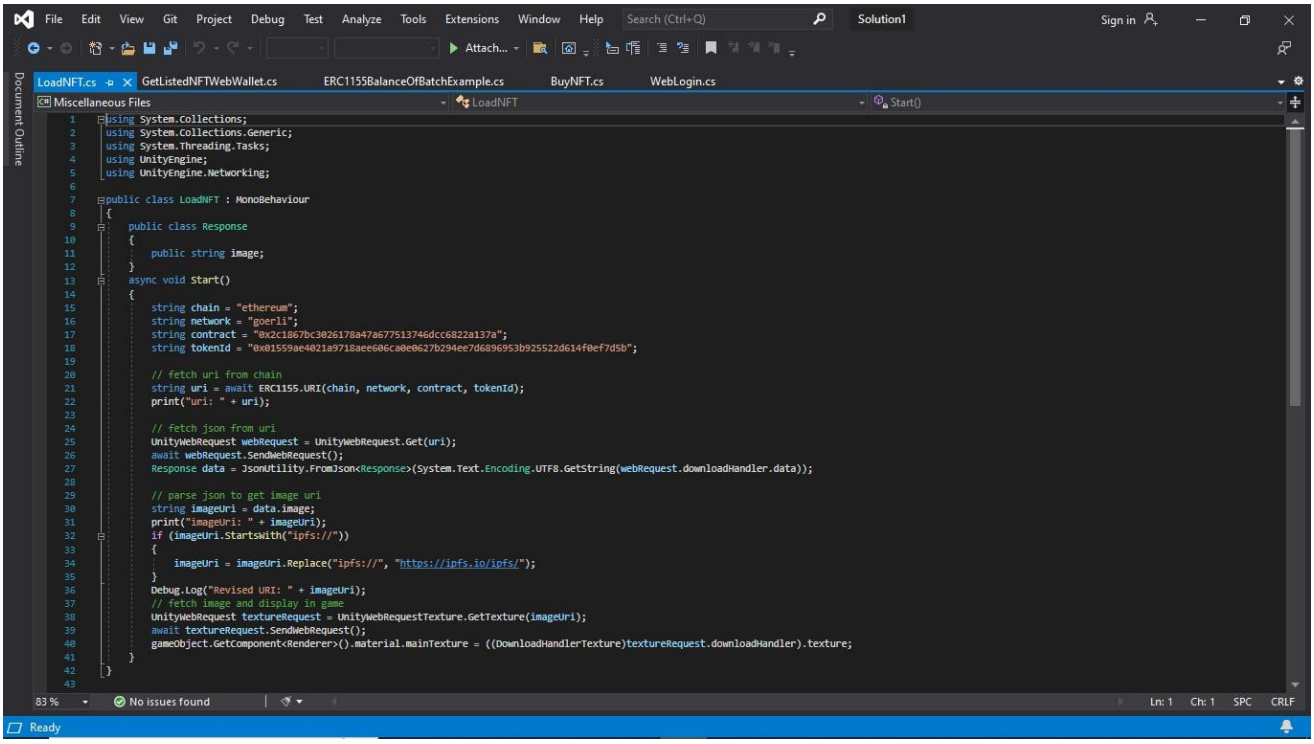

ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
RT
Layers
Layout
Game
Game
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe


ArtGallery - SampleScene - WebGL - Unity 2021.3.13f1 Personal <DX11>
File Edit Assets GameObject Component Jobs Tools Window Help
RT
Layers
Layout
Game
Game
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Revised URI: https://ipfs.io/ipfs/QmQK59Nt85eTQy25zhyKitg4cGLDDS6KC8WW1x4DxEMWQe


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Solution1
Sign in
Attach...
ERC1155BalanceOfBatchExample.cs
BuyNFT.cs
WebLogin.cs
Miscellaneous Files
ERC1155BalanceOfBatchExample
Start()
Document Outline
using System.Collections;
using System.Numerics;
using System.Collections.Generic;
using UnityEngine;
public class ERC1155BalanceOfBatchExample : MonoBehaviour
{
async void Start()
{
string chain = "ethereum";
string network = "goerli";
string contract = "0x2ebecabbbe8a8c629b99ab23ed154d74cd5d4342";
string[] accounts = { "0xaCA9B6D9B1636D99156bB12825c75De1E5a58870", "0xaCA9B6D9B1636D99156bB12825c75De1E5a58870" };
string[] tokenIds = { "17", "22" };
List<BigInteger> batchBalances = await ERC1155.BalanceOfBatch(chain, network, contract, accounts, tokenIds);
foreach (var balance in batchBalances)
{
print ("BalanceOfBatch: " + balance);
}
}
}
121 %
No issues found
Ln: 1
Ch: 1
SPC
CRLF
Ready


File Edit View Git Project Debug Test Analyze Tools
Extensions Window Help
Search...
Solution1
Attach...
LoadNFT.cs
GetListedNFTWebWallet.cs
ERC1155BalanceOfBatchExample.cs
BuyNFT.cs
Miscellaneous Files
GetListedNFTWebWallet
chain
Document Outline
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Models;
using Newtonsoft.Json;
using UnityEngine;
using UnityEngine.Networking;
using UnityEngine.UI;
public class GetListedNFTWebWallet : MonoBehaviour
{
private string chain = "ethereum";
public Renderer textureObject;
private string network = "goerli";
private string chainID = "5";
public Text price;
public Text seller;
public Text description;
public Text listPercentage;
public Text contractAddr;
public Text tokenId;
public Text itemId;
private string _itemPrice = "";
private string _tokenType = "";
private string _itemID = "";
public void Awake()
{
price.text = "";
seller.text = "";
description.text = "";
listPercentage.text = "";
tokenId.text = "";
itemId.text = "";
contractAddr.text = "";
}
// Start is called before the first frame update
async void Start()
{
List<GetNftListModel.Response> response = await EVM.GetNftMarket(chain, network);
price.text = response[0].price;
seller.text = response[0].seller;
Debug.Log("Seller: " + response[0].seller);
if (response[0].uri.StartsWith("ipfs://"))
{
response[0].uri = response[0].uri.Replace("ipfs://", "https://ipfs.io/ipfs/");
Debug.Log("Response URI" + response[0].uri);
}
UnityWebRequest webRequest = UnityWebRequest.Get(response[0].uri);
await webRequest.SendWebRequest();
RootGetNFT data =
JsonConvert.DeserializeObject<RootGetNFT>(
System.Text.Encoding.UTF8.GetString(webRequest.downloadHandler.data));
if (data.description == null)
{
description.text = "";
}
else
{
description.text = data.description;
}
// parse json to get image uri
string imageUri = data.image;
if (imageUri.StartsWith("ipfs://"))
{
imageUri = imageUri.Replace("ipfs://", "https://ipfs.io/ipfs/");
StartCoroutine(DownloadImage(imageUri));
}
else
{
StartCoroutine(DownloadImage(imageUri));
}
if (data.properties != null)
{
foreach (var prop in data.properties.additionalFiles)
{
if (prop.StartsWith("ipfs://"))
{
var additionalUri = prop.Replace("ipfs://", "https://ipfs.io/ipfs/");
}
}
}
listPercentage.text = response[0].listedPercentage;
Debug.Log(response[0].listedPercentage);
contractAddr.text = response[0].nftContract;
itemId.text = response[0].itemId;
_itemID = response[0].itemId;
_itemPrice = response[0].price;
_tokenType = response[0].tokenType;
tokenId.text = response[0].tokenId;
}
// ReSharper disable Unity.PerformanceAnalysis
IEnumerator DownloadImage(string MediaUri)
{
UnityWebRequest request = UnityWebRequestTexture.GetTexture(MediaUri);
yield return request.SendWebRequest();
if (request.result == UnityWebRequest.Result.ProtocolError)
Debug.Log(request.error);
else
{
Texture2D webTexture = ((DownloadHandlerTexture) request.downloadHandler).texture as Texture2D;
Sprite webSprite = SpriteFromTexture2D(webTexture);
textureObject.GetComponent<Image>().sprite = webSprite;
}
}
Sprite SpriteFromTexture2D(Texture2D texture)
{
return Sprite.Create(texture, new Rect(0.0f, 0.0f, texture.width, texture.height), new Vector2(0.5f, 0.5f),
100.0f);
}
public async void PurchaseItem()
{
BuyNFT.Response response = await EVM.CreatePurchaseNftTransaction(chain, network,
PlayerPrefs.GetString("Account"), _itemID, _itemPrice, _tokenType);
Debug.Log("Account: " + response.tx.account);
Debug.Log("To : " + response.tx.to);
Debug.Log("Value : " + response.tx.value);
Debug.Log("Data : " + response.tx.data);
Debug.Log("Gas Price : " + response.tx.gasPrice);
Debug.Log("Gas Limit : " + response.tx.gasLimit);
try
{
string responseNft = await WebWallet.SendTransaction(chainID, response.tx.to, response.tx.value,
response.tx.data, response.tx.gasLimit, response.tx.gasPrice);
if (responseNft == null)
{
Debug.Log("Empty Response Object:");
}
print(responseNft);
Debug.Log(responseNft);
}
catch (Exception e)
{
Debug.LogException(e, this);
}
}
}
47 %
No issues found
Ln: 1
Ch: 1
SPC
CRLF


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Solution1
Sign in
Attach...
Document Outline
WebLogin.cs
Miscellaneous Files
using System;
using System.Runtime.InteropServices;
using UnityEngine;
using UnityEngine.SceneManagement;
#if UNITY_WEBGL
public class WebLogin : MonoBehaviour
{
[DllImport("__Internal")]
private static extern void Web3Connect();
[DllImport("__Internal")]
private static extern string ConnectAccount();
[DllImport("__Internal")]
private static extern void SetConnectAccount(string value);
private int expirationTime;
private string account;
public void OnLogin()
{
Web3Connect();
OnConnected();
}
async private void OnConnected()
{
account = ConnectAccount();
while (account == "") {
await new WaitForSeconds(1f);
account = ConnectAccount();
};
// save account for next scene
PlayerPrefs.SetString("Account", account);
// reset login message
SetConnectAccount("");
// load next scene
SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
}
public void OnSkip()
{
// burner account for skipped sign in screen
PlayerPrefs.SetString("Account", "");
// move to next scene
SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
}
}
#endif
68 %
No issues found
Ln: 51 Ch: 1 SPC CRLF
Ready



6.SYSTEM TESTING

6.1 TEST CASE 1

Gallery X wants to sell an animated gif by their artist, whose work normally sells for $50,000$150,000. The gif is an edition of 1,000 with 10 APs (Artist Proofs) and priced at $150.

The editions sell out, netting $150,000 (less the 5% transaction fee) with no logistics, i.e., no packing, shipping, framing or installation required. The smart contract in this case has been programmed so that the artist automatically receives royalties from every trade when these digital editions are resold in the network. The 10 APs are reserved for museums, who are permitted to purchase the artwork at a discount, as stipulated in the smart contract for the edition. The APs can also be circulated for exhibition purposes, what is traditionally known as a “loan.” If an edition is exhibited at, for example, the MoMA in NYC, this data point would be added to that edition’s metadata on the blockchain, making it a more valuable edition due to its exhibition history. The smart contract would also enforce certain exhibition terms, including the duration of the “loan agreement.

6.2 TEST CASE 2

Bringing existing new media collections to the blockchain.

Collectors who have been acquiring digital and time-based media need a system to manage their collection and track works that have been loaned to various exhibitions (whether at museums, art institutions or biennials). This enables collectors to tokenize their artworks, assign IDs and digitally program terms of exhibition rights into smart contracts. If a collector agrees to loan their video

artwork to the Venice Biennial, they could issue a smart contract from their meta mask account that's attached to the ID of that specific digital art asset (i.e. the video). The Venice Biennial would then register for their own "nonprofit/institution" account, create their own ID and sign into the platform in order to review and accept the terms of the contract. The edition would then be transferred to the Venice Biennial's -Wallet for a limited duration of time, as stipulated by the smart contract, with exhibition rights, credit lines and other important registration details (such as installation requirements including preferred video projector resolution) automatically included. When the loan's exhibition duration (that was stipulated in the smart contract) has been fulfilled, the file would be transferred back to the owner's -Wallet and its exhibition provenance would be updated in their Wallet registry. The owner of the artwork, as well as the artist, do not have to worry about the video file being lost, corrupted, duplicated or stolen.

7.CONCLUSION

Customer can also register online, and they can browse art works that are arranged in different categories scientifically. Each Customer can create their own gallery to see his favorite art works without much difficulty and each user has the right to purchase an artwork using the integrated payment gateway and participate in auction by submitting their bids. Qualified bidder should remit the amount using payment gateway and after each valid payment they get the ownership of the artwork they purchased.

We have accomplished this goal by tokenizing the digital and time-based artworks of our partners as NFTs and developing a marketplace & protocol for their sale (and resale). During the summer of 2020, the NFT market saw a 57% increase in trading volume. As the NFT market increases in value and gains recognition beyond the blockchain ecosystem, this project will be the art industry's most trusted platform for transacting tokenized digital art. By expanding into the digital art market, established galleries/artists/institutions can create new revenue streams at price points that are accessible to a wider base of potential buyers. Digital editions of iconic artworks, such as Maurizio Cattelan's "Comedian" (more widely known as "the banana duct taped to the wall"), can offer an entry point into the art world for a younger audience, specifically Millennials and Gen Z. These works can be resold within our network and the artist will receive automatic royalties from every transaction.

As these collectors mature, they will already be invested in their favorite artists, gallery brands and museums, setting a new standard of brand loyalty. Eventually, these young collectors will be ready to purchase more expensive digital artworks at much higher price points, which they can also buy

through metamask, add to their metamask-Wallet and share on social media. This will be the foundational building block for a new generation of technologically-savvy and digitally native art buyers.

By developing a dedicated blockchain and partnering with established art world brands, we can bypass many of the trappings of the traditional object-based art market: there are no logistics with digital art, no storage fees, and no “insurance-while-in-transit” required. Provenance is verified, uniqueness is assured, and the market is already set up for “peer-to-peer” trades between collectors, allowing the art to seamlessly circulate while accruing value. Blockchain is a self-governing system. There are different types of users, and multiple levels of nodes which serve various functions crucial to maintaining the health and security of the distributed network. One such function is governance, in which certain users either buy or receive tokens that give them a “stake” in governing the blockchain. These users can vote on issues related to system updates and other changes that help blockchain’s technology stay relevant and user-friendly.

8.FUTURE ENHANCEMENT

In future system could be enhanced when the rules of the company changes. The database is designed in such a way that could adhere to the changes in the rules of the company. The future and latest technology can also be implemented, and the system could be enhanced.

Every blockchain has a software repository that contains a source code. This source code defines the blockchain’s implementation – its protocol. People involved in a blockchain project need to determine how updates to the software protocol are made. These updates must be coordinated, and this is where governance comes in. Thus, blockchain governance has been defined as the means of achieving the direction, control, and coordination of stakeholders within the context of a given blockchain project to which they jointly contribute. Blockchain governance is essential to the successful maintenance of the entire network, because it determines what updates will be implemented, when and how, based on votes by the blockchain’s stakeholders.

Not all stakeholders are created equal, however, and some have more voting power than others. Our intention is to create a distributed governance protocol with stakeholders from the art industry such as galleries, museums and collectors. This will control how and when this distributed governance will be implemented across the network.

While we don’t anticipate the need to veto or override votes, this contingency system is designed to occur on the side of caution. Stakeholder status will be based on multiple factors, including current

stake, previous activities (such as the user's art world profile), creation of value for the community, and how early stakeholders joined the BCA network (i.e., "Early supporter" status). Issue and implement modification rights, upon successful implementation of signaling rights Implement quadratic voting - gives minority stake voters a greater stake percentage, so they cannot be so easily overruled by more "powerful" stakeholders within the network. It is confident that by enabling art industry partners a stake in the governance, the entire network will benefit on numerous levels. Through the governance, the blockchain will be more secure, more sustainable over time and will be able to evolve and adapt as technology and the art market continue to evolve and change. By taking the long view, it will provide the art world and its stakeholders a blockchain solution that they are truly invested in, giving them the ability to direct and shape the future of the industry.

BIBLIOGRAPHY

1. Battro, A.M. André Malraux revisited: From the musée imaginaire to the virtual museum. X World Congress Friends of Museums, Sydney, September 1999. http://www.byd.com.ar/xwcfm99. htm/ (accessed on 9 November 2010).

2. Louvre Museum [1], Paris. http://www.louvre.fr/ (accessed on 13 May 2013).

3. Parry, R. Museums in a digital age. Routledge, Abingdon, Oxon OX14 4 RN, 2010.

4. Chen, S.E. Quick time VR: an Image-based approach to virtual environment navigation. In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, 1995. pp. 29-38.

5. White, M., et al. Multimodal mixed reality interfaces for visualising digital heritage. Int. J. of Architectural Comp. (IJAC), 5(2), 322-37.

6. Sylaiou, S., et al. Virtual museums: First results of a survey on methods and tools. In Proceedings of XX International CIPA Symposium, 2005. pp. 1138-43.

7. Gaitatzes, A., Christopoulos, D. & Roussou, M. Reviving the past: Cultural heritage meets virtual reality. In Proceedings of the 2001 Conference on Virtual Reality, Archaeology and Cultural Heritage, VAST 2001, 2001. pp. 103-10.

8. Shum, H.Y. & Kang, S.B. A review of imagebased rendering techniques. In Proceedings of International Conference on Visual Communication and Image Processing (VCIP 00), 2000, SPIE, 4067. pp. 2-13.

9. Remondino, F. & El-Hakim, S. Image-based 3D modelling: A review. The Photogrammetric Record, 2006, 21(115), 269-91.

10. El-Hakim, S., et al. Detailed 3D reconstruction of large-scale heritage sites with integrated techniques. IEEE Comp. Graph. Appli., 2004, 24(3), 21-29.

11. Semião, P.M. & Carmo, M.B. Virtual art gallery tool. In Proceedings of GRAPP 2008, 2008. pp. 471-76..

12. Web 3D. http://www.web3d.org/ (accessed on 13 May 2013).

13. Wojciechowski, R., et al. Building virtual and augmented reality museum exhibitions. In Proceedings of 9th International Conference on 3D Web Tecnology, 2004. pp. 135-144.

14. Lu, W.; Zeng, D. & Pan, J. An XML-based scene description language for 3D virtual museum. In Proceedings of 30th International Conference on Information Technology Interfaces, ITI 2008, 2008. pp. 445-50