



پروژه نهایی | مکتب ۱۳۰

کاستومی بک اند

کاستومی بک اند یک سرویس بک‌اند برای مدیریت یک پلتفرم چند فروشگاهه است. این سیستم امکانات مورد نیاز مشتریان، فروشندگان و مدیران را برای تعامل با هم فراهم می‌کند.

تحويل

- برای پروژه یک **README کامل و مناسب** نوشته شود (نصب، راه‌اندازی، متغیرهای محیطی، اجرای تست‌ها).
- **ERD** (نمودار روابط موجودیت‌ها) ترسیم شود.
- ریپوی گیت‌هاب پروژه تا پایان کار باید **Private** باقی بماند.
- استاد و مدرسان دوره به‌عنوان **collaborator** به ریپو اضافه شوند و پس از انجام، در گروه اطلاع داده شود.

نکات فنی

- رعایت اصول پایه‌ی Git الزامی است: کامیت‌های کوچک و مرتب، پیام‌های واضح، و استفاده از برنچینگ همراه با **Pull Request**
- اپلیکیشن **core** و **Logical Delete** الزامی می‌باشد.
- برای دیپلوی لازمه از **docker** و **docker compose** استفاده کنید.
- تنظیمات پروژه باید تمیز با استفاده از **env** تفکیک شده باشد.
- داشتن داکيومنتیشن به روز الزامی می‌باشد (شامل **request**، **response** و...)
- به‌کارگیری **Redis** و **Celery** بر اساس نیازهای پروژه الزامی است
- مطالعه جنگو HackSoft برای **BaseModel**

[Django Styleguide - Base Model](#)

نیازمندی‌های اصلی (اجباری)

احراز هویت و حساب کاربری

- کاربر بتواند ثبت‌نام و ورود کند (با ایمیل/شماره تلفن)
- ورود با **JWT token** انجام می‌شود.
- امکان ورود و تایید از طریق **OTP** فراهم باشد.
- کاربر باید بتواند اطلاعات پروفایل خود را مشاهده و ویرایش کند.
- کاربر باید بتواند آدرس‌های خود را ثبت، ویرایش و حذف کند.
- کاربر باید بتواند وضعیت سفارش‌های اخیر و تاریخچه سفارش‌های خود را در پروفایل کاربری مشاهده کند.
- نوشتن **تست** برای این app **الزامی** است

فروشنده و فروشگاه

- هر کاربر بتواند درخواست تبدیل به فروشنده بدهد.
- فروشنده **یک** فروشگاه داشته باشد که شامل مشخصات و آدرس است.
- فروشنده بتواند سفارش‌هایی که شامل محصولات اوست را مشاهده و وضعیت هر آیتم سفارش را تغییر دهد.
- نوشتن **تست** برای این app **الزامی** است.

محصولات و دسته‌بندی‌ها

- محصولات در دسته‌بندی‌های جا بگیرند
- محصول شامل نام، توضیحات، دسته، تصاویر، موجودی و قیمت باشد.
- هر محصول می‌تواند توسط چند فروشگاه عرضه شود.
- کاربران بتوانند محصولات را مشاهده، جستجو و فیلتر کنند.
- کاربران بتوانند برای محصولات نظر و امتیاز ثبت کنند.
- نوشتن **تست** برای این app **اختیاری** است.

سبد خرید و سفارش

- کاربر بتواند محصول را از طریق StoreItem¹ به سبد خرید اضافه کند.
- مجموع مبلغ و تخفیف در سبد خرید محاسبه شود.
- کاربر باید بتواند سبد خرید را به سفارش تبدیل کند (Checkout).
- هر سفارش شامل آدرس و آیتم‌های سفارش است.
- نوشتن **تست** برای این app **اختیاری** است.

پرداخت

- ایجاد رکورد پرداخت برای هر سفارش.
- تغییر وضعیت سفارش بعد از تایید پرداخت.
- نوشتن **تست** برای این app **اختیاری** است.

مدیریت (Admin Panel - API)

- ادمین بتواند کاربران را مدیریت کند (لیست، ویرایش، حذف).
- ادمین بتواند دسته‌بندی‌ها را مدیریت کند.
- ادمین بتواند سفارش‌ها و پرداخت‌ها را مدیریت کند.
- نوشتن **تست** برای این app **اختیاری** است.

مدیریت Admin Panel جنگو

- تغییر عنوان (title) پیش‌فرض "Django Administration" به نام پروژه
- تغییر لوگو و رنگ‌بندی مطابق برندینگ پروژه
- تغییر متن فوتر و هدر برای نمایش اطلاعات سازمان یا پروژه
- پیاده‌سازی Action‌های مدیریتی روی مدل‌ها مثلاً تأیید گروهی سفارش‌ها، تغییر وضعیت چند آیتم همزمان، فعال/غیرفعال کردن دسته‌ای کاربران یا محصولات
- نمایش Inline برای روابط (مثلاً نمایش آدرس‌های کاربر داخل صفحه‌ی User)
- استفاده از list_display، list_filter، search_fields برای کاربری بهتر (اختیاری)
- فیلترهای پیشرفته برای جستجو روی کاربران، سفارش‌ها و پرداخت‌ها (اختیاری)

¹ StoreItem: اسم مدلی که بین فروشگاه و محصول رابطه برقرار میکند

نقشه API

احراز هویت و امنیت

- ورود
 - (Login): </api/accounts/login/> → دریافت access و refresh token
 - ثبت نام
 - (Register): </api/accounts/register/>
 - جریان OTP:
 - درخواست </api/accounts/request-otp/>
 - تأیید </api/accounts/verify-otp/>
 - رفرش توکن: </api/accounts/token/refresh/>
-

مدیریت کاربران

- پروفایل من: </api/myuser/> → مشاهده، ویرایش یا حذف حساب کاربری
- آدرسها: </api/myuser/address/> → مدیریت آدرسهای کاربر

- ثبت نام به عنوان فروشنده: /api/myuser/register_as_seller/ → تبدیل کاربر به فروشنده و ایجاد فروشگاه
-

مدیریت کاربران توسط ادمین

- لیست، مشاهده، ویرایش و حذف کاربران </api/admin/users/> →

دسته بندی ها

- عمومی: </api/categories/> → مشاهده لیست دسته بندی ها

- مدیریتی

روی دسته بندی ها CRUD عملیات کامل → </api/admin/categories/> (Admin):

فروشگاه ها

- فروشگاه های عمومی: </api/stores/> → مشاهده فروشگاه ها
 - فروشگاه من (برای فروشنده ها): </api/mystore/> → مدیریت فروشگاه، آدرس ها و کالاها
 - مدیریت فروشگاه ها توسط ادمین: از طریق ماژول های کاربران و محصولات قابل دسترسی است
-

سبد خرید

- دریافت اطلاعات سبد خرید جاری کاربر </api/mycart/> →
- افزودن کالا به سبد [{/api/mycart/add_to_cart/{store_item_id}/](/api/mycart/add_to_cart/{store_item_id}/) →

- </api/mycart/items/> → مدیریت اقلام سبد خرید (مشاهده، به روزرسانی، حذف)
-

سفارش‌ها

- سفارش‌های کاربر: </api/myorders/> → مشاهده سفارش‌های ثبت‌شده توسط کاربر
 - تسویه حساب
 - تبدیل سبد خرید به سفارش → </api/orders/checkout/>: (Checkout)
 - مدیریت سفارش‌ها توسط ادمین: </api/orders/>
-

پرداخت‌ها

- </api/payments/> → ایجاد یا مشاهده پرداخت‌ها
 - </api/payments/{id}/start/> → آغاز فرآیند پرداخت
 - </api/payments/verify/> → بررسی نتیجه پرداخت از سمت درگاه (Zarinpal)
-

نظرات و امتیازات

- محصولات: /api/products/{id}/review_create/
- فروشگاه‌ها: /api/stores/{id}/review_create/