



نویسنده: معصومه حسن پور

تمرین درس مهندسی اینترنت

استاد درس : امیر کیوان شفیعی

موضوع تمرین :

وب سایت با اعتبار سنجی توسط گوگل و اشتراک در شبکه های مجازی مانند تلگرام

مقدمه :

در عصر حاضر که فناوری اطلاعات در تمامی زوایای زندگی ما نفوذ کرده است، وبسایت‌ها دیگر تنها منابعی برای ارائه اطلاعات نیستند، بلکه به بسترهای فعال برای ایجاد ارتباطات کاربردی و موثر تبدیل شده‌اند. امنیت کاربران هنگام استفاده از خدمات آنلاین یکی از مهم‌ترین دغدغه‌های طراحان و توسعه‌دهندگان وب است. اعتبارسنجی (**Authentication**) و استفاده از شبکه‌های مجازی برای اشتراک گذاری محتوا، دو رکن اساسی به شمار می‌روند تا تجربه‌ی کاربری را آسان‌تر و امن‌تر سازند.

اعتبار سنجی از طریق گوگل (**Google Sign-In**) ، یکی از مطمئن‌ترین و کارآمدترین روش‌هایی است که به کاربران اجازه می‌دهد تا با استفاده از یکپارچگی هویت خود در گوگل، به سرعت و بدون نیاز به ثبت‌نام دوباره وارد سایت شوند. این روش، بهره‌وری سایت را افزایش می‌دهد و اعتماد کاربر را استحکام می‌بخشد.

از سوی دیگر، ادغام **API** های شبکه‌های اجتماعی مانند تلگرام به کاربران امکان می‌دهد تا محتوای دلخواه خود را بر روی پلتفرم‌های

گوناگون به اشتراک بگذارند، در نتیجه دامنه‌ی تأثیرگذاری و میزان وفاداری آنها را چند برابر می‌کند.

مستندات :

استفاده از راهکارهای تایید هویت پیشرفته مانند سیستم ورود به وسیله حساب گوگل، گامی به سوی امنیت و راحتی در دسترسی به محتوای وب محسوب می‌شود. این روش نوین به کاربران این امکان را می‌دهد که به سهولت و با امنیت بالا به ویژگی‌های سایت دسترسی یابند.

در ادامه به بررسی فریم‌ورک قابل اعتماد **Flask** می‌پردازیم که با زبان برنامه‌نویسی **Python**، امکان ساخت وب‌سایت‌ها و اپلیکیشن‌هایی کارآمد، واکنش‌گرا و قابل انعطاف را فراهم می‌آورد. این فریم‌ورک از معماری **MVC** پشتیبانی کرده و با ادغام موتور قالب‌گیری **Jinja۲**، تجربه‌ای بی‌نظیر از برنامه‌نویسی را ارائه می‌دهد.

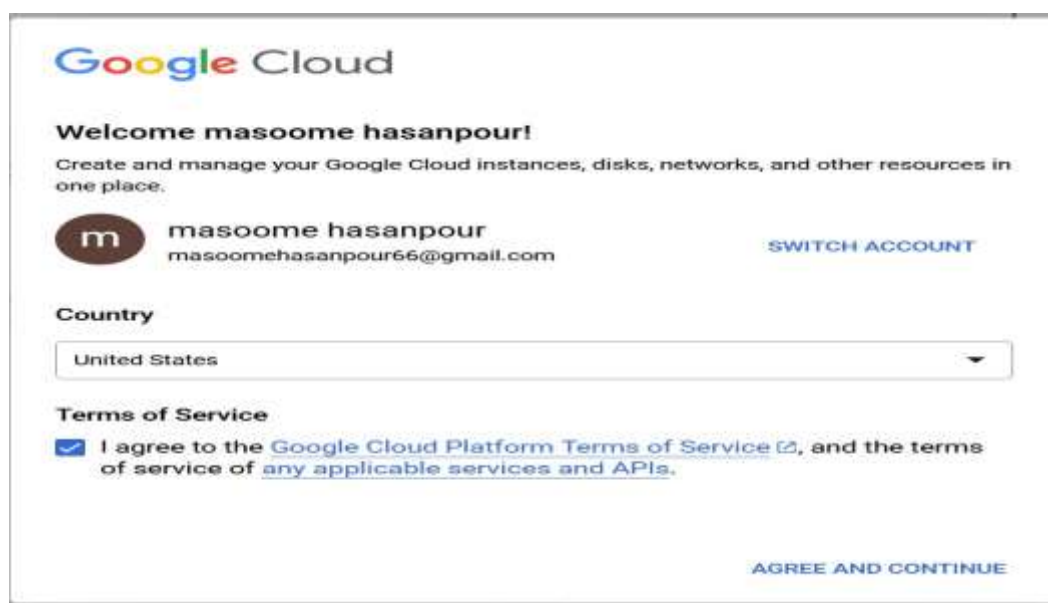
تمرکز این بخش، علاوه بر تایید هویت، بر روی چگونگی ادغام **API**‌های موجود در شبکه‌های اجتماعی نظیر تلگرام، واتس‌آپ و اینستاگرام است. به این صورت کاربران می‌توانند به راحتی محتوا را

به اشتراک بگذارند، که این خود فرصت‌هایی وسیعی برای اشتراک‌گذاری سریع و آسان محتوا و افزایش تعامل کاربری فراهم می‌آورد.

این مستند با زبانی شیوا و در قالب گام‌هایی مشخص، روند پیاده‌سازی ویژگی‌های مذکور را در یک وب‌سایت تعاملی، به همراه توصیف نحوه احراز اعتبار در کنار بهره‌گیری از **API** های شبکه‌های اجتماعی، تشریح می‌کند.

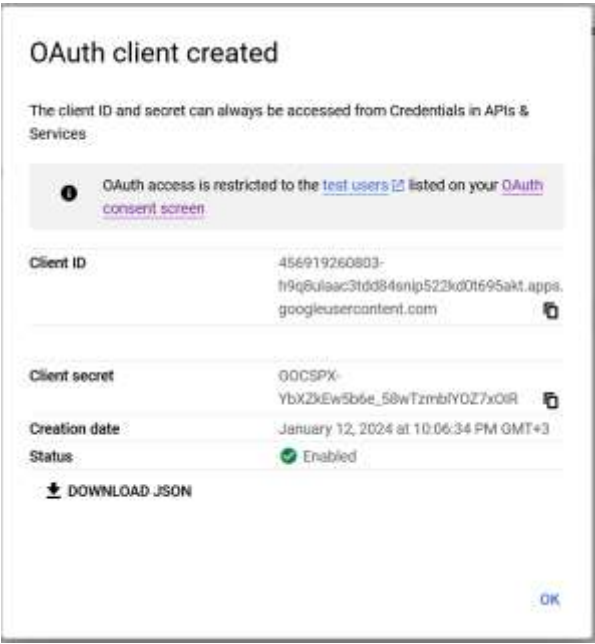
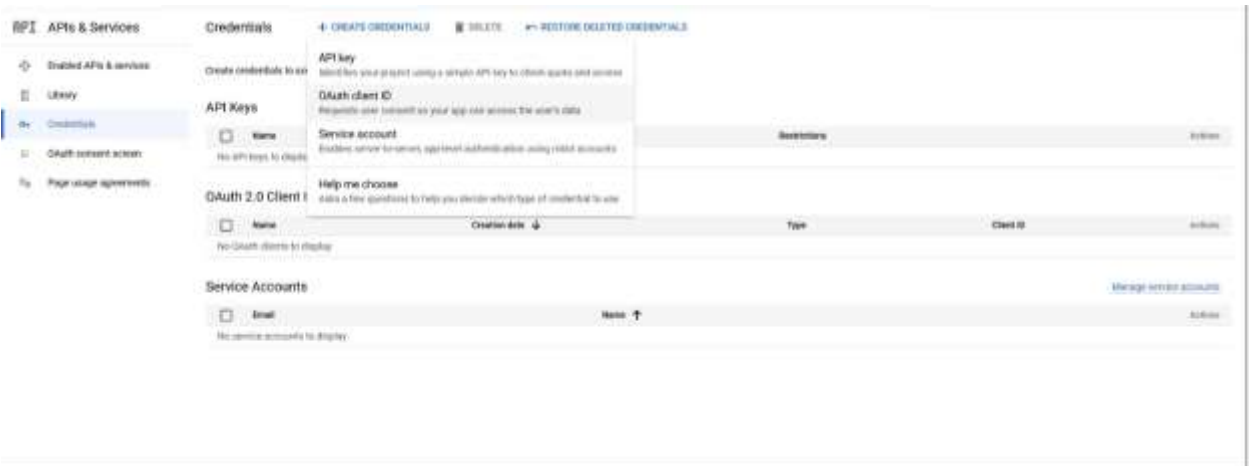
بخش اول: راه‌اندازی اعتبار سنجی Google Sign-In

• توضیحاتی در مورد نحوه‌ی ثبت برنامه در کنسول توسعه‌دهندگان گوگل



The screenshot shows the Google Cloud sign-in interface. At the top is the 'Google Cloud' logo. Below it, a welcome message reads 'Welcome masoome hasanpour!' followed by the instruction 'Create and manage your Google Cloud instances, disks, networks, and other resources in one place.' A user profile card displays a circular icon with the letter 'm', the name 'masoome hasanpour', and the email 'masoomehasanpour66@gmail.com'. To the right of the card is a 'SWITCH ACCOUNT' link. Below the profile card is a 'Country' dropdown menu currently set to 'United States'. Underneath is the 'Terms of Service' section, which includes a checked checkbox and the text 'I agree to the Google Cloud Platform Terms of Service, and the terms of service of any applicable services and APIs.' At the bottom right is an 'AGREE AND CONTINUE' button.

• روش ایجاد شناسه کاربری Client ID و رمز امنیتی Client Secret



- پیکربندی اعتبارسنجی **OAuth ۲.۰** و تعریف دامنه و مسیرهای برگشت

```
• 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
  )
```

- چگونگی استفاده از کتابخانه‌های کلاینت گوگل در سمت سرور و مشتری (**Client & Server libraries**)

```
• from google.oauth2 import id_token
  from google_auth_oauthlib.flow import Flow
  import google.auth.transport.requests
```

- فرایند احراز هویت و دریافت توکن‌ها و به روز رسانی اطلاعات کاربر با استفاده از **API** گوگل

بخش دوم: پیاده‌سازی فرآیند احراز هویت

- شرح مراحل فنی برای تعبیه احراز هویت گوگل در وبسایت:
 - ایجاد **URL** هایی برای شروع و خاتمه‌ی فرآیند احراز هویت
 - استفاده از کتابخانه‌های رسمی و مدیریت پروسه‌ی تأیید و

تبادل توکن

```
• @route.route('/login-by-google')
  def login_bygoogle():

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

  @route.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('show.home'))

```

◦ طراحی رابط کاربری جهت ورود و خروج کاربران با استفاده
از حساب گوگل خود



◦ توضیح معماری کلی برای ذخیره سازی و مدیریت اطلاعات
کاربرانی که وارد سیستم شده اند:

◦ مدل‌سازی دیتابیس برای حفظ اطلاعات کاربران

```
• class User(db.Model, UserMixin):  
    id = db.Column(db.Integer, primary_key=True)  
    email = db.Column(db.String(80), unique=True)  
    password = db.Column(db.String(80))  
    first_name = db.Column(db.String(80))  
  
class Note(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    data = db.Column(db.String(10000))  
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'))
```

◦ جداسازی داده‌های حاصل از احراز هویت

◦ اقدامات امنیتی مربوط به نگهداری توکن‌ها و داده‌های

حساس

بخش سوم: ادغام API شبکه‌های اجتماعی

• گام‌های عملی ادغام API تلگرام:

◦ دریافت لینک و توکن ارسال به تلگرام

◦ طراحی دکمه اشتراک گذاری داخل وبسایت

◦ روش‌های فراخوانی API تلگرام و ارسال پیام

• چگونگی استفاده از **API** های دیگر پلتفرم های محبوب مانند واتس

اپ و اینستاگرام

◦ مراحل ساخت درخواست ها به **API** ها

◦ اقدامات امنیتی برای جلوگیری از دسترسی های بی اجازه

◦ مدیریت خطاها و پاسخ های **API**

این مستند با تأکید بر سهولت درک و گام به گام بودن روند، به توسعه دهندگان کمک می کند تا بتوانند در کوتاه ترین زمان ممکن، سایت خود را به یک سرویس مجهز به اعتبارسنجی مدرن و کارآمد و یک پلتفرم قدرتمند برای به اشتراک گذاری محتوا تبدیل نمایند.

```
<div class="submit-btn">
  <button type="button" class="submit" onclick="location.href='/telegram';">
    <p>Send</p>
    <div class="submit-icon">
      <i class="fa-solid fa-paper-plane"></i>
    </div>
  </button>
  <button type="button" class="submit"
onclick="location.href='/logout';">
    <p>Exit</p>
    <div class="submit-icon">
      <i class="fa-solid fa-sign-out"></i>
    </div>
  </button>
```

بخش چهارم: بررسی نقشه‌ی راه پیاده‌سازی مکانیزم‌های به اشتراک گذاری

• ارائه دستورالعمل‌های مربوط به گنجاندن دکمه‌های اشتراک گذاری:

◦ بررسی رویکردهای UX/UI برای جلب حداکثر مشارکت کاربران

◦ تنظیم خاصیت‌های دسترس‌پذیری و دکمه‌های واکنش‌گرا

• معرفی بهترین شیوه‌های مدیریت و تحلیل وقایع ناشی از اشتراک گذاری:

◦ ردیابی و تجزیه و تحلیل واکنش‌های کاربران به اشتراک گذاری‌ها

◦ استفاده از اطلاعات تحلیلی برای بهبود استراتژی محتوایی و تعاملی

بخش پنجم: امنیت و حریم خصوصی در احراز هویت و

اشتراک گذاری

- ارزیابی ریسک‌ها و توسعه‌ی استراتژی‌های مقابله به منظور تأمین امنیت داده‌ها:

- اجرای سیاست‌های رمزنگاری داده و مدیریت دسترسی‌ها

```
app.secret_key = 'wwwf wwwf wwwf'
```

- تدابیر پیشنهادی برای حفظ حریم خصوصی و اطمینان از رضایت کاربران:

- بررسی قوانین حفاظت از داده‌ها مانند GDPR و تطابق با آنها

- روش‌های صحیح دریافت، ذخیره و پردازش اطلاعات شخصی کاربران

این مستندات به نوبه‌ی خود نقشه‌ی راهی جامع برای توسعه‌دهندگان فراهم می‌سازند تا با درک کامل و دقیق از هر جنبه‌ی فنی و حقوقی مرتبط بتوانند تجربه کاربری همه‌جانبه و ایمنی را در پروژه‌های وبسایت خود به ارمغان آورند.

(توضیحات) :

این برنامه نمونه‌ای از ساخت یک وب‌اپلیکیشن با استفاده از Flask ، که یک چارچوب (framework) سبک برای برنامه‌نویسی وب به زبان پایتون است، را با ادغام احراز هویت گوگل و ارائه‌ی عملکرد اشتراک‌گذاری در شبکه‌های اجتماعی نظیر تلگرام نشان می‌دهد. بخش‌های مختلف کد ارائه شده را توضیح می‌دهیم:

وارد کردن کتابخانه‌ها و تنظیمات اولیه:

```
```python
from flask import Flask, session, abort

...
```

```
db = SQLAlchemy()
DB_NAME = "appdata.db"
```
```

در ابتدا ماژول‌های مختلف Flask وارد می‌شوند `SQLAlchemy`
برای تعامل با پایگاه داده مورد استفاده قرار می‌گیرد و نمونه‌ای از آن
ایجاد می‌شود.

تابع کمکی برای احراز هویت:

```
```python
def login_is_required(function):
 def wrapper(*args, **kwargs):
 if "google_id" not in session:
 return abort(401)
 else:
 return function(*args, **kwargs)
 return wrapper
```
```

تابع `login_is_required` به عنوان یک دکوراتور عمل می کند که دسترسی به تنها کاربران احراز هویت شده را محدود می کند.

ساخت نمونه اپلیکیشن:

```
```python
def create_app():
 ...
 return app
```
```

تابع `create_app` یک کارخانه برنامه‌ساز است. این تابع نمونه‌ای از اپلیکیشن Flask ایجاد می‌کند، پایگاه داده را تنظیم می‌کند و `LoginManager` را برای مدیریت جلسات کاربری راه‌اندازی می‌کند.

مقدمات پایگاه داده:

```
``python
def create_database(app):
    if not path.exists('website/' + DB_NAME):
        db.create_all(app=app)
        print('Created Database!')
...

```

تابع کمکی `create_database` بررسی می‌کند که آیا فایل پایگاه داده وجود دارد و در صورت عدم وجود آن را ایجاد می‌کند.

ثبت Blueprint ها:

```
```python
route = Blueprint('route', __name__)
...
app.register_blueprint(show, url_prefix='/')
app.register_blueprint(route, url_prefix='/')
```
```

Blueprint های `route` برای مسیرهای احراز هویت و `show`
برای مسیرهای محتوای اصلی ایجاد و به اپلیکیشن ثبت می شوند تا
برنامه را به اجزاء سازمان دهی کنند.

تنظیمات OAuth و مسیرها:

```
```python
```

```
client_secrets_file = ...
```

```
flow = Flow.from_client_secrets_file(...)
```

```
...
```

```
@route.route('/login-by-google')
```

```
...
```

```
@route.route('/callback')
```

```
...
```

```
@route.route('/logout')
```

```
...
```

```
```
```

تنظیم مشتری OAuth گوگل و مسیرهای Flask برای کنترل ورود کاربران با استفاده از Google ، فراخوانی OAuth و خروج کاربران فراهم می‌شود.

اشتراک گذاری در تلگرام:

```
```python
@route.route('/telegram')
@login_required
...
```
```

یک نقطه پایانی اشتراک گذاری تلگرام که با استفاده از دکوراتور `@login_required` از Flask-Login محافظت می‌شود، که کاربر را به تلگرام با یک پیام از پیش پر شده حاوی اطلاعات کاربر هدایت می‌کند.

ORM کاربر و یادداشت‌ها:

```
```python
class User(db.Model, UserMixin):
 ...
class Note(db.Model):
 ...
```
```

کلاس‌های `User` ORM و `Note` با استفاده از `SQLAlchemy` برای تعامل با جداول داده‌ای اپلیکیشن تعریف می‌شوند. `UserMixin` یک کلاس کاربردی از Flask-Login است که پیاده‌سازی‌های عمومی که برای بیشتر کلاس‌های مدل کاربر مناسب است را شامل می‌شود.

مسیر اصلی و مدیریت یادداشت:

```
```python
@show.route('/', methods=['GET', 'POST'])
@login_required

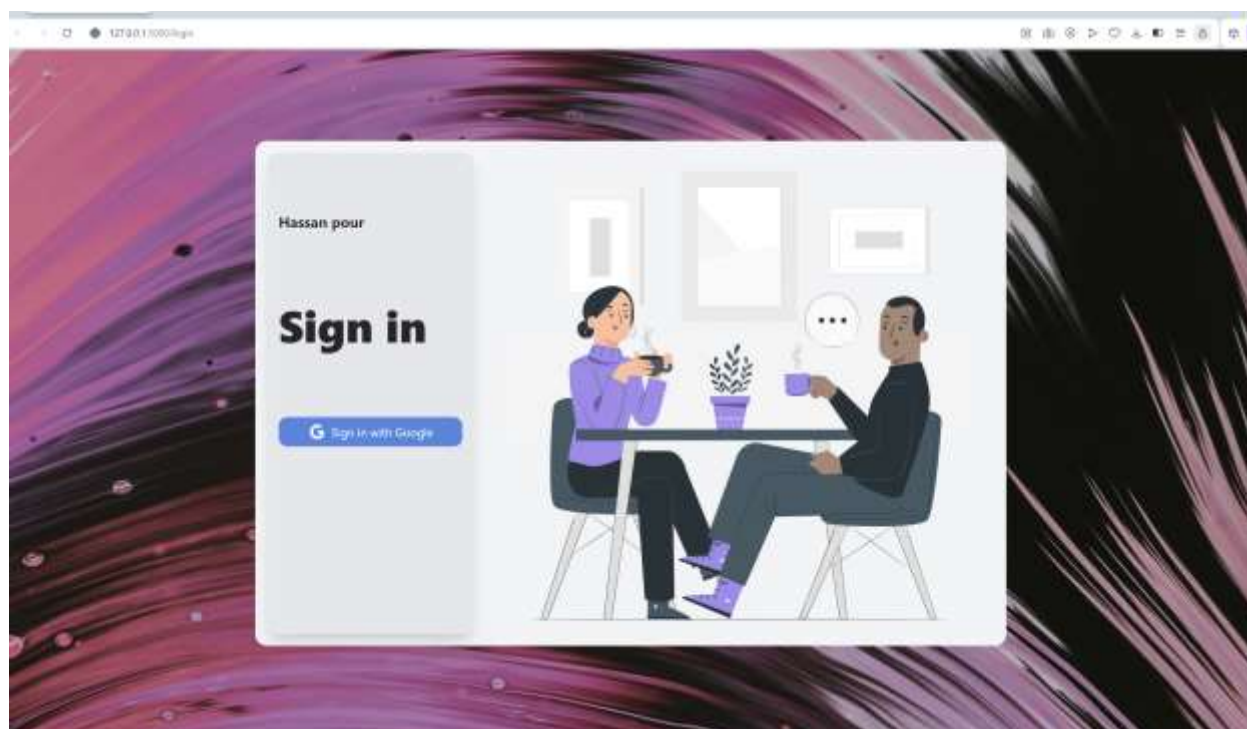
def home():
 ...
```
```

مسیر اصلی خانه که به کاربران وارد شده اجازه می‌دهد تا یادداشت‌های جدید از طریق رابط وب ارسال کنند. اگر کاربر یادداشتی ارسال کند، به پایگاه داده اضافه می‌شود.

اشتباهات و اصلاحات:

- کتابخانه `cachecontrol` برای ارائه‌ی مکانیزم کش مورد استفاده قرار می‌گیرد و از `pip._vendor` وارد می‌شود، که ایده آل نیست. معمولاً باید وابستگی‌ها را از طریق `pip` نصب کرد و به طور مستقیم وارد کرد.

صفحه ورود و اعتبار سنجی گوگل :



در طراحی این صفحه از CSS فریمورک Tailwind و فونت‌های گوگل استفاده می‌کنیم.

این صفحه، طراحی شده برای ورود کاربران، یک تجربه کاربری (UI) مدرن و پاسخگو را فراهم می‌کند که با دستگاه‌های مختلف از جمله تلفن‌های همراه، تبلت‌ها و رایانه‌ها سازگار است. بک‌گراند پویا و طراحی کارت ورود با استفاده از سایه‌های با ظرافت (shadow)، تعادل بصری و همچنین تاکید بر دسترس‌پذیری و خوانایی را ایجاد می‌کند.

با استفاده از Tailwind، کنترل‌رهایی برای این داریم که صفحه به صورت کامل و بدون تکرار تصویر پس‌زمینه و همچنین برای این که پس‌زمینه به صورت پوشش دهنده (cover) بر روی کل صفحه گسترش پیدا کند. تصویر استفاده شده در پس‌زمینه، فضایی گرم و دعوت کننده ایجاد می‌کند.

• بخش ورودی: فرم ورود با زمینه‌ای به رنگ خاکستری متمایل به سیاه و شفافیت مختصر (opacity) به همراه سایه‌های پرتراکم

ارائه شده است. این طراحی هم راحتی را در قرائت فراهم می کند و هم تعامل بصری عمیق تری به کاربر القا می کند.

ورود با گوگل: دکمه ورود که با گوگل استایل داده شده، طراحی تمیز و واضحی دارد، همراه با آیکون Google برای هدایت کاربر به سرویس احراز هویت گوگل. کاربر با کلیک بر روی این دکمه به سرویس ورود با گوگل هدایت می شود.

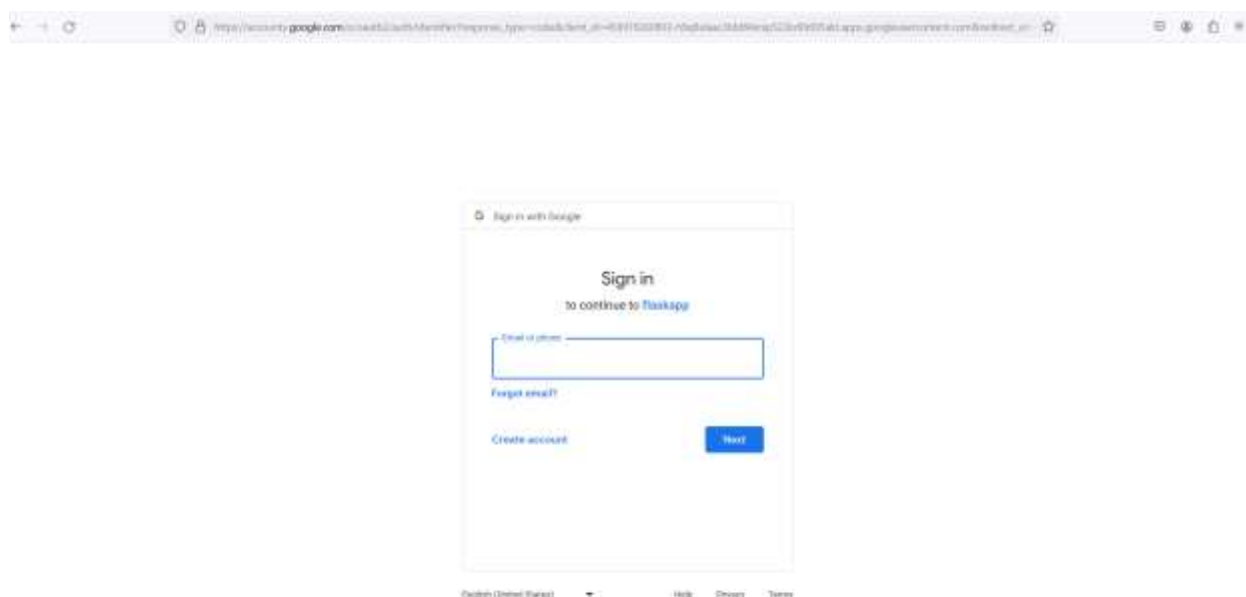
طراحی واکنش گرا: قسمت تصویری که در کنار فرم ورود قرار دارد، در دستگاه های کوچکتر مخفی می شود و تنها در صفحه نمایش های بزرگتر نشان داده می شود که این امر صفحه را برای تجربه های کاربری بر روی دستگاه های متفاوت بهینه می کند.

دستورالعمل های تکنیکی:

• استفاده از Tailwind CSS برای طراحی واکنش گرا و مدرن بدون نیاز به نوشتن کدهای CSS از صفر.

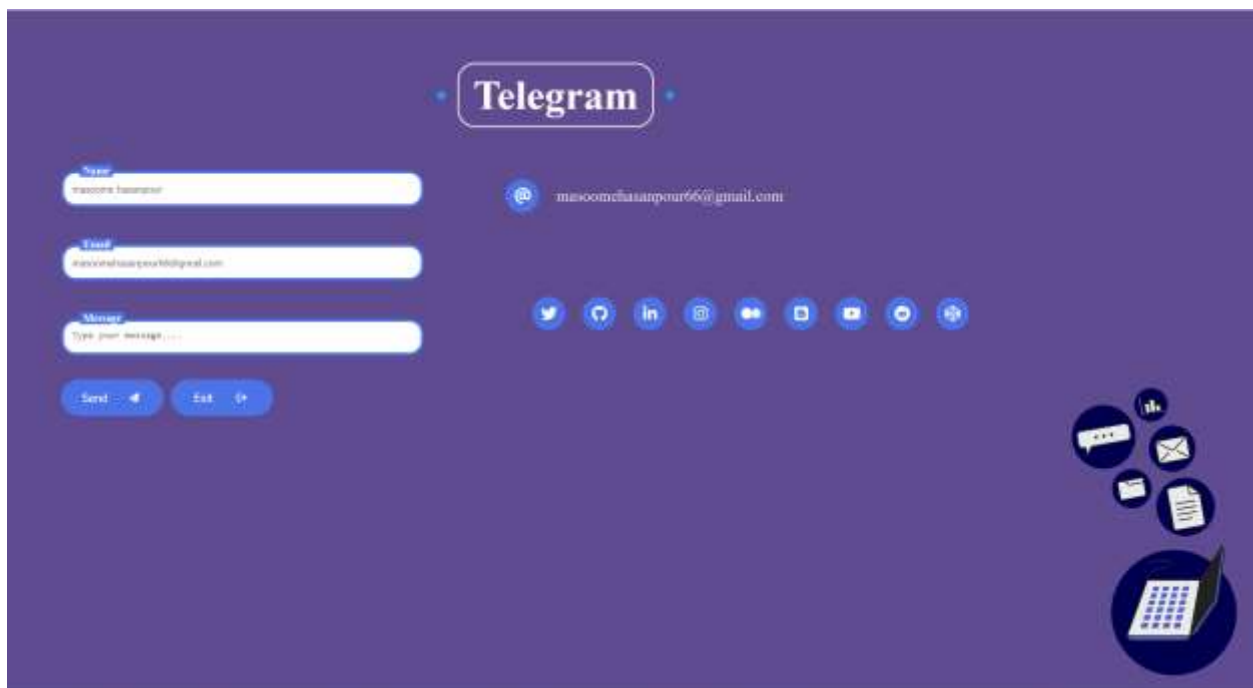
• استفاده از فونت های گوگل فونت برای تضمین یک تجربه محتوایی یکنواخت و قابل دسترس در تمام مرورگرها و دستگاه ها.

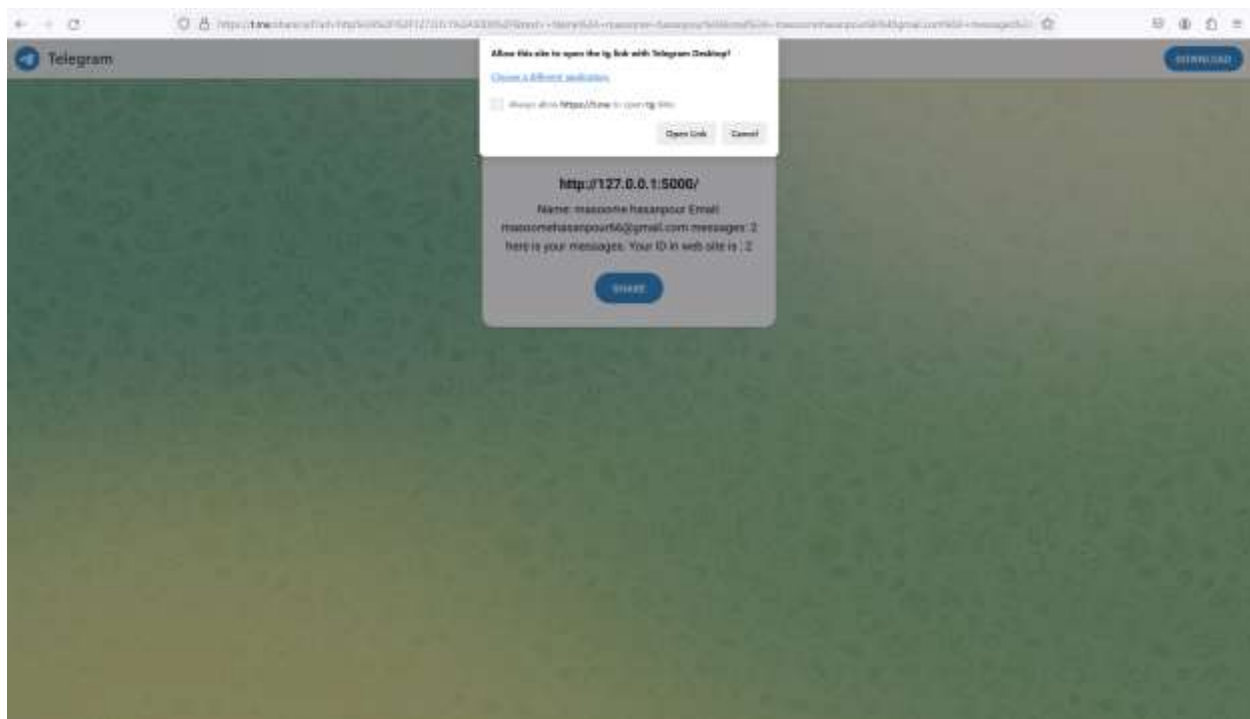
- توجه به UX/UI در طراحی تا حصول اطمینان از اینکه کاربران به راحتی می‌توانند هدایت شوند و به هدف نهایی خود برسند



ارسال در تلگرام :

صفحه زیر یک بخش تماس‌ها (Contacts Section) را توصیف می‌کند که به منظور ارسال پیام به کاربران از طریق تلگرام طراحی شده است. در این صفحه از فونت آوسام برای نمایش آیکون‌ها استفاده شده و سبک‌های بصری از طریق فایل CSS محلی به نام style.css اعمال می‌شوند.





- برای اعمال سبک‌های پیش‌فرض و اصلی استفاده از فونت‌های آوسام و CSS توصیه می‌شود تا قابلیت نمایش در مرورگرهای مختلف تضمین شود.

- عناصر ورودی به همراه برچسب‌های واضح و دکمه‌های دارای تنظیم راحتی بصری برای تجربه‌ی بهتر کاربر تنظیم شده‌اند.

- لینک‌ها و آیکن‌ها به منظور افزایش تعامل پذیری کاربر و تسهیل دسترسی به شبکه‌های مختلف اجتماعی در نظر گرفته شده‌اند.

- برای بهبود دسترس پذیری، ممکن است استفاده از صفتهای aria-label برای دکمه ها و پیوندها لازم باشد تا اهداف و کارکردها به وضوح برای کاربران نابینا توصیف شود.

این صفحه با تأکید بر طراحی کاربر پسند و تعاملی، همراه با فراهم کردن امکان ارتباط مستقیم و فوری، یک رابط کاربری مناسب برای ایجاد ارتباط بین کاربران و سایت پیش رو می گذارد.

