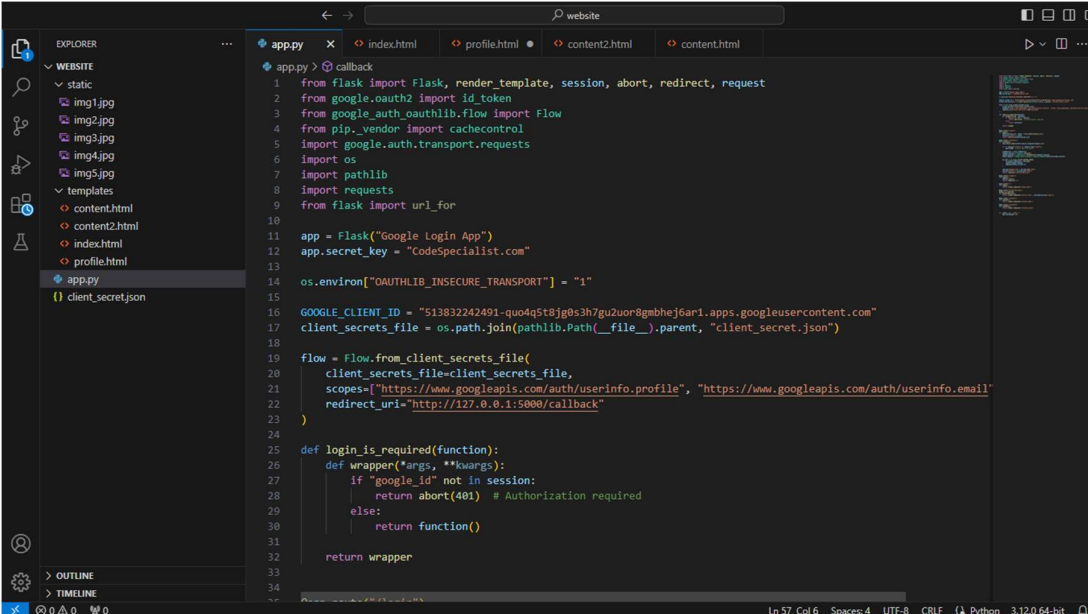


فایل app.py برای بک اند سایت مورد استفاده قرار گرفته است که به صورت زیر است:



```
1 from flask import Flask, render_template, session, abort, redirect, request
2 from google.oauth2 import id_token
3 from google_auth_oauthlib.flow import Flow
4 from pip._vendor import cachecontrol
5 import google.auth.transport.requests
6 import os
7 import pathlib
8 import requests
9 from flask import url_for
10
11 app = Flask("Google Login App")
12 app.secret_key = "CodeSpecialist.com"
13
14 os.environ["OAUTHLIB_INSECURE_TRANSPORT"] = "1"
15
16 GOOGLE_CLIENT_ID = "513832242491-qup4q5t8jg8s3h7gu2uor8gmbhej6ar1.apps.googleusercontent.com"
17 client_secrets_file = os.path.join(pathlib.Path(__file__).parent, "client_secret.json")
18
19 flow = Flow.from_client_secrets_file(
20     client_secrets_file=client_secrets_file,
21     scopes=["https://www.googleapis.com/auth/userinfo.profile", "https://www.googleapis.com/auth/userinfo.email"],
22     redirect_uri="http://127.0.0.1:5000/callback"
23 )
24
25 def login_is_required(function):
26     def wrapper(*args, **kwargs):
27         if "google_id" not in session:
28             return abort(401) # Authorization required
29         else:
30             return function()
31     return wrapper
```

این کد یک وب اپلیکیشن ساده با استفاده از چارچوب Flask برای ورود به سیستم با حساب Google ایجاد می‌کند. این اپلیکیشن از OAuth 2.0 برای احراز هویت کاربران با استفاده از حساب Google استفاده می‌کند.

در ادامه توضیحاتی در مورد هر بخش از کد آورده شده است:

1. استفاده از کتابخانه‌ها:

- `Flask`: برای ایجاد وب اپلیکیشن.

- `render_template`: برای نمایش تمپلیت‌های HTML.

- `session`: برای نگهداری اطلاعات مربوط به جلسه کاربر.

- `abort`: برای نمایش پیام خطا و اجبار به توقف اجرا.

- `redirect`: برای هدایت کاربر به صفحات مختلف.

- `request`: برای درخواست‌های HTTP.

- `google.oauth2.id_token`: برای تأیید توکن‌های ID از سمت Google.

- `google_auth_oauthlib.flow`: برای ایجاد یک جریان OAuth با استفاده از اطلاعات کلاینت.

- `pip._vendor.cachecontrol`: برای مدیریت کش درخواست‌های HTTP.

- `google.auth.transport.requests`: برای انتقال اطلاعات احراز هویت به سمت Google.

2. تنظیمات اولیه:

- `app.secret_key``: یک کلید مخفی برای امنیت جلسه.
- `os.environ["OAUTHLIB_INSECURE_TRANSPORT"] = "1"`: فعال سازی ارتباط ناامن برای آزمون.

3. تنظیمات OAuth و احراز هویت:

- `GOOGLE_CLIENT_ID``: شناسه کلاینت Google API.
- `client_secrets_file``: مسیر فایل حاوی اطلاعات احراز هویت کلاینت.
- `flow``: ایجاد یک جریان OAuth با استفاده از اطلاعات کلاینت و تنظیمات مورد نیاز.

4. توابع دسترسی:

- `login_is_required``: یک دکوراتور برای بررسی لاگین بودن کاربر و اعمال محدودیت های لاگین.

5. مسیرهای اصلی:

- `login/``: مسیر برای هدایت کاربر به صفحه ورود به سیستم Google.
- `callback/``: مسیر بازخورد از سمت Google پس از موافقت کاربر با احراز هویت.
- `logout/``: مسیر خروج از حساب کاربری.
- `/``: صفحه اصلی.

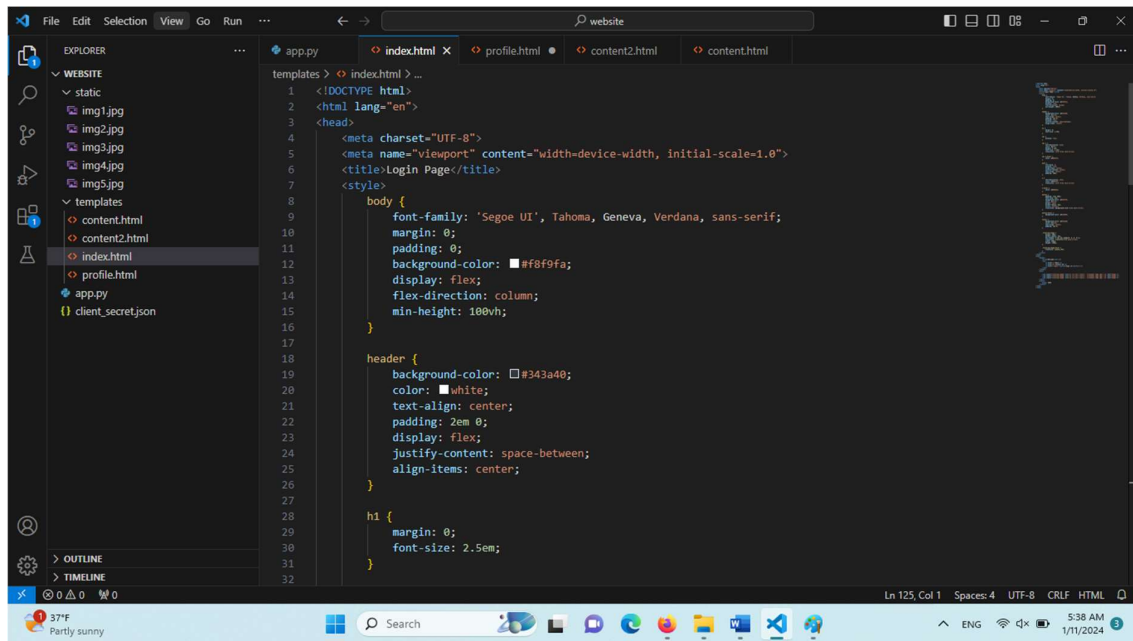
- `protected_area/``: صفحه محافل

6. اجزاء تمپلیت:

- `index.html``: تمپلیت صفحه اصلی.
- `profile.html``: تمپلیت برای نمایش صفحه محافل پس از ورود.
- `content.html`` و `content2.html``: تمپلیت های محتوا.

7. اجرای اپلیکیشن:

- `if __name__ == "__main__": app.run(debug=True)``: اجرای اپلیکیشن در حالت دیباگ.
صفحه اصلی سایت Index.html نام دارد که برای اولین صفحه نمایش برای کاربر مورد استفاده قرار گرفته است:



این کد یک صفحه HTML است که یک صفحه ورود ساده را نمایش می‌دهد. در ادامه، توضیحاتی در مورد هر بخش از کد آورده شده است:

1. Head:

- تعیین مشخصات صفحه شامل زبان (`lang="en"`)، کاراکترست (UTF-8) و ابعاد نمایش.

- تعیین عنوان صفحه به `"Login Page"`.

- اضافه کردن استایل‌های CSS.

2. Body:

- ست کردن استایل بدنه صفحه با فونت و رنگ پس زمینه.

- ایجاد یک هدر با عنوان `"Welcome"` و یک ناوبری با لینک‌های `"Contact Us"`، `"Main"` و یک دکمه `"Sign in"`.

- اضافه کردن یک بخش اصلی (`<main>`) با تصاویر متحرک (floating images) که هنگام هورور بر روی آن‌ها، بزرگ‌تر می‌شوند.

- ایجاد یک فوتر با متن `"copy; 2024&"`.

3. CSS Styles:

- استایل‌های مختلف برای هدر، ناوبری، بخش اصلی، لینک‌ها، دکمه‌ها، و فوتر.

- افکت هورور بر روی لینک‌ها و دکمه‌ها.

- افکت متحرک بر روی تصاویر با استفاده از `transform: scale(1.05);`.

4. Images:

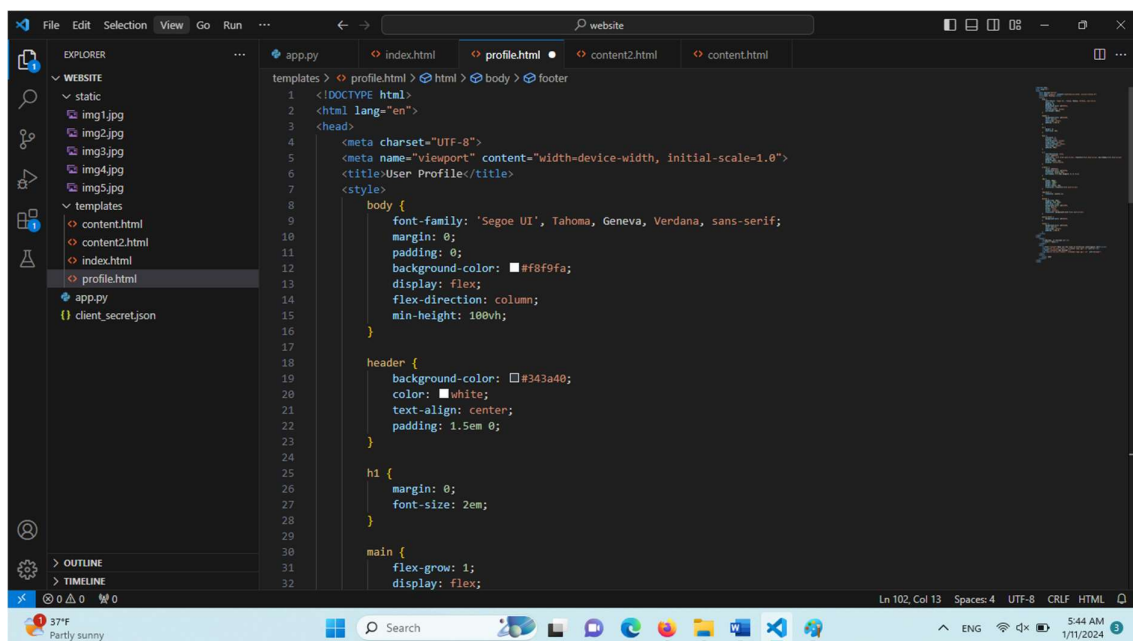
- استفاده از تصاویر با کلاس `floating-image` که با هورر بر روی آن‌ها بزرگ‌تر می‌شوند.

5. ویژگی‌های خاص:

- در استفاده از `url_for('static', filename='img3.jpg')` به نظر می‌رسد که از یک فریم‌ورک وب مانند Flask

استفاده شده است، زیرا `url_for` به طور معمول در این فریم‌ورک برای ایجاد URLهای استاتیک استفاده می‌شود.

Profile.html یک صفحه وب ساده با زبان HTML و CSS برای نمایش پروفایل کاربر ایجاد می‌کند. در اینجا توضیحاتی در مورد بخش‌های کلیدی این کد آورده شده است:



این کد یک صفحه وب ساده با زبان HTML و CSS برای نمایش پروفایل کاربر ایجاد می‌کند. در اینجا توضیحاتی در مورد بخش‌های کلیدی این کد آورده شده است:

1. ساختار اولیه HTML:

- `<DOCTYPE html>`: مشخص می‌کند که این صفحه از نوع HTML5 استفاده می‌کند.

- `<html lang="en">`: تگ اصلی HTML با مشخص کردن زبان صفحه.

- `<head>`: بخش اطلاعات سربرگ شامل متا تگ‌ها و تیتلر صفحه.

- `<meta charset="UTF-8">`: تعیین کدینگ کاراکتر به عنوان UTF-8.

- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: تنظیمات نمایش برای

دستگاه‌های مختلف.

- `<title>User Profile</title>`: عنوان صفحه.

2. استایل‌های CSS:

- تنظیمات استایل برای بدنه صفحه، هدر، مین، و فوتر.

- تعیین فونت، رنگ‌ها و سایر ویژگی‌های ظاهری.

3. هدر (`<header>`) صفحه:

- نمایش خوشامد به کاربر با استفاده از یک تیتل (`<h1>`) و نام کاربر که احتمالاً با استفاده از یک متغیر (`{{ username }}`) پر شده است.

- لینک بازگشت به صفحه اصلی (`Main`).

4. (`<main>`) صفحه:

- لینک‌های دیگر به محتواهای مختلف و تصاویر.

- تصاویر با افکت تغییر اندازه و لینک‌ها با افکت `hover`.

5. فوتر (`<footer>`) صفحه:

- نمایش کپی‌رایت با سال ۲۰۲۴ (`© 2024`).

کد از متغیرهای جلوه‌گر در محل‌هایی مانند `{{ username }}` و `{{ url_for('static', filename='img1.jpg') }}` برای افزودن مقادیر پویا به صفحه استفاده می‌کند که احتمالاً از یک فریم‌ورک وب مانند Flask یا چارچوب مشابه برای پروژه ایجاد شده است.

فایل‌های `content.html` و `content2.html` برای بررسی و نمایش دو سند در مورد هوش مصنوعی و زبان گو استفاده شده است:

```
1 <!DOCTYPE html>
2 <html lang="fa">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Golang</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      margin: 0;
11      padding: 0;
12      box-sizing: border-box;
13    }
14
15    header {
16      background-color: #f3f3f3;
17      color: #fff;
18      padding: 10px;
19      text-align: center;
20      display: flex;
21      justify-content: space-between;
22      align-items: center;
23    }
24
25    header button {
26      background-color: #4CAF50;
27      color: #fff;
28      padding: 10px;
29      border: none;
30      border-radius: 5px;
31      cursor: pointer;
32      transition: transform 0.3s ease, box-shadow 0.3s ease;
```

```
1 <!DOCTYPE html>
2 <html lang="fa">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>AI</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      margin: 0;
11      padding: 0;
12      box-sizing: border-box;
13    }
14
15    header {
16      background-color: #f3f3f3;
17      color: #fff;
18      padding: 10px;
19      text-align: center;
20      display: flex;
21      justify-content: space-between;
22      align-items: center;
23    }
24
25    header button {
26      background-color: #4CAF50;
27      color: #fff;
28      padding: 10px;
29      border: none;
30      border-radius: 5px;
31      cursor: pointer;
32      transition: transform 0.3s ease, box-shadow 0.3s ease;
```

این کد یک صفحه وب HTML است که در زبان فارسی نوشته شده است. در زیر توضیحاتی در مورد هر بخش از کد آورده شده است:

1. تنظیمات HTML:

- `<!DOCTYPE html>`: اعلام نسخه HTML.
- `<html lang="fa">`: تعیین زبان صفحه به فارسی.

2. تنظیمات Head:

- تعیین متا-تگ‌ها برای متن و مقیاس مشاهده (viewport).
- تعیین عنوان صفحه به "Golang".
- استایل‌ها با استفاده از تگ `<style>` برای استایل‌دهی به صفحه.

3. استایل‌دهی به بدنه صفحه (Body):

- استفاده از فونت Arial یا sans-serif برای متن‌ها.
- حذف حاشیه (margin) و فضای پرشده (padding) از بدنه صفحه.
- استفاده از flexbox برای ترتیب مرکزی و فاصله‌گذاری در هدر و دکمه‌ها.

4. هدر (Header):

- یک هدر با پس زمینه سیاه و متن سفید.
- یک عنوان اصلی با متن "Why Golang?".
- دو دکمه Telegram و Twitter که به توابع `shareOnTelegram()` و `shareOnTwitter()` متصل هستند.

5. توابع جاوااسکریپت (JavaScript Functions):

- `shareOnTelegram()`: باز کردن یک پنجره جدید برای ارسال لینک به تلگرام.
- `shareOnTwitter()`: باز کردن یک پنجره جدید برای ارسال لینک به توییتر.

6. محتوا (Main):

- یک محتوا با کلاس `content-box` که شامل یک پاراگراف درباره Golang است.

7. پاورقی (Footer):

- یک پاورقی با پس زمینه سیاه و متن سفید که حاوی سال 2024 است.
- این صفحه به زبان HTML و CSS طراحی شده و از جاوااسکریپت برای ایجاد عملکردی اضافی در دکمه‌ها استفاده می‌کند.

