مستندات
تهیه کننده: پریا عبدالهی

مقدمه

پروژه وبلاگ روانشناسی کودکان به کاربران اجازه می‌دهد مقالات روانشناسی کودکان را منتشر کنند، مشاهده کنند و به اشتراک بگذارند. این پروژه با استفاده از تکنولوژی‌های مدرن و ویژگی‌های خاص، تجربه کاربری خوبی را برای کاربران فراهم می‌کند.

مشخصات فنی پروژه وبلاگ روانشناسی کودکان

پروژه وبلاگ روانشناسی کودکان از تکنولوژی‌های زیر استفاده می‌کند:

- فریمورک فلسک برای توسعه وب‌سایت
- پایتون به عنوان زبان برنامه‌نویسی
- پایگاه داده MySQL برای ذخیره‌سازی داده‌ها
- ماژول Flask-SQLAlchemy برای تعامل با پایگاه داده
- ماژول Flask-Login برای مدیریت کاربران
- ماژول Flask-WTF برای فرم‌های تعاملی

نحوه کارکرد پروژه

پروژه وبلاگ روانشناسی کودکان به شرح زیر کار می‌کند:

- کاربران می‌توانند با ایجاد حساب کاربری، وارد وبلاگ شوند.
- کاربران می‌توانند مقالات جدیدی در وبلاگ منتشر کنند.
- کاربران می‌توانند مقالات موجود را مشاهده کنند و آنها را به اشتراک بگذارند.

ویژگی‌های خاص پروژه

پروژه وبلاگ روانشناسی کودکان دارای ویژگی‌های خاص زیر است:

- سیستم مدیریت کاربران که به کاربران اجازه می‌دهد حساب کاربری ایجاد کنند، نام کاربری و رمز عبور خود را تغییر دهند و مقالات منتشر کنند.
- سیستم مدیریت مقالات که به کاربران اجازه می‌دهد مقالات جدیدی ایجاد کنند، مقالات موجود را ویرایش کنند و مقالات حذف کنند.
- سیستم به اشتراک‌گذاری که به کاربران اجازه می‌دهد مقالات را در شبکه‌های اجتماعی به اشتراک بگذارند.

بررسی ساختار پروژه و نحوه ایجاد آن:

ما فولدری به نام blog را ایجاد کردیم که درون این فولدر، فایل‌ها و فولدرهای پروژه قرار میگیرد.

اولین فایلی که ایجاد شده run.py بوده که برنامه در این فایل اجرا میشود.

و چندین فایل پایتونی دیگر ایجاد کردیم که به‌مرور به همه آنها می‌پردازیم.
فایل run.py :

```
from blog import app
```

مستندات
تهیه کننده: پریا عبدالهی

:'if name == 'main

app.run(debug=True)

فایل run.py نقطه شروع اپلیکیشن است. این فایل، اپلیکیشن را راه‌اندازی می‌کند و آن را در حالت توسعه اجرا می‌کند.

در حالت توسعه، خطاها به صورت خودکار نمایش داده می‌شوند و کدهای جدید بدون نیاز به راه‌اندازی مجدد اپلیکیشن بارگذاری می‌شوند.

برای راه‌اندازی اپلیکیشن در حالت توسعه، می‌توانید از دستور زیر استفاده کنید:

python run.py

این دستور، اپلیکیشن را در پورت 5000 اجرا می‌کند.

فایل __init__.py__ :

این فایل، اپلیکیشن فلسک را راه‌اندازی می‌کند و تنظیمات آن را پیکربندی می‌کند. همچنین، کتابخانه‌های لازم برای تعامل با پایگاه داده، رمزنگاری رمز عبور و احراز هویت کاربر را وارد می‌کند.

from flask import Flask

import SQLAlchemy ی from flask_sqlalchem

این خطوط کتابخانه‌های فلسک و SQLAlchemy را وارد می‌کنند که برای ایجاد یک برنامه وب با پایگاه داده ضروری هستند.

from flask_bcrypt import Bcrypt

from flask_login import LoginManager

این خطوط کتابخانه‌های Bcrypt و LoginManager را وارد می‌کنند که به ترتیب برای رمزنگاری رمز عبور و احراز هویت کاربر استفاده می‌شوند.

app = Flask(name)

'app.config['SECRET_KEY'] = '3e5e4c1b79c0df2a4dbc849e19296d1d

این خطوط اپلیکیشن فلسک را راه‌اندازی می‌کنند و کلید مخفی اپلیکیشن را تنظیم می‌کنند. کلید مخفی برای تولید امضای رمزنگاری ایمن استفاده می‌شود.

'app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///../blog.db

این خط URL پایگاه داده را برای اپلیکیشن تنظیم می‌کند. در این مورد، پایگاه داده یک فایل SQLite به نام blog.db است که در همان دایرکتوری اپلیکیشن فلسک قرار دارد.

db = SQLAlchemy(app)

این خط یک نمونه از شیء SQLAlchemy را ایجاد می‌کند و آن را با اپلیکیشن فلسک مرتبط می‌کند، که تعاملات پایگاه داده را فعال می‌کند.

bcrypt = Bcrypt(app)

مستندات
تهیه کننده: پریا عبدالهی

این خط یک نمونه از شیء Bcrypt را ایجاد می‌کند و آن را با اپلیکیشن فلسک مرتبط می‌کند، که رمزنگاری رمز عبور را با استفاده از الگوریتم bcrypt فعال می‌کند.

login_manager = LoginManager(app)

این خط یک نمونه از شیء LoginManager را ایجاد می‌کند و آن را با اپلیکیشن فلسک مرتبط می‌کند، که احراز هویت کاربر را فعال می‌کند.

'login_manager.login_view = 'login

'login_manager.login_message = 'Please login first

'login_manager.login_message_category = 'info

این خطوط شیء LoginManager را با تنظیمات دلخواه پیکربندی می‌کنند. تنظیم login_view نشان می‌دهد که کاربر در هنگام تلاش برای دسترسی به یک مسیر محدود بدون ورود به سیستم به کجا هدایت می‌شود. تنظیم login_message پیامی را برای نمایش به کاربران غیرمجاز فراهم می‌کند. تنظیم login_message_category دسته پیام را تنظیم می‌کند که تعیین می‌کند سبک پیام چگونه باشد.

from blog import routes

این خط ماژول routes را وارد می‌کند که شامل مسیرهای اپلیکیشن است. مسیرها مسیرهای URL و توابع نمایش مربوطه را که درخواست‌های ورودی را مدیریت می‌کنند، تعریف می‌کنند.

نکته: اگر این خط کد اولین خط کد برنامه باشد و در آخر نباشد ما به مشکل ایمپورت های حلقوی دچار میشویم.

فایل routes.py :

ساختار کلی

فایل routes.py مسیرهای URL و توابع مربوطه را برای یک برنامه وبلاگ تعریف می کند. از کتابخانه های Flask، Flask-Login و Flask-Oauthlib به ترتیب برای مسیریابی، احراز هویت کاربر و عملکرد OAuth استفاده می کند.

توابع مسیریابی

home(): صفحه اصلی را مدیریت می کند و فهرستی از همه پست های منتشر شده را نمایش می دهد.

detail(post_id): جزئیات یک پست خاص را بر اساس ID پست ارائه شده نمایش می دهد.

register(): ثبت نام کاربر را مدیریت می کند، داده های فرم ارسالی را اعتبارسنجی می کند و یک حساب کاربری جدید ایجاد می کند.

login(): ورود کاربر را مدیریت می کند، داده های فرم ارسالی را اعتبارسنجی می کند و کاربر را احراز هویت می کند.

logout(): خروج کاربر را مدیریت می کند، کاربر را امضا می کند و به صفحه اصلی هدایت می کند.

فرم ها

مستندات
تهیه کننده: پریا عبدالهی

RegistrationForm: برای ثبت نام کاربر استفاده می شود، نام کاربری، آدرس ایمیل و ورودی رمز عبور را جمع آوری می کند.

LoginForm: برای ورود کاربر استفاده می شود، ورودی آدرس ایمیل و رمز عبور را جمع آوری می کند.

UpdateProfileForm: برای به روز رسانی اطلاعات نمایه کاربر استفاده می شود، معمولاً شامل نام کاربری و آدرس ایمیل می شود.

PostForm: برای ایجاد و ویرایش پست های وبلاگ استفاده می شود، عنوان، محتوا و اطلاعات نویسنده را جمع آوری می کند.

احراز هویت

current_user: امکان دسترسی به شیء کاربری که در حال حاضر وارد شده است را فراهم می کند.

login_user(user): کاربر ارائه شده را احراز هویت می کند و آن ها را با جلسه فعلی مرتبط می کند.

logout_user(): کاربر فعلی را خارج می کند و جلسه آن ها را باطل می کند.

login_required: دکوراتور مورد استفاده برای محافظت از مسیرهایی که نیاز به احراز هویت دارند.

OAuth

oauth: شیء مشتری OAuth Flask-Oauthlib، که برای ادغام با ارائه دهندگان OAuth استفاده می شود.

home(): این تابع صفحه اصلی را مدیریت می کند. ابتدا، آن یک شیء Post را با استفاده از عبارت پرس و جو Post.query.all() ایجاد می کند. سپس، آن شیء را به یک تابع render_template() ارسال می کند که صفحه home.html را با فهرستی از پست ها ایجاد می کند.

detail(post_id): این تابع جزئیات یک پست خاص را بر اساس ID پست ارائه شده نمایش می دهد. ابتدا، آن یک شیء Post را با استفاده از عبارت پرس و جو Post.query.get_or_404(post_id) ایجاد می کند. سپس، آن شیء را به یک تابع render_template() ارسال می کند که صفحه detail.html را با جزئیات پست ایجاد می کند.

register(): این تابع ثبت نام کاربر را مدیریت می کند. ابتدا، آن یک شیء RegistrationForm را ایجاد می کند. سپس، آن شیء را اعتبارسنجی می کند. اگر فرم معتبر باشد، آن یک شیء User جدید ایجاد می کند و آن را در پایگاه داده ذخیره می کند. سپس، آن یک پیام تأیید را به کاربر نشان می دهد.

login(): این تابع ورود کاربر را مدیریت می کند. ابتدا، آن یک شیء LoginForm را ایجاد می کند. سپس، آن شیء را اعتبارسنجی می کند. اگر فرم معتبر باشد، آن یک کاربر را از پایگاه داده با استفاده از آدرس ایمیل ارائه شده جستجو می کند. سپس، آن رمز عبور کاربر را با رمز عبور ارائه شده مقایسه می کند. اگر رمز عبور مطابقت داشته باشد، آن کاربر را وارد می کند و یک پیام تأیید را به کاربر نشان می دهد.

logout(): این تابع خروج کاربر را مدیریت می کند. ابتدا، آن کاربر فعلی را با استفاده از تابع logout_user() خارج می کند. سپس، آن یک پیام تأیید را به کاربر نشان می دهد.

:detail(post_id)

نمایش یک پست خاص را مدیریت می کند.

مستندات
تهیه کننده: پریا عبدالهی

پست با post_id ارائه شده را با استفاده از روش Post.query.get_or_404(post_id) بازیابی می کند.

پست بازیابی شده را برای رندر کردن به قالب detail.html ارسال می کند.

:post/<int:post_id>/delete

پست مشخص شده را حذف می کند.

با بررسی اینکه آیا نویسنده پست با کاربر فعلی وارد شده مطابقت دارد، اطمینان حاصل می کند که کاربر فعلی مجاز به حذف پست است.

پست را از پایگاه داده با استفاده از روش db.session.delete(post) حذف می کند.

تغییرات را در پایگاه داده ثبت می کند.

پیام موفقیت را با استفاده از flash('post deleted', 'info') و redirect(url_for('home')) نمایش می دهد و به صفحه اصلی هدایت می کند.

:post/<int:post_id>/update

به‌روزرسانی یک پست موجود را مدیریت می کند.

با بررسی اینکه آیا نویسنده پست با کاربر فعلی وارد شده مطابقت دارد، اطمینان حاصل می کند که کاربر فعلی مجاز به به‌روزرسانی پست است.

یک فرم برای ویرایش پست با استفاده از کلاس PostForm() نمایش می دهد.

ارسال فرم را با استفاده از روش validate_on_submit() کلاس PostForm() اعتبارسنجی می کند.

عنوان و محتوای پست را با استفاده از داده های فرم به‌روز می کند.

تغییرات را در پایگاه داده ثبت می کند.

پیام موفقیت را با استفاده از flash('post updated', 'info') و redirect(url_for('detail', post_id=post.id)) نمایش می دهد و به صفحه جزئیات پست هدایت می کند.

:()new_post

ایجاد یک پست جدید را مدیریت می کند.

با بررسی دکوراتور login_required اطمینان حاصل می کند که کاربر فعلی مجاز به ایجاد پست است.

یک فرم برای ایجاد یک پست جدید با استفاده از کلاس PostForm() نمایش می دهد.

ارسال فرم را با استفاده از روش validate_on_submit() کلاس PostForm() اعتبارسنجی می کند.

یک شیء Post جدید با استفاده از داده های فرم ایجاد می کند.

پست جدید را به پایگاه داده با استفاده از روش db.session.add(post) اضافه می کند.

تغییرات را در پایگاه داده ثبت می کند.

پیام موفقیت را با استفاده از flash('post created', 'info') و redirect(url_for('home')) نمایش می دهد و به صفحه اصلی هدایت می کند.

مستندات
تهیه کننده: پریا عبدالهی

این مسیرها عملکرد اصلی برنامه وبلاگ را مدیریت می کنند، از جمله مدیریت پست ها، نمایه های کاربر و احراز هویت کاربر. آنها استفاده از Flask، Flask-Login و Flask-WTF را برای ایجاد یک برنامه وبلاگ ایمن و کاربرپسند نشان می دهند.

توضیحات بیشتر

:()home

این مسیر صفحه اصلی را مدیریت می کند که فهرستی از همه پست های منتشر شده را نمایش می دهد.

این کار با بازیابی تمام پست ها از پایگاه داده با استفاده از روش Post.query.all() انجام می شود.

سپس، این پست ها به قالب home.html ارسال می شوند تا نمایش داده شوند.

:detail(post_id)

این مسیر نمایش یک پست خاص را مدیریت می کند.

این کار با بازیابی پست با post_id ارائه شده از پایگاه داده با استفاده از روش Post.query.get_or_404(post_id) انجام می شود.

سپس، این پست به قالب detail.html ارسال می شود تا نمایش داده شود.

:post/<int:post_id>/delete

این مسیر پست مشخص شده را حذف می کند.

این کار با بررسی اینکه آیا نویسنده پست با کاربر فعلی وارد شده مطابقت دارد، انجام می شود.

علاوه‌بر لاگین معمولی به این پروژه لاگین با گوگل هم اضافه شده.

کد زیر برای احراز هویت Google OAuth با Flask و کتابخانه oauth نوشته شده است. در ادامه توضیحاتی درباره هر بخش آن آمده است:

پروژه Google OAuth:

)google = oauth.remote_app

'google',

consumer_key='510368579935-
'491va33ab3do6mkigl6g9sn2amn34md0.apps.googleusercontent.com',

'consumer_secret='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',

}=request_token_params

```
    'scope': 'email',
},
'base_url='https://www.googleapis.com/oauth2/v1',
request_token_url=None,
'access_token_method='POST',
'access_token_url='https://accounts.google.com/o/oauth2/token',
'authorize_url='https://accounts.google.com/o/oauth2/auth',
)
```

این قسمت پروژه Google OAuth را با اطلاعات مورد نیاز تنظیم می کند، مانند consumer_key و consumer_secret.

روت ورود Google:

```
@app.route('/login/google')
def google_login():
    return google.authorize(callback=url_for('google_authorized', _external=True))
```

این قسمت یک روت برای شروع فرایند ورود به Google تعریف می کند. این روت کاربر را به صفحه احراز هویت Google OAuth هدایت می کند.

روت بازگشت از Google:

```
@app.route('/login/google/callback')
@google.authorized_handler
def google_authorized(resp):
    ...
```

این قسمت یک روت برای دریافت بازگشت از Google پس از احراز هویت کاربر تعریف می کند. دکوریتور @google.authorized_handler بررسی می کند که کاربر احراز شده است و پاسخ را به تابع google_authorized پاس می دهد.

تابع احراز شده Google:

```
if resp is None or resp.get('access_token') is None:
    پردازش رد شدن درخواست
    ...
```

مستندات
تهیه کننده: پریا عبدالهی

:else

ذخیره توکن دسترسی در سشن

session['google_token'] = (resp['access_token'], '')

دریافت اطلاعات کاربر با استفاده از توکن دسترسی

user_info = google.get('userinfo')

نمایش پیام با موفقیت و انتقال به صفحه اصلی

flash('ورود با موفقیت انجام شد: ' + user_info.data['email'])

return redirect(url_for('home'))

این تابع بررسی می کند که کاربر احراز شده است و دسترسی را به کاربر داده می کند. اگر کاربر رد شده است، یک پیام خطا نمایش داده می شود. اگر کاربر احراز شده است، توکن دسترسی را در سشن ذخیره می کند، اطلاعات کاربر را با استفاده از توکن دسترسی دریافت می کند، پیام با موفقیت نمایش داده می شود و به صفحه اصلی هدایت می شود.

فایل models.py :

کدهای این فایل یک مدل کاربر و یک مدل پست برای یک وب سایت وبلاگ را تعریف می کند.

مدل کاربر

id یک کلید اصلی است که هر کاربر را به طور منحصر به فرد شناسایی می کند.

username نام کاربری کاربر را ذخیره می کند.

email آدرس ایمیل کاربر را ذخیره می کند.

password رمز عبور هش شده کاربر را ذخیره می کند.

posts یک رابطه چند به یک بین کاربران و پست ها را تعریف می کند. این بدان معناست که یک کاربر می تواند چندین پست داشته باشد و هر پست متعلق به یک کاربر است. گزینه backref='author' مشخص می کند که ویژگی author کلاس Post به شیء User مربوطه اشاره خواهد کرد.

lazy=True مشخص می کند که پست ها به صورت تنبل بارگیری می شوند، به این معنی که فقط زمانی بارگیری می شوند که واقعاً مورد نیاز باشند. این روشی کارآمدتر برای مدیریت روابط است، به خصوص وقتی یک کاربر تعداد زیادی پست داشته باشد.

مدل پست

id یک کلید اصلی است که هر پست را به طور منحصر به فرد شناسایی می کند.

title عنوان پست را ذخیره می کند.

مستندات
تهیه کننده: پریا عبدالهی

date تاریخ و زمان ایجاد پست را ذخیره می کند.

content محتوای پست را ذخیره می کند.

user_id به ستون id جدول User اشاره می کند، که نشان می دهد کاربری که پست را ایجاد کرده است.

این کد یک کاربر جدید با نام کاربری فلان و آدرس ایمیل فلان ایجاد می کند. سپس یک پست جدید با عنوان عنوان پست، تاریخ ایجاد datetime.now() و محتوای محتوای پست ایجاد می کند. در نهایت، کاربر فعلی را بارگیری می کند و همه پست های آن کاربر را بارگیری می کند.

نکته:

posts = db.relationship('Post', backref='author', lazy=True)

این خط درون دیتابیس ستونی ایجاد نمیکند و فقط روابط را نشان میدهد.

رابطه به صورت چند به یک است یعنی یک کاربر میتواند چند پست ایجاد کند.

فایل form.py :

فرم‌های Flask را برای ثبت نام کاربر، ورود به سیستم، به‌روزرسانی پروفایل و ایجاد پست تعریف می‌کند. این فرم‌ها برای جمع‌آوری ورودی کاربر و اعتبارسنجی آن قبل از ارسال آن به منطق برنامه برای پردازش استفاده می‌شوند.

کلاس‌های فرم

این فایل حاوی چهار کلاس فرم Flask است:

RegistrationForm: این فرم برای ثبت نام کاربر استفاده می‌شود. نام کاربری، آدرس ایمیل، رمز عبور و رمز عبور تأیید را از کاربر جمع‌آوری می‌کند. همچنین داده‌ها را برای اطمینان از اینکه نام کاربری منحصربه‌فرد است، ایمیل معتبر است و گذرواژه‌ها مطابقت دارند، اعتبارسنجی می‌کند.

LoginForm: این فرم برای ورود به سیستم کاربر استفاده می‌شود. آدرس ایمیل و رمز عبور کاربر را جمع‌آوری می‌کند. داده‌ها را برای اطمینان از اینکه ایمیل وجود دارد و رمز عبور صحیح است، اعتبارسنجی می‌کند.

UpdateProfileForm: این فرم برای به‌روزرسانی پروفایل کاربر استفاده می‌شود. نام کاربری و آدرس ایمیل کاربر را جمع‌آوری می‌کند. همچنین داده‌ها را برای اطمینان از اینکه نام کاربری منحصربه‌فرد است، ایمیل معتبر است و کاربر در حال تلاش برای به‌روزرسانی پروفایل خود است، اعتبارسنجی می‌کند.

PostForm: این فرم برای ایجاد پست‌های جدید استفاده می‌شود. عنوان و محتوای پست را جمع‌آوری می‌کند. داده‌ها را برای اطمینان از اینکه عنوان خالی نیست و محتوا خالی نیست، اعتبارسنجی می‌کند.

اعتبارسنجی فرم

هر کلاس فرم از اعتبارسنج‌های WTForms برای اعتبارسنجی ورودی کاربر استفاده می‌کند. این اعتبارسنج‌ها اطمینان حاصل می‌کنند که داده‌ها در قالب مورد نیاز قرار دارند و معیارهای خاصی را برآورده می‌کنند. به عنوان مثال، RegistrationForm

مستندات
تهیه کننده: پریا عبدالهی

از DataRequired() برای اطمینان از پر شدن همه فیلدها، Length() برای محدود کردن طول نام کاربری بین 4 تا 25 کاراکتر و EqualTo() برای مقایسه فیلدهای رمز عبور و تأیید رمز عبور استفاده می‌کند.

ارث‌گیری فرم

RegistrationForm و UpdateProfileForm از کلاس FlaskForm که قابلیت‌های اساسی فرم را فراهم می‌کند، ارث می‌برند. LoginForm از FlaskForm ارث می‌برد و یک فیلد remember را برای اجازه دادن به کاربران برای ماندن در حالت ورود به سیستم اضافه می‌کند. PostForm از FlaskForm ارث می‌برد و هیچ قابلیت اضافی اضافه نمی‌کند.

ادغام با Flask-Login

LoginForm و UpdateProfileForm از Flask-Login برای ادغام با سیستم احراز هویت کاربر برنامه استفاده می‌کنند. آنها از متغیر current_user برای دسترسی به کاربر فعلی استفاده می‌کنند و از آن برای انجام اعتبارسنجی‌های اضافی استفاده می‌کنند.

RegistrationForm

فیلد username باید بین 4 تا 25 کاراکتر طول داشته باشد.

فیلد email باید یک آدرس ایمیل معتبر باشد.

فیلدهای password و `confirm

فولدر templates:

درون فولدر templates چندین فایل html وجود دارد که به تفکیک به آنها میپردازیم.

فایل base.html :

این فایل، فایل پایه برای تمام صفحات وب‌سایت پروژه وبلاگ روانشناسی کودکان است. این فایل شامل کدهای HTML، CSS و JavaScript برای ایجاد طرح کلی و ویژگی‌های مشترک تمام صفحات است.

مستندات:

تگ‌های HTML:

<!DOCTYPE html>: این تگ، مشخص می‌کند که این فایل یک سند HTML است.

<html lang="fa">: این تگ، زبان سند HTML را مشخص می‌کند. در این مورد، زبان فارسی است.

مستندات
تهیه کننده: پریا عبدالهی

<head>: این تگ، سر سند HTML را مشخص می‌کند. سر سند، شامل اطلاعات مربوط به سند مانند عنوان، کدک‌های صفحه و لینک‌های به فایل‌های CSS و JavaScript است.

<title>روانشناسی کودکان</title>: این تگ، عنوان سند HTML را مشخص می‌کند.

link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" > rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNeOA7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">: این تگ، لینک به فایل CSS Bootstrap را اضافه می‌کند.

link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.15.4/css/all.css" > integrity="sha384-DyZ88mC6Up2uqS4h/h+kZ0l87a6j+vUU/I+636h72aXY9kX3yW7x4L7xu6/kZ80eXqU" crossorigin="anonymous">: این تگ، لینک به فایل CSS Font Awesome را اضافه می‌کند.

<link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">: این تگ، لینک به فایل CSS سفارشی پروژه را اضافه می‌کند.

<script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>: این تگ، لینک به فایل JavaScript jQuery را اضافه می‌کند.

تگ‌های CSS:

: این تگ، جهت نوشتن متن را به راست به چپ تغییر می‌دهد.

<h1 class="display-4">روانشناسی کودکان: جذاب و دل‌انگیز</h1>: این تگ، عنوان اصلی صفحه را با اندازه بزرگ نمایش می‌دهد.

سفری به دنیای شگفت‌انگیز روانشناسی کودکان : این تگ، یک پاراگراف با اندازه بزرگ و فونت برجسته نمایش می‌دهد.

تگ‌های HTML و CSS مشترک:

: این تگ، یک بخش بزرگ با پس‌زمینه سفید و متن تیره نمایش می‌دهد.

{% block content %}{% endblock content %}: این تگ، یک بلوک محتوای قابل تنظیم را مشخص می‌کند. محتوای واقعی صفحه در این بلوک قرار می‌گیرد.

توابع و متغیرهای مشترک:

{{ url_for('static', filename='styles.css') }}: این تابع، URL فایل CSS سفارشی پروژه را برمی‌گرداند.

فایل base.html فایل اصلی ما توی کدهای اچ تی ام ال محسوب میشود و باقی فایل ها از این فایل ارث بری میکنند(به وسیله کدهای جینجا2)

مستندات
تهیه کننده: پریا عبدالهی

نکته: درون فایل base کدهای فایل navbar.html include میشوند. یعنی هرچه در نوبار وجود دارد در فایل بیس ایجاد شده و فایل بیس هم به سایر فایل ها این ویژگی را به ارث میگذارد.

فایل home.html :
این فایل، صفحه اصلی پروژه وبلاگ روانشناسی کودکان است. این فایل از فایل base.html به عنوان پایه استفاده می‌کند و محتوای خاص صفحه اصلی را ارائه می‌دهد.
توسعه base.html:

دستور {٪ extends 'base.html' ٪} نشان می‌دهد که این فایل از فایل base.html به عنوان پایه استفاده می‌کند. این بدان معناست که محتوای فایل base.html در فایل صفحه اصلی نیز گنجانده می‌شود، همراه با محتوای خاص تعریف‌شده در این فایل.

پیام خوشامدگویی:

عنصر <h1 class="text-center mt-5 mb-4 welcome-title">{{ current_user.username }}</h1> یک پیام خوشامدگویی با نام کاربری کاربر فعلی را نمایش می‌دهد.

نمایش پست‌های وبلاگ:

عنصر یک ظرف برای نمایش پست‌های وبلاگ ایجاد می‌کند. حلقه {٪ for post in posts ٪} از طریق یک لیست پست‌ها تکرار می‌شود و برای هر پست یک کارت ایجاد می‌کند.

هر کارت دارای ساختار زیر است:

پیوندی به صفحه جزئیات پست با استفاده از برچسب .

عنوان پست با استفاده از برچسب <h5 class="card-title">{{ post.title }}</h5>.

خلاصه‌ای کوتاه از محتوای پست با استفاده از برچسب {{ post.brief_content }} .

کلاس mt-3 حاشیه‌ای به بالای هر کارت اضافه می‌کند، و کلاس‌های text-decoration-none و text-dark به ترتیب خط زیرنویس پیش‌فرض را حذف می‌کنند و رنگ متن پیوند را به خاکستری تیره تغییر می‌دهند.

بلوک محتوا:

بخش {% block content %}{% endblock content %} یک بلوک تعریف می‌کند که محتوای خاص صفحه اصلی در آن رندر می‌شود.

مستندات
تهیه کننده: پریا عبدالهی

فایل detail.html:

این فایل، قالب صفحه جزئیات پست پروژه وبلاگ روانشناسی کودکان است. این فایل از قالب base.html استفاده می‌کند و محتوای خاص صفحه جزئیات پست را ارائه می‌دهد.

توسعه base.html:

دستور {٪ extends 'base.html' ٪} نشان می‌دهد که این فایل از قالب base.html استفاده می‌کند. این بدان معناست که محتوای قالب base.html در قالب صفحه جزئیات پست نیز گنجانده می‌شود، همراه با محتوای خاص تعریف‌شده در این فایل.

عملیات پست:

اگر کاربر فعلی، نویسنده پست باشد، دو دکمه عملیاتی نمایش داده می‌شود: "حذف پست" و "ویرایش پست". این دکمه‌ها قابل دسترسی هستند با استفاده از برچسب‌های <{{ url_for('delete', post_id=post.id) }}"a href="> و < a > {{ url_for('update', post_id=post.id) }}"href="><، به ترتیب.

جزئیات پست به‌صورت ساختارمند نمایش داده می‌شوند:

عنوان پست: عنوان پست با استفاده از برچسب <h2 class="post-title mb-3">{{ post.title }}</h2> نمایش داده می‌شود.

متا پست: نویسنده و تاریخ پست با استفاده از برچسب <small class="post-meta"> نمایش داده می‌شوند. نویسنده با استفاده از برچسب {{ post.author.username }} به پروفایل خود پیوند داده می‌شود.

محتوای پست: محتوای پست با استفاده از برچسب نمایش داده می‌شود. محتوا پیش‌پردازش می‌شود تا کاراکترهای خط جدید را با شکست‌های HTML با استفاده از فیلتر | safe جایگزین کند.

اشتراک‌گذاری اجتماعی:

پست می‌تواند در پلتفرم‌های مختلف رسانه‌های اجتماعی با استفاده از دکمه‌ها به اشتراک گذاشته شود:

تلگرام: < a href="https://t.me/share/url?url={{ url_for('detail', post_id=post.id, _external=True) }}" target="_blank" class="btn btn-info"> تلگرام

توییتر: < a href="https://twitter.com/intent/tweet?url={{ url_for('detail', post_id=post.id, _external=True) }}&text={{ post.title }}" target="_blank" class="btn btn-info"> توییتر

فیس‌بوک: < a href="https://www.facebook.com/sharer/sharer.php?u={{ url_for('detail', post_id=post.id, _external=True) }}" target="_blank" class="btn btn-info"> فیس‌بوک

مستندات
تهیه کننده: پریا عبدالهی

لینکدین: < a href="https://www.linkedin.com/shareArticle?title={{ post.title }}" target="_blank" class="btn btn-info"> لینکدین

واتساپ: < a href="https://wa.me/?text={{ post.title }}" target="_blank" class="btn btn-info"> واتساپ

فایل create_post.html :

دستور {٪ extends 'base.html' ٪} نشان می‌دهد که این فایل از قالب base.html استفاده می‌کند. این بدان معناست که محتوای قالب base.html در قالب صفحه ایجاد پست نیز گنجانده می‌شود.

عنوان:

عنوان صفحه با استفاده از برچسب <h2 class="post-form-title">ایجاد پست جدید</h2> نمایش داده می‌شود.

فرم:

فرم ایجاد پست از طریق دستور <form action="" method="post"> تعریف می‌شود. این فرم دارای دو فیلد است:

عنوان: فیلد <input type="text" class="form-control" id="title" name="title" placeholder="عنوان پست خود را وارد کنید" value="{{ form.title.data }}"> برای وارد کردن عنوان پست استفاده می‌شود.

محتوا: فیلد <textarea class="form-control" id="content" name="content" rows="5">{{ form.content.data }}</textarea> برای وارد کردن محتوای پست استفاده می‌شود.

دکمه ارسال:

دکمه ارسال با استفاده از برچسب <button type="submit" class="btn btn-primary">ایجاد پست</button> برای ارسال فرم استفاده می‌شود.

نمونه‌ای از استفاده از فایل:

فرض کنید کاربری می‌خواهد پستی با عنوان "روانشناسی کودکان" و محتوای "روانشناسی کودکان علم مطالعه رفتار و ذهن کودکان است. این علم به بررسی رشد ذهنی، عاطفی، اجتماعی و جسمی کودکان می‌پردازد." ایجاد کند. برای این کار، کاربر باید به صفحه ایجاد پست مراجعه کرده و فیلدهای عنوان و محتوا را با اطلاعات مورد نظر خود پر کند. سپس، با کلیک بر دکمه ارسال، فرم را ارسال می‌کند.

اگر اطلاعات وارد شده توسط کاربر صحیح باشند، پست جدید با موفقیت ایجاد می‌شود و کاربر به صفحه اصلی وبلاگ منتقل می‌شود. در غیر این صورت، یک پیام خطا نمایش داده می‌شود.

مستندات
تهیه کننده: پریا عبدالهی

فایل login.html:

این فایل، قالب صفحه ورود به حساب کاربری پروژه وبلاگ روانشناسی کودکان است. این فایل از قالب base.html استفاده می‌کند و فرم ورود به حساب کاربری را ارائه می‌دهد.

عنوان صفحه با استفاده از برچسب <h2 class="login-title">ورود به حساب کاربری</h2> نمایش داده می‌شود.

اگر کاربر اطلاعات ورودی نادرستی وارد کند، خطاها در قالب inc/form_errors.html نمایش داده می‌شوند.

فرم ورود به حساب کاربری از طریق دستور <form action="" method="post"> تعریف می‌شود. این فرم دارای سه فیلد است:

ایمیل: فیلد < input type="email" class="form-control" id="email" name="email" placeholder="آدرس ایمیل خود را وارد کنید" required value="{{ form.email.data }}"> برای وارد کردن آدرس ایمیل کاربر استفاده می‌شود.

رمز عبور: فیلد < input type="password" class="form-control" id="password" name="password" placeholder="رمز عبور خود را وارد کنید" required> برای وارد کردن رمز عبور کاربر استفاده می‌شود.

مرا به خاطر بسپار: فیلد < div class="form-group form-check"> <label for="remember" class="form-check-label">{{ form.remember }} {{ form.remember.label }}</label> برای انتخاب گزینه «مرا به خاطر بسپار» استفاده می‌شود. اگر این گزینه انتخاب شود، رمز عبور کاربر در مرورگر ذخیره می‌شود تا در دفعات بعدی ورود به حساب کاربری نیازی به وارد کردن آن نباشد.

دکمه ارسال:

دکمه ارسال با استفاده از برچسب <button type="submit" class="btn btn-primary">ورود</button> برای ارسال فرم استفاده می‌شود.

نمونه‌ای از استفاده از فایل:

فرض کنید کاربری با نام کاربری "ali" و رمز عبور "123456" می‌خواهد وارد حساب کاربری خود شود. برای این کار، کاربر باید به صفحه ورود به حساب کاربری مراجعه کرده و فیلدهای ایمیل و رمز عبور را با اطلاعات مورد نظر خود پر کند. سپس، با کلیک بر دکمه ارسال، فرم را ارسال می‌کند. اگر اطلاعات وارد شده توسط کاربر صحیح باشند، کاربر با موفقیت وارد حساب کاربری خود می‌شود و به صفحه اصلی وبلاگ منتقل می‌شود. در غیر این صورت، یک پیام خطا نمایش داده می‌شود.

فایل profile.html :

این فایل از قالب base.html استفاده می‌کند و محتوای خاص صفحه پروفایل کاربر را ارائه می‌دهد.

اطلاعات پروفایل کاربر به‌صورت ساختارمند نمایش داده می‌شوند:

مستندات
تهیه کننده: پریا عبدالهی

عنوان پروفایل:

عنوان پروفایل شامل نام کاربری و آدرس ایمیل کاربر است.

* نام کاربری `<h2 class="profile-name">{{ current_user.username }}</h2>` :

* ایمیل `<p class="profile-email">{{ current_user.email }}</p>` :

- فرم بروزرسانی پروفایل :یک فرم برای بروزرسانی پروفایل کاربر ارائه می‌شود .این فرم دارای دو فیلد است:
  - نام کاربری <input type="text" class="form-control" id="username" name="username" value="{{ form.username.data }}" required> :
  - ایمیل <input type="email" class="form-control" id="email" name="email" value="{{ form.email.data }}" required> :
- خطاهای فرم :اگر کاربر هنگام بروزرسانی پروفایل خود اطلاعات نامعتبر وارد کند، خطاها در قالب inc/form_errors.htmlنمایش داده می‌شوند.
- دکمه بروزرسانی پروفایل :دکمه <button type="submit" class="btn btn-primary">بروزرسانی پروفایل </button>فرم بروزرسانی پروفایل را ارسال می‌کند.

نمونه‌ای از استفاده از فایل:

فرض کنید کاربری به نام "علی" می‌خواهد پروفایل خود را بروزرسانی کند .برای این کار، کاربر باید به صفحه پروفایل مراجعه کرده و فیلدهای نام کاربری و ایمیل را با اطلاعات مورد نظر خود پر کند .سپس، با کلیک بر دکمه "بروزرسانی پروفایل"، فرم را ارسال می‌کند.

اگر اطلاعات وارد شده توسط کاربر معتبر باشند، پروفایل کاربر با موفقیت بروزرسانی می‌شود .در غیر این صورت، پیام خطا نمایش داده می‌شود.

فایل register.html:

این فایل، قالب صفحه ثبت نام پروژه وبلاگ روانشناسی کودکان است. این فایل از قالب base.html استفاده می‌کند و فرم ثبت نام را ارائه می‌دهد.

عنوان:

عنوان صفحه با استفاده از برچسب <h2 class="register-title">ایجاد حساب کاربری</h2> نمایش داده می‌شود.

نمایش خطاها:

اگر کاربر اطلاعات ورودی نادرستی وارد کند، خطاها در قالب inc/form_errors.html نمایش داده می‌شوند.

مستندات
تهیه کننده: پریا عبدالهی

فرم ثبت نام:

فرم ثبت نام از طریق دستور <"form action="" method="post> تعریف می‌شود. این فرم دارای چهار فیلد است:

نام کاربری: فیلد < "input type="text" class="form-control" id="username" name="username" placeholder="نام کاربری خود را ایجاد کنید" required "{{ form.username.data }}"=value> برای وارد کردن نام کاربری کاربر استفاده می‌شود.

ایمیل: فیلد < "input type="email" class="form-control" id="email" name="email" placeholder="آدرس ایمیل خود را وارد کنید" required "{{ form.email.data }}"=value> برای وارد کردن آدرس ایمیل کاربر استفاده می‌شود.

رمز عبور: فیلد < "input type="password" class="form-control" id="password" name="password" placeholder="یک رمز عبور قوی وارد کنید" required> برای وارد کردن رمز عبور کاربر استفاده می‌شود.

تأیید رمز عبور: فیلد < "input type="password" class="form-control" id="confirm_password" name="confirm_password" placeholder="رمز عبور خود را دوباره وارد کنید" required> برای تأیید رمز عبور کاربر استفاده می‌شود.

دکمه ارسال:

دکمه ارسال با استفاده از برچسب <"button type="submit" class="btn btn-primary">ایجاد حساب کاربری</button> برای ارسال فرم استفاده می‌شود.

نمونه‌ای از استفاده از فایل:

فرض کنید کاربری می‌خواهد حساب کاربری ایجاد کند. برای این کار، کاربر باید به صفحه ثبت نام مراجعه کرده و فیلدهای نام کاربری، ایمیل، رمز عبور و تأیید رمز عبور را با اطلاعات مورد نظر خود پر کند. سپس، با کلیک بر دکمه "ایجاد حساب کاربری"، فرم را ارسال می‌کند. اگر اطلاعات وارد شده توسط کاربر صحیح باشند، حساب کاربری جدید با موفقیت ایجاد می‌شود و کاربر به صفحه اصلی وبلاگ منتقل می‌شود. در غیر این صورت، یک پیام خطا نمایش داده می‌شود.

فایل update.html :
این فایل، قالب صفحه ویرایش پست پروژه وبلاگ روانشناسی کودکان است. این فایل از قالب base.html استفاده می‌کند و فرم ویرایش پست را ارائه می‌دهد.

عنوان:

عنوان صفحه با استفاده از برچسب <"h2 class="post-update-title">ویرایش پست</h2> نمایش داده می‌شود.

فرم ویرایش پست:

فرم ویرایش پست از طریق دستور <"form action="" method="post> تعریف می‌شود. این فرم دارای دو فیلد است:

مستندات
تهیه کننده: پریا عبدالهی

عنوان: فیلد <input type="text" class="form-control" id="title" name="title" placeholder="عنوان جدید پست خود را وارد کنید" required value="{{ form.title.data }}"> برای وارد کردن عنوان جدید پست استفاده می‌شود.

محتوا: فیلد <textarea class="form-control" id="content" name="content" rows="5" placeholder="محتوای جدید پست خود را وارد کنید" required value="{{ form.content.data }}"></textarea> برای وارد کردن محتوای جدید پست استفاده می‌شود.

دکمه ارسال:

دکمه ارسال با استفاده از برچسب <button type="submit" class="btn btn-primary">ویرایش پست</button> برای ارسال فرم استفاده می‌شود.

نمونه‌ای از استفاده از فایل:

فرض کنید کاربری می‌خواهد پستی با عنوان "روانشناسی کودکان" و محتوای "روانشناسی کودکان علم مطالعه رفتار و ذهن کودکان است. این علم به بررسی رشد ذهنی، عاطفی، اجتماعی و جسمی کودکان می‌پردازد." را ویرایش کند. برای این کار، کاربر باید به صفحه ویرایش پست مراجعه کرده و فیلدهای عنوان و محتوا را با اطلاعات مورد نظر خود پر کند. سپس، با کلیک بر دکمه "ویرایش پست"، فرم را ارسال می‌کند.

اگر اطلاعات وارد شده توسط کاربر صحیح باشند، پست با موفقیت ویرایش می‌شود و کاربر به صفحه اصلی وبلاگ منتقل می‌شود. در غیر این صورت، یک پیام خطا نمایش داده می‌شود.

تفاوت‌های این فایل با فایل create_post.html:

تفاوت اصلی این فایل با فایل create_post.html در این است که فرم ویرایش پست، فیلدهای عنوان و محتوا را پر می‌کند که قبلاً در پایگاه داده ذخیره شده‌اند. این فیلدها با استفاده از دستور {{ form.title.data }} و {{ form.content.data }} پر می‌شوند.

درون فولدر templates یک فولدر داریم به نام inc که مخفف include است.

به تفکیک فایل‌های html درون ان را بررسی میکنیم.

فایل navbar.html :

این فایل نوار ناوبری را برای پروژه وبلاگ روانشناسی کودکان تعریف می‌کند. این یک نوار ناوبری ساده با پیوندهایی به صفحه اصلی، پروفایل، ثبت نام، ورود و ورود با گوگل است.

مستندات
تهیه کننده: پریا عبدالهی

سبک‌بندی:

نوار ناوبری با استفاده از کلاس‌های Bootstrap سبک‌بندی می‌شود. دارای رنگ زمینه روشن، رنگ متن روشن و حاشیه گرد است.

برندینگ:

نوار ناوبری دارای یک بخش برندینگ با تصویر لوگوی سایت و نام سایت است.

پیوندهای ناوبری:

نوار ناوبری دارای یک لیست پیوندهایی به صفحه اصلی، پروفایل، ثبت نام، ورود و ورود با گوگل است. این پیوندها به عنوان navbar-links سبک‌بندی می‌شوند و دارای آیکون‌هایی برای شناسایی آسان هستند.

تصدیق هویت کاربر:

نوار ناوبری وضعیت احراز هویت کاربر فعلی را بررسی می‌کند و پیوندهای مختلفی را برای کاربران وارد شده و خارج شده نشان می‌دهد. اگر کاربر وارد شده باشد، پیوندهای خروج، پروفایل و صفحه اصلی نشان داده می‌شوند. اگر کاربر وارد نشده باشد، پیوندهای ثبت نام، ورود و ورود با گوگل نشان داده می‌شوند.

نمونه‌ای از استفاده از فایل:

فرض کنید کاربری به صفحه اصلی وبلاگ روانشناسی کودکان مراجعه می‌کند. در این صورت، پیوندهای «خانه»، «ثبت نام»، «ورود» و «ورود با گوگل» در نوار ناوبری نشان داده می‌شوند. اگر کاربر قبلاً وارد شده باشد، پیوندهای «خروج» و «پروفایل» نیز نشان داده می‌شوند.

توضیحات بیشتر:

پیوند «خانه»: این پیوند کاربر را به صفحه اصلی وبلاگ روانشناسی کودکان منتقل می‌کند.

پیوند «پروفایل»: این پیوند کاربر را به صفحه پروفایل خود منتقل می‌کند.

پیوند «ثبت نام»: این پیوند کاربر را به صفحه ثبت نام منتقل می‌کند.

پیوند «ورود»: این پیوند کاربر را به صفحه ورود منتقل می‌کند.

پیوند «ورود با گوگل»: این پیوند کاربر را به صفحه ورود با گوگل منتقل می‌کند.

فایل form_errors.html:

این فایل برای نمایش پیام‌های خطا برای فرم‌های پروژه وبلاگ روانشناسی کودکان استفاده می‌شود. این فایل خطاهای موجود در فیلدهای نام کاربری، ایمیل، رمز عبور و تأیید رمز عبور را بررسی می‌کند و آنها را در یک جعبه هشدار با پیام خطا نمایش

می‌دهد.
نمایش خطا:

فایل بررسی می‌کند که آیا برای هر یک از چهار فیلد: نام کاربری، ایمیل، رمز عبور و تأیید رمز عبور، خطایی وجود دارد یا خیر. اگر خطا وجود داشته باشد، یک جعبه هشدار با کلاس خطر، عنوان پیام خطا و فهرستی از پیام‌های خطا برای هر فیلد نمایش می‌دهد.

پردازش پیام‌های خطا:

فایل از برچسب‌های {٪ if ٪} و {٪ for ٪} برای نمایش پیام‌های خطا به‌صورت شرطی استفاده می‌کند. برچسب {٪ if ٪} بررسی می‌کند که آیا فیلد فرم دارای خطایی است و برچسب {٪ for ٪} از طریق هر خطا تکرار می‌شود و آن را به‌عنوان یک پاراگراف جداگانه نمایش می‌دهد.

فرمت پیام خطا:

پیام‌های خطا به‌صورت واضح و مختصر نمایش داده می‌شوند، با استفاده از عناصر HTML ساده مانند برچسب‌های <h4> و <p>. این کار باعث می‌شود که کاربران بتوانند خطاها را درک کنند و اقدامات اصلاحی را انجام دهند.

کلاس خطا:

کلاس alert-container برای سبک‌دهی جعبه هشدار با پس‌زمینه قرمز و عنوان پررنگ استفاده می‌شود، که آن را از سایر محتوای صفحه متمایز می‌کند.

فایل messages.html:

این فایل برای نمایش پیام‌های فلاش در پروژه وبلاگ روانشناسی کودکان استفاده می‌شود. پیام‌های فلاش پیام‌های موقتی هستند که پس از انجام یک عمل خاص مانند ورود، ثبت‌نام، ایجاد پست یا به‌روزرسانی پروفایل، برای کاربر نمایش داده می‌شوند.

بازیابی پیام‌های فلاش:

این فایل از تابع get_flashed_messages() برای بازیابی همه پیام‌های فلاش از جلسه استفاده می‌کند. پارامتر with_categories=true اطمینان حاصل می‌کند که پیام‌ها همراه با دسته‌های مربوطه خود بازگردانده می‌شوند.

نمایش شرطی:

بلوک {٪ if messages ٪} اطمینان حاصل می‌کند که بخش پیام‌های فلاش فقط در صورتی نمایش داده می‌شود که پیامی برای نمایش وجود داشته باشد.

گروه‌بندی پیام‌ها: حلقه {٪ for cat, msg in messages ٪} از طریق لیست پیام‌های فلاش تکرار می‌شود و آنها را بر اساس دسته گروه‌بندی می‌کند.

مستندات
تهیه کننده: پریا عبدالهی

سبک‌دهی جعبه هشدار:

هر پیام فلاش در یک جعبه هشدار با کلاسی که با دسته آن مطابقت دارد نمایش داده می‌شود. جعبه‌های هشدار دارای ویژگی قابل حذف هستند که به کاربران امکان می‌دهد با کلیک بر روی دکمه «بستن» آنها را ببندند.

نمایش پیام:

متن پیام فلاش در داخل جعبه هشدار نمایش داده می‌شود.

نمونه استفاده:

فرض کنید کاربری به‌طور موفقیت‌آمیز وارد وبلاگ می‌شود. یک پیام فلاش برای اطلاع کاربر از ورود موفقیت‌آمیز نمایش داده می‌شود. فایل messages.html یک جعبه هشدار با پس‌زمینه سبز و پیام موفقیت نمایش می‌دهد.

HTML

```
<"div class="alert alert-success alert-dismissible fade show" role="alert>
```

موفقیت در ورود!

```
button type="button" class="btn-close" data-bs-dismiss="alert" aria->
<label="Close"></button
```

```
<div/>
```

Use code with caution. Learn more

به‌طور مشابه، پیام‌های فلاش می‌توانند برای سایر اقدامات مانند ثبت‌نام برای حساب، ایجاد پست یا به‌روزرسانی پروفایل نمایش داده شوند.

درون فولدر blog یک فولدر دیگری داریم به نام static که درون آن فولدری وجود دارد به نام images که تصاویر را ذخیره میکند و در کنار این فولدر دوتا فایل styles.css و scripts.js را داریم که به ترتیب کدهای سی اس اس و جاوااسکریپت را درون خود ذخیره میکنند.

فایل scripts.js:

این فایل برای افزودن قابلیت‌های JavaScript به پروژه وبلاگ روانشناسی کودکان استفاده می‌شود. این فایل شامل یک تابع است که روی کلاس .post-container اعمال می‌شود. این تابع دو رویداد hover را ثبت می‌کند:

رویداد mouseenter وقتی موس روی متن قرار می‌گیرد اجرا می‌شود.

رویداد mouseleave وقتی موس از روی متن حرکت می‌کند اجرا می‌شود.

مستندات
تهیه کننده: پریا عبدالهی

توابع:

\(document\)\.ready\(function\(\) \{\}\): این تابع زمانی اجرا می‌شود که DOM بارگذاری شود. ❋ `$(".post-container").hover(function() {}`: این تابع رویدادmouseenterرا برای کلاس.post-container` ثبت می‌کند.

function() {}: این تابع محتوای رویداد mouseenter را تعیین می‌کند.

\("\.post\-container"\)\.css\("background", "rgba\(255, 255, 255, 0\.8\)"\): این دستور زمینه متن را به رنگ سفید با شفافیت 80 درصد تغییر می‌دهد. ❋ `$(".post-container").hover(function() {}`: این تابع رویدادmouseleaveرا برای کلاس.post-container` ثبت می‌کند.

function() {}: این تابع محتوای رویداد mouseleave را تعیین می‌کند.

$(".post-container").css("background", "none");: این دستور زمینه متن را به حالت اولیه خود بازمی‌گرداند.

نمونه استفاده:

فرض کنید یک پست وبلاگ در صفحه detail.html نمایش داده می‌شود. این پست دارای یک کلاس .post-container است. اگر این تابع روی این کلاس اعمال شود، وقتی کاربر موس خود را روی متن پست قرار می‌دهد، زمینه متن به رنگ سفید با شفافیت 80 درصد تغییر می‌کند. وقتی کاربر موس خود را از روی متن حرکت می‌دهد، زمینه متن به حالت اولیه خود بازمی‌گردد.

فایل styles.css:
این فایل حاوی سبک‌های CSS برای پروژه وبلاگ روانشناسی کودکان است. این فایل ظاهر و احساس کلی وب‌سایت را سبک می‌کند، از جمله هدر، بدنه، کارت‌ها، قاب‌های پست و دکمه‌های اشتراک‌گذاری اجتماعی.

عناصر سبک‌دهی:

تصویر پس‌زمینه: عنصر بدنه دارای تصویر پس‌زمینه‌ای است که به تصویر /static/images/lo.jpg تنظیم شده است، که تضمین می‌کند که تصویر پس‌زمینه صفحه کامل باشد و تکرار نشود.

کارت‌ها:

.card-columns: طرح شبکه را برای کارت‌ها در صفحه اصلی با یک ستون واحد و فاصله rem1 بین ستون‌ها تنظیم می‌کند.

.card: سبک را برای کارت‌های فردی تعریف می‌کند، از جمله عدم وجود مرز، گوشه‌های گرد، رنگ پس‌زمینه و جلوه‌های سایه جعبه.

.card:hover: سایه واضح‌تری اضافه می‌کند و رنگ پس‌زمینه را هنگام حرکت ماوس روی کارت به سفید تغییر می‌دهد.

مستندات
تهیه کننده: پریا عبدالهی

.card-title: اندازه قلم و حاشیه را برای عنوان کارت تنظیم می‌کند. .card-text: اندازه قلم و رنگ را برای متن کارت تنظیم می‌کند.

هدر:

.welcome-title: هدر اصلی صفحه اصلی را با یک خانواده قلم خاص، رنگ، سایه متن و جلوه‌های گذار سبک می‌دهد.

قاب‌های پست:

.post-container: سبک را برای یک قاب پست واحد در صفحه جزئیات، از جمله رنگ پس‌زمینه، مرز، شعاع گوشه، حاشیه، و جلوه‌های سایه جعبه تعریف می‌کند.

.post-container:hover: هنگام حرکت ماوس روی قاب پست، جلوه سایه جعبه را قوی‌تر می‌کند.

.post-action: حاشیه را برای دکمه‌های عمل پست تنظیم می‌کند.

.post-header: هدر پست (عنوان و متا) را به سمت راست هم‌تراز می‌کند.

.post-title: اندازه قلم و رنگ را برای عنوان پست تنظیم می‌کند.

.post-meta: اندازه قلم و رنگ را برای اطلاعات متا پست تنظیم می‌کند.

.post-content: خط فاصله و رنگ را برای محتوای پست تنظیم می‌کند.

اشتراک‌گذاری اجتماعی:

.post-share: حاشیه بالا را برای دکمه‌های اشتراک‌گذاری اجتماعی تنظیم می‌کند.

.social-share a: دکمه‌های اشتراک‌گذاری اجتماعی را با رنگ سفید، تزئین متن، پدینگ، شعاع گوشه، حاشیه، اندازه قلم و جلوه گذار سبک می‌دهد.

.social-share a:hover: هنگام حرکت ماوس روی دکمه اشتراک‌گذاری اجتماعی، رنگ پس‌زمینه را به #333 تغییر می‌دهد.

نمونه استفاده:

فرض کنید یک کارت در صفحه اصلی وبلاگ نمایش داده می‌شود. این کارت دارای کلاس .card است. اگر این کلاس روی آن اعمال شود، کارت دارای ویژگی‌های زیر خواهد بود:

هیچ مرزی نخواهد داشت.

گوشه‌های آن گرد خواهد بود.

رنگ پس‌زمینه آن سفید با شفافیت 60.5 درصد خواهد بود.

وقتی کاربر ماوس خود را روی آن قرار می‌دهد، سایه‌ای با شعاع 8 پیکسل و شفافیت 30 درصد به آن اضافه می‌شود.

مستندات
تهیه کننده: پریا عبدالهی

وقتی کاربر ماوس خود را از روی آن حرکت می‌دهد، سایه حذف می‌شود.

عنوان آن با اندازه قلم rem1.1 و حاشیه rem0.3 نمایش داده می‌شود.

متن آن با اندازه قلم rem0.9 و رنگ #666 نمایش داده می‌شود.

فایل blog.db:

فایل blog.db یک پایگاه داده SQLite است که برای پروژه وبلاگ روانشناسی کودکان استفاده می‌شود. این پایگاه داده اطلاعات مربوط به کاربران، پست‌ها و نظرات را ذخیره می‌کند.

جدول کاربران اطلاعات مربوط به کاربران را ذخیره می‌کند، از جمله نام کاربری، ایمیل، رمز عبور و نقش.

جدول پست‌ها اطلاعات مربوط به پست‌ها را ذخیره می‌کند، از جمله عنوان، متن، تاریخ ایجاد و نویسنده.

جدول نظرات اطلاعات مربوط به نظرات را ذخیره می‌کند، از جمله نویسنده، متن و تاریخ ایجاد.

نمونه داده‌ها برای هر جدول ارائه شده است.

تغییرات احتمالی که می‌توان در آینده به این پایگاه داده اضافه کرد شامل موارد زیر است:

اطلاعات مربوط به دسته‌های پست

اطلاعات مربوط به برچسب‌های پست

اطلاعات مربوط به نظرات تأیید شده

همچنین می‌توان از این پایگاه داده برای ذخیره اطلاعات مربوط به سایر ویژگی‌های وب‌سایت، مانند:

اطلاعات مربوط به سفارشات

اطلاعات مربوط به پرداخت‌ها

اطلاعات مربوط به کاربران عضو

در اینجا یک خلاصه کلی از نحوه استفاده از این پایگاه داده آورده شده است:

برای ثبت نام کاربران جدید، اطلاعات کاربر را در جدول users وارد کنید.

برای ایجاد یک پست جدید، اطلاعات پست را در جدول posts وارد کنید.

برای ارسال یک نظر، اطلاعات نظر را در جدول comments وارد کنید.

مستندات
تهیه کننده: پریا عبدالهی

فایل requeriments.txt:

فایل الزامات

جدول زیر نسخه هر بسته‌ای را که برای پروژه ضروری است نشان می‌دهد:

نام بسته | توضیحات

absl-py | کتابخانه‌ای برای دسترسی به ابزارهای کاربردی Google

alembic | ابزاری برای مدیریت پایگاه داده‌های SQL

attrs | کتابخانه‌ای برای کار با ویژگی‌های داده‌ها

Authlib | کتابخانه‌ای برای احراز هویت و مجوز

bcrypt | کتابخانه‌ای برای رمزگذاری گذرواژه‌ها

cachelib | کتابخانه‌ای برای مدیریت حافظه نهان

cachetools | کتابخانه‌ای برای بهبود کارایی حافظه نهان

certifi | کتابخانه‌ای برای مدیریت گواهینامه‌های SSL

cffi | کتابخانه‌ای برای رابطهای سطح پایین C/C++ در Python

charset-normalizer | کتابخانه‌ای برای تبدیل کدگذاری‌های کاراکتر

click | کتابخانه‌ای برای ایجاد برنامه‌های خط فرمان

colorama | کتابخانه‌ای برای پشتیبانی از رنگ‌ها در خروجی خط فرمان

cryptography | کتابخانه‌ای برای امنیت رمزنگاری

cycler | کتابخانه‌ای برای تکرار الگوها

decorator | کتابخانه‌ای برای تزئین توابع و کلاس‌ها

dnspython | کتابخانه‌ای برای کار با سیستم نام دامنه (DNS)

email-validator | کتابخانه‌ای برای اعتبارسنجی آدرس‌های ایمیل

مستندات
تهیه کننده: پریا عبدالهی

Flask | چارچوب وب Python

Flask-Bcrypt | افزونه‌ای برای Flask برای رمزگذاری گذرواژه‌ها

Flask-JWT-Extended | افزونه‌ای برای Flask برای مدیریت توکن‌های JWT

Flask-Login | افزونه‌ای برای Flask برای مدیریت ورود به سیستم

Flask-Migrate | افزونه‌ای برای Flask برای مدیریت مهاجرت‌های پایگاه داده

Flask-OAuthlib | افزونه‌ای برای Flask برای احراز هویت OAuth

Flask-Reuploaded | افزونه‌ای برای Flask برای بارگذاری مجدد فایل‌ها

Flask-SQLAlchemy | افزونه‌ای برای Flask برای کار با پایگاه داده‌های SQL

Flask-Uploads | افزونه‌ای برای Flask برای بارگذاری فایل‌ها

Flask-WTF | افزونه‌ای برای Flask برای فرم‌های HTML

flatbuffers | کتابخانه‌ای برای ذخیره و بازیابی داده‌ها در ساختارهای داده‌ای ثابت

fonttools | کتابخانه‌ای برای کار با فونت‌ها

google-auth | کتابخانه‌ای برای احراز هویت با خدمات Google

google-auth-oauthlib | کتابخانه‌ای برای احراز هویت OAuth با خدمات Google

greenlet | کتابخانه‌ای برای اجرای کد به صورت موازی

httplib2 | کتابخانه‌ای برای کار با پروتکل HTTP

idna | کتابخانه‌ای برای کار با نام‌های دامنه بین‌المللی

importlib-metadata | کتابخانه‌ای برای دسترسی به اطلاعات مربوط به بسته‌ها

importlib-resources | کتابخانه‌ای برای دسترسی به منابع بسته‌ها

is-disposable-email | کتابخانه‌ای برای تشخیص آدرس‌های ایمیل یکبار مصرف

itsdangerous | کتابخانه‌ای برای تولید و بررسی توکن‌های امنیتی

Jinja2 | موتور قالب‌بندی مورد استفاده توسط Flask

مستندات
تهیه کننده: پریا عبدالهی

kiwisolver | کتابخانه‌ای برای حل سیستم‌های معادلات خطی

Mako | موتور قالب‌بندی دیگری که می‌توان با Flask استفاده کرد

MarkupSafe | کتابخانه‌ای برای کار با کد HTML در متن

matplotlib | کتابخانه‌ای برای رسم نمودارها

mediapipe | کتابخانه‌ای برای پردازش تصویر و ویدیو

MouseInfo | کتابخانه‌ای برای دسترسی به اطلاعات مربوط به ماوس

numpy | کتابخانه‌ای برای محاسبات عددی

oauth2client | کتابخانه‌ای برای احراز هویت OAuth با سرویس‌های Google

oauthlib | کتابخانه‌ای عمومی برای احراز هویت OAuth

opencv-contrib-python | کتابخانه‌ای برای پردازش تصویر و ویدیو

opencv-python | کتابخانه‌ای برای پردازش تصویر و ویدیو

opencv-python-headless | نسخه بدون رابط کاربری کتابخانه opencv-python

packaging | کتابخانه‌ای برای کار با بسته‌های Python

Pillow | کتابخانه‌ای برای کار با تصاویر

pip | ابزاری برای نصب و مدیریت بسته‌های Python

protobuf | کتابخانه‌ای برای کار با پروتکل پیامی protobuf

pyasn1 | کتابخانه‌ای برای کار با ASN.1

pyasn1-modules | کتابخانه‌ای برای کار با ASN.1 در Python

PyAutoGUI | کتابخانه‌ای برای کنترل ماوس و صفحه کلید از طریق Python

pycparser | کتابخانه‌ای برای تجزیه کد C

PyGetWindow | کتابخانه‌ای برای دسترسی به اطلاعات مربوط به پنجره‌های ویندوز

کتابخانه‌ای برای تولید | PyJWT

مستندات
تهیه کننده: پریا عبدالهی

در پایان باید گفت که پروژه وبلاگ روانشناسی کودکان، یک پروژه ارزشمند و قابل توسعه است. با اضافه کردن ویژگی‌های جدید به این پروژه، می‌توان آن را کاربردی‌تر و جذاب‌تر کرد.