سند نیازمندی‌ها



**Version 1.0**

**Document Identification**

|  |  |  |  |
| --- | --- | --- | --- |
| Document Name | Requirements | | |
| Document ID | CP\_DOC\_0002 | Version | 1.0 |
| Created by | Parsa Enayati | Created Date | 2025-07-04 |
| Updated by |  | Updated Date |  |
| Reviewed by |  | Reviewed Date |  |
| Approved by |  | Approved Date |  |
| Document Status | Done | Business Owner | SUT |
| Stakeholders |  | Environment | Calmative Portfolio Application |

# مقدمه

سامانه «Calmative Portfolio» یک بستر تحت ‌وب برای مدیریت سبد دارایی‌های متنوع است. این سند نیازمندی‌های عملکردی (Functional) و غیرعملکردی (Non-Functional) را به‌صورت تفکیک‌شده ارائه می‌دهد تا مبنای طراحی، توسعه و آزمون قرار گیرد.

# نیازمندی‌های عملکردی (Functional Requirements)

1. ثبت‌نام کاربر با ایمیل، گذرواژه و ارسال لینک تأیید.
2. ورود ایمن و دریافت توکن JWT؛ ذخیره در کوکی HTTP Only .
3. بازیابی گذرواژه از طریق ایمیل بازیابی.
4. ایجاد پورتفولیو با نام و توضیح؛ ذخیره در DB .
5. ویرایش / حذف پورتفولیو (Cascade حذف دارایی‌ها).
6. افزودن دارایی به پورتفولیو: نام، نماد، نوع، مقدار، قیمت خرید و فعلی، تاریخ خرید.
7. ویرایش / حذف دارایی؛ در حذف، تاریخچه قیمت مرتبط نیز حذف می‌شود.
8. ثبت تاریخچه قیمت به‌ازای هر تغییر قیمت دارایی.
9. مشاهده پورتفولیو (لیست دارایی‌ها، سود/زیان، نمودار توزیع، نمودار قیمت).
10. داشبورد تجمیعی برای همه پورتفولیوها با KPIها و نمودارها.

توصیه‌گر هوشمند: تولید توصیه خرید/نگهداری/کاهش برای هر دارایی و توصیه تنوع سبد.

1. برچسب اولویت توصیه (زیاد، متوسط، کم) بر اساس میزان رشد/افت.
2. پنل مدیریت برای مدیر سیستم (مدیریت کاربران، انواع دارایی سفارشی، گزارش‌ها).
3. جستجوی سریع در پورتفولیوها و دارایی‌ها.
4. مستندسازی خودکار API در swagger .
5. واکنش‌گرایی رابط در موبایل و تبلت.

# نیازمندی‌های غیرعملکردی (Non-Functional Requirements)

1. کارایی | ۹۵٪ درخواست‌های API باید ≤ ۲ثانیه پاسخ دهند.
2. امنیت | رمزنگاری گذرواژه با BCrypt؛ همه ارتباطات بر بستر HTTPS .
3. بومی‌سازی | رابط کاملاً راست‌به‌چپ و فارسی.
4. دسترسی‌پذیری | رعایت WCAG 2.1 سطح AA برای عناصر اصلی.
5. مقیاس‌پذیری | امکان استقرار افقی API پشت Load Balancer .
6. دسترس‌پذیری | Uptime سالانه ≥ ۹۹٪.
7. نگه‌داری | پوشش تست واحد ≥ ۶۰٪ برای لایه سرویس؛ کد طبق اصول Clean Code .
8. لاگ و پایش | ثبت لاگ سطوح Info/Warning/Error؛ متریک‌های Prometheus .
9. پشتیبان‌گیری | بکاپ روزانه پایگاه‌داده و فایل‌های بارگذاری‌شده.
10. حریم خصوصی | داده‌های کاربر بدون مجوز در اختیار شخص ثالث قرار نگیرد.
11. سازگاری مرورگر | پشتیبانی از آخرین نسخه Chrome ,Edge, Firefox .
12. نصب و اجرا | راه‌اندازی محلی با یک دستور docker compose up برای وابستگی‌ها (DB و SMTP).

# قیود طراحی (Design Constraints)

1. استفاده از .NET 6 و EF Core
2. SQL Server به‌عنوان DBMS
3. Chart.js برای نمودارها
4. Bootstrap ۵ RTL برای UI

# معیار پذیرش (Acceptance Criteria)

* تمام نیازمندی‌های عملکردی و غیر‌عملکردی در تست پذیرش باید تأیید شوند.
* خروجی توصیه‌گر برای سناریوهای تست از پیش تعریف‌شده بررسی و مطابقت داشته باشد.
* رابط کاربری بدون خطا در مرورگرهای هدف بارگذاری شود.