

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

پروژه درس مهندسی اینترنت

استاد : مهندس امیرکیوان شفیعی

دانشجو : محمدیان

موضوع :

اپلیکیشن وب رسیانسیو و پویا با سیستم احراز هویت گوگل
و قابلیت ارسال و تعامل با شبکه های اجتماعی در جهت تعامل
با کاربر

- ۴.....: چکیده پروژه
- ۷.....: احراز هویت با گوگل ، ورود به برنامه
- ۱۳.....: ارسال مطالب در شبکه های اجتماعی
- ۱۵.....: توضیحات برنامه و توابع
- ۱۶.....: وایر فریم و فرانت اند
- ۱۸.....: رسیانسیو سازی

چکیده پروژه :

Flask یک فریمورک لایه بالا برای زبان برنامه نویسی پایتون است که برای ساخت وب سایت ها و وب اپلیکیشن ها استفاده می شود. این فریمورک سبک و منعطف است و کنترل کاملی بر معماری MVC ارائه می دهد. همچنین اجازه استفاده از موتور قالب Jinja2 را می دهد که امکان سفارشی سازی زیادی را فراهم می کند.

یکی از ویژگی های مهم یک وب سایت کاربر محور، روش احراز هویت کاربران است. احراز هویت از طریق گوگل یکی از روش های محبوب و امن است که به کاربران اجازه می دهد با استفاده از حساب گوگل خود وارد سیستم شوند بدون نیاز به نام کاربری و کلمه عبور جداگانه. این روش با کاهش موانع ورود و افزایش امنیت، تجربه کاربری را بهبود می بخشد.

رشد شبکه های اجتماعی باعث شده کاربران بیشتر از قبل خواستار امکان اشتراک گذاری محتوا با دیگران در شبکه هایی مثل توییتر، فیسبوک و اینستاگرام باشند. ادغام API های شبکه های اجتماعی در فلسک نه تنها این فرایند را برای کاربران آسان می کند، بلکه راه مؤثری برای افزایش دامنه دید و اشتراک گذاری محتوا برای توسعه دهندگان ایجاد می کند. این برنامه با استفاده از فریمورک Flask و پایگاه داده SQLAlchemy توسعه یافته است. برای ورود از طریق حساب گوگل از OAuth2.0 و OpenID Connect استفاده شده است. برنامه از Flask-Login برای مدیریت کاربران و احراز هویت و Flask-Migrate برای مدیریت پایگاه داده بهره می برد.

این برنامه رابط یک برنامه وب با قابلیت ورود از طریق حساب گوگل است. کاربران پس از ورود موفق می توانند به صفحه اصلی برنامه منتقل شوند. در این برنامه امکان اشتراک گذاری از طریق تلگرام و واتساپ نیز وجود دارد.

توسعه وب اپلیکیشن های پویا و رسپانسیو که قادر به احراز هویت از طریق سرویس های اجتماعی محبوبی مانند گوگل باشند و همچنین به کاربران امکان ارتباط و اشتراک گذاری محتوا در شبکه های اجتماعی را بدهند، امروزه بخشی ضروری از توسعه نرم افزار در دنیای امروز است. این نوع از اپلیکیشن ها که به کاربران اجازه می دهد با استفاده از اطلاعات حساب کاربری گوگل وارد شوند، ارتباط و تعامل را آسان تر می کنند و تجربه کاربری بهتری ارائه می دهند.

در ادامه این مستندات به بررسی کدهای Flask ، که برای ساخت یک اپلیکیشن وب رسپانسیو و پویا با امکان احراز هویت از طریق حساب کاربری گوگل و همچنین ارسال و اشتراک گذاری اطلاعات کاربر از طریق شبکه های اجتماعی استفاده شده اند، می پردازیم. برنامه به کاربران این امکان را می دهد تا با استفاده از حساب کاربری گوگل خود به سیستم وارد شوند و سپس اطلاعات مورد نظرشان را از طریق تلگرام و واتساپ به اشتراک بگذارند.

در ادامه ، به عمیق ترین جزئیات فنی می پردازیم. ابتدا، کتابخانه ها و ماژول های لازم برای اجرای اپلیکیشن از جمله Flask برای ایجاد یک برنامه تحت وب ، flask_login برای مدیریت جلسات کاربران، و flask_migrate برای مدیریت مهاجرت های پایگاه داده فراخوانی می شوند.

پس از پیکربندی اولیه اپلیکیشن و معرفی Blueprint برای مسیریابی درخواست‌ها، به سراغ احراز هویت حساب‌های گوگل می‌رویم. با استفاده از کتابخانه‌های مرتبط با OAuth 2.0، پروسه احراز هویت توضیح داده می‌شود تا در نهایت، بعد از ورود موفقیت آمیز، وضعیت و داده‌های کاربر در جلسه وب (session) ذخیره شوند.

همچنین، فرآیند ثبت نام کاربران جدید که در پایگاه داده ذخیره می‌شوند با استفاده از یک کلمه عبور تصادفی رمزگذاری شده تشریح می‌گردد. این اطمینان را برای مدیریت گذر واژه‌ها و حفظ امنیت حساب کاربران فراهم می‌کند.

در نهایت، به توضیح ویژگی‌های اشتراک‌گذاری از طریق تلگرام و واتساپ می‌پردازیم. کدهای خاص در این جا به بررسی نحوه ساخت پیام‌هایی می‌پردازند که حاوی اطلاعات کاربر هستند و چگونه از طریق URL های مخصوص هر پلتفرم شبکه اجتماعی به اشتراک گذاشته می‌شوند.

این مستند به طور خلاصه، تصویری کلی پیرامون اجزای مختلف و کدهایی که برای ایجاد یک تجربه کاربری روان و بی دردسر برای ورود و تعامل کاربران با برنامه‌های وب مورد استفاده قرار می‌گیرد، ارائه می‌دهد. این برنامه به عنوان یک مثال عالی از تلفیق فناوری‌ها و کتابخانه‌های مختلف عمل می‌کند تا تجربه کاربری جذاب و همزمان امنی را فراهم آورد.

احراز هویت با گوگل ، ورود به برنامه :

مراحل احراز هویت و ورود به برنامه وب ما توسط تابع login_bygoogle و callback در برنامه route.py انجام می گیرد که در زیر بخش کد برنامه قرار گرفته است.

```
@routing.route('/loginbygoogle')
def login_bygoogle():

    authorization_url, state = flow.authorization_url()
    session["state"] = state
    return redirect(authorization_url)

@routing.route('/callback')
def callback():

    flow.fetch_token(authorization_response=request.url)

    if not session["state"] == request.args["state"]:
        abort(500)

    credentials = flow.credentials
    request_session = requests.session()
    cached_session = cachecontrol.CacheControl(request_session)
    token_request =
google.auth.transport.requests.Request(session=cached_session)

    id_info = id_token.verify_oauth2_token(
        id_token=credentials._id_token,
        request=token_request,
        audience=GOOGLE_CLIENT_ID

    )
    session["google_id"] = id_info.get("sub")
    session["name"] = id_info.get("name")
```

```

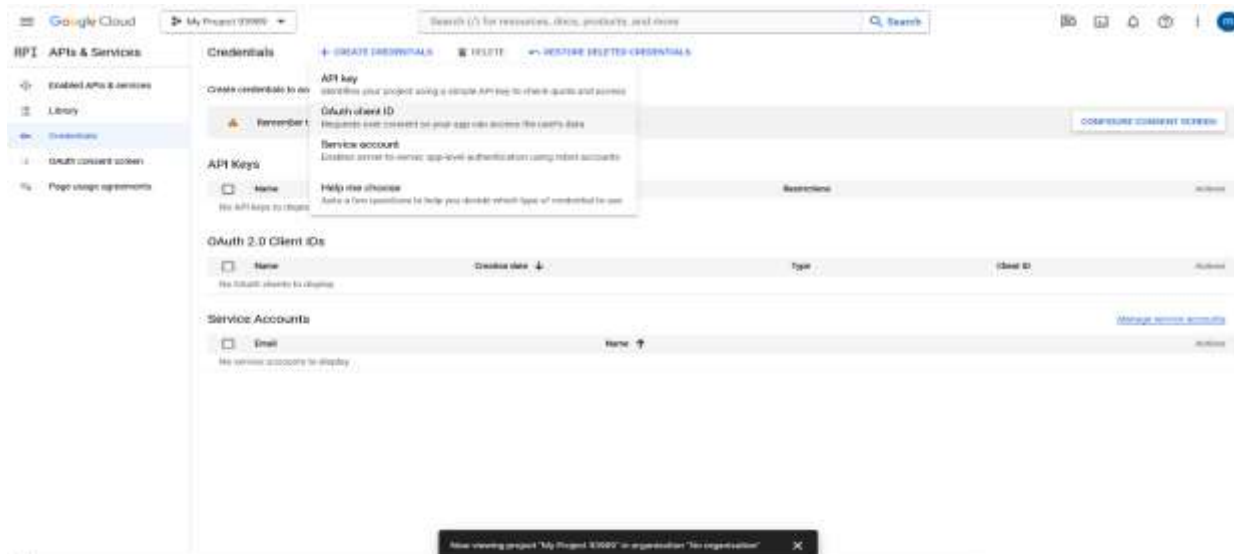
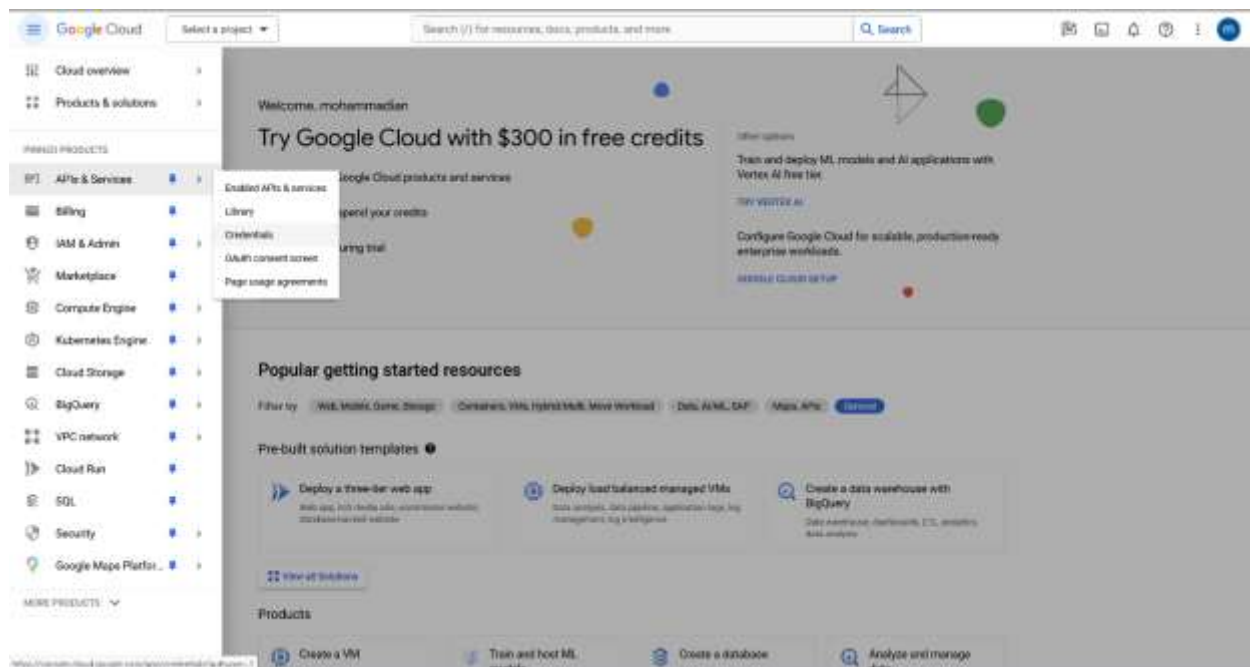
session["email"] = id_info.get("email")
user = User.query.filter_by(email=session["email"]).first()
if not user:
    plain_password = ''.join(random.choice(string.ascii_letters) for i in
range(10))
    hashed_password = generate_password_hash(plain_password,
method='pbkdf2:sha256')

    new_user = User(
        email=session["email"],
        first_name=session["name"],
        password=hashed_password
    )
    db.session.add(new_user)
    db.session.commit()
    user = new_user
flash('Logged in successfully!', category='success')
login_user(user, remember=True)
return redirect(url_for('routing.home'))

```

برای اینکه بتوان مراحل احراز هویت را کامل نمود نیاز به توکن گوگل برای دسترسی می باشد که این توکن را می توان از Google Console Cloud بدست آورد، مراحل کامل در زیر شرح داده شده است:

۱. تنظیم پروژه در Google Cloud Platform : قبل از شروع، باید یک پروژه در Google Cloud Platform ایجاد کنید و API های لازم را فعال کنیم. سپس باید اعتبارنامه های OAuth 2.0 را ایجاد کنیم که شامل client_id و client_secret است.



۲. ایجاد URL برای احراز هویت: برنامه‌ی ما باید یک URL برای احراز هویت کاربر ایجاد کند. به این صورت که Google Flow را با استفاده از client_id، client_secret، حوزه‌های دسترسی مورد نظر (scopes) مانند userinfo.email و userinfo.profile، و redirect_uri که نشانی است که گوگل پس از احراز هویت کاربر به آن داده‌ها را ارسال می‌کند، پیکربندی می‌کنیم.

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

i The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

Authorized JavaScript origins **i**

For use with requests from a browser

+ ADD URI

Authorized redirect URIs **i**

For use with requests from a web server

URIs 1 *

http://127.0.0.1:5000/callback

URIs 2 *

https://127.0.0.1:5000/callback



+ ADD URI

Note: It may take 5 minutes to a few hours for settings to take effect

CREATE

CANCEL

Now viewing project "My Project 12290" in organization "No organization"



OAuth consent screen — Scopes — Test users — 4 Summary

OAuth consent screen

EDIT

User type

External

App name

احراز هویت گوگل

Support email

mohammadian8787@gmail.com

App logo

Not provided

Application homepage link

Not provided

Application privacy policy link

Not provided

Application terms of service link

Not provided

Authorized domains

Not provided

Contact email addresses

mohammadian8787@gmail.com

Now viewing project "My Project 93989" in organization "No organization"

X

۳. انتقال کاربر به URL احراز هویت گوگل : کاربر را به URL خود کار ایجاد شده توسط Google Flow هدایت می کنیم. کاربر باید وارد حساب کاربری گوگل خود شود و اجازه دسترسی به اطلاعات خود را بدهد.

۴. دریافت کد موقت : پس از اجازه دادن توسط کاربر، گوگل کاربر را به `redirect_uri` باز می گرداند و در درخواست، یک `code` موقت درج می کند.

۵. دریافت توکن از گوگل : اکنون برنامه می باید با استفاده از کد موقت دریافتی، یک درخواست به سرور گوگل ارسال کند تا توکن های دسترسی (`access token`) و تازه سازی (`refresh token`) را دریافت کند.

۶. استفاده از توکن برای دریافت اطلاعات کاربر: با استفاده از توکن دریافتی، درخواست دیگری به گوگل ارسال می‌کنیم تا اطلاعات پروفایل کاربر مانند نام، ایمیل و شناسه یکتا را دریافت نماییم.

۷. احراز هویت در برنامه و ایجاد جلسه کاربری: پس از دریافت اطلاعات کاربر، برنامه‌ی ما باید این داده‌ها را بررسی و تایید کند. اگر کاربر در پایگاه داده وجود نداشته باشد، یک حساب جدید برای او ایجاد شده و اگر وجود داشته باشد، جلسه‌ای (session) برای او آغاز می‌شود تا بدون نیاز به ورود مجدد، از سرویس‌های برنامه استفاده کند.

تمامی این مراحل با رعایت استانداردها و امنیت اطلاعات کاربران انجام می‌شود تا فرآیند ورود به وب اپلیکیشن هم برای توسعه‌دهنده و هم برای کاربران، راحت و امن باشد.



ارسال مطالب در شبکه های اجتماعی :

این برنامه فلسک دو تابع برای اشتراک گذاری محتوا در شبکه های اجتماعی تلگرام و واتساپ دارد:

تابع: share_on_telegram

این تابع یک پیام متنی درباره نام کاربر و ایمیل وی تولید می کند و آن را به همراه لینک به صفحه اصلی وبسایت کد گذاری می کند. سپس URL اشتراک گذاری در تلگرام را تولید می کند که شامل متن کد گذاری شده و لینک کد گذاری شده است. در نهایت کاربر را به این URL هدایت می کند تا محتوا را در تلگرام به اشتراک بگذارد.

```
@routing.route('/share_on_telegram')
@login_required
def share_on_telegram():
    message = f"نام صاحب اکانت : {current_user.first_name}\nایمیل: {current_user.email}\nمی توانید در این قسمت اطلاعات ورود کننده به حساب را مشاهده کنید و این اطلاعات را به اشتراک بگذارید."
    encoded_message = quote_plus(message)

    url_to_share = url_for('routing.home', _external=True)

    telegram_url =
    f"https://t.me/share/url?url={quote_plus(url_to_share)}&text={encoded_message}"

    return redirect(telegram_url)
```

تابع: share_on_whatsapp

این تابع نیز یک پیام مشابه درباره نام کاربر و ایمیل وی و همچنین لینک به صفحه اصلی تولید می کند. سپس URL اشتراک گذاری در واتساپ را تولید می کند که شامل متن کدگذاری شده و لینک است. در نهایت کاربر را به این URL هدایت می کند تا محتوا را در واتساپ به اشتراک بگذارد.

این توابع امکان اشتراک گذاری ساده محتوا در دو شبکه اجتماعی پرتعداد را فراهم می کنند.

```
@routing.route('/share_on_whatsapp')
def share_on_whatsapp():

    message = f" نام صاحب اکانت \n می توانید اطلاعات مربوط به ورود و نام کاربری را در این لینک مشاهده کنید "
    {current_user.first_name}\nایمیل: {current_user.email}"
    encoded_message = quote_plus(message)
    url_to_share = url_for('routing.home', _external=True)

    url_to_share = request.args.get('url', 'http://127.0.0.1:5000')
    whatsapp_url =
    f"https://wa.me/?text={quote_plus(url_to_share)}&text={encoded_message}"
    return redirect(whatsapp_url)
```

توضیحات برنامه و توابع :

در این برنامه ابتدا app از فلسک ایجاد می شود.

سپس routing که یک Blueprint است تعریف می شود تا مسیرها را مدیریت کند.

متغیرهای محیطی برای اجازه ارتباطات غیر امن تنظیم می شود و Migrate برای مدیریت پایگاه داده ایجاد می شود.

مسیرهای زیر تعریف می شوند:

(/): صفحه اصلی که نیاز به ورود کاربر دارد.

(/login): صفحه ورود کاربر

(/loginbygoogle): برای ورود کاربر از طریق گوگل

(/callback): برای گرفتن توکن دسترسی از گوگل و ایجاد/ورود کاربر

(/logout): برای خروج کاربر

(/share_on_telegram): برای اشتراک گذاری در تلگرام

(/share_on_whatsapp): برای اشتراک گذاری در واتساپ

در تابع callback پس از دریافت توکن از گوگل، اطلاعات کاربر استخراج می شود و یک کاربر جدید ایجاد یا کاربر موجود وارد می شود. سپس کاربر به صفحه اصلی هدایت می شود.

در تابع logout کاربر از سیستم خارج می شود.

در تابع home صفحه اصلی نمایش داده می شود.

وایر فریم و فرانت اند :

وایر فریم (Wireframe) یک نسخه ساده و ابتدایی از طراحی یک صفحه وب است که به عنوان نقشه‌ای برای ارائه چگونگی چیدمان عناصر در صفحه، استفاده می‌شود. وایر فریم‌ها معمولاً رنگ، گرافیک یا استایل دقیق را نمایش نمی‌دهند، بلکه تمرکز آن‌ها بر روی ساختار و کارکرد است.

برای مثال:

در وایرفریم صفحه `login.html` ، محل قرارگیری لوگو، فیلد ورود، دکمه ورود با اکانت گوگل و سایر عناصر را بدون جزئیات طراحی از قبیل رنگ یا فونت مشخص می کنیم. هدف این است که درک کلی از چیدمان صفحه ارائه شود.

در مقابل، فرانت اند (Front-end) به بخشی از وبسایت گفته می شود که کاربران با آن تعامل دارند. این شامل طراحی رابط کاربری، طراحی گرافیک، انیمیشن ها و همچنین کدهای HTML ، CSS و جاوااسکریپت است.

برای `login.html` ، فرانت اند شامل موارد زیر خواهد بود:

۱: `HTML`. ساختار اصلی صفحه را مشخص می کند.

۲: `CSS`. استایل دهی به صفحه، مانند قرار دادن رنگ ها، فونت ها و دیگر عناصر طراحی.

۳. تصاویر و گرافیک ها: شامل لوگو و سایر عناصر تصویری که به صورت استاتیک و یا با استفاده از CSS ساخته شده اند.

در نمونه فرانت‌اند برای `home.html` و `base.html`، از مفاهیم فرانت‌اند مانند ارث‌بری قالب‌ها استفاده از `{% extends "base.html" %}` برای ساخت صفحات داینامیک، استفاده از جینجا تمپلیت‌ها مانند `{{ user.first_name }}` برای نمایش داده‌ها و همچنین پیوندها به سایر صفحات از طریق `url_for` در فلسک استفاده کرده‌ایم.

ریسپانسیو سازی :

ریسپانسیو سازی یک وبسایت به معنی طراحی صفحات به شکلی است که در دستگاه‌های مختلف با اندازه‌های مختلف نمایشگر به خوبی قابل مشاهده باشند. در ریسپانسیو سازی موارد زیر اهمیت دارند:

۱. استفاده از واحدهای سنجش مثل درصد یا `em` برای اندازه‌گذاری که نسبت به مادر خود مقیاس‌پذیر هستند.

۲. استفاده از تکنیک‌های فلکس باکس یا گرید برای چیدمان عناصر.

۳. استفاده از `Media Queries` در `CSS` برای تنظیم استایل‌ها بسته به اندازه نمایشگرها.

۴. تصاویر و الگوهای پس‌زمینه‌ای که در اندازه‌های مختلف قابل تطبیق هستند.

۵. استفاده از فریم‌ورک‌های CSS مانند Bootstrap که از پیش ریسپانسیو ساخته شده‌اند.

```
@media (min-width: 600px) {  
  .menu-icon {  
    display: none;  
  }  
}  
  
@media (max-width: 600px) {  
  .navigation {  
    display: none;  
  }  
  
  .menu-icon {  
    display: block;  
  }  
  
  .navigation--mobile {  
    top: var(--navbar-height);  
    position: absolute;  
    right: 0px;  
    display: flex;  
    padding: 80px 60px;  
    min-height: 100%;  
    background-color: var(--background);  
    gap: 8px;  
    flex-direction: column;  
    align-items: flex-start;  
    opacity: 0.95;  
    animation: fadein 0.3s forwards;  
  }  
  
  @keyframes fadein {  
    0% {  
      opacity: 0;  
      width: 0;  
      height: 0;  
    }  
  }  
}
```

```
    100% {
      opacity: 1;
      width: 100%;
      height: calc(100vh - var(--navbar-height));
    }
  }

  .navigation--mobile--fadeout {
    animation: fadeout 300ms forwards;
  }

  @keyframes fadeout {
    0% {
      opacity: 1;
      width: 100%;
      height: calc(100vh - var(--navbar-height));
    }

    100% {
      opacity: 0;
      width: 0;
      height: 0;
    }
  }
}
```

تصاویر:

