

تقسی

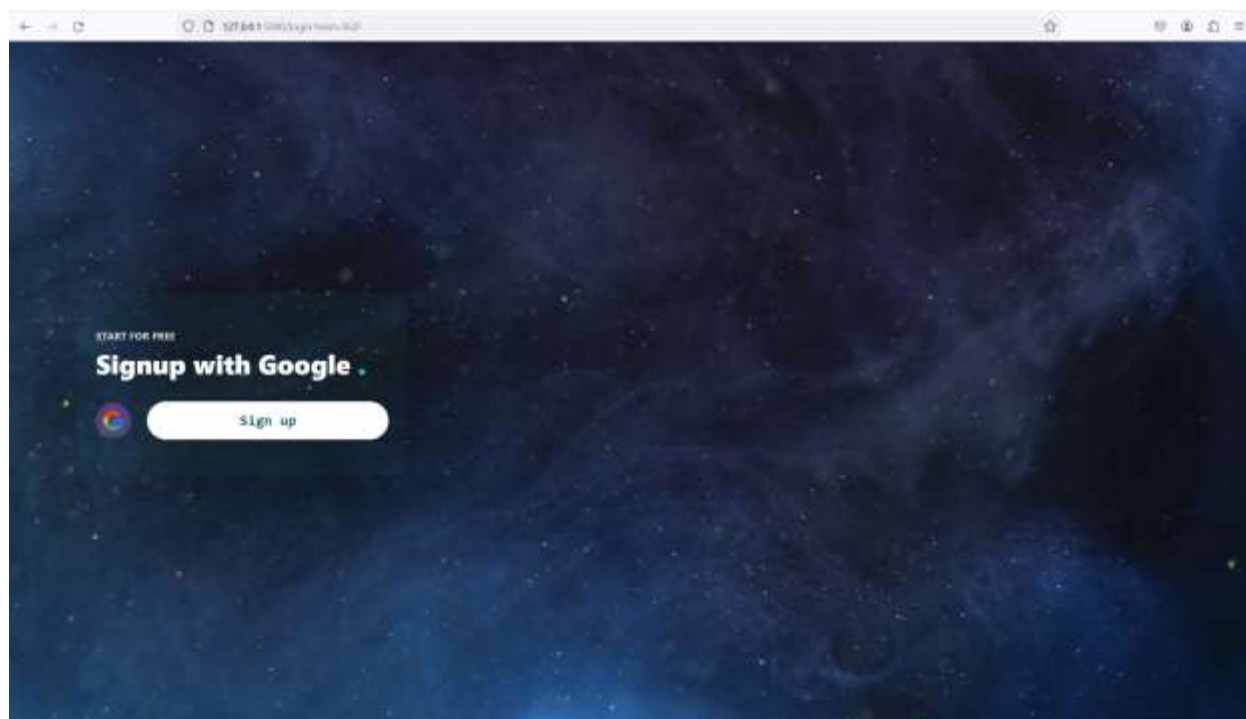
پروژه مهندسی اینترنت

عنوان :

**Google Auth Web App**

نسترن بیان جمجاه

- مقدمه : ..... ۴
- استفاده از کنسول توسعه دهنده گوگل : ..... ۶
- برنامه نویسی : ..... ۱۰
- تصاویر برنامه : ..... ۱۶



.....

## مقدمه :

از آغاز رونق فضای دیجیتال و گسترش روز افزون وبسایت‌ها، نیاز به راهکارهای امن و بهینه برای احراز هویت کاربران به موضوعی کلیدی تبدیل شده است. با پیشرفت تکنولوژی و وجود تهدیدات فزاینده سایبری، تضمین امنیت داده‌های حساس کاربران بیش از پیش اهمیت یافته است. در این راستا، بسیاری از توسعه‌دهندگان وب به سمت راه‌اندازی سیستم‌های احراز هویتی معتبر و قابل اتکا سوق پیدا کرده‌اند. یکی از شیوه‌های پرطرفدار این استفاده از سرویس اعتبارسنجی ارائه شده توسط شرکت‌های بزرگ فناوری مانند گوگل است؛ روشی که از لحاظ امنیتی مورد تایید است و به کاربران اجازه می‌دهد با استفاده از اکانت‌های کاربری موجود خود، به سرویس‌های مختلف وارد شوند.

در این میان، فریم‌ورک Flask در دنیای پایتون با ارائه یک بستر ساده و در عین حال قدرتمند، امکان توسعه وبسایت‌های سریع و انعطاف‌پذیر را فراهم آورده است. ترکیب قابلیت‌های Flask با سیستم اعتبارسنجی Google می‌تواند به توسعه‌دهندگان این توانایی را بدهد که یک وبسایت محکم با تجربه کاربری بی‌نقص را ارائه دهند. در مستندات این تمرین قصد داریم ببینیم چگونه می‌توانیم با استفاده از Flask و Google Authentication به توسعه یک وبسایت امن که به کاربران امکان ورود آسان و سریع را می‌دهد، بپردازیم.

این ترکیب نه تنها روند ورود به سیستم را برای کاربران ساده‌تر می‌کند، بلکه برای توسعه‌دهندگان نیز از پیچیدگی‌های مرتبط با مدیریت کلمات عبور و احراز هویت دستی می‌کاهد. به محض اینکه یک کاربر با استفاده از حساب گوگل خود وارد یک وبسایت مبتنی بر Flask شود، اطلاعاتی که توسط Google تایید و دریافت شده‌اند، می‌توانند به آسانی در سیستمی که ایمنی، دقت، و سرعت را تضمین می‌کند، مورد استفاده قرار گیرند. این روش که در بستر OAuth 2.0 عمل می‌کند، نه تنها به اشتراک‌گذاری امن اطلاعات کاربری بین وبسایت و گوگل کمک کند بلکه به واسطه استانداردهای مود نیاز امروزی، از استقبال گسترده‌ای نیز برخوردار شده است.

با پیاده‌سازی نمونه کدی که در بالا به آن اشاره شد و بررسی عمیق‌تر مراحل تک تک، ما در این مقاله سعی خواهیم کرد تا با نحوه تنظیم و پیکربندی Google Authentication در فریم‌ورک Flask آشنا شویم.

## استفاده از کنسول توسعه دهنده گوگل :

در ادامه‌ی مستندات ، ما شروع به بررسی جزییات تکنیکی خواهیم کرد، از جمله چگونگی استفاده از کنسول توسعه‌دهنده گوگل برای ایجاد یک پروژه جدید، دریافت شناسه‌های OAuth و پیکربندی مجوزهای لازم برای اعتبارسنجی کاربران. این فرآیند، در حالی که امکان ارتباط بین فلسک و سرویس‌های گوگل را فراهم می‌آورد، به ما اجازه می‌دهد تا احراز هویت را به نحو احسن درون برنامه‌های خود ادغام کنیم. سپس، وارد عرصه پیاده‌سازی کد خواهیم شد، جایی که به صورت عملی نحوه ایجاد مسیرها (routes) در Flask را برای کنترل ورود و خروج کاربران از طریق Google OAuth مورد بررسی قرار می‌دهیم. با استفاده از کتابخانه‌های معتبر و اثبات شده مانند authlib، یک دید کلی نسبت به ساختار درخواست‌ها و پردازش پاسخ‌ها ارائه خواهیم کرد و نشان می‌دهیم چطور می‌توان این اطلاعات را برای ایجاد یک جلسه‌ی کاربری معتبر و امن استفاده کرد.

برای استفاده از کنسول توسعه‌دهنده گوگل (Google Developer Console) جهت ایجاد یک پروژه جدید و تنظیم سرویس‌های Google Authentication ، شما نیاز به انجام چند مرحله اساسی دارید. در زیر، به صورت گام‌به‌گام به توضیح این فرآیند می‌پردازیم:

## گام ۱: ایجاد پروژه جدید

۱. وارد حساب کاربری Google خود می شویم و به [Google Cloud Platform Console](https://console.cloud.google.com/) می رویم.

۲. بر روی دکمه "ساخت پروژه" کلیک می کنیم.

۳. یک نام برای پروژه انتخاب می کنیم.

۴. بر روی "ساخت" کلیک می کنیم تا پروژه همان ایجاد شود.

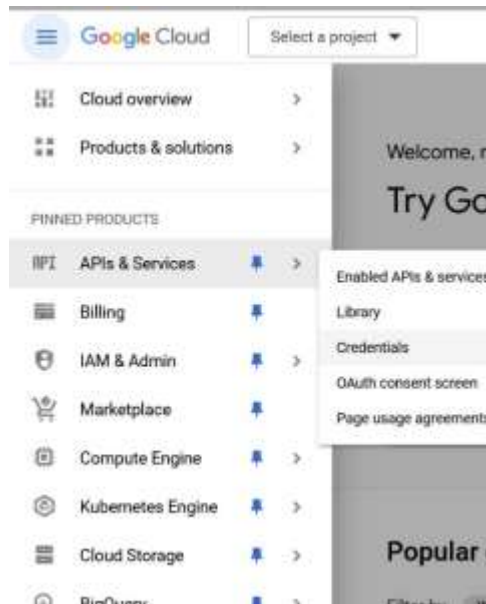


## گام ۲: تنظیم API و سرویس ها

۱. پس از ایجاد پروژه، به داشبورد پروژه می رویم.

۲. به بخش "API ها و سرویس ها" < "کتابخانه" می رویم.

۳. API های احراز هویت Google را انتخاب می کنیم و آن را فعال می کنیم.



### گام ۳: ایجاد اعتبارها

۱. به داشبورد API ها و سرویس ها و به بخش "اعتبارها" می رویم.
۲. بر روی "ساخت اعتبارها" کلیک می کنیم و گزینه "شناسه مشتری OAuth" را انتخاب می کنیم.
۳. نوع برنامه ی کاربردی خود را انتخاب می کنیم (ما web application را انتخاب کردیم).



## ← Create OAuth client ID

Application type \*  
Web application

۴. نام محصول و اطلاعات مورد نیاز دیگر مانند ریدایرکت یوآرال‌های مجاز را تنظیم می‌کنیم، این باید آدرسی باشد که Google بعد از احراز هویت کاربر به آن کاربر را ریدایرکت می‌کند.

### Authorized JavaScript origins ⓘ

For use with requests from a browser

+ ADD URI

### Authorized redirect URIs ⓘ

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

۵. پس از اتمام تنظیمات، شناسه مشتری (OAuth (Client Secret) به ما داده خواهد شد و آن را دانلود می‌کنیم.

با انجام این مراحل، ما هم اکنون اعتبار OAuth 2.0 را برای پروژه خود در Google Cloud Platform ایجاد کرده‌ایم و می‌توانید شناسه مشتری و مخفی‌گاه را در اپلیکیشن Flask خود استفاده کنیم، تا از هویت کاربران توسط Google احراز هویت کنیم.

## برنامه نویسی :

در پروژه وب جاری ما، استفاده از فناوری‌ها و فریم‌ورک‌های پیشرفته به ما امکان داده تا تجربه‌ای نوآورانه و بی‌نظیر در عرصه‌ی وب ایجاد کنیم. با تکیه بر ماژول‌ها و اکستنشن‌های قدرتمند Python و Flask، به ایجاد یک پلتفرم تعاملی دست یافته‌ایم که بر کارایی و امنیت کاربر اهمیت ویژه‌ای می‌دهد.

بهره‌گیری از محیط توسعه‌ی Visual Studio Code با امکانات پیشرفته‌ی آن، به ما اجازه داد تا فرآیندهای فرانت‌اند و بک‌اند را بهینه‌سازی کنیم. ماژول `init.py` در این میان، هسته‌ی مرکزی اپلیکیشن Flask ما را تشکیل می‌دهد و به ما این امکان را می‌دهد که تک تک بخش‌های پروژه را با دقت مهندسی کرده و اجرا کنیم.

تابع `create_app()` در هسته‌ی اپلیکیشن تعریف شده، نه تنها اپلیکیشن Flask را مقدمه‌چینی می‌کند، بلکه مسئولیت پیکربندی دیتابیس‌ها، مدیریت جلسات کاربری و حفاظت از نقاط دسترسی احساسی را عهده‌دار است. همچنین، می‌توانیم

از دکوراتورهایی بهره ببریم که تضمین کنند دسترسی‌ها تنها پس از احراز هویت کاربران فراهم خواهد شد.

اکستنشن Flow در کتابخانه google\_auth\_oauthlib، پیاده‌سازی فرآیند OAuth 2.0 را ساده‌سازی کرده و محافظت از توکن‌ها و اطلاعات شخصی کاربران را از طریق ابزارهای امنیتی تضمین می‌کند. استفاده از موتور الگوسازی Jinja2 نیز به ما امکان می‌دهد تا محتوای سایت را به صورت پویا بر اساس داده‌ها و ترجیحات کاربران به‌روز کنیم، که این امر تجربه‌ای شخصی و خصوصی را برای هر بازدیدکننده بوجود می‌آورد.

تمامی این عناصر با یکدیگر متحد شده‌اند تا بستری کامل و کاربردی را پدید آورند؛ جایی که امنیت، سادگی دسترسی و پویایی محتوا، سه رکن اصلی را تشکیل می‌دهند. این پروژه یک تجربه‌ی کاربری جامع و دلنشین را ارائه می‌کند.

### پیکربندی اپلیکیشن:

در ماژول init.py، ساختار اصلی اپلیکیشن را مشخص می‌کنیم. با استفاده از اکستنشن‌های Flask مانند SQLAlchemy برای مدیریت دیتابیس و LoginManager برای مدیریت حالت‌های ورود و انجام عملیات احراز هویت، اطلاعات کاربران را به صورت امن و بهینه مدیریت می‌کنیم.

### دکوراتورهای امنیتی:

برای محافظت از بخش‌های حساس اپلیکیشن، ما دکوراتور `@login_required` را تعریف کرده‌ایم، این دکوراتور اطمینان حاصل می‌کند که فقط کاربرانی که احراز هویت شده‌اند دسترسی به منابع محفوظ دارند.

### پویاسازی محتوا:

با استفاده از موتور قالب‌بندی `jinja2`، محتویات صفحات وب را بر اساس داده‌های کاربر مثل نام و وضعیت ورود، پویا می‌سازیم تا هر کاربر تجربه شخصی‌سازی شده‌ای را دریافت کند.

### ادغام API ها:

برای دریافت و نمایش اطلاعات به‌روز مانند اطلاعات کاربری یا داده‌های تعاملی، ما از API ها استفاده می‌کنیم. این باعث می‌شود تا کاربران قادر به دریافت داده‌ها و اطلاعات مناسب در زمان واقعی باشند.

### ولیداسیون تجربه کاربری:

ما روی UX/UI اهمیت زیادی قرار می‌دهیم و مطمئن می‌شویم که فلوهای کاربری با وضوح بالا و با سهولت استفاده طراحی شده‌اند تا هر کاربری، چه تازه‌کار و چه حرفه‌ای، بتواند بدون سختی از برنامه استفاده کند.

با ادغام همه این مؤلفه‌ها، اپلیکیشن ما یک نمونه عالی از چگونگی استفاده از فریم‌ورک‌ها و ابزارهای مدرن توسعه وب است، که هم امنیت و هم قابلیت استفاده

کاربری را بهبود می‌بخشد. پروژه ما تعادل بین عملکرد تکنیکی و زیبایی‌شناختی را به نمایش می‌گذارد، از این رو، یک چارچوب را پیشنهاد می‌کند که با امکاناتی که ارائه می‌دهد، اطمینان حاصل می‌کند تجربه کاربران همیشه اصلی‌ترین اولویت است.

برنامه ما یک اپلیکیشن فلسک است که احراز هویت کاربران را از طریق گوگل انجام می‌دهد و امکان به اشتراک‌گذاری اطلاعات را در تلگرام و واتس‌آپ فراهم می‌کند. در ادامه، توضیحات فنی هر بخش از کد را مرور می‌کنیم:

### پیکربندی و آغاز اپلیکیشن:

- ابتدا ماژول‌های لازم از جمله `os`, `requests`, و `flask` را وارد (`import`) می‌کنیم.
- کد یک نمونه از `Flask` اپلیکیشن ساخته و یک مسیر (`Blueprint`) به نام `route` تعریف می‌کند.
- یک رمز جلسه (`session secret key`) برای امنیت جلسه کاربری تعریف می‌شود.
- مهاجرت دیتابیس با استفاده از `Flask-Migrate` به اپلیکیشن اضافه می‌شود.

### تنظیمات: OAuth 2.0

- یک فایل شامل اطلاعات امنیتی مربوط به احراز هویت گوگل وارد می‌شود.

- Flow از کتابخانه google\_auth\_oauthlib برای پیکربندی فرایند احراز هویت گوگل استفاده می‌شود.

### به اشتراک‌گذاری اطلاعات:

- تابع telegram یک پیام متشکل از نام و ایمیل کاربر جاری را برای اشتراک‌گذاری در تلگرام آماده می‌کند.
- تابع whatsapp به عملکرد مشابهی برای واتساپ دارد و اطلاعات کاربر را با افزودن به یک URL واتساپ منتقل می‌کند.

### احراز هویت گوگل:

- مسیر login-by-google کاربر را به URL احراز هویت گوگل ریدایرکت می‌کند.
- callback مسیری است که گوگل پس از احراز هویت به آن باز می‌گردد. اینجا است که توکن احراز هویت از گوگل دریافت و اعتبار آن بررسی می‌شود.
- اگر کاربر قبلاً وجود ندارد، به صورت خودکار در دیتابیس ایجاد می‌شود.

### خروج کاربر:

- تابع logout کاربر را از حساب خود خارج (logout) می‌کند و به صفحه ورود ریدایرکت می‌کند.

### دکوراتور امنیتی:

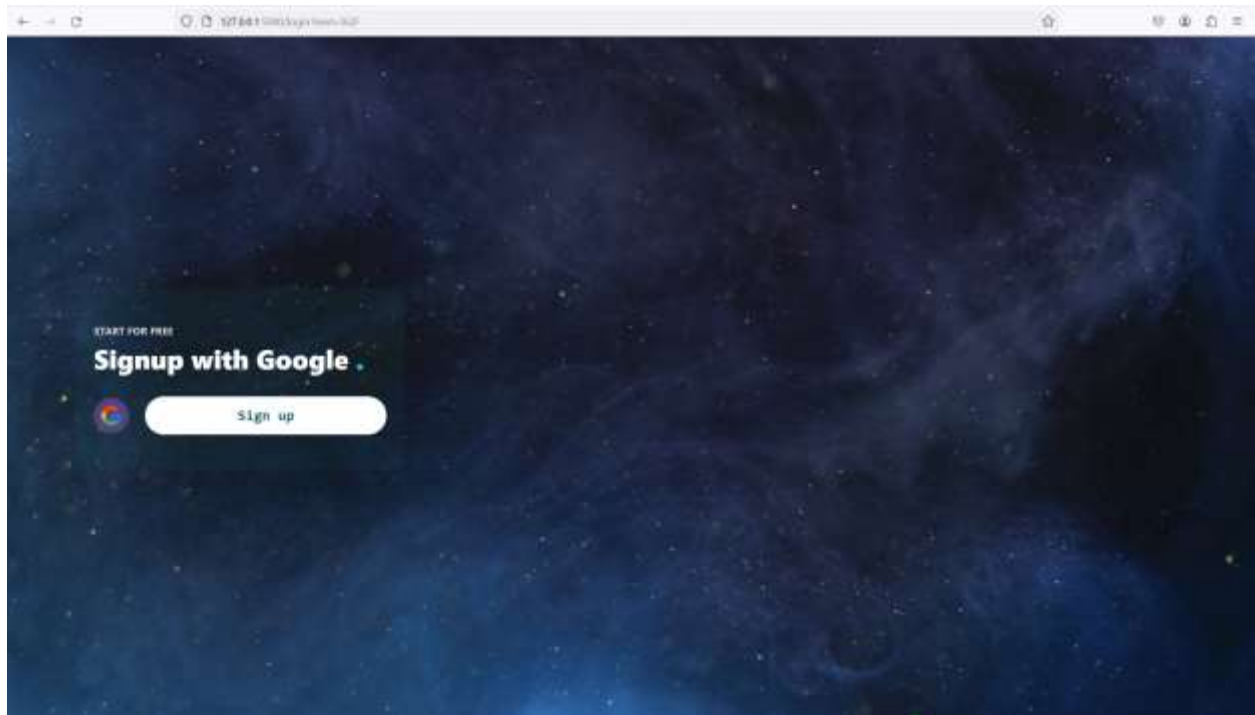
• `login_is_required` دکوراتوری است که بررسی می‌کند کاربر دارای `google_id` در جلسه (`session`) است. اگر نیست، به کد ارور ۴۰۱ (`Unauthorized`) می‌رود.

این کد یک پایه‌ای استوار برای احراز هویت مدرن و به اشتراک‌گذاری آسان اطلاعات فراهم می‌آورد. گوگل OAuth2 یک روش ایمن برای ورود کاربران به وب‌اپلیکیشن‌ها است و برنامه ما به خوبی این فرایند را پیاده‌سازی کرده است.

در ادامه تصاویری از برنامه طراحی شده قرار گرفته است :

تصاویر برنامه :

صفحه ورود به برنامه :



این تصویر یک صفحه ورود ساده و جذاب با امکان احراز هویت از طریق گوگل است که با HTML پیاده‌سازی می‌شود .

با استفاده از کتابخانه CSS شناخته شده و فریمورک Tailwind، این صفحه طراحی را به شکلی عصری و واکنش‌گرا ارائه می‌دهد.

ساختار کلی صفحه:



`<head>` - تگ‌های مربوط به متا دیتا، عنوان صفحه و اسکریپت‌های مورد نیاز را از جمله Tailwind CSS را شامل می‌شود.

`<body>` - همه عناصر مربوط به نمایش صفحه را در خود جای داده است.

### بخش‌های تشکیل دهنده صفحه:

- یک `

` که به کل صفحه امتداد دارد و پس‌زمینه‌ای شامل تصویری از Engin Akyurt که از Unsplash دریافت شده است را نمایش می‌دهد.

- داخل این `

` یک `

` اصلی وجود دارد که فرم ورود را مرتب می‌کند، این فرم دکمه‌ای جهت ورود با حساب گوگل دارد.

- فرم همچنین شامل یک عنوان `` و `

# ` است که برای جلب توجه به استفاده از گوگل برای ثبت نام همراه با کلمه "Signup" به شکل خلاقانه‌ای به کار رفته است.

- دکمه ورود با استایل‌های Tailwind CSS طراحی شده است تا با کلیک کاربر به مسیر '/login-by-google' هدایت شود.

## ظاهر و احساس صفحه:

- استفاده از سایه‌ها و لایه‌های نیمه شفاف در طراحی تجربه کاربری را افزایش می‌دهد.

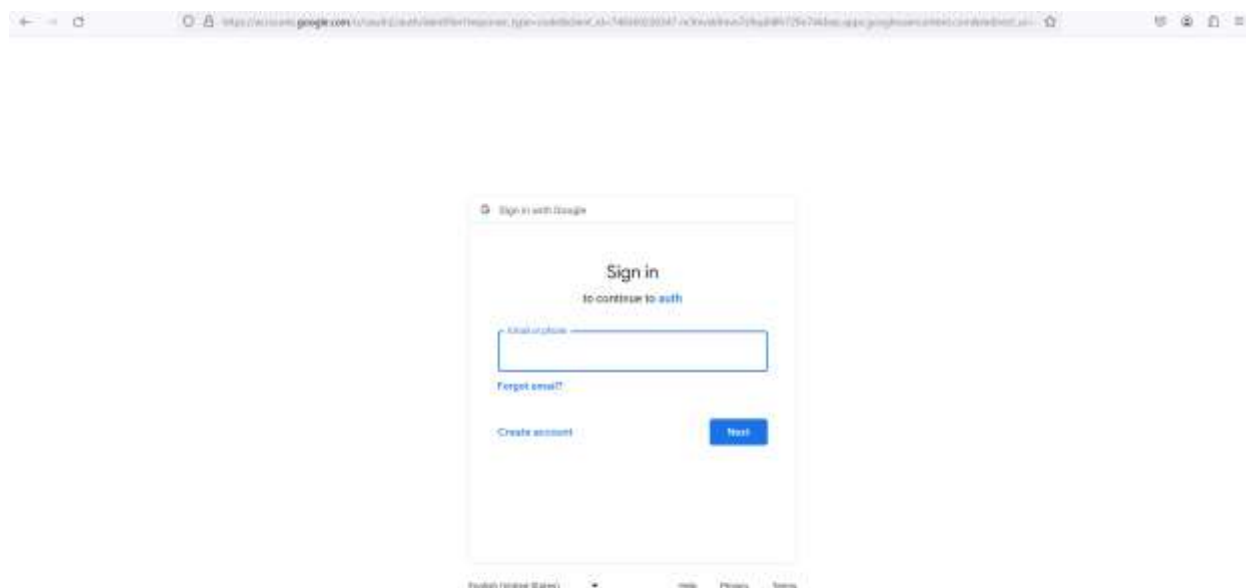
- تصویر پس‌زمینه نمایی زیبا و آرامش‌بخش ایجاد می‌کند که به جذابیت صفحه کمک می‌کند.

- کاربرد رنگ‌های تیره در کنار طیف‌های رنگی روشن، تضاد بصری جذابی برای کاربر فراهم می‌کند.

- دکمه ورود به گونه‌ای طراحی شده که با هور ماوس، انیمیشن و تغییر رنگ ایجاد می‌کند که چشم کاربر را به خود جلب کند و تعامل او را تشویق کند.

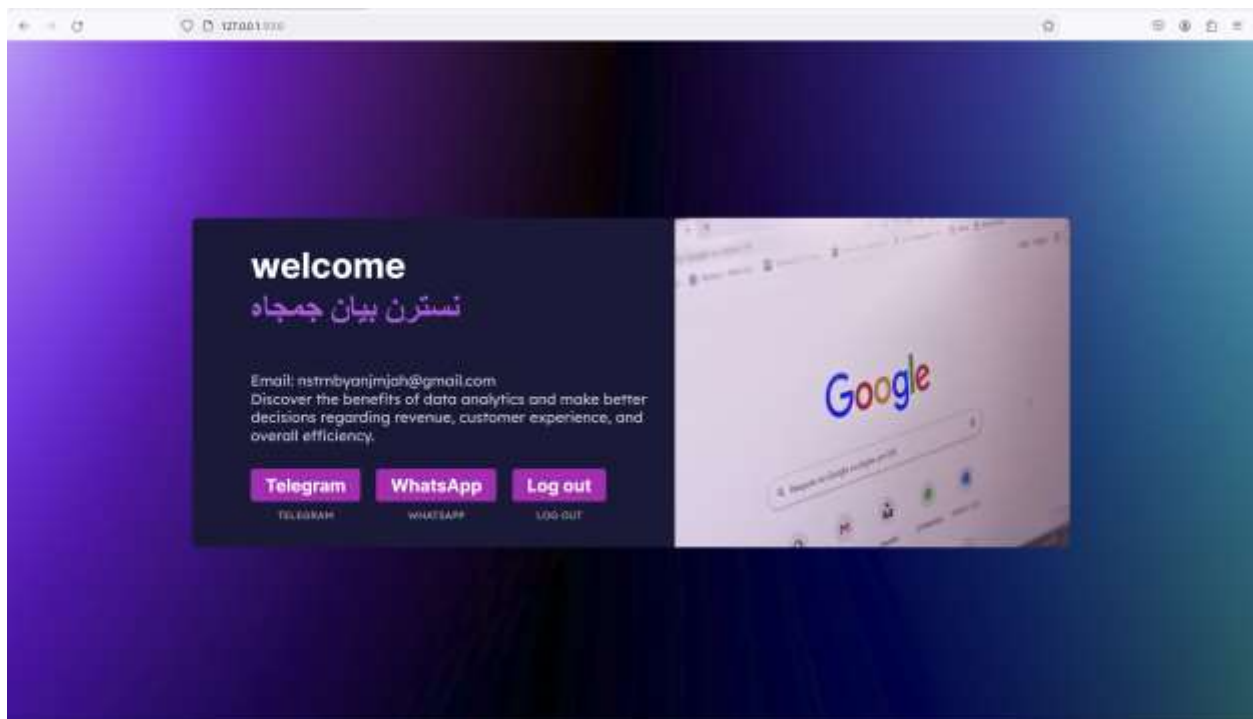
در کل، صفحه این پیام را ارسال می‌کند که ورود به سایت فوق‌العاده ساده و راحت است، و استفاده از Google sign-in این فرایند را آسان‌تر و امن‌تر می‌کند. این صفحه نه تنها از نظر زیبایی‌شناسی جذاب است بلکه از طراحی واکنش‌گرا و کاربرپسند بهره می‌برد که برای کاربران مدرن از اهمیت بالایی برخوردار است.

## صفحه احراز هویت :



این صفحه ، صفحه احراز هویت با اکانت گوگل است که بعد از کلیک کردن بر روی دکمه کاربر به آن هدایت می شود.

صفحه خانه :



این صفحه HTML یک پنل کاربری برای وبسایت است که به کاربران امکان می‌دهد به سادگی اطلاعات خود را در پلتفرم‌های اجتماعی به اشتراک بگذارند و یا از حساب کاربری خود خارج شوند. در ادامه، جزئیات بیشتری درباره‌ی هر بخش از این صفحه به منظور استفاده در مقاله ارائه می‌دهیم:

## توضیحات هدر:

- استفاده از `<meta charset="UTF-8">` برای پشتیبانی از تمام نمادها و کاراکترها در زبان‌های مختلف.

- `<meta name="viewport" content="width=device-width, initial-scale=1.0">` موجب واکنش‌گرا بودن صفحه در دستگاه‌های مختلف می‌شود.

`<title>` - عنوان صفحه است که در برگه مرورگر کاربر نمایش داده می‌شود.

## تنظیمات CSS کدهای طراحی:

- دو فایل CSS جداگانه برای تعریف سبک‌های دسکتاپ و موبایل لینک شده‌اند.

## سکشن اصلی:

`<main class="container">` - به عنوان کانتینر اصلی صفحه عمل می‌کند و تمامی المان‌های داخل صفحه را در خود جای می‌دهد.

- در `<div class="box">` دو بخش اصلی صفحه تعبیه شده است: بخش متن (`text-sec`) و بخش تصویر (`img-sec`).

## بخش متن:

`<div class="text-sec">` - شامل دو بخش فرزند `heading` و `table` است.

`<div class="main-heading">` - یک سرخط بزرگ با استفاده از `

# ` دارد که نام کاربر را از طریق یک متغیر جینجا نمایش می‌دهد.

`<div class="sub-heading">` - شرح کوتاهی درباره مزایای آنالیز داده‌ها ارائه می‌دهد.

`<div class="table">` - یک سری دکمه‌های عملیاتی مثل به اشتراک‌گذاری در `Telegram` و `WhatsApp` و همچنین دکمه `Log out` را شامل می‌شود.

## بخش تصویر:

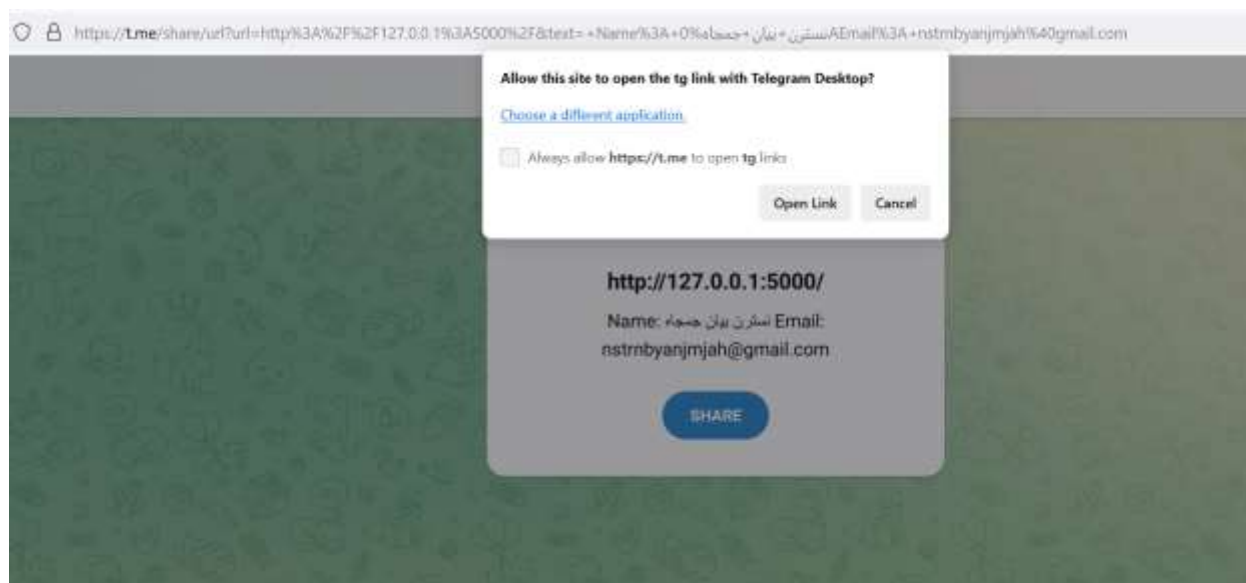
`<div class="img-sec">` - تصویری تزئینی به صفحه اضافه می‌کند که حس نشاط و انگیزه را به کاربر منتقل می‌کند.

`<div class="main-img">` - از یک حالت در حال نمایش تصویر استفاده می‌کند که توسط تگ `` به نمایش در می‌آید.

مناسب با دستگاه‌های موبایل و ریسپانسیو:

-سبک‌های مربوط به موبایل در فایل `mobile.css` تعریف شده‌اند تا تجربه کاربر را در دستگاه‌های مختلف بهینه کنند.

این صفحه خطاب به کاربرانی است که مایل به دسترسی سریع و آسان به امکانات پایه وبسایت هستند و همچنین تمایل دارند که به راحتی اطلاعاتی را به اشتراک بگذارند یا از حساب کاربری خود خارج شوند. طراحی واکنش‌گرا به کاربران این امکان را می‌دهد که از هر جایی و با هر دستگاهی به صورت فعال در وبسایت مشارکت کنند.



Send the following on WhatsApp

Continue to Chat

Name: نسترن بیان جمجاه Email: nstrnbyanjmjah@gmail.com

---

Don't have WhatsApp yet?

[Download](#)