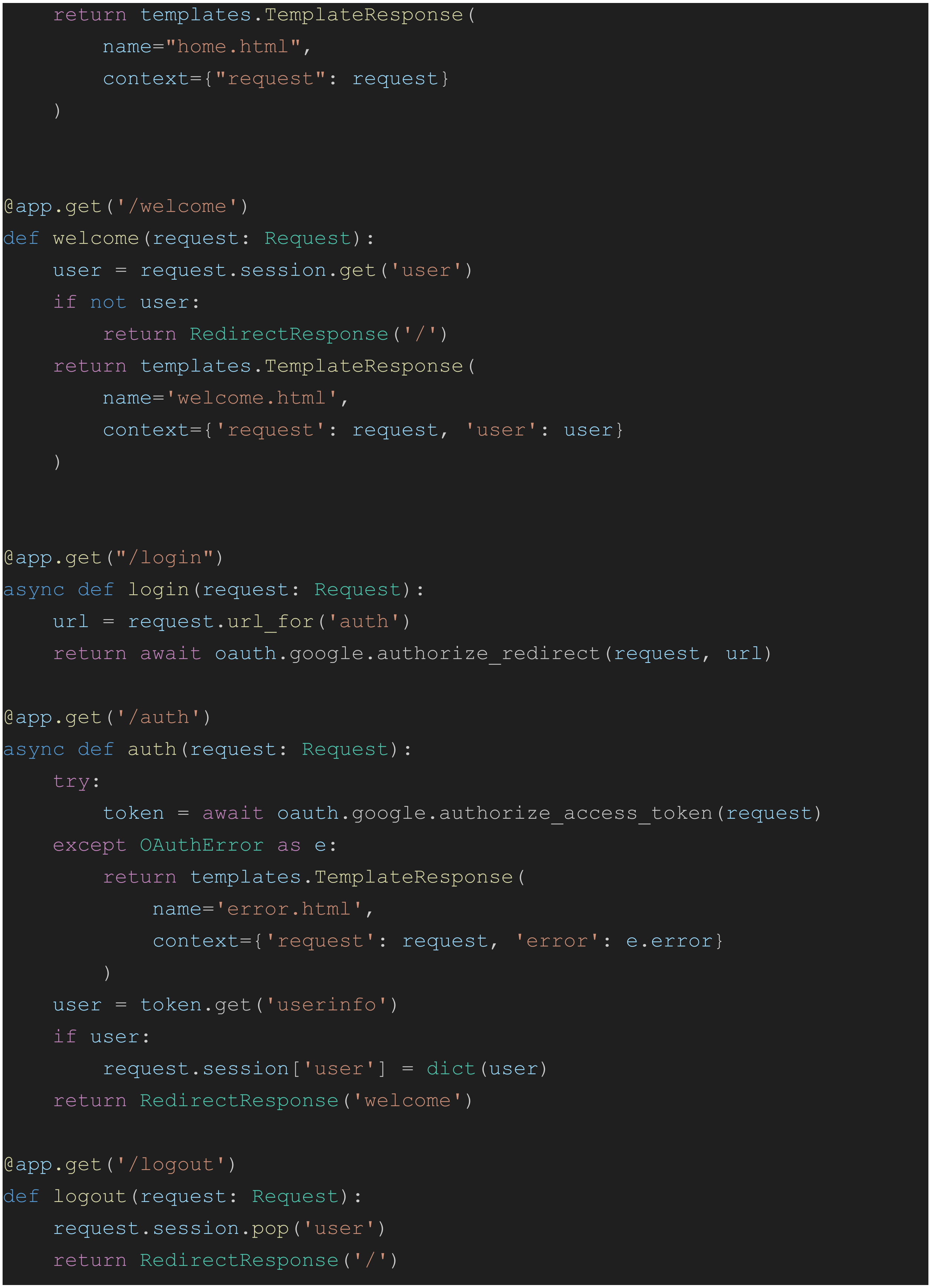
SOURCE CODE:

main.py

```
from fastapi import FastAPI
from fastapi.templating import Jinja2Templates
from starlette.requests import Request
from starlette.responses import RedirectResponse
from starlette.middleware.sessions import SessionMiddleware
from authlib.integrations.starlette_client import OAuth, OAuthError
from .config import CLIENT_ID, CLIENT_SECRET
from fastapi.staticfiles import StaticFiles

app = FastAPI()
app.add_middleware(SessionMiddleware, secret_key="add any string...")
app.mount("/static", StaticFiles(directory="static"), name="static")

oauth = OAuth()
oauth.register(
    name='google',

server_metadata_url='https://accounts.google.com/.well-known/openid-config
uration',
    client_id=CLIENT_ID,
    client_secret=CLIENT_SECRET,
    client_kwargs={
        'scope': 'email openid profile',
        'redirect_url': 'http://localhost:8000/auth'
    }
)


templates = Jinja2Templates(directory="templates")


@app.get("/")
def index(request: Request):
    user = request.session.get('user')
    if user:
        return RedirectResponse('welcome')
```



```
    return templates.TemplateResponse(
        name="home.html",
        context={"request": request}
    )


@app.get('/welcome')
def welcome(request: Request):
    user = request.session.get('user')
    if not user:
        return RedirectResponse('/')
    return templates.TemplateResponse(
        name='welcome.html',
        context={'request': request, 'user': user}
    )


@app.get("/login")
async def login(request: Request):
    url = request.url_for('auth')
    return await oauth.google.authorize_redirect(request, url)

@app.get('/auth')
async def auth(request: Request):
    try:
        token = await oauth.google.authorize_access_token(request)
    except OAuthError as e:
        return templates.TemplateResponse(
            name='error.html',
            context={'request': request, 'error': e.error}
        )
    user = token.get('userinfo')
    if user:
        request.session['user'] = dict(user)
    return RedirectResponse('welcome')

@app.get('/logout')
def logout(request: Request):
    request.session.pop('user')
    return RedirectResponse('/')
```

config.py

```
import os
from dotenv import load_dotenv

load_dotenv()

CLIENT_ID = os.environ.get('client-id', None)
CLIENT_SECRET = os.environ.get('client-secret', None)
```

Templates

error.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
    <title>Error</title>
  </head>
  <body>
    <h2>{{error}}</h2>
  </body>
</html>
```

home.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Home Page</title>
    <link rel="stylesheet" href="{{ url_for('static',
path='css/style.css') }}" />
  </head>
  <body>
```

```
    <div class="container">
      <div class="header">
        <h4>FastAPI with Google Oauth</h4>
        <div class="img-box">
          <img class="img" src="{{ url_for('static',
path='icons/avatar.png') }}" />
        </div>
      </div>
      <a href="/login" class="google-btn">
        <div class="google-icon"></div>
        <span class="btn-text">Signin with Gmail</span>
      </a>
    </div>
  </body>
</html>
```

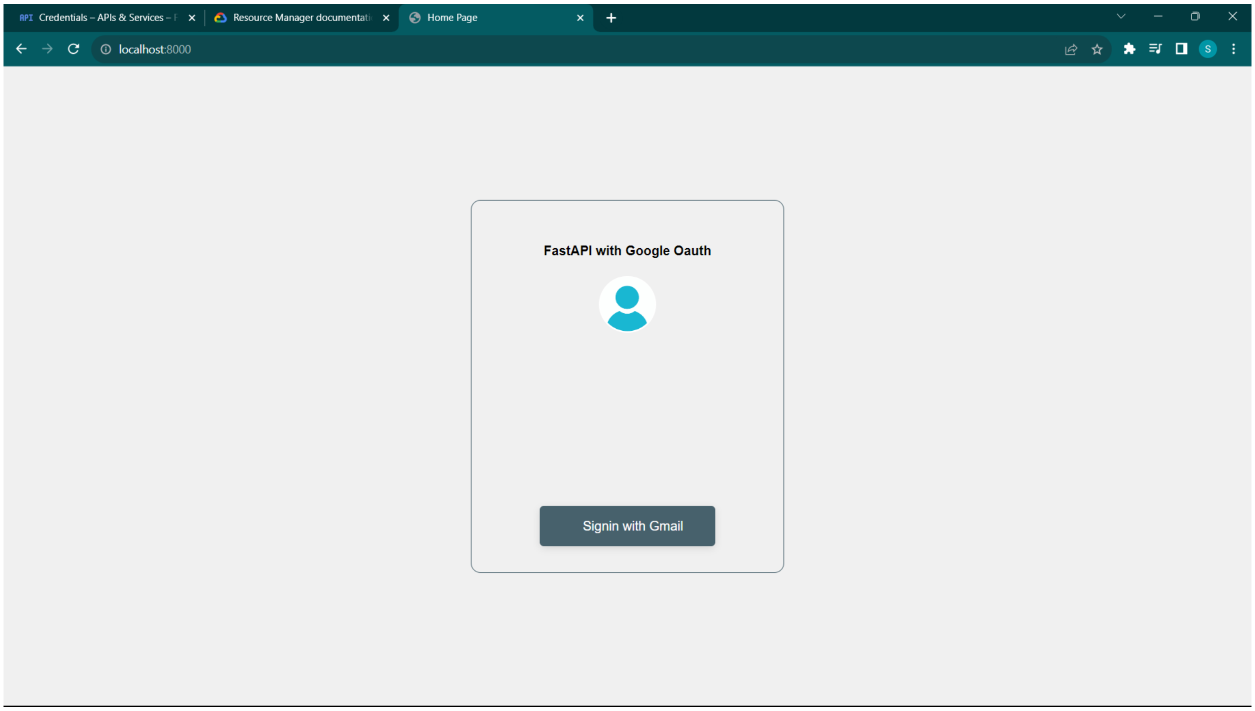
welcome.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Home Page</title>
    <link rel="stylesheet" href="{{ url_for('static',
path='css/style.css') }}" />
  </head>
  <body>
    <div class="container">
      <div class="header">
        <h4>Welcome, {{user['family_name']}}</h4>
        <div class="img-box">
          <img class="img" src="{{user['picture']}}" />
        </div>
      </div>
      <a href="/logout" class="google-btn">
        <div class="google-icon"></div>
        <span class="btn-text">Logout</span>
      </a>
    </div>
  </body>
</html>
```

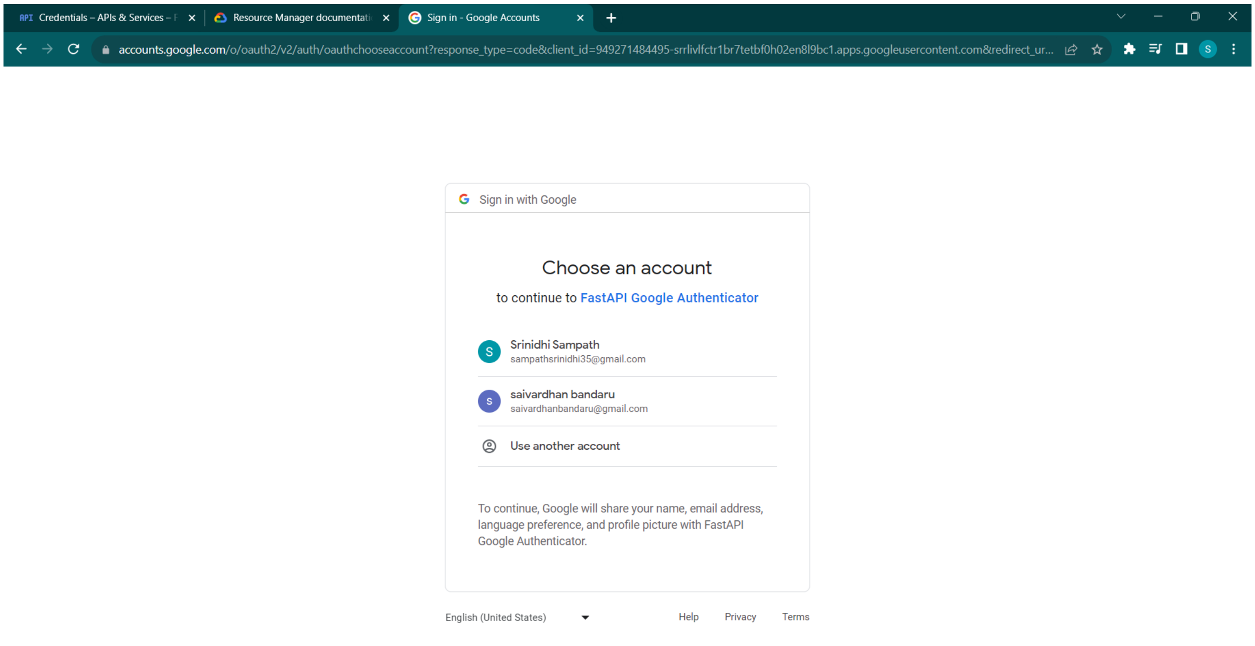
main.py

```
import uvicorn

if __name__ == "__main__":
    uvicorn.run(
        app="app.main:app",
        host="localhost",
        port=8000,
        reload=True
    )
```

RESULTS:


Credentials – APIs & Services – F
Resource Manager documentati
Home Page
localhost:8000

FastAPI with Google Oauth


Credentials – APIs & Services – F
Resource Manager documentati
Sign in - Google Accounts
accounts.google.com/o/oauth2/v2/auth/oauthchooseaccount?response_type=code&client_id=949271484495-srrlivlfctr1br7tetbf0h02en8l9bc1.apps.googleusercontent.com&redirect_ur...
Sign in with Google

Choose an account

to continue to FastAPI Google Authenticator

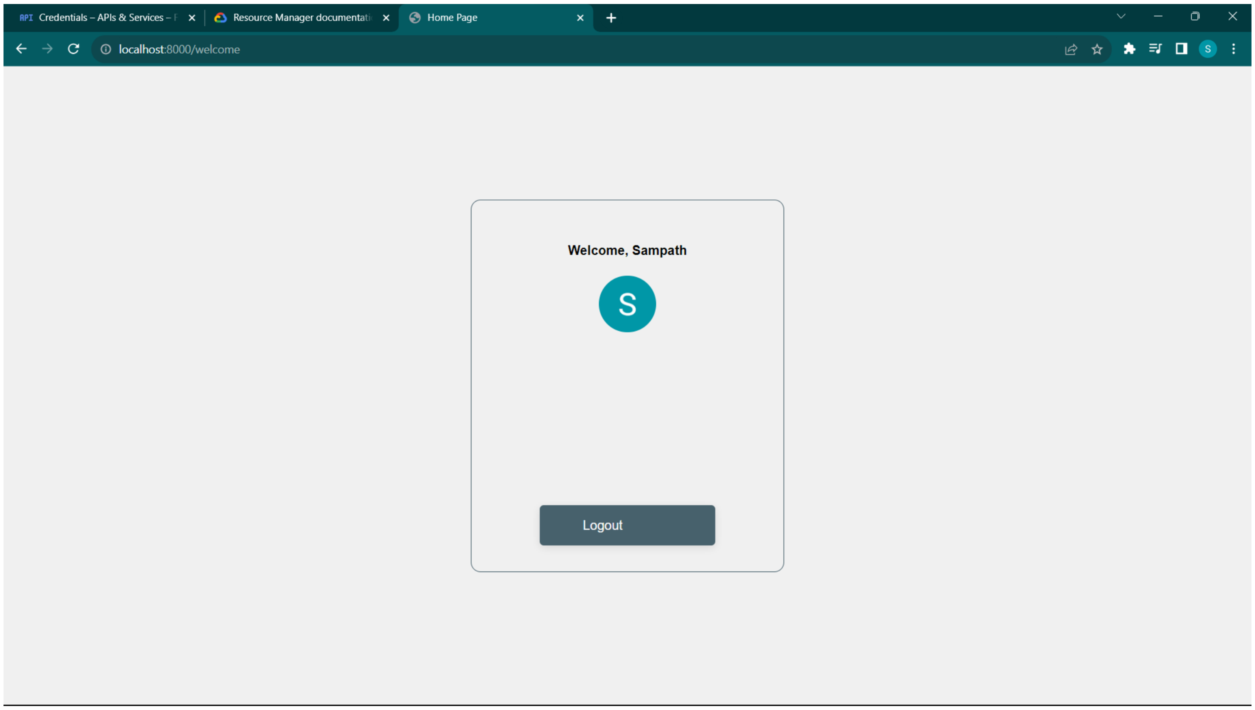
Srinidhi Sampath
sampathsrinidhi35@gmail.com

saivardhan bandaru
saivardhanbandaru@gmail.com

Use another account

To continue, Google will share your name, email address, language preference, and profile picture with FastAPI Google Authenticator.

English (United States)
Help
Privacy
Terms


Credentials – APIs & Services – F
Resource Manager documentati
Home Page
localhost:8000/welcome
S
Logout

Welcome, Sampath

REQUIREMENTS:

- Google Cloud
- Google API
- Fast API
- Uvicorn
- Authlib
- Dotenv
- Jinja