

مستندات پروژه درس برنامه سازی پیشرفته

استاد : دکتر داود آبادی

اعضای گروه:

سید علیرضا حسینی

امیرمهدی ملکوتی خواه

محمد سیدی

معرفی پروژه

پلتفرم توصیه‌گر محصولات مراقبت پوستی یک وب‌اپلیکیشن مبتنی بر Django است که با هدف ارائه‌ی تجربه‌ی خرید شخصی‌سازی‌شده برای کاربران طراحی شده. کاربران بعد از ثبت‌نام و ورود، یک کوئیز اولیه درباره وضعیت پوست خود تکمیل می‌کنند. پاسخ‌های این کوئیز به عنوان SkinProfile در سیستم ذخیره می‌شود. سپس سیستم با توجه به این اطلاعات، محصولات مناسب برای نوع پوست و دغدغه‌های پوستی کاربر را پیشنهاد می‌دهد.

۱. ویژگی‌های کلیدی این پروژه

- a. ثبت‌نام و ورود با فعال‌سازی ایمیل: کاربران تنها پس از تأیید ایمیل می‌توانند وارد شوند.
- b. پروفایل کاربری: هر کاربر می‌تواند اطلاعات شخصی خود (مثل شماره تماس، آدرس و عکس پروفایل) را مدیریت کند.
- c. کوئیز پوستی (Skin Quiz): اطلاعات پایه‌ی پوستی کاربر (خشکی، چربی، حساسیت، آکنه و ...) جمع‌آوری می‌شود.
- d. سیستم توصیه‌گر: محصولات بر اساس SkinProfile و رفتار کاربران مشابه پیشنهاد می‌شوند.
- e. مدیریت سبد خرید: کاربران می‌توانند محصولات پیشنهادی را به سبد خرید اضافه کنند و خرید خود را مدیریت کنند.
- f. برنامه‌های مراقبت پوستی (Routines): کاربران بر اساس پروفایل خود برنامه‌های مختلفی مثل روتین کامل، روتین آبرسانی و روتین مینیمال دریافت می‌کنند.

۲. هدف نهایی

ترکیب یک فروشگاه آنلاین محصولات پوستی با یک سیستم توصیه‌گر هوشمند برای ارائه تجربه‌ای منحصربه‌فرد و متناسب با نیاز هر کاربر.

۳. ساختار اپلیکیشن

این پروژه بر پایه‌ی معماری چنداپلیکیشنی Django ساخته شده است. هر اپلیکیشن مسئول یک بخش مشخص از سیستم است و این ماژولار بودن باعث توسعه‌پذیری و نگهداری آسان‌تر می‌شود.

a. mainpage

نقش: صفحه اصلی و مدیریت نمایش محصولات.

امکانات:

- i. جستجو (ساده و پیشرفته) با الگوریتم شخصی‌سازی شده.
- ii. فیلتر و مرتب‌سازی محصولات (بر اساس برند، نوع پوست، قیمت و ...).
- iii. نمایش جزئیات محصول + بخش نظرات کاربران.
- iv. ثبت رفتار کاربر (بازدید، لایک).
- v. API عمومی برای نمایش محصولات (با DRF).
- vi. تابع‌های کلیدی:

۱. تابع `filter(request, products)`

a. وظیفه: اعمال فیلتر روی لیست محصولات.

b. ورودی: `request` و `queryset` محصولات.

c. فیلترها:

i. `brand` → برند محصول

ii. `category` → دسته‌بندی

iii. `skin_type` → نوع پوست

iv. `concern` → دغدغه پوستی

v. `min_price / max_price` → محدوده قیمت

d. خروجی: لیست محصولات فیلترشده

e. اهمیت: کاربر خیلی سریع می‌تواند محصول موردنظر خودش را پیدا کند.

۲. تابع `sort(request, products)`

a. وظیفه: مرتب‌سازی لیست محصولات.

b. گزینه‌های مرتب‌سازی:

i. price_low → ارزان‌ترین

ii. price_high → گران‌ترین

iii. rating → پرامتیازترین

iv. popularity → پربازدیدترین

c. خروجی: لیست محصولات مرتب‌شده

d. اهمیت: کنترل تجربه کاربری در نمایش محصولات.

۳. تابع bayesian_average(product, total_rating_average)

a. وظیفه: محاسبه یک امتیاز واقعی‌تر برای محصول بر اساس

امتیاز کاربران + تعداد فروش.

b. این فرمول ترکیبی از میانگین امتیاز کل و امتیاز خود محصول

است.

c. باعث می‌شود محصولاتی که تعداد رأی کمی دارند به صورت

غیرواقعی بالا نروند.

d. اهمیت: رتبه‌بندی منصفانه‌تر محصولات.

۴. تابع search(request, products, ...)

a. این مهم‌ترین تابع پروژه است و نقش موتور جستجوی هوشمند

را دارد. search قلب سیستم توصیه و جستجوی هوشمند است.

این تابع داده‌های کاربر (SkinProfile) و تاریخچه کاربران مشابه را

ترکیب می‌کند تا نتایج جستجو و توصیه شخصی‌سازی شوند.

i. مراحل کار:

۱. دریافت Query

گرفتن عبارت جستجو (q) از کاربر

پاک‌سازی (حذف نویسه‌های اضافی، نیم‌فاصله،

کاراکترهای غیرمجاز)

۲. ذخیره تاریخچه

اگر کاربر وارد شده باشد، عبارت جستجو در مدل SearchHistory ذخیره می شود.

فقط ۳۰ جستجوی آخر نگه داشته می شود.

۳. استفاده از Cache

نتایج جستجو با کلید مخصوص ذخیره می شوند تا در جستجوهای بعدی سریع تر بازیابی شوند.

۴. ایجاد Query در دیتابیس

جستجو در name, brand, description
اگر نتایج کم باشند از کل محصولات استفاده می شود.

۵. محاسبه امتیاز محصولات (Scoring)

a. معیارها:

i. شباهت نام

(NAME_BASE_SCORE)

ii. شباهت برند

(BRAND_BASE_SCORE)

iii. ترکیبات محصول (ingredients)

iv. دغدغه های پوستی (concerns)

v. انواع پوست (skin_types)

vi. امتیاز و فروش

(bayesian_average)

b. برای کاربرانی که کوئیز پر کرده اند:

i. مقایسه SkinProfile کاربر با

محصولات

- ii. اضافه کردن امتیاز برای محصولاتى که کاربران مشابه خرید کرده‌اند (Collaborative Filtering ساده)
- iii. مرتب‌سازی و خروجی
- iv. محصولات براساس امتیاز نهایی مرتب می‌شوند
- v. نتیجه نهایی برگردانده می‌شود
- ۶. دلیل خاص بودن تابع:

a. ترکیب چند رویکرد مختلف برای جستجو:

- i. Text Matching (مقایسه متن نام و برند)
- ii. Semantic Matching (مقایسه دغدغه‌ها و نوع پوست)
- iii. Behavioral Matching (استفاده از خریدهای کاربران مشابه)

۵. توابع دیگر

- a. more_products: نمایش محصولات بیشتر براساس یک Query خاص
- b. live_search: پیشنهاد سریع (Autocomplete) هنگام تایپ
- c. product_detail: نمایش جزئیات محصول، ثبت نظر و امتیاز
- d. like_product: لایک یا آن لایک کردن محصول

b. Accounts

- نقش: مدیریت کاربران و احراز هویت.
- i. ثبت نام کاربر با ارسال لینک فعال سازی ایمیل.

ii. ورود و خروج کاربر.

iii. ذخیره و مدیریت تاریخچه جستجو.

iv. ذخیره تعامل کاربر با محصولات (بازدید، لایک، خرید).

v. ثبت تاریخچه خرید کاربر برای توصیه‌های آینده.

c. Profiles

نقش: مدیریت پروفایل شخصی و سبد خرید.

i. مشاهده و ویرایش اطلاعات پروفایل (شماره تماس، آدرس، عکس پروفایل).

ii. مدیریت سبد خرید (اضافه، کم کردن، حذف محصول).

iii. خرید محصولات موجود در سبد.

iv. نمایش و مدیریت روتین‌های پوستی کاربر (کامل، مینیمال، آبرسانی).

d. Quiz

نقش: جمع‌آوری داده‌های اولیه از کاربر درباره وضعیت پوست.

i. نمایش سوالات کوئیز پوستی.

ii. ذخیره نتایج در مدل SkinProfile.

iii. استفاده از اطلاعات SkinProfile در سیستم توصیه‌گر و تولید روتین‌های

پوستی.

e. Routine

نقش: تولید برنامه‌های پوستی بر اساس اطلاعات کاربر.

i. تعریف چند نوع روتین (کامل، مینیمال، آبرسانی).

ii. استفاده از اطلاعات SkinProfile برای انتخاب محصولات مناسب هر مرحله.

iii. نمایش مراحل روتین همراه با محصولات مرتبط.

۴. اجزای دیگر پروژه

a. /templates → قالب‌های HTML برای هراپ.

b. /static → فایل‌های CSS، JavaScript و تصاویر.

c. /media → فایل‌های آپلودی کاربران (مثل عکس پروفایل و تصاویر محصول).

db.sqlite3 → دیتابیس پیش فرض SQLite (قابل جایگزینی با
MySQL/PostgreSQL).

مدل های داده

مدل های داده در این پروژه به سه دسته کلی تقسیم می شوند:

۱. کاربران و پروفایل ها

a. User (Django Auth)

مدل پیش فرض Django برای مدیریت کاربران (username, password, email)

و (...).

b. UserProfile (در اپ profiles)

نگهداری اطلاعات تکمیلی کاربر.

فیلدها

۱. user (OneToOne با User)

۲. phone_number

۳. address

۴. profile_picture

هدف: نگهداری داده های شخصی (در کنار SkinProfile که داده های پوستی کاربر را

ذخیره می کند).

c. SkinProfile (در اپ quiz)

ذخیره اطلاعات پوستی کاربر بر اساس کوئیز.

فیلدها:

i. skin_type → خشک، چرب، حساس، نرمال ...

ii. concerns → آکنه، لک، خشکی ...

iii. quiz_completed → آیا کوئیز تکمیل شده یا خیر.

iv. quiz_skipped → آیا کاربر کوئیز را رد کرده است.

هدف: استفاده در سیستم توصیه گر و تولید روتین های پوستی.

۲. محصولات و تعاملات کاربران

a. Product (در اپ mainpage)

نماینده محصولات موجود در فروشگاه.

فیلدهای اصلی:

- i. name, brand, category, price
- ii. skin_types (MultiSelectField)
- iii. concerns_targeted (JSONField)
- iv. ingredients (JSONField)
- v. rating, views, likes, sales_count
- vi. image, description, usage

متدها:

vii. average_rating → محاسبه میانگین امتیاز نظرات.

viii. get_skin_types_fa, get_category_display_fa → نمایش ترجمه

فارسی.

b. Comment (در اپ mainpage)

ذخیره نظرات کاربران درباره محصولات

فیلدها:

- i. product (ForeignKey → Product)
- ii. user (ForeignKey → User)
- iii. text, rating, created_at

محدودیت: هر کاربر فقط یک بار برای هر محصول نظر می دهد.

c. SearchHistory (در اپ accounts)

ذخیره تاریخچه جستجوی کاربران.

فیلدها:

- i. user
- ii. query
- iii. searched_at

d. ProductSearchHistory (در اپ accounts)

ثبت تعاملات کاربر با محصولات

فیلدها:

i. user, product

ii. interaction_type (View, Like, Wishlist, Cart, Purchase)

iii. timestamp

اهمیت: این داده در الگوریتم توصیه‌گر استفاده می‌شود (پیشنهاد محصولاتی که کاربران مشابه خریدند).

۳. سبد خرید و خریدها

a. ShoppingCartItem (در اپ profiles)

آیتم‌های موجود در سبد خرید هر کاربر

فیلدها:

i. user, product

ii. quantity

iii. added_at

متد:

iv. total_price(): محاسبه قیمت کل آیتم.

نقش: اتصال مستقیم بین سیستم فروشگاه و سیستم توصیه‌گر (چون

خریدها در تاریخچه ذخیره می‌شوند).

مدل‌های پروژه ترکیبی از اطلاعات کاربری (UserProfile + SkinProfile)، اطلاعات

محصولات (Product, Comment) و تعاملات کاربر (SearchHistory, PurchaseHistory, Cart)

هستند. این ترکیب به پروژه قابلیت فروشگاه آنلاین + سیستم توصیه‌گر پویا رو می‌ده.

۴.

مسیرها و ویوها (URLs & Views)

سیستم URL ها در Django مسئول اتصال درخواست های کاربر به منطق مربوطه (views) است. در این پروژه هر اپلیکیشن فایل urls.py مخصوص خودش را دارد و در SkincareRecommendationPlatform/urls.py جمع شده است.

۱. اپ accounts (مدیریت کاربر)

a. مسیرها:

- i. /accounts/register/ → ثبت نام
- ii. /accounts/login/ → ورود
- iii. /accounts/logout/ → خروج
- iv. /accounts/activate/<uidb64>/<token/ → فعال سازی حساب کاربری
- v. /accounts/ → بازگشت به صفحه اصلی

b. ویوهای کلیدی:

- i. register: ساخت حساب جدید + ارسال لینک فعال سازی به ایمیل.
- ii. user_login: ورود کاربر + هدایت به کوئیز (اگر تکمیل نشده باشد).
- iii. activate_account: فعال سازی کاربر از طریق لینک ایمیل.
- iv. home: بازگرداندن کاربر به صفحه اصلی.

۲. اپ profiles (پروفایل و سبد خرید)

a. مسیرها

- i. /profiles/profile/ → مشاهده پروفایل
- ii. /profiles/editprofile/ → ویرایش پروفایل
- iii. /profiles/add-to-cart/<id/ → افزودن محصول به سبد
- iv. /profiles/remove-from-cart/<id/ → کاهش تعداد
- v. /profiles/delete-cart/<id/ → حذف کامل محصول از سبد
- vi. /profiles/buy_products/ → خرید کل محصولات سبد
- vii. /profiles/user-routines/ → نمایش وضعیت روتین کاربر

viii. /profiles/routine/full/ → برنامه کامل

ix. /profiles/routine/hydration/ → برنامه آبرسانی

x. /profiles/routine/mini/ → برنامه مینیمال

b. ویوهای کلیدی:

i. profile_view: نمایش پروفایل + لیست محصولات سبد خرید.

ii. profile_edit: ویرایش اطلاعات پروفایل.

iii. add_to_cart / decrease_cart_item / remove_cart_item: مدیریت سبد خرید.

iv. buy_products: ثبت خرید و خالی کردن سبد.

v. full_routine_view / hydration_routine_view /

mini_routine_view: نمایش روتین‌های پوستی.

۳. اپ mainpage (صفحه اصلی و محصولات)

a. مسیرها:

i. / → صفحه اصلی (لیست محصولات)

ii. API/ → api/mainpage/ محصولات (DRF)

iii. /<product/<id/ → جزئیات محصول

iv. /search/ → جستجوی محصولات

v. /more-products/ → نمایش نتایج بیشتر

vi. /live-search/ → جستجوی زنده (Autocomplete)

vii. /<like/<id/ → لایک کردن محصول

b. ویوهای کلیدی

i. mainpage: نمایش محصولات با فیلتر، مرتب‌سازی و جستجو.

ii. search: الگوریتم هوشمند جستجو (ترکیب شباهت اسمی، نوع پوست،

تاریخچه کاربران مشابه)

iii. product_detail: نمایش جزئیات محصول + ثبت نظر کاربر.

iv. live_search: پیشنهاد سریع بر اساس ورودی کاربر.

v. like_product: لایک یا آن لایک کردن محصول.

vi. MainpageAPIView: API برای دریافت لیست محصولات.

۴. اپ quiz (کوئیز پوستی)

a. مسیرها

i. /quiz/ → نمایش سوالات کوئیز

ii. /quiz/quiz/ → اجرای کوئیز از داخل پروفایل

iii. /quiz/quiz/ → ریست وضعیت کوئیز برای انجام مجدد

iv. /quiz/quiz/ → رد کردن کوئیز

b. ویدیوهای کلیدی

i. skin_quiz_view: نمایش پرسشنامه و ثبت پاسخها.

ii. skin_quiz_view_quiz: نمایش پرسشنامه وقتی کاربر از پروفایل وارد

می شود.

iii. from_quiz: ریست وضعیت و اجازه انجام مجدد آزمون.

iv. skip_quiz: رد کردن آزمون و ذخیره وضعیت در پروفایل.

۵. اپ routine (برنامه پوستی)

a. مسیرها

i. /routine/routine/ → تولید روتین شخصی سازی شده

b. ویدیوهای کلیدی

i. routine_generator: تولید یا به روز رسانی برنامه مراقبتی بر اساس داده های

.SkinProfile

ii. find_step_products (تابع کمکی): انتخاب محصولات مناسب برای هر

مرحله از روتین (مثل پاک کننده، مرطوب کننده و ...).

۶. ساختار URL ها ماژولار است و هر اپ مسیرهای خودش را مدیریت می کند.

ویدیوها ترکیبی از:

a. CRUD ساده (ثبت نام، ویرایش پروفایل، مدیریت سبد خرید)

b. منطق هوشمند (جستجوی شخصی سازی شده، کوئیز پوستی، تولید روتین)
این بخش ستون فقرات تعامل کاربر با سیستم است

جریان کاربری

۱. ثبت نام و فعالسازی حساب

- a. کاربر وارد صفحه ثبت نام می شود.
 - b. اطلاعات پایه (ایمیل، نام کاربری، رمز عبور) را وارد می کند.
 - c. سیستم یک ایمیل فعال سازی ارسال می کند.
 - d. کاربر روی لینک فعال سازی کلیک می کند و حسابش فعال می شود.
- هدف: اطمینان از صحت ایمیل و جلوگیری از ثبت نام های غیر واقعی.

۲. ورود به سیستم

- a. کاربر وارد صفحه ورود می شود.
 - b. نام کاربری / ایمیل و رمز عبور را وارد می کند.
 - c. اگر کاربر برای اولین بار وارد شود:
 - d. به صورت خودکار به کوئیز پوستی هدایت می شود.
 - e. اگر قبلاً کوئیز تکمیل شده باشد → هدایت به صفحه اصلی.
- هدف: جمع آوری داده های اولیه برای سیستم توصیه گر.

۳. تکمیل کوئیز پوستی

- a. کاربر به سوالات کوئیز (نوع پوست، دغدغه ها، مشکلات پوستی و ...) پاسخ می دهد.
 - b. نتایج در مدل SkinProfile ذخیره می شود.
- این داده ها در آینده برای:
- i. پیشنهاد محصولات مناسب.
 - ii. تولید روتین های شخصی سازی شده.
- استفاده می شوند.

- c. هدف: ایجاد یک پروفایل پوستی منحصربه فرد برای هر کاربر

۴. مرور محصولات

- a. کاربر وارد صفحه اصلی می شود. می تواند محصولات را:
- i. جستجو کند (ساده یا زنده).

ii. براساس برند، نوع پوست، قیمت و ... فیلتر کند.

iii. براساس محبوبیت، امتیاز یا قیمت مرتب کند.

b. سیستم محصولات را با استفاده از:

i. SkinProfile کاربر

ii. رفتار کاربران مشابه

iii. تاریخچه جستجو و خرید

iv. شخصی سازی می کند.

۵. جزئیات محصول و تعامل

a. کاربر روی یک محصول کلیک می کند → وارد صفحه جزئیات محصول می شود.

b. اطلاعاتی مثل: توضیحات، ترکیبات، نوع پوست هدف، دغدغه های هدف و ... نمایش داده می شود.

c. کاربر می تواند:

i. نظر (Comment) و امتیاز ثبت کند.

ii. محصول را لایک کند.

iii. محصول را به سبد خرید اضافه کند.

هدف: افزایش تعامل کاربر با محصولات و جمع آوری داده برای توصیه های آینده.

۶. مدیریت پروفایل و سبد خرید

a. کاربر وارد بخش پروفایل می شود.

i. مشاهده و ویرایش اطلاعات شخصی

ii. مشاهده محصولات موجود در سبد خرید.

iii. تغییر تعداد محصولات یا حذف آنها.

iv. مشاهده جمع کل قیمت.

۷. دریافت برنامه مراقبت پوستی

a. کاربر به بخش رژیم پوستی (routines) مراجعه می کند.

b. با توجه به SkinProfile، سیستم برنامه های مختلف ارائه می دهد:

i. روتین کامل.

ii. روتین آپرسانی.

iii. روتین مینیمال.

c. هر مرحله از روتین همراه با محصولات پیشنهادی نمایش داده می شود.

۸. خرید محصولات

a. کاربر روی خرید کلیک می کند.

b. محصولات سبد خرید به عنوان خرید ثبت می شوند (در

ProductPurchaseHistory).

c. شمارش فروش محصول (sales_count) به روزرسانی می شود.

d. سبد خرید کاربر خالی می شود.

نیازمندی ها

۱. نرم افزار و ابزارهای پایه

- a. Python 3.10+ (توصیه شده: ۳/۱۱)
- b. Django 5.2.4 (نسخه‌ای که پروژه با آن ساخته شده)
- c. پایگاه داده پیش فرض: SQLite (قابل تغییر به MySQL یا PostgreSQL)
- d. ابزارهای توسعه:

- i. pip یا pipenv برای مدیریت پکیج‌ها
- ii. virtualenv یا محیط مجازی داخلی پایتون

۲. کتابخانه های پایتون

در فایل requirements.txt در دسترس هستند.

a. یک اند:

- i. django → فریمورک اصلی وب
- ii. djangorestframework → ساخت API
- iii. pillow → پشتیبانی از فایل‌های تصویری (عکس پروفایل و محصولات)
- iv. django-multiselectfield → ذخیره چند نوع پست برای هر محصول
- v. pandas → پردازش داده برای پیدا کردن کاربران مشابه
- vi. numpy → عملیات عددی (غیرمستقیم در pandas)

۳. تنظیمات اولیه

a. ایمیل:

- i. سیستم برای فعال سازی حساب کاربری از SMTP استفاده می‌کند.
- ii. در settings.py:

۱. نیاز به یک App Password در Gmail وجود دارد.

۲. مسیر فایل‌های استاتیک: /static/

۳. مسیر فایل‌های کاربر (media): /media/

۴. مسیر آپلود عکس محصولات: /media/product_images/

۵. مسیر آپلود عکس پروفایل: /media/profile_pictures/