

# Project Outline – Virtual 2D-Art Gallery

## #1 Web-Application Design motivations and ideas:

This **Virtual 2D-Art Gallery web-application** aims to provide users a glimpse of some amazing 2D-artworks (focused on paintings, drawings, prints) exhibited/hosted by National Gallery of Art (NGA). With the current restrictions and inconvenience in international travel, we'd like to provide an alternative virtual gallery browsing experience to the people who cannot travel to visit these artworks in person.

Meanwhile, this web-application also serves an educational and research purpose, as it provides an introduction to some of the finest artworks by a group of prominent artists around the world. For instance, our **searching functionality** will allow users browse artworks by artists, by title of artworks, by geological origins, by chronology, etc.. We will also perform some **analyses** of these artworks based on their **contents/styles/historical/geographical/etc. associations**. These analytical results can provide more insights about the rationales and motivations behind the creation of these artworks.

Additionally, we wish our website to be a source of **aesthetic inspiration** for art creators. We will achieve this by tailoring the aesthetic designs of our webpages with the visual/color components of the artworks being displayed. (i.e. with the help of some amazing REACT-library tools such as the “react-color-extractor” package). There will also be some interesting functionalities such as “to give the user an idea about the actual dimensions of artworks by asking the height of the user, and display some artworks with regard to this height”. We also want to try to implement a **user Login system**, which can support some user-customized functionalities such as “having a collection of favorite artworks for each user-account” (the implementation of this feature is still in discussion).

Lastly, we hope our virtual site experience can encourage people to make physical tours to NGA, which will complement them with a more holistic perceptual experience of these artworks.

## #2 Definite Features to be implemented:

### Feature #1 – Search Artworks (Search Page)

1. **Filter By (Dynamic Query)**
  - provide cues/options for searching, i.e. buttons that have “Nationality”, “Time Span”, “Style”, “Artist Last Name”, etc.
2. **Specific keyword Search (Dynamic Query):**
  - with an input box, user enters name of the **artist**
    - Display artworks by the same artist in a chronological manner (ex. With a time axis)
  - With an input box, user enters name of the **artwork**
3. **Recommend by similarity (Dynamic Query)**
  - When User is browsing a specific artwork, at the bottom of the page, we will also recommend several other pieces of artwork by the same artist, or from the same collection/portfolio/series, or from the same style

### Feature #2 – Naughty Search (Search Page)

1. Search by User's Height:
  - User enters his/her height, we return a set of artworks that is around the similar height
2. Search by User's Birthyear:
  - User enters birth year, return the artwork that was created 100/200/300 years ago, return the results by dimension of artworks Large → Small

### Feature #3 – Artwork Analysis (Analysis Page)

## Project Outline for Team--DataOmini

- 
1. **Semantic Analysis:**
    - By Contents of Artwork (i.e. keyword from relation-“object\_terms”), static analysis (may be able to achieve dynamic within a smaller range) => end result will be made into some collections/albums of artworks, put on Homepage → section box “analysis result collections”
    - Portrait change across time
  2. **Geographical Analysis (Put on Home Page):**
    - analyze the artwork geographical density by constituent’s Nationality, display the result as a world-map view, with regions involved as colored sections.
  3. **Artworks collections in different time span or originated regions:**
    - Similar function: List art pieces for the same artist and arrange by time course

## #3 Potential Features to be implemented:

### Feature #4: An User Login System (Login/Account Page)

- Features an additional set of tables in our database:
- Accounts(UserID, Password)
- UserArtCollection(ArtworkID, UserID)

### Feature #5: Artwork Analysis (Analysis Page)

- Color palette analysis on 4000~painting, generate top 10~20 most popular color range/combinations, feature them as recommendation on the home page

## #4 Web Pages:

### Home Page: /home or /

- Section 1: Welcoming message
- Section 2: General information about our artwork collection (i.e. quantity and geographical origins)
- Section 3: recommendations about popular artworks and artists. (hyperlink to these artworks (to object page) or artists to search page)
- Section 4: A highlight/preview of our analysis results and hyperlinks to our analysis page

### Search Page: /search

- For searching artworks and display search result
- Users can use our suggested filters to search for specific artworks. Also support keyword search: user type-in **artist’s name, or artwork’s name**.
- subpage/nested page: artwork-object:
  - When view individual artworks, each artwork get its own page of detailed visual display, along with information about this artwork
  - At the bottom of the object page, we will recommend some similar artworks, depending on the artist/style/year/etc.
- Contents on this page mostly involving dynamic query

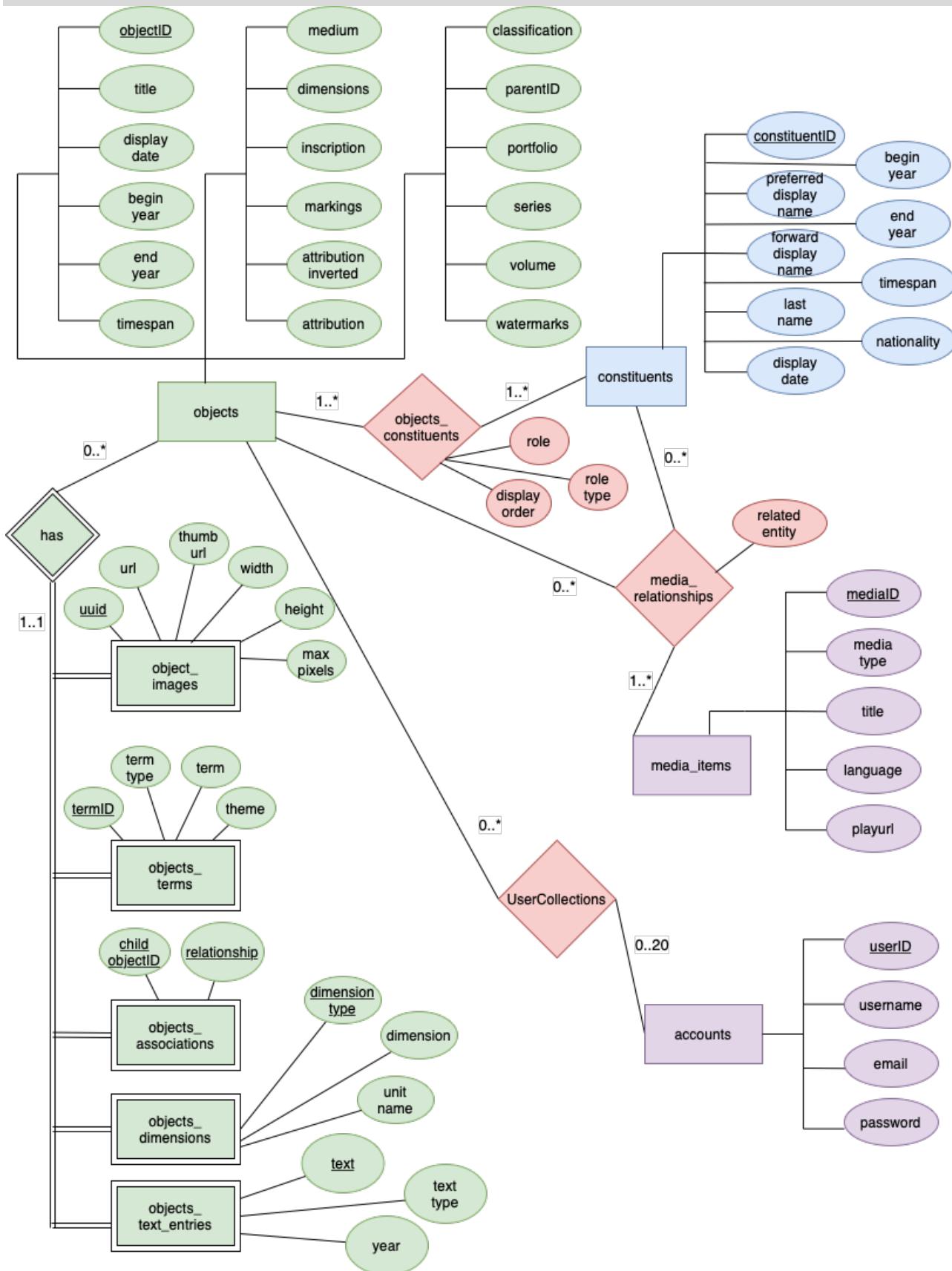
### Analysis Page: /analysis

- For showing the analyzes result of our artworks collection
- Analyzes are predetermined by us (website maker), query are mostly static with little to no adjustability of dynamic contents

### LoginPage (optional): /login:userID or /auth

- Prompt user to create an account, or login as an existing user
- If a user is logged in, we can allow the user to store a collection of 20 artworks of his/her preference.

## #5 ER Diagram:



## #6 SQL DDL for database creating:

## Definite/Core Relations

```
#1
CREATE TABLE objects (
    objectID int,
    title varchar(2048),
    displayDate varchar(256),
    beginYear int,
    endYear int,
    timeSpan varchar(32),
    medium varchar(2048),
    dimensions varchar(2048),
    inscription varchar,
    markings varchar,
    attributionInverted varchar(1024),
    attribution varchar(1024),
    classification varchar(32),
    parentID int,
    portfolio varchar(2048),
    series varchar(850),
    volume varchar(850),
    watermarks varchar(512),
    PRIMARY KEY (objectid),
    FOREIGN KEY (parentid) REFERENCES objects(objectid)
);

#2
CREATE TABLE objects_terms (
    termID int,
    objectID int,
    termType varchar(64) NOT NULL,
    term varchar(256) NOT NULL,
    visualBrowserTheme varchar(32),
    PRIMARY KEY(termID, objectID),
    FOREIGN KEY(objectID) REFERENCES objects(objectID)
);

#3
CREATE TABLE objects_associations (
    parentObjectID int,
    childObjectID int,
    relationship varchar(32) NOT NULL,
    PRIMARY KEY(parentObjectID, childObjectID, relationship),
    FOREIGN KEY(parentObjectID) REFERENCES objects(objectID),
    FOREIGN KEY(childObjectID) REFERENCES objects(objectID)
);

#4
CREATE TABLE objects_dimensions (
    objectID int,
    dimensionType varchar(32) NOT NULL,
    dimension decimal(22,10) NOT NULL,
    unitName varchar(32) NOT NULL,
    PRIMARY KEY(objectID, dimensionType)
    FOREIGN KEY(objectID) REFERENCES objects(objectID),
)
```

> InnoDB (65xxx  
bytes)  
> MyISAM  
  
TEXT  
BLOB  
  
> utf8mb4  
utf8mb3

```
#5
CREATE TABLE objects_text_entries (
    objectID int,
    text varchar NOT NULL,
    textType varchar(32) NOT NULL,
    Year int varchar(4) NOT NULL,
    PRIMARY KEY(objectID, text)
    FOREIGN KEY(objectID) REFERENCES
objects(objectID),
)

#6
CREATE TABLE objects_constituents (
    objectID int,
    constituentID int,
    displayOrder int NOT NULL,
    roleType varchar(64) NOT NULL,
    role varchar(64) NOT NULL,
    PRIMARY KEY(objectID, constituentID)
    FOREIGN KEY(objectID) REFERENCES
objects(objectID)
    FOREIGN KEY(constituentID) REFERENCES
constituents(constituentID)
);

#7
CREATE TABLE objects_images (
    uuid varchar(64),
    URL varchar(512),
    thumbURL varchar(512),
    width int,
    height int,
    maxpixels int,
    objectID int,
    PRIMARY KEY (uuid),
    FOREIGN KEY (objectID) REFERENCES
objects(objectID)
);

#8
CREATE TABLE constituents (
    constituentID int,
    preferredDisplayName varchar(256),
    forwardDisplayName varchar(256),
    lastName varchar(256),
    displayDate varchar(256),
    beginYear int,
    endYear int,
    visualBrowserTimeSpan varchar(32),
    visualBrowserNationality varchar(128),
    PRIMARY KEY(constituentID)
)
```

**Optional Relations:**

To display supplementary media(audio/video) of an artwork:	For the User Login System:
<pre>#9 CREATE TABLE media_items (     mediaID int,     mediaType varchar(32),     title varchar(2048),     language varchar(12),     playURL varchar(1024),     PRIMARY KEY(mediaID) )  #10 CREATE TABLE media_relationships (     mediaID int,     relatedID int,     relatedEntity varchar(32),     PRIMARY KEY(mediaID, relatedID, relatedEntity)     FOREIGN KEY(mediaID) REFERENCES media_items(mediaID)     FOREIGN KEY(relatedID) REFERENCES objects(objectID) )</pre>	<pre>#11 CREATE TABLE accounts (     userID int(11) NOT NULL AUTO_INCREMENT,     username varchar(50) NOT NULL,     email varchar(100) NOT NULL,     password varchar(255) NOT NULL,     PRIMARY KEY (userID) ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;  #12 CREATE TABLE UserCollections (     userID int(11) NOT NULL,     objectID int NOT NULL,     PRIMARY KEY (userID, objectID),     FOREIGN KEY (userID) REFERENCES accounts(userID),     FOREIGN KEY (objectID) REFERENCES objects(objectID) );</pre>

## #7 Data Cleaning

Our data was collected from the official Github repository of the National Gallery of Art (NGA), thanks to their [Open Access Project](#). So most of our data-tables are already in a well designed schema and relationships.

Hence, our data cleaning will be focusing on extracting the artworks of types of our interest. For instance, our aim is to build a gallery for 2D-artworks only, which are of the types: **paintings, prints and drawings**. In NGA's collection, there are also many photographs, sculptures, and virtual artworks, which are outside our interest. So we will use the help of **Python Pandas library**, to clean our data with the following protocol:

- **Step 1:** for our most essential data-table “**objects.csv**”, we will use the column named “**visualbrowserclassification**” to identify if an object is of one of the three types: painting, drawing, or print. We will only keep objects of these types. We get a partially processed data-table named “**objects\_2D.csv**”
- **Step 2:** Due to the visualizing nature of our web-application, which is an art gallery, our main functionalities are centered around displaying artworks. Therefore, we need to check if every artwork of our interest has a digital image to be displayed, which depends on if NGA grants the access of such an image as open data. So we will use the data-table “**published\_images.csv**”, to check for digital images availability. And we will **join** the data in “**objects\_2D.csv**” with “**published\_images.csv**”, and extract for tuples that have a digital image. At the end of this step, we will get a reduced data-table “**objects\_cleaned.csv**”, and we can further clean other data-tables with the constraints of this key table.
- **Step 3:** for data-table “**constituents.csv**”, we will keep the constituents that are the creators of the artworks of our interests, discard everyone else (i.e. donors). We get “**constituents\_cleaned.csv**”
- **Step 4:** for data-table “**objects\_constituents.csv**”, we use “**objects\_cleaned.csv**” and “**constituents\_cleaned.csv**” to extract the object-constituent relationships that are of our interest, which gives the “**objects\_constituents\_cleaned.csv**” data-table.
- **Step 5:** for data-tables “**objects\_terms.csv**”, “**objects\_associations.csv**”, “**objects\_dimensions.csv**” and “**objects\_text\_entries.csv**”, we use “**objects\_cleaned.csv**” to extract only the informations about artwork objects of our interests, which give us “**objects\_terms\_cleaned.csv**”, “**objects\_associations\_cleaned.csv**”, “**objects\_dimensions\_cleaned.csv**” and “**objects\_text\_entries\_cleaned.csv**”
- During the entire cleaning process, we will also check if there are any **NUL / Missing values for the key columns**, and if so we will discard these tuples (i.e. discard this artwork object from our collection).

## #8 Technologies

- MySQL, Python (Pandas), Javascript, React, Express.js, Node.js

## #9 Duty Assignment:

Back-End / Database & Query Design	Front-End / Web Page Design
<b>Data Cleaning</b> <ul style="list-style-type: none"><li>• Xianjie (Iris) Ma</li><li>• Yinjie Zhang</li></ul>	<b>UI- Aesthetic Design</b> <ul style="list-style-type: none"><li>• Xianjie (Iris) Ma</li><li>• Yinjie Zhang</li></ul>
<b>Database Design &amp; Database Management</b> <ul style="list-style-type: none"><li>• Yinjie Zhang</li><li>• Xianjie (Iris) Ma</li></ul>	<b>Front-End Website Architecture Design</b> <ul style="list-style-type: none"><li>• Lingzi Xiao</li><li>• Xue Liu</li></ul>
<b>Database &amp; Query Optimization</b> <ul style="list-style-type: none"><li>• Yinjie Zhang</li><li>• Lingzi Xiao</li></ul>	<b>Front-End Functionality Development</b> <ul style="list-style-type: none"><li>• Lingzi Xiao</li><li>• Xue Liu</li></ul>
<b>Back-end: Database &amp; Query Testing</b> <ul style="list-style-type: none"><li>• Xianjie (Iris) Ma</li><li>• Xue Liu</li></ul>	<b>Front-End Functionality Testing</b> <ul style="list-style-type: none"><li>• Yinjie Zhang</li><li>• Xianjie(Iris) Ma</li></ul>

# Website Design Prototype

[home](#) [search](#) [analysis](#) [login](#)



i.e. shows popular geographical origins of artworks as a coloured world map

WELCOME TO THE VIRUAL GALLERY !!

explore our collection of

+4000 paintings,

+30000 drawings,

and +60000 prints

from more than 25 popular regions worldwide

Display some Recommendations:

the popular/famous artworks in our collection

the popular/famous artists in our collection

Pablo Picasso

Vincent van Gogh

Claude Monet

Leonardo da Vinci

Paul Cézanne

Analysis Highlights:

Semantic Analysis By Contents of Artwork

surrealism

lanscape

Wars

Baroque

Protraits

Artworks collections in different time span or originated regions

1500s

1700s

1900s

French

American

search by artwork:



search by artist



Filters:

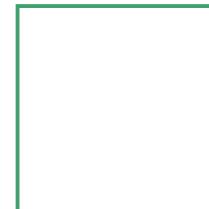
 nationality

object

object -&gt; term

object -&gt; constituent

object

 style time span medium

1..2.3.4.5.6.....100

**Naughty Search:**

height :

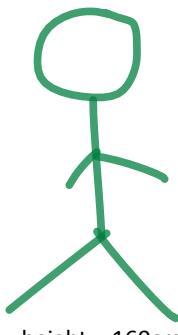
&gt; find artwork with dimensions close to your height

enter your height (cm):

birthyear:

&gt;search artworks by ~100/~200/~300 years before user's birthyear, then rank by dimensions of artwork from largest to closes

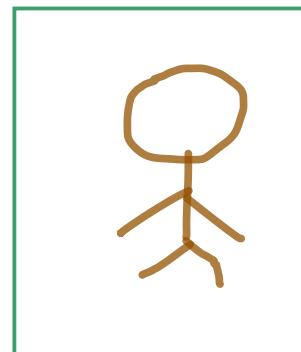
enter your birthyear:



height = 160cm

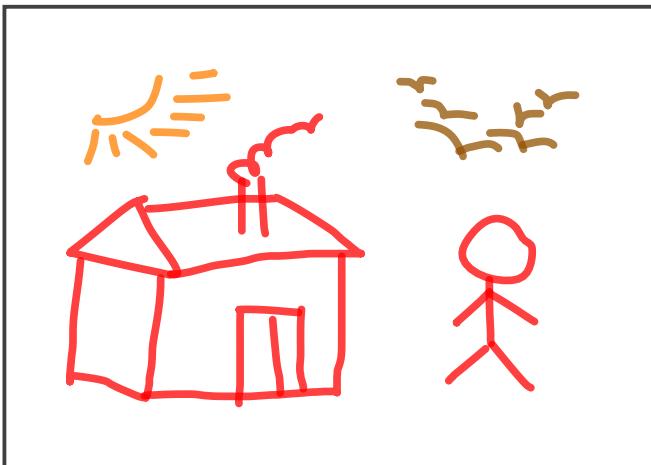


“afternoon tea”



“a portrait of Lisa”

home **search** analysis login



Title: A person and his house

Medium: Water-colour on Paper

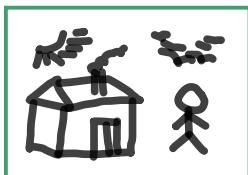
Dimension: 20cm x 30cm

Artists: Fake Name

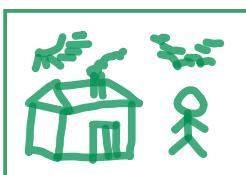
Color Palette:



Similar Artworks:



A person and his house-Black



A person and his house-Green



A person and his house-Purple



A person and his house-Blue

### Semantic Analysis By Contents of Artwork

surrealism

lanscape

Wars

Baroque

female

### Artworks collections in different time span or originated regions

1500s

1700s

1900s

French

American

analysis by time

portrait change across time

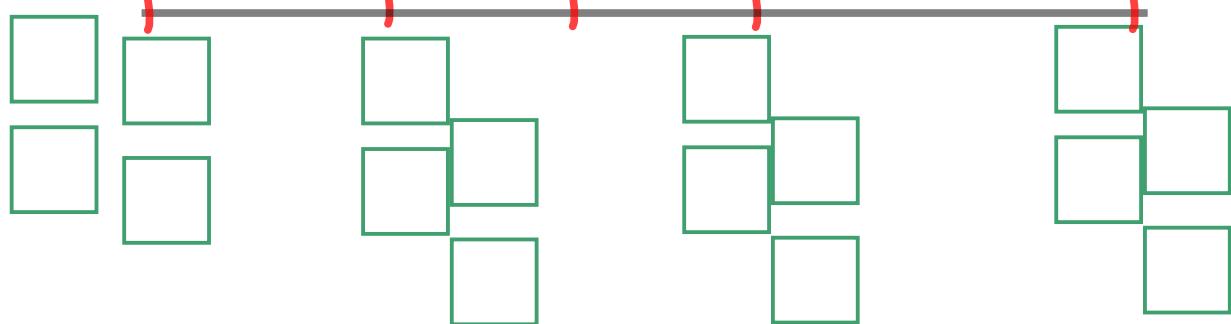
1500

1600

1700

1800

1900



if we were to login system feature, the login page would look like this

Register for A New Account:

email:   
username:   
password:

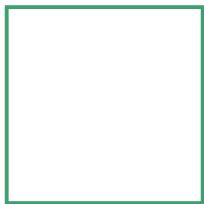
REGISTER

Existing User Login

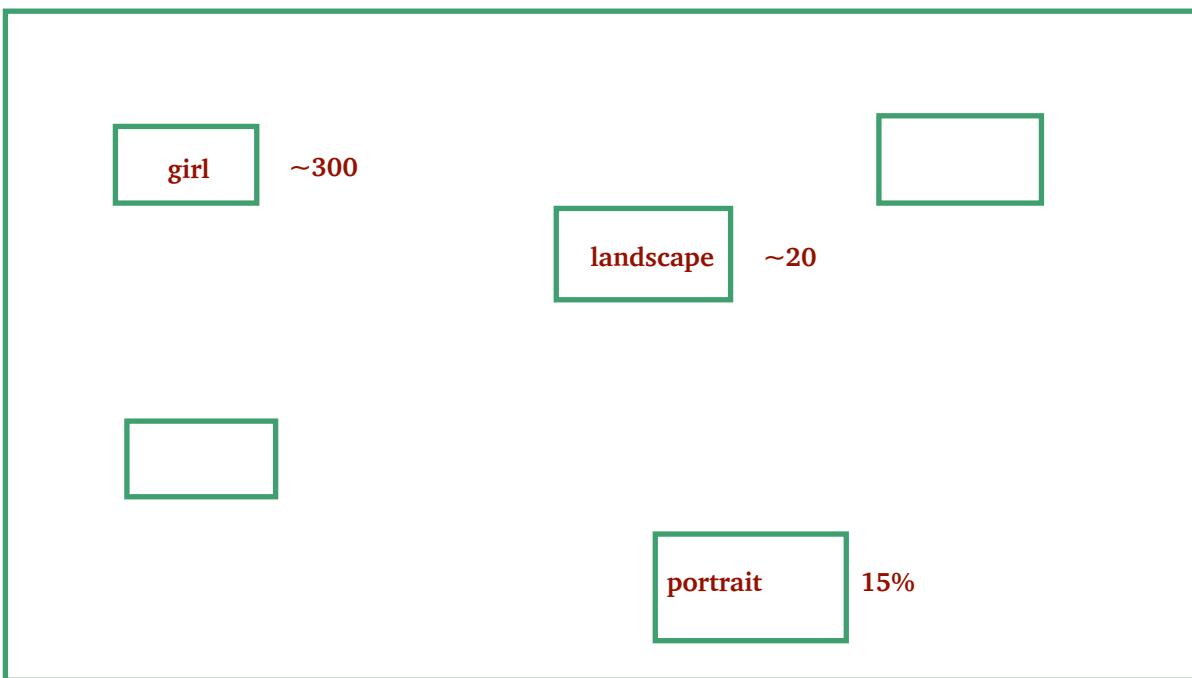
email:   
password:

LOGIN

MY ARTWORK COLLECTION (MAX 20):



top 20 popular keywords:



girl objectID  
landscape ↗