



استاد : مهندس شفیعی

زهرا نور زاده

تمرین درس مهندسی اینترنت

عنوان :

وب سایت کاربر محور با احراز هویت گوگل و ارسال به شبکه های

اجتماعی با استفاده از **FLASK**

فهرست

۳ مقدمه :
۴ کتابخانه ها :
۷ برنامه نویسی :
۹ احراز هویت گوگل :
۱۲ وایر فریم و فرانت اند :

مقدمه :

در عصر حاضر، تکنولوژی های وب کاربردی و ارتباطات اجتماعی در بستر دیجیتال از جمله مؤلفه های جدایی ناپذیر زندگی روزمره مان شده اند. سکوهایی وب کاربر محور که به کاربران امکان می دهند به راحتی و با امنیت بالا وارد سیستم شده و مطالب را به اشتراک بگذارند، به نقاط اتصال ارزشمندی در فضای مجازی تبدیل گشته اند.

Flask ، یک فریم ورک وب سبک و منعطف برای زبان برنامه نویسی Python ، به عنوان ابزاری برای ساخت چنین سکوهایی استفاده می شود . در توضیحات این برنامه به بررسی چگونگی استفاده از Flask برای ایجاد این وب سایت کاربر محور با قابلیت احراز هویت از طریق گوگل و همچنین عملکرد ارسال به شبکه های اجتماعی می پردازیم.

رابط کاربری (User Interface - UI) و تجربه کاربری (User Experience - UX) در یک وب سایت، اغلب به عنوان فاکتورهای تعیین کننده در موفقیت یک پلتفرم آنلاین به حساب می آیند.

Flask با ارائه کنترل دقیق بر ساختار MVC (Model-View-Controller) و امکان استفاده از جینجا ۲ (Jinja2) برای Template Engine ، فضای زیادی برای سفارشی سازی فراهم می کند.

یکی از ویژگی های کلیدی که یک وب سایت کاربر محور باید به آن توجه داشته باشد، روش احراز هویت است. احراز هویت گوگل، یکی از روش های محبوب و امن است که به کاربران اجازه می دهد با استفاده از حساب کاربری گوگل خود بدون نیاز به ایجاد یک نام کاربری و کلمه عبور جداگانه وارد سیستم شوند. این فرآیند، که اغلب بخشی از “احراز هویت فدراسیون” نامیده می شود، با کاهش موانع ورود و ارتقاء امنیت، تجربه کاربری را به طور قابل ملاحظه ای بهبود می بخشد.

با رشد شبکه‌های اجتماعی، کاربران بیش از پیش خواستار امکان به اشتراک‌گذاری محتوا با دیگران در پلتفرم‌هایی مانند توییتر، فیسبوک، و اینستاگرام هستند. ادغام API های شبکه‌های اجتماعی در Flask نه تنها این فرآیند را برای کاربران تسهیل می‌کند، بلکه راهی مؤثر برای افزایش دامنهی دید و اشتراک‌گذاری محتوا توسعه‌دهندگان وب فراهم می‌آورد.

کتابخانه ها :

به منظور برنامه‌نویسی مؤثر بخش بک‌اند یک وب‌سایت که در عین حال کاربردی باشد و تجربه‌ای روان را برای کاربران خود فراهم آورد، استفاده از فریم‌ورک‌ها و کتابخانه‌های معتبر و مدرن الزامی است. در این پروژه، ما بر پایه فریم‌ورک Flask و همچنین چندین افزونه کاربردی از Flask از جمله:

OAuth ، Flask-Login ، Flask-SQLAlchemy، و پکیج‌های مرتبط با OAuth استاندارد گوگل به منظور احراز هویت ایمن و قابل اتکا، نهادینه‌سازی شده‌ایم. بیایید کمی عمیق‌تر به نقش هر یک از این مؤلفه‌ها و شیوه استفاده از آن‌ها در پروژه‌ی ما بپردازیم.

فایل route.py ، که هسته‌ی اصلی سیستم احراز هویت در پروژه ما را تشکیل می‌دهد، شامل چندین ماژول اساسی است که کاربرد هر یک به ترتیب زیر است:

- requests برای ارسال درخواست‌های HTTP به سرویس‌های خارجی همانند سرورهای OAuth گوگل مورد استفاده قرار می‌گیرد.
- Blueprint از Flask به منظور تعریف مسیرهای مرتبط با احراز هویت.
- render_template برای پاسخ‌دهی به درخواست‌ها با صفحه‌های HTML که توسط Jinja2 تحلیل و جایگزینی شده‌اند.
- flash و url_for از Flask برای ارائه پیغام‌های کاربر و هدایت او در وب‌سایت.
- session برای ذخیره مقادیر و اطلاعات کاربران در حین جلسات (session) وب.
- abort و redirect برای هدایت کاربران به صفحات خاص یا نمایش خطاهای HTTP.
- Flask-Login برای مدیریت جلسه‌ی کاربر و احراز هویت.
- ماژول‌های مربوط به گوگل OAuth برای فرآیند ورود به سیستم از طریق حساب‌های گوگل.
- Flask-Migrate برای مدیریت مهاجرت‌های دیتابیس.

در `init.py`، پروژهای ما را به عنوان یک بند از Flask تعریف می کنیم و پیکربندی ها و متغیرهای اولیه مورد نیاز برای کارکرد صحیح اپلیکیشن را تنظیم می کنیم:

- Flask برای ایجاد نمونه ی Flask و تنظیمات مربوط به جلسه ی کاربر.
- SQLAlchemy برای ارتباط با دیتابیس و مدیریت ORM.
- os و path برای تعریف مسیرها و پردازش متغیرهای محیطی.
- LoginManager از Flask-Login برای احراز هویت و مدیریت جلسه ها.

مدل‌های مورد استفاده در `models.py` ارتباطات و جدول‌های دیتابیس را تعریف می‌کنند:

- `SQLAlchemy db` از برای ایجاد مدل‌های دیتابیس.
- `UserMixin` از `Flask-Login` که امکان‌های پایه مربوط به کاربر (مثل فعال بودن، احراز هویت شده، آنانیموس بودن) را ارائه می‌دهد.

و نهایتاً، `app.py` که ورودی اصلی به برنامه‌ی ما است و تمامی این مؤلفه‌ها را با همدیگر ادغام می‌کند تا یک وبسایت متعامل و پاسخگو ارائه دهد.

برنامه نویسی :

در کدنویسی وب، پیشرفت‌های اخیر در فریم‌ورک‌ها و محیط‌های توسعه به ما اجازه می‌دهند تا تجربیات کاربری همه‌جانبه و تعاملی را بسازیم که سال‌ها پیش تصورش دشوار بود. در تمرین ما از ماژول‌ها و افزونه‌های غنی پایتون و `Flask` برای ساخت یک تجربه کاربری خوب استفاده شده است که احراز هویت را راحت‌تر کرده و اطلاعات را به صورت پویا ارائه می‌دهد.

با استفاده از محیط توسعه‌ی پیشرفته‌ی `vscode`، توسعه‌دهندگان می‌توانند بهره‌وری و تمرکز خود را در قسمت‌های فرانت‌اند و بک‌اند بهینه‌سازی نمایند. در ادامه، به توضیح ساختار و جزئیات بیشتری از کدهای مربوط به این پروژه می‌پردازیم.

ماژول `__init__.py` به عنوان هسته اصلی `Flask` در اپلیکیشن عمل می‌کند، جایی که اشیا مهمی مانند اشیای دیتابیس `SQLAlchemy` تعریف شده و اپلیکیشن `Flask` ساخته می‌شود.

تابع `create_app` تنها اپلیکیشن `Flask` را راه اندازی می کند بلکه از روال های پیکربندی برای تنظیم ارتباط با دیتابیس و مدیریت جلسات کاربری استفاده می کند. در اینجا نیز یک دکوراتور خاص تعریف شده که بررسی می کند کاربر قبل از دسترسی به بخش های خاصی از وبسایت احراز هویت شده باشد.

هر بخش از کد، از جمله مدل های دیتابیس و بلوپرینت های رابط کاربری تا مکانیزم های احراز هویت گوگل و اپل، با دقت طراحی شده اند تا اطمینان حاصل شود که کاربران می توانند به راحتی و بدون دغدغه از امنیت، به بخش های مختلف وبسایت دسترسی پیدا کنند. به عنوان مثال، استفاده از `Flow` در کتابخانه `google_auth_oauthlib` برای پیاده سازی فرآیند `OAuth 2.0` و احتیاط های لازم برای محافظت از متغیرهای محیطی نمایش داده می شود.

در نهایت، از `jinja2` به عنوان موتور قالب بندی استفاده می شود تا امکان پویاسازی محتوا و شخصی سازی صفحات بر اساس داده های کاربر فراهم شود. متغیرها و دستورهای قالب بندی درون `HTML` به ما اجازه می دهند تا با استفاده از داده ها در سرور، صفحات وبسایت را به گونه ای بسازیم که هر کاربر تجربه ای منحصر به فرد داشته باشد.

این توانایی در ترکیب با `API` ها، به ما اجازه می دهد تا اطلاعات به روز را به صورت پویا دریافت و نمایش دهیم. این هر دو کارکرد - پویاسازی و به روزرسانی محتوا - کمک می کنند تا اپلیکیشن ما در راستای وب ۲.۰ حرکت کرده و تجربه ای تعاملی برای کاربران بوجود آورد.

به این ترتیب، پروژه ای که ما توسعه داده ایم نه تنها راهکارهای قوی و مدرن برای تأیید هویت کاربران ارائه می کند، بلکه بستری را فراهم می آورد که از هر زاویه ای مطلوب و کاربرپسند است. از

داده‌های پویا گرفته تا فرآیندهای پیشرفته OAuth و از استفاده‌ی هوشمندانه‌ی API ها تا قالب‌بندی قدرتمند زبان نشانه‌گذاری.

در قسمت‌هایی مانند فایل `app.py`، ما شاهد اصول اساسی و اجرای برنامه هستیم، جایی که نمونه‌ی اپلیکیشن ساخته و به فرمان اجرایی سپرده می‌شود. این سادگی در فرم، به راحتی کار با پلتفرم‌های توسعه‌ی وب و قدرت پیاده‌سازی و اجرای آسان اپلیکیشن‌ها، بدون نیاز به دانش و زمان زیادی برای پیکربندی‌های پیچیده، اشاره دارد.

مدل‌ها، مانند `models.py`، به طور خاص تعریف و پیاده‌سازی شده‌اند تا تعاملات داده‌ای را از طریق ORM مدیریت نمایند، این روش داده‌های مرتبط با کاربران را در پایگاه داده به طور کارآمد و ایمن ذخیره می‌کند.

نهایتاً، بدون شک، امکاناتی مانند موتور قالب‌بندی `jinja2` نقش بی‌بدیلی در ارائه‌ی یک Frontend داینامیک و انعطاف‌پذیر دارند. این امکانات به توسعه‌دهندگان اجازه می‌دهند تا محتوا را بر اساس نیازها و اطلاعات کاربران سفارشی‌سازی کنند، به نحوی که بدون نیاز به نوشتن کدهای زیاد تغییراتی ایجاد شود.

احراز هویت گوگل :

احراز هویت گوگل یکی از روش‌های امن و قابل اعتماد برای ورود به سیستم‌های وب است که به کاربران امکان می‌دهد تا به جای ایجاد حساب‌های جدید و به خاطر سپاری رمز عبورهای متفاوت برای هر سایت، از اعتبار هویت گوگل خود برای دسترسی به اپلیکیشن‌ها استفاده کنند. این روش که

بخشی از استانداردهای فدراسیون OAuth 2.0 را رعایت می‌کند، از مکانیزم‌های محکم امنیتی گوگل بهره می‌برد و از آن توقع می‌رود که تجربه‌ی ورود به سیستم را تا حد زیادی ساده‌تر و ایمن‌تر کند.

روند احراز هویت گوگل به این شرح است که ابتدا کاربر درخواست ورود به سایت را می‌دهد و سپس به صفحه‌ی ورود گوگل هدایت می‌شود، که در آنجا با وارد کردن اطلاعات حساب کاربری گوگل، احراز هویت خود را انجام می‌دهد. پس از اینکه کاربر به طور موفقیت‌آمیز توسط گوگل احراز هویت می‌شود، گوگل یک توکن برای سایت درخواست‌کننده ارسال می‌کند که نشان‌دهنده‌ی تأیید هویت کاربر است.

در بخش فنی، فرآیند احراز هویت گوگل با استفاده از کتابخانه‌ها و ماژول‌هایی که Flask فراهم می‌کند، پیاده‌سازی شده است:

مقدمات احراز اعتبار OAuth 2.0 :

برای شروع، اپلیکیشن باید در سرویس‌های گوگل به ثبت برسد و شناسه‌ی کلاینت (Client ID) و رمز کلاینت (Client Secret) را دریافت کند. این اطلاعات باید در فایل‌های پیکربندی اپلیکیشن و یا متغیرهای محیطی به صورت محافظت شده نگهداری شوند.

جریان OAuth 2.0:

با استفاده از بلوپرینت‌های فلسک (Blueprint) ، مسیرهای ورودی و بازگشت (callback) تنظیم و پیکربندی می‌شوند تا با گوگل ارتباط برقرار کنند. هنگامی که کاربر تلاش می‌کند تا با

حساب گوگل خود وارد شود، یک درخواست به سرور گوگل فرستاده می شود و فرآیند OAuth آغاز می گردد.

احراز هویت و دریافت توکن:

پس از اینکه کاربر اطلاعات حساب کاربری گوگل خود را وارد می کند، گوگل او را به مسیر بازگشت (`redirect_uri`) که در اپلیکیشن تعریف شده، باز می گرداند. همراه با این بازگشت، یک کد احراز هویت ارسال می گردد که باید برای دریافت توکن نهایی (`access token`) به سرور گوگل فرستاده شود.

اعتبارسنجی و ورود به سیستم:

پس از دریافت توکن، اپلیکیشن از توکن به منظور دریافت اطلاعات بیشتر درباره کاربر استفاده می کند. این اطلاعات شامل `sub` (یک شناسه منحصر به فرد برای کاربر) و اطلاعات پایه مانند نام و ایمیل می باشد. این اطلاعات سپس برای ساخت یا به روزرسانی حساب کاربر داخل دیتابیس اپلیکیشن و ورود وی به سیستم مورد استفاده قرار می گیرند.

توسط این روش، احراز هویت در اپلیکیشن‌های مدرن به یک فرآیند سریع، امن و کاربرپسند تبدیل می‌شود که ریسک‌های امنیتی مرتبط با ذخیره رمز عبور را به حداقل می‌رساند و از سیستم‌های پیشرفته امنیتی گوگل بهره می‌گیرد.

وایر فریم و فرانت اند :

وب سایتی که طراحی کردیم از چند صفحه شامل صفحه ورود (login)، صفحه ثبت نام (signup) و صفحه خانه (Home) می‌شود.

صفحه‌ی ورود به سیستم و صفحه ثبت نام:

اولین بخش کد یک صفحه‌ی ورود ساده و مدرن را توضیح می‌دهد. این صفحه ترکیبی از مینی‌مالیسم تصویری و کاربرد عملکردی است که با استفاده از زبان طراحی مدرن و رابط کاربری انعطاف‌پذیر می‌باشد. بهره‌گیری از فونت‌های Google ، SVG برای ایجاد آیکون‌ها، و صفحه‌بندی واکنش‌گرا نشان‌دهنده دقت در زیبایی‌شناسی و کارایی است.

عناصر ورودی (Input elements) با لیبل‌ها و آیکون‌های مربوطه بطور مؤثری طراحی شده تا تجربه‌ی کاربری نرم و روانی ارائه دهد.

طراحی حساس به بافت‌پذیری این اطمینان را می‌دهد که چیدمان صفحه به خوبی بر روی دستگاه‌های مختلف، از جمله گوشی‌های هوشمند و تبلت‌ها، به نمایش درآید.

در این صفحات از فرایند OAuth گوگل برای ورود به سیستم نیز استفاده می‌شود؛ یک دکمه‌ی اختصاصی با تصویر گوگل که کاربران را هدایت می‌کند تا با استفاده از حساب‌های کاربری گوگل خود وارد شوند، نشان‌دهنده‌ی به کارگیری آخرین روش‌های احراز هویت می‌باشد.

صفحه نمایش (خانه) :

در این صفحه به کمک Jinja2 از `base.html`، قسمتی از یک صفحه وب را به تصویر می کشد که مقالات یا پست های وبلاگ را نمایش می دهد. این بخش به طور موثر از قدرت Jinja2 برای نمایش محتوای پویا استفاده می کند. استفاده از دستور `for` در Jinja2 به ما امکان می دهد تا از یک `iterable` برای ایجاد لیستی از مقالات در صفحه استفاده کنیم.

اضافه شدن قابلیت اشتراک گذاری مستقیم محتوا به شبکه های اجتماعی همانند تونیترا با استفاده از پارامترهای URL که برای تونیت کردن محتوای پیش تعریف شده استفاده می شود، نمایش دهنده ی اندیشه ی گسترش دادن تعامل کاربران با محتوا و تشویق به اشتراک گذاری آن در پلتفرم های اجتماعی است.

تصاویر بزرگ با عناوین و خلاصه هایی جذاب، به همراه دکمه های اشتراک گذاری اجتماعی، یک رویکرد کاربر مدارانه را دنبال می کنند که نه تنها به ارائه محتوا می پردازد بلکه تعامل کاربر با محتوا و دیگران را نیز تسهیل می بخشد.

در زیر صفحات طراحی شده قرار گرفته است .


صفحه ی ورود به سیستم و صفحه ثبت نام:

Login Example

+

← → ↻ http://127.0.0.1:5000/login

☆ ⬇️ 🏠 » ☰



Welcome

Please enter your details!


Email:

@

Password

🔒

Login

 Log in with Google

Don't have an account?


[Sign Up](#)

Login Example

+

← → ↻ http://127.0.0.1:5000/sign-up

☆ ⬇️ 🏠 » ☰



Create new account

your Name:

☰

Email:

@


Password

🔒

Confirm Password

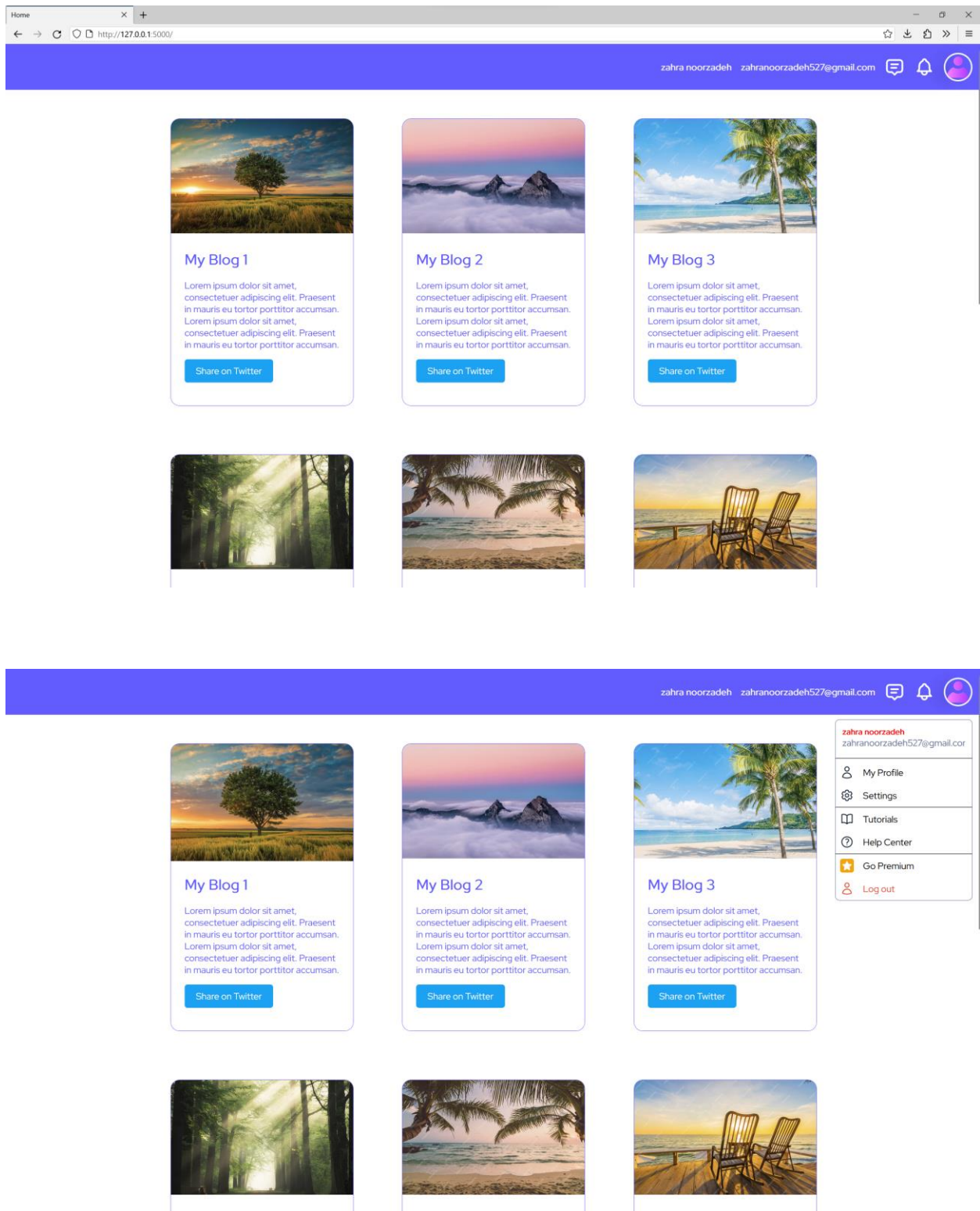
🔒

Sign up

 Log in with Google

login

صفحه نمایش (خانه) :



صفحه احراز هویت :

The screenshot shows a web browser window with the address bar displaying a Google sign-in URL. The main content area features a 'Sign in with Google' dialog box. Inside the dialog, the text 'Sign in to continue to Google auth Flask App' is centered. Below this, there is a text input field labeled 'Email or phone' containing the text 'zahranoorzadeh527'. A link 'Forgot email?' is positioned below the input field. Further down, a paragraph explains that Google will share the user's name, email address, language preference, and profile picture with the Flask App. At the bottom of the dialog, there are two buttons: 'Create account' on the left and 'Next' on the right. The footer of the page includes the text 'English (United States)' with a dropdown arrow, and links for 'Help', 'Privacy', and 'Terms'.

صفحه اشتراک گذاری تویتر :

The screenshot displays the Twitter interface with a login prompt overlay. The overlay is a white box with a blue Twitter bird icon and the text 'Want to log in first? You'll need to log in before you can share that post.' At the bottom of the overlay are two buttons: 'Sign up' and 'Log in'. In the top right corner of the Twitter page, there is a 'Sign in with Google' button. The background of the page is dark gray, and the bottom of the page has a dark blue footer with the text 'Don't miss what's happening People on X are the first to know.' and links for 'Log in' and 'Sign up'.