Artiverse: A Map-Based Art Community Platform

Team Members:

Risa Xie (yantongx) | Yuxin Deng (yuxinden) | Karry Mao (karrym)

Product Backlog

The product backlog consists of all major functionalities, organized into modules:

A. Core Features (Essential for MVP)

1. **User Authentication System**: Email/password registration & login, OAuth, password reset.
2. **Profile Management**: Update user profile, avatar, and bio.
3. **Artwork Upload:** Users upload paintings with title, description, and location. Location will be extracted automatically from uploaded photos (if metadata is available) or manually defined by the user.
4. **Artwork Showcase**: Display uploaded paintings in timeline and map view.
5. **Community Map**:
    a. Gallery(Left): Scrollable display of artworks sorted by date/popularity.
    b. Map(Right):View global landscape artworks on an interactive map.
6. **Search by Location**: Find artworks based on geographic location.
7. **Cloud Storage Integration**: AWS S3 for image storage.
8. **Database Management**: MySQL for storing user and artwork data.
9. **API Development**: RESTful APIs for frontend-backend communication.

B. Optional Features (Potential Enhancements)

13. **Comment System**: Users can comment on artworks.
14. **Like System**: Users can like artworks.
15. **Location Bookmarking**: Users can save locations for future inspiration.
16. **User Follow System**: Follow artists to see their updates.

Data Model:

Python
```python
from django.db import models
from django.contrib.auth.models import User
from django.conf import settings

class Post(models.Model):
    content = models.TextField()
    time = models.DateTimeField(auto_now_add=True)
    author = models.ForeignKey(User, on_delete=models.CASCADE)
    id = models.AutoField(primary_key=True)

    def __str__(self):
        return f'{self.id} Post'

    def to_json(self):
        return {
            'id': self.id,
            'author_username': self.author.username,
            'author_firstname': self.author.first_name,
            'author_lastname': self.author.last_name,
            'content': self.content,
            'time': self.time.isoformat(),
            'comments': [comment.to_json() for comment in self.comments.all()]
        }

class Comment(models.Model):
    content = models.TextField()
    time = models.DateTimeField(auto_now_add=True)
    author = models.ForeignKey(User, on_delete=models.CASCADE)
    post = models.ForeignKey('Post', on_delete=models.CASCADE,
related_name='comments')

    def __str__(self):
        return f'{self.id} Comment'

    def to_json(self):
        return {
            'id': self.id,
            'author_username': self.author.username,
            'author_firstname': self.author.first_name,
            'author_lastname': self.author.last_name,
            'content': self.content,
            'time': self.time.isoformat()
        }


class Profile(models.Model):
       user = models.OneToOneField(User, on_delete=models.CASCADE)
       profile_picture = models.FileField(upload_to='avatars/', blank=True,
null=True)
       cover_photo = models.ImageField(upload_to='covers/', blank=True,
null=True)
       following = models.ManyToManyField(User, related_name='followers')
       bio = models.TextField(blank=True)
       location = models.CharField(max_length=30, blank=True)
       fname = models.CharField(max_length=30, blank=True)
       lname = models.CharField(max_length=30, blank=True)
       created_at = models.DateTimeField(default=timezone.now)
       featured_artwork = models.ForeignKey(
               'Artwork',
               null=True,
               blank=True,
               on_delete=models.SET_NULL,
               related_name='featured_in_profiles'
           )

       def __str__(self):
               return f'{self.user.username}\'s Profile'
class Bookmark(models.Model):
       user = models.ForeignKey(User, on_delete=models.CASCADE,
       related_name='bookmarks')
       artwork = models.ForeignKey(Artwork, on_delete=models.CASCADE,
       related_name='bookmarked_by')
       created_at = models.DateTimeField(auto_now_add=True)

       class Meta:
              unique_together = ('user','artwork')
              ordering = [-'create_at']
```

First Sprint Backlog

The first sprint will focus on establishing the core functionality needed for user onboarding, profile management, and basic artwork uploading.

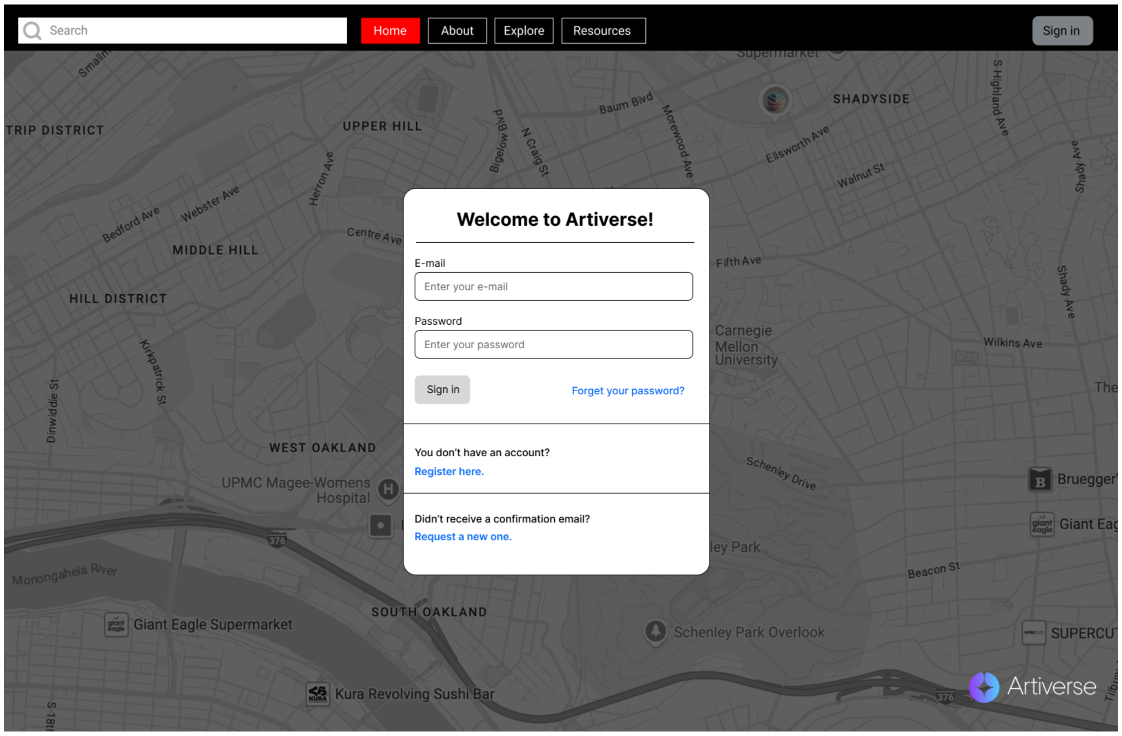
Planned Features for Sprint 1 (March 11 - March 17) [Yuxin Deng (yuxinden)]:

1. **Wire frame design (Risa)**
    - **Walk-through Demo**
    - UI
    - User Flow
2. **Artwork extracted position** (Karry)
    - Extract photo location, time, camera parameters, etc. from photo EXIF fields.
    - If the information is empty, the user can fill it voluntarily.
3. **User Authentication System** (Yuxin)
    - **Implemented user registration and login using Django authentication.**
4. **Profile Management** (Yuxin)
    - **Created user profile models with avatar upload, bio, and location.**
    - **Implemented profile update functionality.**
    - **Added profile picture management.**
5. **Artwork Upload System** (Yuxin)
    - **Implemented image upload with local storage.**
    - **Developed artwork models supporting title, description, location, and creation date.**
6. **Basic Frontend Setup** (Yuxin)
    - Initial HTML form for login, register, profile and artwork creation.

Wireframes & Navigation Overview

Walk-through Demo Video

1. ***Login Page: Users can log in via email/password or Google OAuth.***


Search

Home About Explore Resources

Sign in

Welcome to Artiverse!

E-mail

Enter your e-mail

Password

Enter your password

Sign in

Forget your password?

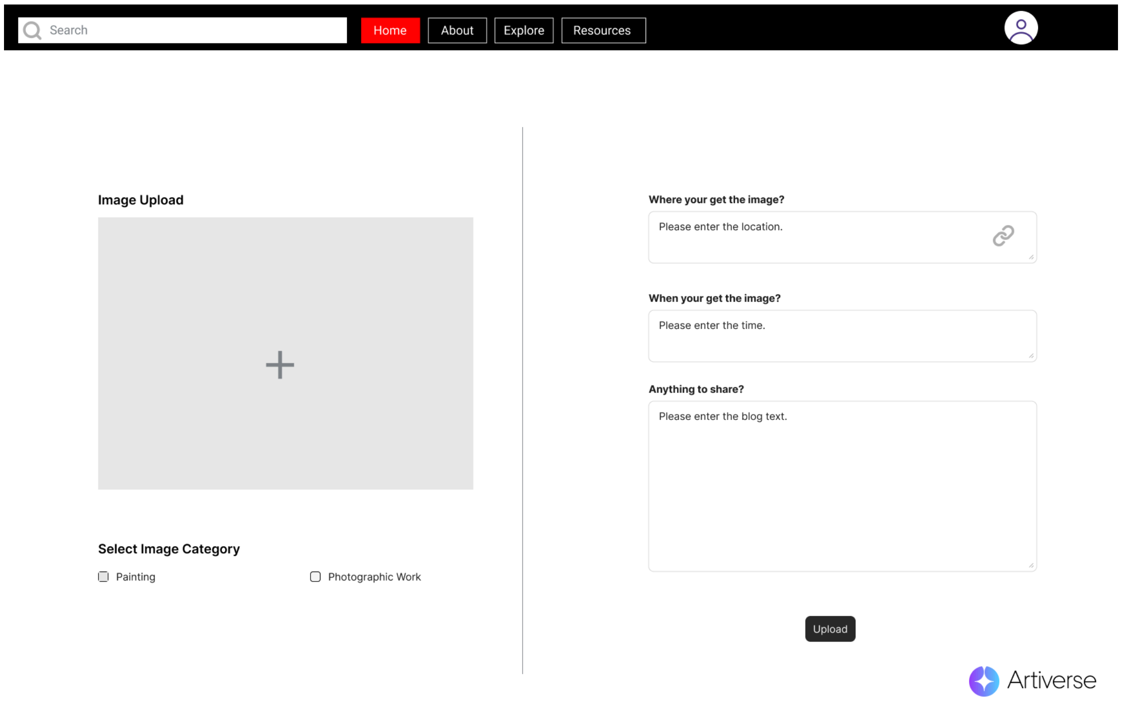
You don't have an account?

Register here.

Didn't receive a confirmation email?

Request a new one.

2. Upload Page: Users upload new artwork, providing title, description, and location.


Search

Home About Explore Resources

Image Upload

Select Image Category

Painting Photographic Work

Where your get the image?

Please enter the location.

When your get the image?

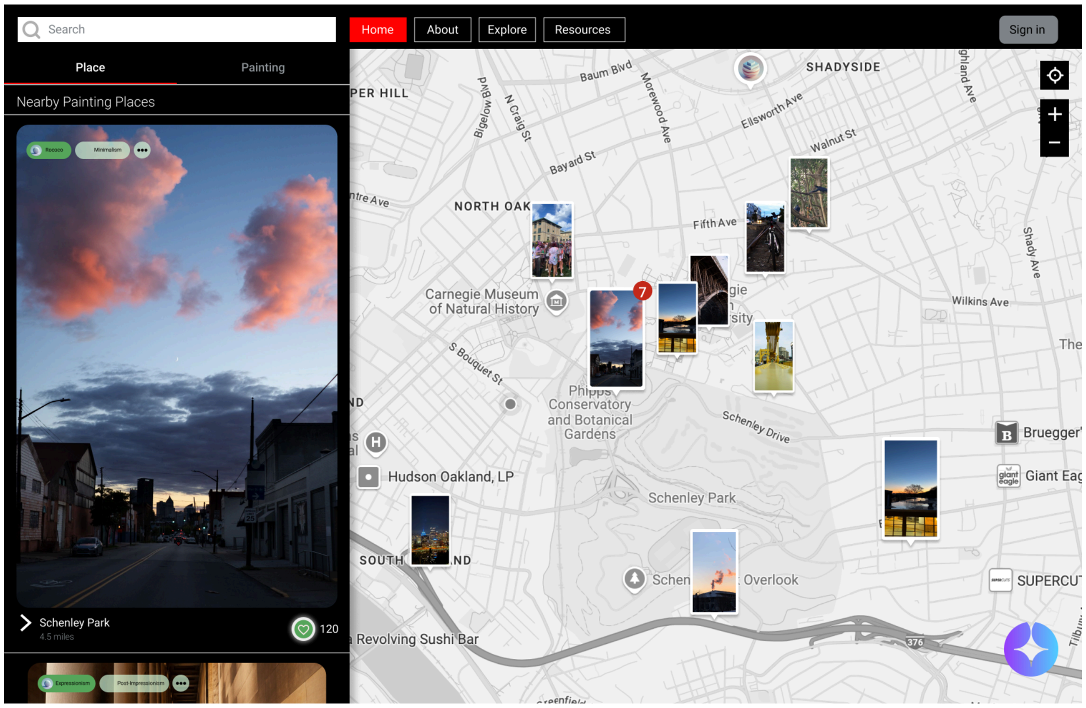
Please enter the time.

Anything to share?

Please enter the blog text.

Upload

3. ***Community Map:***
    a. ***Map (Right): Interactive map displaying uploaded artworks.***
    b. ***Gallery (Left): Displays all artworks in a scrollable feed, sorted by popularity or date.***



4. ***Profile Page: Displays user profile info and uploaded artworks.***
5. ***Artwork Detail Page: Displays artwork with location, comments, and Street View integration.***