



دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر
مهندسی نرم افزار ۱

"فاز ۳ پروژه"

موضوع پروژه: Wiki Service

استاد درس: دکتر گلپاسی

نام گروه:

Nullterminated

فهرست بخش‌ها

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

سناریو پروژه

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

جایگاه Wiki Service در سامانه کلی

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، ویدئوهای محتوای چندسانه‌ای 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 معنابی:

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

برای طیف گسترده‌ای از کاربران طراحی شده است: **Wiki Service**

- 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
- علاقمندان به فرهنگ و تاریخ ایران که می‌خواهند دانش خود را گسترش دهنند
- نویسندهای کارشناسان که می‌خواهند دانش خود را با دیگران به اشتراک بگذارند
- سایر سرویس‌های سامانه که به یک منبع اطلاعاتی ساختار یافته نیاز دارند

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

دامنه موضوعی **Wiki Service**

بر روی چهار حوزه اصلی متمرکز است: **Wiki Service**

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"، "خامنشیان"، "تخت جمشید"

2. جغرافیا و مکان‌ها

- استان‌ها و شهرهای ایران
- روستاهای و مناطق کمتر شناخته شده
- 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 (قاليابافی، 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

متادینیا:

- 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

نسخه‌داری:

- شماره نسخه فعلی

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

کاربران و نقش‌های مختلف

۱. بازدیدکننده مهمان (Guest Visitor)

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

۲. کاربر ثبت‌نام‌شده (Registered User)

- تمام امکانات مهمان
- ذخیره مقالات مورد علاقه
- دریافت پیشنهادهای شخصی‌سازی شده
- ثبت نظرات و امتیازدهی

۳. نویسنده/کارشناس (Contributor)

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

۴. ویراستار/مدیر محتوا (Editor/Moderator)

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

حدودیت‌ها و قیود

حدودیت‌های 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

حدودیت‌های فنی:

- حداقل طول مقاله: ۲۰۰۰۰ کاراکتر
- حداقل تعداد تصاویر: ۲۰
- فرمت‌های مجاز: WebP, JPEG, PNG
- حداقل حجم تصویر: ۵ مگابایت
- حداقل 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 منتهی ساده
- جستجوی معنایی (Semantic Search) با NLP
- فیلتر بر اساس دسته، برچسب، تاریخ انتشار
- مرتب‌سازی بر اساس ارتباط، محبوبیت، تاریخ

2. نسخه‌داری 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 محتوا خودکار:

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

5. کنترل کیفیت:

- سیستم امتیازدهی کاربران
- بررسی املایی و دستوری
- تشخیص محتوا کپی شده (Plagiarism Detection)
- سیستم گزارش‌دهی برای محتوا 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

(Actors)

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

(Main Operational Flow)

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

(System Behavior)

- جمع‌آوری و پردازش داده‌ها
- استخراج مستقیم داده‌ها
- تشخیص موارد غیر عادی
- نمایش مسیرهای جایگزین در صورت خطا

(Post-Conditions)

- ایجاد حساب کاربر
- فعال بودن **Session**
- پردازش و ارسال داده‌ها
- ثبت تراکنش‌ها و لاغه‌ها

(Edge Cases & Exceptions)

- اطلاعات اشتباه -> پیام خطأ

- قطع سرویس خارجی -> **Retry** + پیام
- 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 موردنی (**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 علاقمند 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 محظوظ

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 فیلهای 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: بهروزرسانی 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

دانستان 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 برچسب‌های موضوعی خودکار

بازیگر: سیستم (فرآیند پس زمینه)

هدف: سازماندهی بهتر مقالات و بهبود جستجو

سناریوی تعامل:

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

معیارهای پذیرش:

- حداقل ۵ برچسب مرتبط تولید شود
- برچسب‌ها با محتواهای مقاله همخوانی داشته باشند
- امکان تأیید یا رد برچسب‌ها توسط نویسنده
- زمان تولید برچسب‌ها کمتر از ۵ ثانیه

وابستگی‌ها: US-03

داستان کاربر US-08: گزارش مقاله نامناسب

بازیگر: کاربر

هدف: اطلاع‌رسانی درباره محتواهی 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 (محتواهی نادرست، تصاویر نامناسب، اسپم و...)
- ثبت تاریخ و کاربر گزارش‌دهنده

- اطلاع‌رسانی خودکار به مدیران
- نمایش پیام موفقیت‌آمیز پس از ارسال

وابستگی‌ها: US-01

داستان کاربر US-09: فیلتر مقالات بر اساس دسته‌بندی

بازیگر: کاربر

هدف: یافتن سریع مقالات مرتبط با یک دسته خاص

سناریوی تعامل:

1. کاربر وارد بخش مقالات می‌شود
2. گزینه فیلتر بر اساس دسته‌بندی را انتخاب می‌کند
3. سیستم مقالات مرتبط با دسته انتخاب شده را نمایش می‌دهد
4. کاربر مقالات را مشاهده و انتخاب می‌کند

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

وابستگی‌ها: 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 می‌کند
2. سیستم تغییرات جدید در مقاله یا دسته را بررسی می‌کند
3. در صورت وجود تغییر، اعلان برای کاربر ارسال می‌شود
4. کاربر می‌تواند اعلان را بررسی و به مقاله دسترسی پیدا کند

معیارهای پذیرش:

- نمایش نوع تغییر (ویرایش متن، افزودن تصویر، تغییر برچسب)
- 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)

1.1-FR – پردازش جستوجوی معنایی

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

محرك: ورود عبارت جستوجو توسط کاربر

پیش‌شرط‌ها: پایگاه داده شامل مقالات ایندکس شده باشد

پس‌شرط‌ها: لیست مقالات مرتبط تولید شده باشد

وروهدی‌ها: عبارت جستوجو (رشته متى)

خروجی‌ها: لیست مقالات شامل عنوان، خلاصه و امتیاز ارتباط

داستان مرتبط: US-01

1.2-FR – نمایش نتایج جستوجو

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

ورويدى‌ها: لیست مقالات، شماره صفحه

خروجى‌ها: صفحه HTML شامل نتایج

داستان مرتبه: 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 باشد

پس‌شرطها: نتایج مطابق فیلتر نمایش داده شوند

ورويدى‌ها: نوع فیلتر

خروجى‌ها: لیست فیلتر شده مقالات

داستان مرتبه: 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

ورويدى‌ها: شناسه مقاله

خروجى‌ها: محتوای مقاله کامل

داستان مرتبه: 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

ورويدى‌ها: پیوند داخلی

خروجى‌ها: نمایش مقاله مقصد

داستان مرتبه: US-02

3.1-FR – ایجاد مقاله جدید

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

محرك: انتخاب گزینه "ایجاد مقاله جدید"

پیش‌شرط‌ها: نویسنده وارد سیستم شده باشد

پس‌شرط‌ها: مقاله ذخیره و منتشر شود

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

خروجی‌ها: مقاله جدید منتشر شده

دانستان مرتبط: US-03

3.2-FR – پیش‌نمایش مقاله قبل از انتشار

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

محرك: انتخاب گزینه "پیش‌نمایش"

پیش‌شرط‌ها: فرم مقاله پر شده باشد

پس‌شرط‌ها: پیش‌نمایش دقیق مقاله نمایش داده شود

ورودی‌ها: محتوای مقاله

خروجی‌ها: صفحه پیش‌نمایش

دانستان مرتبط: US-03

4.1-FR – ویرایش مقاله موجود

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

محرك: انتخاب گزینه "ویرایش"

پیش‌شرط‌ها: مقاله موجود باشد

پس‌شرط‌ها: تغییرات ذخیره و نسخه جدید ایجاد شود

ورودی‌ها: محتوای ویرایش شده

خروجی‌ها: نسخه جدید مقاله

دانستان مرتبط: US-04

4.2-FR – مشاهده تفاوت نسخه‌ها

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

محرك: انتخاب گزینه "تفاوت نسخه‌ها"

پیش‌شرط‌ها: چند نسخه از مقاله موجود باشد

پس‌شرط‌ها: 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.

محرك: انتخاب گزینه "بازگرداندن نسخه قدیمی"

پیش‌شرط‌ها: نسخه قدیمی موجود باشد

پس‌شرط‌ها: مقاله به نسخه قبلی بازگردد

ورويدی‌ها: 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.

محرك: باز کردن مقاله

پیش‌شرط‌ها: مقاله شامل برچسب یا محتوا باشد

پس‌شرط‌ها: لیست پیشنهادات نمایش داده شود

ورويدی‌ها: 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.

محرك: ذخیره مقاله

پیش‌شرط‌ها: مقاله جدید ذخیره شده باشد

پس‌شرط‌ها: برچسب‌ها تولید و ذخیره شوند

ورويدی‌ها: متن مقاله

خروجی‌ها: برچسب‌های پیشنهادی

داستان مرتبط: US-07

8.1-FR – گزارش مقاله نامناسب

شرح: کاربر باید بتواند مقاله را گزارش کند.

محرك: انتخاب گزینه "گزارش مقاله"

پیش‌شرط‌ها: مقاله موجود باشد

پسشرط‌ها: 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
پیششرط‌ها: کاربر مقاله یا دسته را دنبال کرده باشد
پسشرط‌ها: 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
پیششرط‌ها: دسته‌بندی‌ها موجود باشند
پسشرط‌ها: 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"
پیششرط‌ها: کاربر باید مجوز حذف داشته باشد و مقاله موجود باشد
پسشرط‌ها: 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

ورويدی‌ها: متن مقاله یا شناسه مقاله

خروجی‌ها: خلاصه مقاله (Short Summary / Extended Summary)

داستان مرتبه: 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

ورويدی‌ها: متن مقاله

خروجی‌ها: یک یا چند دسته‌بندی پیشنهادی

داستان مرتبه: US-07

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

شرح: کاربر باید بتواند داخل یک مقاله، با جست‌وجوی معنایی، بخش‌های مرتبط را پیدا کند.

محرك: وارد کردن عبارت جست‌وجو داخل بخش "Find"

پیش‌شرط‌ها: مقاله لود شده باشد

پس‌شرط‌ها: بخش‌های مرتبط هایلایت و لیست شوند

ورويدی‌ها: عبارت جست‌وجو

خروجی‌ها: بخش‌های یافته شده (Paragraph matches)

داستان مرتبه: 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

ورويدی‌ها: متن در حال تایپ

خروجی‌ها: فهرست پیوندهای پیشنهادی

داستان مرتبه: US-03, US-04

نیازمندی‌های غیر عملکردی (Non-Functional Requirements)

1.1-NFR – کارایی (Performance)

شرح: سیستم باید قادر به پاسخ‌دهی سریع به درخواست‌های جستجو باشد.

معیار ارزیابی: در شرایط ۱۰۰۰ کاربر همزمان، زمان پاسخ کمتر از ۲ ثانیه باشد.

سناریو: کاربر در اوج استفاده روزانه جستجو می‌کند.

داستان مرتبط: US-01

1.2-NFR – کیفیت نتایج جستجو (Quality)

شرح: نتایج جستجو باید ارتباط بالایی با عبارت جستجو داشته باشند.

معیار ارزیابی: حداقل ۸۰٪ از نتایج صفحه اول مرتب باشند ($0.8 < \text{Precision}$).

سناریو: کاربر عبارت "تاریخ صفویه" را جستجو می‌کند.

داستان مرتبط: US-01

2.1-NFR – امنیت (Security)

شرح: تنها کاربران مجاز 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 مازولار و مستند باشد تا توسعه و تغییرات آسان باشد.

معیار ارزیابی: تمام توابع دارای Docstring و پوشش تست $\leq 80\%$.

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

(Reliability) – قابلیت اطمینان 4.1-NFR

شرح: سیستم باید پایدار بوده و مقالات ذخیره شده را از دست ندهد.

معیار ارزیابی: بدون از دست رفتن داده در ۹۹٪ موقع.

سناریو: نویسنده مقاله ایجاد می‌کند و سیستم در حال بروزرسانی است.

داستان مرتبط: US-03, US-04

(Security) – امنیت داده‌ها 4.2-NFR

شرح: تمام داده‌های کاربران و تغییرات مقالات در ذخیره‌سازی و انتقال رمزگذاری شوند.

معیار ارزیابی: استفاده از AES-256 برای ذخیره‌سازی و HTTPS برای انتقال.

سناریو: کاربر در حال ایجاد یا ویرایش مقاله است.

داستان مرتبط: US-03, US-04

NFR-5.1 – 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

(Reliability) – جلوگیری از گزارش تکراری 6.1-NFR

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

(Maintainability) – لากذاری 8.1-NFR

شرح: سیستم باید قابلیت لاگگیری کامل از تغییرات و گزارش‌ها داشته باشد.

معیار ارزیابی: تمام فعالیت‌ها در فایل لاغ 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 قادر اطلاعات غلط باشد.

معیار ارزیابی: بررسی دستی ۵۰ خلاصه تولیدی، $0.75 \leq \text{Semantic Similarity}$.

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 خطأ، پیام مناسب نمایش دهد و عملکردهای پایه ادامه یابند.

سناریو: کاربر خلاصه می‌خواهد ولی LLM در دسترس نیست.

داستان مرتبه: **US-03, US-04, US-02**

اعتبارسنجی نیازمندی‌ها

بخش اول – نیازمندی‌های عملکردی (FR)

1.1-FR – پردازش جستجوی معنایی

: کاربران جستجوکننده (عمومی، پژوهشگر، سرویس‌های داخلی/خارجی) Who
: پیاده‌سازی موتور Semantic Search برای یافتن و رتبه‌بندی مقالات بر مبنای درک معنایی عبارت جستجو What
: افزایش کیفیت نتایج، افزایش ارزش اطلاعاتی سرویس و بهبود تجربه کاربری Why
: When

• اولویت: High

• پیشنهاد: 1 – هسته اولیه سیستم؛ پیاده‌سازی پایه‌ی جستجوی معنایی با مدل سبک برای MVP.
Where: در میکروسرویس جستجو (Search Service) و لایه‌ی API.
How: استفاده از مدل‌های embedding (مثلًا SBERT یا transformer-based) یا محاسبه similarity.
Why: ایندکس‌سازی متن (مثلًا dense vector یا سرویس Elasticsearch) و رتبه‌بندی 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 جستجو

Who: همه کاربران جستجوکننده.
What: نمایش نتایج به صورت صفحه‌بندی 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 صفحه‌بندی، طراحی کامپونت‌های UI برای نتایج، مدیریت ناوبری صفحات.
How Much: زمان ۱–۱.۵ هفته؛ منابع: ۱ بکاند؛ هزینه کم؛ پیچیدگی: کم.

1.3-FR – فیلتر نتایج جستجو

: کاربران پیشرفت و عمومی. Who

: فیلتر نتایج براساس تاریخ، محبوبیت و دسترسی. What

: کاهش زمان جستجو و دسترسی سریع‌تر به نتایج مرتبط. Why

: اولویت: Medium-High – پیشنهاد: 2-1 Sprint (پیاده‌سازی اولیه در 1 Sprint، ارتقاء در 2 Sprint). When

: Where: Search Service (query params) و UI فیلترها.

: افزودن پارامترهای query، نگاشت فیلترها به شاخص‌های ایندکس شده، و UI کنترل فیلتر. How

: زمان 1.5-2 هفته؛ منابع: 1 بگاند؛ هزینه: کم تا متوسط؛ پیچیدگی: متوسط. How Much

2.1-FR – باز کردن مقاله و نمایش محتوا

: تمام کاربران خواننده. Who

: نمایش کامل مقاله شامل متن، تصاویر و پیوندها. What

: هدف اصلی سرویس – دسترسی به محتوا کامل. Why

: اولویت: High – پیشنهاد: 1 Sprint. When

: Article Retrieval API (Where: Article View Page (Frontend

: How: API و اکتشافی مقاله بر اساس شناسه، CDN برای تصاویر، رندر سمت مشتری. API

: زمان 1-2 هفته؛ منابع: 1 بگاند؛ هزینه: کم؛ پیچیدگی: متوسط. How Much

2.2-FR – پیوندهای داخلی قابل کلیک

: خوانندگان مقالات. Who

: لینک‌های داخلی مقاله باید قابل کلیک و باز کردن مقاله مقصد بدون از دست رفتن وضعیت باشند. What

: بهبود پیمایش و کشف محتوا مرتبط. Why

: اولویت: High – پیشنهاد: 1 Sprint یا بلافاصله پس از FR-2.1. When

: Frontend routing و Where: Rendering layer در محتوا ذخیر شده. rendering

: لینکدهی به شناسه‌ها، استفاده از تاریخچه مرورگر (pushState) برای حفظ وضعیت. How

: زمان 3-5 روز؛ منابع: 1 فرانت‌اند؛ هزینه: بسیار کم؛ پیچیدگی: کم. How Much

3.1-FR – ایجاد مقاله جدید

: نویسندها و کاربران دارای مجوز نوشتن. Who

: قابلیت ایجاد مقاله شامل عنوان، متن، تصاویر، برچسب‌ها و پیوندها. What

: افزایش محتوا سامانه و بهروزرسانی پایگاه دانش. Why

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

3.2-FR – پیش‌نمایش مقاله

Who: نویسنده‌گان.
What: پیش‌نمایش مقاله قبل از انتشار بدون ایجاد نسخه رسمی.
Why: جلوگیری از خطاهای 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 render endpoint 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 نویسنده‌گان مجاز.
What: امکان ویرایش محتوای مقاله و ذخیره به صورت نسخه جدید.
Why: نگهداری کیفیت محتوا و رفع اشکال/بروزرسانی.
When: اولویت: High – پیشنهاد: 2 Sprint
Where: Editor UI، Article API، Versioning Service
How: پیاده‌سازی قابلیت ویرایش در Editor و ذخیره نسخه جدید با metadata نسخه.
How Much: زمان 1–2 هفته؛ منابع: 1 فرانت‌اند، 1 بک‌اند؛ هزینه: متوسط؛ پیچیدگی: متوسط.

4.2-FR – مشاهده تفاوت نسخه‌ها

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

5.1-FR – بازیابی نسخه قدیمی مقاله

: ویراستار/مدیر (کاربران مجاز). Who

: بازگرداندن مقاله به یک نسخه قبلی و ایجاد نسخه جدید جایگزین. What

: توانایی برگرداندن تغییرات اشتباه یا حذف ناخواسته. Why

. Sprint 3 – پیشنهاد: When

. Admin UI و Where: Versioning Service

: ثبت کپی نسخه قدیمی به عنوان نسخه جدید، ثبت تاریخچه بازگردانی. How

: زمان 4–7 روز؛ منابع: 1 بکاند؛ هزینه: کم؛ پیچیدگی: کم-متوسط. How Much

6.1-FR – پیشنهاد مقالات مرتب

: خوانندگان مقاله. Who

: نمایش لیست مقالات مرتب بر اساس محتوا/برچسب/شناخت معنایی. What

: افزایش تعامل کاربر و کشف محتوا. Why

. Sprint 3 – پیشنهاد: When

. UI؛ Search Service و Where: Recommendation Engine

: محاسبه similarity (content/tag-based)، احتمالاً استفاده از مدل‌های embedding یا گراف. How

: زمان 2–3 هفته؛ منابع: 1 بکاند، 1 متخصص similarity؛ هزینه: متوسط؛ پیچیدگی: متوسط-زیاد. How Much

7.1-FR – تولید برچسب خودکار

: نویسنده‌گان (پس از ذخیره مقاله) و سیستم. Who

: استخراج و پیشنهاد برچسب‌ها به صورت خودکار با استفاده از NLP. What

: سهولت برچسب‌گذاری و بهبود دسته‌بندی و جستجو. Why

. Sprint 3 – پیشنهاد: When

. Article Creation flow و Where: Tag Generator Service

: کلیدواژه‌کاوی، NER، TF-IDF یا مدل embedding برای استخراج برچسب‌های کاندید. How

: زمان 1.5–2.5 هفته؛ منابع: 1 NLP/ML، 1 بکاند؛ هزینه: متوسط؛ پیچیدگی: متوسط-زیاد. How Much

8.1-FR – گزارش مقاله نامناسب

: تمام کاربران. Who

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

.Why: حفظ کیفیت محتوا و امکان بررسی تخلفات.
.Sprint 2: اولویت: High – پیشنهاد: When
.Admin Dashboard و Where: Report Service
.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 نگهداری کاربران.
.Sprint 3: اولویت: Medium – پیشنهاد: When
.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.
.Sprint 2: اولویت: Medium – پیشنهاد: When
.Where: Category Browser و API
.How: تقسیم 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 مطلب نامعتبر یا غیرقابل استفاده و کنترل نسخه‌ها.
.When: پیشنهاد: 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
.Summary: فراخوانی API مدل زبانی، پردازش متن و تولید/نمایش یا ذخیره.
.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: نویسنده‌گان 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: Sprint 4 یا 3 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 – جست‌وجوی هوشمند داخل مقاله (Semantic In-Article Search)

.Who: کاربران مطالعه‌کننده مقاله.
.What: امکان جست‌وجوی معنایی در متن مقاله و نمایش بخش‌های مرتبط.
.Why: افزایش سرعت دسترسی به بخش‌های موردنیاز در مقالات طولانی.
.When: Sprint 4
.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: نویسنده‌گان هنگام تایپ محتوا.
.What: پیشنهاد مقالات مرتبط برای ایجاد لینک داخلی در لحظه نوشتن.

.Sprint 4: **When**
.Why: غنی‌سازی ساختار مقاله و بهبود پیوند داخلی.
.Link Suggestion Service و Where: Editor UI
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 غیرعملیاتی (NFR)

1.1-NFR – کارایی (Performance)

.Who: تمام کاربران سیستم.
.What: سیستم باید در شرایط ۱۰۰۰ کاربر همزمان، زمان پاسخ جستجو را < 2 ثانیه نگه دارد.
.Why: حفظ تجربه کاربری و کاهش نرخ ترک.
.When: اولویت: High – اعمال از 1
.Where: Search Service
.DB, API Gateway، زیرساخت شبکه و
How: بهینه‌سازی ایندکس‌ها، (caching (Redis)، توزیع بار، auto-scaling و مانیتورینگ (Prometheus/Grafana)
.Sprint 1–3: پیشنهاد
.Large: 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 جستجوکنند.
.What: دقت نتایج صفحه اول ≤ 0.8 (Precision > 0.8)
.Why: اطمینان از مرتبط بودن نتایج.
.When: اولویت: High – پیشنهاد
.Where: Semantic Search pipeline
.evaluation datasets، How: fine-tune
How: مدل‌ها، ارزیابی IR، بازخورد کاربران.
.Large: زمان 3–5 هفته؛ منابع: 1 NLP/ML؛ هزینه: متوسط تا زیاد؛ پیچیدگی: Large

2.1-NFR – امنیت (Authorization)

Who: نویسنده‌گان، ویراستاران، مدیران.

What: فقط کاربران مجاز بتوانند مقالات را ایجاد/ویرایش کنند.

Why: حفاظت از محتوا و جلوگیری از دسترسی غیرمجاز.

When: اولویت: 1 – پیشنهاد High .Sprint 1

Where: Auth Service در سرویس‌ها.

How: RBAC، API، UI، Where: Auth Service در سرویس‌ها.

How Much: زمان 1–2 هفته؛ منابع: 1 بکاند، 1 Security Engineer پاره وقت؛ هزینه: متوسط؛ پیچیدگی: متوسط – زیاد.

3.1-NFR – نگهداری (Maintainability)

Who: تیم توسعه و توسعه‌دهندگان جدید.

What: کد مازولار و مستند با Docstring و پوشش تست $\leq 80\%$.

Why: تسهیل توسعه و کاهش هزینه نگهداری.

When: اولویت: 1 – 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 2.1 WCAG سطح AA در UI.

Why: دسترسی‌پذیر کردن سرویس برای همه کاربران.

When: اولویت: 1–2 .Medium-High – Sprint 1–2

Where: Frontend components, templates, CSS, ARIA attributes

How: semantic HTML .accessibility، ARIA roles

How Much: زمان 1–2 هفته بازبینی UI؛ منابع: 1 فرانت‌اند؛ هزینه: کم؛ پیچیدگی: متوسط.

4.1-NFR – قابلیت اطمینان (Reliability)

Who: نویسنده‌گان و کاربران.

What: عدم از دست رفتن مقالات؛ پایداری بالا (مثلاً uptime 99.9%).

Why: حفظ محتوا ارزشمند و اعتماد کاربران.

اولویت: High – طراحی backup/replication از ابتدا.
.versioning system، storage، Where: DB
.monitoring، transactional writes منظم، backup، How: replication
زمان 2–3 هفته طراحی + پیکربندی؛ منابع: 1 بکاند، 1 DevOps؛ هزینه: متوسط تا زیاد؛ پیچیدگی: How Much
.Large

(Encryption) – امنیت 4.2-NFR

Who: همه کاربران و داده‌ها.
What: رمزگذاری در انتقال (HTTPS) و در ذخیره‌سازی (AES-256).
Why: حفظ محترمانگی و الزامات قانونی.
When: اولویت: High – فعال از روز اول.
.DB encryption at rest، Where: Transport layer (TLS)
.storage برای کلیدها، رمزگذاری در KMS، How: HTTPS/TLS
زمان 4–7 روز؛ منابع: 1 بکاند، 1 DevOps؛ هزینه: کم تا متوسط؛ پیچیدگی: کم–متوسط.

(Load time) – کارایی بارگذاری 5.1-NFR

Who: خواننده‌گان مقاله.
What: زمان بارگذاری مقاله > 1 ثانیه در ۵۰۰ درخواست همزمان.
Why: تجربه کاربری روان.
When: اولویت: 2–1 High – Sprint 1
.Where: Frontend, CDN, backend APIs, DB
.caching استاتیک، How: CDN
z lazy loading تصاویر، How: CDN
زمان 2–3 هفته؛ منابع: 1 بکاند، 1 فرانت‌اند، 1 DevOps؛ هزینه: متوسط؛ پیچیدگی: Large
How Much

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

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

(Tag quality) 7.1-NFR

: نویسنده‌گان و کاربران. Who

: برچسب‌ها $\leq 70\%$ نطبق با محتوای مقاله. What

: کاهش نیاز به ویرایش دستی و بهود جستجو. Why

: اولویت: Sprint 3 – پیشنهاد 3. When

. Where: Tag Generator Service

: ارزیابی نمونه‌ای، بازخورد انسانی، الگوریتم‌های NER/keyword extraction. How

: زمان 1.5–3 هفته؛ منابع: 1 NLP/ML؛ هزینه: متوسط؛ پیچیدگی: متوسط–زیاد. How Much

(Logging) 8.1-NFR

: تیم پشتیبانی و توسعه. Who

: لاگ‌گیری کامل از فعالیت‌های ایجاد، ویرایش و گزارش‌ها. What

: ردیابی خطأ، تحلیل رفتار و حسابرسی. Why

: اولویت: 1 – فعال از 1. When

. Log Aggregation، Audit Logs، Where: Middlewares

: retention policy، How: structured logging

: زمان 1–2 هفته؛ منابع: 1 DevOps؛ هزینه: متوسط؛ پیچیدگی: متوسط. How Much

(Restore access control) 9.1-NFR

: ویراستاران/مدیران مجاز. Who

: فقط کاربران مجاز توانایی بازیابی نسخه‌های قدیمی را داشته باشند. What

: جلوگیری از دستکاری غیرمجاز تاریخچه محتوا. Why

: اولویت: 3 – پیشنهاد 3. When

. Authorization checks، Where: Versioning Service

: policy-based access control یا How: RBAC

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

(Notifications display) 10.1-NFR

: کاربران موبایل و دسکتاپ. Who

: اعلان‌ها و پیشنهادات باید در رزولوشن‌های مختلف قابل مشاهده باشند. What

: تجربه یکنواخت در پلتفرم‌ها.

.**Sprint 3** – **پیشنهاد Medium** : اولویت

.**Notification UI**, **Where: Frontend responsive layouts**

.**adaptive components**, **How: طراحی responsive**, تست در رزولوشن‌های مختلف،

.**How Much: زمان 3–6 روز؛ منابع: 1** فرانت‌اند؛ هزینه: کم؛ پیچیدگی: کم–متوسط.

11.1-NFR – امنیت عملیات حذف مقاله

: 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 2 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

: کاربران استفاده‌کننده از خلاصه‌سازی.

.**What: زمان تولید خلاصه ≥ 3 ثانیه برای مقاله ~ 2000 کلمه.**

.**Why: حفظ تجربه کاربری روان.**

.**When: پیشنهاد 3–4 Sprint 3**

.**Where: LLM Service** و API مرتبط.

.**How: cache** نتایج، مدل سبکتر، موازی‌سازی.

.**How Much: زمان 1–2 هفته بین‌مسازی؛ منابع: 1** ML/NLP، 1 بکاند؛ هزینه: متوسط؛ پیچیدگی: Medium–Large

13.1-NFR – کیفیت خروجی LLM

: نویسنده‌گان و کاربران خلاصه.

.**What: خلاصه با Semantic Similarity ≤ 0.75 و بدون hallucination**

.**Why: جلوگیری از خلاصه‌های نادرست و حفظ اعتبار.**

.**When: پیشنهاد 4 Sprint 4**

.**Where: LLM Evaluation Pipeline**

.**How: نمونه‌برداری دستی، تست خودکار، مدل‌های تشخیص خطأ.**

.**How Much: زمان 2–3 هفته؛ منابع: 1** NLP/ML؛ هزینه: متوسط تا زیاد؛ پیچیدگی: Large

14.1-NFR – پایداری در قطع سرویس LLM

: تمام کاربران Who

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

: جلوگیری از اختلال تجربه کاربری Why

.Sprint 4 When پیشنهاد

.LLM Service در UI و Where: Fallback Layer

، پیام کاربرپسند، ثبت لاغ fallback handler، How: timeout

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

اطلاعات جمع‌آوری شده از مطالعات موردي

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 شبیه باشد.

نمونه انتخاب شده:

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

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 جستجو میکند.
- لینک ویکی‌پدیا یا وبسایت‌های گردشگری در نتایج دیده میشود.
- کاربر وارد صفحه میشود.

الگوهای استخراج شده:

- ۹۰٪ کاربران از طریق **Wiki Search** وارد **Home Page** میشوند، نه از طریق **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 استخراج شده:

- 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 کلیک دیگر روی لینک‌های داخلی دارند.
- کاربران عمیق‌خوان، گاهی تا ۸-۷ کلیک متوالی انجام می‌دهند.

الگوهای استخراج شده:

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

مرحله ۴: جستجوی عمیق‌تر

در **TripAdvisor** یا **Wikipedia**

- کاربر عبارتی مانند "Safavid architecture" یا "جاده‌های اصفهان" را جستجو میکند.

سیستم پیشنهادهای مرتب نمایش می‌دهد:

- معماری صفوی
- میدان نقش‌جهان
- مسجد شیخ لطف‌الله

بینش کاربردی:

- نیت کاربر (Search Intent) مهم است.
- جستجوی 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 (Notification Service) و خدمات اطلاع‌رسانی (Favorite Storage) است.

مرحله ۶: ثبت بازخورد و مشارکت

در Wikipedia:

- کاربران حرفه‌ای محتوا را ویرایش می‌کنند.
- کاربران عادی پیشنهاد تغییر می‌دهند.
- تاریخچه نسخه‌های زامنی 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: جستجوی ضعیف و وابسته به کلمات کلیدی

- Wikipedia فارسی جستجوی 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 غیر‌هدفمند

• در بسیاری صفحات:

- تعداد زیادی لینک وجود دارد
- برخی بی‌ربط هستند
- برخی مهم‌ها لینک نشده‌اند

: نتیجه

• پیشنهاد خودکار لینک‌های داخلی (15.1-FR) 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 غیرشخصی‌سازی 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

- استفاده نمی‌شود **CDN**
 - زمان لود صفحه بیشتر از ۴ ثانیه است
- نتیجه:
- بسیار حیاتی است. **(NFR-5.2 Load Time (<1s)**

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 پیشرفت (الزامی و محوری) **Semantic Search**
- .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 غیرعملیاتی (**NFR**)

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

۵. الگوهای طلایی استخراج شده

- .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. ارزش واقعی سرویس است. **Semantic Search**
5. مهمتر از **WYSIWYG Editor Versioning** است.
6. تاثیر مستقیم روی تجربه کاربری دارد. **CDN + caching**
7. پیشنهاد خودکار لینک‌های داخلی از تکرار اشتباهات و یکی‌پدیا 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.