1. نظرة عامة

يهدف نظام إدارة المشاريع (PMS) إلى تسهيل تتبع المهام، تعزيز التعاون، وإدارة الموارد بكفاءة. سيكون النظام قابلاً للتوسع، ومرنًا، وسهل الاستخدام، ويناسب أحجام المشاريع المختلفة وتعقيداتها.

2. بنية النظام

سيتبع النظام بنية العميل-الخادم مع المكونات التالية:

الواجهة الأمامية (الجانب العميل):

واجهة ويب يمكن الوصول إليها عبر المتصفحات والأجهزة المحمولة.

تصميم واجهة المستخدم (UI) سهل التنقل ومتوافق مع جميع الأجهزة.

التقنيات: HTML، CSS، JavaScript (React، Angular، أو Vue.js).

الواجهة الخلفية (الخادم):

معالجة المنطق التجاري، معالجة البيانات، والمصادقة للمستخدمين.

التقنيات: Node.js، Python (Django/Flask)، Ruby on Rails.

واجهة برمجة التطبيقات (API) RESTful للتواصل بين الواجهة الأمامية والخلفية.

قاعدة البيانات:

تخزين بيانات المستخدمين، تفاصيل المشاريع، معلومات المهام، ومقاييس التقدم.

التقنيات: MySQL أو PostgreSQL (علاقاتية) أو MongoDB (NoSQL، للتوسع).

الاستضافة/البنية التحتية السحابية:

سيتم استضافة النظام على منصات سحابية مثل AWS أو Azure أو Google Cloud لتحقيق قابلية التوسع والتوافر.

سيتم تنفيذ توازن التحميل، النسخ الاحتياطي الآلي، والنسخ الاحتياطي المزدوج.

3. التصميم الوظيفي

أ. أدوار المستخدمين والصلاحيات

المسؤول:

الوصول الكامل إلى جميع ميزات النظام (إنشاء المشاريع، إدارة المستخدمين، عرض جميع التقارير).

مدير المشروع:

الوصول لإدارة المشاريع، تعيين المهام، تتبع التقدم، وإنشاء التقارير.

عضو الفريق:

عرض المهام، تحديث التقدم، والتعاون مع أعضاء الفريق الآخرين.

العميل (اختياري):

وصول محدود لعرض حالة المشروع، والمعالم، والتقارير.

ب. الميزات الأساسية

1. اللوحة الرئيسية (Dashboard):

عرض شامل للمشاريع الجارية، المهام، والمواعيد النهائية.

تصورات مثل مخططات جانت، لوحات كانبان، ومؤشرات التقدم.

2. إدارة المشاريع:

إنشاء وتعديل وتتبع المشاريع.

تحديد الجداول الزمنية، المعالم، والتسليمات.

تعيين الأدوار والمسؤوليات لأعضاء الفريق.

3. إدارة المهام:

إنشاء وتعيين المهام لأعضاء الفريق.

تحديد المواعيد النهائية، الأولويات، والتبعيات.

تتبع التقدم وتحديث الحالات.

4. إدارة الموارد:

إدارة توافر أعضاء الفريق، المهارات، وأحمال العمل.

تخصيص الموارد للمهام بناءً على التوافر.

5. أدوات التعاون:

رسائل داخل التطبيق، لوحات المناقشة، ومشاركة الملفات.

إشعارات في الوقت الفعلي لتحديثات المهام، التعليقات، والمواعيد النهائية.

6. التقارير والتحليلات:

إنشاء تقارير حول حالة المشروع، استخدام الموارد، وإتمام المهام.

رسومات بيانية للمساعدة في اتخاذ القرارات (مثل مخططات الانتهاء، تخصيص الموارد).

7. إدارة الوثائق:

تخزين مركزي للوثائق الخاصة بالمشروع (الملفات، المواصفات، إلخ).

التحكم في الإصدار وقدرات مشاركة الوثائق.

8. الإشعارات والتنبيهات:

تنبيهات تلقائية للمواعيد النهائية القادمة، تحديثات المهام، أو معالم المشاريع.

إشعارات عبر البريد الإلكتروني أو الدفع أو في التطبيق.

4. التصميم غير الوظيفي

أ. الأمان:

المصادقة:

التحكم في الوصول بناءً على الأدوار (RBAC) للمستويات المختلفة للمستخدمين.

المصادقة الثنائية (2FA) للإجراءات الحساسة.

أمان البيانات:

تشفير البيانات الحساسة (مثل تفاصيل المشروع، معلومات المستخدم).

إجراء تدقيقات أمنية واختبارات اختراق دورية.

ب. الأداء:

القابلية للتوسع:

البنية التحتية السحابية تسمح بالتوسع بناءً على عدد المستخدمين وحجم البيانات.

توازن التحميل وتقسيم قواعد البيانات خلال فترات الازدحام.

الموثوقية:

ضمان وقت تشغيل بنسبة 99.9%، مع أنظمة احتياطية وآليات النسخ الاحتياطي.

النسخ الاحتياطي الآلي وآليات استعادة البيانات.

ج. قابلية الاستخدام:

تصميم متجاوب:

النظام سيكون متاحًا عبر الأجهزة (الكمبيوتر، الأجهزة اللوحية، الهواتف المحمولة).

واجهة مستخدم/UI بديهية وسهلة الاستخدام.

إمكانية الوصول:

الامتثال لمعايير WCAG (إرشادات الوصول إلى محتوى الويب) للمستخدمين ذوي الإعاقات.

5. مخطط تدفق البيانات (DFD)

المستوى 0 (مخطط السياق):

يتفاعل المستخدمون (المسؤول، مدير المشروع، عضو الفريق، العميل) مع نظام إدارة المشاريع، حيث يمكنهم الوصول إلى المشاريع، المهام، والتقارير.

المستوى 1 (تدفق البيانات):

تتدفق البيانات من المستخدمين إلى النظام لإنشاء المهام، تتبع المشاريع، وإنشاء التقارير.

تتم معالجة البيانات وإرجاعها كتعليقات (تحديثات المهام، تقارير المشروع، إلخ).

6. تكامل النظام

التكامل مع الأطراف الثالثة:

التكامل مع أدوات الاتصال (مثل Slack، Microsoft Teams) للإشعارات والدردشة.

التكامل مع خدمات التخزين السحابي (مثل Google Drive، Dropbox) لإدارة الوثائق.

واجهات برمجة التطبيقات الخارجية:

التكامل مع تطبيقات التقويم (مثل Google Calendar، Outlook) للجدولة والتذكيرات.

التكامل مع بوابات الدفع (لإصدار الميزات أو الاشتراكات المدفوعة).

7. مجموعة التكنولوجيا

الواجهة الأمامية: HTML، CSS، JavaScript (React، Angular، أو Vue.js).

الواجهة الخلفية: Node.js، Python (

قاعدة البيانات: MySQL،

البنية التحتية السحابية: AWS، Azure، Google Cloud.

المراسلة والإشعارات: Firebase Cloud Messaging (FCM) أو WebSockets.

8. الخلاصة

يضمن هذا التصميم أن يكون نظام إدارة المشاريع قابلًا للتوسع، آمنًا، وسهل الاستخدام. تتيح البنية المعمارية القابلة للتوسيع إمكانية إضافة ميزات مستقبلية وتكامل مع أطراف ثالثة. مع مجموعة من الميزات القوية والسمات غير الوظيفية، سيكون النظام مؤهلًا لإدارة المشاريع المعقدة وتحسين إنتاجية الفريق بشكل عام.