ASSIGNMENT – 2

|  |  |
| --- | --- |
| Student Name | Vedharathanabai E |
| Student Roll Number | 811719104112 |

**STEP 1** : Create an Open ID Connect config file

Create a new file named client\_secrets.json in the root of our project folder and insert the following code

{

"web": {

"client\_id": "{{ OKTA\_CLIENT\_ID }}",

"client\_secret": "{{ OKTA\_CLIENT\_SECRET }}",

"auth\_uri": "{{ OKTA\_ORG\_URL }}/oauth2/default/v1/authorize",

"token\_uri": "{{ OKTA\_ORG\_URL }}/oauth2/default/v1/token",

"issuer": "{{ OKTA\_ORG\_URL }}/oauth2/default",

"userinfo\_uri": "{{ OKTA\_ORG\_URL }}/oauth2/default/userinfo",

"redirect\_uris": [

"<http://localhost:5000/oidc/callback>"

]

}

}

**STEP 2** : Configure Flask-OIDC  
  
Open up app.py and paste in the following code

from flask import Flask, render\_template

from flask\_oidc import OpenIDConnect

app = Flask(\_\_name\_\_)

app.config["OIDC\_CLIENT\_SECRETS"] = "client\_secrets.json"

app.config["OIDC\_COOKIE\_SECURE"] = False

app.config["OIDC\_CALLBACK\_ROUTE"] = "/oidc/callback"

app.config["OIDC\_SCOPES"] = ["openid", "email", "profile"]

app.config["SECRET\_KEY"] = "{{ LONG\_RANDOM\_STRING }}"

oidc = OpenIDConnect(app)

@app.route("/")

def index():

return render\_template("index.html")

@app.route("/dashboard")

def dashboard():

return render\_template("dashboard.html")

**STEP 3** : Inject the User into each request  
  
Open up app.py and paste in the following code  
  
from flask import Flask, render\_template, g

from flask\_oidc import OpenIDConnect

from okta import UsersClient

app = Flask(\_\_name\_\_)

app.config["OIDC\_CLIENT\_SECRETS"] = "client\_secrets.json"

app.config["OIDC\_COOKIE\_SECURE"] = False

app.config["OIDC\_CALLBACK\_ROUTE"] = "/oidc/callback"

app.config["OIDC\_SCOPES"] = ["openid", "email", "profile"]

app.config["SECRET\_KEY"] = "{{ LONG\_RANDOM\_STRING }}"

app.config["OIDC\_ID\_TOKEN\_COOKIE\_NAME"] = "oidc\_token"

oidc = OpenIDConnect(app)

okta\_client = UsersClient("{{ OKTA\_ORG\_URL }}", "{{ OKTA\_AUTH\_TOKEN }}")

@app.before\_request

def before\_request():

if oidc.user\_loggedin:

g.user = okta\_client.get\_user(oidc.user\_getfield("sub"))

else:

g.user = None

@app.route("/")

def index():

return render\_template("index.html")

@app.route("/dashboard")

def dashboard():

return render\_template("dashboard.html")

What we’re doing here is importing the okta Python library, and using it to define the okta\_client object. This client object will be used to retrieve a robust User object that you can use to:

* Identify the currently logged in user
* Make changes to the user’s account
* Store and retrieve user information

@app.before\_request

def before\_request():

if oidc.user\_loggedin:

g.user = okta\_client.get\_user(oidc.user\_getfield("sub"))

else:

g.user = None

The code above is where the magic happens. This function will be

executed each time a user makes a request to view a page on the site

before the normal view code runs.

**STEP 4** : Enable User Registration, Login and Logout  
  
Open up app.py and insert the following code  
  
rom flask import Flask, render\_template, g, redirect, url\_for

from flask\_oidc import OpenIDConnect

from okta import UsersClient

app = Flask(\_\_name\_\_)

app.config["OIDC\_CLIENT\_SECRETS"] = "client\_secrets.json"

app.config["OIDC\_COOKIE\_SECURE"] = False

app.config["OIDC\_CALLBACK\_ROUTE"] = "/oidc/callback"

app.config["OIDC\_SCOPES"] = ["openid", "email", "profile"]

app.config["SECRET\_KEY"] = "{{ LONG\_RANDOM\_STRING }}"

app.config["OIDC\_ID\_TOKEN\_COOKIE\_NAME"] = "oidc\_token"

oidc = OpenIDConnect(app)

okta\_client = UsersClient("{{ OKTA\_ORG\_URL }}", "{{ OKTA\_AUTH\_TOKEN }}")

@app.before\_request

def before\_request():

if oidc.user\_loggedin:

g.user = okta\_client.get\_user(oidc.user\_getfield("sub"))

else:

g.user = None

@app.route("/")

def index():

return render\_template("index.html")

@app.route("/dashboard")

@oidc.require\_login

def dashboard():

return render\_template("dashboard.html")

@app.route("/login")

@oidc.require\_login

def login():

return redirect(url\_for(".dashboard"))

@app.route("/logout")

def logout():

oidc.logout()

return redirect(url\_for(".index"))

Test Our New Flask App  
  
Now that your app is fully built, go test it out! Open up [http://localhost:5000](http://localhost:5000/), create an account, log in, etc.

<https://developer.okta.com/assets-jekyll/flask-simple-user-registration-and-login/app-usage-5c09e769946882c8dffcf556df4a029ff87106096cdca87d4f94a0fc28ec569b.gif>