

**پروژه 1:** "پلتفرم جامعه" نمای کلی پروژه: یک پلتفرم جامعه با استفاده از جنگو REST Framework ایجاد کنید که در آن کاربران می‌توانند از طریق JSON API ها تعامل داشته باشند. این پروژه به دانشجویان اجازه می‌دهد که با ساختن API هایی برای ثبت‌نام کاربر، ارسال پست، نظر دادن، لایک کردن و قابلیت‌های تعامل کاربر آشنا شوند.

ویژگی‌های اصلی:

**ثبت‌نام و احراز هویت کاربر:**

- نقاط پایانی API برای ثبت‌نام، ورود و خروج کاربران.
- API برای بازیابی رمز عبور.
- API برای مدیریت پروفایل‌های کاربر (GET، PUT).

**تعامل کاربر:**

**ارسال محتوا:**

- نقاط پایانی API برای ایجاد، ویرایش و حذف پست‌ها (POST، PUT، DELETE).
- API برای بارگذاری تصاویر و مدیریت تگ‌های اختیاری (POST، PUT).

**نظر دادن:**

- نقاط پایانی API برای ارسال نظرات و نظرات تودرتو (POST).
- API برای ویرایش و حذف نظرات (PUT، DELETE).

**لایک کردن:**

- API برای لایک یا عدم لایک کردن پست‌ها و نظرات (POST، DELETE).

**فید محتوا:**

- API برای بازیابی یک فید از پست‌های اخیر (GET).
- API برای فیلتر کردن فید بر اساس تگ‌ها یا کلمات کلیدی جستجو (GET با پارامترهای جستجو).
- API برای مرتب‌سازی فید بر اساس جدیدترین، بیشترین لایک شده یا بیشترین نظر داده شده (GET با پارامترهای مرتب‌سازی).

**پروفایل‌های کاربر:**

- API برای بازیابی داده‌های پروفایل کاربر، شامل پست‌ها، نظرات و محتوای لایک شده (GET).
- API برای دنبال کردن/عدم دنبال کردن کاربران و مدیریت فیدهای شخصی‌سازی شده (POST، DELETE).
- API برای ارسال پیام‌های مستقیم بین کاربران (POST).

**اطلاعه‌ها:**

- API برای مدیریت اطلاعیه‌ها (GET، POST، DELETE).
  - درخواست‌های GET باید همه اطلاعیه‌های جدید از آخرین بار دریافت اطلاعیه‌ها را برگردانند.
  - درخواست‌های POST اجازه می‌دهند که اطلاعیه‌ها در هر بازه زمانی بازیابی شوند.

**ویژگی‌های اختیاری:**

- API ها برای ایجاد و مدیریت گروه‌ها، انجمن‌ها، گالری‌های رسانه‌ای، رویدادها و غیره (عملیات CRUD).
- یکپارچه‌سازی REST API برای تعامل با برنامه‌های خارجی.

## ساختار پروژه:

### مدل‌ها:

- User، Post، Comment، Like، Profile، Notification و مدل‌های اختیاری مانند Group، Event، Media.

### ویوها:

- ویوهای API (با استفاده از ViewSet ها یا APIView های جنگو REST Framework) برای عملیات CRUD روی پست‌ها، نظرات و پروفایل‌ها.
- ویوهای API مبتنی بر تابع برای اقداماتی مانند لایک کردن پست، دنبال کردن کاربر و غیره.

### سریالایزرها:

- سریالایزرها برای همه مدل‌ها برای تبدیل داده‌های JSON.

### URL ها:

- الگوهای URL به نقاط پایانی API برای هر اقدام نگاشت شده‌اند.

### معیارهای ارزیابی:

- عملکرد API: پیاده‌سازی صحیح نقاط پایانی برای ثبت‌نام، ارسال پست، نظر دادن، لایک کردن و غیره.
- ساختار کد: استفاده صحیح از ویوهای DRF، سریالایزرها و اجزای قابل استفاده مجدد.
- سفارشی‌سازی: تشویق دانشجویان به افزودن نقاط پایانی سفارشی یا تغییر موجود.
- مستندات API: مستندات واضح و دقیق برای هر نقطه پایانی.
- یکپارچه‌سازی ویژگی‌های اختیاری: امتیاز اضافی برای دانشجویانی که API های اضافی پیاده‌سازی می‌کنند.

## پروژه 2: "پلتفرم تجارت الکترونیک"

مرور پروژه: توسعه یک پلتفرم تجارت الکترونیک با استفاده از فریمورک Django REST که کاربران می‌توانند از طریق API‌های JSON با فروشگاه آنلاین تعامل داشته باشند. این پروژه شامل توسعه API برای مرور محصولات، مدیریت سبد خرید، فرآیند پرداخت، پردازش سفارش و مدیریت موجودی است.

ویژگی‌های اصلی:

ثبت‌نام و احراز هویت کاربر:

- نقاط پایانی API برای ثبت‌نام کاربر، ورود به سیستم و مدیریت پروفایل (POST, PUT, GET).

مدیریت محصول:

- نقاط پایانی API برای مدیران جهت ایجاد، ویرایش و حذف محصولات (POST, PUT, DELETE).
- API برای مدیریت تصاویر محصول، توضیحات، برچسب‌ها و موجودی (عملیات CRUD).
- API برای مدیریت دسته‌بندی با اعتبارسنجی JSON schema (عملیات CRUD).
- پیاده‌سازی یک مدل دسته‌بندی با ساختار درختی. هر دسته‌بندی باید یک فیلد اضافی به نام `attributes_schema` داشته باشد که حاوی `jsonschema` ویژگی‌های محصول است.
- هر محصول باید یک دسته‌بندی (تنها دسته‌بندی‌های برگ مجاز هستند) و یک فیلد ویژگی (در قالب JSON) داشته باشد که باید با `jsonschema` دسته‌بندی اعتبارسنجی شود.

سبد خرید:

- نقاط پایانی API برای افزودن، به‌روزرسانی یا حذف اقلام از سبد خرید (POST, PUT, DELETE).
- API برای حفظ سبد خرید بین جلسات (با استفاده از پایگاه داده).

فرآیند پرداخت:

- نقاط پایانی API برای مرور سبد خرید و پیشرفت به مرحله پرداخت (GET, POST).
- API برای ایجاد سفارشات و پاکسازی سبد خرید پس از اتمام پرداخت (POST).

مدیریت سفارش:

- API برای کاربران جهت مشاهده سفارشات گذشته و پیگیری سفارشات فعلی (GET).
- API برای مدیران جهت مدیریت سفارشات، به‌روزرسانی وضعیت‌ها و رسیدگی به بازپرداخت‌ها (PUT, DELETE).

اعلان‌ها:

- API برای مدیریت اعلان‌ها (GET, POST, DELETE).
- درخواست‌های GET باید تمام اعلان‌های جدید از آخرین باری که اعلان‌ها دریافت شده‌اند را برگرداند.
- درخواست‌های POST به کاربران اجازه می‌دهد اعلان‌ها را در هر بازه زمانی دریافت کنند.
- مدیران اعلان‌هایی از سفارشات جدید دریافت می‌کنند و کاربران از هرگونه تغییر وضعیت در سفارشات خود مطلع می‌شوند.

جستجو و فیلتر محصولات:

- API برای جستجوی محصولات با نام یا توضیحات و فیلتر کردن بر اساس محدوده قیمت، دسته‌بندی یا برچسب‌ها (GET با پارامترهای کوئری).

ویژگی‌های اختیاری:

- API ها برای مدیریت تخفیف ها، کوپن ها، نظرات محصولات، لیست علاقه مندی ها و غیره (عملیات CRUD).

#### ساختار پروژه:

#### مدل ها:

- کاربر، محصول، سبد خرید، سفارش، پرداخت و مدل های اختیاری مانند نظر، کوپن.

#### ویوها:

- ویوهای API (با استفاده از Django REST Framework) برای عملیات CRUD و مدیریت سبد خرید.
- ویوهای API مبتنی بر تابع برای پرداخت، پردازش پرداخت و مدیریت سفارش.

#### سریالایزرها:

- سریالایزرها برای تمام مدل ها جهت تبدیل داده های JSON.

#### آدرس ها:

- الگوهای URL برای نقاط پایانی API جهت کاتالوگ محصولات، سبد خرید، پرداخت و مدیریت سفارش.

#### معیارهای ارزیابی:

- عملکرد API: پیاده سازی صحیح نقاط پایانی برای کاتالوگ محصولات، سبد خرید، پرداخت و غیره.
- ساختار کد: استفاده مناسب از ویوها، سریالایزرها و اجزای قابل استفاده مجدد DRF.
- سفارشی سازی: تشویق دانشجویان به افزودن نقاط پایانی سفارشی یا تغییر موارد موجود.
- مستندات API: مستندات واضح و دقیق برای هر نقطه پایانی.
- ادغام ویژگی های اختیاری: امتیاز اضافی برای دانشجویانی که API های اضافی را پیاده سازی می کنند.

## پروژه ۳: "سیستم نوبت‌دهی در حوزه بهداشت و درمان"

مروری بر پروژه: ایجاد یک سیستم نوبت‌دهی در حوزه بهداشت و درمان با استفاده از چارچوب جنگو REST که در آن بیماران و پزشکان از طریق JSON API ها با پلتفرم تعامل دارند. این پروژه بر روی ساخت API برای نقش‌های کاربری، زمان‌بندی نوبت‌ها و مدیریت پرونده‌های پزشکی تمرکز دارد.

### ویژگی‌های اصلی:

1. نقش‌های کاربری:
  - نقاط پایانی API برای نقش‌های مختلف کاربری: ادمین، پزشک، و بیمار (عملیات CRUD).
  - کنترل دسترسی مبتنی بر نقش پیاده‌سازی شده از طریق مجوزها و محدودیت‌های دیدگاه DRF.
2. مدیریت پزشکان:
  - API برای ادمین‌ها جهت مدیریت پروفایل‌های پزشکان (عملیات CRUD).
  - API برای پزشکان جهت مدیریت دسترسی‌پذیری و زمان‌های نوبت‌دهی (GET, PUT).
3. مدیریت بیماران:
  - نقاط پایانی API برای بیماران جهت مدیریت پروفایل‌های خود (عملیات CRUD).
  - API برای مشاهده پزشکان موجود و رزرو نوبت‌ها (GET, POST).
4. زمان‌بندی نوبت‌ها:
  - API برای مرور پزشکان موجود و رزرو نوبت‌ها بر اساس دسترسی‌پذیری (GET, POST).
  - API برای پزشکان جهت مشاهده و مدیریت زمان‌بندی نوبت‌های خود (GET, PUT).
  - API برای یادآوری نوبت‌ها از طریق ایمیل یا SMS (POST).
5. پرونده‌های پزشکی:
  - API برای پزشکان جهت ایجاد و به‌روزرسانی پرونده‌های پزشکی بیماران (POST, PUT).
  - API برای بیماران جهت مشاهده تاریخچه پزشکی خود (GET).

### ویژگی‌های اختیاری:

- API ها برای مدیریت نسخه‌ها، صورتحساب‌ها، و ادغام با سیستم‌های خارجی پرونده‌های بهداشتی یا ارائه‌دهندگان بیمه (عملیات CRUD).

### ساختار پروژه:

1. مدل‌ها:
  - کاربر (با نقش‌های گسترده شده)، پزشک، بیمار، نوبت، پرونده پزشکی، و مدل‌های اختیاری مانند نسخه و صورتحساب.
2. ویوها:
  - ویوهای API (با استفاده از چارچوب جنگو REST) برای عملیات CRUD بر روی پزشکان، بیماران، و نوبت‌ها.
  - ویوهای مبتنی بر عملکرد برای زمان‌بندی، اعلان‌ها، و یادآوری‌ها.
3. سریالایزرها:
  - سریالایزرها برای تمامی مدل‌ها جهت تبدیل داده‌های JSON.
4. آدرس‌ها (URLs):
  - الگوهای URL که به نقاط پایانی API برای اقدامات نقش-محور، رزرو نوبت و مدیریت نقش‌برداري شده‌اند.

### معیارهای ارزیابی:

1. عملکرد API: پیاده‌سازی صحیح نقاط پایانی برای نقش‌های کاربری، رزرو نوبت‌ها و مدیریت زمان‌بندی.
2. ساختار کد: استفاده مناسب از ویوها، سریالایزرها و کنترل دسترسی مبتنی بر نقش DRF.
3. سفارشی‌سازی: تشویق دانشجویان به افزودن نقاط پایانی سفارشی یا تغییر نقاط موجود.
4. مستندسازی API: مستندسازی شفاف و دقیق برای هر نقطه پایانی.
5. ادغام ویژگی‌های اختیاری: امتیاز اضافی برای دانشجویانی که API های اضافی را پیاده‌سازی می‌کنند.