



SkyShare

Chenghao Xiong, Mateusz Muszynski, Jennifer Kim, Genesse Miles



What is Skyshare?

A platform dedicated to showcasing ✨***the best space photos*** ✨ taken by people from all across the country!

Users can...



Upload their
best space
photos

with the **location** where it
was taken



Connect with
fellow space
enthusiasts

by **following** other
accounts



View NASA's
photos with a
simple click

conveniently accessible
via the nav bar

Tools we Used



Project Tracker: GitHub

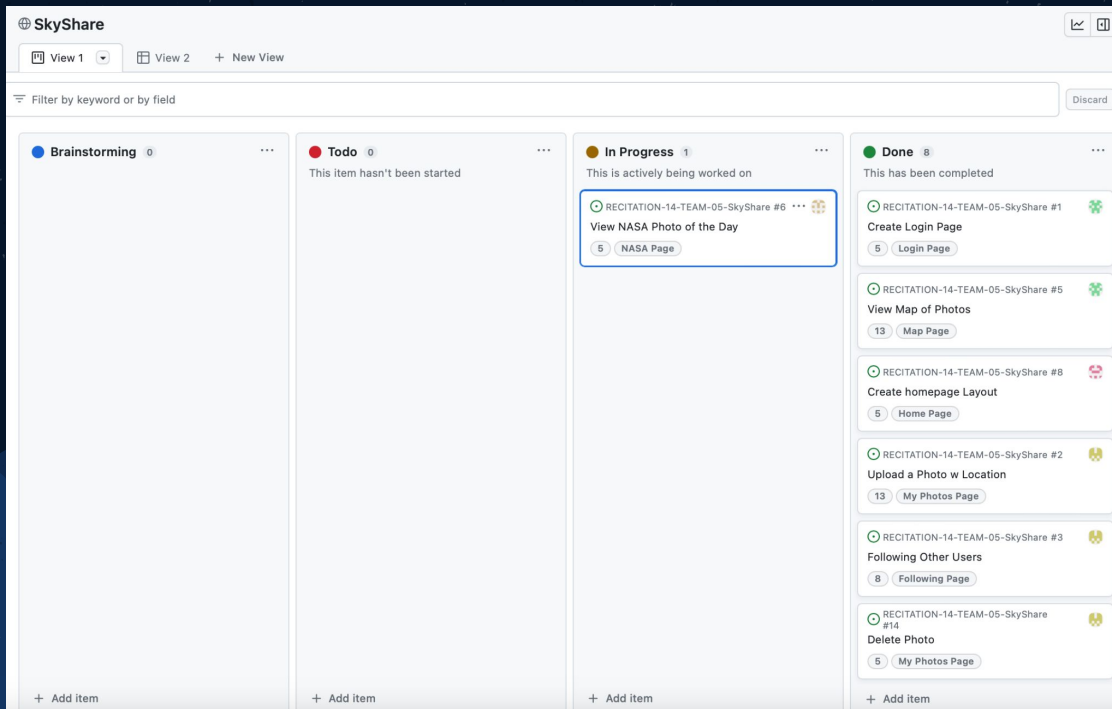


How useful was this?



4/5 stars; useful

- Clear & organized
- For our group of 4, helpful but not necessary



VCS repo: GitHub











How useful was this?



5/5 stars; extremely useful

- Very organized
- Great for collaboration and merging code
- Easy to track contributions and view past edits

 jenniferrkimm added lab13 screenshot		e03683e 2 days ago	 75 commits
 RECITATION-14-TEAM-05-SkySh...	Second time to submit code		5 days ago
 all project code&components	Homepage Beautification		3 days ago
 meeting_logs	added Apr 25th meeting mins		3 days ago
 milestoneSubmissions	added lab13 screenshot		2 days ago
 wireframes	wireframes		2 weeks ago
 README.md	updated readme w basic info		3 weeks ago

Database: PostgreSQL



How useful was this?



5/5 stars; extremely useful

```
DROP TABLE IF EXISTS users CASCADE;
CREATE TABLE IF NOT EXISTS users (
  user_id SERIAL PRIMARY KEY NOT NULL,
  username VARCHAR(50) unique,
  password CHAR(60) NOT NULL
);
```

```
DROP TABLE IF EXISTS followers CASCADE;
CREATE TABLE IF NOT EXISTS followers (
  user_id INT NOT NULL,
  following_id INT NOT NULL,
  PRIMARY KEY (user_id, following_id),
  FOREIGN KEY (user_id) REFERENCES users(user_id),
  FOREIGN KEY (following_id) REFERENCES users(user_id)
);
```

```
DROP TABLE IF EXISTS states CASCADE;
CREATE TABLE IF NOT EXISTS states (
  state_id SERIAL PRIMARY KEY NOT NULL,
  state_name VARCHAR(200),
  lng DECIMAL NOT NULL,
  lat DECIMAL NOT NULL
);
```

```
DROP TABLE IF EXISTS photos CASCADE;
CREATE TABLE IF NOT EXISTS photos (
  photo_id SERIAL PRIMARY KEY NOT NULL,
  user_id INT NOT NULL,
  photo_state VARCHAR(250),
  photo_description VARCHAR(200),
  photo_url VARCHAR(300) NOT NULL,
  FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

- Allowed us to:
 - store user info
 - query photos associated w/ each user
 - implement the following feature
 - display photos taken at x location on the map

IDE: VSCode



How useful was this?



5/5 stars; extremely useful

- Wrote all our code here
- Connected w GitHub repo



UI Tools: HTML, CSS, EJS

How useful was this?



4/5 stars; useful

- Each nav bar item is a page coded by an ejs file w/ **HTML & CSS** elements integrated
- **Bootstrap** for things like carousel & cards

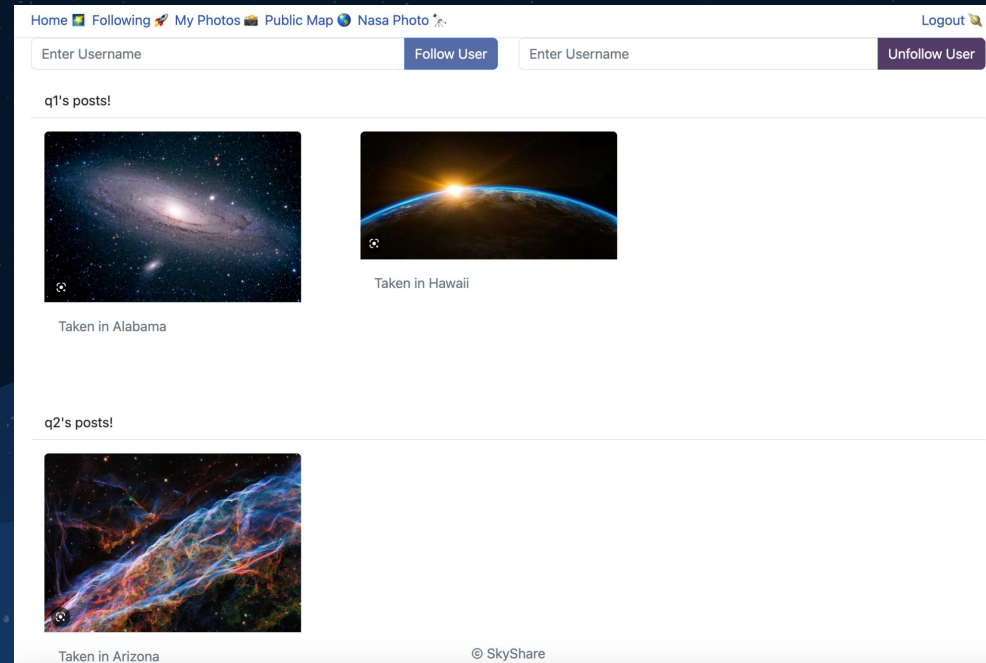
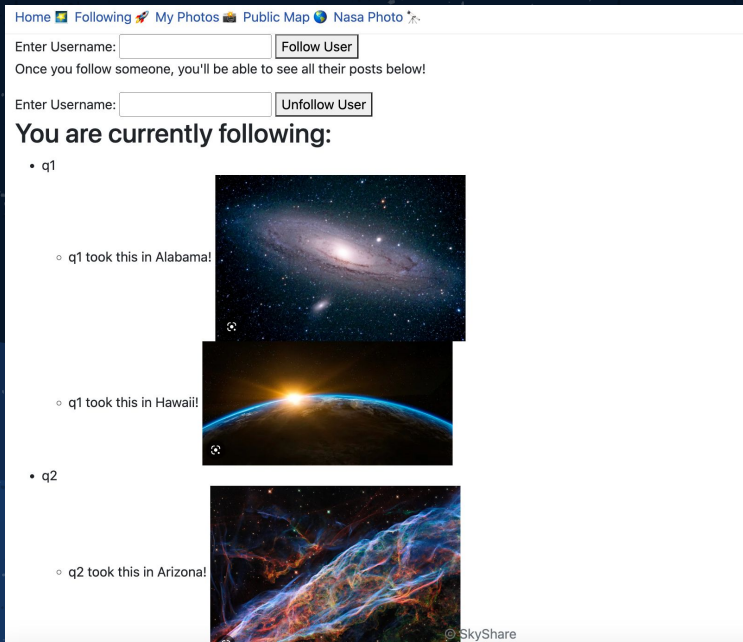
Our nav bar:

Home 🌟 Following 🚀 My Photos 📷 Public Map 🌐 Nasa Photo 🚀

UI Tools: HTML, CSS, EJS

Example: Bootstrap to generate a **card** for each followed users' posts

Before Bootstrap: After Bootstrap:



Ui Tools: HTML, CSS, EJS

Example: CSS to improve appearance of buttons on “My Photos” page


Before CSS:

Upload Photo


No file chosen

Where did you take this?

Your images!



You took this in Alabama!



You took this in Hawaii!


After CSS:

Upload Photo ✨


No file chosen

Where did you take this?

Your images!



You took this in Hawaii



You took this in Alabama

UiTools: HTML, CSS, EJS

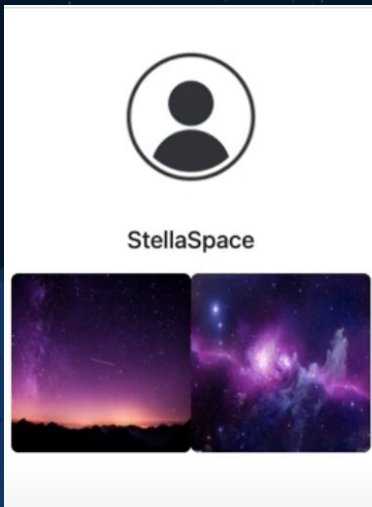
How useful was this?



5/5 stars; extremely useful

The **Bootstrap profile card** organizes the profile of the user

It displays the user's username and underneath, utilizing EJS, fetches the two most recent images the user has uploaded.



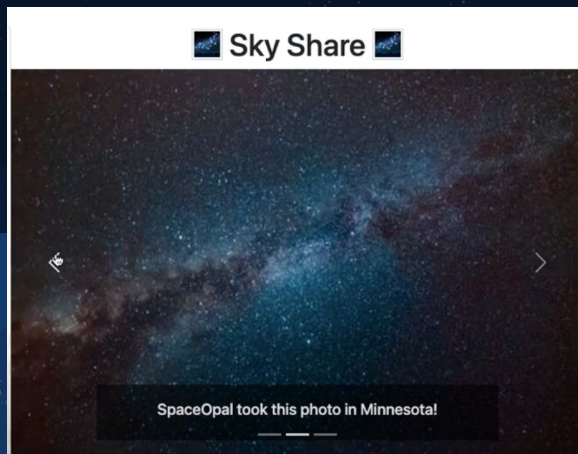
Ui Tools: HTML, CSS, EJS

How useful was this?



5/5 stars; extremely useful

For the homepage, utilizing the **Bootstrap Carousel** allowed for the implementation of a slideshow of random images which are uploaded from those who you follow through utilizing EJS.



App. Server: NodeJS



How useful was this?



5/5 stars; extremely useful

- APIs connecting to Postgres
- Basically what makes our platform's features function
- Each user request is associated with GET and POST API routes



Deployment environment: localhost & azure



How useful was this?



5/5 stars; extremely useful

- Allows users to access the website



★ External API #1: Cloudinary



How useful was this?



5/5 stars; extremely useful

Purpose:

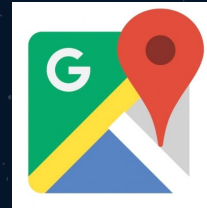
- Photo uploading + display photo on the page

How it works:

- Given a photo file, Cloudinary generates a url for it
- Store the url in photos table
- Display photo by passing in this url in the ejs file



External API #2: Google Maps



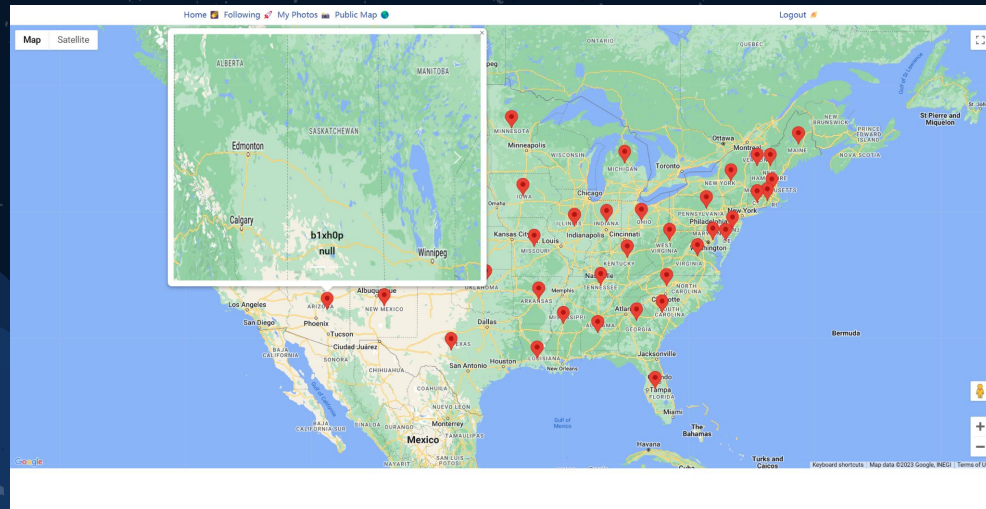
How useful was this?



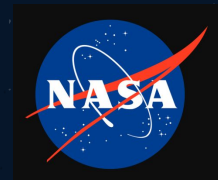
5/5 stars; extremely useful



You can click on the red marker to view photos of the location.



External API #3: NASA



How useful was this?



5/5 stars; extremely useful

- HD Images straight from NASA
- Generated Daily
- Detailed description



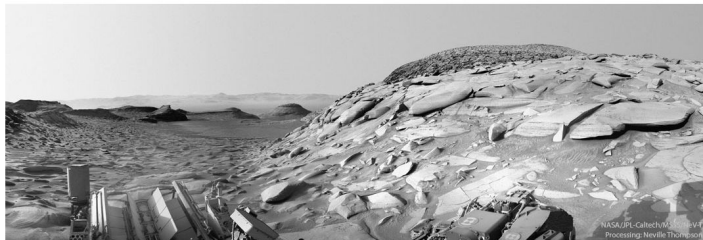
Home Following My Photos Public Map Nasa Photo

Logout

NASA Astronomy Picture of the Day

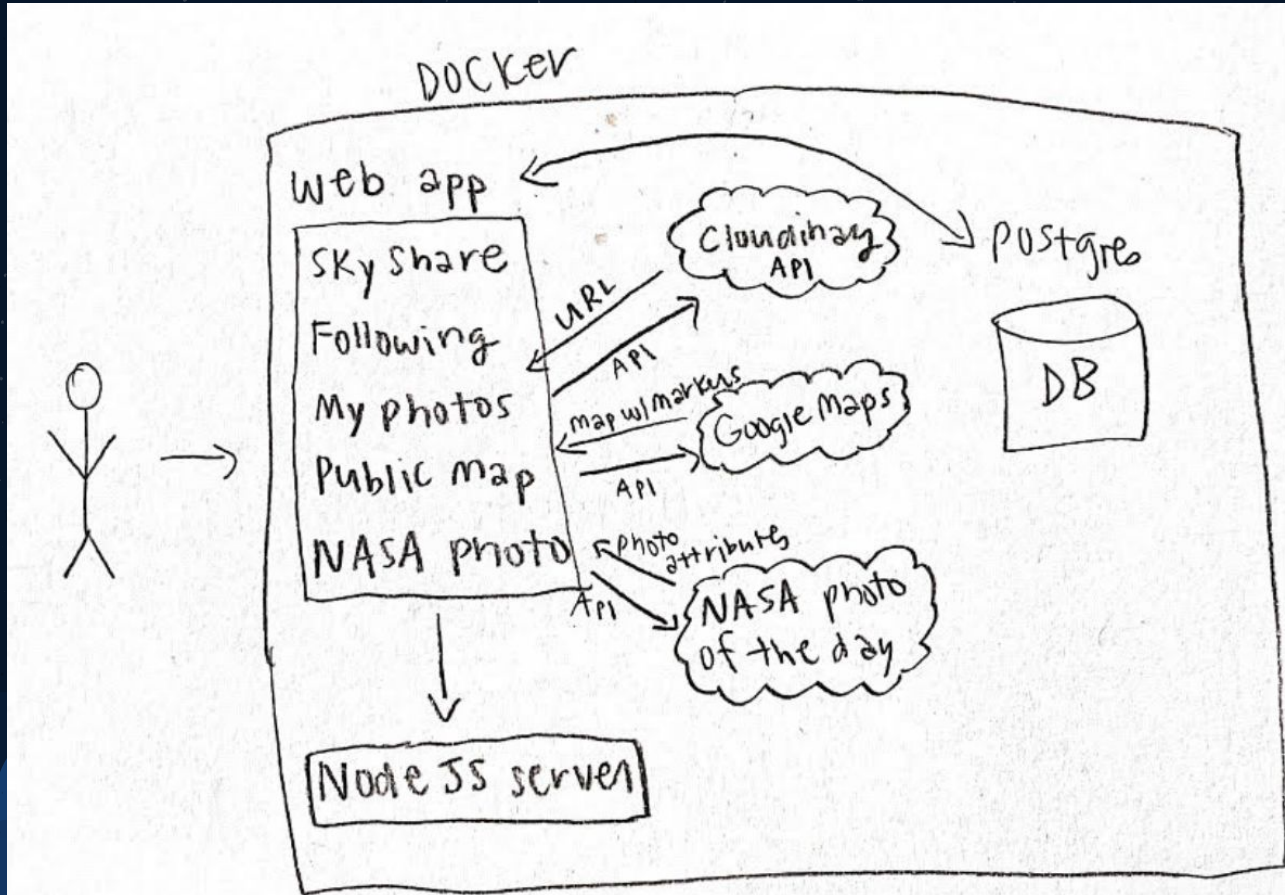
Flat Rock Hills on Mars

2023-05-02



Why are there so many flat rocks on Mars? Some views of plains and hills on Mars show many rocks that are unusually flat when compared to rocks on Earth. One reason for this is a process that is common to both Mars and Earth: erosion. The carbon-dioxide wind on Mars can act like sandpaper when it blows around gritty Martian sand. This sand can create differential erosion, smoothing over some rocks, while wearing down the tops of other long-exposed stones. The featured image capturing several hills covered with flat-topped rocks was taken last month by NASA's Curiosity Rover on Mars. This robotic rover has now been rolling across Mars for ten years and has helped uncover many details of the wet and windy past of Earth's planetary neighbor. After taking this and other images, Curiosity carefully navigated stones and slippery sand to climb up Marker Band Valley.

Architecture Diagram





Methodologies

✨ **Peer Code Reviews** (5/5 stars for usefulness):

- Before merging to main, other members looked over the code

✨ **Agile** (5/5 stars for usefulness):

- Broke up project into smaller tasks
- Worked on implementing small functional features in weekly phases
- Met every Wed/ Fri to discuss updates w the team, met TA every Tues

The background is a dark blue night sky filled with small white stars. There are three stylized dark blue clouds: one in the top left, one in the top right, and a large, dense cluster of overlapping circles at the bottom. A large, light blue full moon with darker blue spots is positioned on the right side. In the top right corner, there is a small planet with a ring system. The word "Challenge(s)" is written in a white, sans-serif font in the center-left area.

Challenge(s)

Google Maps longitude, latitude

Problem:

Google Maps API requires
longitude & latitude coordinates
to display a marker at that place

Solution:

Only ask user to input the state of
where the photo was taken.

In our sql file we inserted 50 values
of the long/lat of the middle of each
US state, so each state has 1
marker, and on this marker, we
display the photos taken in that
state

```
csci_3308 > RECITATION-14-TEAM-05-SkyShare > all project code&components > init_data > insert.sql
5  insert into states(state_name, lat, lng) Values('West Virginia', 39.000010, -80.500001);
6  insert into states(state_name, lat, lng) Values('Vermont', 44.000010, -72.699997);
7  insert into states(state_name, lat, lng) Values('Texas', 31.000010, -100.000010);
8  insert into states(state_name, lat, lng) Values('South Dakota', 44.500001, -100.000010);
9  insert into states(state_name, lat, lng) Values('Rhode Island', 41.742325, -71.742332);
10 insert into states(state_name, lat, lng) Values('Oregon', 44.000010, -120.500001);
11 insert into states(state_name, lat, lng) Values('New York', 43.000010, -75.000010);
12 insert into states(state_name, lat, lng) Values('New Hampshire', 44.000010, -71.500001);
13 insert into states(state_name, lat, lng) Values('Kansas', 38.500001, -98.000010);
14 insert into states(state_name, lat, lng) Values('Nebraska', 41.500001, -100.000010);
15 insert into states(state_name, lat, lng) Values('Mississippi', 33.000010, -90.000010);
16 insert into states(state_name, lat, lng) Values('Illinois', 40.000010, -89.000010);
17 insert into states(state_name, lat, lng) Values('Delaware', 39.000010, -75.500001);
18 insert into states(state_name, lat, lng) Values('Connecticut', 41.599998, -72.699997);
```


Demo!

