



Department  
of Industrial  
Engineering

دانشکده مهندسی صنایع - دانشگاه صنعتی شریف

نیم سال تحصیلی ۰۱۰۳

برنامه نویسی پیشرفته (۲۱۰۶۰)

## ساخت نرم افزار وقت گیری آنلاین

علی ابوسعیدی ۹۸۱-۴۵۹۶

فرزام اسدیان ۹۸۱-۴۶۳۶

سید مهدی ساعدی ۴۰۱۱-۴۱۵۷

حسین اکبری ۴۰۱۱-۳۸۹۳

در فاز قبلی، ما پروژه را به صورت command-based پیاده سازی کردیم و در این فاز به دنبال این هستیم که کد جنگو آن را ارائه دهیم.

در این گزارش فایل های models, views, forms توضیح داده شده اند.

حال با توضیح مدل های ساخته شده آغاز می کنیم.

کلاس CustomUserManager:

این کلاس یک custom user manager است که از BaseUserManager گسترش یافته است. این دو متد اساسی را فراهم می کند:

create\_user: برای ایجاد یک کاربر عادی با ایمیل، نام کاربری و رمز عبور استاندارد.

create\_superuser: برای ایجاد یک superuser با فیلدهای اضافی مانند is\_staff و is\_superuser که به True تنظیم شده اند. این متد برای admin functionality در پنل ادمین Django حیاتی است.

مدل کاربر (user model): از AbstractUser به ارث برده، که امکان گسترش مدل کاربر داخلی Django را فراهم می کند. این مدل کاربر سفارشی شامل موارد زیر است:

USER\_TYPE\_CHOICES: یک تاپل که انواع مختلف کاربران را تعریف می کند. پیش نمایش شروع این تاپل را با حداقل یک نوع کاربر، 'بیمار'، نشان می دهد که پیشنهاد می کند سیستم بین نقش های کاربری (بیماران، و کارکنان) تفاوت قائل است.

مشاهدات و تبعات سفرashi سازی برای دسترسی بر اساس نقش: استفاده از یک custom user model با USER\_TYPE\_CHOICES نشان دهنده این است که سیستم برای مدیریت نقش های مختلف در برنامه طراحی شده است، مانند بیماران و کارکنان کلینیک، هر کدام با سطوح دسترسی و اجازه های متفاوت.

امنیت و مدیریت کاربر: استفاده از یک custom user manager برای ایجاد کاربران و superusers، به Django کمک می کند تا user authentication و authorization securely را مدیریت کند.

قابلیت گسترش: گسترش AbstractUser امکان اصلاحات و افزودنی های آتی به مدل کاربر را بدون تغییر در سیستم مدیریت کاربر داخلی Django فراهم می کند. این تنظیمات یک پایه محکم برای سیستم رزرواسیون کلینیک است، که برای تفکیک بین عملکردهای بیمار و کارکنان در سیستم ضروری است.

ویژگی های کاربر: user\_type: یک CharField که نقش کاربر را ذخیره می کند، با استفاده از USER\_TYPE\_CHOICES ذکر شده برای اعتبارسنجی.

use\_otp: یک BooleanField که نشان می دهد آیا کاربر رمز یک بار مصرف (OTP) را برای امنیت اضافی فعال کرده است یا خیر.

otp: یک CharField که OTP فعلی را، در صورت لزوم، ذخیره می کند.

otp\_expiry: یک DateTimeField که زمان انقضای OTP را ثبت می کند.

متدهای کاربر:

generate\_otp: یک OTP شش رقمی تصادفی تولید می کند، زمان انقضا را تنظیم می کند (۵ دقیقه از تولید) و این جزئیات را در پروفایل کاربر ذخیره می کند.

check\_otp: یک OTP ورودی را با OTP ذخیره شده کاربر تأیید می کند و بررسی می کند که آیا در بازه زمانی معتبر است.

update\_profile: به روزرسانی ایمیل و رمز عبور کاربر را امکان پذیر می سازد و بر اهمیت مدیریت حساب کاربری تأکید دارد.

User Management (مدیریت کاربر): متد update\_profile نشان می دهد که کاربران کنترل تنظیمات حساب خود را دارند، اجازه می دهد تا آنها ایمیل و رمز عبور خود را مستقیماً به روز کنند، که برای حفظ حساب های کاربری به روز و امن ضروری است.

بقیه ی کلاس های داخل model نیز به اختصار در زیر توضیح داده شده اند.

## کلاس Clinic

- هدف: این کلاس اطلاعات مربوط به کلینیک‌ها مانند نام، آدرس و اطلاعات تماس را ذخیره می‌کند.
- متدها:
- Update\_clinic\_info: این متد اجازه می‌دهد تا آدرس یا اطلاعات تماس کلینیک به‌روزرسانی شوند، که قابلیت انعطاف‌پذیری در مدیریت اطلاعات کلینیک‌ها را فراهم می‌کند.

## کلاس Availability

- هدف: نگهداری اطلاعات در دسترس بودن کلینیک‌ها برای تاریخ‌های مشخص.
- مشخصات: با استفاده از foreign key به کلینیک مرتبط می‌شود و برای هر تاریخ، وضعیت در دسترس بودن را ذخیره می‌کند.
- کنترل یکتایی: با تعریف unique\_together برای clinic و date، اطمینان حاصل می‌شود که برای هر کلینیک در هر تاریخ، فقط یک رکورد در دسترس بودن وجود دارد.
- متدها:
- set\_availability\_for\_clinic: این متد اجازه می‌دهد تا برای یک کلینیک و تاریخ مشخص، وضعیت در دسترس بودن تعیین شود.

## کلاس Appointment

- هدف: مدیریت قرارهای ملاقات، شامل وضعیت، زمان و مشخصات کلینیک و کاربر.
- مشخصات: شامل وضعیت‌های قرار ملاقات مانند در انتظار، تایید شده و لغو شده.
- متدها:
- Save: قبل از ذخیره‌سازی، بررسی می‌کند که آیا زمان انتخاب شده برای قرار ملاقات در دسترس است.
- is\_time\_slot\_available: بررسی می‌کند که آیا زمان انتخابی برای قرار ملاقات در دسترس است یا خیر.
- Cancel: قرار ملاقات را با تغییر وضعیت به لغو شده، لغو می‌کند.
- Reschedule: قرار ملاقات را به زمان جدیدی موکول کرده و وضعیت آن را به در انتظار تغییر می‌دهد.

## کلاس Notification

- هدف: ارسال اطلاعیه به کاربران.
- مشخصات: شامل کاربر، پیام و زمان ارسال اطلاعیه.
- متدها:
- `send_notification`: برای ارسال اطلاعیه به یک کاربر خاص.
- `send_bulk_notifications`: برای ارسال اطلاعیه به چندین کاربر.

حال به بررسی `views.py` می‌پردازیم.

### ثبت نام کاربر (sign\_up\_view)

این تابع از فرم `CustomUserCreationForm` برای ثبت نام کاربران جدید استفاده می‌کند. در صورتی که درخواست به روش POST باشد، فرم با داده‌های ارسالی پر می‌شود. اگر فرم معتبر باشد، یک نمونه کاربر جدید ایجاد شده و بر اساس نوع کاربر (مثلاً بیمار)، اطلاعات بیشتری ممکن است به آن اضافه شود. در نهایت، کاربر ذخیره شده و به صفحه اصلی هدایت می‌شود.

### ورود کاربر (log\_in\_view)

این تابع از فرم `AuthenticationForm` برای احراز هویت کاربران و ورود آنها استفاده می‌کند. پس از اعتبارسنجی موفقیت آمیز، کاربر وارد سیستم شده و به صفحه موفقیت یا پروفایل کاربری هدایت می‌شود. در صورت عدم اعتبارسنجی، پیغام خطایی نمایش داده می‌شود.

### مشاهده نوبت‌های کاربر (user\_appointments\_view)

این تابع نوبت‌های کاربر را با استفاده از متد `get_appointments_for_user` نمایش می‌دهد. این اطلاعات در صفحه‌ای به کاربر نمایش داده می‌شود.

### حذف نوبت (remove\_appointment)

این تابع به کاربران با دسترسی مدیر (`admin`) امکان می‌دهد تا نوبت‌ها را حذف کنند.

### افزودن ظرفیت نوبت‌دهی (add\_appointment\_capacity)

این تابع اجازه می‌دهد که ظرفیت نوبت‌دهی افزوده شود. فرم `AppointmentForm` برای این منظور استفاده می‌شود و پس از اعتبارسنجی، نوبت جدید ذخیره و کاربر به لیست نوبت‌ها هدایت می‌شود.

نمایش نوبت‌های موجود (`fetch_available_appointments`)

این تابع نوبت‌های موجود را از یک API خارجی دریافت و نمایش می‌دهد. در صورت بروز خطا، پیغام خطا به کاربر نمایش داده می‌شود.

تنظیم مجدد ظرفیت کلینیک (`adjust_clinic_capacity`)

با استفاده از فرم `ClinicCapacityForm`، این تابع امکان تنظیم مجدد ظرفیت کلینیک را فراهم می‌کند. داده‌های فرم پس از اعتبارسنجی به یک API ارسال شده و در صورت موفقیت یا شکست، پاسخ مناسب به کاربر نمایش داده می‌شود.

رزرو نوبت (`book_appointment`)

کاربران می‌توانند با استفاده از فرم `BookAppointmentForm`، نوبت رزرو کنند. پس از اعتبارسنجی، نوبت ذخیره شده و کاربر به مشاهده نوبت‌ها هدایت می‌شود.

به‌روزرسانی پروفایل (`update_profile`)

کاربران می‌توانند اطلاعات پروفایل خود را با استفاده از فرم `UpdateProfileForm` به‌روزرسانی کنند. پس از اعتبارسنجی، تغییرات ذخیره شده و کاربر به مشاهده پروفایل خود هدایت می‌شود.

مشاهده اطلاعیه‌ها (`view_notifications`)

کاربران می‌توانند اطلاعیه‌های مرتبط با خود را مشاهده کنند.

به‌روزرسانی اطلاعات کلینیک (`update_clinic_info`)

کارکنان می‌توانند اطلاعات کلینیک را با استفاده از فرم `UpdateClinicInfoForm` به‌روزرسانی کنند.

حال به بررسی `forms.py` می‌پردازیم:

فرم رزرو نوبت (AppointmentForm)

این فرم برای ایجاد و ویرایش رکوردهای رزرو نوبت استفاده می‌شود. شامل فیلدهای زمان‌بندی، کلینیک، کاربر و وضعیت است.

فرم ایجاد حساب کاربری (CustomUserCreationForm)

این فرم برای ثبت نام کاربران جدید استفاده می‌شود و از UserCreationForm گسترش یافته است. فیلد user\_type با انتخاب‌های 'patient' و 'staff' به کاربر اجازه می‌دهد نقش خود را انتخاب کند.

فرم به‌روزرسانی پروفایل (UpdateProfileForm)

این فرم برای به‌روزرسانی اطلاعات پروفایل کاربری استفاده می‌شود و شامل فیلدهایی مانند نام کاربری، ایمیل، نام و نام خانوادگی است. این فرم امکان ویرایش اطلاعات شخصی کاربر را فراهم می‌کند.

فرم رزرو نوبت (BookAppointmentForm)

این فرم برای رزرو نوبت‌ها توسط کاربران استفاده می‌شود. فرم از مدل Appointment استفاده می‌کند و شامل فیلدهای کلینیک و زمان‌بندی است. در متد \_\_init\_\_، کاربر فعلی به فرم اضافه می‌شود تا در پردازش‌های بعدی قابل استفاده باشد.

فرم به‌روزرسانی اطلاعات کلینیک (UpdateClinicInfoForm)

این فرم برای به‌روزرسانی اطلاعات کلینیک‌ها توسط کاربران دارای دسترسی مدیریتی استفاده می‌شود. شامل فیلدهایی مانند نام، آدرس و اطلاعات تماس است.