



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر
مهندسی نرم افزار ۱

"فاز ۳ پروژه"

Wiki Service: موضوع پروژه:

استاد درس: دکتر کلباسی

نام گروه:

Nullterminated

فهرست بخش‌ها

1. سناریو پروژه 1
2. داستان‌های کاربر 8
3. نیازمندی‌های استخراج شده از داستان‌های کاربر و تفکیک آن‌ها 14
4. اعتبارسنجی نیازمندی‌ها 24
5. اطلاعات جمع‌آوری شده از مطالعات موردی 36
6. گزارش کوتاهی از جلسات برگزار شده با تدریس‌یار 41

سناریو پروژه

بخش اول: تحلیل زمینه و ضرورت وجود سرویس (Context & Rationale)

جایگاه Wiki Service در سامانه کلی

Wiki Service هسته محتوایی و اطلاعاتی سامانه است و نقش مرکزی در تأمین زمینه‌ای برای سایر میکروسرویس‌ها ایفا می‌کند:

1. برای Map Service:
هنگامی که کاربر روی یک شهر یا جاذبه در نقشه کلیک می‌کند، Wiki Service اطلاعات تاریخی، فرهنگی و جغرافیایی آن مکان را فراهم می‌آورد.
2. برای Timeline Service:
رویدادهای تاریخی که در خط زمان نمایش داده می‌شوند، به مقالات تفصیلی در Wiki پیوند می‌خورند تا کاربر بتواند زمینه و تأثیرات آن رویداد را درک کند.
3. برای Recommendation Service:
محتوای مقالات، برچسب‌های موضوعی و تاریخچه مطالعه کاربران به عنوان ورودی برای الگوریتم‌های پیشنهاددهنده استفاده می‌شود.
4. برای Media Service:
تصاویر، ویدئوها و محتوای چندرسانه‌ای که کاربران آپلود می‌کنند، در مقالات Wiki جای‌گذاری می‌شوند تا تجربه‌ای غنی‌تر ارائه شود.
5. برای Trip Plan Service:
اطلاعات موجود در مقالات (مانند بهترین فصل بازدید، ساعات بازدید، نکات ایمنی) در برنامه‌ریزی سفر مورد استفاده قرار می‌گیرد.

ارزش افزوده Wiki Service

این سرویس دو ارزش اصلی ایجاد می‌کند:

1. یکپارچه‌سازی دانش:
به جای اینکه کاربر مجبور باشد از چندین منبع مختلف اطلاعات جمع‌آوری کند، Wiki Service تمام اطلاعات مرتبط را در یک مکان متمرکز می‌کند.
2. ایجاد ارتباطات معنایی:
با استفاده از پیوندهای داخلی و برچسب‌گذاری هوشمند، کاربر می‌تواند به راحتی از یک موضوع به موضوعات مرتبط دیگر منتقل شود و دانش خود را عمیق‌تر کند.

مخاطبان و ذینفعان

Wiki Service برای طیف گسترده‌ای از کاربران طراحی شده است:

- گردشگران داخلی و خارجی که قصد دارند قبل از سفر اطلاعات کافی درباره مقصد کسب کنند
- پژوهشگران و دانشجویان که نیاز به منبع معتبر برای تحقیقات خود دارند
- علاقه‌مندان به فرهنگ و تاریخ ایران که می‌خواهند دانش خود را گسترش دهند
- نویسندگان و کارشناسان که می‌خواهند دانش خود را با دیگران به اشتراک بگذارند
- سایر سرویس‌های سامانه که به یک منبع اطلاعاتی ساختاریافته نیاز دارند

بخش دوم: تبیین دامنه مسئله و حدود عملکرد سرویس (Domain Characterization)

دامنه موضوعی Wiki Service

Wiki Service بر روی چهار حوزه اصلی متمرکز است:

1. تاریخ و میراث فرهنگی

- دوره‌های تاریخی ایران (از تمدن‌های باستانی تا دوران معاصر)
- شخصیت‌های تاریخی و فرهنگی مؤثر
- رویدادهای مهم تاریخی
- بناهای تاریخی و آثار باستانی
- سنت‌ها و آیین‌های سنتی

نمونه مقالات: "دوره صفویه"، "میدان نقش جهان"، "شاه عباس کبیر"، "هخامنشیان"، "تخت جمشید"

2. جغرافیا و مکان‌ها

- استان‌ها و شهرهای ایران
- روستاها و مناطق کمتر شناخته‌شده
- ویژگی‌های جغرافیایی هر منطقه (آب و هوا، ارتفاع، موقعیت)
- اطلاعات جمعیت‌شناختی و اقتصادی

نمونه مقالات: "استان اصفهان"، "شهر کاشان"، "روستای ماسوله"، "دشت لوت"

3. فرهنگ و هنر

- هنرهای سنتی ایرانی (معماری، خوشنویسی، نگارگری، سفالگری)
- ادبیات فارسی و شعر کلاسیک
- موسیقی سنتی و آوازی
- صنایع دستی (قالی‌بافی، خاتم‌کاری، میناکاری)
- آداب و رسوم محلی

نمونه مقالات: "معماری ایرانی"، "حافظ شیرازی"، "موسیقی دستگاهی"، "قالی فارس"

4. طبیعت و محیط زیست

- پارک‌های ملی و مناطق حفاظت‌شده
- کوهستان‌ها، دریاچه‌ها و رودخانه‌ها
- گونه‌های گیاهی و جانوری بومی
- اقلیم و ویژگی‌های طبیعی مناطق مختلف

نمونه مقالات: "پارک ملی گلستان"، "کوه دماوند"، "دریاچه ارومیه"، "یوز ایرانی"

ساختار داده‌ای مقالات

اطلاعات اصلی:

- شناسه یکتا (UUID)
- عنوان (فارسی و انگلیسی)
- متن اصلی (فارسی، با امکان ترجمه)
- خلاصه مقاله (۲-۳ جمله)
- تصویر شاخص

متادیتا:

- نویسنده اولیه
- تاریخ و زمان ایجاد
- آخرین ویراستار
- تاریخ و زمان آخرین ویرایش
- تعداد بازدید
- امتیاز کیفیت (۱-۵ ستاره)

دسته‌بندی و سازماندهی:

- دسته اصلی (تاریخ، جغرافیا، فرهنگ، طبیعت)
- زیردسته‌ها
- برچسب‌های موضوعی (Tags)
- کلمات کلیدی

ارتباطات:

- پیوندهای داخلی به سایر مقالات
- پیوندهای به مکان‌های جغرافیایی (Map Service)
- پیوندهای به رویدادهای تاریخی (Timeline Service)
- منابع و مراجع خارجی

نسخه‌برداری:

- شماره نسخه فعلی

- لیست نسخه‌های قبلی
- تاریخچه تغییرات (Revision History)
- امکان مقایسه نسخه‌ها (Diff)

کاربران و نقش‌های مختلف

1. بازدیدکننده مهمان (Guest Visitor)

- مشاهده مقالات
- جستجو در محتوا
- دریافت پیشنهاد مقالات مرتبط

2. کاربر ثبت‌نام‌شده (Registered User)

- تمام امکانات مهمان
- ذخیره مقالات مورد علاقه
- دریافت پیشنهادهاى شخصی‌سازی‌شده
- ثبت نظرات و امتیازدهی

3. نویسنده/کارشناس (Contributor)

- تمام امکانات کاربر عادی
- ایجاد مقاله جدید
- ویرایش مقالات موجود
- پیشنهاد تغییرات

4. ویراستار/مدیر محتوا (Editor/Moderator)

- تمام امکانات نویسنده
- تأیید یا رد مقالات جدید
- بررسی و تأیید ویرایش‌ها
- حذف محتوای نامناسب
- مدیریت برچسب‌ها و دسته‌بندی‌ها

محدودیت‌ها و قیود

محدودیت‌های محتوایی:

- محتوا باید به زبان فارسی باشد
- مقالات باید مرتبط با چهار حوزه اصلی باشند
- منابع معتبر و قابل استناد
- محتوای سیاسی، مذهبی یا تبلیغاتی مجاز نیست

محدودیت‌های فنی:

- حداکثر طول مقاله: ۲۰۰۰۰ کاراکتر
- حداکثر تعداد تصاویر: ۲۰
- فرمت‌های مجاز: WebP، PNG، JPG
- حداکثر حجم تصویر: ۵ مگابایت
- حداکثر تعداد برچسب‌ها: ۱۰

محدودیت‌های دسترسی:

- ایجاد و ویرایش مقالات نیاز به احراز هویت دارد
- مقالات جدید پس از تأیید ویراستار منتشر می‌شوند
- کاربران عادی نمی‌توانند مقالات را حذف کنند

قابلیت‌های کلیدی

1. جستجوی پیشرفته:

- جستجوی متنی ساده
- جستجوی معنایی (Semantic Search) با NLP
- فیلتر بر اساس دسته، برچسب، تاریخ انتشار
- مرتب‌سازی بر اساس ارتباط، محبوبیت، تاریخ

2. نسخه‌برداری و تاریخچه:

- ذخیره خودکار نسخه‌ها
- مشاهده تفاوت میان نسخه‌ها (Visual Diff)
- بازگشت به نسخه‌های قبلی
- ردیابی تغییرات هر نویسنده

3. پیوندهای هوشمند:

- شناسایی خودکار موضوعات مرتبط
- پیشنهاد پیوندهای داخلی
- ایجاد شبکه دانش (Knowledge Graph)

4. تولید محتوای خودکار:

- خلاصه‌سازی خودکار مقالات
- برچسب‌گذاری موضوعی خودکار
- پیشنهاد عنوان‌های بهتر
- تشخیص و پیشنهاد پیوندهای داخلی

5. کنترل کیفیت:

- سیستم امتیازدهی کاربران
- بررسی املائی و دستوری
- تشخیص محتوای کپی شده (Plagiarism Detection)
- سیستم گزارش‌دهی برای محتوای نامناسب

بخش سوم: سناریوی عملیاتی (Operational Scenario)

پیش‌شرط‌ها (Pre-Conditions)

- کاربر به اینترنت دسترسی دارد
- سیستم و سرورها فعال هستند
- سرویس‌های خارجی (Payment Gateway، Email Service) آماده پاسخ‌گویی هستند

- کاربر ممکن است حساب داشته یا نداشته باشد

بازیگران (Actors)

- **Normal User**: کاربر عادی
- **Admin**: مدیریت کاربران و تنظیمات
- **External Services**: سرویس ایمیل، پرداخت، API های اطلاعاتی

جریان کلی سناریو (Main Operational Flow)

1. کاربر وارد صفحه اصلی می شود
2. تصمیم می گیرد ثبت نام کند، وارد شود یا به عنوان مهمان استفاده محدود داشته باشد
3. در صورت ثبت نام:
 - وارد کردن اطلاعات
 - اعتبارسنجی سیستم
 - ارسال تأییدیه به **Email Service**
 - ایجاد حساب و ورود
4. در صورت ورود:
 - ارسال اطلاعات هویتی
 - تطبیق با داده های ذخیره شده
 - فعال شدن **Session**
5. دسترسی به بخش های مختلف سیستم
6. انجام عملیات خاص (خرید، سرویس ویژه) → **Payment Gateway**
7. ثبت رفتار کاربر برای تحلیل و بهبود

رفتار سیستم (System Behavior)

- جمع آوری و پردازش داده ها
- استخراج مستقیم داده ها
- تشخیص موارد غیر عادی
- نمایش مسیرهای جایگزین در صورت خطا

پس شرطها (Post-Conditions)

- ایجاد حساب کاربر
- فعال بودن **Session**
- پردازش و ارسال داده ها
- ثبت تراکنش ها و لاگ ها

سناریوهای مرزی / خطا (Edge Cases & Exceptions)

- اطلاعات اشتباه -> پیام خطا

- قطع سرویس خارجی -> **Retry** + پیام
- **Timeout** یا اختلال شبکه -> ادامه/لغو
- درخواست **API** خارجی شکست بخورد -> استفاده از **cache**
- دسترسی ناکافی -> نمایش **Access Denied**

هدف نهایی این سناریو

- تصویر کامل تعاملات سیستم
- استخراج **User Story** بدون بازنویسی
- پایه‌ای برای تحلیل رفتار کاربران و طراحی معماری سرویس
- ایجاد دید مستقل برای مطالعات موردی (**Case Study**)

داستان‌های کاربر

داستان کاربر US-01: جستجوی مقاله در Wiki

بازیگر: کاربر عادی

هدف: یافتن اطلاعات درباره یک موضوع خاص

سناریوی تعامل:

1. کاربر عبارت جستجو را در کادر جستجو وارد می‌کند
2. سیستم با استفاده از جستجوی معنایی، مقالات مرتبط را پیدا می‌کند
3. نتایج به صورت رتبه‌بندی شده نمایش داده می‌شوند
4. کاربر روی مقاله مورد نظر کلیک کرده و محتوا را مشاهده می‌کند

معیارهای پذیرش:

- نتایج در کمتر از ۲ ثانیه نمایش داده شوند
- حداقل ۵ نتیجه مرتبط نمایش داده شود
- هر نتیجه شامل عنوان، خلاصه کوتاه و امتیاز ارتباط باشد
- در صورت عدم یافتن نتیجه، پیام مناسب نمایش داده شود
- امکان فیلتر نتایج بر اساس تاریخ و محبوبیت وجود داشته باشد

وابستگی‌ها: ندارد

داستان کاربر US-02: مشاهده مقاله با پیوندهای مرتبط

بازیگر: کاربر علاقه‌مند به یادگیری عمیق

هدف: خواندن مقاله و دسترسی به مطالب مرتبط

سناریوی تعامل:

1. کاربر مقاله‌ای را باز می‌کند
2. سیستم محتوای مقاله را همراه با پیوندهای داخلی نمایش می‌دهد

3. کاربر روی پیوندهای مرتبط کلیک می‌کند
4. مقاله جدید بدون قطع تجربه کاربری باز می‌شود

معیارهای پذیرش:

- پیوندهای داخلی به وضوح مشخص باشند
- کلیک روی پیوند باعث بارگذاری سریع مقاله جدید شود
- امکان بازگشت به مقاله قبلی وجود داشته باشد
- تاریخچه مرور ذخیره شود
- نمایش پیوندهای مرتبط بر اساس بیشترین ارتباط محتوا

وابستگی‌ها: US-01

داستان کاربر US-03: ایجاد مقاله جدید توسط نویسنده

بازیگر: نویسنده محتوا / کارشناس

هدف: افزودن اطلاعات جدید به پایگاه دانش

سناریوی تعامل:

1. نویسنده وارد بخش ایجاد مقاله می‌شود
2. فرم ایجاد مقاله شامل عنوان، متن، برچسب‌ها و تصاویر نمایش داده می‌شود
3. نویسنده اطلاعات را وارد می‌کند
4. سیستم مقاله را ذخیره و منتشر می‌کند

معیارهای پذیرش:

- امکان استفاده از ویرایشگر غنی (Rich Text Editor) وجود داشته باشد
- امکان افزودن پیوندهای داخلی و خارجی
- امکان آپلود تصویر
- اعتبارسنجی فیلدهای اجباری
- ذخیره خودکار در حین نوشتن
- نمایش پیش‌نمایش مقاله قبل از انتشار

وابستگی‌ها: احراز هویت کاربر

داستان کاربر US-04: ویرایش مقاله موجود

بازیگر: ویراستار / نویسنده

هدف: به روز رسانی یا اصلاح اطلاعات موجود

سناریوی تعامل:

1. ویراستار مقاله را باز کرده و گزینه ویرایش را انتخاب می‌کند
2. محتوای فعلی در ویرایشگر بارگذاری می‌شود
3. ویراستار تغییرات را اعمال می‌کند
4. سیستم نسخه جدید را ذخیره و تاریخچه تغییرات را حفظ می‌کند

معیارهای پذیرش:

- تمام نسخه‌های قبلی حفظ شوند
- امکان مشاهده تفاوت‌ها بین نسخه‌ها
- ذخیره اطلاعات ویرایشگر و زمان ویرایش
- امکان بازگشت به نسخه قبلی

وابستگی‌ها: US-03، احراز هویت

داستان کاربر US-05: مشاهده تاریخچه تغییرات مقاله

بازیگر: کاربر / مدیر محتوا

هدف: بررسی تغییرات اعمال شده بر مقاله

سناریوی تعامل:

1. کاربر گزینه "تاریخچه" را انتخاب می‌کند
2. سیستم لیست نسخه‌ها را نمایش می‌دهد
3. کاربر دو نسخه را انتخاب کرده و تفاوت‌ها را مشاهده می‌کند
4. در صورت نیاز، نسخه قدیمی را بازیابی می‌کند

معیارهای پذیرش:

- نمایش تاریخ، ساعت و نویسنده هر نسخه
- مقایسه بصری نسخه‌ها
- امکان بازیابی نسخه‌های قدیمی
- نمایش توضیح تغییرات (اگر وارد شده باشد)

وابستگی‌ها: US-04

داستان کاربر US-07: تولید برجسب‌های موضوعی خودکار

بازیگر: سیستم (فرآیند پس‌زمینه)

هدف: سازماندهی بهتر مقالات و بهبود جست‌وجو

سناریوی تعامل:

1. پس از ذخیره مقاله جدید، سیستم متن را تحلیل می‌کند
2. با استفاده از مدل‌های NLP، برجسب‌های موضوعی استخراج می‌شوند
3. برجسب‌ها به مقاله اضافه می‌شوند
4. نویسنده می‌تواند برجسب‌ها را ویرایش کند

معیارهای پذیرش:

- حداقل ۵ برجسب مرتبط تولید شود
- برجسب‌ها با محتوای مقاله همخوانی داشته باشند
- امکان تأیید یا رد برجسب‌ها توسط نویسنده
- زمان تولید برجسب‌ها کمتر از ۵ ثانیه

وابستگی‌ها: US-03

داستان کاربر US-08: گزارش مقاله نامناسب

بازیگر: کاربر

هدف: اطلاع‌رسانی درباره محتوای نادرست یا نامناسب

سناریوی تعامل:

1. کاربر گزینه "گزارش مقاله" را انتخاب می‌کند
2. فرم گزارش شامل نوع مشکل و توضیحات نمایش داده می‌شود
3. کاربر اطلاعات را وارد کرده و ارسال می‌کند
4. سیستم گزارش را ثبت و به تیم مدیریت محتوا اطلاع می‌دهد

معیارهای پذیرش:

- امکان انتخاب نوع مشکل (محتوای نادرست، تصاویر نامناسب، اسپم و...)
- ثبت تاریخ و کاربر گزارش‌دهنده

- اطلاع رسانی خودکار به مدیران
- نمایش پیام موفقیت آمیز پس از ارسال

وابستگی ها: US-01

داستان کاربر US-09: فیلتر مقالات بر اساس دسته بندی

بازیگر: کاربر

هدف: یافتن سریع مقالات مرتبط با یک دسته خاص

سناریوی تعامل:

1. کاربر وارد بخش مقالات می شود
2. گزینه فیلتر بر اساس دسته بندی را انتخاب می کند
3. سیستم مقالات مرتبط با دسته انتخاب شده را نمایش می دهد
4. کاربر مقالات را مشاهده و انتخاب می کند

معیارهای پذیرش:

- امکان انتخاب چند دسته به صورت همزمان
- نمایش تعداد مقالات موجود در هر دسته
- زمان بارگذاری کمتر از ۲ ثانیه
- امکان بازگشت به نمایش همه مقالات

وابستگی ها: US-01

داستان کاربر US-10: اعلان تغییرات مقاله

بازیگر: کاربر / مدیر محتوا

هدف: اطلاع از تغییرات جدید در مقالات مورد علاقه

سناریوی تعامل:

1. کاربر مقاله یا دسته مورد علاقه را دنبال می کند
2. سیستم تغییرات جدید در مقاله یا دسته را بررسی می کند
3. در صورت وجود تغییر، اعلان برای کاربر ارسال می شود
4. کاربر می تواند اعلان را بررسی و به مقاله دسترسی پیدا کند

معیارهای پذیرش:

- نمایش نوع تغییر (ویرایش متن، افزودن تصویر، تغییر برجسب)
- امکان فعال/غیرفعال کردن اعلان‌ها
- ارسال اعلان در کمتر از ۵ ثانیه پس از تغییر
- آرشیو اعلان‌ها برای مراجعات بعدی

وابستگی‌ها: US-04، US-05

نیازمندی‌های استخراج شده از داستان‌های کاربر و تفکیک آن‌ها

نیازمندی‌های عملکردی (Functional Requirements)

FR-1.1 – پردازش جستجوی معنایی

شرح: سیستم باید قادر باشد با استفاده از الگوریتم‌های جستجوی معنایی، مقالات مرتبط با عبارت جستجوی کاربر را یافته و رتبه‌بندی کند.

محرک: ورود عبارت جستجو توسط کاربر

پیش‌شرط‌ها: پایگاه داده شامل مقالات ایندکس شده باشد

پس‌شرط‌ها: لیست مقالات مرتبط تولید شده باشد

ورودی‌ها: عبارت جستجو (رشته متنی)

خروجی‌ها: لیست مقالات شامل عنوان، خلاصه و امتیاز ارتباط

داستان مرتبط: US-01

FR-1.2 – نمایش نتایج جستجو

شرح: سیستم باید نتایج جستجو را به صورت صفحه‌بندی شده نمایش دهد.

محرك: دریافت نتایج از موتور جستجو

پیش‌شرط‌ها: جستجو انجام شده باشد

پس‌شرط‌ها: نتایج در رابط کاربری نمایش داده شده باشد

ورودی‌ها: لیست مقالات، شماره صفحه

خروجی‌ها: صفحه HTML شامل نتایج

داستان مرتبط: US-01

1.3-FR – فیلتر نتایج جستجو

شرح: کاربر باید بتواند نتایج جستجو را بر اساس تاریخ، محبوبیت یا دسته‌بندی فیلتر کند.

محرك: انتخاب فیلتر توسط کاربر

پیش‌شرط‌ها: نتایج جستجو تولید شده باشند

پس‌شرط‌ها: نتایج مطابق فیلتر نمایش داده شوند

ورودی‌ها: نوع فیلتر

خروجی‌ها: لیست فیلتر شده مقالات

داستان مرتبط: US-01, US-09

2.1-FR – باز کردن مقاله و نمایش محتوا

شرح: کاربر باید بتواند مقاله موردنظر را باز کرده و متن کامل، تصاویر و پیوندها را مشاهده کند.

محرك: کلیک کاربر روی مقاله

پیش‌شرط‌ها: مقاله موجود باشد

پس‌شرط‌ها: مقاله در رابط کاربری نمایش داده شود

ورودی‌ها: شناسه مقاله

خروجی‌ها: محتوای مقاله کامل

داستان مرتبط: US-02

2.2-FR – پیوندهای داخلی قابل کلیک

شرح: هر پیوند داخلی داخل مقاله باید قابل کلیک و باز کردن مقاله جدید باشد.

محرك: کاربر روی پیوند داخلی کلیک می‌کند

پیش‌شرط‌ها: مقاله شامل پیوند باشد

پس‌شرط‌ها: مقاله مقصد باز شود بدون از دست رفتن وضعیت مقاله فعلی

ورودی‌ها: پیوند داخلی

خروجی‌ها: نمایش مقاله مقصد

داستان مرتبط: US-02

3.1-FR – ایجاد مقاله جدید

شرح: نویسنده باید بتواند مقاله جدید شامل عنوان، متن، تصاویر و برجسبها ایجاد کند.

محرک: انتخاب گزینه "ایجاد مقاله جدید"

پیش شرطها: نویسنده وارد سیستم شده باشد

پس شرطها: مقاله ذخیره و منتشر شود

ورودیها: عنوان، متن، برجسبها، تصاویر، پیوندهای داخلی یا خارجی

خروجیها: مقاله جدید منتشر شده

داستان مرتبط: US-03

3.2-FR – پیش‌نمایش مقاله قبل از انتشار

شرح: نویسنده باید بتواند پیش‌نمایش مقاله را قبل از انتشار مشاهده کند.

محرک: انتخاب گزینه "پیش‌نمایش"

پیش شرطها: فرم مقاله پر شده باشد

پس شرطها: پیش‌نمایش دقیق مقاله نمایش داده شود

ورودیها: محتوای مقاله

خروجیها: صفحه پیش‌نمایش

داستان مرتبط: US-03

4.1-FR – ویرایش مقاله موجود

شرح: ویراستار باید بتواند محتوای مقاله موجود را ویرایش و ذخیره کند.

محرک: انتخاب گزینه "ویرایش"

پیش شرطها: مقاله موجود باشد

پس شرطها: تغییرات ذخیره و نسخه جدید ایجاد شود

ورودیها: محتوای ویرایش شده

خروجیها: نسخه جدید مقاله

داستان مرتبط: US-04

4.2-FR – مشاهده تفاوت نسخه‌ها

شرح: کاربر باید بتواند تغییرات بین نسخه‌های مختلف یک مقاله را مشاهده کند.

محرک: انتخاب گزینه "تفاوت نسخه‌ها"

پیش شرطها: چند نسخه از مقاله موجود باشد

پس شرطها: تغییرات بین نسخه‌ها نمایش داده شوند

ورودیها: دو نسخه انتخاب شده برای مقایسه

خروجی‌ها: تفاوت‌ها به صورت بصری

داستان مرتبط: US-05

5.1-FR – بازیابی نسخه قدیمی مقاله

شرح: کاربر باید بتواند نسخه قبلی مقاله را بازیابی کند.

محرک: انتخاب گزینه "بازگرداندن نسخه قدیمی"

پیش‌شرط‌ها: نسخه قدیمی موجود باشد

پس‌شرط‌ها: مقاله به نسخه قبلی بازگردد

ورودی‌ها: نسخه مورد نظر

خروجی‌ها: مقاله به حالت نسخه انتخابی

داستان مرتبط: US-05

6.1-FR – پیشنهاد مقالات مرتبط

شرح: سیستم باید مقالات مرتبط با مقاله در حال مطالعه را به کاربر پیشنهاد دهد.

محرک: باز کردن مقاله

پیش‌شرط‌ها: مقاله شامل برچسب یا محتوا باشد

پس‌شرط‌ها: لیست پیشنهادات نمایش داده شود

ورودی‌ها: مقاله جاری

خروجی‌ها: لیست مقالات مرتبط

داستان مرتبط: US-06

7.1-FR – تولید برچسب خودکار

شرح: سیستم باید پس از ذخیره مقاله، برچسب‌های مرتبط را استخراج کند.

محرک: ذخیره مقاله

پیش‌شرط‌ها: مقاله جدید ذخیره شده باشد

پس‌شرط‌ها: برچسب‌ها تولید و ذخیره شوند

ورودی‌ها: متن مقاله

خروجی‌ها: برچسب‌های پیشنهادی

داستان مرتبط: US-07

8.1-FR – گزارش مقاله نامناسب

شرح: کاربر باید بتواند مقاله را گزارش کند.

محرک: انتخاب گزینه "گزارش مقاله"

پیش‌شرط‌ها: مقاله موجود باشد

پس شرطها: گزارش ثبت و به مدیر اطلاع داده شود
ورودی‌ها: نوع مشکل، توضیحات
خروجی‌ها: تایید ثبت گزارش
داستان مرتبط: US-08

9.1-FR – اعلان تغییرات مقاله

شرح: سیستم باید تغییرات جدید مقالات مورد علاقه کاربر را اطلاع دهد.
محرك: ویرایش مقاله یا انتشار نسخه جدید
پیش شرطها: کاربر مقاله یا دسته را دنبال کرده باشد
پس شرطها: اعلان برای کاربر ارسال شود
ورودی‌ها: تغییرات مقاله
خروجی‌ها: اعلان در رابط کاربری
داستان مرتبط: US-10

10.1-FR – فیلتر مقالات بر اساس دسته‌بندی

شرح: کاربر باید بتواند مقالات را بر اساس دسته‌بندی مشاهده کند.
محرك: انتخاب دسته‌بندی
پیش شرطها: دسته‌بندی‌ها موجود باشند
پس شرطها: مقالات مرتبط نمایش داده شوند
ورودی‌ها: دسته‌بندی انتخاب شده
خروجی‌ها: لیست مقالات مرتبط بر اساس دسته‌بندی
داستان مرتبط: US-09

11.1-FR – حذف مقاله

شرح: سیستم باید امکان حذف یک مقاله توسط نویسنده یا مدیر محتوا را فراهم کند.
محرك: انتخاب گزینه "حذف مقاله"
پیش شرطها: کاربر باید مجوز حذف داشته باشد و مقاله موجود باشد
پس شرطها: مقاله حذف شده و در وضعیت "Archived / Soft Delete" ذخیره شود
ورودی‌ها: شناسه مقاله
خروجی‌ها: پیام موفقیت حذف و انتقال مقاله به بخش آرشیو
داستان مرتبط: US-04، US-05

12.1-FR – خلاصه‌سازی خودکار مقاله با LLM

شرح: سیستم باید بتواند محتوای مقالات را با استفاده از مدل زبانی (LLM) خلاصه کرده و نسخه خلاصه را نمایش دهد.
محرك: انتخاب گزینه "خلاصه‌سازی" توسط کاربر
پیش‌شرط‌ها: مقاله موجود باشد
پس‌شرط‌ها: خلاصه مقاله تولید و ذخیره یا نمایش داده شود
ورودی‌ها: متن مقاله یا شناسه مقاله
خروجی‌ها: خلاصه مقاله (Short Summary / Extended Summary)
داستان مرتبط: US-02

13.1-FR – دسته‌بندی خودکار مقاله بر اساس محتوای LLM

شرح: سیستم باید بتواند با تحلیل محتوای مقاله، دسته‌بندی مناسب را به صورت خودکار پیشنهاد دهد.
محرك: ذخیره یا ایجاد مقاله جدید
پیش‌شرط‌ها: متن مقاله وجود داشته باشد
پس‌شرط‌ها: دسته‌بندی پیشنهادی تولید شود
ورودی‌ها: متن مقاله
خروجی‌ها: یک یا چند دسته‌بندی پیشنهادی
داستان مرتبط: US-07

14.1-FR – جستجوی هوشمند داخل مقاله (Semantic In-Article Search)

شرح: کاربر باید بتواند داخل یک مقاله، با جستجوی معنایی، بخش‌های مرتبط را پیدا کند.
محرك: وارد کردن عبارت جستجو داخل بخش "Find"
پیش‌شرط‌ها: مقاله لود شده باشد
پس‌شرط‌ها: بخش‌های مرتبط هایلایت و لیست شوند
ورودی‌ها: عبارت جستجو
خروجی‌ها: بخش‌های یافت شده (Paragraph matches)
داستان مرتبط: US-02

15.1-FR – پیشنهاد هوشمند پیوندهای داخلی هنگام نوشتن مقاله

شرح: هنگام تایپ مقاله، سیستم باید پیوندهای داخلی مناسب را پیشنهاد دهد.
محرك: ورود متن توسط نویسنده
پیش‌شرط‌ها: مقالات دیگر در سیستم موجود باشند
پس‌شرط‌ها: پیشنهادات لینک به نویسنده نمایش داده شود
ورودی‌ها: متن در حال تایپ
خروجی‌ها: فهرست پیوندهای پیشنهادی
داستان مرتبط: US-03, US-04

نیازمندی‌های غیر عملکردی (Non-Functional Requirements)

1.1-NFR – کارایی (Performance)

شرح: سیستم باید قادر به پاسخ‌دهی سریع به درخواست‌های جست‌وجو باشد.
معیار ارزیابی: در شرایط ۱۰۰۰ کاربر همزمان، زمان پاسخ کمتر از ۲ ثانیه باشد.
سناریو: کاربر در اوج استفاده روزانه جست‌وجو می‌کند.
داستان مرتبط: US-01

1.2-NFR – کیفیت نتایج جست‌وجو (Quality)

شرح: نتایج جست‌وجو باید ارتباط بالایی با عبارت جست‌وجو داشته باشند.
معیار ارزیابی: حداقل ۸۰٪ از نتایج صفحه اول مرتبط باشند ($0.8 < \text{Precision}$).
سناریو: کاربر عبارت "تاریخ صفویه" را جست‌وجو می‌کند.
داستان مرتبط: US-01

2.1-NFR – امنیت (Security)

شرح: تنها کاربران مجاز باید بتوانند مقالات را ایجاد یا ویرایش کنند.
معیار ارزیابی: احراز هویت و مجوزدهی برای تمام عملیات نوشتن.
سناریو: کاربر ناشناس سعی در ویرایش مقاله می‌کند.
داستان مرتبط: US-03, US-04

3.1-NFR – نگهداری (Maintainability)

شرح: کد باید ماژولار و مستند باشد تا توسعه و تغییرات آسان باشد.
معیار ارزیابی: تمام توابع دارای Docstring و پوشش تست $\geq 80\%$.
سناریو: توسعه‌دهنده جدید قصد اضافه کردن ویژگی دارد.
داستان مرتبط: همه

3.2-NFR – قابلیت دسترسی (Accessibility)

شرح: رابط کاربری باید مطابق با نیازهای کاربران دارای محدودیت باشد.
معیار ارزیابی: تطابق با WCAG 2.1 سطح AA.

سناریو: کاربر دارای محدودیت بینایی از سایت استفاده می‌کند.
داستان مرتبط: همه

4.1-NFR – قابلیت اطمینان (Reliability)

شرح: سیستم باید پایدار بوده و مقالات ذخیره شده را از دست ندهد.
معیار ارزیابی: بدون از دست رفتن داده در ۹۹/۹٪ مواقع.
سناریو: نویسنده مقاله ایجاد می‌کند و سیستم در حال بروزرسانی است.
داستان مرتبط: US-03, US-04

4.2-NFR – امنیت داده‌ها (Security)

شرح: تمام داده‌های کاربران و تغییرات مقالات در ذخیره‌سازی و انتقال رمزگذاری شوند.
معیار ارزیابی: استفاده از 256-AES برای ذخیره‌سازی و HTTPS برای انتقال.
سناریو: کاربر در حال ایجاد یا ویرایش مقاله است.
داستان مرتبط: US-03, US-04

5.1-NFR – کارایی بارگذاری مقاله

شرح: زمان بارگذاری مقاله کمتر از ۱ ثانیه باشد.
معیار ارزیابی: آزمایش در ۵۰۰ درخواست همزمان.
سناریو: کاربر مقاله را باز می‌کند.
داستان مرتبط: US-02

6.1-NFR – جلوگیری از گزارش تکراری (Reliability)

شرح: سیستم باید از ثبت چندین گزارش یک مقاله توسط یک کاربر جلوگیری کند.
معیار ارزیابی: هر کاربر تنها یک بار می‌تواند گزارش ثبت کند.
سناریو: کاربر چندین بار روی "گزارش مقاله" کلیک می‌کند.
داستان مرتبط: US-08

7.1-NFR – کیفیت برچسب‌ها

شرح: برچسب‌های تولید شده باید با محتوای مقاله حداقل ۷۰٪ تطابق داشته باشند.
معیار ارزیابی: بررسی نمونه‌ای از ۱۰۰ مقاله تولیدی.
سناریو: مقاله جدید ایجاد شده و برچسب‌ها پیشنهاد می‌شوند.
داستان مرتبط: US-07

8.1-NFR – لاگ‌گذاری (Maintainability)

شرح: سیستم باید قابلیت لاگ‌گیری کامل از تغییرات و گزارش‌ها داشته باشد.
معیار ارزیابی: تمام فعالیت‌ها در فایل لاگ ثبت شوند.
سناریو: هر عملیات ایجاد، ویرایش یا گزارش انجام شود.
داستان مرتبط: همه

9.1-NFR – کنترل دسترسی برای بازیابی نسخه‌ها (Security)

شرح: امکان بازیابی نسخه‌های قدیمی باید فقط برای کاربران مجاز فراهم شود.
معیار ارزیابی: تست دسترسی کاربران غیرمجاز.
سناریو: کاربر عادی سعی در بازگرداندن نسخه قدیمی مقاله می‌کند.
داستان مرتبط: US-05

10.1-NFR – دسترسی‌پذیری اعلان‌ها (Availability)

شرح: اعلان‌ها و پیشنهادات باید برای دستگاه‌های موبایل و دسکتاپ قابل مشاهده باشند.
معیار ارزیابی: نمایش صحیح در رزولوشن‌های مختلف موبایل و دسکتاپ.
سناریو: کاربر با موبایل اعلان دریافت می‌کند.
داستان مرتبط: US-06, US-10

11.1-NFR – کنترل حذف (Access Control)

شرح: عملیات حذف مقاله باید فقط برای نویسنده مقاله یا مدیر محتوا فعال باشد.
معیار ارزیابی: کاربران غیرمجاز حتی با دسترسی مستقیم API نتوانند حذف انجام دهند.
سناریو: کاربر عادی تلاش می‌کند مقاله‌ای را حذف کند.
داستان مرتبط: US-04, US-05

12.1-NFR – عملکرد LLM برای خلاصه‌سازی (LLM Performance)

شرح: تولید خلاصه مقاله توسط LLM باید در زمان مناسب انجام شود.
معیار ارزیابی: زمان تولید خلاصه کمتر از ۳ ثانیه برای مقاله ~۲۰۰۰ کلمه‌ای.
سناریو: کاربر گزینه "خلاصه‌سازی" را انتخاب می‌کند.
داستان مرتبط: US-02

13.1-NFR – کیفیت خروجی (LLM Output Quality)

شرح: خلاصه مقاله باید حداقل ۷۵٪ همپوشانی معنایی با متن اصلی داشته باشد و فاقد اطلاعات غلط باشد.
معیار ارزیابی: بررسی دستی ۵۰ خلاصه تولیدی، $\text{Semantic Similarity} \geq 0.75$.
سناریو: نویسنده خلاصه تولید شده را بررسی می‌کند.
داستان مرتبط: US-02

14.1-NFR – تحمل خرابی وابستگی‌های خارجی (Resilience)

شرح: سیستم باید حتی در قطع ارتباط با سرویس LLM (خارجی) عملکرد پایه خود را حفظ کند.
معیار ارزیابی: در صورت Down بودن LLM، سیستم به‌جای خطا، پیام مناسب نمایش دهد و عملکردهای پایه ادامه یابند.
سناریو: کاربر خلاصه می‌خواهد ولی LLM در دسترس نیست.
داستان مرتبط: US-03, US-04, US-02

اعتبارسنجی نیازمندی ها

بخش اول – نیازمندی های عملکردی (FR)

1.1-FR – پردازش جستوجوی معنایی

Who: کاربران جستوجوکننده (عمومی، پژوهشگر، سرویس های داخلی/خارجی)
What: پیاده سازی موتور **Semantic Search** برای یافتن و رتبه بندی مقالات بر مبنای درک معنایی عبارت جستوجو.
Why: افزایش کیفیت نتایج، افزایش ارزش اطلاعاتی سرویس و بهبود تجربه کاربری.
When:

• اولویت: High

- پیشنهاد: **1 Sprint** – هسته اولیه سیستم؛ پیاده سازی پایهی جستوجوی معنایی با مدل سبک برای MVP.
Where: در میکروسرویس جستوجو (**Search Service**) و لایه API.
How: استفاده از مدل های embedding (مثلاً **transformer-based** یا **SBERT**)، محاسبه **similarity**، ایندکس سازی متن (مثلاً **Elasticsearch** با **dense vector** یا سرویس **embedding**) و رتبه بندی نتایج.
How Much: منابع محاسباتی متوسط تا زیاد (**GPU** برای آموزش/توسعه احتمالی)، زمان توسعه ۲-۴ هفته برای MVP با تیم ۲-۳ نفره، هزینه عملیاتی متوسط.

1.2-FR – نمایش نتایج جستوجو

Who: همه کاربران جستوجوکننده.
What: نمایش نتایج به صورت صفحه بندی شده همراه عنوان، خلاصه و امتیاز.
Why: بهبود خوانایی و کارایی مرور نتایج.
When: اولویت: High – پیشنهاد: **1 Sprint** (همزمان با 1.1-FR، بخش **Frontend**).
Where: **Frontend (Search Results Page)** و API مربوطه.
How: ایجاد **endpoint** صفحه بندی، طراحی کامپوننت های UI برای نتایج، مدیریت ناوبری صفحات.
How Much: زمان 1-1.5 هفته؛ منابع: 1 فرانت اند، 1 بک اند؛ هزینه کم؛ پیچیدگی: کم.

1.3-FR – فیلتر نتایج جستجو

Who: کاربران پیشرفته و عمومی.
What: فیلتر نتایج براساس تاریخ، محبوبیت و دسته‌بندی.
Why: کاهش زمان جستجو و دسترسی سریع‌تر به نتایج مرتبط.
When: اولویت: Medium-High – پیشنهاد: 1 Sprint (پیاده‌سازی اولیه در 1 Sprint، ارتقاء در Sprint 2).
Where: Search Service (query params) و UI فیلترها.
How: افزودن پارامترهای query، نگاشت فیلترها به شاخص‌های ایندکس شده، و UI کنترل فیلتر.
How Much: زمان 1.5-2 هفته؛ منابع: 1 فرانت‌اند، 1 بک‌اند؛ هزینه: کم تا متوسط؛ پیچیدگی: متوسط.

2.1-FR – باز کردن مقاله و نمایش محتوا

Who: تمام کاربران خواننده.
What: نمایش کامل مقاله شامل متن، تصاویر و پیوندها.
Why: هدف اصلی سرویس – دسترسی به محتوای کامل.
When: اولویت: High – پیشنهاد: Sprint 1.
Where: Article View Page (Frontend) و Article Retrieval API.
How: API واکنشی مقاله بر اساس شناسه، CDN برای تصاویر، رندر سمت مشتری.
How Much: زمان 1-2 هفته؛ منابع: 1 فرانت‌اند، 1 بک‌اند؛ هزینه: کم؛ پیچیدگی: متوسط.

2.2-FR – پیوندهای داخلی قابل کلیک

Who: خوانندگان مقالات.
What: لینک‌های داخلی داخل مقاله باید قابل کلیک و باز کردن مقاله مقصد بدون از دست رفتن وضعیت باشند.
Why: بهبود پیمایش و کشف محتوای مرتبط.
When: اولویت: High – پیشنهاد: 1 Sprint یا بلافاصله پس از FR-2.1.
Where: Rendering layer و routing در Frontend؛ لینک‌ها در محتوا ذخیره شده.
How: لینک‌دهی به شناسه‌ها، استفاده از تاریخچه مرورگر (pushState) برای حفظ وضعیت.
How Much: زمان 3-5 روز؛ منابع: 1 فرانت‌اند؛ هزینه: بسیار کم؛ پیچیدگی: کم.

3.1-FR – ایجاد مقاله جدید

Who: نویسندگان و کاربران دارای مجوز نوشتن.
What: قابلیت ایجاد مقاله شامل عنوان، متن، تصاویر، برچسب‌ها و پیوندها.
Why: افزایش محتوای سامانه و به‌روزرسانی پایگاه دانش.

When: اولویت: High — پیشنهاد: 2 Sprint (پس از آماده شدن خواندن و جست‌وجو).
Where: Authoring UI و Article Creation API، پایگاه داده.
How: فرم ویرایشگر (WYSIWYG/Markdown)، آپلود تصویر، نسخه‌بندی اولیه.
How Much: زمان 2-3 هفته؛ منابع: 1 فرانت‌اند (Editor)، 1 بک‌اند؛ هزینه: متوسط؛ پیچیدگی: متوسط-زیاد.

3.2-FR — پیش‌نمایش مقاله

Who: نویسندگان.
What: پیش‌نمایش مقاله قبل از انتشار بدون ایجاد نسخه رسمی.
Why: جلوگیری از خطاها و اطمینان از قالب‌بندی صحیح.
When: اولویت: Medium — پیشنهاد: 2 Sprint (همزمان با FR-3.1).
Where: Editor UI (Preview mode).
How: رندر محتوای فرم در حالت client-side یا با render endpoint بدون ذخیره.
How Much: زمان 3-6 روز؛ منابع: 1 فرانت‌اند؛ هزینه: کم؛ پیچیدگی: کم.

4.1-FR — ویرایش مقاله موجود

Who: ویراستاران و نویسندگان مجاز.
What: امکان ویرایش محتوای مقاله و ذخیره به‌صورت نسخه جدید.
Why: نگهداری کیفیت محتوا و رفع اشکال/بهرورسانی.
When: اولویت: High — پیشنهاد: Sprint 2.
Where: Editor UI، Versioning Service، Article API.
How: پیاده‌سازی قابلیت ویرایش در Editor و ذخیره نسخه جدید با metadata نسخه.
How Much: زمان 1-2 هفته؛ منابع: 1 بک‌اند، 1 فرانت‌اند؛ هزینه: متوسط؛ پیچیدگی: متوسط.

4.2-FR — مشاهده تفاوت نسخه‌ها

Who: ویراستاران، نویسندگان، مدیران.
What: نمایش بصری تفاوت میان دو نسخه (متن و تغییرات ساختاری).
Why: شفافیت تغییرات و کنترل کیفیت محتوا.
When: اولویت: Medium — پیشنهاد: Sprint 3.
Where: Version Comparison Page و Versioning Service.
How: محاسبه تفاوت با ابزارهای متن (مثلاً diff-match-patch) و رندر بصری.
How Much: زمان 1-1.5 هفته؛ منابع: 1 بک‌اند، 1 فرانت‌اند؛ هزینه: متوسط؛ پیچیدگی: متوسط.

5.1-FR — بازیابی نسخه قدیمی مقاله

Who: ویراستار/مدیر (کاربران مجاز).
What: بازگرداندن مقاله به یک نسخه قبلی و ایجاد نسخه جدید جایگزین.
Why: توانایی برگرداندن تغییرات اشتباه یا حذف ناخواسته.
When: اولویت: Medium-High — پیشنهاد: Sprint 3.
Where: Versioning Service و Admin UI.
How: ثبت کپی نسخه قدیمی به عنوان نسخه جدید، ثبت تاریخچه بازگردانی.
How Much: زمان 4-7 روز؛ منابع: 1 بک‌اند؛ هزینه: کم؛ پیچیدگی: کم-متوسط.

6.1-FR — پیشنهاد مقالات مرتبط

Who: خوانندگان مقاله.
What: نمایش لیست مقالات مرتبط بر اساس محتوا/برچسب/شباهت معنایی.
Why: افزایش تعامل کاربر و کشف محتوا.
When: اولویت: Medium — پیشنهاد: Sprint 3 (پس از پایه‌گذاری Tag/Embedding).
Where: Recommendation Engine یا Search Service؛ بخش پیشنهادات.
How: محاسبه (similarity (content/tag-based)، احتمالاً استفاده از مدل‌های embedding یا گراف.
How Much: زمان 2-3 هفته؛ منابع: 1 بک‌اند، 1 متخصص similarity؛ هزینه: متوسط؛ پیچیدگی: متوسط-زیاد.

7.1-FR — تولید برچسب خودکار

Who: نویسندگان (پس از ذخیره مقاله) و سیستم.
What: استخراج و پیشنهاد برچسب‌ها به صورت خودکار با استفاده از NLP.
Why: سهولت برچسب‌گذاری و بهبود دسته‌بندی و جست‌وجو.
When: اولویت: Medium — پیشنهاد: Sprint 3.
Where: Tag Generator Service و Article Creation flow.
How: کلیدواژه‌کاو، TF-IDF، NER، embedding برای استخراج برچسب‌های کاندید.
How Much: زمان 1.5-2.5 هفته؛ منابع: 1 NLP/ML، 1 بک‌اند؛ هزینه: متوسط؛ پیچیدگی: متوسط-زیاد.

8.1-FR — گزارش مقاله نامناسب

Who: تمام کاربران.
What: قابلیت ثبت گزارش (نوع مشکل، توضیحات) و ارسال به مدیر/صف بررسی.

Why: حفظ کیفیت محتوا و امکان بررسی تخلفات.

When: اولویت: High – پیشنهاد: Sprint 2.

Where: Report Service و Admin Dashboard.

How: فرم گزارش، ذخیره گزارش، نوتیفیکیشن به مدیر، امکان مدیریت گزارش‌ها.

How Much: زمان 3–5 روز؛ منابع: 1 بک‌اند، 1 فرانت‌اند؛ هزینه: کم؛ پیچیدگی: کم.

9.1-FR – اعلان تغییرات مقاله

Who: کاربرانی که مقاله/دسته را دنبال کرده‌اند.

What: ارسال اعلان هنگام انتشار یا ویرایش مقاله.

Why: افزایش تعامل و نگهداشت کاربران.

When: اولویت: Medium – پیشنهاد: Sprint 3.

Where: Notification Service و UI برای subscription.

How: event-driven triggers → push/in-app/email notification.

How Much: زمان 1–2 هفته؛ منابع: 1 بک‌اند، 1 فرانت‌اند؛ هزینه: کم تا متوسط؛ پیچیدگی: متوسط.

10.1-FR – فیلتر مقالات بر اساس دسته‌بندی

Who: تمام کاربران.

What: امکان مشاهده مقالات بر پایه دسته‌بندی انتخابی.

Why: دسترسی سریع به موضوعات مرتبط.

When: اولویت: Medium – پیشنهاد: Sprint 1–2.

Where: Category Browser و API.

How: تفکیک مقالات بر اساس دسته‌بندی، ذخیره‌سازی و نمایش در UI.

How Much: زمان 4–6 روز؛ منابع: 1 بک‌اند، 1 فرانت‌اند؛ هزینه: کم؛ پیچیدگی: کم.

11.1-FR – حذف مقاله

Who: نویسنده مقاله یا مدیر محتوا (کاربران دارای سطح دسترسی حذف).

What: امکان حذف مقاله و انتقال آن به حالت Archived/Soft Delete.

Why: مدیریت محتوا، حذف مطالب نامعتبر یا غیرقابل استفاده و کنترل نسخه‌ها.

When: پیشنهاد: Sprint 2 یا Sprint 3 (پس از تکمیل ایجاد/ویرایش و تاریخچه).

Where: Article Service (Back-end) و بخش مدیریت مقاله در UI.

How: علامت‌گذاری مقاله به "Deleted/Archived"، جلوگیری از نمایش عمومی، ثبت رویداد حذف در تاریخچه.

How Much: زمان 4–6 روز؛ منابع: 1 بک‌اند، 1 فرانت‌اند؛ هزینه: کم؛ پیچیدگی: Small–Medium.

12.1-FR – خلاصه‌سازی خودکار مقاله با LLM

Who: تمام کاربران مطالعه‌کننده مقاله.
What: تولید خلاصه کوتاه یا بلند از مقاله با استفاده از مدل‌های LLM.
Why: سرعت‌بخشیدن به مطالعه و ارائه درک سریع از محتوا.
When: پیشنهاد: 3 Sprint یا 4 Sprint (پس از زیرساخت LLM).
Where: LLM Integration Service و بخش نمایش مقاله (UI).
How: فراخوانی API مدل زبانی، پردازش متن و تولید/نمایش یا ذخیره Summary.
How Much: زمان 2-4 هفته؛ منابع: 1 NLP/ML، 1 بک‌اند؛ هزینه: متوسط تا زیاد؛ پیچیدگی: Large.

13.1-FR – دسته‌بندی خودکار مقاله با LLM

Who: نویسندگان و سیستم هنگام ایجاد/ذخیره مقاله.
What: پیشنهاد دسته‌بندی مقاله بر اساس تحلیل محتوای LLM.
Why: بهبود مدیریت موضوعی و کاهش نیاز به دسته‌بندی دستی.
When: پیشنهاد: 3 Sprint یا 4 Sprint.
Where: Article Processing Service / LLM Service.
How: تحلیل محتوا و استخراج موضوعات سپس پیشنهاد دسته‌ها.
How Much: زمان 1.5-3 هفته؛ منابع: 1 NLP/ML، 1 بک‌اند؛ هزینه: متوسط؛ پیچیدگی: Medium-Large.

14.1-FR – جستجوی هوشمند داخل مقاله (Semantic In-Article Search)

Who: کاربران مطالعه‌کننده مقاله.
What: امکان جستجوی معنایی در متن مقاله و نمایش بخش‌های مرتبط.
Why: افزایش سرعت دسترسی به بخش‌های موردنیاز در مقالات طولانی.
When: پیشنهاد: 4 Sprint.
Where: Frontend Article Reader + In-Article Semantic Search Engine.
How: embedding جملات/پاراگراف‌ها، محاسبه semantic similarity، هایلایت بخش‌های مرتبط.
How Much: زمان 2-3 هفته؛ منابع: 1 NLP/ML، 1 فرانت‌اند؛ هزینه: متوسط؛ پیچیدگی: Medium-Large.

15.1-FR – پیشنهاد هوشمند پیوندهای داخلی هنگام نوشتن مقاله

Who: نویسندگان هنگام تایپ محتوا.
What: پیشنهاد مقالات مرتبط برای ایجاد لینک داخلی در لحظه نوشتن.

Why: غنی‌سازی ساختار مقاله و بهبود پیوند داخلی.
When: پیشنهاد: Sprint 4.
Where: Editor UI و Link Suggestion Service.
How: تحلیل جملات در حال تایپ، یافتن مقالات مرتبط و نمایش پیشنهاد.
How Much: زمان 2-3 هفته؛ منابع: 1 فرانت‌اند، 1 بک‌اند، 1 NLP؛ هزینه: متوسط؛ پیچیدگی: Medium-Large.

بخش دوم – نیازمندی‌های غیر عملیاتی (NFR)

1.1-NFR – کارایی (Performance)

Who: تمام کاربران سیستم.
What: سیستم باید در شرایط ۱۰۰۰ کاربر همزمان، زمان پاسخ جست‌وجو را $2 >$ ثانیه نگه دارد.
Why: حفظ تجربه کاربری و کاهش نرخ ترک.
When: اولویت: High – اعمال از Sprint 1.
Where: Search Service، API Gateway، زیرساخت شبکه و DB.
How: بهینه‌سازی ایندکس‌ها، (caching (Redis، توزیع بار، auto-scaling و مانیتورینگ (Prometheus/Grafana).
How Much: زمان 2-4 هفته تست و بهینه‌سازی؛ منابع: 1 بک‌اند، 1 DevOps؛ هزینه: متوسط تا زیاد؛ پیچیدگی: Large.

1.2-NFR – کیفیت (Precision)

Who: کاربران جست‌وجوکننده.
What: دقت نتایج صفحه اول $0.8 \leq$ (Precision > 0.8).
Why: اطمینان از مرتبط بودن نتایج.
When: اولویت: High – پیشنهاد Sprint 1-3.
Where: Semantic Search pipeline، evaluation datasets.
How: fine-tune مدل‌ها، ارزیابی IR، بازخورد کاربران.
How Much: زمان 3-5 هفته؛ منابع: 1 NLP/ML؛ هزینه: متوسط تا زیاد؛ پیچیدگی: Large.

2.1-NFR – امنیت (Authorization)

Who: نویسندگان، ویراستاران، مدیران.
What: فقط کاربران مجاز بتوانند مقالات را ایجاد/ویرایش کنند.
Why: حفاظت از محتوا و جلوگیری از دسترسی غیرمجاز.
When: اولویت: High – پیشنهاد Sprint 1.
Where: Auth Service، UI، API، RBAC در سرویس‌ها.
How: JWT/OAuth2، نقش‌ها و سطوح دسترسی، کنترل‌های سمت سرور.
How Much: زمان 1-2 هفته؛ منابع: 1 بک‌اند، Security Engineer 1 پاره‌وقت؛ هزینه: متوسط؛ پیچیدگی: متوسط-زیاد.

3.1-NFR – نگهداری (Maintainability)

Who: تیم توسعه و توسعه‌دهندگان جدید.
What: کد ماژولار و مستند با Docstring و پوشش تست $\geq 80\%$.
Why: تسهیل توسعه و کاهش هزینه نگهداری.
When: اولویت: High – رعایت مستمر از Sprint 1.
Where: کلیه مخازن کد.
How: قوانین کدنویسی، CI، code review، مستندسازی خودکار.
How Much: زمان 10-20٪ از هر اسپرینت؛ همه اعضای تیم؛ هزینه: کم اما مداوم.

3.2-NFR – قابلیت دسترسی (Accessibility)

Who: کاربران دارای نیازهای ویژه.
What: رعایت WCAG 2.1 سطح AA در UI.
Why: دسترس‌پذیر کردن سرویس برای همه کاربران.
When: اولویت: Medium-High – Sprint 1-2.
Where: Frontend components, templates, CSS, ARIA attributes.
How: semantic HTML، ARIA roles، تست با ابزارهای accessibility.
How Much: زمان 1-2 هفته بازبینی UI؛ منابع: 1 فرانت‌اند؛ هزینه: کم؛ پیچیدگی: متوسط.

4.1-NFR – قابلیت اطمینان (Reliability)

Who: نویسندگان و کاربران.
What: عدم از دست رفتن مقالات؛ پایداری بالا (مثلاً 99.9% uptime).
Why: حفظ محتوای ارزشمند و اعتماد کاربران.

When: اولویت: High — طراحی backup/replication از ابتدا.
Where: DB ،storage ،versioning system.
How: replication ،backup منظم ،transactional writes ،monitoring.
How Much: زمان 2-3 هفته طراحی + پیکر بندی؛ منابع: 1 بک اند، 1 DevOps؛ هزینه: متوسط تا زیاد؛ پیچیدگی: Large.

4.2-NFR — امنیت (Encryption)

Who: همه کاربران و داده ها.
What: رمزگذاری در انتقال (HTTPS) و در ذخیره سازی (AES-256).
Why: حفظ محرمانگی و الزامات قانونی.
When: اولویت: High — فعال از روز اول.
Where: Transport layer (TLS) ،DB encryption at rest.
How: HTTPS/TLS ،KMS برای کلیدها، رمزگذاری در storage.
How Much: زمان 4-7 روز؛ منابع: 1 بک اند، 1 DevOps؛ هزینه: کم تا متوسط؛ پیچیدگی: کم-متوسط.

5.1-NFR — کارایی بارگذاری (Load time)

Who: خوانندگان مقاله.
What: زمان بارگذاری مقاله > ۱ ثانیه در ۵۰۰ درخواست همزمان.
Why: تجربه کاربری روان.
When: اولویت: High — Sprint 1-2 بهینه سازی.
Where: Frontend, CDN, backend APIs, DB.
How: CDN استاتیک، lazy loading تصاویر، caching.
How Much: زمان 2-3 هفته؛ منابع: 1 بک اند، 1 فرانت اند، 1 DevOps؛ هزینه: متوسط؛ پیچیدگی: Large.

6.1-NFR — جلوگیری از گزارش تکراری (Reports deduplication)

Who: تمام کاربران.
What: هر کاربر تنها یک بار بتواند گزارش یک مقاله را ثبت کند.
Why: جلوگیری از اسپم و تکرار داده ها.
When: اولویت: Medium — پیشنهاد Sprint 2.
Where: Report Service ،DB constraints یا logic.
How: چک در سمت سرور قبل از ثبت / unique constraint یا business rule.
How Much: زمان 2-3 روز؛ منابع: 1 بک اند؛ هزینه: بسیار کم؛ پیچیدگی: کم.

7.1-NFR – کیفیت برچسب‌ها (Tag quality)

Who: نویسندگان و کاربران.
What: برچسب‌ها $\leq 70\%$ تطابق با محتوای مقاله.
Why: کاهش نیاز به ویرایش دستی و بهبود جست‌وجو.
When: اولویت: Medium – پیشنهاد Sprint 3.
Where: Tag Generator Service.
How: ارزیابی نمونه‌ای، بازخورد انسانی، الگوریتم‌های NER/keyword extraction.
How Much: زمان 1.5–3 هفته؛ منابع: 1 NLP/ML؛ هزینه: متوسط؛ پیچیدگی: متوسط–زیاد.

8.1-NFR – لاگ‌گذاری (Logging)

Who: تیم پشتیبانی و توسعه.
What: لاگ‌گیری کامل از فعالیت‌های ایجاد، ویرایش و گزارش‌ها.
Why: ردیابی خطا، تحلیل رفتار و حسابرسی.
When: اولویت: High – فعال از Sprint 1.
Where: Middlewares، Audit Logs، Log Aggregation.
How: structured logging، retention policy، ارسال لاگ به سرویس مرکزی.
How Much: زمان 1–2 هفته؛ منابع: 1 بک‌اند، 1 DevOps؛ هزینه: متوسط؛ پیچیدگی: متوسط.

9.1-NFR – کنترل بازیابی نسخه‌ها (Restore access control)

Who: ویراستاران/مدیران مجاز.
What: فقط کاربران مجاز توانایی بازیابی نسخه‌های قدیمی را داشته باشند.
Why: جلوگیری از دستکاری غیرمجاز تاریخچه محتوا.
When: اولویت: High – پیشنهاد Sprint 3.
Where: Versioning Service، Authorization checks.
How: RBAC یا policy-based access control.
How Much: زمان 3–5 روز؛ منابع: 1 بک‌اند؛ هزینه: کم؛ پیچیدگی: کم.

10.1-NFR – نمایش اعلان‌ها (Notifications display)

Who: کاربران موبایل و دسکتاپ.
What: اعلان‌ها و پیشنهادات باید در رزولوشن‌های مختلف قابل مشاهده باشند.

Why: تجربه یکنواخت در پلتفرم‌ها.
When: اولویت: Medium – پیشنهاد Sprint 3.
Where: Frontend responsive layouts، Notification UI.
How: طراحی responsive، تست در رزولوشن‌های مختلف، adaptive components.
How Much: زمان 3–6 روز؛ منابع: 1 فرانت‌اند؛ هزینه: کم؛ پیچیدگی: کم–متوسط.

11.1-NFR – امنیت عملیات حذف مقاله

Who: نویسنده، مدیر و کاربران دارای نقش معتبر.
What: محدودسازی حذف مقاله فقط برای افراد مجاز و جلوگیری از حذف از طریق API مستقیم.
Why: جلوگیری از حذف غیرمجاز محتوا.
When: پیشنهاد Sprint 2 (همزمان با حذف مقاله).
Where: Authorization Layer + Delete Endpoint.
How: RBAC، اعتبارسنجی نقش و مالکیت، لاگ فعالیت.
How Much: زمان 3–5 روز؛ منابع: 1 بک‌اند؛ هزینه: کم؛ پیچیدگی: Small–Medium.

12.1-NFR – کارایی LLM در خلاصه‌سازی

Who: کاربران استفاده‌کننده از خلاصه‌سازی.
What: زمان تولید خلاصه ≥ 3 ثانیه برای مقاله ~ 2000 کلمه.
Why: حفظ تجربه کاربری روان.
When: پیشنهاد Sprint 3–4.
Where: LLM Service و API مرتبط.
How: cache، نتایج، مدل سبک‌تر، موازی‌سازی.
How Much: زمان 1–2 هفته بهینه‌سازی؛ منابع: 1 ML/NLP، 1 بک‌اند؛ هزینه: متوسط؛ پیچیدگی: Medium–Large.

13.1-NFR – کیفیت خروجی LLM

Who: نویسندگان و کاربران خلاصه.
What: خلاصه با $\text{Semantic Similarity} \leq 0.75$ و بدون hallucination.
Why: جلوگیری از خلاصه‌های نادرست و حفظ اعتبار.
When: پیشنهاد Sprint 4.
Where: LLM Evaluation Pipeline.
How: نمونه‌برداری دستی، تست خودکار، مدل‌های تشخیص خطا.
How Much: زمان 2–3 هفته؛ منابع: 1 NLP/ML؛ هزینه: متوسط تا زیاد؛ پیچیدگی: Large.

14.1-NFR – پایداری در قطع سرویس LLM

Who: تمام کاربران.

What: اگر سرویس LLM در دسترس نبود، سیستم پیام مناسب نمایش دهد و عملکرد پایه حفظ شود.

Why: جلوگیری از اختلال تجربه کاربری.

When: پیشنهاد Sprint 4.

Where: Fallback Layer در UI و LLM Service.

How: timeout، fallback handler، پیام کاربر پسند، ثبت لاگ.

How Much: زمان 1-1.5 هفته؛ منابع: 1 بک‌اند، 1 DevOps؛ هزینه: کم تا متوسط؛ پیچیدگی: Medium.

اطلاعات جمع‌آوری شده از مطالعات موردی

۱. معرفی نمونه واقعی مورد بررسی

برای انجام یک مطالعه موردی مفید، نمونه‌ای انتخاب می‌کنیم که از نظر دامنه، رفتار کاربر، و ساختار به Wiki Service ما شبیه باشد.

نمونه انتخاب‌شده:

- Wikipedia
- TripAdvisor (برای تعاملات گردشگری و کامنت/علاقه‌مندی) : <https://www.tripadvisor.com/>

ترکیب بررسی Wikipedia و TripAdvisor تصویر نسبتاً کاملی از رفتار یک کاربر گردشگر ارائه می‌دهد و الگوهای عملی مفیدی برای طراحی Wiki Service فراهم می‌کند.

۲. جریان واقعی رفتار کاربر در نمونه مورد بررسی

سناریوی واقعی: گردشگر در حال تحقیق درباره "اصفهان"

مرحله ۱: ورود کاربر به سایت

- کاربر از طریق گوگل عبارت "Isfahan history" یا "جاهای دیدنی اصفهان" را جست‌وجو می‌کند.
- لینک ویکی‌پدیا یا وبسایت‌های گردشگری در نتایج دیده می‌شود.
- کاربر وارد صفحه می‌شود.

الگوهای استخراج‌شده:

- ۹۰٪ کاربران از طریق **Search** وارد **Wiki** می‌شوند، نه از طریق **Home Page**.
- بنابراین **Semantic Search** مهم‌ترین نقطه ورودی است.

مرحله ۲: مشاهده مقاله اصلی

در ویکی‌پدیا خلاصه اولیه در بالای صفحه، تصویر شاخص، جدول اطلاعات پایه و لینک‌های داخلی فراوان در وسط متن دارد.

الگوهای استخراج‌شده:

- کاربران اسکن می‌کنند، نه اینکه همه متن را بخوانند.
- خلاصه‌ی ۲-۳ جمله‌ای مهم‌ترین بخش مقاله است.
- لینک‌های داخلی نقش مسیرهای یادگیری را دارند.

مرحله ۳: پیمایش پیوندهای داخلی

الگوی رفتار کاربران در ویکی‌پدیا:

- کاربران بعد از ورود به مقاله اول، حداقل ۲/۲ کلیک دیگر روی لینک‌های داخلی دارند.
- کاربران عمیق‌خوان، گاهی تا ۷-۸ کلیک متوالی انجام می‌دهند.

الگوهای استخراج‌شده:

- ساخت شبکه دانش (**Knowledge Graph**) تجربه کاربری را چند برابر بهتر می‌کند.
- پیوندهای داخلی باید بهینه‌شده، دقیق و مفید باشند.
- موتور جست‌وجوی معنایی باید این لینک‌ها را تقویت کند، نه اینکه فقط یک جست‌وجوی ساده داشته باشد.

مرحله ۴: جست‌وجوی عمیق‌تر

در **Wikipedia** یا **TripAdvisor**:

- کاربر عبارتی مانند "**Safavid architecture**" یا "جاذبه‌های اصفهان" را جست‌وجو می‌کند.
- سیستم پیشنهادی مرتبط نمایش می‌دهد:
 - معماری صفوی
 - میدان نقش جهان
 - مسجد شیخ لطف‌الله

بینش کاربردی:

- نیت کاربر (Search Intent) مهم است.
- جستوجوی Semantic در اینجا ارزش اصلی سرویس ماست.

مرحله ۵: ذخیره، علاقه‌مندی، یا ساخت برنامه سفر (مثل TripAdvisor)

- کاربر مقاله یا مکان موردنظر خود را ذخیره می‌کند.
- کاربر می‌تواند نظر خود را ثبت کند.
- گاهی کاربر به بخش "Plan Trip" منتقل می‌شود.

الگوهای مهم:

- ذخیره‌سازی محتوا یکی از نیازمندی‌های اساسی کاربران است.
- قابلیت علاقه‌مندی و ذخیره‌سازی نیازمند احراز هویت کاربران (User Authentication)، سیستم ذخیره علاقه‌مندی‌ها (Favorite Storage) و خدمات اطلاع‌رسانی (Notification Service) است.

مرحله ۶: ثبت بازخورد و مشارکت

در Wikipedia:

- کاربران حرفه‌ای محتوا را ویرایش می‌کنند.
- کاربران عادی پیشنهاد تغییر می‌دهند.
- تاریخچه نسخه الزامی است.

بینش مهم:

- بدون Versioning کامل، اعتماد به محتوا از بین می‌رود.

۳. تحلیل مشکلات، خطاهای رایج و نقاط ضعف سرویس‌های مشابه

با تحلیل سرویس‌های واقعی، چند خطای رایج کشف شد:

خطای ۱: جستوجوی ضعیف و وابسته به کلمات کلیدی

- Wikipedia فارسی جستوجوی معنایی ندارد.
- کاربران بسیاری نمی‌توانند مقاله مناسب پیدا کنند.

نتیجه:

- **Semantic Search** باید نیازمندی سطح بالا (High Priority) باشد.

خطای ۲: لینک‌های داخلی زیاد اما غیر هدفمند

- در بسیاری صفحات:
 - تعداد زیادی لینک وجود دارد
 - برخی بی‌ربط هستند
 - برخی مهم‌ها لینک نشده‌اند

نتیجه:

- پیشنهاد خودکار لینک‌های داخلی (FR-15.1) واقعاً ارزشمند است.

خطای ۳: نسخه‌بندی ناقص در سایت‌های گردشگری

- بسیاری سایت‌ها نسخه‌بندی ندارند. محتوا اشتباه شود، اصلاح آن سخت است.

نتیجه:

- **Versioning Service** ضرورت دارد.
- **Versioning** یعنی تاریخچه کامل تغییرات، قابلیت بازگشت به نسخه‌های قبلی و اطمینان از صحت محتوا ولی **WYSIWYG Editor** فقط ظاهر و سهولت ویرایش متن را فراهم می‌کند، بدون تضمین صحت یا کنترل نسخه.

خطای ۴: تجربه کاربری غیر شخصی‌سازی شده

- در **Wikipedia** فارسی و سایت‌های ایرانی، محتوایی مطابق علایق کاربر نمایش نمی‌شود.

نتیجه:

- **Recommendation Service** حتماً باید باشد.

خطای ۵: کندی بارگذاری و حجم زیاد تصاویر

- به‌خصوص در پلتفرم‌های ایرانی:
 - تصاویر بزرگ‌اند

- CDN استفاده نمی‌شود
- زمان لود صفحه بیشتر از ۴ ثانیه است

نتیجه:

- (NFR-5.2 Load Time <1s) بسیار حیاتی است.

۴. استخراج نیازمندی‌ها از Case Study

بر اساس رفتار واقعی کاربران و تحلیل نمونه‌ها، نیازمندی‌های زیر مستقیماً نتیجه Case Study هستند:

نیازمندی‌های عملیاتی (FR)

1. Semantic Search پیشرفته (الزامی و محوری)
2. لینک‌دهی داخلی هوشمند
3. نمایش خلاصه مقاله در ابتدای صفحه
4. صفحه مقاله با ساختار منظم، بخش‌بخش و قابل اسکن
5. ویرایش مقاله با versioning دقیق
6. ثبت و نمایش بازخورد کاربران
7. ذخیره‌سازی مقالات مورد علاقه
8. گالری تصاویر با CDN

نیازمندی‌های غیر عملیاتی (NFR)

1. Precision جست‌وجو < 0.8
2. Load time < 1s برای 500 درخواست همزمان
3. بازدهی Semantic Search > 2 ثانیه برای 1000 کاربر همزمان
4. Reliability: نسخه‌بندی + backup
5. Security: دسترسی نقش‌محور (RBAC)
6. Accessibility مطابق WCAG 2.1
7. بدون تولید محتوای اشتباه یا ساختگی توسط مدل زبانی
8. Error Handling مناسب

۵. الگوهای طلایی استخراج‌شده

1. یک مقاله خوب همیشه یک خلاصه کوتاه در ابتدای صفحه دارد.
2. ساختار مقاله باید قطعه‌قطعه باشد (history, geography, culture).

3. پیوندهای داخلی مهمتر از طول مقاله هستند.
4. **Semantic Search** ارزش واقعی سرویس است.
5. **Versioning** مهمتر از **WYSIWYG Editor** است.
6. **CDN + caching** تاثیر مستقیم روی تجربه کاربری دارد.
7. پیشنهاد خودکار لینک‌های داخلی از تکرار اشتباهات ویکی‌پدیا جلوگیری می‌کند.

گزارش کوتاهی از جلسات برگزار شده با تدریس‌یار

در این فاز دو جلسه اسکرام برگزار شد:

جلسه اول – دوشنبه ۱۰ آذر ۱۴۰۴

در این جلسه کلیت میکروسرویس و اهداف آن بررسی شد. بخش‌های اصلی شامل **Semantic Search** و خلاصه‌سازی با **LLM** شناسایی شدند. تقسیم وظایف بین اعضای تیم برای فاز اول انجام شد و نکاتی برای شروع پیاده‌سازی و برنامه‌ریزی فنی اولیه مطرح شد.

جلسه دوم – پنجشنبه ۱۳ آذر ۱۴۰۴

در جلسه دوم نیازمندی‌های عملیاتی و غیرعملیاتی مرور و موارد ابهام‌دار بررسی شد. تصمیم گرفته شد هر گروه می‌تواند میکروسرویس خود را به صورت مستقل انجام دهد و برای هماهنگی با سایر گروه‌ها یک **Leader** تعیین شود. همچنین بر ضرورت حفظ هماهنگی و سازگاری بین فایل‌ها و بخش‌های مختلف پروژه تأکید شد.