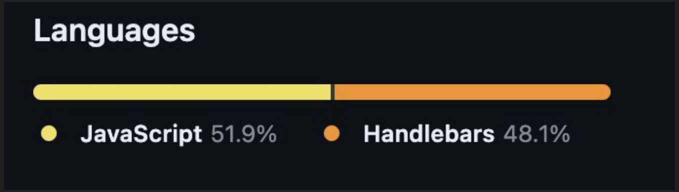
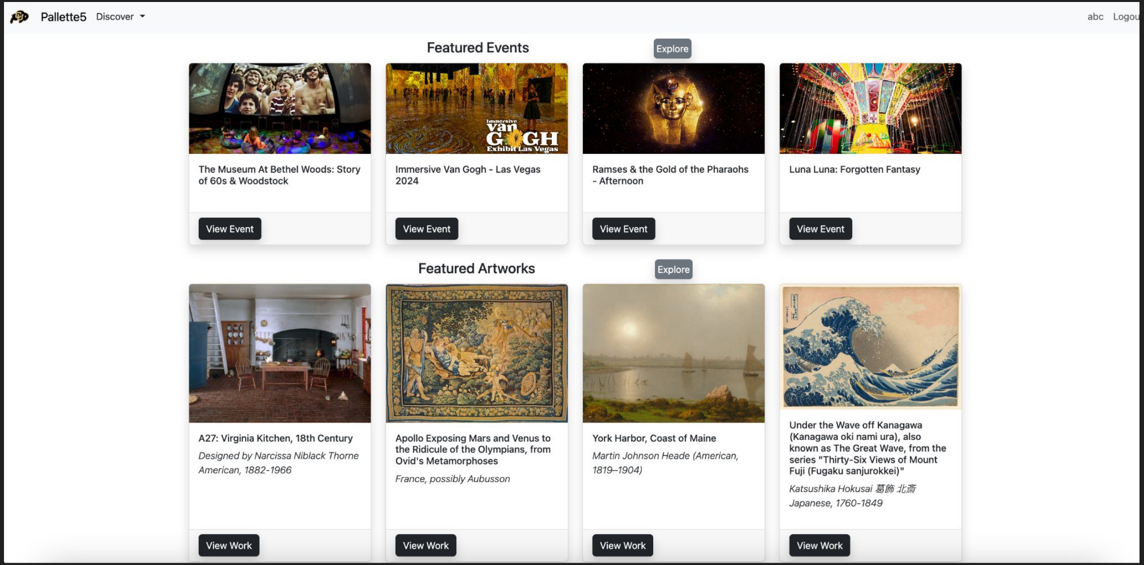
Palette5

Amy Bowers, Nathaniel Covington, Ethan Epperson, Austin McCutcheon, Khizar Pasha, Catherine Xie.



Description of Project

We created a user-interactable artworks page. We allow the user to browse artists, artwork, and local events. The user is then free to follow artists and add a comment to artwork. The user is also able to add events to the event calendar and upload their own artworks, of which show up on the user profile page. The goal of the website is to allow artists to interact with other artists at a more local and personal level.



Featured Events

The Museum At Bethel Woods: Story of 60s & Woodstock

Immersive Van Gogh - Las Vegas 2024

Ramses & the Gold of the Pharaohs - Afternoon

Luna Luna: Forgotten Fantasy

Featured Artworks

A27: Virginia Kitchen, 18th Century
Designed by Narcissa Niblack Thorne American, 1882-1966

Apollo Exposing Mars and Venus to the Ridicule of the Olympians, from Ovid's Metamorphoses
France, possibly Aubusson

York Harbor, Coast of Maine
Martin Johnson Heade (American, 1819–1904)

Under the Wave off Kanagawa (Kanagawa oki nami ura), also known as The Great Wave, from the series "Thirty-Six Views of Mount Fuji (Fugaku sanjurokkei)"
Katsushika Hokusai 葛飾 北斎 Japanese, 1760-1849

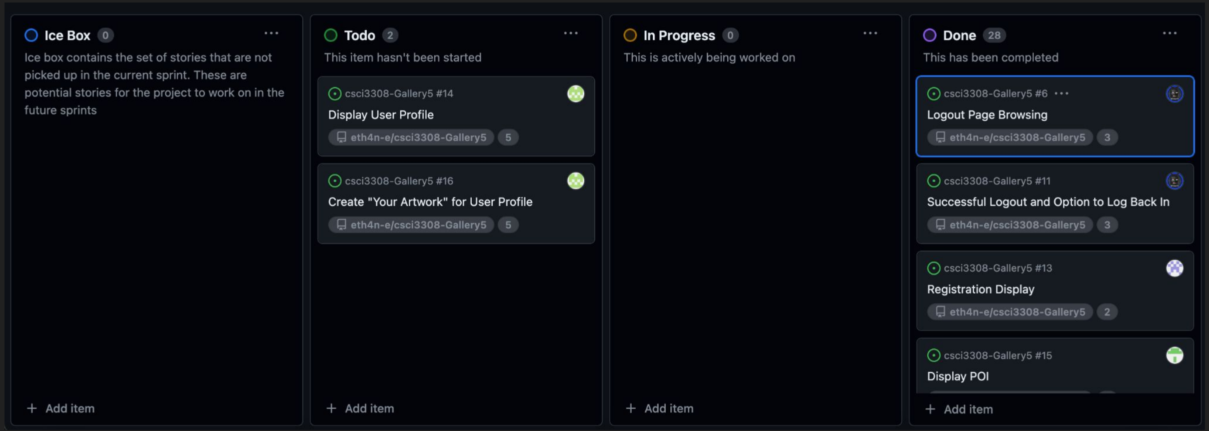
Tools and their respective ratings

Tool	Tool Used	Rating	Notes
Project Tracker	GitHub	4/5	Helpful for stating responsibilities and story points
VCS	GitHub	5/5	Provided helpful snapshots of project state
Database	PostgreSQL	5/5	Easy implementation of database; to include posting queries / getting information
IDE	VSCode	5/5	Least problematic in entire project
User Interface	Handlebars	4.5/5	Easy to work with for the most part
Application Server	NodeJS	4/5	Javascript, async, npm; Some errors occurred without explicit reasoning, not being typed was annoying.
Testing	Mocha/Chai	3.5/5	Had to go through loopholes to get them to work, for this specific project they weren't much more helpful or time saving then just manually trying to break the application and using print debugging.

Tools and their respective ratings (2)

Tool	Tool Used	Rating	Notes
Deployment Environment	LocalHost -Docker	4/5	Worked as expected
API	Google Maps	4/5	It was a bit tricky to use, but Google has a ton of documentation so it was fine.
API	Ticketmaster	3/5	Ticketmaster was very finicky about its parameters. Its responses were also cluttered and had to be manually parsed in the back end
API	Art Institute of Chicago	4/5	Solid artwork calls; did not generate artist's image.
Deployment Environment	Azure	4/5	Had trouble with making an account, otherwise worked fine.
Middleware	Multer	5/5	Very easy to set up, solid middleware.
Cloud Storage	Cloudinary	5/5	Extremely easy to use, more so than Azure. Free.
API	Artsy	1/5	Was very poorly made, often times newer data couldn't be searched, and not much variety in search parameters. Little accurate documentation.

Project Tracking in GitHub



Ice Box 0
Ice box contains the set of stories that are not picked up in the current sprint. These are potential stories for the project to work on in the future sprints

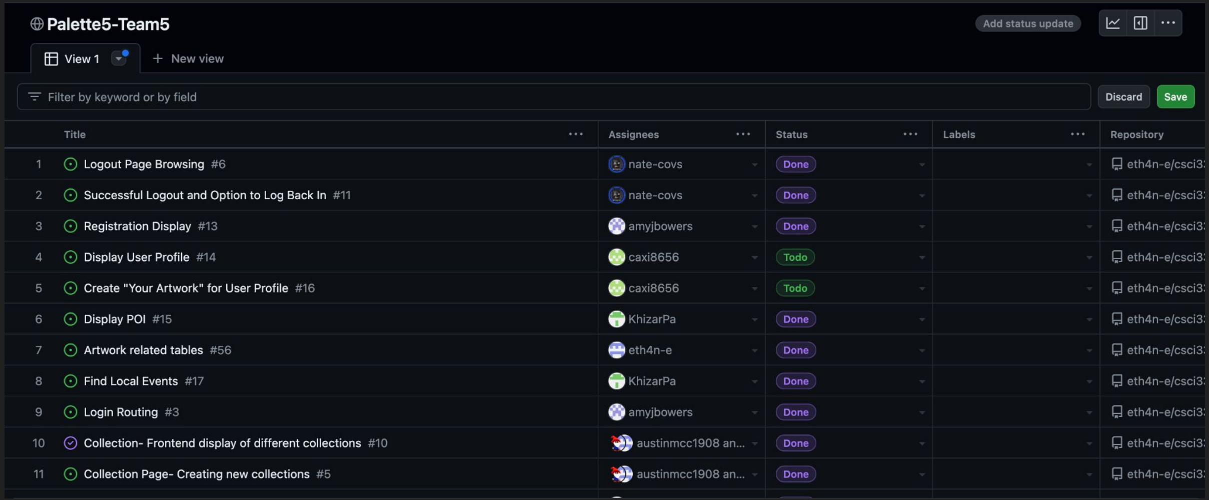
Todo 2
This item hasn't been started

- csci3308-Gallery5 #14
  Display User Profile
  eth4n-e/csci3308-Gallery5 5
- csci3308-Gallery5 #16
  Create "Your Artwork" for User Profile
  eth4n-e/csci3308-Gallery5 5

In Progress 0
This is actively being worked on

Done 28
This has been completed

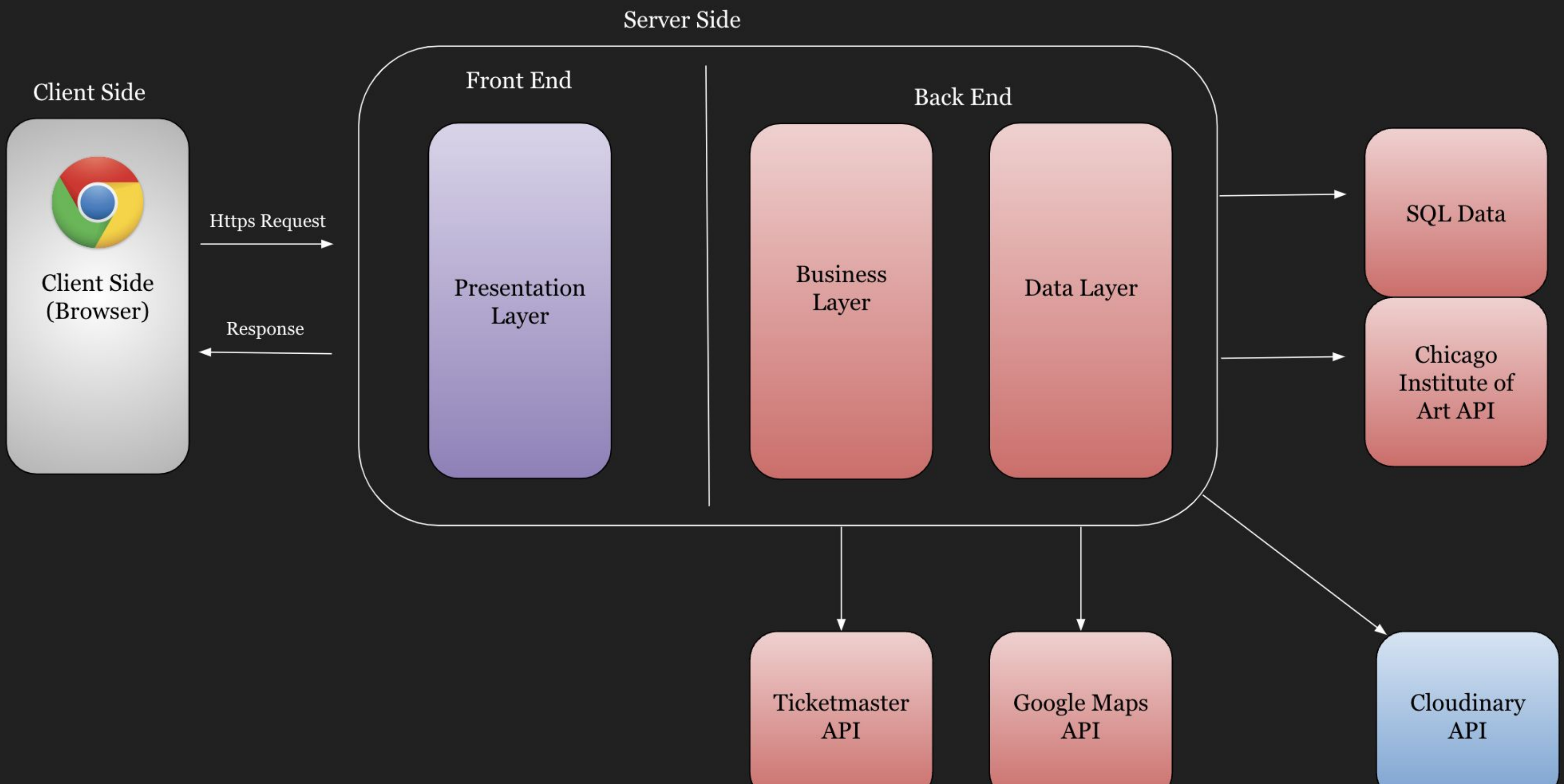
- csci3308-Gallery5 #6
  Logout Page Browsing
  eth4n-e/csci3308-Gallery5 3
- csci3308-Gallery5 #11
  Successful Logout and Option to Log Back In
  eth4n-e/csci3308-Gallery5 3
- csci3308-Gallery5 #13
  Registration Display
  eth4n-e/csci3308-Gallery5 2
- csci3308-Gallery5 #15
  Display POI



Palette5-Team5

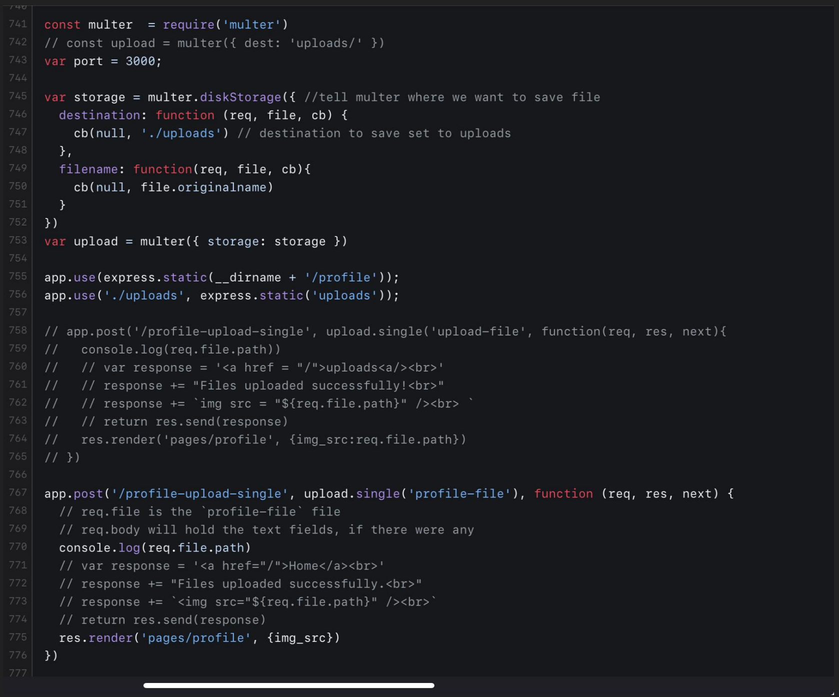
	Title	Assignees	Status	Labels	Repository
1	Logout Page Browsing #6	nate-covs	Done		eth4n-e/csci3
2	Successful Logout and Option to Log Back In #11	nate-covs	Done		eth4n-e/csci3
3	Registration Display #13	amyjbowers	Done		eth4n-e/csci3
4	Display User Profile #14	caxi8656	Todo		eth4n-e/csci3
5	Create "Your Artwork" for User Profile #16	caxi8656	Todo		eth4n-e/csci3
6	Display POI #15	KhizarPa	Done		eth4n-e/csci3
7	Artwork related tables #56	eth4n-e	Done		eth4n-e/csci3
8	Find Local Events #17	KhizarPa	Done		eth4n-e/csci3
9	Login Routing #3	amyjbowers	Done		eth4n-e/csci3
10	Collection- Frontend display of different collections #10	austinmcc1908 an...	Done		eth4n-e/csci3
11	Collection Page- Creating new collections #5	austinmcc1908 an...	Done		eth4n-e/csci3

Architecture Diagram



Challenges

- Figuring out a way for users to upload
- Setting up Multer and the Docker
- Actually saving with Multer



```
const multer  = require('multer')
// const upload = multer({ dest: 'uploads/' })
var port = 3000;

var storage = multer.diskStorage({ //tell multer where we want to save file
  destination: function (req, file, cb) {
    cb(null, './uploads') // destination to save set to uploads
  },
  filename: function(req, file, cb){
    cb(null, file.originalname)
  }
})
var upload = multer({ storage: storage })

app.use(express.static(__dirname + '/profile'));
app.use('./uploads', express.static('uploads'));

// app.post('/profile-upload-single', upload.single('upload-file', function(req, res, next){
//   console.log(req.file.path))
//   // var response = '<a href = "/">uploads<a/><br>'
//   // response += "Files uploaded successfully!<br>"
//   // response += `img src = "${req.file.path}" /><br> `
//   // return res.send(response)
//   res.render('pages/profile', {img_src:req.file.path})
// })

app.post('/profile-upload-single', upload.single('profile-file'), function (req, res, next) {
  // req.file is the `profile-file` file
  // req.body will hold the text fields, if there were any
  console.log(req.file.path)
  // var response = '<a href="/">Home</a><br>'
  // response += "Files uploaded successfully.<br>"
  // response += `<img src="${req.file.path}" /><br>`
  // return res.send(response)
  res.render('pages/profile', {img_src})
})
```

Challenges With Axios

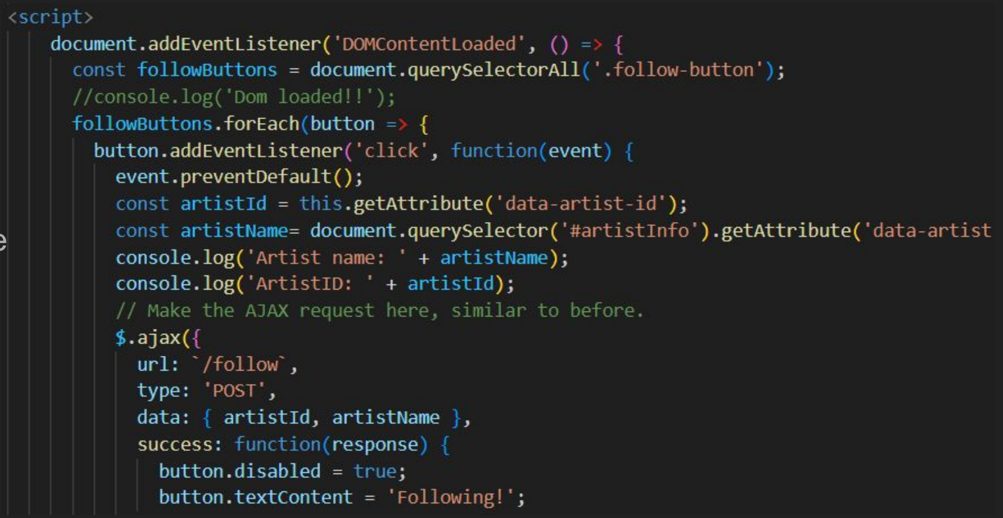
Generally easy to use, but:

- Lots of issues with types, had to typecast a lot
- Didn’t realize Axios automatically encrypted queries to HTML escape
- How to handle multiple axios requests

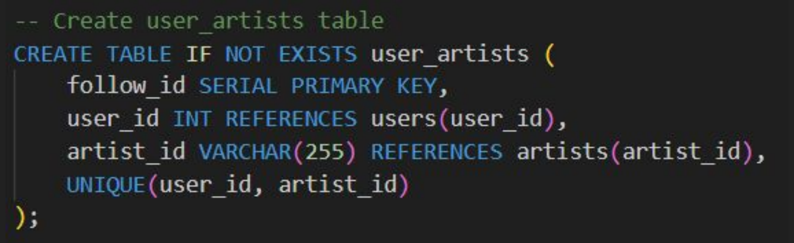
Challenges with Implementing Follow / Collections

- Implementing a following feature
- Artists Table in SQL
- Ajax vs Axios
  - AJAX: Allowed us to make request from the server without refreshing the browser in order to preserve state
- Updating a follow list

- Similar complications with adding an artwork to a collection

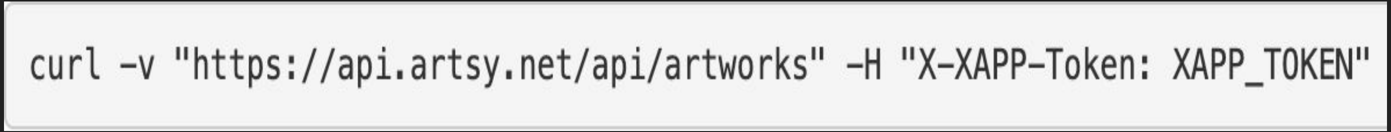


```
<script>
  document.addEventListener('DOMContentLoaded', () => {
    const followButtons = document.querySelectorAll('.follow-button');
    //console.log('Dom loaded!!');
    followButtons.forEach(button => {
      button.addEventListener('click', function(event) {
        event.preventDefault();
        const artistId = this.getAttribute('data-artist-id');
        const artistName= document.querySelector('#artistInfo').getAttribute('data-artist
        console.log('Artist name: ' + artistName);
        console.log('ArtistID: ' + artistId);
        // Make the AJAX request here, similar to before.
        $.ajax({
          url: `/follow`,
          type: 'POST',
          data: { artistId, artistName },
          success: function(response) {
            button.disabled = true;
            button.textContent = 'Following!';
```

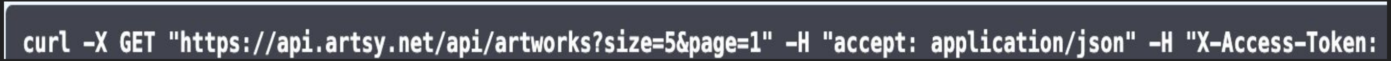


```
-- Create user_artists table
CREATE TABLE IF NOT EXISTS user_artists (
    follow_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES users(user_id),
    artist_id VARCHAR(255) REFERENCES artists(artist_id),
    UNIQUE(user_id, artist_id)
);
```

Challenges With APIs

- Artsy and Art Institute of Chicago
  - Usable offset values were not reflective of the total number of results
  - Artsy: artworks did not list the artist
  - Art Institute of Chicago: artists did not have an associated image
- Documentation
  - Inconsistent and unclear
    - Ex: Artsy request for artworks refers to the token in different manners:
      - 

```
curl -v "https://api.artsy.net/api/artworks" -H "X-XAPP-Token: XAPP_TOKEN"
```

      - 

```
curl -X GET "https://api.artsy.net/api/artworks?size=5&page=1" -H "accept: application/json" -H "X-Access-Token:
```

Challenges With Handlebars

- Wanted to be able to compare two strings
- Making a custom if didn’t work, only literals for second param
- Had to process in back end

Future Scope and Enhancements

- Infinite scrolling
- Unfollowing users + artists
- Username shown on comments
- User profile pictures
- Social aspect
- Admin editing
- Add user photos to collection

Demo!

Thank you for listening to our presentation.

Thank you!

-Team 5