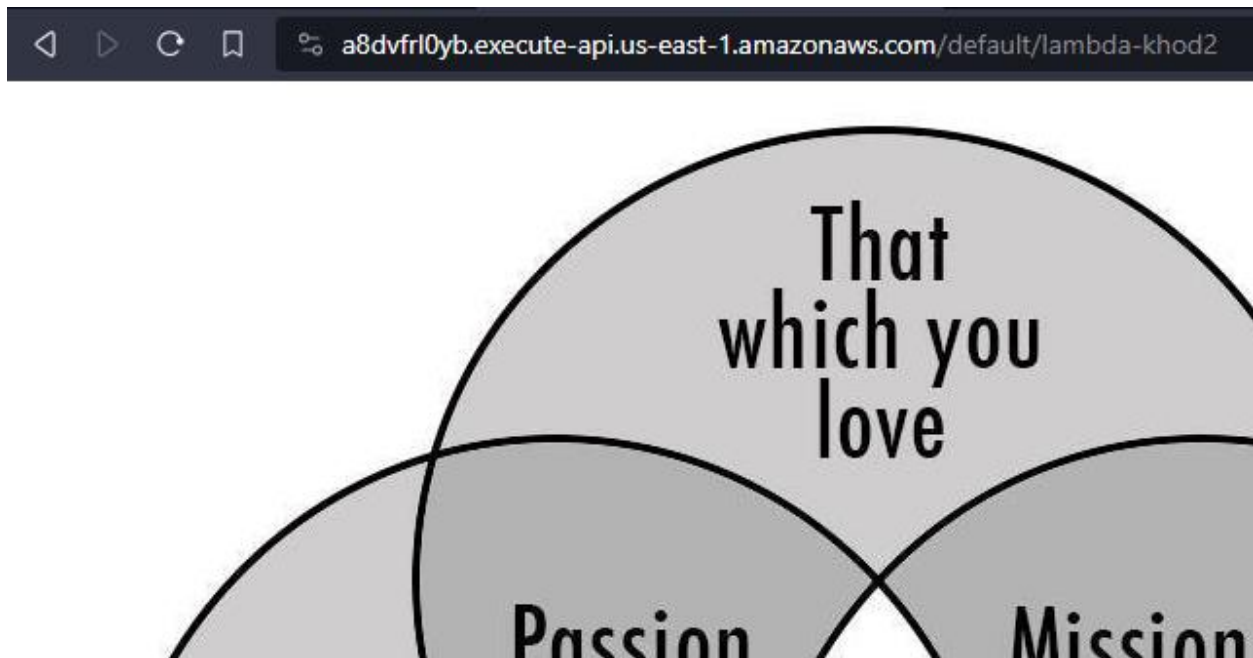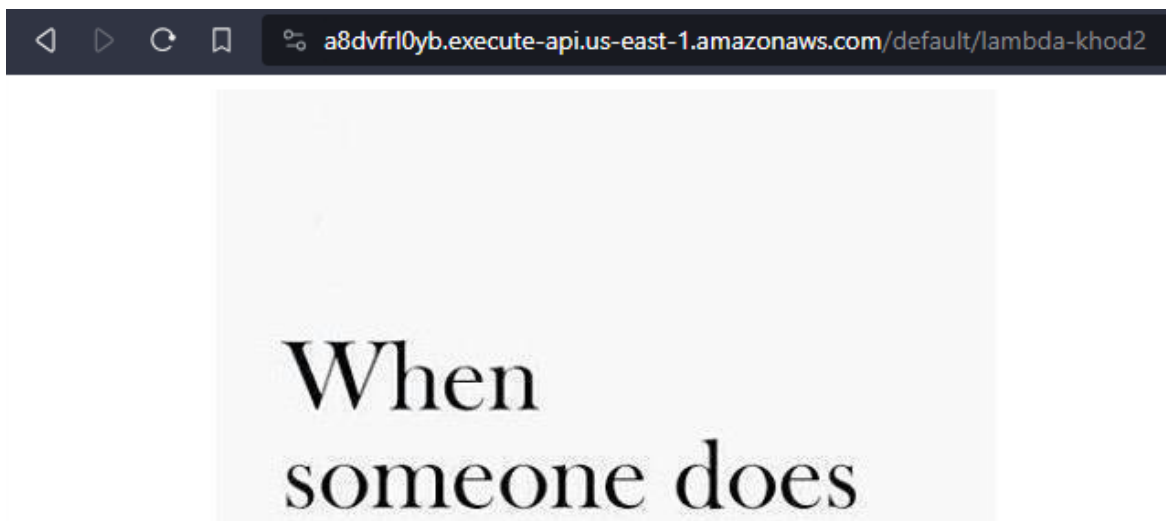Lab 8 Notebook

8.1a

4. **Take a screenshot of the resulting page including the URL bar**.

Click "Reload" in the browser and take another screenshot showing the image has changed:

8. **Use `curl` on your Linux VM to access the API endpoint and show the results. Take a screenshot for your lab notebook.**

```
khod2@SamPC:~$ curl https://2gfzcajhh4.execute-api.us-east-1.amazonaws.com/default/gettime-khod2
{"currentTime": "2023-11-15 07:13:19.345362"}khod2@SamPC:~$
```

8.2a

10. **Take a screenshot that shows that you can view the entries in the backend database.**

Documents/MAIN/Github%20repos/CS430/cs430-src/06_aws_restapi_lambda/frontend/src/index.html

Guestbook

Name:

Email:

Message:

Sign

Entries

Sam <khod2@pdx.edu>
signed on 2023-10-25 18:04:46.304495
Hello DynamoDB!

Sam <khod2@pdx.edu>
signed on 2023-10-26 01:21:27.851345
Hello Docker DynamoDB!

**12. Take a screenshot showing that the submission worked.**

/entry - POST method test results

Request	Latency	Status
/entry	2389	200

Response body

[{"message": "Hello DynamoDB!", "date": "2023-10-25 18:04:46.304495", "email": "khod2@pdx.edu", "name": "Sam"}, {"message": "Hello Docker DynamoDB!", "date": "2023-10-26 01:21:27.851345", "email": "khod2@pdx.edu", "name": "Sam"}, {"message": "Hello Cloud9!", "date": "2023-10-26 01:44:21.534043", "email": "khod2@pdx.edu", "name": "Sam"}, {"message": "Hello EC2!", "date": "2023-10-26 02:39:29.710922", "email": "khod2@pdx.edu", "name": "Sam"}, {"message": "Hello Elastic Beanstalk!", "date": "2023-11-02 00:50:47.486585", "email": "khod2@pdx.edu", "name": "Sam"}, {"message": "Hello ECS!", "date": "2023-11-03 07:10:57.041588", "email": "khod2@pdx.edu", "name": "Sam"}, {"message": "Hello API Gateway!", "date": "2023-11-15 08:47:35.918529", "email": "khod2@pdx.edu", "name": "Sam"}]

Take a screenshot as before that shows your entry and the static website hosting URL.

⚠ Not secure | khod2-frontend.s3-website-us-east-1.amazonaws.com
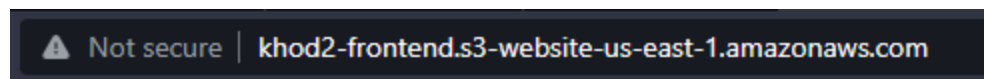
Hello API Gateway!

Sam <khod2@pdx.edu>
signed on 2023-11-15 08:58:07.489464
Hi API Gateway from local HTML file!

Sam <khod2@pdx.edu>
signed on 2023-11-15 09:18:59.810589
Hello S3, API Gateway and Lambda!

3

8.2g

10. Include the hostname for the API gateway in your lab notebook.

gbapigw-dpychv45.uc.gateway.dev

11. Take a screenshot of the loop and its output.

```
>>> for guest in guestbook_json:\
... print(guest['name'], guest['email'], guest['date'], guest['message'])
...
Sam khod2@pdx.edu 2023-10-27 07:47:32.579962+00:00 Hello Compute Engine!
Sam khod2@pdx.edu 2023-10-26 22:27:02.518371+00:00 Hello Datastore!
Sam khod2@pdx.edu 2023-11-09 08:12:56.989038+00:00 Hello Cloud Build!
Sam khod2@pdx.edu 2023-11-03 07:33:54.219216+00:00 Hello Cloud Run!
Sam khod2@pdx.edu 2023-11-09 07:13:04.358324+00:00 Hello Kubernetes!
Sam khod2@pdx.edu 2023-11-02 01:25:17.054584+00:00 Hello App Engine!
Sam khod2@pdx.edu 2023-10-27 07:10:17.800231+00:00 Hello Docker Datastore!
Sam khod2@pdx.edu 2023-10-27 07:35:45.303855+00:00 Hello Cloud Shell!
>>>
```
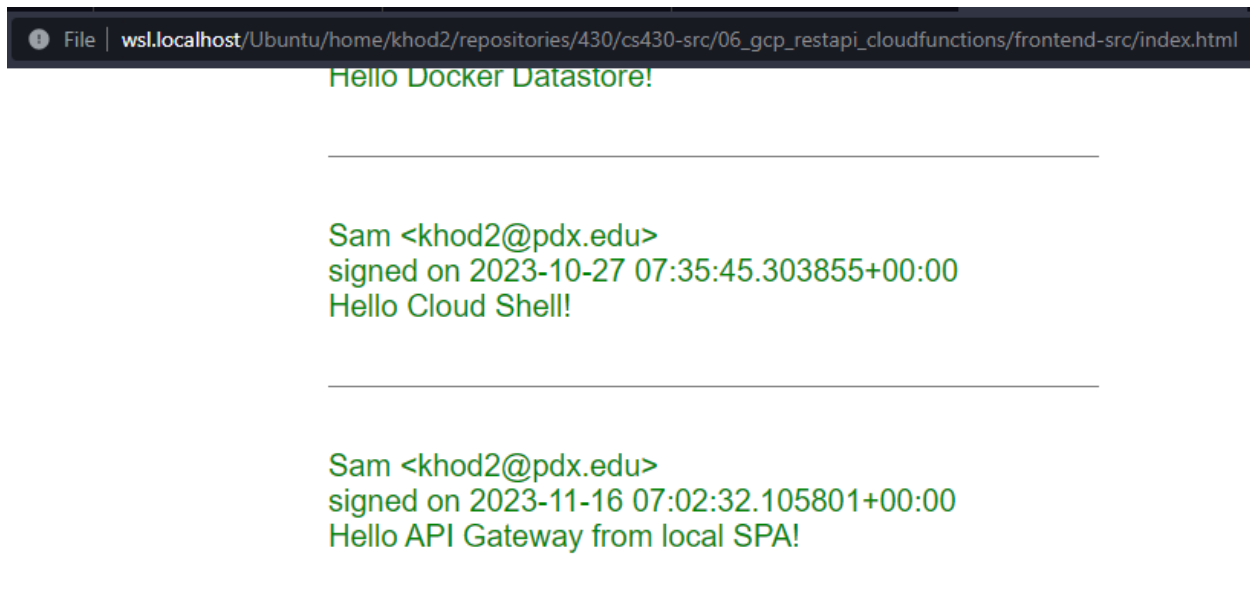
12. Take a screenshot of the output for your lab notebook.

```
>>> print(resp.status_code)
200
>>> print(resp.headers)
{'content-type': 'application/json', 'access-control-allow-origin': '*', 'function-execution-id': 'nq6v188gzyw0', 'x-cloud-trace-con
text': 'd871b2cd1c190b79caa76b7970fe8c0c;o=1', 'alt-svc': 'h3=":443"; ma=2592000,h3-29=":443"; ma=2592000', 'Date': 'Thu, 16 Nov 202
3 06:29:45 GMT', 'Server': 'Google Frontend', 'Content-Length': '1103'}
>>> print(resp.text)
[{"name": "Sam", "email": "khod2@pdx.edu", "date": "2023-10-27 07:47:32.579962+00:00", "message": "Hello Compute Engine!"}, {"name":
 "Sam", "email": "khod2@pdx.edu", "date": "2023-10-26 22:27:02.518371+00:00", "message": "Hello Datastore!"}, {"name": "Sam", "email
": "khod2@pdx.edu", "date": "2023-11-09 08:12:56.989038+00:00", "message": "Hello Cloud Build!"}, {"name": "Sam", "email": "khod2@pd
x.edu", "date": "2023-11-03 07:33:54.219216+00:00", "message": "Hello Cloud Run!"}, {"name": "Sam", "email": "khod2@pdx.edu", "date"
: "2023-11-09 07:13:04.358324+00:00", "message": "Hello Kubernetes!"}, {"name": "Sam", "email": "khod2@pdx.edu", "date": "2023-11-02
 01:25:17.054584+00:00", "message": "Hello App Engine!"}, {"name": "Sam", "email": "khod2@pdx.edu", "date": "2023-11-16 06:29:45.530
223+00:00", "message": "Hi API Gateway with python Interpreter!"}, {"name": "Sam", "email": "khod2@pdx.edu", "date": "2023-10-27 07:
10:17.800231+00:00", "message": "Hello Docker Datastore!"}, {"name": "Sam", "email": "khod2@pdx.edu", "date": "2023-10-27 07:35:45.3
03855+00:00", "message": "Hello Cloud Shell!"}]
```
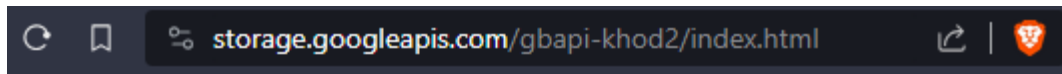
15. Take a screenshot showing the preflight request to the API that allows API access, as well as the subsequent fetch request have been successful.

Name	Status	Protocol	Type	Initiator
KHOD2				
index.html	200	file	document	Other
style.css	200	file	stylesheet	index.html:5
guestbook.js	200	file	script	index.html:6
entries	200	h3	fetch	guestbook.js:68
entries	200	h3	preflight	Preflight

Take a screenshot of the Guestbook including the URL.

File | wsl.**localhost**/Ubuntu/home/khod2/repositories/430/cs430-src/06_gcp_restapi_cloudfunctions/frontend-src/index.html

Hello Docker Datastore!

_____

Sam <khod2@pdx.edu>
signed on 2023-10-27 07:35:45.303855+00:00
Hello Cloud Shell!

_____

Sam <khod2@pdx.edu>
signed on 2023-11-16 07:02:32.105801+00:00
Hello API Gateway from local SPA!

_____

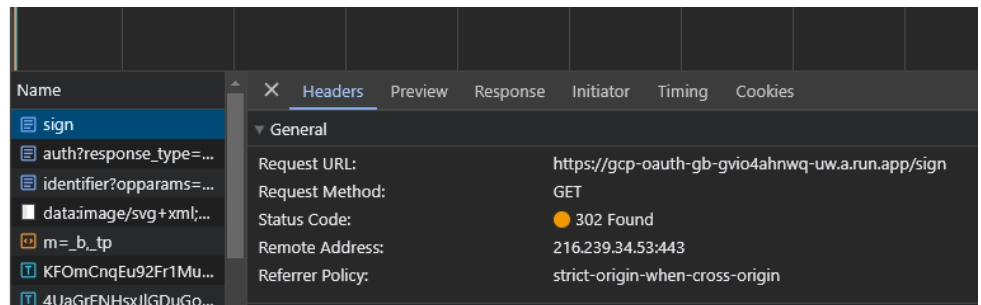**16. Take a screenshot of the Guestbook including the URL.**

8.3g

14. **Take a screenshot of the Headers that includes the URL and the returned HTTP status code for the first two requests for your lab notebook.**

Based on the description of the source code, what lines of code in our application are responsible for the second request shown?

I'm guessing this second request is the request to Google's server to provide an authentication URL, after the initial request for the sign page. The lines of code responsible would be 18-23 (for just the url request).

Take a screenshot of the permissions you (as a user) are granting the Guestbook access to.

Sign in

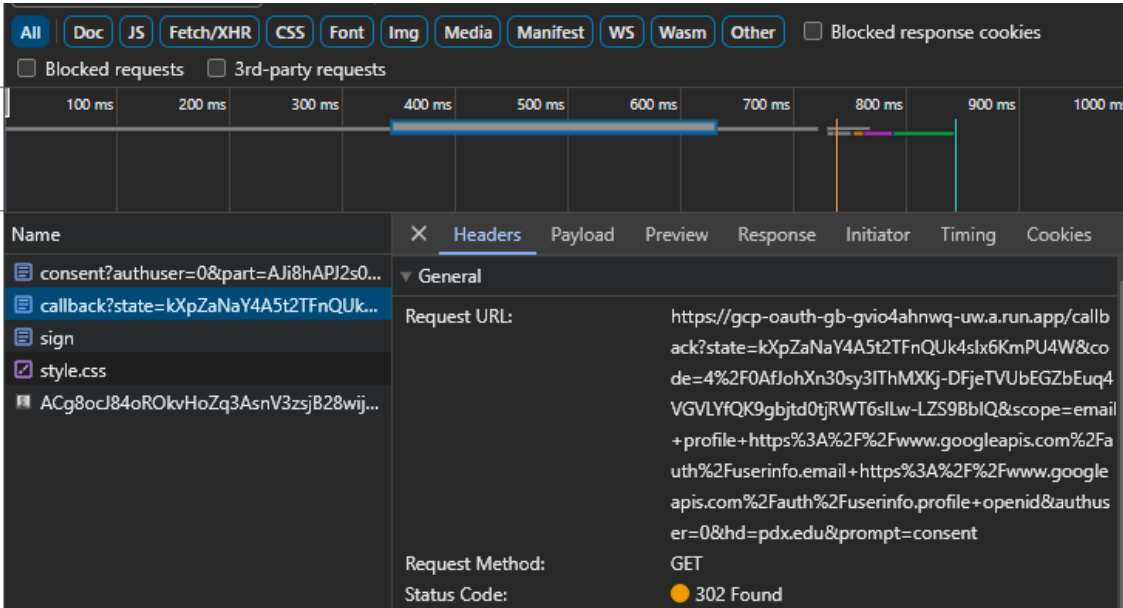to continue to

gcp-oauth-gb-gvio4ahnwq-uw.a.run.app

Email or phone

khod2@pdx.edu

Forgot email?

To continue, Google will share your name, email address, language preference, and profile picture with gcp-oauth-gb-gvio4ahnwq-uw.a.run.app.

Take a screenshot of the Headers that includes the entire Callback URL and its returned HTTP status code. What is the URI for the Location that the User sent to by the Callback?

<khod2@pdx.edu>

Request URL:	https://gcp-oauth-gb-gvio4ahnwq-uw.a.run.app/callback?state=kXpZaNaY4A5t2TFnQUk4slx6KmPU4W&code=4%2F0AfJohXn30sy3IThMXKj-DFjeTVUbEGZbEuq4VGVLYfQK9gbjtd0tjRWT6sILw-LZS9BbIQ&scope=email+profile+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.profile+openid&authuser=0&hd=pdx.edu&prompt=consent
Request Method:	GET
Status Code:	● 302 Found

The URI in the callback request's location is /sign. This is where the user is sent to after the callback request is completed.

Find the request within Developer Tools that fetches the embedded image and take a screenshot of its URL.
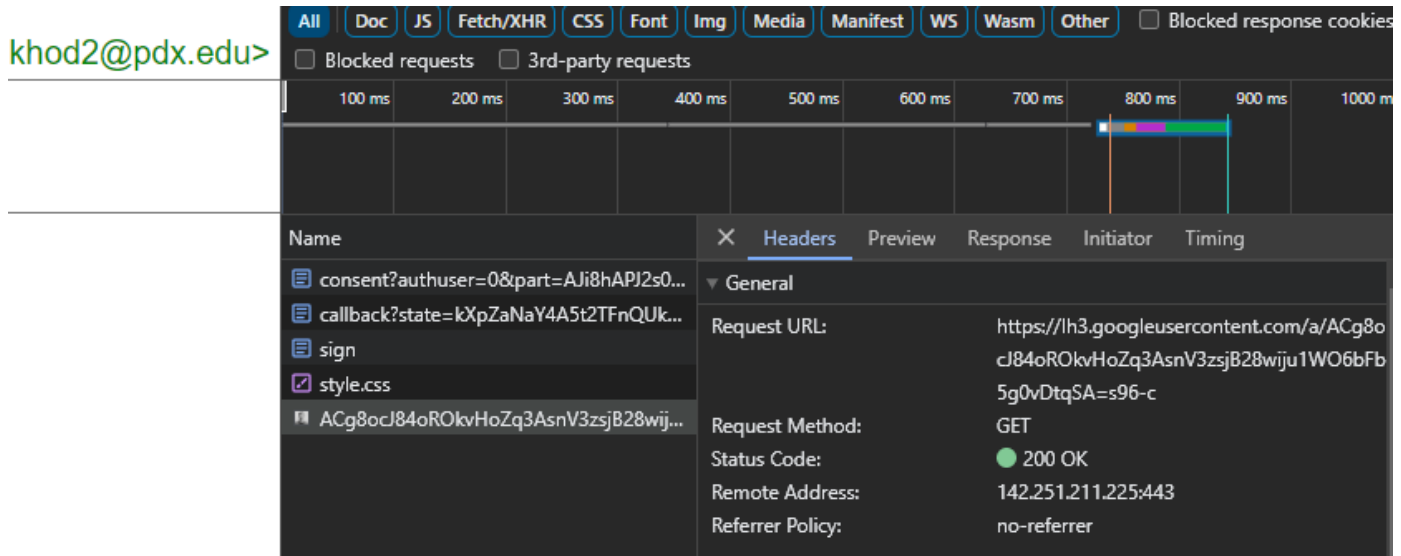
15. **Take a screenshot showing multiple authenticated accounts have been able to sign the Guestbook.**

16. Take a screenshot of the expanded information that includes your OdinId for your lab notebook.

The expanded info doesn't include your odinID anymore, so I hope this works.

8.4g

4. **What other domains are given access to this Firebase project by default?**

```
localhost
fir-khod2.firebaseapp.com
fir-khod2.web.app
```

8. **Take a screenshot of the first 10 lines of the produced file.**

12. **What missing functions deal with user authentication?**

signIn(), signOutUser(), initFirebaseAuth(). The next following are login functions.

What missing functions deal with sending and receiving messages?

loadMessages(), saveMessage(), saveImageMessage(), saveMessagingDeviceToken().
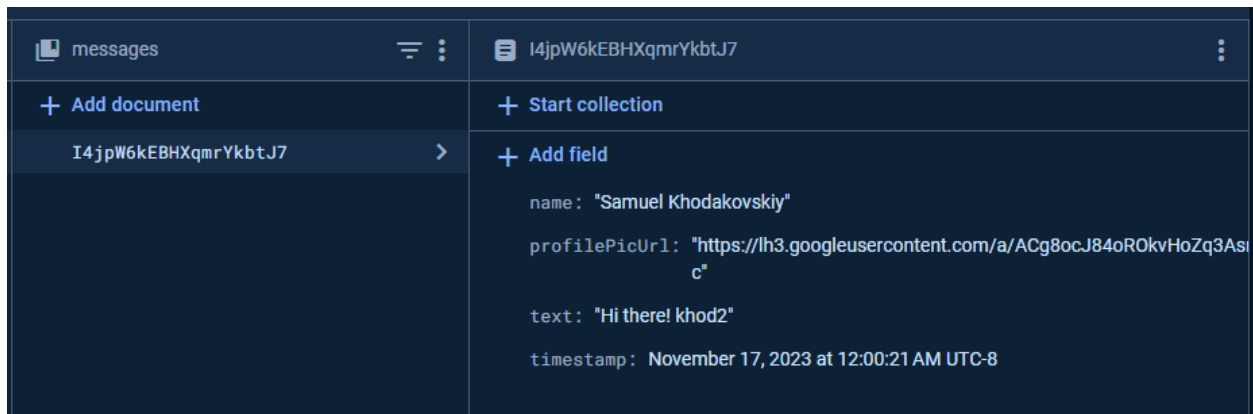
13. What are the names of the elements that are hidden when the user is signed out?

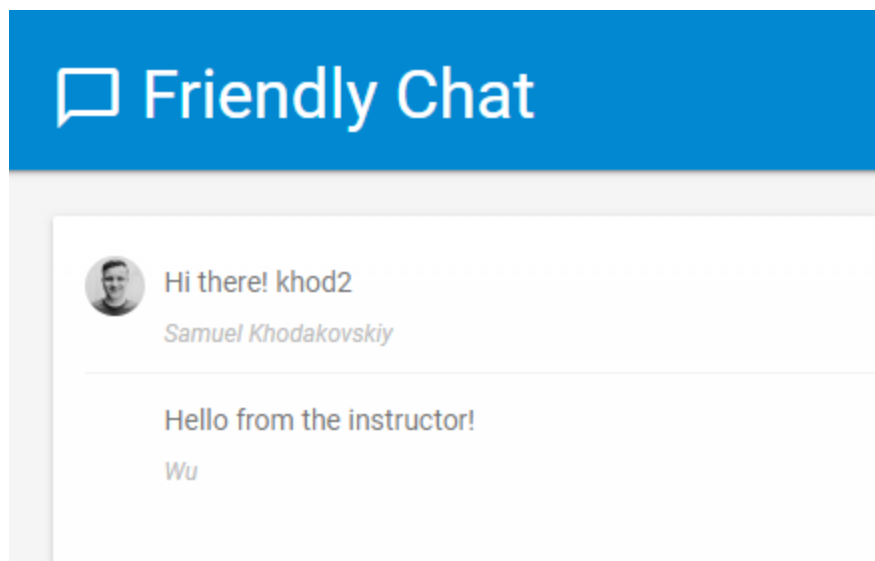userNameElement, userPicElement, signOutButtonElement.

What is the name of the element that is not hidden when the user is signed out?

signInButtonElement.

16. Include a screenshot of the message and its fields in the database for your lab notebook.

17. Include a screenshot of the application with its two messages for your lab notebook.

18. What is the URL of the image that is first shown in the UI as the message is loading?

The URL is https://www.google.com/images/spin-32.gif?a.

19. How do the fields in an image document differ from that of the text document?

The image document has an added "imageUrl" and "storageUri" property. The "text" property is not included.

**What URL and storage location can the image be found at?**

imageUrl:

"https://firebasestorage.googleapis.com/v0/b/firebase-khod2.appspot.com/o/bolVEaReNuYOha7DUpDfd6hryOf2%2FGGdMKpZf2m0b1eA4dX5D%2Fhawkgirl-shayera-hol-justice-league-3.72.jpg?alt=media&token=1be2181c-c07d-464b-a88c-fc050046d2bc"

storageUri:

"bolVEaReNuYOha7DUpDfd6hryOf2/GGdMKpZf2m0b1eA4dX5D/hawkgirl-shayera-hol-justice-league-3.72.jpg"

Take a screenshot of the image in the storage bucket for your lab notebook.
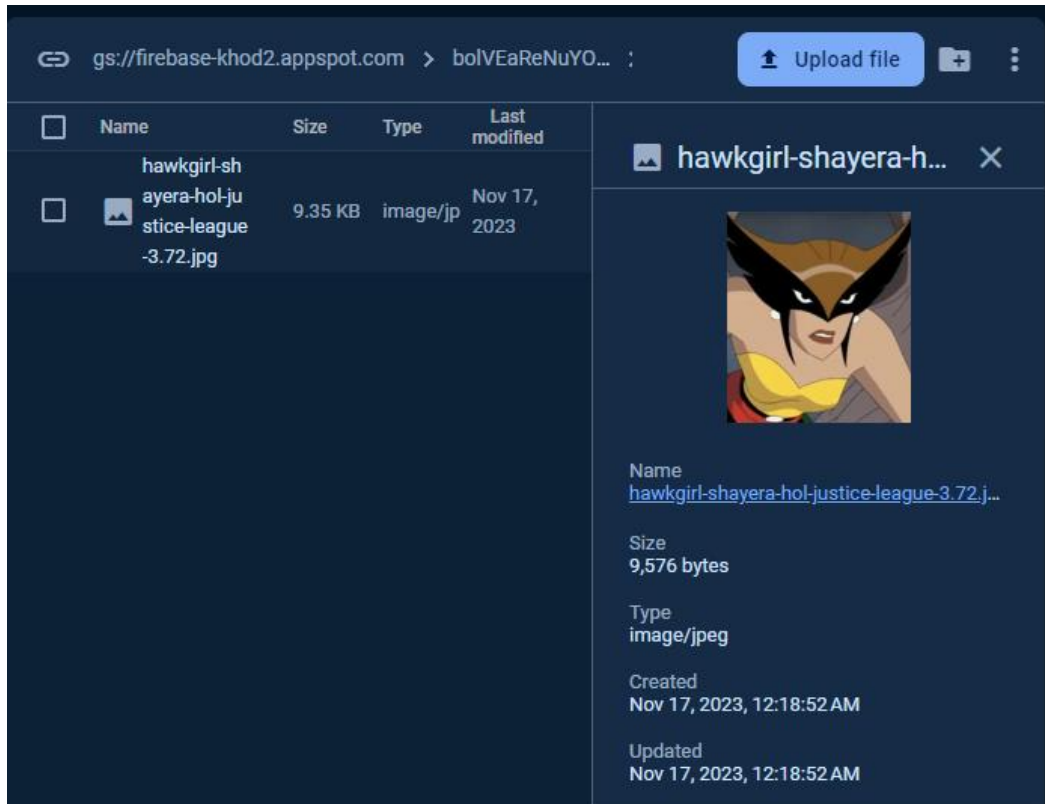
20. What directory is the application going to be served from?

codelab-friendlychat-web/web-start.

Take a screenshot of the message including the URL for your lab notebook.