SI Final Project
Face Detection and Verification API

Shruti Kharkar
801261325

Project Overview:

API (Application Programming Interface) is a software interface that enables two apps to communicate with one another.
The Azure Face service uses artificial intelligence (AI) to discover, recognize, and analyze human faces in photos. Facial recognition software is useful in a variety of situations, including identity verification, touchless access control, and privacy face blurring.
In this project, MS Azure's Face API is used to detect and verify the faces in the images provided through the image URL as input. Followed the steps given in the Canvas and implemented the project.

- **Face Detection:** This helps to identify one or more human faces in an image, as well as features such as stance, facial coverings, and face location, using 27 landmarks for each face. Currently for this project we are just detecting the face from the image.

- **Face Verify:** Verifies the face by receiving a confidence score after determining the chance that two faces belong to the same individual through the image provided.

- **Features of using the Face API:**
  1. Advanced Facial Recognition
  2. Easy to use and implement
  3. Security purposes

To test the API:

POST method has been used for both face detection and verification using Swagger which makes more presentable and structured design for the API.
Visit http://143.198.162.71:3000/docs/ for performing the detection and verification.

1) Face Detection:

- In **/detect**, input request parameters used are as below where the string is the URL of the image from which the face needs to be detected.

```
{
   "Image_URL": "string"
}
```

- If the request is successful, the response/output will be generated in the JSON object itself with the face coordinates and the faceId.
- If the request is unsuccessful for certain reason, the error will be displayed.
- Below Error codes are used:

  200: The detection was SUCCESSFUL and the output provides us with an array object containing information

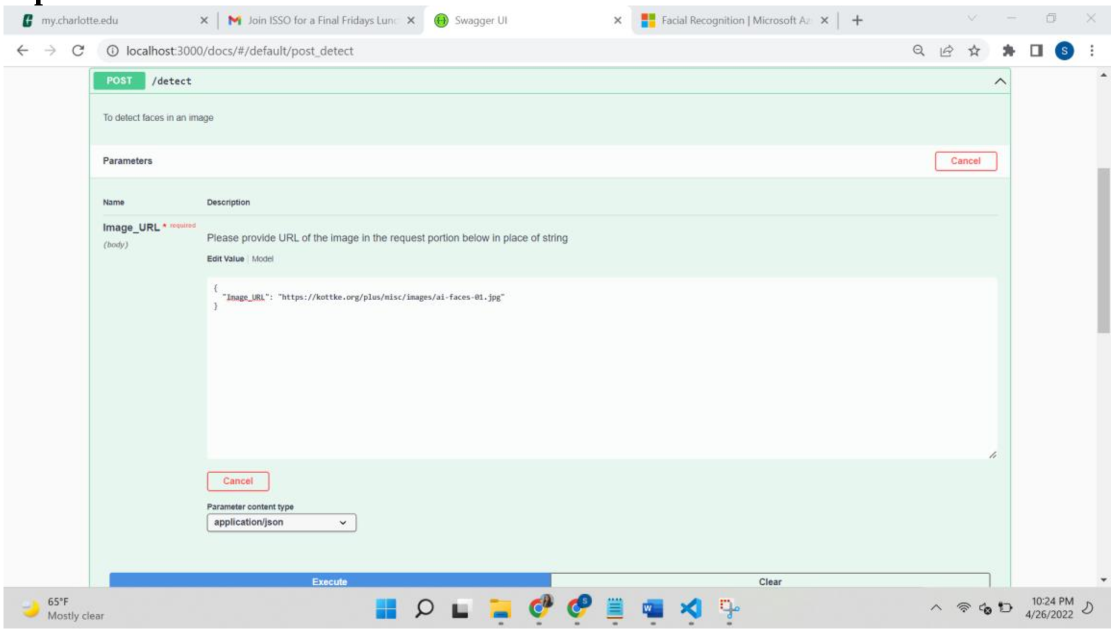
  400: Please provide valid input

  500: Some internal server error

Please find attached below screenshot of the test cases for face detection:

1) Image URL used in the input parameter:
https://kottke.org/plus/misc/images/ai-faces-01.jpg

Input:

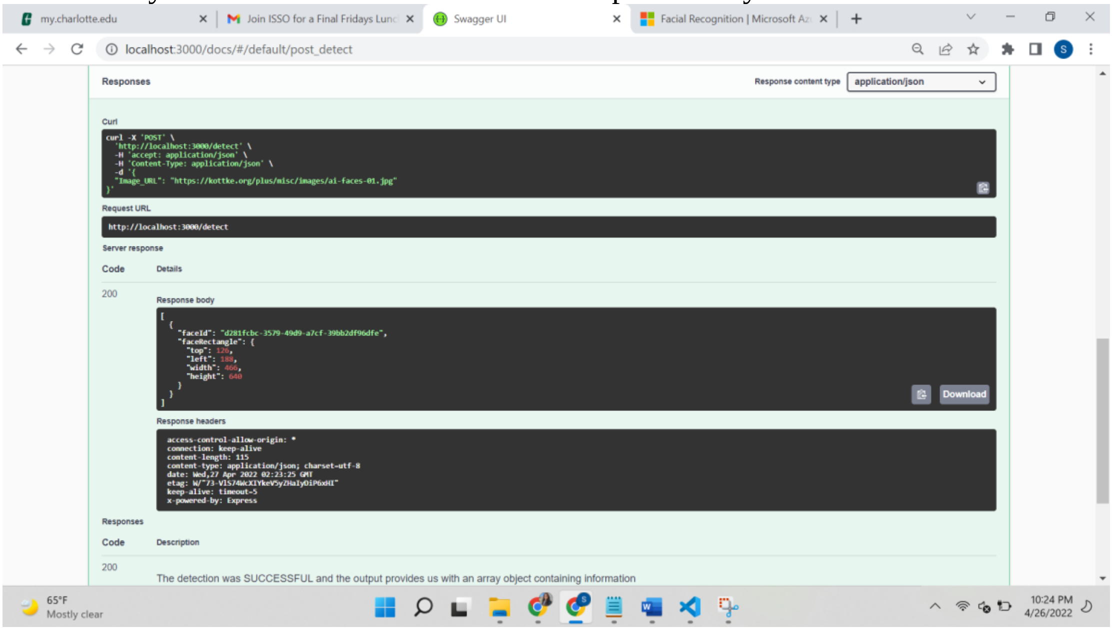


To detect faces in an image

Image_URL * required (body) — Please provide URL of the image in the request portion below in place of string

```
{
  "Image_URL": "https://kottke.org/plus/misc/images/ai-faces-01.jpg"
}
```

Ouput:
Successfully detected the face as seen in the response body with the faceId


localhost:3000/docs/#/default/post_detect

Responses — Response content type: application/json

Curl
```
curl -X 'POST' \
  'http://localhost:3000/detect' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "Image_URL": "https://kottke.org/plus/misc/images/ai-faces-01.jpg"
}'
```
Request URL
```
http://localhost:3000/detect
```
Server response

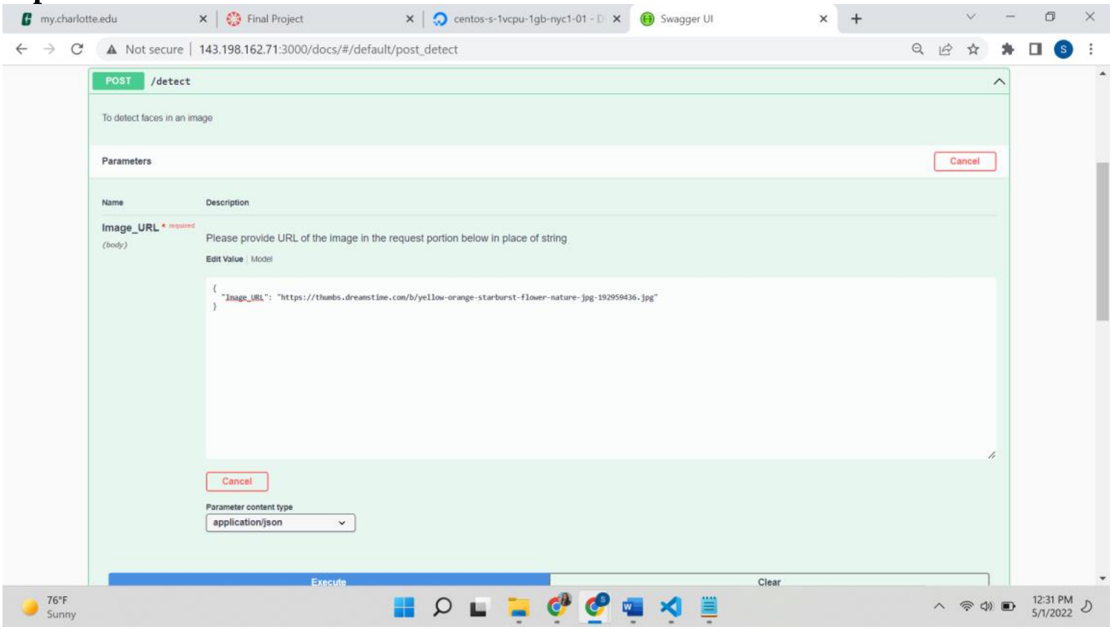
Code: 200 — Response body
```
[
  {
    "faceId": "d281fcbc-3579-49d9-a7cf-39bb2df96dfe",
    "faceRectangle": {
      "top": 126,
      "left": 188,
      "width": 466,
      "height": 640
    }
  }
]
```
Response headers
```
access-control-allow-origin: *
connection: keep-alive
content-length: 115
content-type: application/json; charset=utf-8
date: Wed,27 Apr 2022 02:23:25 GMT
etag: W/"73-V1S74WcXIYkeV5yZHaIyDiP6xHI"
keep-alive: timeout=5
x-powered-by: Express
```
Responses

Code	Description
200	The detection was SUCCESSFUL and the output provides us with an array object containing information

my.charlotte.edu
Join ISSO for a Final Fridays Lunc
Swagger UI
Facial Recognition | Microsoft Az
Download
65°F
Mostly clear
10:24 PM
4/26/2022

2) Image URL used in the input parameter is the Image without a face:
https://thumbs.dreamstime.com/b/yellow-orange-starburst-flower-nature-jpg-192959436.jpg

Input:


Not secure | 143.198.162.71:3000/docs/#/default/post_detect

POST /detect

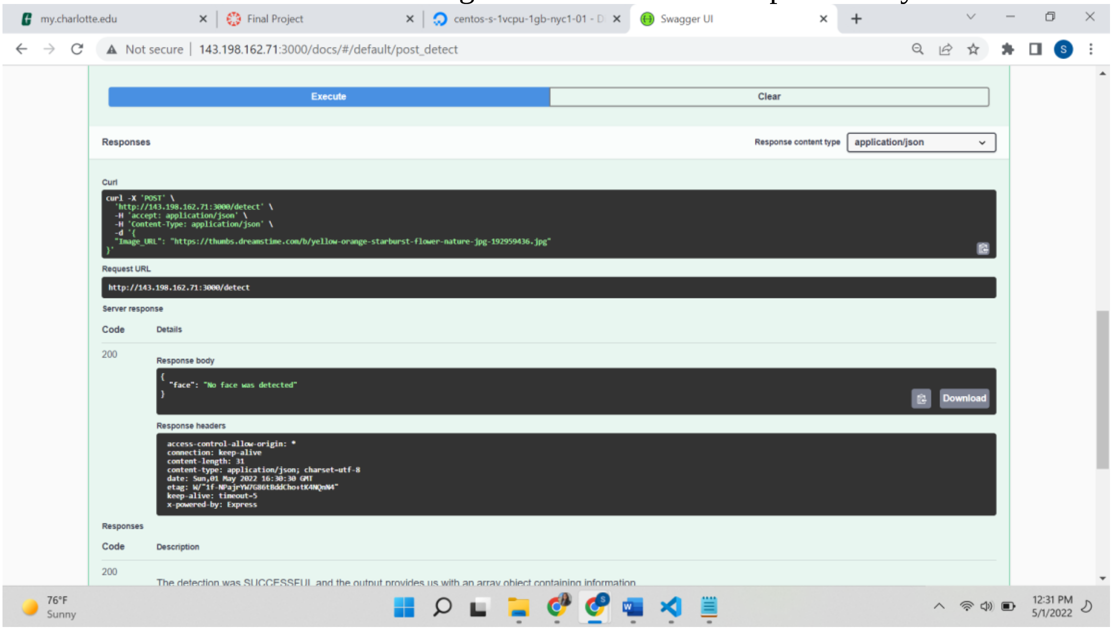
To detect faces in an image

Parameters

Name	Description
Image_URL * required (body)	Please provide URL of the image in the request portion below in place of string

Edit Value | Model
```
{
  "Image_URL": "https://thumbs.dreamstime.com/b/yellow-orange-starburst-flower-nature-jpg-192959436.jpg"
}
```
Parameter content type: application/json

Ouput:
Could not detect the face in the image can be seen in the response body



143.198.162.71:3000/docs/#/default/post_detect

Responses — Response content type: application/json

Curl
```
curl -X 'POST' \
  'http://143.198.162.71:3000/detect' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "Image_URL": "https://thumbs.dreamstime.com/b/yellow-orange-starburst-flower-nature-jpg-192959436.jpg"
}'
```

Request URL
```
http://143.198.162.71:3000/detect
```

Server response

Code 200 — Response body
```
{
  "face": "No face was detected"
}
```

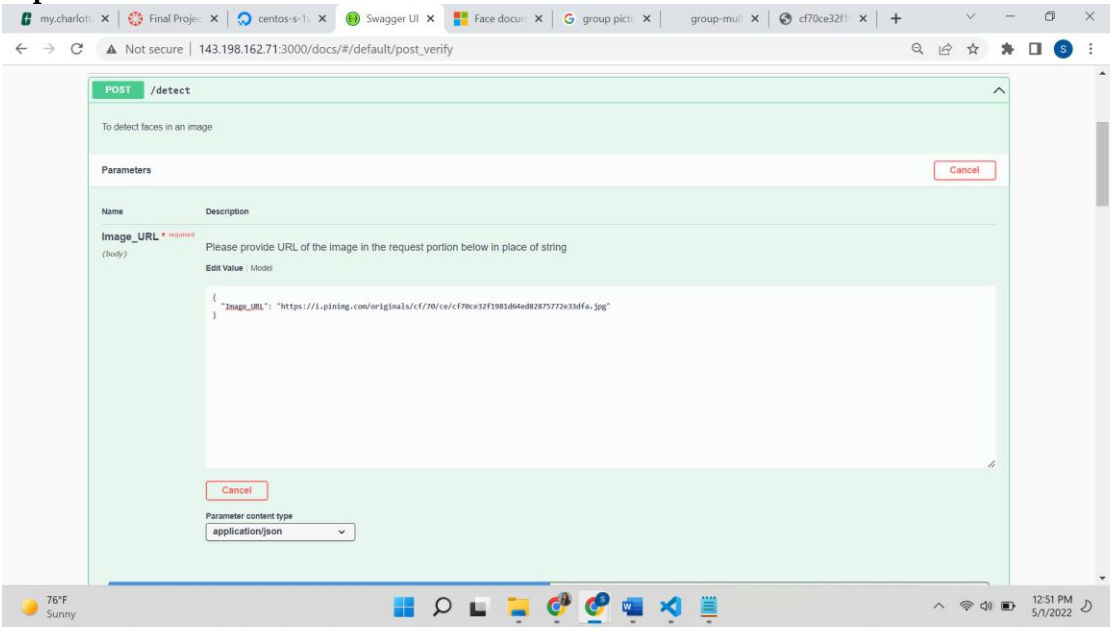
Response headers
```
 access-control-allow-origin: *
 connection: keep-alive
 content-length: 31
 content-type: application/json; charset-utf-8
 date: Sun,01 May 2022 16:30:30 GMT
 etag: W/"1f-NPajrYW7G86tBddCho+tK4NQmN4"
 keep-alive: timeout-5
 x-powered-by: Express
```

Responses

Code 200 — Description: The detection was SUCCESSFUL and the output provides us with an array object containing information

3) Image URL used in the input parameter is the group picture with multiple faces to detect:
https://i.pinimg.com/originals/cf/70/ce/cf70ce32f1981d64ed82875772e33dfa.jpg

Input:



143.198.162.71:3000/docs/#/default/post_verify

POST /detect

To detect faces in an image

Parameters

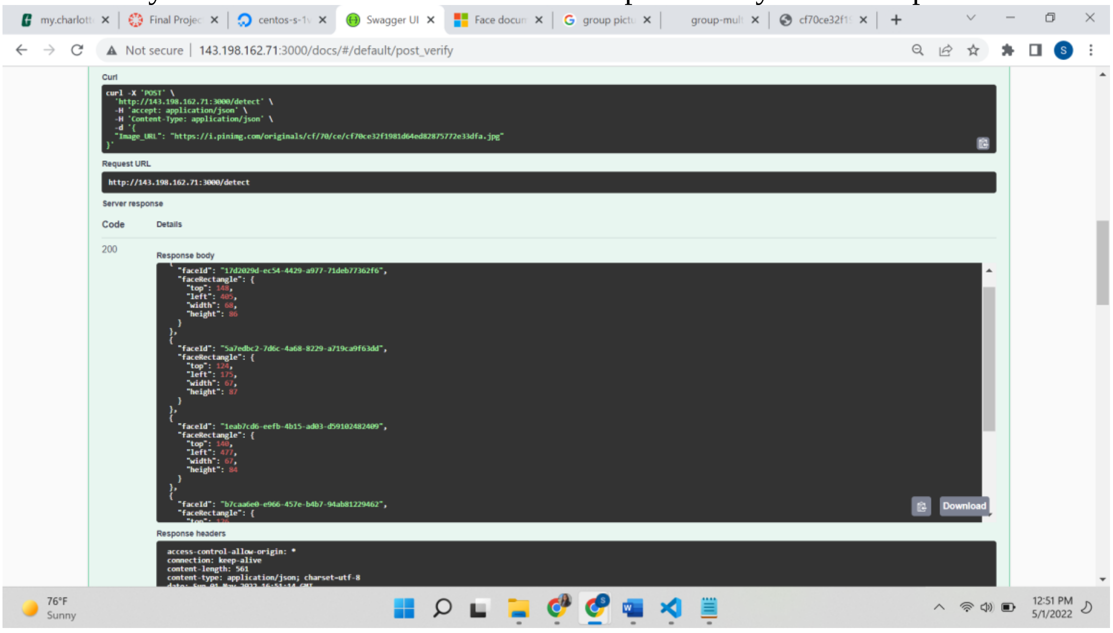
Name: Image_URL * required (body)

Description: Please provide URL of the image in the request portion below in place of string

```
{
  "Image_URL": "https://i.pinimg.com/originals/cf/70/ce/cf70ce32f1981d64ed82875772e33dfa.jpg"
}
```

Parameter content type: application/json

Output:
Successfully detected the faces as seen in the response body with multiple faceIDs



143.198.162.71:3000/docs/#/default/post_verify

Curl

```
curl -X 'POST' \
  'http://143.198.162.71:3000/detect' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "Image_URL": "https://i.pinimg.com/originals/cf/70/ce/cf70ce32f1981d64ed82875772e33dfa.jpg"
}'
```

Request URL

```
http://143.198.162.71:3000/detect
```

Server response

Code Details

200

Response body

```
    "faceId": "17d2029d-ec54-4429-a977-71deb77362f6",
    "faceRectangle": {
      "top": 148,
      "left": 405,
      "width": 68,
      "height": 86
    }
  },
  {
    "faceId": "5a7edbc2-7d6c-4a68-8229-a719ca9f63dd",
    "faceRectangle": {
      "top": 124,
      "left": 175,
      "width": 67,
      "height": 87
    }
  },
  {
    "faceId": "1eab7cd6-eefb-4b15-ad03-d59102482409",
    "faceRectangle": {
      "top": 140,
      "left": 477,
      "width": 67,
      "height": 84
    }
  },
  {
    "faceId": "b7caa6e0-e966-457e-b4b7-94ab81229462",
    "faceRectangle": {
```

Response headers

```
access-control-allow-origin: *
connection: keep-alive
content-length: 561
content-type: application/json; charset=utf-8
```

1) Face Verification:

- In **/verify**, input request parameters used are as below where the string is the URL of the image from which the face needs to be verified with each other.

  {
  "Image_URL1": "string",
  "Image_URL2": "string"
  }

- If the request is successful, the response/output will be generated in the JSON object with the isIdentical values as 'true' and confidence values as 1 and If not able to verify then the isIdentical value will be 'false' with certain value for confidence.

- If the request is unsuccessful for certain reason, the error will be displayed.

- Below Error codes are used:

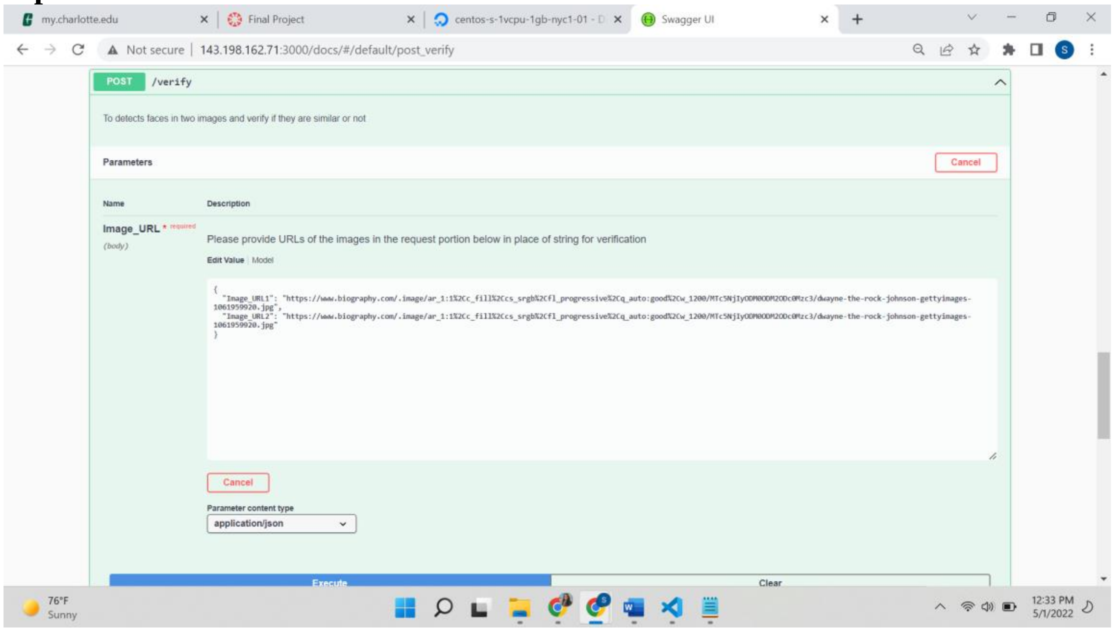
  200: The detection was SUCCESSFUL and the output provides us with an array object containing information that two faces are similar or not

  400: Please provide valid input

  500: Some internal server error

Please find attached below screenshot of the test cases for face verification:
1) Image URL used in the input parameter is:
https://www.biography.com/.image/ar_1:1%2Cc_fill%2Ccs_srgb%2Cfl_progressive%2Cq_auto:good%2Cw_1200/MTc5NjIyODM0ODM2ODc0Mzc3/dwayne-the-rock-johnson-gettyimages-1061959920.jpg

Input:



To detects faces in two images and verify if they are similar or not

Parameters

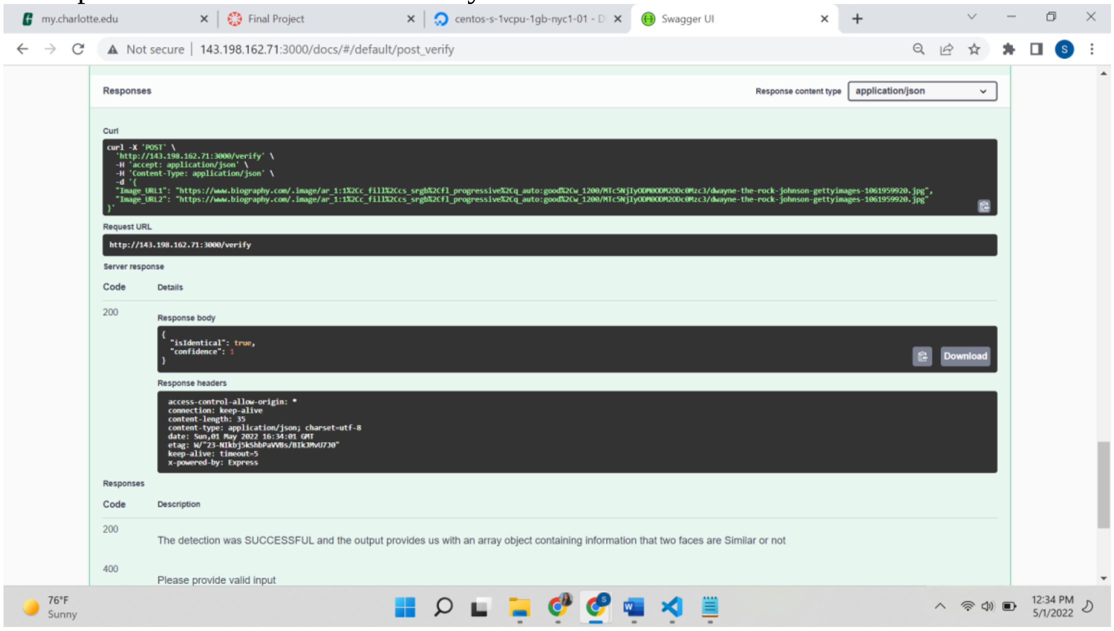
Name	Description
Image_URL * required (body)	Please provide URLs of the images in the request portion below in place of string for verification

Edit Value | Model

```
{
  "Image_URL1": "https://www.biography.com/.image/ar_1:1%2Cc_fill%2Ccs_srgb%2Cfl_progressive%2Cq_auto:good%2Cw_1200/MTc5NjIyODM0ODM2ODc0Mzc3/dwayne-the-rock-johnson-gettyimages-
1061959920.jpg",
  "Image_URL2": "https://www.biography.com/.image/ar_1:1%2Cc_fill%2Ccs_srgb%2Cfl_progressive%2Cq_auto:good%2Cw_1200/MTc5NjIyODM0ODM2ODc0Mzc3/dwayne-the-rock-johnson-gettyimages-
1061959920.jpg"
}
```

Parameter content type: application/json

Output:
In response body, we can see isIdentical = true which mean the face belongs to the same person. The face is successfully verified.


my.charlotte.edu
Final Project
centos-s-1vcpu-1gb-nyc1-01
Swagger UI
Not secure | 143.198.162.71:3000/docs/#/default/post_verify
Download
76°F Sunny
12:34 PM 5/1/2022

Responses

Response content type: application/json

Curl

```
curl -X 'POST' \
  'http://143.198.162.71:3000/verify' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "Image_URL1": "https://www.biography.com/.image/ar_1:1%2Cc_fill%2Ccs_srgb%2Cfl_progressive%2Cq_auto:good%2Cw_1200/MTc5NjIyODM0ODM2ODc0Mzc3/dwayne-the-rock-johnson-gettyimages-1061959920.jpg",
  "Image_URL2": "https://www.biography.com/.image/ar_1:1%2Cc_fill%2Ccs_srgb%2Cfl_progressive%2Cq_auto:good%2Cw_1200/MTc5NjIyODM0ODM2ODc0Mzc3/dwayne-the-rock-johnson-gettyimages-1061959920.jpg"
}'
```

Request URL

```
http://143.198.162.71:3000/verify
```

Server response

Code Details

200

Response body

```
{
  "isIdentical": true,
  "confidence": 1
}
```

Response headers

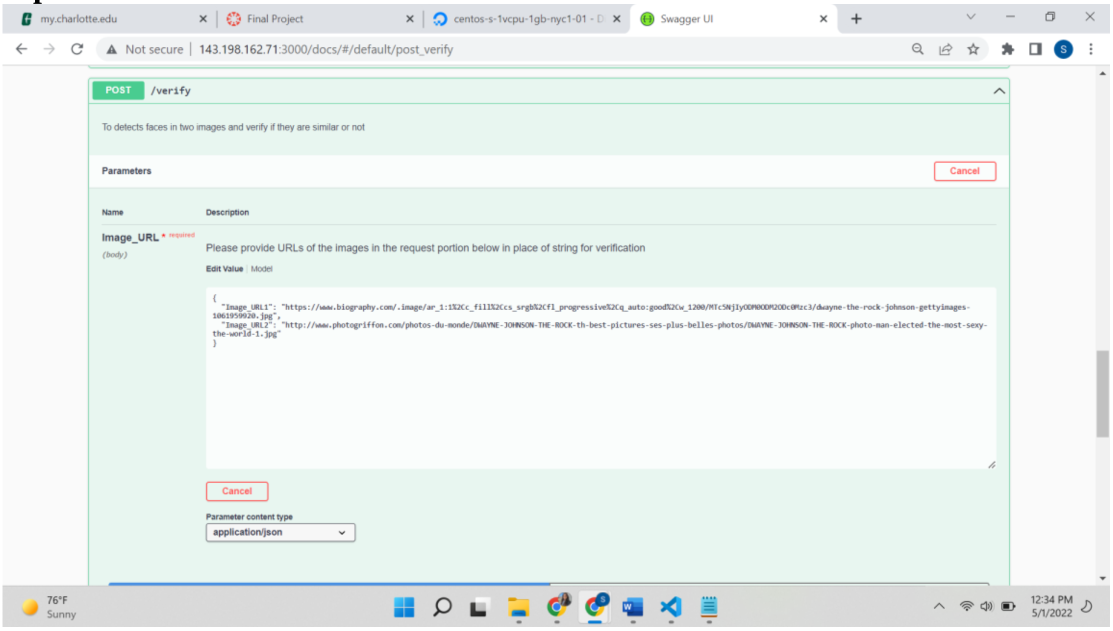
```
 access-control-allow-origin: *
 connection: keep-alive
 content-length: 35
 content-type: application/json; charset=utf-8
 date: Sun,01 May 2022 16:34:01 GMT
 etag: W/"23-NIkbj5kShbPaVVBs/BIkJMvU7J0"
 keep-alive: timeout=5
 x-powered-by: Express
```

Responses

Code	Description
200	The detection was SUCCESSFUL and the output provides us with an array object containing information that two faces are Similar or not
400	Please provide valid input

2) Image URL used in the input parameter is of the same person but a different image of them.
Url1 used:
https://www.biography.com/.image/ar_1:1%2Cc_fill%2Ccs_srgb%2Cfl_progressive%2Cq_auto:good%2Cw_1200/MTc5NjIyODM0ODM2ODc0Mzc3/dwayne-the-rock-johnson-gettyimages-1061959920.jpg
Url2 used:
http://www.photogriffon.com/photos-du-monde/DWAYNE-JOHNSON-THE-ROCK-th-best-pictures-ses-plus-belles-photos/DWAYNE-JOHNSON-THE-ROCK-photo-man-elected-the-most-sexy-the-world-1.jpg

Input:



/verify

To detects faces in two images and verify if they are similar or not

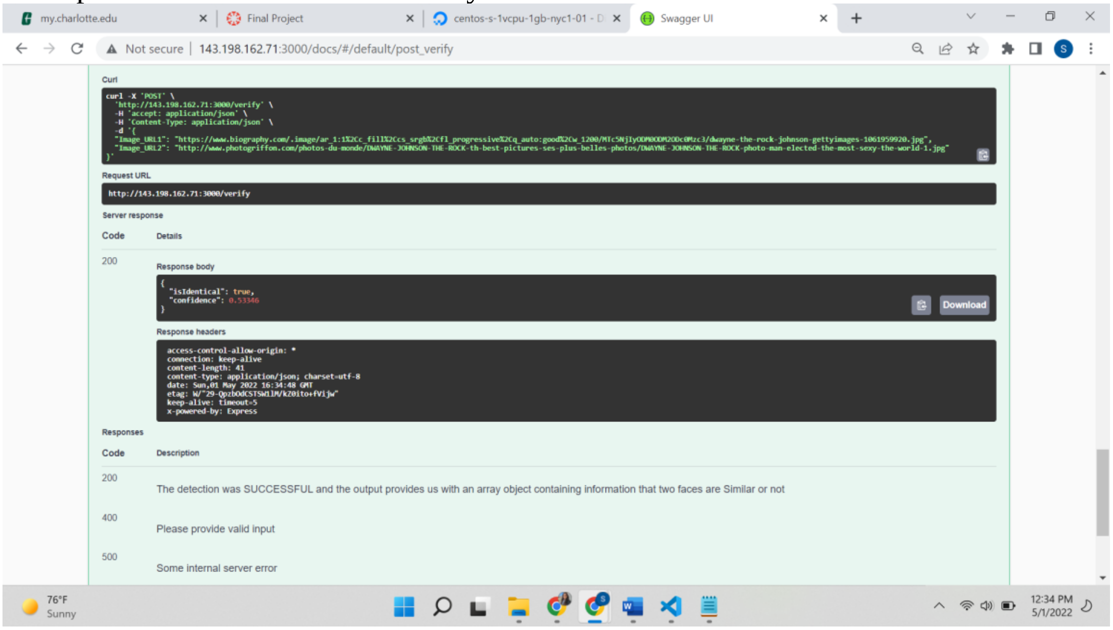
Parameters

Name	Description
Image_URL * required (body)	Please provide URLs of the images in the request portion below in place of string for verification

```
{
  "Image_URL1": "https://www.biography.com/.image/ar_1:1%2Cc_fill%2Ccs_srgb%2Cfl_progressive%2Cq_auto:good%2Cw_1200/MTc5NjIyODM0ODM2ODc0Mzc3/dwayne-the-rock-johnson-gettyimages-1061959920.jpg",
  "Image_URL2": "http://www.photogriffon.com/photos-du-monde/DWAYNE-JOHNSON-THE-ROCK-th-best-pictures-ses-plus-belles-photos/DWAYNE-JOHNSON-THE-ROCK-photo-man-elected-the-most-sexy-the-world-1.jpg"
}
```

Parameter content type: application/json

Output:
In response body, we can see isIdentical = true which mean the face belongs to the same person. The face is successfully verified.


my.charlotte.edu
Final Project
centos-s-1vcpu-1gb-nyc1-01 - D
Swagger UI
Not secure | 143.198.162.71:3000/docs/#/default/post_verify
Download
76°F
Sunny
12:34 PM
5/1/2022

Curl

```
curl -X 'POST' \
  'http://143.198.162.71:3000/verify' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "Image_URL1": "https://www.biography.com/.image/ar_1:1%2Cc_fill%2Ccs_srgb%2Cfl_progressive%2Cq_auto:good%2Cw_1200/MTc5NjIyODM0ODM2ODc0Mzc3/dwayne-the-rock-johnson-gettyimages-1061959920.jpg",
  "Image_URL2": "http://www.photogriffon.com/photos-du-monde/DWAYNE-JOHNSON-THE-ROCK-th-best-pictures-ses-plus-belles-photos/DWAYNE-JOHNSON-THE-ROCK-photo-man-elected-the-most-sexy-the-world-1.jpg"
}'
```

Request URL

```
http://143.198.162.71:3000/verify
```

Server response

Code Details

200

Response body

```
{
  "isIdentical": true,
  "confidence": 0.53346
}
```

Response headers

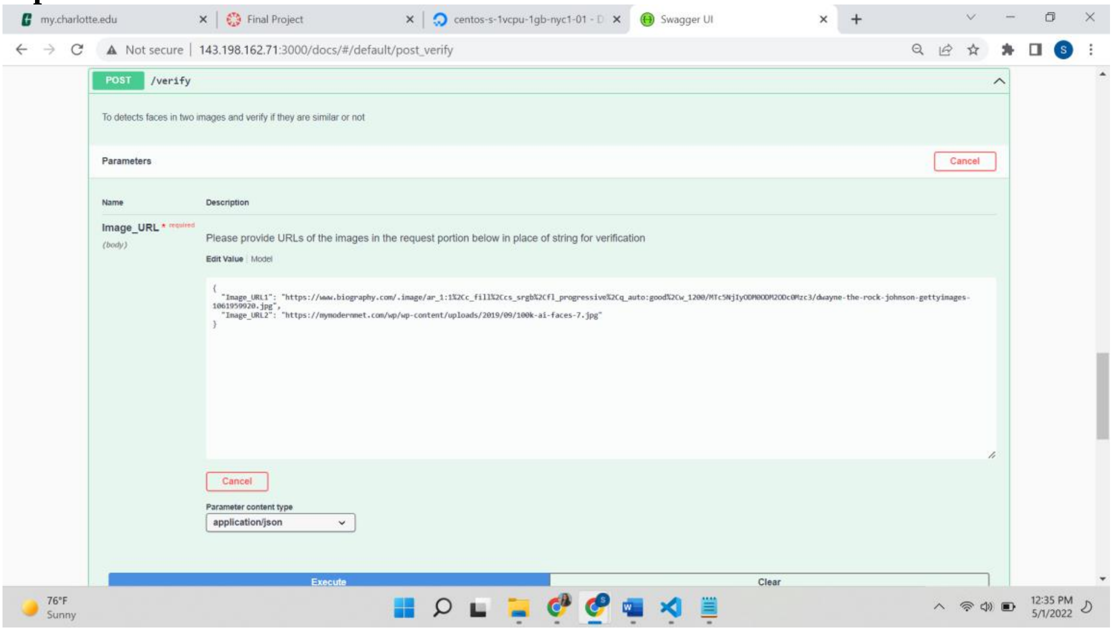
```
access-control-allow-origin: *
connection: keep-alive
content-length: 41
content-type: application/json; charset=utf-8
date: Sun,01 May 2022 16:34:48 GMT
etag: W/"29-QpzbOdCSTSW1lM/kZ0ito+fVijw"
keep-alive: timeout=5
x-powered-by: Express
```

Responses

Code	Description
200	The detection was SUCCESSFUL and the output provides us with an array object containing information that two faces are Similar or not
400	Please provide valid input
500	Some internal server error

3) Image URL used in the input parameter is of two different people:
Url1:
https://www.biography.com/.image/ar_1:1%2Cc_fill%2Ccs_srgb%2Cfl_progressive%2Cq_auto:good%2Cw_1200/MTc5NjIyODM0ODM2ODc0Mzc3/dwayne-the-rock-johnson-gettyimages-1061959920.jpg
Url2:
https://mymodernmet.com/wp/wp-content/uploads/2019/09/100k-ai-faces-7.jpg

Input:



143.198.162.71:3000/docs/#/default/post_verify

POST /verify

To detects faces in two images and verify if they are similar or not

Parameters

Name: Image_URL * required (body)

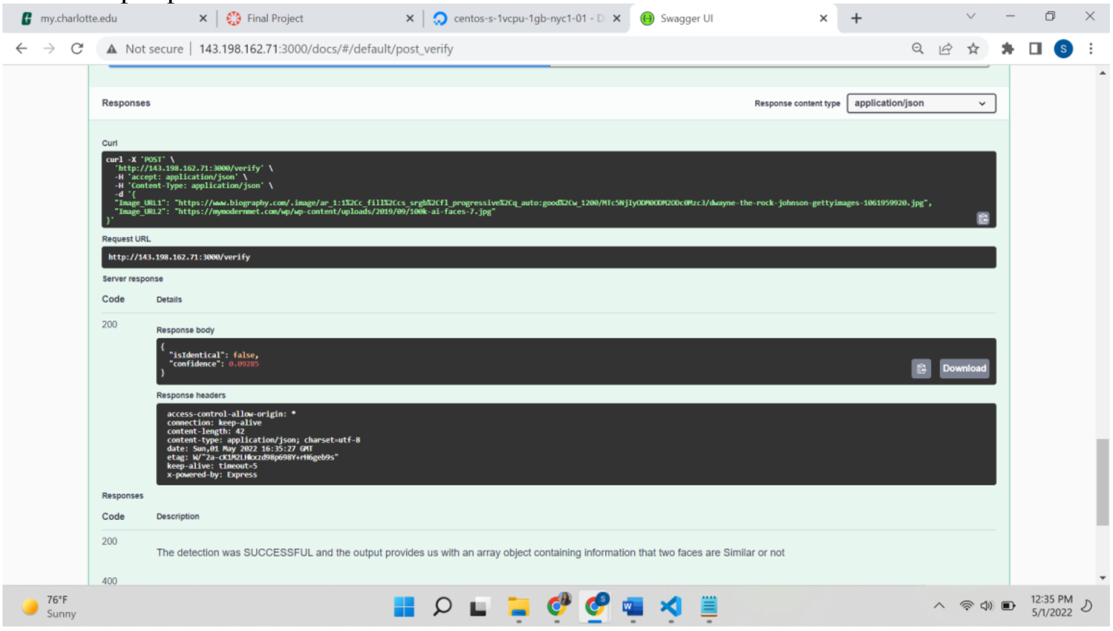
Description: Please provide URLs of the images in the request portion below in place of string for verification

Edit Value | Model

```
{
  "Image_URL1": "https://www.biography.com/.image/ar_1:1%2Cc_fill%2Ccs_srgb%2Cfl_progressive%2Cq_auto:good%2Cw_1200/MTc5NjIyODM0ODM2ODc0Mzc3/dwayne-the-rock-johnson-gettyimages-1061959920.jpg",
  "Image_URL2": "https://mymodernmet.com/wp/wp-content/uploads/2019/09/100k-ai-faces-7.jpg"
}
```

Parameter content type: application/json

Output:
In response body, we can see isIdentical = false which mean the face belongs to the different people.



143.198.162.71:3000/docs/#/default/post_verify

Responses

Response content type: application/json

Curl

```
curl -X 'POST' \
  'http://143.198.162.71:3000/verify' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "Image_URL1": "https://www.biography.com/.image/ar_1:1%2Cc_fill%2Ccs_srgb%2Cfl_progressive%2Cq_auto:good%2Cw_1200/MTc5NjIyODM0ODM2ODc0Mzc3/dwayne-the-rock-johnson-gettyimages-1061959920.jpg",
  "Image_URL2": "https://mymodernmet.com/wp/wp-content/uploads/2019/09/100k-ai-faces-7.jpg"
}'
```

Request URL

```
http://143.198.162.71:3000/verify
```

Server response

Code	Details
200	Response body

```
{
  "isIdentical": false,
  "confidence": 0.09285
}
```

Response headers

```
access-control-allow-origin: *
connection: keep-alive
content-length: 42
content-type: application/json; charset=utf-8
date: Sun,01 May 2022 16:35:27 GMT
etag: W/"2a-cK1M2LHkxzd98p698Y+rH6geb9s"
keep-alive: timeout=5
x-powered-by: Express
```

Responses

Code	Description
200	The detection was SUCCESSFUL and the output provides us with an array object containing information that two faces are Similar or not
400	