**Photo Bazaar**

Shiyuan Xu | Chang Liu | Xiaoxing Pan

# PART1:

### **PROJECT DESCRIPTION**

“Need visual inspiration? Explore on PHOTOBAZAAR,

Share your creative vision? Post on PHOTOBAZAAR!”

Our project focuses on creating an Online artwork Sharing and Trading Platform, a digital marketplace where users can share, manage, and trade their original artworks.

User Management: The platform will offer user management capabilities, allowing users to create accounts, customize profiles, and manage their personal information. They can also upload, organize, and showcase their artworks. The platform will support image uploads, descriptions, and tagging to facilitate easy discovery by potential buyers.

Artwork Trading: Users can explore a diverse range of artworks, purchase them, and complete transactions, while they can monitor sales and earnings.

In summary, our project aims to empower artists and enthusiasts with a user-friendly platform for sharing, trading, and appreciating original artworks and images in the digital area.

### **FEATURES**

* User Profiles
* Image Upload
* Keyword and Description Tags
* View artwork and artwork details
* Purchase Options
* Reuse Rights
* Search Filters
* Admin manage users
* Admin manage images

### **TECHNOLOGIES & LIBRARIES CHANLLENGES**

* MERN (***MongoDB***, Express, React, Node.js)
* RESTful API
* Jsonwebtoken, CORS, Axios
* ***AWS S3 BUCKET***
* ***Cart.js***
* ***Stripe***
* ***Messages between seller and buyer***
* ***MailTrap***

### **URLS**

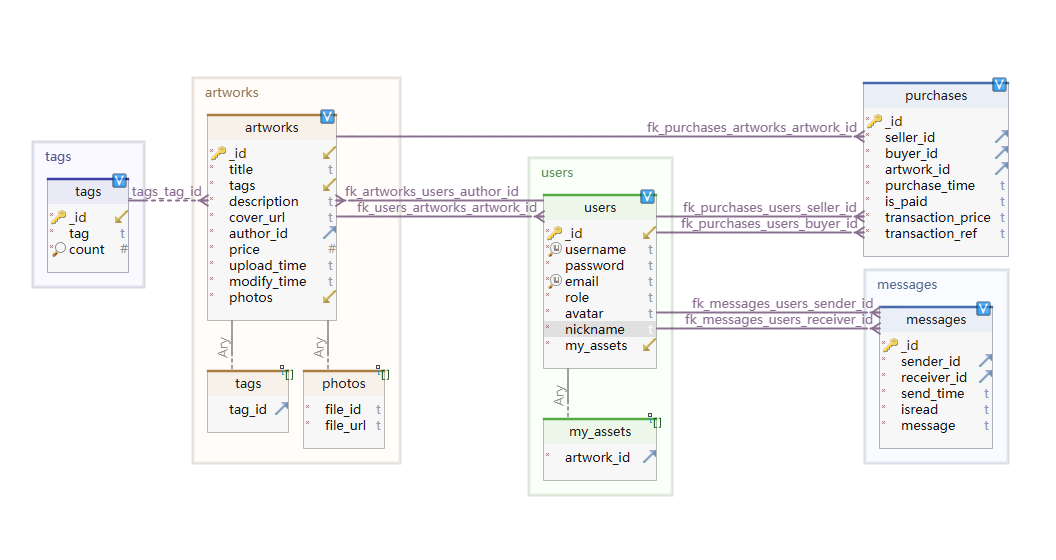
* /login
* /register
* /index
* /search
* /artwork/123 ( info about an artist, list of their pieces)
* /messages/123
* /myaccount
* /myartworks
* /mycart
* /admin/users
* /admin/artworks

### **API CALLS**

* /api/users/login
* /api/users/register
* /api/users/admin
* /api/artworks
* /api/artworks/purchase
* /api/artworks/admin

# PART 2:

### **DATABASE DESIGN: E.R. DIAGRAM**



This database structure is implemented using MongoDB, allowing to store user information, artwork details, and purchase records. The relationships between different collections are established through ID references, enabling data to be queried and utilized relatedly.

**Users Collection** is used to store user information. Each user document includes the following fields:

\_id: Unique identifier for the user.

username: User's username.

password: Hashed password.

email: User's email address.

role: User's role, typically indicating whether the user is a regular user or an administrator.

avatar: Url of user’s avatar

nickname: user’s nickname

my\_assets: an array containing IDs of artworks created or purchased by the user

**Artworks Collection** is used to store information about artworks. Each artwork document includes the following fields:

\_id: Unique identifier for the artwork.

title: Title of the artwork.

tags: An array containing tag objects describing the artwork's tags.

description: Description of the artwork.

cover\_url: URL of the artwork's cover image.

author\_id: User ID of the author, linking to a user in the Users collection.

price: Price of the artwork.

upload\_time: Timestamp of when the artwork was uploaded.

modify\_time: Timestamp of when the artwork was modified.

photos: An array containing photos. It includes file IDs and URLs.

**Purchases Collection** is used to store purchase records. Each purchase record document includes the following fields:

\_id: Unique identifier for the purchase record.

seller\_id: User ID of the seller, linking to a user in the Users collection.

buyer\_id: User ID of the buyer, linking to a user in the Users collection.

artwork\_id: ID of the purchased artwork, linking to an artwork in the Artwork collection.

purchase\_time: Timestamp of when the purchase was made.

is\_paid: Boolean indicating whether the purchase has been paid for.

transaction\_price: The price at which the purchase was made.

Transaction\_ref： Unique reference for payment transaction

**Messages Collection** is used to store messages between users. Each message record document includes the following fields:

\_id: Unique identifier for the message record.

seller\_id: User ID of the sender, linking to a user in the Users collection.

buyer\_id: User ID of the receiver, linking to a user in the Users collection.

send\_time: Timestamp of when the message was sent.

is\_read: If the message was read by the receiver

message: the content of the message

Tags Collection is used to store tags attached to artworks. Each tag record documents includes the following fields:

\_id: Unique identifier for the tag record.

tag: the tag string

count: how many artworks were attached with the tag.

# PART 3:

### CASE DIAGRAM: ACTORS & ACTIONS

A screenshot of a computer

Description automatically generated

# PART 4:**MOCKUPS**

Home page:

A screenshot of a computer

Description automatically generated

Admin profile :

A screenshot of a computer

Description automatically generated

My profile:

A screenshot of a computer

Description automatically generated

My assets:

A screenshot of a website

Description automatically generated