

تم تصميمه من قبل:

**محمود أنس داؤودية**

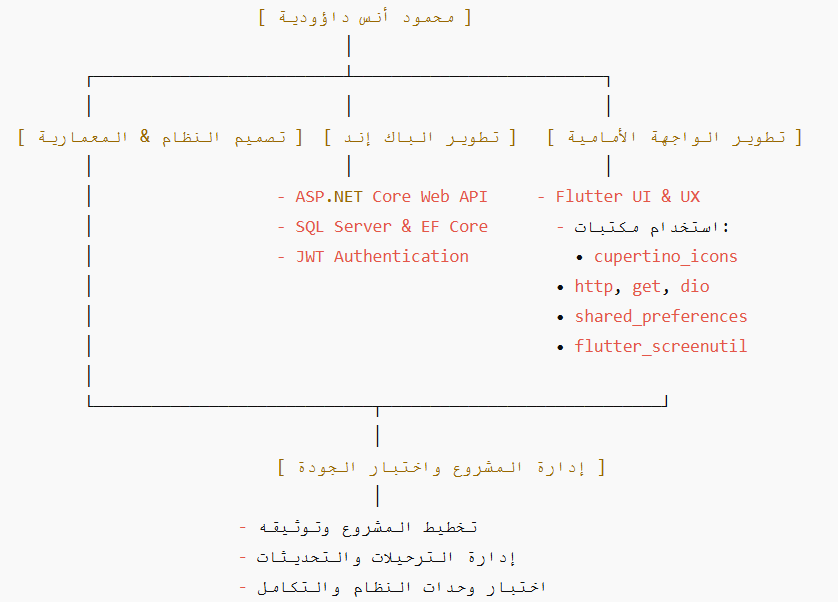
**اسم التدريب: flutter**

**المشرف: الاستاذ عبد اللطيف عيدة**

**التاريخ والمكان:**

ادارة القضايا الكترونيا

# ملخص المشروع ونطاقه

يهدف المشروع إلى إنشاء نظام رقمي متكامل لإدارة القضايا القانونية، بحيث يتمكن المستخدمون (سواء كانوا محامين أو عملاء) من تسجيل القضايا، متابعة الإجراءات القانونية، تحميل المستندات الخاصة بالقضية، وتقديم الشكاوى أو الاستشارات دون الحاجة للذهاب شخصياً إلى المحكمة. كما يتضمن المشروع ميزة تقديم شكاوى إلى الجهات الحكومية مع إمكانية إرفاق صور أو فيديوهات توضيحية للمشكلة.  
التعريف بالفريق  
**1تعريف الفريق بشكل نصي مع توزيع المهام  
اسم الفريق:** فريق MyCase  
**عضو الفريق: محمود أنس داؤودية  
دور المشروع:** المطور الرئيسي / مدير المشروع  
**المسؤوليات:  
تصميم النظام:** وضع الهيكلية المعمارية للمشروع واعتماد نمط التصميم (Fetchers) وتنظيم الطبقات وفق مبادئ Clean Architecture.  
**تطوير الباك إند:** إنشاء واجهات برمجة التطبيقات باستخدام ASP.NET Core، إعداد قاعدة البيانات باستخدام SQL Server وEntity Framework Core، وتطبيق نظام المصادقة باستخدام JWT.  
**تطوير الواجهة الأمامية:** بناء تطبيق Flutter مع استخدام مكتبات مثل cupertino\_icons، http، get، dio، shared\_preferences، وflutter\_screenutil لضمان تجربة مستخدم متكاملة ومتجاوبة.  
**اختبار النظام وضمان الجودة:** التأكد من تنفيذ جميع متطلبات المشروع من خلال اختبارات متكاملة والتحقق من استقرار النظام.  
**إدارة المشروع:** التخطيط الزمني، التوثيق، وإدارة الترحيلات والتحديثات لضمان توافق النظام مع احتياجات المستخدمين.رسم بياني تبسيطي لتوزيع المهام   
  
 **تعريف مختصر يبرز تعدد الأدوار  
فريق MyCase** هو مشروع تم تطويره بالكامل بواسطة **محمود أنس داؤودية**، الذي قام بتولي جميع المهام بدءاً من التصميم المعماري للنظام وتطوير واجهات برمجة التطبيقات (Back-End) باستخدام ASP.NET Core وSQL Server، مروراً بتطوير الواجهة الأمامية باستخدام Flutter، وانتهاءً بإدارة المشروع واختبار جودة التطبيق. يُظهر هذا التوزيع أن المشروع قد تم تنفيذه وفقاً لمبادئ التقسيم الواضح للمهام، مما يعكس قدرة العضو على تحمل عدة أدوار وتقديم منتج متكامل من حيث التقنية والإدارة.

# مراجعة الأدبيات

# **4. أمثلة على الاستفادة من المصادر والمشاريع**

# **تنظيم الكود والمعمارية:** من خلال دراسة مبادئ Clean Architecture والـ Fetchers، تم تبني نموذج يسمح بعزل واجهات المستخدم عن منطق الأعمال والبيانات، مما يسهل عملية الصيانة والتطوير المستقبلي.

# **تأمين النظام:** ساعدت المصادر المتعلقة بتطبيق JWT Authentication في بناء آلية مصادقة قوية، حيث تم تطبيق أفضل الممارسات لتحديد صلاحيات المستخدمين وتأمين البيانات الحساسة.

# **تحسين تجربة المستخدم:** اعتماد مكتبات Flutter المستندة إلى تجارب سابقة ومراجع تقنية أساسية مكن من بناء واجهة مستخدم متجاوبة وسلسة، مع دمج حلول مثل flutter\_screenutil لضمان توافق التصميم مع مختلف أحجام الشاشات. **إدارة البيانات والتحديثات:** الاعتماد على التوثيق الرسمي لـ Entity Framework Core وSQL Server مكن من تبني نهج Code-First مع استخدام الترحيلات، مما أدى إلى نظام إدارة بيانات متكامل ومتجدد بسهولة مع تحديث متطلبات المشروع.

# المشكلة والتحديات (الدافع)

# **أ. المشكلة الأساسية**

# يعاني قطاع الإدارة القانونية من عدد من التحديات الأساسية تؤثر على كفاءة العمل وجودة الخدمات المقدمة، وتتجلى أبرزها في:

# **الإجراءات الورقية والتقليدية:** تعتمد الكثير من الجهات القانونية والمحاكم على الأنظمة اليدوية أو الورقية في إدارة القضايا، مما يؤدي إلى بطء الإجراءات وزيادة احتمال الخطأ.

# **نقص التكامل الرقمي:** عدم وجود نظام متكامل يجمع كافة عمليات تسجيل وإدارة القضايا، مما يصعب متابعة الإجراءات القانونية وتحميل المستندات وإدارة البيانات بشكل مركزي.

# **التواصل بين الأطراف:** وجود فجوة في التواصل بين المحامين والعملاء، إضافة إلى صعوبة متابعة المستجدات في القضية بشكل فوري.

# **أمن البيانات:** التحديات المرتبطة بتأمين المعلومات القانونية الحساسة، وعدم توفر آليات فعالة لتحديد صلاحيات المستخدمين وتوفير الحماية المناسبة.

# الحل

# **أ. وصف الحل**

# يقوم مشروع **MyCase** بتقديم حل رقمي متكامل لإدارة القضايا القانونية يهدف إلى معالجة كافة أوجه القصور المذكورة أعلاه، وذلك من خلال:

# **إنشاء منصة متكاملة:** نظام واحد يجمع عمليات تسجيل القضايا، إدارة المستندات، تقديم الشكاوى والاستشارات، ومتابعة الإجراءات القانونية.

# **تحسين سرعة وكفاءة الإجراءات:** اعتماد واجهات برمجة تطبيقات (API) متطورة باستخدام ASP.NET Core وEntity Framework Core لتقليل زمن الاستجابة وتحسين تجربة المستخدم.

# **تعزيز الأمان:** تطبيق نظام JWT Authentication لتأمين البيانات وتحديد صلاحيات المستخدمين بدقة.

# **التكامل مع حلول حديثة:** استخدام تقنيات Flutter في الواجهة الأمامية مع مكتبات متخصصة (مثل dio وflutter\_screenutil) لضمان تجربة مستخدم متجاوبة وسلسة على مختلف الأجهزة.

# **إمكانية التوسع والصيانة:** اعتماد نمط تصميم Fetchers مع مبادئ Clean Architecture لعزل منطق الأعمال عن واجهة المستخدم، مما يسهل عملية الصيانة والتطوير المستقبلي.

# **ب. الأثر الاجتماعي والتجاري للمشروع**

# **الأثر الاجتماعي:**

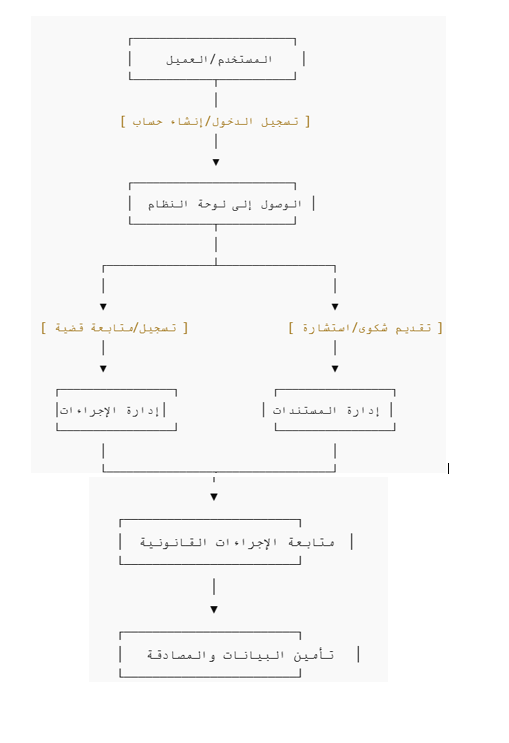
# **تحسين الوصول للخدمات القانونية:** يتيح للمستخدمين متابعة قضاياهم واستلام الاستشارات دون الحاجة للتواجد الفعلي، مما يسهم في تقليل العبء على المحاكم وتحقيق العدالة الرقمية.

# **تعزيز الشفافية:** يوفر النظام آلية متابعة دقيقة للإجراءات القانونية، مما يعزز الثقة بين الأطراف ويقلل من التأخيرات والخلل في العمليات.

# **الأثر التجاري:**

# **زيادة الكفاءة وتقليل التكاليف:** يقلل النظام من الحاجة للإجراءات اليدوية، مما يوفر وقت وتكاليف إضافية للجهات القانونية.

# **تحقيق ميزة تنافسية:** بتقديم نظام متكامل وحديث، يمكن للمكاتب القانونية والجهات الحكومية الاستفادة من أدوات متطورة في إدارة القضايا، مما يعزز مكانتهم في السوق.

يمكن توضيح آلية عمل النظام عبر مخطط تدفّق مبسط على النحو التالي:  


# المنهجية والتقنيات المستخدمة

**2. التقنيات المستخدمة في الباك إند**

**أ. ASP.NET Core**

* **وظيفتها:** إطار عمل قوي لبناء واجهات برمجة التطبيقات (Web API) عالية الأداء.
* **الفائدة:** يتيح التعامل مع الطلبات المتعددة بكفاءة، ويوفر بنية متينة لتطبيقات الويب الحديثة.

**ب. Entity Framework Core**

* **وظيفتها:** أداة ORM (Object-Relational Mapping) لإدارة قواعد البيانات من خلال نموذج Code-First.
* **الفائدة:** تسهيل إنشاء وتحديث الجداول دون الحاجة لكتابة استعلامات SQL يدوياً، مع إمكانية التعامل مع الترحيلات (Migrations) لتحديث قاعدة البيانات بسهولة.

**ج. SQL Server**

* **وظيفتها:** قاعدة بيانات علائقية لتخزين كافة البيانات الخاصة بالقضايا، المستخدمين، المستندات، وغيرها.
* **الفائدة:** توفر أداءً عالياً وأماناً متقدماً للبيانات.

**د. JWT Authentication**

* **وظيفتها:** نظام مصادقة لتأمين الواجهة باستخدام رموز JSON Web Token.
* **الفائدة:** يضمن تحديد الصلاحيات للمستخدمين (مثل محامي أو عميل) وتوفير حماية قوية للبيانات والعمليات.

**هـ. DTOs (Data Transfer Objects)**

* **وظيفتها:** كائنات تستخدم لنقل البيانات بين طبقة الواجهة الأمامية والباك إند.
* **الفائدة:** تنظيم البيانات المتبادلة وضمان عدم تسرب معلومات غير مرغوب فيها، مما يحسن من أمان التطبيق.

**و. Services**

* **وظيفتها:** فصل منطق الأعمال في خدمات مستقلة.
* **الفائدة:** تحسين نظافة الكود وسهولة الصيانة وإعادة الاستخدام.

**ز. Migrations**

* **وظيفتها:** إدارة التحديثات والتغييرات في قاعدة البيانات باستخدام Entity Framework Core.
* **الفائدة:** تسهيل تتبع التعديلات على قاعدة البيانات وتنفيذها بدون الحاجة للتدخل اليدوي.

**4. التقنيات والمكتبات المستخدمة في التطبيق المبني بفلاتر**

في جانب الواجهة الأمامية باستخدام Flutter، تم استخدام المكتبات التالية:

**أ. cupertino\_icons: ^1.0.8**

* **الشرح:** توفر أيقونات على نمط iOS (Cupertino).
* **الفائدة:** تحسين تصميم الواجهة وتوفير مظهر متناسق مع أجهزة iOS.

**ب. http: ^1.3.0**

* **الشرح:** مكتبة لإجراء طلبات HTTP.
* **الفائدة:** تسهيل عملية الاتصال بواجهة برمجة التطبيقات (API) لجلب وإرسال البيانات.

**ج. get: ^4.7.2**

* **الشرح:** مكتبة لإدارة الحالة (State Management) والتنقل (Routing).
* **الفائدة:** تبسيط تنظيم الكود وتحسين تجربة المستخدم من خلال تقليل التعقيدات في إدارة الحالة.

**د. dio: ^5.8.0+1**

* **الشرح:** مكتبة متقدمة للتعامل مع طلبات HTTP.
* **الفائدة:** توفر ميزات إضافية مثل التحكم في المهلات وإعادة المحاولة وتحميل البيانات بكفاءة عالية.

**هـ. shared\_preferences: ^2.5.3**

* **الشرح:** مكتبة لحفظ البيانات محلياً على الجهاز (مثل إعدادات المستخدم أو بيانات الجلسة).
* **الفائدة:** تسهيل عملية تخزين واسترجاع البيانات البسيطة دون الحاجة لاستخدام قواعد بيانات معقدة على الجوال.

**و. flutter\_screenutil: ^5.9.3**

* **الشرح:** مكتبة لتصميم واجهات متجاوبة تناسب مختلف أحجام الشاشات.
* **الفائدة:** ضمان تناسق المظهر وتجربة مستخدم مثالية عبر مختلف الأجهزة.

**5. نمط التصميم (Design Pattern) المستخدم**

**نمط التصميم: Fetchers**

* **الشرح:**  
  يعتمد هذا النمط على فصل مكونات تحميل البيانات (Fetching) عن باقي طبقات التطبيق، مما يسمح بفصل منطق جلب البيانات عن منطق العرض (UI) والتفاعل.
* **الفائدة:**
  + **تنظيم الكود:** تقسيم المشروع إلى وحدات واضحة يسهل صيانتها وتطويرها.
  + **قابلية إعادة الاستخدام:** يمكن إعادة استخدام مكونات Fetchers في أجزاء متعددة من التطبيق دون تكرار الكود.
  + **اختبار الوحدة:** تسهيل عملية اختبار كل وحدة على حدة بفضل فصل المنطق عن العرض.

كما يستلهم المشروع من مبادئ الـ **Clean Architecture** التي تهدف إلى تقسيم التطبيق إلى طبقات واضحة، مما يعزز من الاستقلالية بين منطق الأعمال وطبقة العرض والبيانات.

.

# الجوانب المبتكرة في مشروع

# **1. منصة متكاملة لإدارة القضايا القانونية**

# **توحيد العمليات:** يجمع النظام كافة العمليات القانونية من تسجيل القضايا وإدارة المستندات إلى تقديم الشكاوى والاستشارات في منصة واحدة متكاملة، مما يُقلل من التعقيد والازدواجية في الإجراءات.

# **متابعة الإجراءات دون الحاجة للحضور الشخصي:** يوفر النظام إمكانية متابعة القضايا وإجراءاتها عبر واجهة رقمية، مما يُسهم في تخفيف الضغط على المحاكم وتقليل وقت الانتظار.

# **2. استخدام تقنيات حديثة ومتطورة**

# **ASP.NET Core مع Entity Framework Core:** اعتماد إطار عمل ASP.NET Core في بناء واجهة برمجة التطبيقات يُحقق أداء عالي وقدرة على التعامل مع طلبات متعددة بكفاءة، بينما تُسهّل Entity Framework Core إدارة قاعدة البيانات باستخدام منهجية Code-First.

# **JWT Authentication:** تطبيق نظام مصادقة JWT يضمن أمان البيانات ويحدد صلاحيات المستخدمين بدقة، مما يُعزز من حماية المعلومات الحساسة.

# **SQL Server:** اختيار SQL Server كقاعدة بيانات أساسية يضمن استقرار وأداء عالي لتخزين البيانات، مع دعم قوي للترحيلات والتحديثات.

# **3. تصميم معماري متقدم وفق مبادئ Clean Architecture**

# **عزل منطق الأعمال:** يعتمد المشروع على نمط التصميم الذي يفصل بين طبقات العرض، منطق الأعمال والبيانات. هذا التصميم يعزز من قابلية التوسع والصيانة ويُسهّل عملية اختبار كل وحدة على حدة.

# **اعتماد نمط Fetchers:** يتم استخدام هذا النمط لفصل عمليات جلب البيانات عن واجهة المستخدم، مما يُحسّن من تنظيم الكود ويساهم في بناء نظام مرن يمكن تطويره وتحديثه بسهولة.

# **4. واجهة مستخدم متجاوبة باستخدام Flutter**

# **تصميم تفاعلي:** بفضل استخدام Flutter ومكتبات مثل flutter\_screenutil، يتم ضمان توافق التصميم مع مختلف أحجام الشاشات، مما يُوفر تجربة مستخدم متكاملة وسلسة.

# **اعتماد مكتبات متخصصة:** استخدام مكتبات مثل dio وget يُسهل عمليات التواصل مع الواجهة الخلفية، مما يُقلل من زمن الاستجابة ويُحسن من أداء التطبيق.

# **5. ابتكار في تقديم الخدمات القانونية الإلكترونية**

# **تقديم الشكاوى والاستشارات:** يتيح النظام للمستخدمين تقديم شكاوى واستشارات قانونية مباشرة عبر المنصة مع إمكانية إرفاق الصور أو الفيديوهات التوضيحية، مما يُعتبر خطوة مبتكرة في تسهيل الوصول للخدمات القانونية.

# **إدارة المستندات بشكل مركزي:** دمج آلية تحميل وتنزيل وتحديث المستندات داخل النظام يُساعد في تنظيم الوثائق القانونية والتقليل من الأخطاء الناتجة عن الإجراءات الورقية التقليدية.

# إمكانية التطبيق والتسويق

# **1. إمكانية التطبيق**

# **أ. تلبية احتياجات القطاع القانوني**

# **حل شامل:** يقدم المشروع منصة متكاملة لإدارة القضايا القانونية تشمل تسجيل القضايا، إدارة المستندات، تقديم الشكاوى والاستشارات، مما يُلبي احتياجات الجهات القانونية سواء كانت حكومية أو خاصة.

# **تحسين كفاءة الإجراءات:** من خلال تقليل الاعتماد على الإجراءات الورقية وتبسيط متطلبات المتابعة، يُمكن للمستخدمين متابعة القضايا والإجراءات بشكل أسرع وأكثر دقة.

# **أمان البيانات:** اعتماد تقنيات متقدمة مثل JWT Authentication يوفر طبقة حماية قوية للمعلومات القانونية الحساسة، مما يُعزز الثقة في النظام.

# **ب. سهولة التحديث والتطوير**

# **تصميم معماري مرن:** استخدام مبادئ Clean Architecture مع نمط Fetchers يتيح إمكانية صيانة وتطوير النظام بسهولة، وبالتالي يمكن التوسع لاحقًا بإضافة مميزات جديدة.

# **قابلية التوسع:** تم تصميم النظام بحيث يكون قابلًا للتوسع سواء من حيث عدد المستخدمين أو الوظائف الإضافية التي قد تتطلبها تطورات القطاع القانوني مستقبلاً.

**2. إمكانية التسويق وتحويل المشروع إلى منتج تجاري**

**أ. القيمة المضافة للمشروع**

* **تكامل الخدمات القانونية:**  
  بدمج كافة الخدمات القانونية في منصة واحدة، يُمكن تقديم تجربة متكاملة للمستخدم مقارنة بالحلول الجزئية أو التقليدية، مما يُشكّل قيمة مضافة كبيرة.
* **توفير الوقت والتكاليف:**  
  يُسهم النظام في تقليل الوقت اللازم لاتخاذ الإجراءات القانونية وتقليل التكاليف الإدارية، مما يرفع من كفاءة العمل ويزيد من جاذبية المنتج للمؤسسات.
* **تجربة مستخدم متقدمة:**  
  استخدام تقنيات حديثة في الواجهة الأمامية (مثل Flutter) يضمن تجربة تفاعلية وسلسة، مما يُعزز من رضا المستخدمين ويعمل على تمييز المنتج عن المنافسين.  
  التكلفة التقديرية   
  **. نظرة عامة على التكاليف المتوقعة**يُعتمد تقدير الميزانية على عدة عوامل رئيسية منها:  
  **تكاليف التطوير البرمجي:** تشمل تطوير الواجهة الخلفية باستخدام ASP.NET Core وقاعدة البيانات باستخدام SQL Server، وتطوير الواجهة الأمامية باستخدام Flutter.  
  **تكاليف التصميم والاختبار:** تشمل إعداد التصاميم والواجهات (UI/UX)، وتنفيذ اختبارات الوحدة والتكامل لضمان جودة المنتج.  
  **تكاليف الصيانة والدعم:** تشمل التحديثات الدورية، وتصحيح الأخطاء، ودعم المستخدمين بعد الإطلاق.  
  **التكاليف التشغيلية والتسويقية:** تشمل استضافة النظام، ونفقات التسويق للترويج للمنتج في السوق.

# **2. خطة الميزانية حسب مراحل المشروع**

# **أ. مرحلة التصميم والتحليل**

# **تحليل المتطلبات وتصميم النظام:**

# العمل على إعداد المخططات المعمارية وتصميم الواجهات (UI/UX).

# قد يتطلب ذلك استثماراً منخفضاً نسبيًا إذا كان التطوير من قِبَل مطور واحد أو فريق صغير.

# الفئة المستهدفة (المستخدمون)

# **1. الفئة القانونية**

# **المحامون والمستشارون القانونيون:** يعاني المحامون من تحديات تتعلق بإدارة ملفات القضايا ومتابعة الإجراءات القانونية بشكل يدوي أو باستخدام أنظمة غير متكاملة.

# **المشكلة:**

# ازدواجية البيانات وتأخر الإجراءات نتيجة الاستخدام الورقي أو الأنظمة الجزئية.

# صعوبة متابعة تحديثات القضايا وتنسيق الجلسات والمستندات.

# **كيف يخدمهم المشروع:**

# يوفر النظام منصة رقمية موحدة لتسجيل القضايا، متابعة الإجراءات، إدارة المستندات، وتقديم الاستشارات، مما يساعد على تقليل الوقت والجهد وتحسين دقة العمل.

# يعزز الأمان من خلال تقنيات حديثة (JWT Authentication) تحمي المعلومات الحساسة.

# **2. العملاء والموكلين**

# **الأفراد أو الشركات الذين يحتاجون لخدمات قانونية:** غالباً ما يواجه العملاء صعوبة في متابعة قضاياهم وفهم الإجراءات القانونية التي تخضع لها ملفاتهم، مما يؤدي إلى شعورهم بعدم الشفافية والارتباك.

# **المشكلة:**

# قلة التواصل المباشر والمستمر مع المحامين وعدم القدرة على متابعة تحديثات القضايا.

# الاعتماد على المعلومات غير المنظمة والطرق التقليدية في تقديم الشكاوى أو طلب الاستشارات.

# **كيف يخدمهم المشروع:**

# يقدم واجهة مستخدم مبسطة وسهلة الاستخدام تتيح للعملاء متابعة قضاياهم بشكل فوري والحصول على استشارات قانونية إلكترونية.

# إمكانية تحميل المستندات وإرفاق الصور أو الفيديوهات عند تقديم الشكاوى، مما يساهم في توضيح المشكلة وتسريع عملية المتابعة.

# **3. الجهات القانونية والإدارية**

# **المكاتب القانونية والجهات الحكومية:** تواجه الجهات القانونية صعوبة في تنظيم وإدارة البيانات القانونية بشكل متكامل وموحد، مما يؤثر على سرعة وكفاءة تقديم الخدمات القانونية للمواطنين

# إدارة المخاط وتحليل SWOT

# **1. إدارة المخاطر**

# **أ. تحديد المخاطر المحتملة**

# **المخاطر اللوجستية:**

# **تأخير في تسليم المكونات البرمجية:** مشاكل في الجدولة أو تسليم الكود في الوقت المحدد.

# **مشكلات في التكامل بين الواجهة الخلفية والأمامية:** تحديات تقنية قد تؤدي إلى تأخير في دمج الأنظمة المختلفة.

# **الموارد البشرية:**

# **اعتماد المشروع على مطور أو عضو واحد:** احتمال تعرض المشروع لتأثيرات سلبية في حالة غياب أو توقف المطور.

# **نقص الخبرة في بعض التقنيات الحديثة:** قد يؤدي إلى صعوبة في حل بعض المشكلات التقنية أو التعامل مع أخطاء غير متوقعة.

# **التمويل:**

# **عدم كفاية الميزانية المخصصة:** إمكانية عدم توافر التمويل اللازم لتغطية كافة مراحل التطوير، الاختبار، والصيانة.

# تأخير التمويل أو عدم تحقيق العائد المتوقع في المراحل الأولية.

# **الظروف البيئية والتكنولوجية:**

# **التغيرات السريعة في التقنيات:** قد يتطلب المشروع تحديثات مستمرة لمواكبة التطورات التكنولوجية.

# **تغيرات في اللوائح القانونية والتنظيمية:** تأثيرها على كيفية إدارة وحماية البيانات القانونية

**ب. استراتيجيات إدارة المخاطر وخطط الطوارئ**

1. **تأمين الجداول الزمنية والتخطيط المرن:**
   * **إعداد خطط بديلة لكل مرحلة من مراحل التطوير:**  
     تخصيص هوامش زمنية للتأخير غير المتوقع.
   * **استخدام أدوات إدارة المشاريع لمتابعة التقدم:**  
     مثل Trello أو Jira لتتبع المهام والموارد.
2. **توسيع قاعدة الموارد البشرية:**
   * **التعاون مع مطورين مستقلين أو فرق خارجية عند الحاجة:**  
     لتأمين استمرارية العمل في حالة غياب العضو الرئيسي.
   * **تنفيذ برامج تدريبية قصيرة لتحسين المهارات التقنية:**  
     خاصة في التقنيات التي تعتمد عليها المشروع.
3. **التحكم في التكاليف وتأمين التمويل:**
   * **إعداد خطة مالية مرنة:**  
     تشمل مراحل متعددة مع تحديد الأولويات والميزانيات الطارئة

اليك بعضا من الامور التي كانت تحصل خلف الكواليس التي تبين لكم اهمية التخطيط الورقي قبل بدء التنفيذ   
  
