گزارش نهایی پروژه وب

سامانه اتوماسیون اداری پلیس

گروه ۸
اعضای تیم:
سیدامیرحسین موسوی‌فرد
رضا اسلامی‌ابیانه
سیداحمد موسوی‌اول

استاد: دکتر علی ابریشمی
ترم پاییز ۱۴۰۴-۱۴۰۵

۷ اسفند ۱۴۰۴

فهرست مطالب

۱ مقدمه ۳

۲ معماری کلی پروژه ۳

۳ گزارش فنی بک‌اند ۳
۱.۳ استک فناوری ۳
۲.۳ ساختار اپلیکیشن‌ها ۳
۳.۳ طراحی API ۴
۴.۳ قوانین داده و دسترسی ۴
۵.۳ پرداخت (اختیاری) ۴
۶.۳ اجرای محلی ۴

۴ گزارش فنی فرانت‌اند ۴
۱.۴ استک فناوری ۴
۲.۴ ساختار پروژه ۵
۳.۴ مسیریابی ۵
۴.۴ یکپارچگی با API ۵
۵.۴ اجرای محلی ۵

۵ گزارش فنی Docker ۵
۱.۵ هدف ۵
۲.۵ سرویس‌های Compose Docker ۶
۳.۵ پورت‌ها و هاVolume ۶
۴.۵ دستورات پرکاربرد ۶
۵.۵ تذکر محیط توسعه در مقابل تولید ۶

۶ گزارش نهایی پروژه ۶
۱.۶ اعضای تیم و مسئولیت‌ها ۶
۲.۶ قراردادهای توسعه ۷
۳.۶ مدیریت پروژه و تقسیم وظایف ۷
۴.۶ موجودیت‌های کلیدی سامانه ۸
۵.۶ پکیج‌های NPM استفاده‌شده (حداکثر ۶ مورد) ۸
۶.۶ نمونه کدهای تولیدشده با هوش مصنوعی ۹
۱.۶.۶ منطق پرداخت و Verify در بک‌اند ۹
۲.۶.۶ اعتبارسنجی قانون پرداخت ۹
۳.۶.۶ فیلتر مظنون در فرانت‌اند ۹
۷.۶ قوت‌ها و ضعف‌های هوش مصنوعی در توسعه فرانت‌اند ۹
۸.۶ قوت‌ها و ضعف‌های هوش مصنوعی در توسعه بک‌اند ۱۰
۹.۶ نیازسنجی اولیه و نهایی پروژه ۱۰

۷ جمع‌بندی ۱۰

۱ مقدمه

این سند گزارش نهایی پروژه‌ی طراحی و پیاده‌سازی سامانه اتوماسیون اداری پلیس است که با استفاده از معماری مبتنی بر Django ***(Backend)*** و React ***(Frontend)*** توسعه یافته است. هدف از این سند ارائه‌ی نمای کلی از معماری، تصمیمات فنی، ساختار پروژه، و فرآیند توسعه می‌باشد.

۲ معماری کلی پروژه

پروژه از سه لایه اصلی تشکیل شده است:

- **بک‌اند (Backend)**: مبتنی بر ***Django*** و ***Django REST Framework*** با پایگاه داده ***SQLite*** (قابل تغییر به ***PostgreSQL*** در محیط عملیاتی).
- **فرانت‌اند (Frontend)**: ***React*** همراه با ***Vite***، ***React Router*** و ***Axios*** برای ارتباط با ***API***.
- **ocker**: جهت ایجاد محیط توسعه یکسان برای تمام اعضای تیم.

۳ گزارش فنی بک‌اند

۱.۳ استک فناوری

- ***Django 4.x***
- ***Django REST Framework***
- ***Simple JWT*** برای احراز هویت مبتنی بر توکن
- ***SQLite*** (در حال حاضر)

۲.۳ ساختار اپلیکیشن‌ها

پروژه بک‌اند شامل اپلیکیشن‌های مستقل زیر است:

accounts مدیریت کاربران، ثبت‌نام، ورود و پروفایل

rbac کنترل دسترسی پویا بر اساس نقش (Role) و مجوز (Permission)

cases مدیریت پرونده‌ها، شکوائیه‌ها، صحنه جرم و چرخه حیات پرونده

evidence مدیریت انواع ادله (شاهدی، بیولوژیک، خودرو، مدارک هویتی و ...)

investigation تخته کارآگاه، مظنونین، بازجویی و اعلان‌ها

judiciary جلسات دادگاه و صدور رأی

rewards ثبت گزارش‌های مردمی و فرآیند پاداش

payments پرداخت وثیقه و جریمه (اختیاری، متصل به درگاه زرین‌پال)

dashboard آمار و نمایش ماژول‌ها بر اساس نقش

۳.۳ طراحی API

- **نشانی پایه**: ***http://localhost:8000/api/***
- **سبک**: ***RESTful*** همراه با نقاط پایانی خاص گردش کار (نه صرفاً ***CRUD***)
- **احراز هویت**: ارسال توکن ***JWT*** در هدر ***Authorization***
- **مستندسازی**: با استفاده از ***drf-spectacular*** (Swagger)

۴.۳ قوانین داده و دسترسی

- هر کاربر دارای فیلدهای یکتا مانند نام کاربری، ایمیل، شماره ملی و تلفن است.
- سیستم ***RBAC*** پویا بوده و مدیر می‌تواند نقش‌ها را بدون تغییر کد ویرایش کند.
- کاربران می‌توانند چند نقش داشته باشند.
- قوانین کسب‌وکار در سطح سریالایزرها و ویوها اعمال می‌شود.

۵.۳ پرداخت (اختیاری)

اپلیکیشن ***payments*** امکان تعریف مبلغ وثیقه توسط گروهبان، بررسی مظنونین واجد شرایط، اتصال به درگاه آزمایشی زرین‌پال و بازگشت به صفحه نتیجه را فراهم می‌کند.

۶.۳ اجرای محلی

بدون ***Docker***:

```
cd /Users/reza/Documents/sag/backend
source .venv/bin/activate
python manage.py migrate
python manage.py seed_roles
python manage.py runserver
```

۴ گزارش فنی فرانت‌اند

۱.۴ استک فناوری

- ***React 18***
- ***Vite*** (به عنوان build tool و dev server)
- ***React Router DOM*** برای مسیریابی
- ***Axios*** برای درخواست‌های HTTP
- ***html2canvas*** برای خروجی تصویری از تخته کارآگاه

۲.۴ ساختار پروژه

- ***src/pages***: صفحات اصلی (ورود، داشبورد، پرونده‌ها، ادله، تخته کارآگاه، دادگاه، پاداش، پرداخت)
- ***src/components***: کامپوننت‌های قابل استفاده مجدد (مانند ***ProtectedRoute***)
- ***src/context***: ***AuthContext*** برای مدیریت وضعیت احراز هویت و نقش کاربر
- ***src/api***: تنظیمات ***Axios*** و ارتباط با بک‌اند
- ***src/App.jsx***: تعریف مسیرها
- ***src/ThemeContext.jsx***: کنترل تم روشن/تاریک

۳.۴ مسیریابی

مسیرهای عمومی:

- **/** — صفحه اصلی
- ***/login*** — ورود
- ***/register*** — ثبت‌نام

مسیرهای محافظت‌شده:

- ***/dashboard*** ،***/cases*** ،***/evidence*** ،***/board*** ،***/reports*** ،***/judiciary*** ،***/rewards*** ،***/payments*** ،***/admin-rbac***

۴.۴ یکپارچگی با API

- ارتباط با ***API*** از طریق ***Axios*** و استفاده از توکن ***JWT*** ذخیره‌شده.
- نمایش پیام‌های خطای دریافتی از سرور به کاربر.

۵.۴ اجرای محلی

```
cd /ProjectRoot/frontend
npm install
npm run dev -- --host
```

۵ گزارش فنی Docker

۱.۵ هدف

استفاده از ***Docker*** برای ایجاد محیط توسعه یکسان، حذف مشکلات وابستگی‌ها و ساده‌سازی راه‌اندازی پروژه برای اعضای تیم.

۲.۵ سرویس‌های Compose Docker

فایل ***docker-compose.yml*** شامل دو سرویس است:

- ***backend***: اجرای مهاجرت‌ها، بذر نقش‌ها و راه‌اندازی سرور ***Django*** روی پورت ***8000***
- ***frontend***: اجرای سرور توسعه ***Vite*** روی پورت ***5173*** و وابسته به ***backend***

۳.۵ پورت‌ها و هاVolume

- پورت‌ها: ***8000:8000*** (بک‌اند) و ***5173:5173*** (فرانت‌اند)
- ***Volume*** برای ***backend***: ***./backend:/app*** — برای بازتاب تغییرات لحظه‌ای کد
- ***Volume*** برای ***frontend***: ***./frontend:/app*** و ***/app/node$_{m}odules$***

۴.۵ دستورات پرکاربرد

```
#
docker compose up

#
docker compose up backend
docker compose up frontend

#
docker compose down

#
docker compose logs -f backend
docker compose logs -f frontend
```

۵.۵ تذکر محیط توسعه در مقابل تولید

تنظیمات فعلی برای توسعه سریع بهینه شده است. برای محیط تولید باید از فایل‌های ایستا فرانت‌اند و سرور ***WSGI/ASGI*** برای بک‌اند استفاده شود.

۶ گزارش نهایی پروژه

۱.۶ اعضای تیم و مسئولیت‌ها

- **رضا اسلامی ابیانه**: طراحی و پیاده‌سازی بک‌اند با ***Django/DRF***، مدل‌سازی موجودیت‌ها، ***RBAC***، ***JWT***، مستندسازی ***Swagger*** و تست‌نویسی.
- **سیدامیرحسین موسوی‌فرد**: طراحی و پیاده‌سازی فرانت‌اند با ***React + Vite***، صفحات اصلی، داشبورد ماژولار، مدیریت ***State*** و اتصال ***API***.

- **سیداحمد موسوی‌اول**: یکپارچه‌سازی فرانت و بک‌اند، ***Docker Compose***، رفع باگ‌های سناریویی، تست نهایی و مستندسازی.

بازه زمانی کلی: از تحلیل نیازمندی تا تکمیل فازهای پرونده، شواهد، تحقیق، محاکمه، پاداش و پرداخت.

۲.۶ قراردادهای توسعه

- **نام‌گذاری**:
  - بک‌اند: ***snake_case*** برای فیلدها و توابع، ***PascalCase*** برای نام مدل‌ها.
  - فرانت‌اند: ***PascalCase*** برای کامپوننت‌ها، ***camelCase*** برای هوک‌ها و توابع.
  - ***Endpoint***ها: ***RESTful*** و معنادار مانند ***/api/cases/cases/*** و ***/api/payments/bail/***.
- **قالب پیام کامیت**:
  - ***feat***: ... برای ویژگی جدید
  - ***fix***: ... برای رفع باگ
  - ***refactor***: ... برای بازآرایی
  - ***test***: ... برای تست
  - ***docs***: ... برای مستندات
- **قوانین Request Pull**:
  - هر ***PR*** شامل توضیح تغییر، دلیل، اسکرین‌شات (در صورت نیاز) و وضعیت تست.
- **قوانین کیفیت**:
  - عدم ادغام بدون گذراندن تست‌های اصلی.
  - عدم تغییر رفتار حساس بدون بررسی سطح دسترسی.

۳.۶ مدیریت پروژه و تقسیم وظایف

کارها به ***Epic***های اصلی تقسیم شد:

۱. احراز هویت و نقش‌ها
۲. تشکیل پرونده (شکوائیه / صحنه جرم)
۳. شواهد
۴. تخته کارآگاه و روند بررسی مظنون
۵. بازجویی و امتیازدهی

۶. محاکمه

۷. تحت پیگیری شدید

۸. پاداش

۹. پرداخت وثیقه و جریمه (اختیاری)

اولویت اجرا: بک‌اند پایدار و API سپس ***UI*** سپس اصلاح مجوزها و سناریوهای واقعی سپس تست و ریزه‌کاری ***UX***.

۴.۶ موجودیت‌های کلیدی سامانه

- ***User***: کاربر پایه با اطلاعات هویتی یکتا.
- ***Role***، ***Permission***، ***UserRole***: پیاده‌سازی ***RBAC*** پویا.
- ***Case***: هسته پرونده با منبع، شدت جرم، وضعیت و افراد درگیر.
- ***ComplaintSubmission***: چرخه ثبت شکوائیه و بازگشت به شاکی/کارآموز/افسر.
- ***CaseComplainant***، ***CaseWitness***، ***CaseLog***: ثبت چند شاکی، شاهدان و تاریخچه.
- موجودیت‌های شواهد (شاهدی، زیستی، خودرو، مدارک هویتی، سایر).
- ***DetectiveBoard***، ***BoardNode***، ***BoardEdge***: تخته کارآگاه و اتصال مدارک.
- ***Suspect***، ***SuspectSubmission***، ***Interrogation***، ***Notification***: مظنون، تأیید گروهبان، بازجویی و اعلان.
- ***CourtSession***: محاکمه و ثبت رأی برای هر مظنون.
- ***Reward*** / ***Tip***: ثبت اطلاعات مردمی، بررسی، صدور کد یکتا.
- ***BailPayment***: مدیریت پرداخت وثیقه/جریمه و اتصال به درگاه.

۵.۶ پکیج‌های NPM استفاده‌شده (حداکثر ۶ مورد)

۱. ***react***: هسته کتابخانه ساخت ***UI***.

۲. ***react-dom***: رندر ***React*** در مرورگر.

۳. ***react-router-dom***: مسیریابی ***SPA***.

۴. ***axios***: ارتباط با ***API***.

۵. ***vite***: سرعت در توسعه و بیلد.

۶. ***html2canvas***: خروجی تصویری از تخته کارآگاه.

۶.۶ نمونه کدهای تولیدشده با هوش مصنوعی

۱.۶.۶ منطق پرداخت و Verify در بک‌اند

```
payload = {
    "merchant_id": merchant_id,
    "amount": int(obj.amount),
    "description": f"Bail/Fine payment for case #{obj.case_id} suspect #{obj.suspect_id}",
    "callback_url": callback_url,
}
result = zarinpal_post(settings.ZARINPAL_REQUEST_URL, payload)
```

۲.۶.۶ اعتبارسنجی قانون پرداخت

```
if suspect.status == Suspect.Status.ARRESTED:
    if case.severity not in [Case.Severity.LEVEL_2, Case.Severity.LEVEL_3]:
        raise ValidationError("Only level 2 and level 3 arrested suspects are eligible.")
elif suspect.status == Suspect.Status.CRIMINAL:
    if case.severity != Case.Severity.LEVEL_3 or not sergeant_approved:
        raise ValidationError("Sergeant approval is required for level 3 criminal release.")
```

۳.۶.۶ فیلتر مظنون در فرانت‌اند

```
const filteredSuspects = useMemo(() => {
  if (!selectedCaseId) return []
  return suspects.filter((s) => Number(s.case) === selectedCaseId)
}, [suspects, selectedCaseId])
```

۷.۶ قوت‌ها و ضعف‌های هوش مصنوعی در توسعه فرانت‌اند

- **قوت‌ها:**
  - تولید سریع اسکلت صفحات و فرم‌ها.
  - کمک در اتصال ***API*** و مدیریت وضعیت‌های بارگذاری/خطا.
  - سرعت بالا در رفع باگ‌های کوچک رابط کاربری.
- **ضعف‌ها:**
  - احتمال ناهماهنگی با منطق دقیق مجوزها.
  - نیاز به بازبینی انسانی برای تجربه کاربری واقعی سناریوهای پیچیده.
  - احتمال تولید پیام خطای عمومی و غیرکاربردی.

۸.۶ قوت‌ها و ضعف‌های هوش مصنوعی در توسعه بک‌اند

- **قوت‌ها:**
  - سرعت در ساخت ***CRUD***، ***ViewSet***، ***Serializer*** و مسیرهای ***REST***.
  - کمک در مدل‌سازی اولیه و نوشتن اعتبارسنجی‌های کسب‌وکاری.
  - کمک مؤثر در تست‌نویسی و پوشش سناریوهای اصلی.
- **ضعف‌ها:**
  - احتمال خطای اولیه در قوانین دقیق ***RBAC*** و چندنقشی.
  - نیاز به اصلاح دستی در سناریوهای ***stateful*** چندمرحله‌ای.
  - نیاز به بازبینی امنیتی و پایداری برای محیط عملیاتی.

۹.۶ نیازسنجی اولیه و نهایی پروژه

- **نیازسنجی اولیه**: پوشش نقش‌ها، ثبت پرونده، شواهد، تحلیل، محاکمه، پاداش، پرداخت. تمرکز روی پیاده‌سازی فنی.
- **نیازسنجی نهایی**: تمرکز بیشتر روی گردش کار واقعی، مجوزهای دقیق، تجربه کاربری، تفکیک دقیق ***Complaint*** و ***Case***، پشتیبانی از چندنقشی.
- **نقاط قوت تصمیم‌ها:**
  - معماری ماژولار ***app-by-app*** در ***Django***.
  - ***RBAC*** قابل تغییر بدون تغییر کد.
  - جداسازی واضح لایه‌ها در ***React***.
- **نقاط ضعف تصمیم‌ها:**
  - پیچیدگی بالا در مدیریت حالت‌های پرونده.
  - نیاز به تست‌های بیشتر برای سناریوهای مرزی.
  - وابستگی به تست دستی بیشتر در گردش‌کارهای طولانی.

۷ جمع‌بندی

سامانه با موفقیت با استفاده از ***Django + DRF + JWT*** در بک‌اند و ***React + Vite*** در فرانت‌اند پیاده‌سازی شد. تمام الزامات اصلی پروژه پوشش داده شد و بخش اختیاری پرداخت نیز به درگاه آزمایشی متصل گردید. پروژه برای توسعه‌های بعدی از جمله بهبود ***CI/CD***، افزایش تست‌های ***E2E*** و ارتقای امنیت آماده است.