You're building an API that allows users to:

Info: please choose "a" or "b" depending on how comfortable you are with the subject (if you want you can of course do both :)

a) tag images - user submits an image url and receives set of tags describing the image (pairs of `feature: probability score`)

Request:

```
import requests

response = requests.post(
    'http://example.com/tags',
    json={
        'url': 'https://cdn.filestackcontent.com/5IkgQW1tS56RAXPnLlFG'
    }
)
```

Response:

```
{
    "tags": {
        "sail": 95,
        "sailboat": 94,
        "water transportation": 92,
        "dinghy sailing": 92,
        "sailing": 86,
        "boat": 88,
        "sailing ship": 86,
        "scow": 84,
        "yawl": 83
    }
}
```

b) detect objects in the image - user submits an image url and receives a list of detected images with corresponding bounding boxes

Request:

```
import requests
response = requests.post(
    'http://example.com/objects,
    json={
        'url': 'https://cdn.filestackcontent.com/5IkgQW1tS56RAXPnLlFG'
    }
)
```

Response:

```
{
    'objects': {
        'boat': {
            'confidence': 90,  # optional
            'bounding_box': {
                'x': 260,
                'y': 25,
                'width': 350,
                'height': 420
            },
            # crop_url is also optional, but useful for debugging :)
            'crop_url':
'https://cdn.filestackcontent.com/crop=dim:[260,25,350,420]/5IkgQW1tS56RA
XPnLlFG'
        }
        'something-else': { … }
    }
}
```

Please use one of pretrained models that are available online.
Responses given above are just an example, tags/object names/confidence scores returned by your API can of course be different.

Additional requirements (do as many as you can)

1. Include unit/functional tests
2. Add authentication for your API (JWT/basic auth/something else? Choice is yours)
3. Sometimes users can submit two different urls that contain the same image (for example https://cdn.filestackcontent.com/5IkgQW1tS56RAXPnLlFG and https://cdn.filestackcontent.com/JVAunkzQoGtxyWkrucej).
   Add a caching mechanism that will allow your API to skip unnecessary image processing and just return cached result.
4. Prepare a Dockerfile so that it's easy to build your app and run it as a docker container