

Week8

Lab8

Armant Touche

Class/Instructor: CS430P/ Dr. Wu-Chang
Date: 11/21/22

Table of Contents

1. Lab7

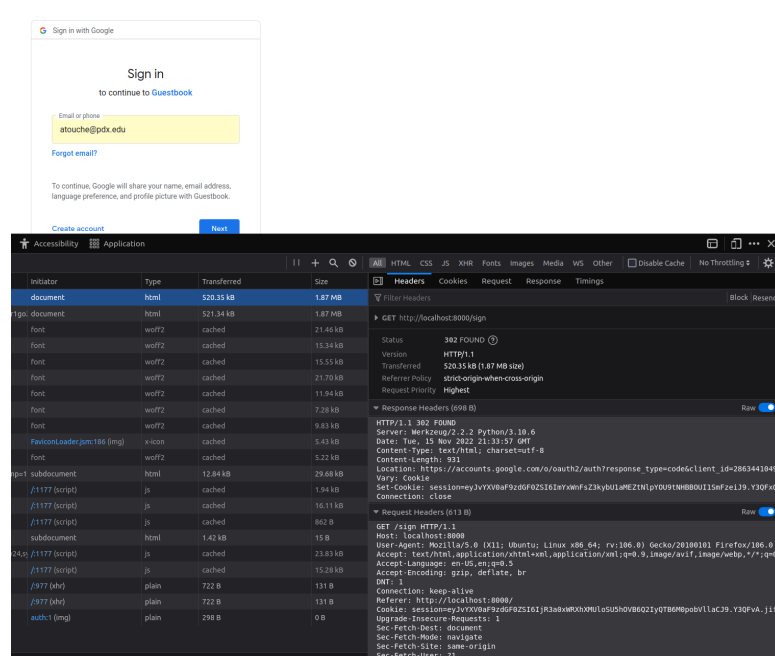
1.1. OAuth2

1.2. ML APIs

1.3. Firebase

OAuth2 Guestbook ([Link](#))

- ☐ OAuth2 Guestbook
- ☐ OAuth2 Authorization Code flow
- ☐ Set up Identity Provider
- ☐ -
- ☐ Checkout code
- ☐ OAuth initiation code
- ☐ Identity provider interaction
- ☐ Callback code
- ☐ -
- ☐ Signing page
- ☐ Model code
- ☐ Running the code
- ☐ **Take a screenshot of the Headers that includes the URL and the returned HTTP status code for each request for your lab notebook.**
 - /sign



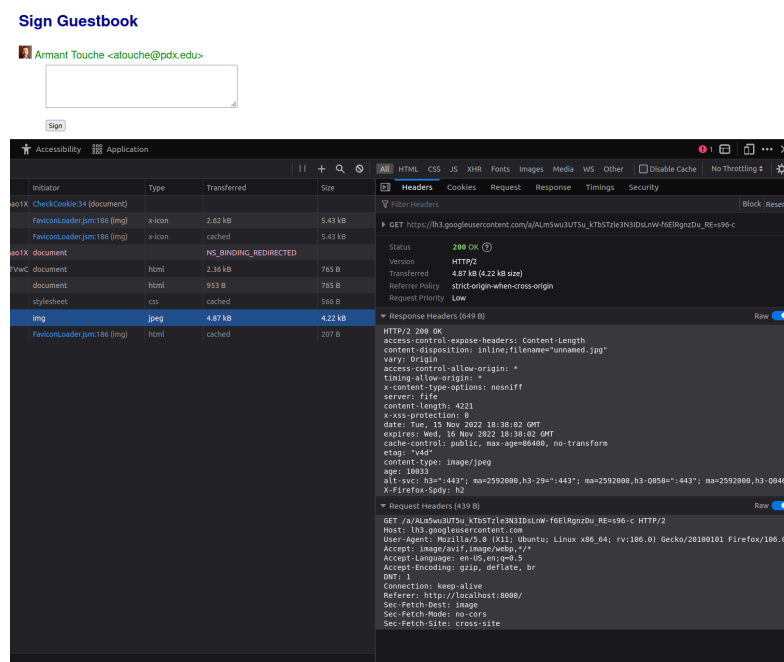
- Redirect img

[illegible]

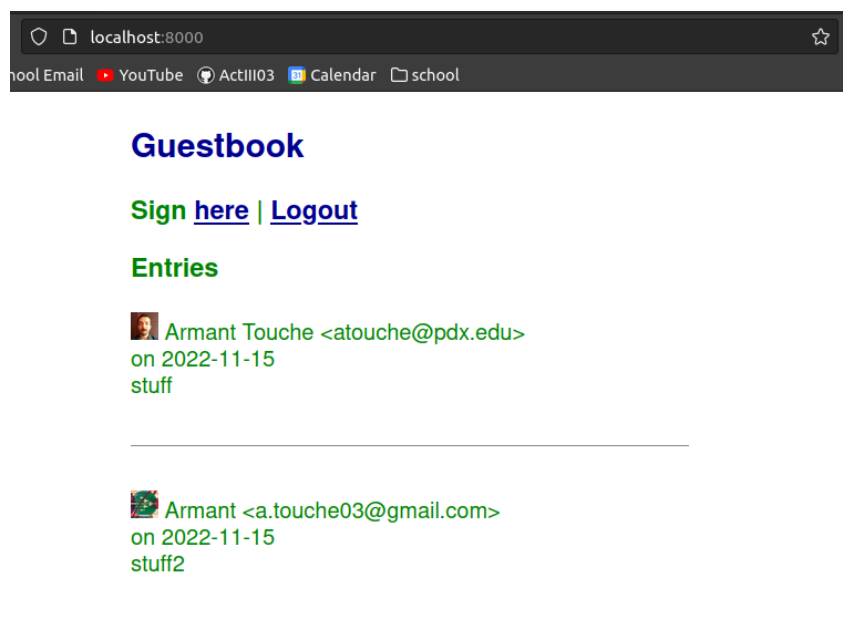
- ☐ **Based on the description of the source code, what lines of code in our application are responsible for the second request?**
 - `sign.py:15-28`
- ☐ **Take a screenshot of the Headers that includes the entire Callback URL and its returned HTTP status code. What location is the User sent to as a result of this request?**
 - Sent to callback along with an update to the state of the session.

[illegible]

- ☐ Find the request within Developer Tools that fetches the embedded image and take a screenshot of its URL.



- ☐ Take a screenshot showing multiple authenticated accounts have been able to sign the Guestbook.



ML APIs ([Link](#))

- ☐ APIs #1
- ☐ IAM service account setup
- ☐ Vision
- ☐ Show the output for your lab notebook

```
(env) atouche@cloudshell:~/Documents/lab8/python-vision/samples/snippets/detect (cloud-touche-atouche)$ python detect.py labels-uri gs://ml-api-codelab/birds.jpg
Labels:
Bird
Ratite
Cloud
Sky
Beak
Plant
Neck
Ostrich
Casuariiformes
Terrestrial animal
(env) atouche@cloudshell:~/Documents/lab8/python-vision/samples/snippets/detect (cloud-touche-atouche)$
```

- ☐ What is the name of the function?
 - def detect_labels_uri(uri)
- ☐ What type of Vision client is instantiated in it?
 - client = vision.ImageAnnotatorClient()
- ☐ What method is invoked in the Vision client to perform the detection?
 - response = client.label_detection(image=image)
- ☐ What is the name of the attribute in the response object that contains the results we seek?
 - description
- ☐ Take a screenshot of the output for the above commands

```
(env) atouche@cloudshell:~/.../snippets/detect (cloud-touche-atouche)$ python detect.py logos ./photo.jpg
Logos:
Portland State University
(env) atouche@cloudshell:~/.../snippets/detect (cloud-touche-atouche)$
```

- ☐ What method is invoked in the Vision client to perform the detection?
 - def detect_logos()
- ☐ Speech
- ☐ Show the output for your lab notebook

```
[notice] To update, run: pip install --upgrade pip
(env) atouche@cloudshell:~/.../samples/snippets (cloud-touche-atouche)$ python transcribe.py resources/audio.raw
Transcript: how old is the Brooklyn Bridge
(env) atouche@cloudshell:~/.../samples/snippets (cloud-touche-atouche)$
```

- ☐ What is the name of the function?
 - def transcribe_file()
- ☐ What method is invoked in the Speech client to perform the detection?
 - response = client.recognize(config=config, audio=audio)

- ☐ What is the name of the attribute in the response object that contains the results we seek?

- response.results.alternatives[0].transcript

- ☐ Translate

- ☐ Show the output for your lab notebook

```
(env) atouche@cloudshell:~/.../samples/snippets (cloud-touche-atouche)$ python snippets.py translate-text en '你有沒有帶外套'
Text: 你有沒有帶外套
Translation: do you have a coat
Detected source language: zh-TW
(env) atouche@cloudshell:~/.../samples/snippets (cloud-touche-atouche)$
```

- ☐ What is the name of the function?

- def translate_text()

- ☐ What method is invoked in the Translate client to perform the detection?

- result = translate_client.translate(text, target_language=target)

- ☐ What is the name of the attribute in the response object that contains the results we seek?

- Text: result["input"]

- Translation: result["translatedText"]

- Detected source language: result["detectedSourceLanguage"]

- ☐ Natural Language

- ☐ Show the output for your lab notebook

```
(env) atouche@cloudshell:~/.../samples/snippets (cloud-touche-atouche)$ ./analysis.sh
"homework is awful!" has sentiment=-0.800000011920929

Entities are:
name: homework
"homework is ok" has sentiment=0.30000001192092896

Entities are:
name: homework
"homework is awesome?" has sentiment=0.4000000059604645

Entities are:
name: homework
"homework is awesome!" has sentiment=0.8999999761581421

Entities are:
name: homework
"The protestors in Oregon put on gas masks and wore yellow t-shirts" has sentiment=-0.6000000238418579

Entities are:
name: protestors
name: gas masks
name: Oregon
name: t-shirts
(env) atouche@cloudshell:~/.../samples/snippets (cloud-touche-atouche)$
```

- ☐ Integration

☐ Code

- ☐ **What is the name of the function that performs the transcription?**
 - transcription = transcribe(language, audio)
- ☐ **What is the name of the function that performs the translation?**
 - translation = translate_text('en', transcription)
- ☐ **What is the name of the function that performs the entity analysis on the translation?**
 - entities = entities_text(translation)
- ☐ **What is the name of the function that performs the entity analysis on the image?**
 - labels = detect_labels_uri(image)

☐ Test integration

- ☐ **If the program deems them unrelated, then based on the results from the APIs, what must be changed in the program to address this?**
 - At solution.py:135 (if entity in labels:), this condition does account for case sensitivity. 'soccer' == 'Soccer' —> False but 'Soccer' == 'Soccer' —> True. Change the condition to account for case sensitivity differences since Python is explicit in casing.
- ☐ **If the program deems them unrelated, then based on the results from the APIs, what must be changed in the program to address this?**
 - At solution.py:129, if there is an option, generate more labels that are synonymous w/ the entities in the image. Bike is synonymous with Bicycle and are related but since Bike is not present in labels, the program addresses it as unrelated
- ☐ **If the program deems them unrelated, then based on the results from the APIs, what must be changed in the program to address this?**
 - At this point, it is apparent that synonyms, case sensitivity, and in this 3rd case, plurality are not intended or addressed in this program. ostriches is related to Ostrich.

☐ APIs #2

☐ Video setup

☐ Video Intelligence labeling script

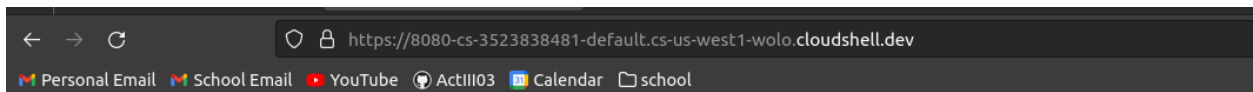
☐ Video Intelligence

- ☐ **What are the top 3 labels that the Video Intelligence API associates with the video and what is its confidence in them?**
 1. sports w/ confidence=0.92188
 2. basketball w/ confidence=0.91378
 3. player w/ confidence=0.84465
- ☐ **What is the name of the client class in the package that is used?**
 - videointelligence from google.cloud

- ☐ What method is used in that class to perform the annotation?
 - operation = video_client.annotate_video(input_uri=path, features=features)

- ☐ APIs #3
- ☐ IAM service account setup
- ☐ Application

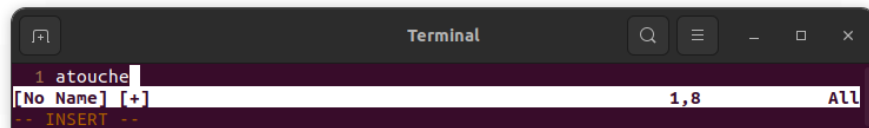
- ☐ Take a screenshot for your lab notebook that includes the URL.



Google Cloud Platform - Face Detection Sample

This Python Flask application demonstrates App Engine Flexible, Google Cloud Storage, Datastore, and the

Upload File: No file selected.



profile-pic.jpg was uploaded 2022-11-16 23:24:28.973769+00:00.

Joy Likelihood for Face: Unlikely

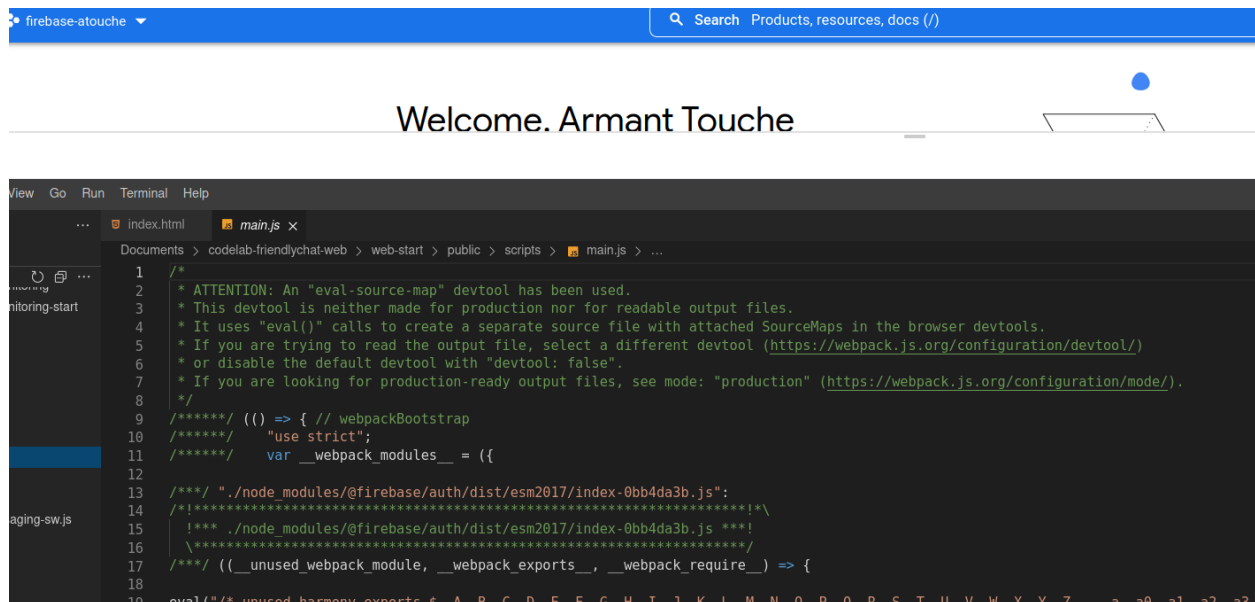
- ☐ Code
 - ☐ main.py
 - ☐ What line of code creates the query for previous detections?
 - main.py:39
 - ☐ What line of code sends the query to Cloud Datastore?
 - main.py:40 → query.fetch()
 - ☐ Upload route
 - ☐ Show the line that retrieves the name of the storage bucket to use.
 - Main.py: CLOUD_STORAGE_BUCKET = os.environ.get("CLOUD_STORAGE_BUCKET")

- ☐ **What form field is used to specify the uploaded photo?**
 - `request.files["file"]`
- ☐ **Show the line that copies the photo's contents to the storage bucket.**
 - `main.py:113`
- ☐ **What method in Vision's annotation client is used to perform the analysis?**
 - `faces = vision_client.face_detection(image=image).face_annotations`
- ☐ **What fields are stored in Cloud Datastore for each image?**
 - `blob_name`
 - `image_public_url`
 - `timestamp`
 - `joy`
- ☐ **What happens at the end of the `upload_photo` route?**
 - redirect to `/` path in flask application occurs

☐ Clean up

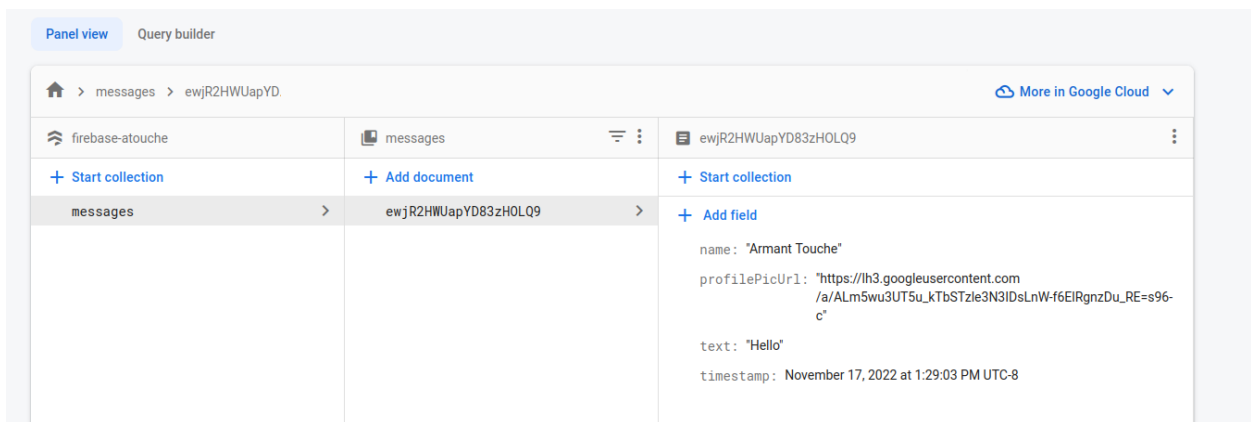
Firestore (Link)

- ☐ Firestore web application
- ☐ Project setup
- ☐ Application setup
- ☐ Authentication setup
 - ☐ What other domains are given access to this Firestore project by default?
 - localhost
 - fir-atouche.firebaseio.com
 - fir-atouche.firebaseio.web.app
- ☐ Database setup
- ☐ Storage setup
- ☐ CLI setup
- ☐ Bundling with Webpack
 - ☐ Take a screenshot of the first 10 lines of the produced file.

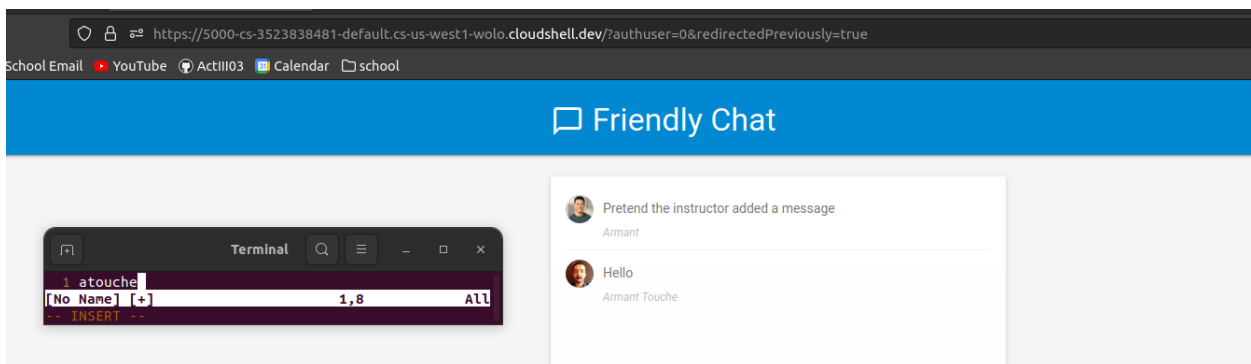


- ☐ Configure Firestore within application
- ☐ Initialize Firestore within application
- ☐ Test application
- ☐ Add authentication
 - ☐ What missing functions deal with user authentication?
 - signIn()
 - ☐ What missing functions deal with sending and receiving messages?
 - onMessageFormSubmit() calls saveMessage() to send

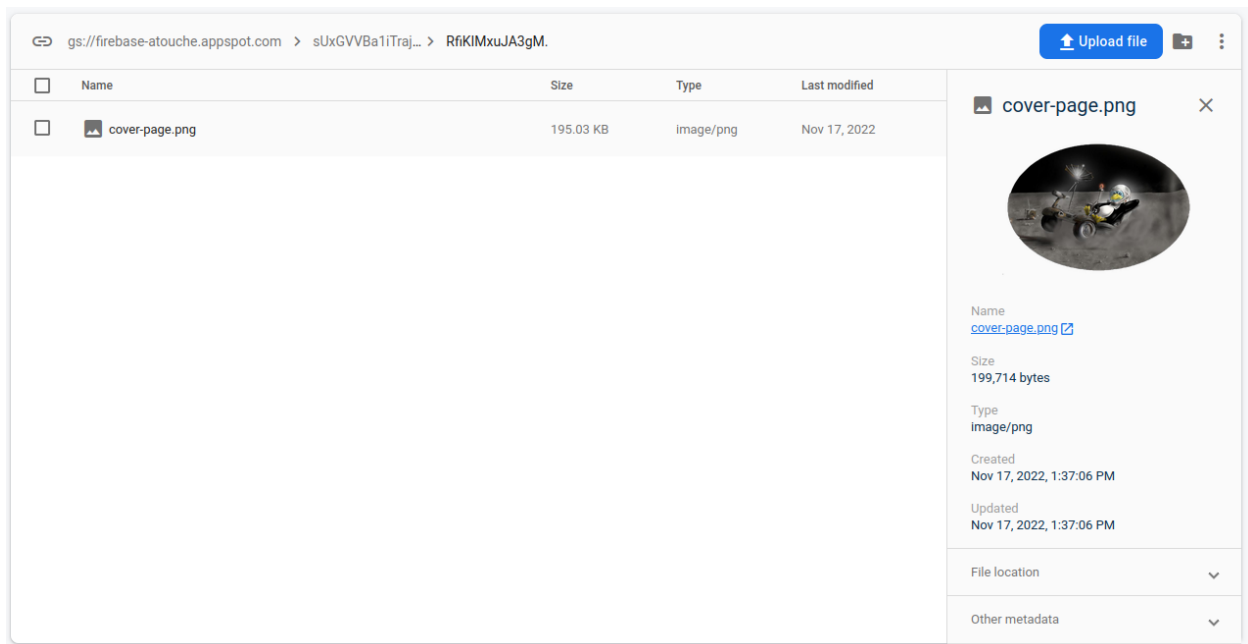
- loadMessages() loads/receives messages and listens for upcoming ones
- ☐ Update UI
 - ☐ **What are the names of the elements that are hidden when the user is signed out?**
 - userNameElement
 - userPicElement
 - signOutButtonElement
 - ☐ **What is the name of the element that is not hidden when the user is signed out?**
 - signInButtonElement has 'hidden' removed in order to allow user to sign-in again
- ☐ Test application with authentication
- ☐ Add text messaging
- ☐ Test application with text messaging
 - ☐ **Include a screenshot of the message and its fields in the database for your lab notebook**



- ☐ Manual message insertion
 - ☐ **Include a screenshot of the application with its two messages for your lab notebook**

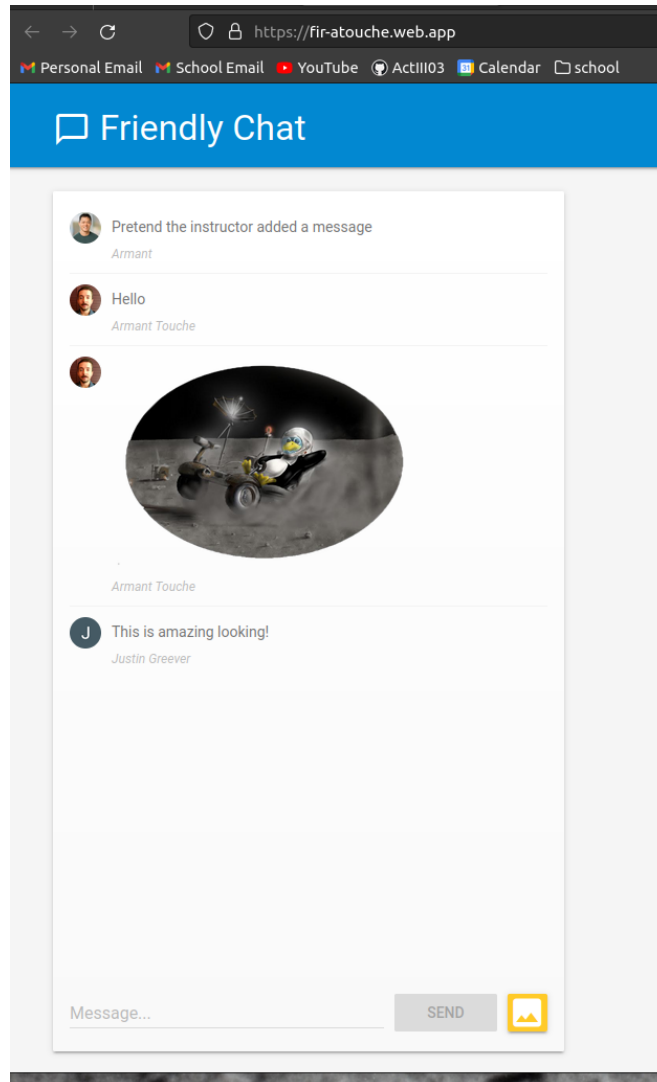


- ☐ Add image messaging
 - ☐ **What is the URL of the image that is first shown in the UI as the message is loading?**
 - `LOADING_IMAGE_URL = 'https://www.google.com/images/spin-32.gif?a'`
- ☐ Test application with image messaging
 - ☐ **How do the fields in an image document differ from that of the text document?**
 - `imageUrl` and `storageUri` are present vs prior messages
 - ☐ **What URL and storage location can the image be found at?**
 - ☐ `imageUrl =`
<https://firebasestorage.googleapis.com/v0/b/firebase-atouche.appspot.com/o/sUxGVVBa1iTrajuSlYe6reAdJQI3%2FRfiKIMxuJA3gM4Uma7a1%2Fcover-page.png?alt=media&token=ca8c9e49-de27-499e-a26b-80fdca44972f>
 - ☐ `storageUri = sUxGVVBa1iTrajuSlYe6reAdJQI3`
 - ☐ **Take a screenshot of the image in the storage bucket for your lab notebook.**



- ☐ Deploy application
 - ☐ **What directory is the application going to be served from?**
 - ☐ `"public": "./public"`

- ☐ What does the Cache-Control setting configured for the HTML and Javascript files do?
 - max-age is 0 seconds and Cache-Control ensures browser must refresh its version of the resource (./public/*) by sending another request to a server where response is *more* recent version of the public resources.
- ☐ Take a screenshot of the message including the URL for your lab notebook.



- Response from Justin G. (classmate)
- ☐ Clean up