

## مستندات پروژه API فروشگاه اینترنتی

### ۱. معرفی پروژه

این پروژه یک API برای یک سیستم فروشگاه اینترنتی است که با زبان Python و فریم‌ورک FastAPI توسعه داده شده. امکانات آن شامل مدیریت کاربران، محصولات و سفارشات است. این API از احراز هویت JWT و ساختار معماری تمیز (Clean Architecture) استفاده می‌کند.

---

### ۲. تکنولوژی‌های استفاده شده

FastAPI\* برای پیاده‌سازی RESTful API

SQLite\* به عنوان دیتابیس سبک

PyJWT\* برای پیاده‌سازی احراز هویت

Pydantic\* برای مدیریت اسکیمای داده‌ها

Pytest\* برای تست یونیت

Postman\* برای تست دستی API

---

### ###۳. ساختار معماری پروژه

پروژه با معماری Clean Architecture پیاده‌سازی شده که شامل لایه‌های زیر است:

routes\* مدیریت مسیرهای API

schemas\* تعریف ساختار داده‌ها با Pydantic

services\* منطق تجاری سیستم

auth\* سیستم احراز هویت با JWT

models\* مدل‌های داده

tests\* تست‌های یونیت

---

### ###۴. احراز هویت و نقش‌ها

احراز هویت پروژه با JWT انجام می‌شود. کاربران با ارسال توکن Bearer در هدر درخواست، احراز هویت می‌شوند.

دو سطح دسترسی در نظر گرفته شده:

\*کاربر عادی (User)

\*مدیر (Admin)

برخی مسیرها فقط برای کاربران احراز هویت‌شده در دسترس هستند.

---

#### ۵. مدل‌های موجود در سیستم

\*\*User:\*\* شامل شناسه و نام کاربری

\*\*Product:\*\* شامل شناسه و نام محصول

\*\*Order:\*\* شامل شناسه، شناسه کاربر و شناسه محصول

---

#### ۶. عملیات‌های قابل انجام (CRUD)

\*لیست کاربران: /users/ (نیاز به احراز هویت دارد)

\*لیست محصولات: /products/

\*لیست سفارشات: /orders/

---

#### ۷. تست‌ها

در مسیر `tests/` تست‌هایی قرار دارد که با Pytest اجرا می‌شوند. یک نمونه تست اولیه داخل فایل `test_sample.py` پیاده‌سازی شده است و امکان توسعه بیشتر آن وجود دارد.

---

#### ۸. مجموعه Postman

یک فایل **Postman Collection** برای تست API آماده شده است. این فایل شامل توکن‌های نمونه برای تست مسیرهای محافظت‌شده نیز هست.

---

#### ۹. نحوه اجرا

برای اجرای پروژه مراحل زیر را انجام دهید:

۱. نصب کتابخانه‌های مورد نیاز:

```
pip install fastapi uvicorn python-jose[cryptography] pytest
```

۲. اجرای برنامه:

```
uvicorn main:app --reload
```

۳. اجرای تست‌ها:

```
pytest
```