قسم: الـBackend (نسخه محدثه)

الفهرس

1. مقدمة عامة (General Introduction)

الغرض

- إنشاء منصة/موقع عام يستطيع المستخدمون من خلاله مشاهدة فيديوهات حصرية ومقالات وقصص، مع إمكانية إضافة ميزة صغيرة لرفع القصص من قبل المستخدمين.
- الاستفادة من الذكاء الاصطناعي في تحويل النصوص إلى فيديوهات وصوتيات متعددة اللغات مع مؤثرات صوتية (360، عاطفية، اهتزازات Haptic).
- وجود نظام اشتراكات (مجاني/مدفوع) واستخدام بوابات دفع متعددة تناسب 12 لغة/دولة.

---

2. قسم: الـBackend

2.1 نظرة عامة

سيكون الـBackend مسؤولًا عن كل العمليات الإدارية والربط مع الذكاء الاصطناعي وواجهات الدفع، وإدارة المحتوى. يجب تصميمه بأسلوب نظيف يمكن توسيعه لاحقًا (قابلية التوسع Microservices، أو Modular).

2.2 المتطلبات الوظيفية (Functional Requirements)

1. **إدارة المستخدمين (User Management)**

- تسجيل المستخدمين وتوثيق حساباتهم (قد يكون بريد إلكتروني/هاتف).
- تسجيل دخول (Login) وخروج (Logout) باستخدام JWT أو OAuth2.
- تخصيص صلاحيات (المالك/المستخدم العادي/كاتب معتمد).

2. **إدارة الاشتراكات (Subscription Management)**
   - خطط اشتراك: مجاني، قياسي، مميز (وغيرها لو تطلب الأمر).
   - ضبط حدود الاستهلاك (عدد المشاهدات، جودة الفيديو، الخواص الصوتية).
   - تجديد تلقائي (Recurring) للمستخدمين الذين يختارون ذلك.
   - حفظ سجلات المدفوعات وتواريخ التجديد.
3. **إدارة المحتوى (Content Management)**
   - **المقالات (Articles)**: إضافة/تحرير/حذف المقالات، تصنيفها، تحديد الظهور في الرئيسية أو قسم المدونة.
   - **الفيديوهات (Videos)**: رفع رابط الفيديو (على سيرفر داخلي أو YouTube/Vimeo) مع حفظ البيانات الوصفية (العنوان، الوصف، الصورة المصغرة).
   - **القصص القصيرة (Short Stories)**:

     •

     من إنتاجك أنت (المالك) أو من كتّاب معتمدين.

     •

     تمييز المحتوى حسب التصنيف (قصص خيالية، رعب، تاريخية... إلخ).
   - **قسم رفع القصص من المستخدمين** (Submit Your Story):

     •

     استقبال الطلبات في حالة "معلق/قيد المراجعة".

     •

     إمكانية قبول/رفض القصة من لوحة التحكم.
4. **واجهات API مع الذكاء الاصطناعي**
   - إرسال نص القصة (أو المقال) إلى خادم الذكاء الاصطناعي لتحليله (اكتشاف الانتحال، الأخطاء اللغوية).
   - استقبال نتيجة التحليل (نسبة التشابه مع أعمال معروفة، مدى الأصالة،

الأخطاء...).

- إرسال طلب تحويل نص إلى صوت/فيديو، وإدارة حالة المعالجة (قيد التشغيل/ مكتمل).

5. **التعامل مع بوابات الدفع (Payment Gateways)**
   - التكامل مع مزوّد دولي (Stripe أو Adyen أو 2C2P) لتوفير الطرق الأساسية (بطاقات ائتمان، PayPal...).
   - تتبّع نجاح الدفع وفشل الدفع، وتحديث الاشتراك في قاعدة البيانات.
   - حفظ إيصالات الدفع.
6. **إدارة التحليلات (Analytics)**
   - عدد المشاهدات/الزيارات لكل مقال أو فيديو أو قصة.
   - عدد القصص المرفوضة/المقبولة.
   - إحصائيات الاشتراكات (معدل التحويل، إيرادات شهرية...).
   - إمكانية عرض تحليلات مختلفة في لوحة التحكم.
7. **إرسال التنبيهات (Notifications)**
   - تنبيه المالك عند وصول قصة جديدة من المستخدمين.
   - تنبيه المستخدم حين يتم قبول القصة أو رفضها.
   - تنبيه المستخدم بانتهاء اشتراكه أو نجاح دفعه.
   - دعم قنوات التنبيه (بريد إلكتروني/رسائل داخل الموقع/WebSockets).

2.3 المتطلبات غير الوظيفية (Non-Functional Requirements)

1. **الأداء (Performance)**
   - استجابة سريعة لمعظم عمليات الـAPI (< 2-3 ثوانٍ).
   - إدارة حمل العمليات الثقيلة (التحويل بالـAI) بواسطة طوابير أو Jobs Queue (Celery + Redis).
2. **الأمان (Security)**
   - بروتوكول HTTPS فقط.
   - تشفير كلمات المرور (Salt + Hash).
   - صلاحيات واضحة (RBAC) بين المالك والكاتب والمستخدم العادي.
   - تأمين مفاتيح الـAPI الخاصة بالذكاء الاصطناعي وبوابات الدفع.
3. **القابلية للتوسع (Scalability)**
   - هيكلة تسمح بإضافة سيرفرات جديدة (Load Balancer) أو فصل بعض الوحدات.
   - إمكانية فصل خدمة التحليل والمحتوى في Microservices إذا زاد الحمل.
4. **قابلية الصيانة (Maintainability)**
   - استخدام أطر عمل (Framework) مثل Flask/FastAPI أو Django.
   - كود نظيف وواضح مع توثيق الـAPI (Swagger أو OpenAPI).

- اعتماد Docker/Kubernetes (اختياري) للتنصيب السريع.

5. **النسخ الاحتياطي (Backup & Recovery)**
   - نسخ دوري لقاعدة البيانات والمحتوى المخزّن.
   - وجود خطة استعادة في حال تعطل السيرفر.

2.4 البنية الهيكلية وقاعدة البيانات

- **قاعدة بيانات (Database)**: PostgreSQL أو MySQL.
- جداول مقترحة:

•

users (حسابات المستخدمين).

•

subscriptions (تفاصيل الخطط والاشتراكات).

•

payments (سجلات الدفع).

•

articles (المقالات).

•

videos (الفيديوهات).

•

stories (القصص القصيرة).

•

stories_submissions (القصص المرفوعة من

المستخدمين).

•

analytics (التتبع والإحصائيات).

- **إدارة المهام:**
  - Redis + Celery لمعالجة الطلبات الضخمة (تحويل نص إلى فيديو مثلاً).

2.5 تدفق العمليات (Workflow)

1. **إضافة محتوى من المالك**
   1. المالك يدخل لوحة التحكم → ينشئ مقالًا أو فيديوًا أو قصة قصيرة.
   2. يحفظ المحتوى وتُحدّث قاعدة البيانات.
   3. يظهر المحتوى فوريًا في الصفحة الرئيسية (إن تم تمكينه) أو ينتظر موافقة.
2. **رفع قصة من المستخدم**
   1. المستخدم (زائر أو مسجل) يضغط زر "Submit Your Story" في الصفحة الرئيسية.
   2. يُدخل النص (و/أو ملف مرفق إن وُجد) → تُحفظ في جدول stories_submissions بحالة Pending.
   3. لوحة التحكم تعرض تنبيهًا للمالك → يقرر قبولها/رفضها.
   4. إن قُبلت، قد تُنشر في الموقع. وإن رغب المالك، يرسل النص للـAI للتحويل/التحليل.
3. **نظام الاشتراكات**
   1. المستخدم يختار خطة (مجاني/قياسي/مميز).
   2. عند اختيار خطة مدفوعة → يذهب لبوابة الدفع → يعود إشعار نجاح أو فشل.
   3. نجاح الدفع = تفعيل الخطة في جدول subscriptions.
4. **طلب تحليل AI**
   1. عند إضافة قصة/مقال: يستدعي الـBackend واجهة //ai analyze (خادم الـAI).
   2. ينتظر النتيجة (عبر WebSocket أو طلب GET دوري).
   3. يستقبل التحليل (نسبة الانتحال، الأخطاء، إلخ) ويخزنه في قاعدة البيانات.