

Lab notebook Week 8

Submitted by: Vishrut Sharma (OdinID: vishrut)

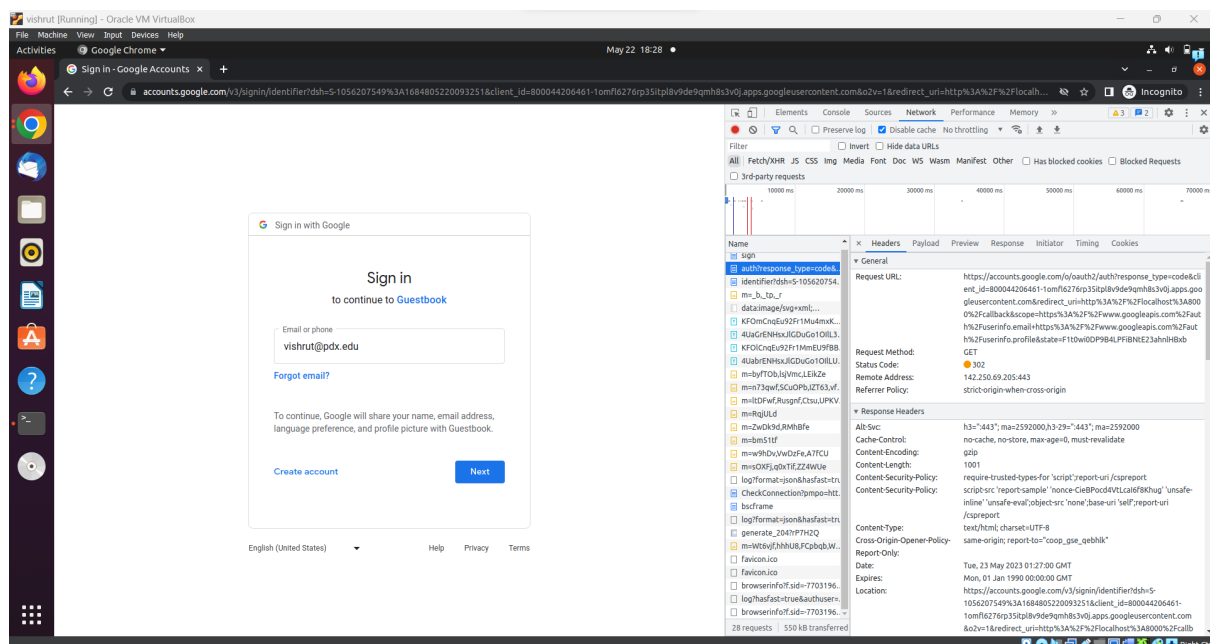
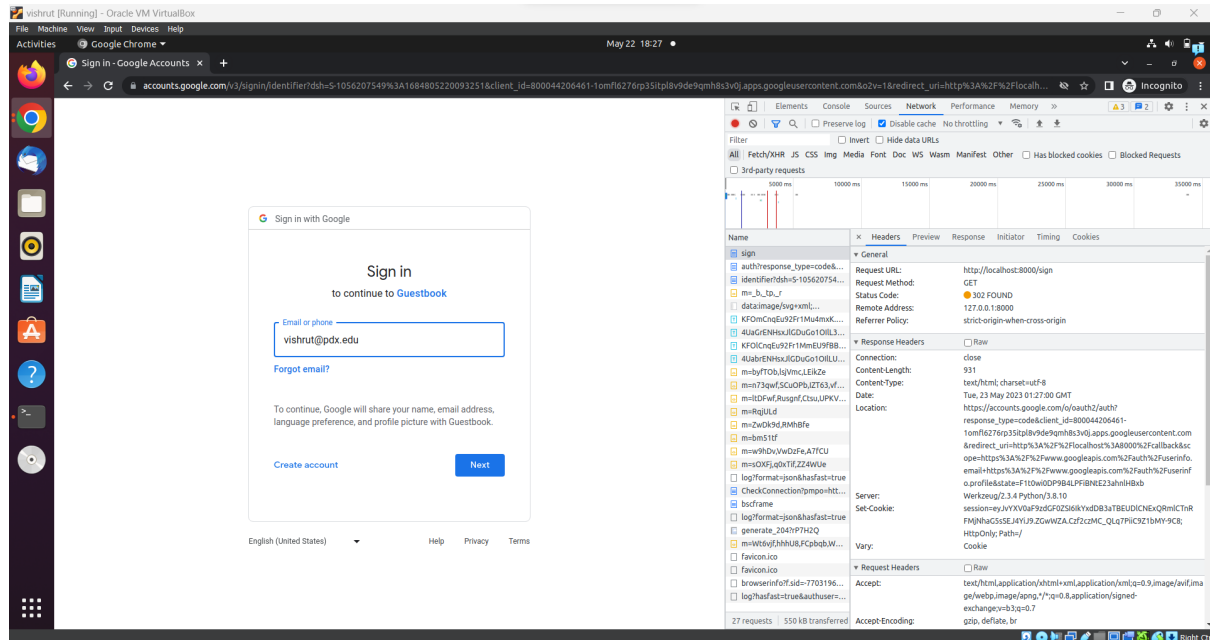
Table of Contents

08.1 OAuth2 Guestbook.....	2
12. Running the code.....	2
13. Removing access.....	4
08.2g: ML APIs.....	5
3. Vision.....	5
4. Speech.....	5
5. Translate.....	6
6. Natural Language.....	6
8. Code.....	7
9. Test integration.....	7
13. Video Intelligence.....	8
16. Application.....	8
17. Code.....	9
08.3g: Firebase.....	9
4. Authentication setup.....	9
8. Bundling with Webpack.....	10
12. Add authentication.....	10
13. Update UI.....	10
16. Test application with text messaging.....	11
17. Manual message insertion.....	11
18. Add image messaging.....	12
19. Test application with image messaging.....	12
20. Deploy application.....	12

08.1 OAuth2 Guestbook

12. 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.



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

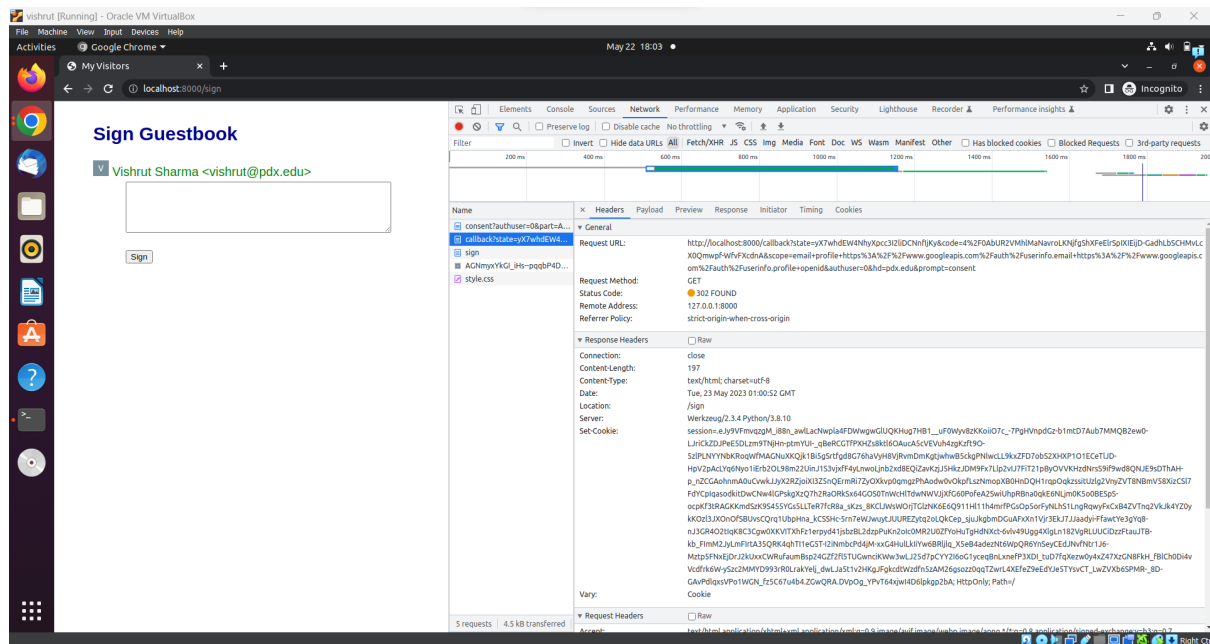
Answer: The following 3 lines of code are responsible for the second request.

- 1) `authorization_url, state = google.authorization_url(authorization_base_url)`

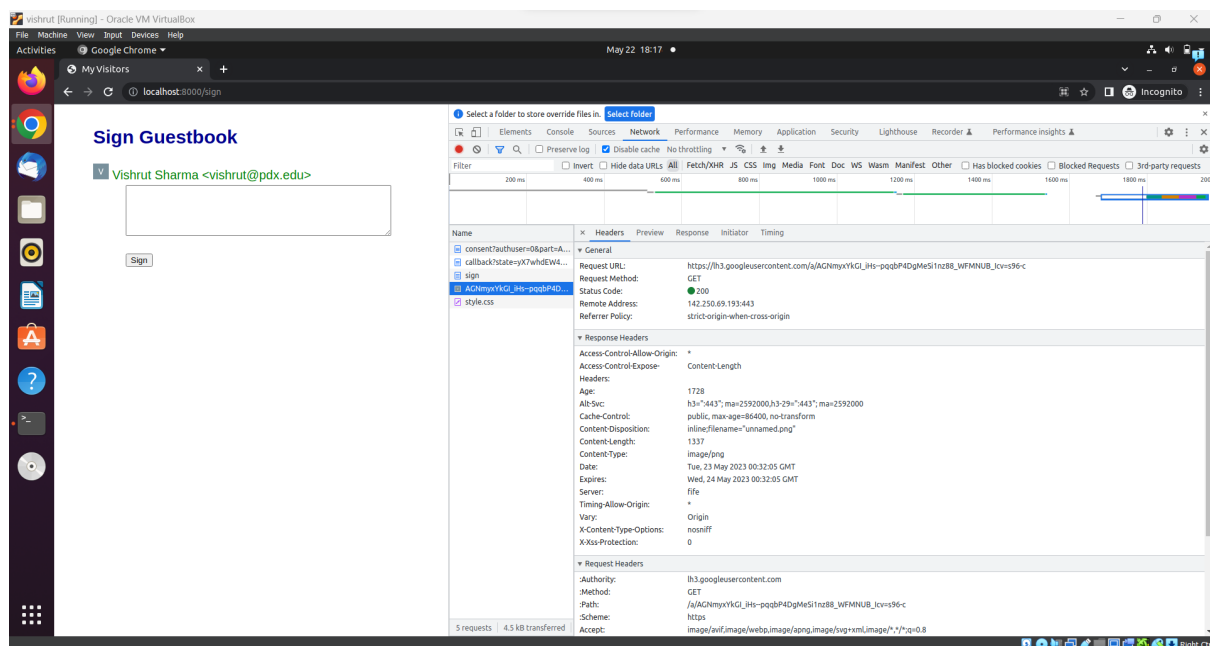
- 2) session['oauth_state'] = state
- 3) return redirect(authorization_url)

- 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?

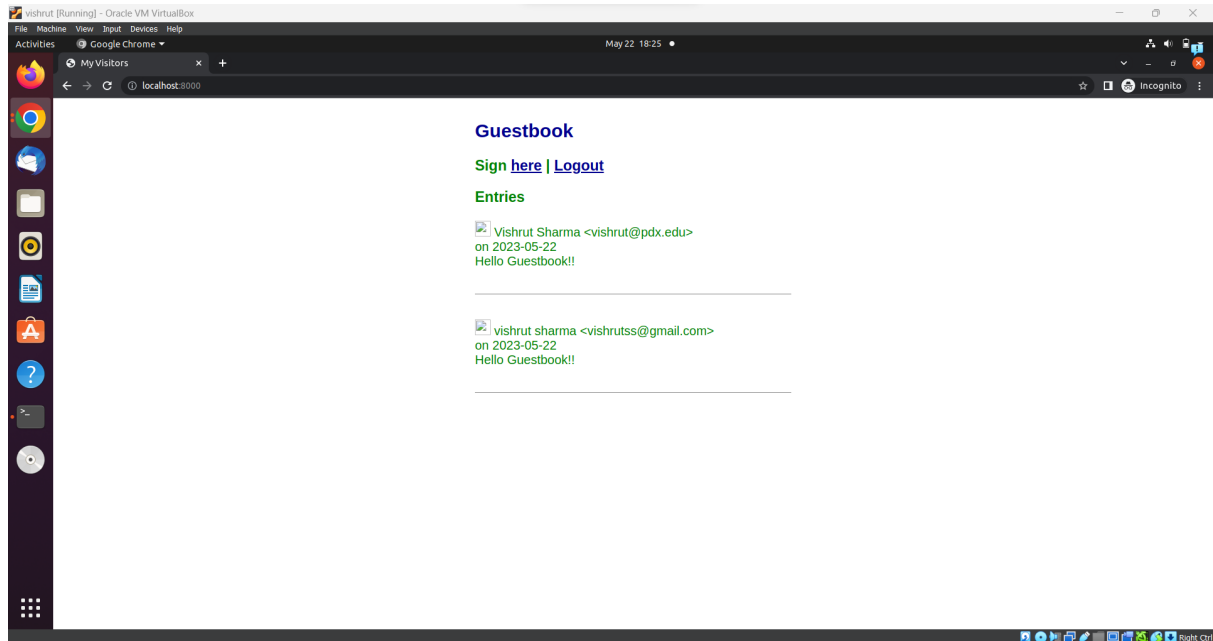
Answer: The user is sent to “/sign”.



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

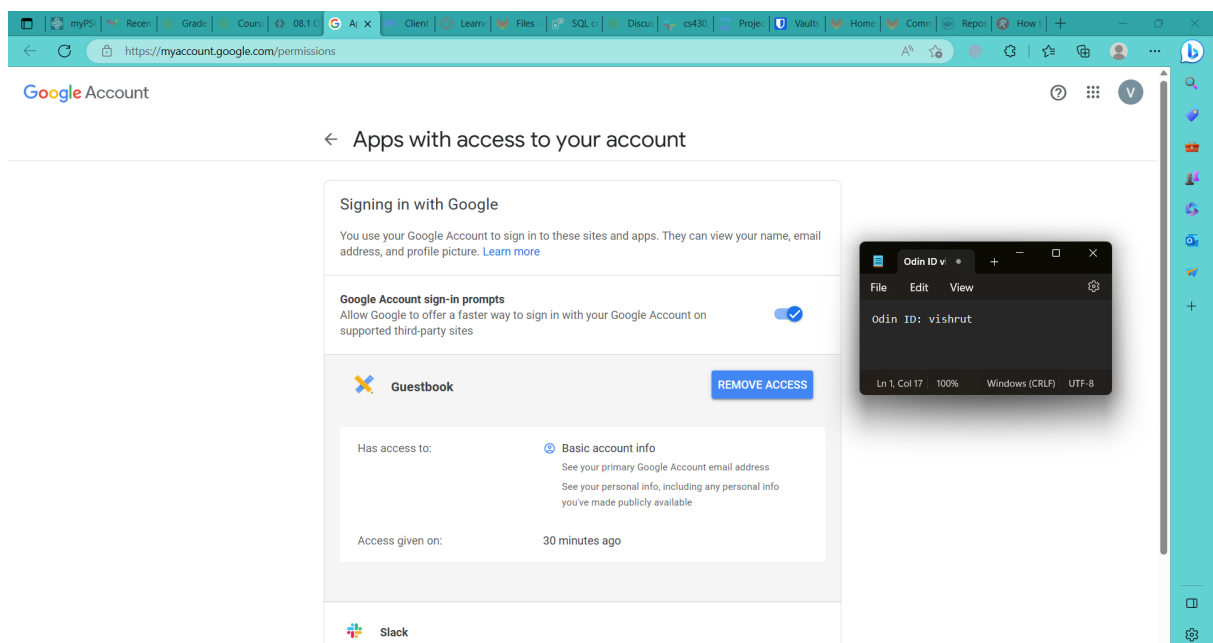


- Take a screenshot showing multiple authenticated accounts have been able to sign the Guestbook.



13. Removing access

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

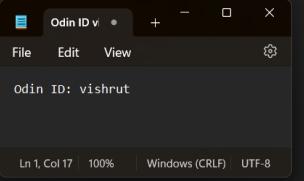


08.2g: ML APIs

3. Vision

- Show the output for your lab notebook

```
(env) vishrut@cloudshell:~/python-docs-samples/vision/snippets/detect (cloud-sharma-vishrut)$ python detect.py labels-uri gs://ml-api-codelab/birds.jpg
Labels:
Bird
Ratite
Cloud
Sky
Beak
Plant
Green
Neck
Ostrich
Casuariiformes
(env) vishrut@cloudshell:~/python-docs-samples/vision/snippets/detect (cloud-sharma-vishrut)$
```



- What is the name of the function?

Answer: detect_labels_uri(uri)

- What type of Vision client is instantiated in it?

Answer: ImageAnnotatorClient

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

Answer: label_detection(image=image)

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

Answer: response.label_annotations

- Take a screenshot of the output for the above commands

```
(env) vishrut@cloudshell:~/python-docs-samples/vision/snippets/detect (cloud-sharma-vishrut)$ wget https://1000logos.net/wp-content/uploads/2022/07/Portland-State-University-Logo.png -O psu_logo
--2023-05-23 01:49:41-- https://1000logos.net/wp-content/uploads/2022/07/Portland-State-University-Logo.png
Resolving 1000logos.net (1000logos.net)... 104.26.8.175, 172.67.71.45, 104.26.9.175, ...
Connecting to 1000logos.net (1000logos.net)|104.26.8.175|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 89725 (88K) [image/png]
Saving to: 'psu_logo'

psu_logo
100%[=====] 87.62K --.-KB/s in 0.008s

2023-05-23 01:49:41 (11.3 MB/s) - 'psu_logo' saved [89725/89725]

[1]+ Exit 8 wget https://www.google.com/search?q=psu+logo
(env) vishrut@cloudshell:~/python-docs-samples/vision/snippets/detect (cloud-sharma-vishrut)$ python detect.py detect_logos psu_logo
usage: detect.py [-h]
                 {faces,faces-uri,labels,labels-uri,landmarks,landmarks-uri,text,text-uri,logos,logos-uri,safe-search,safe-search-uri,properties,properties-uri,web,web-uri,web-geo,web-geo-uri,crophints,crophints-uri,document,document-uri,ocr-uri,object-localization,object-localization-uri}
                 ...
detect.py: error: argument command: invalid choice: 'detect_logos' (choose from 'faces', 'faces-uri', 'labels', 'labels-uri', 'landmarks', 'landmarks-uri', 'text', 'text-uri', 'logos', 'logos-uri', 'safe-search', 'safe-search-uri', 'properties', 'properties-uri', 'web', 'web-uri', 'web-geo', 'web-geo-uri', 'crophints', 'crophints-uri', 'document', 'document-uri', 'ocr-uri', 'object-localization', 'object-localization-uri')
(env) vishrut@cloudshell:~/python-docs-samples/vision/snippets/detect (cloud-sharma-vishrut)$ python detect.py logos psu_logo
logos:
Portland State University
(env) vishrut@cloudshell:~/python-docs-samples/vision/snippets/detect (cloud-sharma-vishrut)$
```

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

Answer: client.logo_detection(image=image)

4. Speech

- Show the output for your lab notebook

```
vishrut@cloudshell:~/python-docs-samples/speech/snippets (cloud-sharma-vishrut)$ python transcribe.py resources/audio.raw
Transcript: how old is the Brooklyn Bridge
vishrut@cloudshell:~/python-docs-samples/speech/snippets (cloud-sharma-vishrut)$
```

- What is the name of the function?

Answer: transcribe_file(speech_file)

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

Answer: client.recognize(config=config, audio=audio)

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

Answer: response.results

5. Translate

- Show the output for your lab notebook

```
vishrut@cloudshell:~/python-docs-samples/translate/samples/snippets (cloud-sharma-vishrut)$ python snippets.py translate-text en '你有没有带外套'
Text: 你有没有带外套
Translation: do you have a coat
Detected source language: zh-TW
vishrut@cloudshell:~/python-docs-samples/translate/samples/snippets (cloud-sharma-vishrut)$
```

- What is the name of the function?

Answer: translate_text(target: str, text: str) -> dict

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

Answer: translate_client.translate(text, target_language=target)

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

Answer: result["translatedText"]

6. Natural Language

- Show the output for your lab notebook

```
(env) vishrut@cloudshell:~ (cloud-sharma-vishrut)$ python language.py 'homework is awful!'
"homework is awful!" has sentiment=-0.800000011920929

Entities are:
name: homework
(env) vishrut@cloudshell:~ (cloud-sharma-vishrut)$ python language.py 'homework is ok'
"homework is ok" has sentiment=0.30000001192092896

Entities are:
name: homework
(env) vishrut@cloudshell:~ (cloud-sharma-vishrut)$ python language.py 'homework is awesome?'
"homework is awesome?" has sentiment=0.4000000059604645

Entities are:
name: homework
(env) vishrut@cloudshell:~ (cloud-sharma-vishrut)$ python language.py 'homework is awesome!'
"homework is awesome!" has sentiment=0.8999999761581421

Entities are:
name: homework
(env) vishrut@cloudshell:~ (cloud-sharma-vishrut)$ python language.py 'The protestors in Oregon put on gas masks and wore yellow t-shirts'
"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) vishrut@cloudshell:~ (cloud-sharma-vishrut)$
```

8. Code

- **What is the name of the function that performs the transcription?**

Answer: transcribe_gcs

- **What is the name of the function that performs the translation?**

Answer: translate_text

- **What is the name of the function that performs the entity analysis on the translation?**

Answer: entities_text

- **What is the name of the function that performs the entity analysis on the image?**

Answer: detect_labels_uri

9. 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?**

Answer: Within the "compare_audio_to_image" function, we can try to adjust the confidence threshold that determines a match between entities and labels. At present, the function only verifies if the entity name precisely matches a label. To enhance the comparison logic, we can enable partial matches or explore the possibility of utilizing a similarity metric to evaluate the connection between entities and labels.

- **If the program deems them unrelated, then based on the results from the APIs, what must be changed in the program to address this?**

Answer: We can modify the confidence threshold for matching entities and labels. It is worth exploring different similarity metrics or considering the integration of additional APIs or models to extract more comprehensive details from both the audio and image. This could involve object detection or scene analysis to obtain more detailed information from the data.

- **If the program deems them unrelated, then based on the results from the APIs, what must be changed in the program to address this?**

Answer: We can reassess the confidence threshold and the matching logic employed for entities and labels. Furthermore, it is worth considering the exploration of advanced techniques, such as leveraging deep learning-based models, for audio and image analysis. These models possess the capability to capture more sophisticated features and semantics, facilitating a more profound comprehension and comparison between the audio and image data.

13. Video Intelligence

- What are the top 3 labels that the Video Intelligence API associates with the video and what is its confidence in them?

```
(env) vishrut@cloudshell:~ (cloud-sharma-vishrut)$ python labels.py gs://{CLOUD_STORAGE_BUCKET}/SportsBloopers2016.mp4

Processing video for label annotations:

Finished processing.
Video label description: sport venue
  Label category description: location
  Label category description: structure
  Segment 0: 0s to 178s
  Confidence: 0.3432130515575409

Video label description: audience
  Label category description: people
  Segment 0: 0s to 178s
  Confidence: 0.4254549741744995

Video label description: stadium
  Label category description: location
  Label category description: structure
  Segment 0: 0s to 178s
  Confidence: 0.5119114518165588
```

- What is the name of the client class in the package that is used?

Answer: `videointelligence.VideoIntelligenceServiceClient`

- What method is used in that class to perform the annotation?

Answer: `annotate_video`

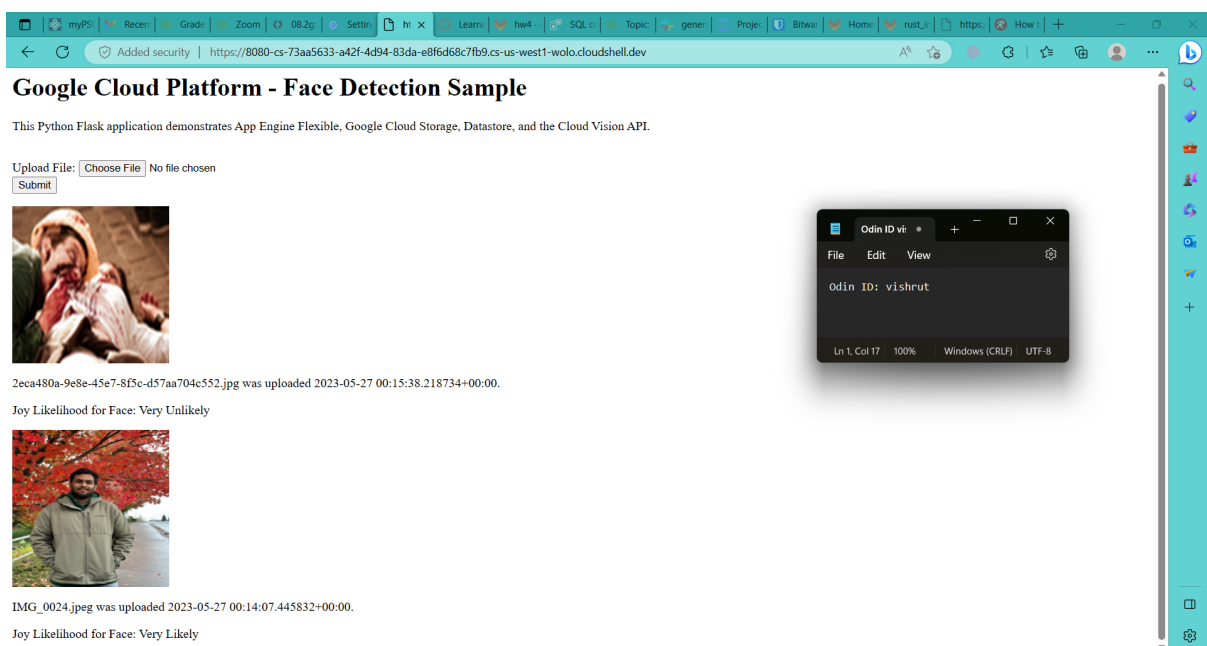
16. Application

- 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 Cloud Vision API.

Upload File: No file chosen



2eca480a-9e8e-45e7-8f5c-d57aa704c552.jpg was uploaded 2023-05-27 00:15:38.218734+00:00.

Joy Likelihood for Face: Very Unlikely



IMG_0024.jpeg was uploaded 2023-05-27 00:14:07.445832+00:00.

Joy Likelihood for Face: Very Likely

Odin ID vishrut

Odin ID: vishrut

Ln 1, Col 17 100% Windows (CRLF) UTF-8

17. Code

- **What line of code creates the query for previous detections?**

Answer: Line 39. `Query=datastore_client.query(kind="Faces")`

- **What line of code sends the query to Cloud Datastore?**

Answer: Line 40. `Image_entities=list(query.fetch())`

- **Show the line that retrieves the name of the storage bucket to use.**

Answer: `bucket = storage_client.get_bucket(CLOUD_STORAGE_BUCKET)`

- **What form field is used to specify the uploaded photo?**

Answer: `photo = request.files["file"]`

- **Show the line that copies the photo's contents to the storage bucket.**

Answer: `blob = bucket.blob(photo.filename)`

`blob.upload_from_string(photo.read(), content_type=photo.content_type)`

- **What method in Vision's annotation client is used to perform the analysis?**

Answer: `faces = vision_client.face_detection(image=image).face_annotations`

- **What fields are stored in Cloud Datastore for each image?**

Answer:

blob_name: The name of the blob (image file) in the storage bucket.

image_public_url: The publicly accessible URL of the image.

timestamp: The date and time of the upload.

joy: The likelihood that the detected face displays 'joy.'

- **What happens at the end of the upload_photo route?**

Answer: Upon uploading the photo and analyzing the face, a fresh entity will be generated within the cloud datastore. This entity will contain all the relevant information pertaining to the photo and the analyzed face. Subsequently, the entity will be stored in Cloud Datastore, and we will then be redirected to the home page.

08.3g: Firebase

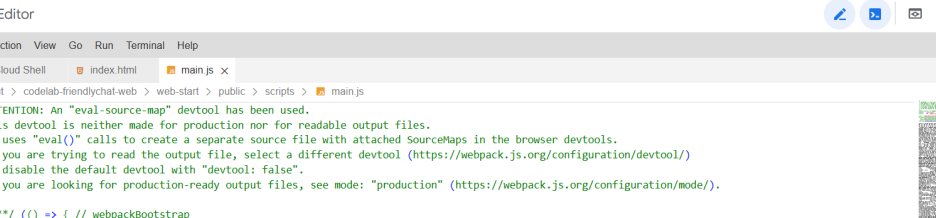
4. Authentication setup

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

Answer: localhost, fir-vishrut.firebaseio.com, fir-vishrut.web.app

8. Bundling with Webpack

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



```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000
```

12. Add authentication

- **What missing functions deal with user authentication?**

Answer: signIn, signOutUser, and initFirebaseAuth()

- **What missing functions deal with sending and receiving messages?**

Answer: loadMessages

13. Update UI

- **What are the names of the elements that are hidden when the user is signed out?**

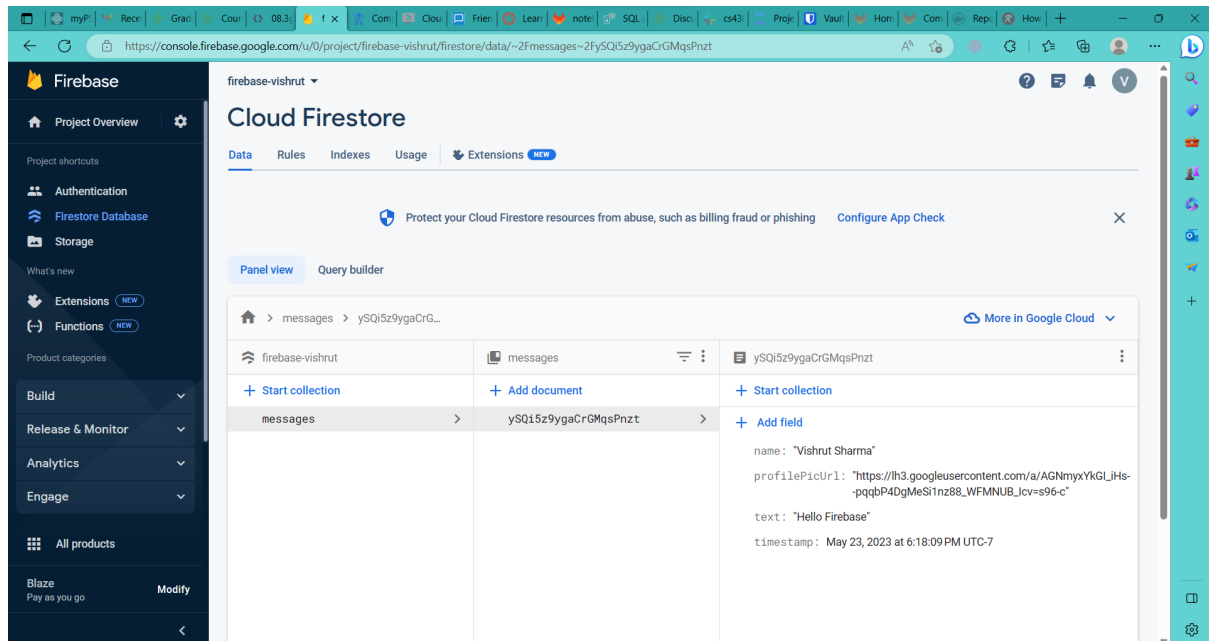
Answer: userNameElement, userPicElement and signOutButtonElement

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

Answer: signInButtonElement

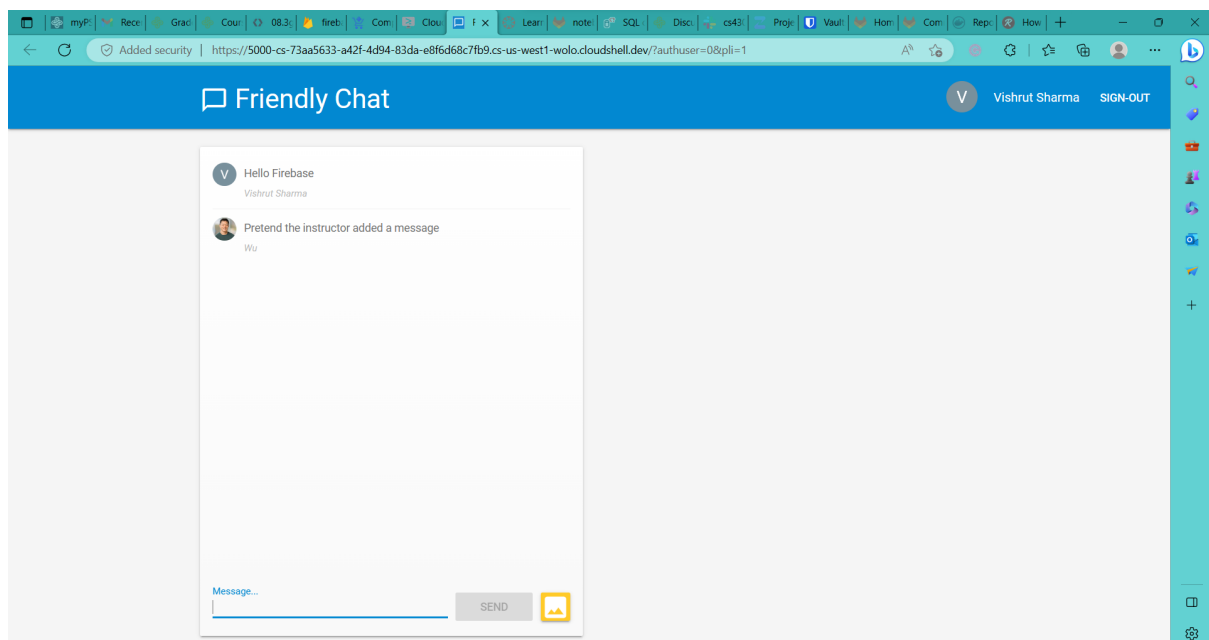
16. Test application with text messaging

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



17. Manual message insertion

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



18. Add image messaging

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

Answer: <https://www.google.com/images/spin-32.gif?a>

19. Test application with image messaging

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

Answer: Image document has imageUrl, name, profilePicUrl, storageUri and timestamp.
Text document has name, profilePicUrl, text and timestamp

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

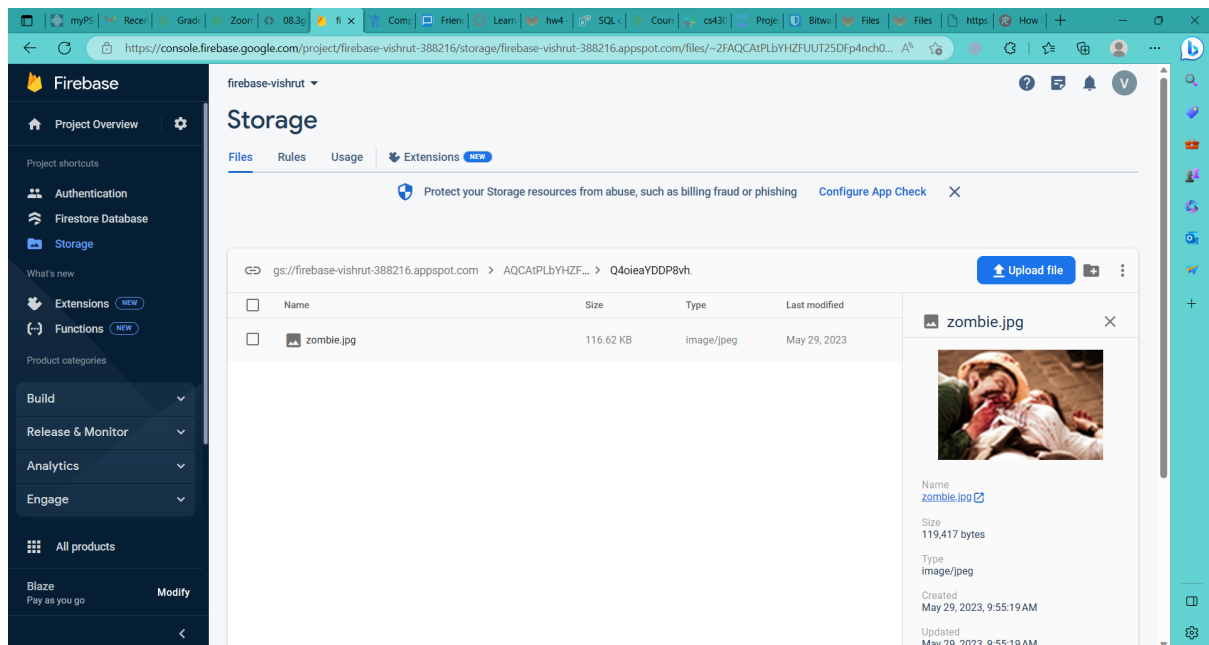
Answer: URL:

<https://firebasestorage.googleapis.com/v0/b/firebase-vishrut-388216.appspot.com/o/AQCAtPLbYHZFUUT25DFp4nch0o33%2FQ4oieaYDDP8vh4lWg1fa%2Fzombie.jpg?alt=media&token=362e508a-a268-4e43-9fe8-4b0245ae6db4>

Storage location:

gs://firebase-vishrut-388216.appspot.com/AQCAtPLbYHZFUUT25DFp4nch0o33/Q4oieaYDDP8vh4lWg1fa/zombie.jpg

- Take a screenshot of the image in the storage bucket for your lab notebook.



20. Deploy application

- What directory is the application going to be served from?

Answer: ~/codelab-friendlychat-web/web-start/src

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

