

مقدمة

مقدمة:

يشهد العالم اليوم تطورا هائلا في جميع المجالات وخاصة في مجال العلوم الحاسوبية حيث يشكل الحاسوب وعلومه عصب هذا التطور وكأي اختراع بشري بعد تداوله تتجاذبه الأيدي وتبنى عليه الأفكار العديدة ليكون أداة مطوعة ومرنة في أيدي مستخدميها، علوم الحاسب ببساطة هي لغة عصر الحديث عصر الثورة الصناعية في جيلها الرابع المتمثلة في التطور التقني حيث أصبح التفكير في إيجاد حلول لأي مشكلة لا يخلو من البرمجة اليوم.

منذ منتصف التسعينيات غدت صناعة الويب كواحدة من أسرع الصناعات نمواً، بقدر ما هنالك أدوات ومنصات معتبرة لصناعة الويب والتي قد تشكل عبئا ماليا للمستخدم فإنه بالمقابل هنالك الكثير من الخيارات المجانية والمفتوحة المصدر بنفس الوقت.

انطلاقاً من " إيجاد حلول للمستخدمين والتواصل عن بعد "

كان لابد من منصة جامعة كلا من الأشخاص والشركات لفرص العمل.

تشير الإحصائيات الى أن عدد العالمين قد تزداد بنحو 2.4 ملايين في فترة استخدام الموقع الإلكتروني لإيجاد فرص عمل وقد تصل في عام 2030 إلى 9.18 مليون عامل وذلك بسبب توفر فرص العمل المناسبة وتأتي فكرة المشروع موقع الإلكتروني لتقديم فرص عمل على انضمام أكبر عدد من الشركات والتي تضم فئات وأصناف وتخصصات مختلفة وتمكنها أيضا من طلب اشخاص ذو مهارات وكفاءات عالية من جميع أطراف العالم أصبح من الممكن تأمين موظفين للعمل بسرعة كبيرة دون هدر في إدارة المهام والعمل هذا هو عمل الموقع نشر فرص عمل تخدم الشواغر الموجودة في أي شركة وبطريقة سريعة وبكفاءات موظفين عالية وكذلك الأمر لدى متصفح الموقع أصبح بإمكانه طلب عمل مناسب لخبراته وقدراته بمستوى المالي الذي يرغب به هذا الموقع يقوم على البرمجة،

يظن البعض أن دور المبرمج ينحصر في كتابة مجموعة من الأكواد البرمجية وحسب ولكن هذا غير صحيح لأن مهامه أعمق من ذلك بكثير مهما كانت تبدو بسيطة.

تعتمد البرمجة على التفكير النقدي للخروج بحلول المناسبة للمشكلات فيقوم المبرمج بتقسيم المشكلة الى اجزاء صغيرة ويوظف التفكير الإبداعي ومهارة حل المشكلات للتعامل معها بشكل سليم

◀ مستخدم المشروع

◀ الشركات التي تطلب موظفين

◀ الباحثين عن فرص عمل متاحة أو طلب عمل محدد

◀ متصفحين الموقع

ملخص عن المشروع:

فرص عمل

تهدف المنصة إلى توفير فرص العمل الموجه إلى كافة الفئات ، حيث يسعى في المنصة على انتقاء الشركات والمنظمات والمؤسسات المانحة لفرص العمل من أجل توفير فرص عمل مميزة لجميع الدرجات العلمية والتي تتوافق مع شغف المتابعين وميولهم، إذ يتم نشر فرص عمل حسب خبرات محددة مع طبيعة العمل وكافة التفاصيل والمتطلبات من قبل الشركات بالإضافة لإمكانية طرح أعمال خارجية من قبل المستخدمين بما يتناسب مع شروط المنصة لحفظ حقوق الموظفين من خلالها وضمان الاتصال الكامل مع بين المستخدمين ، كما نسعى أيضا إلى توفير فرص العمل الجزئية أو فرص العمل عن بعد Freelance أو فرص عمل من المنزل، التي قد تضيف قيمة إضافية إلى سيرتك الذاتية . اذا كنت تبحث عن فرصة عمل بالخارج أو فرصة عمل للخريجين فأنت في المكان الصحيح.

حيث اتاحة البحث عن آخر فرص العمل وعلى الراغبين بالانضمام إلى المنصة تسجيل حساب مؤكد وإدخال البيانات الرئيسية مع إمكانية ارفاق السير الذاتية والخبرات والاعمال السابقة والمهارات الفردية ومكان الإقامة الذي يعزز من ترشيح الفرص الأقرب والمناسبة للمستخدم مع اخر الاخبار والمتابعة من قبل الشركات والأصدقاء

فهو ليس فقط مجرد اطروحات توظيف وانما شبكة اجتماعية تتضمن أي مستخدم رياضي متابع لأخر الاعمال والشركات.

تتيح المنصة طرح توظيف بنوع "مشروع" من قبل المستخدمين حيث يتم نشر صيغة مشروع من قبل المستخدم المستفيد من استقبال أكبر عدد من المستخدمين للعمل ضمن مشروع قد قام بطرحه دون الحاجة إلى العمل بشكل دائم وثابت لدى الشركات

الباب الأول:

مدخل إلى البنية العامة للمشروع

الفصل الأول:

لغات البرمجة المستخدمة

مقدمة

سنتناول في هذا الفصل اللغات البرمجية المستخدمة في بناء وتطوير منصة الويب الخاصة، بالإضافة إلى توضيح وتبيان المواضيع التي تم استخدامها ضمن المشروع ولماذا تم اختيارهم عوضاً عن غيرهم من اللغات

1-1 لغة C#

تمثل الحجر الرئيسي في بناء وتطوير المنصة بشكل عام حيث تم اعتمادها كلغة أساسية لتطوير الشيفرة البرمجية الخاصة بالمشروع ككل. قمنا باعتماد الإصدار 7.3/8.0 من اللغة.

تتميز هذه اللغة بكونها من أكثر اللغات دعماً وانتشاراً في عالم البرمجة، حيث تم طرحها من قبل Microsoft عام 2000 كلغة غرضية التوجه متعددة الاستعمالات، وما لبثت أن أصبحت لغة لبرمجة لكافة المنصات وأنظمة التشغيل وإطارات العمل فهي تدعم التنسيقات المختلفة من Script syntax و Syntax Basic مما يجعلها قابلة الترجمة وسهولة استخدامها لكافة المبرمجين القادمين من لغات مختلفة

ولكن لماذا تم اعتمادها ضمن المنصة بشكل أساسي!!

تعتبر الآن الـ C# هي Cross platform، أي بناء موقع يمكن بسهولة تحويله لتطبيقات الهواتف أو أجهزة الحاسب بالإضافة للذكاء الصناعي والدعم،

في تعتمد على "Don't repeat yourself" وقابلية تطبيق العديد من الـ "Design Pattern" ولعمل ضمن فريق موزع فهي بطبيعتها "Full OOP"

1-2 لغة JavaScript

استخدمت بشكل رئيسي لكتابة السكريبتات والأكواد الخاصة بصفحات الويب والتي سيتم تنفيذها ضمن جهاز المستخدم "Client"، كأحداث الأزرار أو إرسال الطلبات غير المتزامنة إلى السيرفر "Ajax" وغيرها من العمليات البسيطة، حيث قمنا باعتماد إصدار ES6 والذي يعد قفزة نوعية ضمن اللغة من ناحية الأداء وطريقة كتابة الكود.

وهي تعد الحجر الأساس للتعامل مع "DOM" (Document Object Model)

1-3 اللغة التوضيفية HTML 5 / HTML

تم اعتماد لغة HTML أو language markup Hypertext وذلك لكتابة الكود الأساسي لعناصر الواجهات ضمن موقع الويب كالأزرار والقوائم المنسدلة وغيرها من مكونات الصفحات، حيث قمنا بالاعتماد على الإصدار الخامس الذي يضم العديد من المزايا كدعم المواقع الجغرافية وملفات الصوت والفيديو وغيرها.

1-4 لغة CSS / CSS3

حيث تم تضمينها جنباً إلى جنب مع أكواد لغة HTML وذلك لإعطاء الموقع تنسيق خاص بالعناصر المستخدمة ولجعله تفاعلي "Responsive" ضمن شاشات الهواتف الذكية. قمنا باعتماد الإصدار الأخير من لغة CSS وهو الإصدار الثالث الذي يضم العديد من المزايا المتعلقة بالحركات "animations" وتحسينات على الإصدارات السابقة.

1-5 لغة الاستعلامات SQL

SQL (لغة الاستعلام الهيكلية) هي لغة برمجة تستخدم للتواصل مع البيانات المخزنة في نظام إدارة قواعد البيانات العلائقية. يشبه بناء جملة SQL اللغة الإنجليزية، مما يجعل من السهل نسبياً الكتابة والقراءة والتفسير.

تستخدم العديد من أنظمة RDBMS SQL (وأشكال مختلفة من SQL) للوصول إلى البيانات الموجودة في الجداول. على سبيل المثال، يعد SQLite نظامًا لإدارة قواعد البيانات العلائقية. يحتوي SQLite على الحد الأدنى من مجموعة أوامر SQL (والتي هي نفسها في جميع أنظمة إدارة قواعد البيانات (RDBMS)). قد تستخدم أنظمة RDBMS الأخرى متغيرات أخرى.

(غالبًا ما يتم نطق SQL بإحدى طريقتين. يمكنك نطقه من خلال نطق كل حرف على حدة مثل "S-Q-L"، أو نطقه باستخدام كلمة "تكملة").

6-1 الاستعلامات NoSQL Query

هي مثيلة لـ Sql ولكن تتعامل مع نوع قواعد بيانات خاص حيث تجري العمليات على ملفات أقل تعقيداً مما يجعلها السبب الرئيسي لاستخدامها ضمن بنية اللغات الأساسية التي تعتمد عليها المنصة فهي تتيح التعامل مع الدتا الكبيرة والزمن الحقيقي من تطبيقات.

7-1 SASS

ماهي إلا تغليف للغة الـ CSS التي تقوم بتسهيل تكرار التأثيرات والموازنة والعديد من الميزات التي تجعل كتابة الملفات الكبيرة من CSS سهل وممتع.

الفصل الثاني:

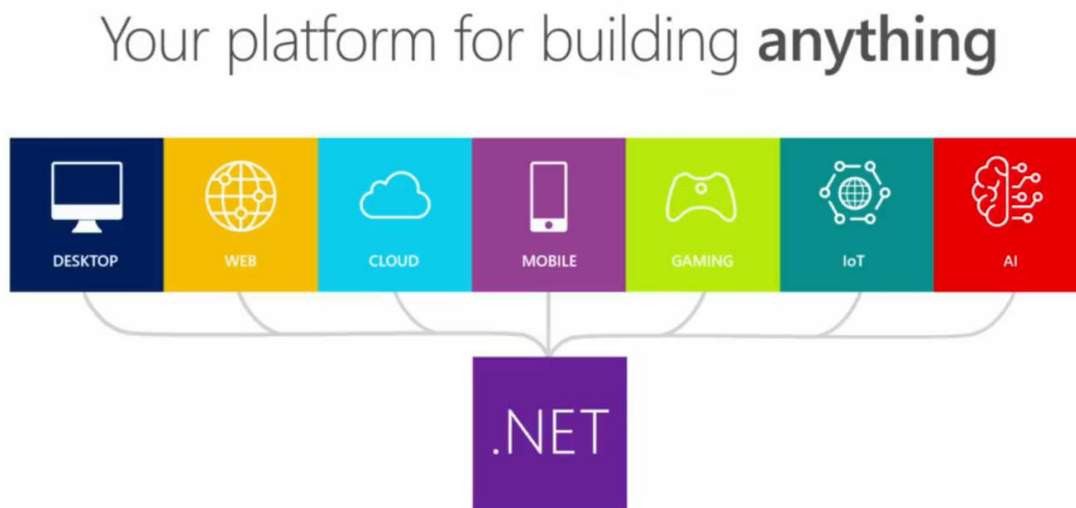
التقنيات الأساسية المستخدمة

مقدمة

سنتناول في هذا الفصل التقنيات الأساسية المستخدمة في بناء وتطوير منصة الويب الخاصة، بالإضافة إلى توضيح وتبيان المواضيع التي تم استخدامها ضمن المشروع ولماذا تم اختيارهم عوضاً عن غيرهم من تقنيات.

1-2 إطار عمل .net core

يعتبر إطار العمل الأحدث من Microsoft، مفتوح المصدر ومجاني بشكل كامل، كما يتميز بدعمه للعديد من أنظمة التشغيل مثل Windows, Linux, MacOS فهو متعدد المنصات "Cross-platform" كما موضح بالشكل رقم (1-2)، إضافة إلى احتوائه العديد من المكاتب الجديدة والمصممة للعمل بكفاءة عالية ضمن أي نظام تشغيل يتم دعمه من قبل إطار العمل السابق، حيث يعتبر حجر الأساس في تطوير المشروع وبنائه.



الشكل (1-2)

webApplication 2-2

تطبيقات الويب، مصطلح جديد في عالم التقنية، وهو يصف بشكل خاص تلك البرمجيات التي يقدمها مطورو الويب) بالإنجليزية(Web Developers : عبر تقنيات الويب 2 التي ظهرت فكرتها مع مطلع العام 2006.

تُقدم تطبيقات الويب للمستخدم برمجيات صغيرة تقوم بعمل مشابه لعمل البرامج الموجودة على أجهزة الحاسوب أو الهاتف الذكي كما موضح بالشكل رقم (2-2). من الأمثلة الشهيرة لتطبيقات الويب: بريد الويب، الويكي، المدونات، منتديات الإنترنت.



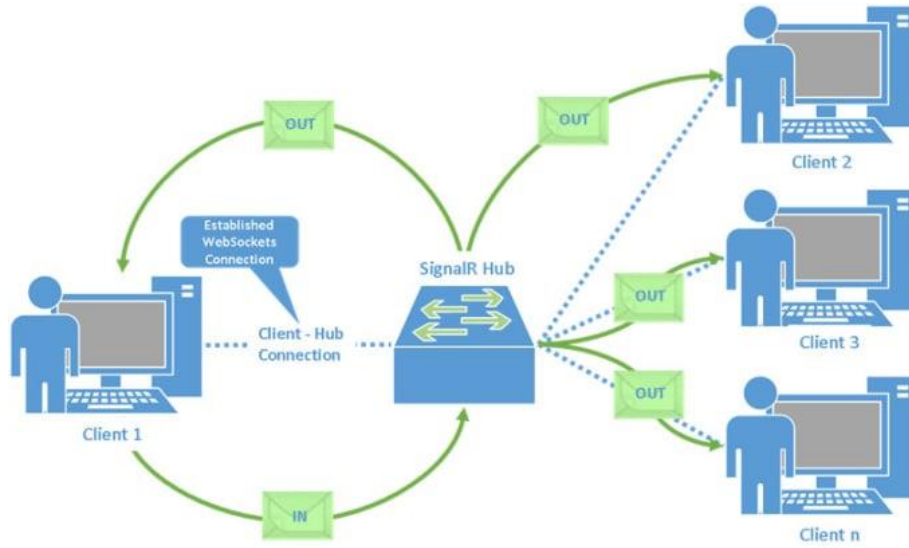
الشكل (2-2)

لماذا webApplication؟

تعتبر تطبيقات الويب اليوم هي هدف معظم المبرمجين فهم يرون أنها تقدم للمستخدمين المبتدئين والمحترفين الواجهة البسيطة والتقنية المتفوقة والوصول السريع، فهي متوفرة عبر رابط إنترنت ويمكن الحصول عليها في أي وقت ومن أي مكان، كما أن أداء معظم تطبيقات الويب بدأ يتحسن وبدأت تقدم خدمات متميزة. وفي حين أن معظم تطبيقات الويب مجانية ولا تحتاج إلا إلى رابط التطبيق.

تطبيق الـ RealTime

الهدف الأول والرئيسي عند بناء منصة بحجم منصات التواصل الاجتماعي فإنها تكون قابلة للتحديث الانني للواجهات والتفاعل المباشر السبب الذي أدى إلى دراسة المشروع والتقنيات بعناية من استخدام بنيات برمجية بنيت ذاتيا تقوم بتحقيق الاتصال كالا WebSocket،SignalR وبناءه اتصال بغطاء TCPضمن معايير أمنة ببيئة موحدة الشكل رقم (2-ب) يوضح عملية التوزيع ضمن الـ HUBs



الشكل (2-ب)

ASP.NET Core 3-2

عبارة عن إطار عمل مفتوح المصدر وعالي الأداء ومتعدد الأنظمة الأساسية لبناء تطبيقات حديثة ومتصلة بالسحابة ومتصلة بالإنترنت. باستخدام ASP.NET Core، يمكنك: أنشئ تطبيقات وخدمات الويب، وتطبيقات إنترنت الأشياء (IoT)، وخلفيات الأجهزة المحمولة. استخدم أدوات التطوير المفضلة لديك على أنظمة التشغيل Windows و macOS و Linux. انشر على السحابة أو في مكان العمل.

وتشمل بعض الميزات:

- نظام التوجيه "Routing": حيث يتيح إمكانية تحويل الطلبات بناءً على الرابط "URL" إلى المتحكم "Controller" الملائم.
- ربط النماذج "Model binding": يقوم إطار العمل بعملية تعبئة البيانات القادمة من الواجهات في صفوف وكائنات برمجية ليتم استخدامها في الشيفرة البرمجية من قبل المطورين.
- حقن الاعتماديات "Dependency injection": يقوم على مبدأ عزل مكونات التطبيق ضمن صفوف يغلفها واجهات برمجية "Interfaces" بحيث يعتمد كل مكون على Interface خاصة بمكون آخر وتقوم مكتبة الحقن بإنشاء Object من هذا المكون وإعطائه لمكون الذي يعتمد عليه.
- إمكانية الاختبار "Testability": يتميز إطار العمل السابق بكون جميع مكوناته قابلة للاستبدال مما يتيح إمكانية اختبارها بشكل مستقل ومنفصل بشكل مؤتمت باستخدام مفهوم "Unit Test".

محرك الواجهات "Razor view engine": تضم ASP.NET Core MVC محرك خاص للواجهات يدعى "Razor" يتيح لنا إمكانية كتابة كود سي شارب ضمن واجهات الـ Html ويقوم عند التنفيذ بتفسير كود السي شارب وتحويله إلى كود Html, CSS, JavaScript يمكن لمتصفحات الويب المختلفة فهمه وعرضه للمستخدم.

تم إصدار ASP.NET Core لأول مرة في عام 2016

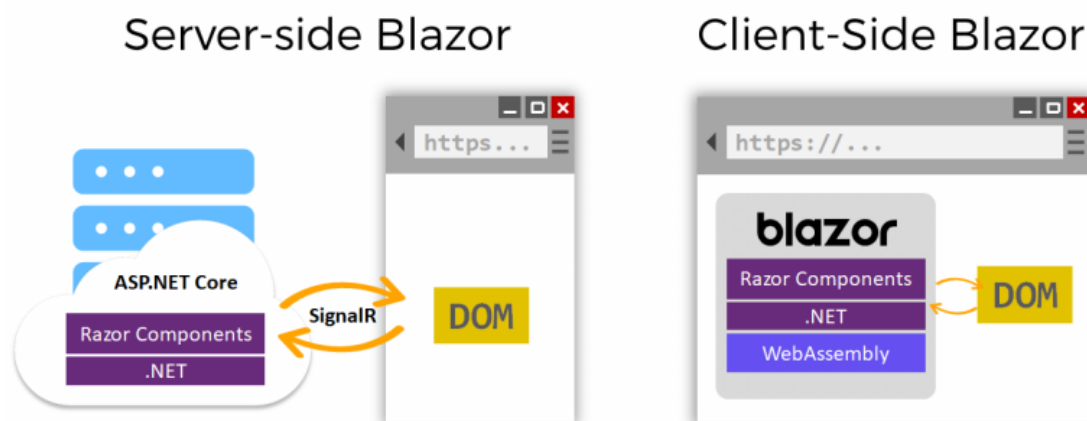
إصدار العمل 3.1 تم إصداره 2019-12-03 (حيث استخدم بالمشروع 3.1.5 من إصدار مكاتب وتحسينات مترافقة مع 3.1)

لماذا تم استخدام ASP.NET Core؟

تعد الآن هي من أسرع إطارات العمل والأكثر أماناً فهي تشمل الأمان العالي المقدم من مايكروسوفت بالإضافة التقوى التقني ودعم الكثير من الميزات وقابلية التطوير الذاتي والاستمرارية

Blazor 4-2

Blazor هي إطار عمل تعدل (SPA) وهي الصفحة الواحدة طورت من فريق ASP.NET. الهدف هي القدرة على كتابة تصاميم وتجربة المستخدم ثرية لواجهة مستخدم الويب باستخدام HTML و CSS و C# بدلاً من JavaScript - وهو أمر يحلم به العديد من المطورين الشكل رقم (2-3) يوضح المبدأ الأساسي بالعمل مع المتصفحات.



الشكل (1-4-2)

تقسم لعدة أنماط:

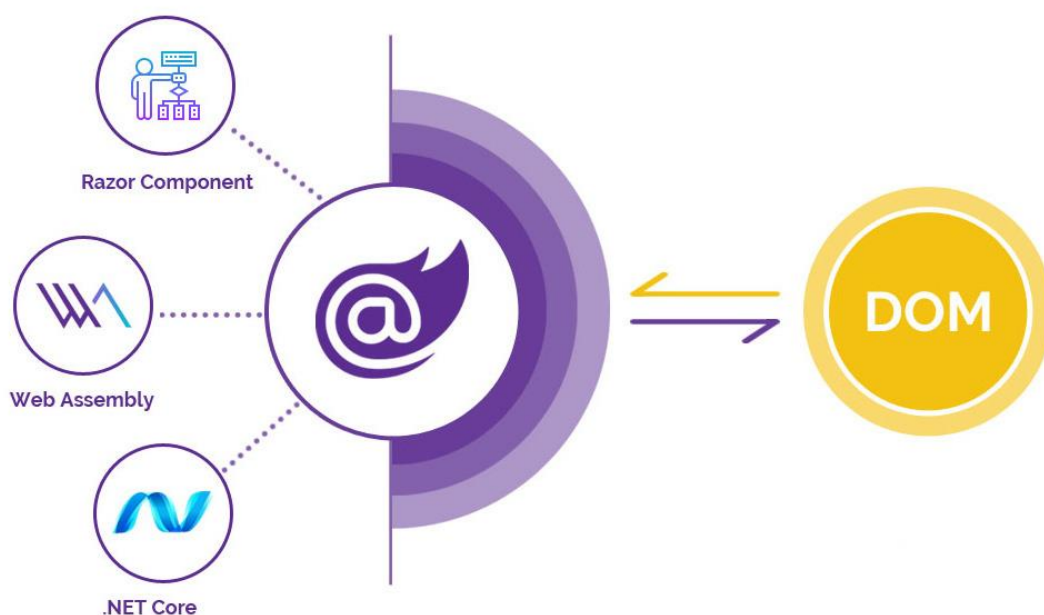
Blazor-ServerSide

تتم استضافة هذه التطبيقات على خادم ASP.NET Core بتنسيق ASP.NET Razor. تعمل الأجهزة العميلة البعيدة كعملاء خفيف المستوى، مما يعني أن الجزء الأكبر من حمل المعالجة يقع على الخادم. يقوم متصفح الويب الخاص بالعميل بتنزيل صفحة صغيرة وتحديث واجهة المستخدم الخاصة به عبر اتصال SignalR.

Blazor WebAssembly

تعد القفزة النوعية في مجال الويب والتطبيقات الحديثة.. اتفقنا مسبقاً انه تم استخدام البلزور لسرعة وتلافي ترجمة JavaScript فالأن بين أيدينا تطبيق مستقل يعمل بشكل أسرع بأضعاف من الـ Js ضمن حاسب العميل Client ودعم العديد من التقنيات وسهولة الـ RealTime وبختصار

هي: تطبيقات من صفحة واحدة يتم تنزيلها على متصفح الويب الخاص بالعميل قبل التشغيل. حجم التنزيل أكبر من حجم خادم Blazor، ويعتمد على التطبيق، وتتم المعالجة بالكامل على أجهزة العميل. ومع ذلك، يتمتع هذا النوع من التطبيقات بوقت استجابة سريع. كما يوحي اسمه، تتم كتابة إطار عمل جانب العميل هذا في WebAssembly ، بدلاً من JavaScript كما موضح بالشكل رقم (2-4-2). تم إصدار Blazor WebAssembly 3.2.0 في 19 تموز 2020.



الشكل (2-4-2)

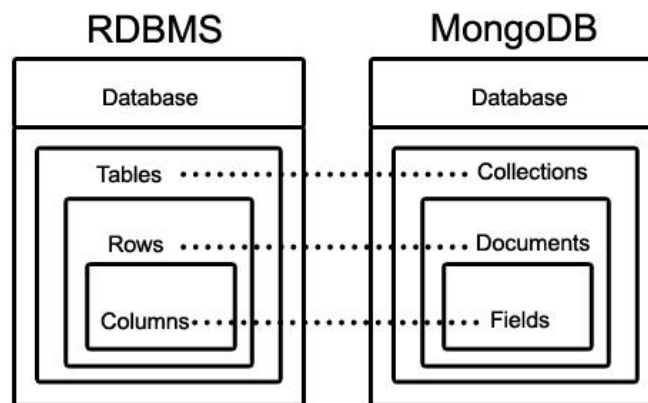
فهي تقنية حديثة ومتينة بأول إصدارها تم تطبيقها بالمشروع ضمن وحدة بنوية متكاملة فهي تتعاون مع الـ C# ضمنياً دون الدخول بتفاصيل التحويل للغة Assembly حيث:

يستطيع Blazor تشغيل كود C# من جانب العميل مباشرة في المتصفح، باستخدام WebAssembly. نظرًا لأنه حقيقي. NET يعمل على WebAssembly ، يمكنك إعادة استخدام التعليمات البرمجية والمكتبات من أجزاء من جانب الخادم لتطبيقك.

وهو إطار عمل للمراسلة في الوقت الفعلي RealTime. بمجرد اكتمال التنفيذ، يتم إرسال تغييرات واجهة المستخدم المطلوبة إلى العميل ودمجها في DOM.

MongoDB 5-2

MongoDB هي قاعدة بيانات للمستندات ، مما يعني أنها تخزن البيانات في مستندات تشبه JSON. نعتقد أن هذه هي الطريقة الأكثر طبيعية للتفكير في البيانات، وهي أكثر تعبيراً وقوة من نموذج الصف / العمود التقليدي الشكل رقم (2-5-1) يوضح الاختلاف بين تسميات الـ MongoDB و RDBMS .



الشكل (2-5-1)

يقوم MongoDB بتخزين البيانات في مستندات مرنة تشبه JSON، مما يعني أن الحقول يمكن أن تختلف من مستند إلى مستند ويمكن تغيير بنية البيانات بمرور الوقت

يقوم نموذج المستند بتعيين الكائنات الموجودة في كود التطبيق الخاص بك، مما يسهل التعامل مع البيانات

توفر الاستعلامات المخصصة والفهرسة والتجميع في الوقت الفعلي طرقاً قوية للوصول إلى بياناتك وتحليلها

MongoDB هي قاعدة بيانات موزعة في جوهرها ، لذا فإن التوافر العالي والقياس الأفقي والتوزيع الجغرافي مدمجان وسهل الاستخدام

MongoDB مجاني للاستخدام. تم نشر الإصدارات التي تم إصدارها قبل 16 أكتوبر 2018 تحت AGPL. يتم نشر جميع الإصدارات التي تم إصدارها بعد 16 أكتوبر 2018.

بما في ذلك إصلاحات التصحيح للإصدارات السابقة، بموجب الترخيص العام من جانب الخادم (SSPL) الإصدار 1. الشكل رقم (2-5-2) يوضح قابلية الـ MongoDB



الشكل (2-5-2)

حيث تم اعتمادها ضمن المشروع بشكل رئيسي كقاعدة بيانات لحفظ المستخدمين والمنشورات وكل ما يتعلق بالبيانات الحية

RDMS 6-2

لتنق على "نظام إدارة قواعد البيانات العلائقية". RDBMS هو نظام DBMS مصمم خصيصًا لقواعد البيانات العلائقية. لذلك، RDBMSes هي مجموعة فرعية من نظم إدارة قواعد البيانات.

تشير قاعدة البيانات الارتباطية إلى قاعدة بيانات تخزن البيانات بتنسيق منظم، باستخدام الصفوف والأعمدة. هذا يجعل من السهل تحديد موقع قيم محددة والوصول إليها داخل قاعدة البيانات. إنها "علائقية" لأن القيم الموجودة في كل جدول مرتبطة ببعضها البعض. قد تكون الجداول مرتبطة أيضًا بجداول أخرى. تتيح البنية العلائقية تشغيل الاستعلامات عبر جداول متعددة في وقت واحد.

بينما تصف قاعدة البيانات العلائقية نوع قاعدة البيانات التي يديرها RDMBS، يشير RDBMS إلى برنامج قاعدة البيانات نفسه. هو البرنامج الذي ينفذ الاستعلامات على البيانات، بما في ذلك إضافة القيم وتحديثها والبحث عنها. قد يوفر نظام RDBMS أيضًا تمثيلًا مرئيًا للبيانات. على سبيل المثال، قد يعرض البيانات في جداول مثل جدول البيانات، مما يسمح لك بعرض القيم الفردية في الجدول وحتى تحريرها. تسمح لك بعض برامج RDMBS بإنشاء نماذج يمكنها تسهيل إدخال البيانات وتحريرها وحذفها.

تقع معظم تطبيقات DBMS المعروفة ضمن فئة RDBMS. تتضمن الأمثلة Oracle Database وMySQL وMicrosoft SQL Server وIBM DB2. تدعم بعض هذه البرامج قواعد البيانات غير العلائقية، لكنها تستخدم بشكل أساسي لإدارة قواعد البيانات العلائقية.

تتضمن أمثلة قواعد البيانات غير العلائقية Apache HBase وIBM Domino وOracle NoSQL Database. تتم إدارة هذا النوع من قواعد البيانات بواسطة برامج DMBS أخرى تدعم NoSQL، والتي لا تندرج ضمن فئة RDBMS.

حيث ضمن المشروع تم اعتماد ال sql ضمن طبقة ال Security وذلك للتعامل الوثيق مع AspCore مما يضمن Authentication , Authorization.

الفصل الثالث:

الأدوات المستخدمة

مقدمة

سنتناول في هذا الفصل الأدوات المستخدمة في كيفية برمجة وإتمام العمل ضمن بيئة العمل التي تم العمل عليها.

Visual Studio 2019 1-3

Microsoft Visual Studio هو بيئة تطوير متكاملة من Microsoft. يتم استخدامه لتطوير برامج الكمبيوتر، وكذلك مواقع الويب وتطبيقات الويب وخدمات الويب وتطبيقات الأجهزة المحمولة حيث تم الاعتماد عليه بشكل رئيسي لإنشاء مشروع الـ AspCore واحتواء الـ Blazor مع بناء الـ Class Libs ضمن أعضاء الفريق .. الشكل رقم (1-3) يعبر عن ايقونة الإصدار الأخير بتاريخ 2020.9.11

الإصدارات المستخدمة 16.6.x



الشكل (1-3)

Visual Studio Code 2-3

Microsoft Visual Studio Code هو محرر شفرة مصدر مجاني تم إنشاؤه بواسطة Microsoft لأنظمة Windows و Linux و macOS. تتضمن الميزات دعم تصحيح الأخطاء وإبراز بناء الجملة وإكمال التعليمات البرمجية الذكي والمقتطفات وإعادة بناء التعليمات البرمجية وGit المضمنة.

حيث تم استخدامه ضمن المشروع فقط لكتابة ال SASS/HTML/CSS

SQL Server management 3-3

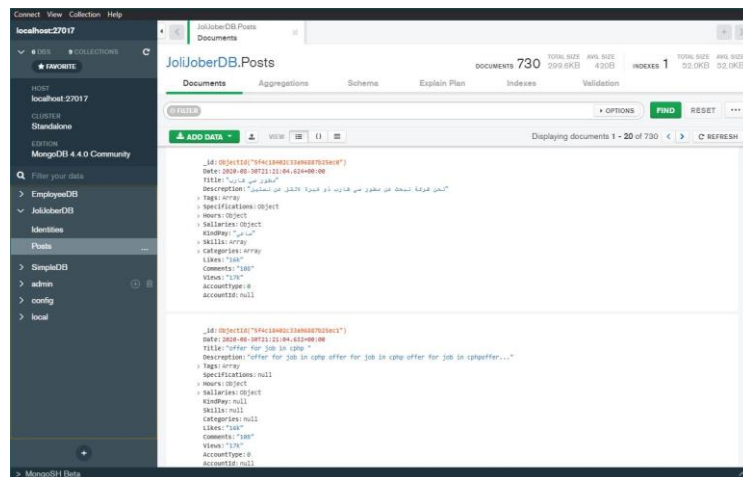
Microsoft Visual Studio SQL Server Management Studio هو تطبيق برمجي تم إطلاقه لأول مرة مع Microsoft SQL Server 2005 ويستخدم لتكوين وإدارة جميع المكونات داخل Microsoft SQL Server.

حيث تم استخدامه ضمن المشروع لإدارة الجداول المتخصصة بطبقه الامن والحماية وتنفيذ الاستعلامات.

تم استخدام ضمن المشروع إصدار 18.3.1

MongoDB Compass Community 4-3

مجرد واجهه تخاطبية مع ال LocalHost أو ال Server الذي يتم حفظ فيه قاعدة البيانات وإنشاء ال Collection و وملفات ال Json كقاعدة ال noSql ... يبين الشكل الواجهة التخاطبية رقم (3-4)



الشكل (3-4)

GitHub 5-3

العمل ضمن فريق لابد من مكان سحابي يتم فيه حفظ الكود والتعامل معه عن بعد وتنظيم الأنشطة والمهام والتشارك عن بعد

فهو منصة عالمية وأشهر المنصات الـ OpenSource والعمل ضمن نطاق الـ Private.



الفصل الرابع:

التقنيات والتضمينات الرئيسية

مقدمة

سنتناول في هذا الفصل التقنيات والتضمينات (المكاتب) الرئيسية المستخدمة في بناء وتطوير منصة الويب الخاصة، بالإضافة إلى توضيح وتبيان المواضيع التي تم استخدامها ضمن المشروع ولماذا استخدمهم.

4-1 معمارية MVC

واحدة من أشهر المماريات وأوسعها انتشاراً في عالم الويب حيث بدأ استخدامها في تطوير تطبيقات الدسكوب التقليدية ضمن لغة جافا وغيرها في سبعينيات وثمانينيات القرن الماضي، ولكن مع ظهور الويب وانتشاره، بدأت بالانتشار بشكل واسع وتم بناء العديد من اطر العمل للغات البرمجة بالاعتماد عليها مثل Django ضمن لغة Python و Sprint ضمن لغة Java، وأخيراً ASP.NET MVC ضمن لغة سي شارب

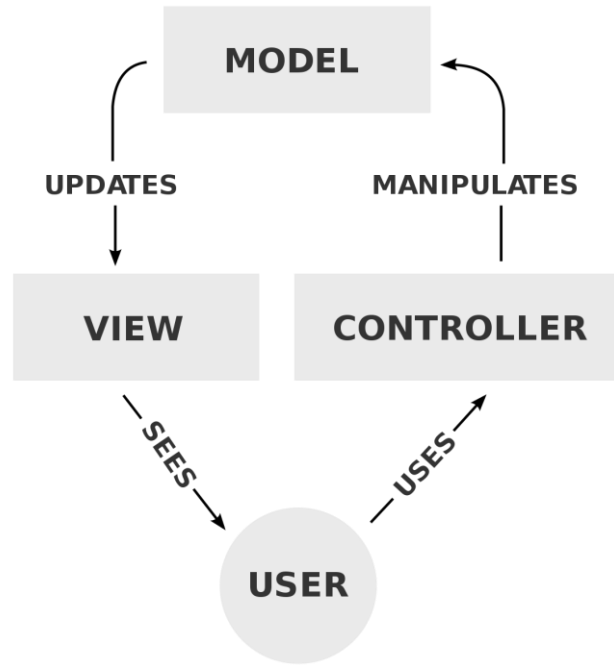
تتألف الممارية السابقة من الأقسام التالية:

المتحكم أو "Controller": يعد المسؤول عن استلام طلبات المستخدم ومعالجتها بالإضافة إلى القيام بالعمليات المنطقية والتحقق من هوية المستخدم وصحة المعلومات المرسلّة أو المدخلة من قبله.

الصفحة أو "View": تعد المسؤولة عن عرض البيانات للمستخدم والتفاعل معه بشكل مباشر، حيث تقوم بتسجيل الاحداث كالضغط على أحد الأزرار وإرسال هذه الاحداث مع البيانات المرسلّة إلى المتحكم ليتم معالجتها.

النموذج أو "Model": يقوم بشكل رئيسي بتأمين الاتصال مع مصدر البيانات، والذي غالباً ما يكون قاعدة بيانات علائقية "Relational database" وتنفيذ التعليمات والاستعلامات ضمنها، مع إعادة النتائج للمتحكم كي يكمل إجراء العمليات عليها.

كما يوضح الشكل رقم (4-1) البنيات الثلاث

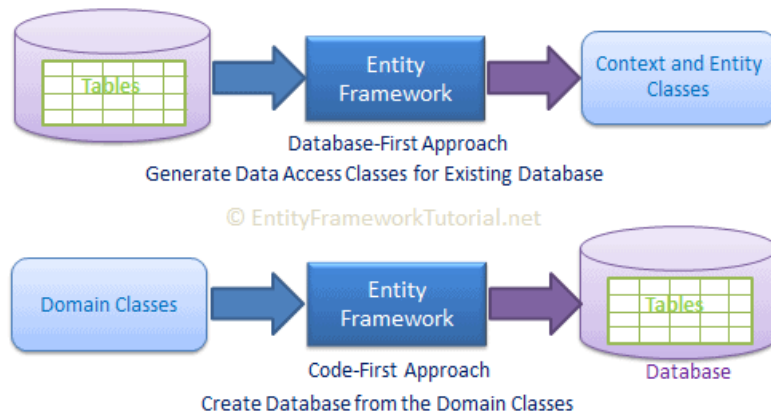


الشكل (1-4)

حيث تم عمل المشروع ضمن MVC Core LastVersion_3.

2-4 مكتبة Entity Framework Core

تم اعتماد مكتبة Entity framework core وذلك لربط قاعدة بيانات SQL Server مع تطبيق سي شارب، هذه المكتبة تعتمد مفهوم ORM أو Object relational mapper والذي يعتمد على تحويل جداول قاعدة البيانات إلى صفوف برمجية "Classes" وبالعكس، كما نتيح لنا هذه المكتبة مزايا أخرى مثل إنشاء قاعدة البيانات انطلاقاً من الصفوف البرمجية بالإضافة إلى مراقبة التعديلات على هيكلية قاعدة البيانات "Schema" والاستعلام من قاعدة البيانات باستخدام مكتبة LINQ. يوضح الشكل رقم (2-4) كيفية الفرق بين الـ CodeFirst و Database-First وكيفية تخاطب الـ EF



الشكل (2-4)

كما أتاحت بشكل خاص بناء طبقة ال Security والاستعلام الانبي من sql

تم استخدام الإصدار 3.1.5

3-4 مكتبة Language Integrated Query (LINQ)

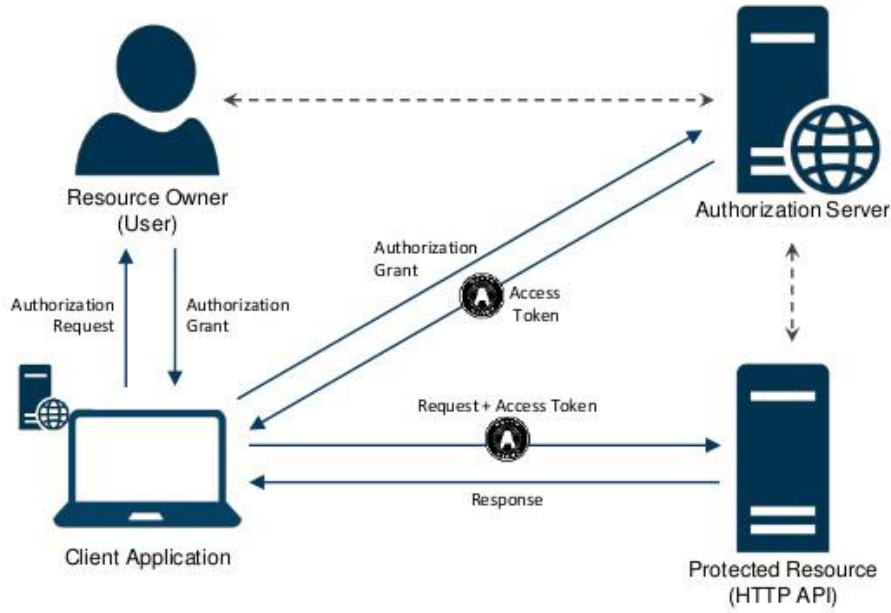
Language-Intergrated Query او LINQ وهي مكتبة تستخدم لتوحيد طريقة الاستعلام من مصادر مختلفة للبيانات مثل XML, JSON, Documents, Relational database باستخدام كود سي شارب مباشرة وذلك عن طريق كتابة تعابير Expressions يتم تحويلها للغة الاستعلام الاصلية وتنفيذها ومن ثم إعادة النتيجة ككائنات ضمن لغة سي شارب، قمنا باعتماد إحدى تضمينات هذه المكتبة والذي يدعى LINQ-To-SQL وذلك من أجل كتابة الاستعلامات بلغة سي شارب ومن ثم تقوم المكتبة بعملية تحسين "Optimize" لهذه الاستعلامات ومن ثم إرسالها لقاعدة البيانات ليتم تنفيذها وتعبئة الجدول الناتج ضمن كائنات مناسبة.

حيث تعتمد بشكل رئيسي على ال foreach yield من بنيات Enumerable و Queryable.

4-4 تضمين IdentityDbContext / ApiAuthorizationDbContext

هو API التي تدعم وظيفة تسجيل الدخول واجهة المستخدم UI وإدارة المستخدمين وكلمات المرور وبيانات الملف الشخصي والأدوار والمطالبات والرموز المميزة وتأكد البريد الإلكتروني والمزيد. يمكن للمستخدمين إنشاء حساب مع معلومات تسجيل الدخول المخزنة في الهوية أو يمكنهم استخدام موثر تسجيل دخول خارجي. وتشمل مقدمي خدمات تسجيل الدخول الخارجية المدعومة فيسبوك و Google وحساب Microsoft، Twitter. يتم تكوين الهوية عادة باستخدام قاعدة بيانات SQL Server لتخزين أسماء المستخدمين وكلمات المرور وبيانات ملف التعريف والذاتيات لحسابات المستخدمين. يوضح الشكل رقم (4-4) كيفية المصادقة وتوليد الشيفرة الخاصة لكل عميل

مع إمكانية ضبط الأمان في ال Client عن طريق IdentityServer4 .



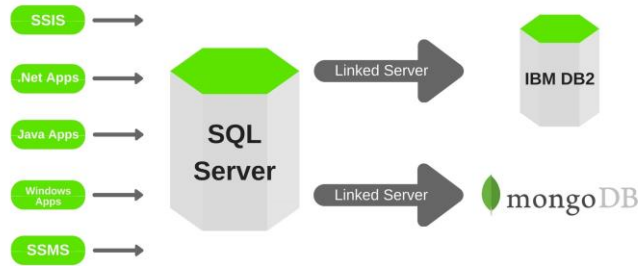
الشكل (4-4)

MongoDB.Driver 5-4

المكتبة المسؤولة عن الاتصال وتوفير التوابع لمخاطبة قاعدة البيانات مع دعم الـ Query وإنشاء Collections .

حيث تعد المنفذ بين الـ C# ولغة الاستعلام ضمن الـ MongoDB NoSql.

الشكل رقم (5-4) يوضح بشكل بسيط عمل وسهولة انتقاء بين كلا الطرفين



الشكل (5-4)

6-4 معمارية Dependency injection

إن AspCore دعم نمط تصميم برنامج حقن التبعية DI، وهو تقنية لتحقيق عكس التحكم ICO بين الفئات وتبعياتها.

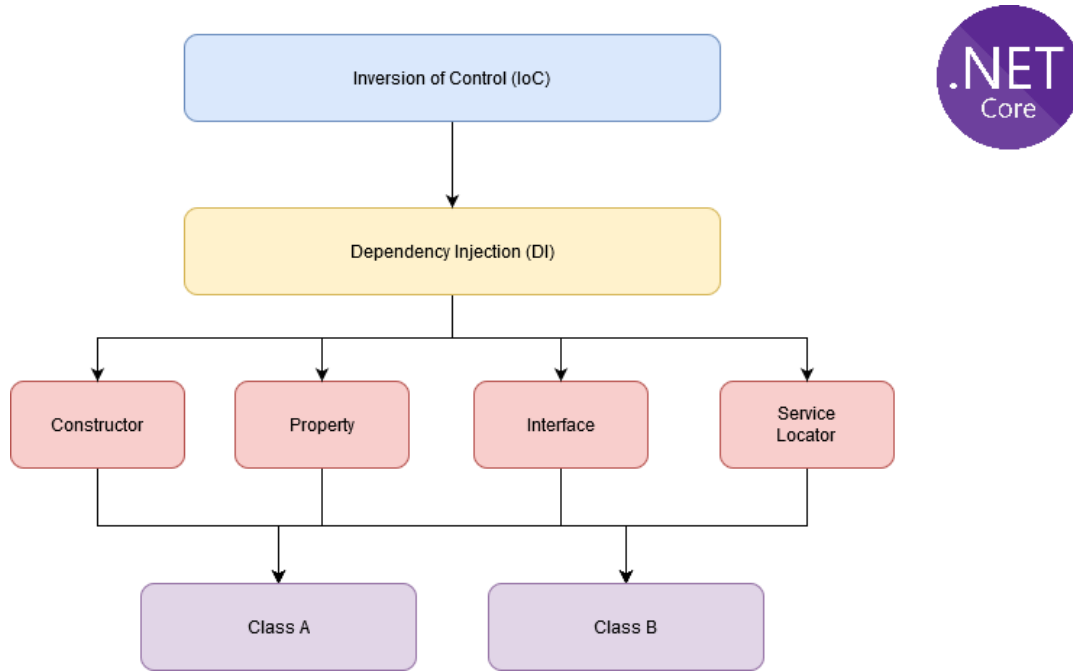
حيث تمكن من تقليل انشاء الكلاسات والوَبجكتات ضمن كل Request ويتم ضمن اعتماديات لحقن فترة حياة الـ اوبجكت وإعادة استخدامه وعدم ارهاق السيرفر على الأمد اطويل وتقسم لعدة أنواع منها:

Transient: وتعني فترة حياة عابرة تنتهي لكل طلب يتم مناداته.

Scoped: وتعني فترة حياة خاصة لبنية ضمن حياة تستمر حتى انتهاء التعريف مجال الاستدعاء لها.

Singleton: وتعني حياة طويلة الأمد تستمر من بدء السيرفر.

الشكل ادنى رقم (6-4) يوضح نمط الحقن بين الطبقات الأخرى



الشكل (6-4)

حيث تم الاستخدام بالمشروع كبنية اتصال أساسية لـ Controller وربط الدائم مع قواعد البيانات ضمن حياة وسهولة حقن بين الطبقات المنفصلة لتشكل رابطة مباشرة بين مكونات الموقع والـ AspCore

7-4 مفهوم Single Page Apps

هناك نهجين عامين لبناء تطبيقات الويب اليوم: تطبيقات الويب التقليدية التي تؤدي معظم منطق التطبيق على الخادم، وتطبيقات صفحة واحدة (SPAs) التي تؤدي معظم منطق واجهة المستخدم في متصفح الويب، والتواصل مع خادم الويب في المقام الأول باستخدام واجهات برمجة التطبيقات على شبكة الإنترنت. نهج هجين هو أيضا ممكن، وأبسط يكون المضيف واحد أو أكثر من التطبيقات الفرعية الغنية سبا مثل داخل تطبيق ويب التقليدية أكبر. الشكل رقم (7-4) يوضح الفرق بين وضع الصفحات التقليدي بين الصفحة الواحدة

Single Page Applications Work Differently



الشكل (7-4)

حيث تم اعتماد في المشروع الـ Blazor كإطار عمل يتيح الـ SPAs لسهولة تحويل المنصة لتطبيق مع إمكانية تخفيف إعادة تحميل المكاتب والسرعة ودعم الـ UI بشل أساسي مع الـ RealTime

Progressive web application 8-4

هو نوع من برامج التطبيقات التي يتم تسليمها عبر الويب، والتي تم بناؤها باستخدام تقنيات الويب الشائعة بما في ذلك HTML و CSS و JavaScript. ويهدف إلى العمل على أي منصة يستخدم متصفح متوافق مع المعايير. تتضمن الوظيفة العمل دون اتصال، ودفع الأخطار، والوصول إلى الأجهزة، تمكين إنشاء تجارب المستخدم مشابهة للتطبيقات الأصلية على سطح المكتب والأجهزة المحمولة. نظرًا لأن تطبيق الويب التقدمي هو نوع من صفحات الويب أو موقع الويب المعروف باسم تطبيق الويب، فلا يوجد أي شرط للمطورين أو المستخدمين لتثبيت تطبيقات الويب عبر أنظمة التوزيع الرقمية مثل Apple App Store أو Google Play.

في حين أن تطبيقات الويب كانت متاحة للأجهزة المحمولة منذ البداية، إلا أنها كانت بشكل عام أبطأ، وكان لها ميزات أقل، وكانت أقل استخدامًا من التطبيقات الأصلية. ولكن مع القدرة على العمل دون اتصال، التي كانت متاحة في السابق فقط للتطبيقات الأصلية، يمكن لـ PWAs الذي يعمل على الأجهزة المحمولة أن يؤدي بشكل أسرع ويوفر المزيد من الميزات، مما يسد الفجوة مع التطبيقات الأصلية، بالإضافة إلى كونه محمولًا عبر كل من منصات سطح المكتب والجوال.

PWAs لا تتطلب جميع منفصلة أو توزيع. نشر تطبيق ويب تقدمي كما سيكون لأي صفحة ويب أخرى. تعمل PWAs في أي متصفح، ولكن ميزات "app-like" مثل كونها مستقلة عن الاتصال، وتثبيت الشاشة الرئيسية، ودفع الرسائل تعتمد على دعم المتصفح. وتسلط العديد الضوء على التحسينات الهامة في مجموعة واسعة من مؤشرات الأداء الرئيسية بعد تنفيذ PWA ، مثل زيادة الوقت الذي يقضيه في الصفحة أو التحويلات أو الإيرادات

حيث تم استخدام ميزة الـ offline ضمن الـ Client-Side بالمشروع بحيث يتم التعامل مع طلبات مجمدة ريثما يتم إعادة الاتصال بالإنترنت وتوفير بيئة استخدام تضمن التغيرات من قبلة وعم ضياع التأثيرات على الموقع الحي.

4-9 مكتبة الزمن الحقيقي SignalR

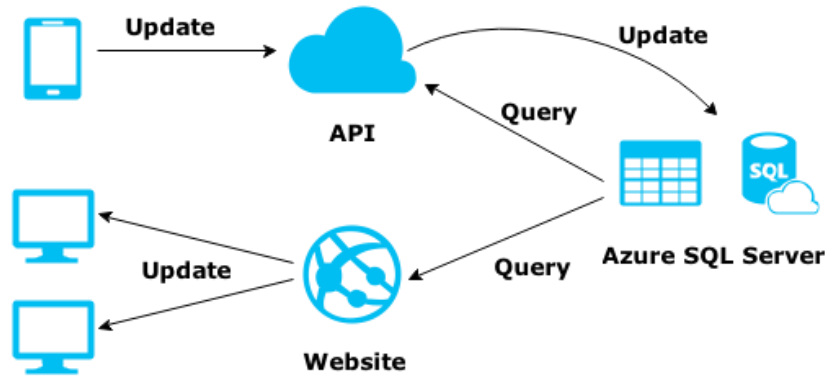
من المتوقع أن تقدم التطبيقات الحديثة اليوم معلومات حديثة دون الضغط على زر التحديث. أضف وظائف في الوقت الحقيقي إلى لوحات المعلومات والخرائط والألعاب وغيرها.

ما هي وظيفة الوقت الحقيقي؟ انها القدرة على أن يكون لديك التعليمات البرمجية من جانب الخادم دفع المحتوى إلى العملاء المتصلين كما يحدث ، في الوقت الحقيقي. تستخدم SignalR WebSockets عندما تكون متوفرة، و تقع بأمان مرة أخرى على تقنيات أخرى عندما لا يكون، بينما يبقى رمز التطبيق الخاص بك كما هو.

SignalR سريع وقابل للتطوير

مثل بقية ASP.NET Core ، تم بناء SignalR لأداء عالي وهو واحد من أسرع

في الوقت الحقيقي حولها. الشكل الاتي رقم (4-9) يوضح كيفية الاتصال والاشعار لحدوث تغير في قاعدة البيانات



الشكل (4-9)

حيث تم تطبيق ضمن المشروع استخدام الـ RealTime بشكل أساسي حيث يشمل 90% من الطلبات بين المستخدمين والتفاعل بشكل مباشر

شمل العمل على المنصة أكثر من موزع Hub

حيث يتوفر هنالك الـ Blazor Hub ضمن التوابع الستاتيكية التي تعمل ضمن الـ ServerSide بالإضافة لـ Hub يقوم بإدارة العمليات الخارجية ومزامنة لمنشورات وآخر التحديثات والتفاعل عليها من تعليقات ومراسلات وتهيئة بين مكونات المنصة بشكل كامل.

10-4 مكتبة Bootstrap

تم تصميم وبناء هذه المكتبة من قبل ثلاثة مهندسين ضمن شركة Twitter ثم ما لبثت أن أصبحت مستقلة عن الشركة السابقة للتحويل إلى مكتبة مفتوحة المصدر تهدف إلى توفير الوظائف والتنسيقات الأساسية لعناصر الويب كالأزرار والقوائم وغيرها.

لذلك هي مكتبة أساسية تم اعتمادها ضمن المشروع لبناء الواجهات وسهولة تضمين الأصناف الجاهزة من CSS وإعطاء مظهر UI غني ومرن وقابل لتحويل بشكل سلس بالإضافة الاتساق مع شاشات والأبعاد المختلفة Responsive. حيث الشكل رقم (4-10) يمثل الأيقونة المتعارفة



الشكل (10-4)

حيث استخدم ضمن المشروع اصدار 4.4 مع اربع تضمينات إضافية تشمل الايقونات و الانميشن والـ js لها

11-4 مكتبة JQuery

تعد واحدة من أشهر وأقدم مكاتب JavaScript حيث تستخدمها كبرى الشركات كـ Google و Microsoft و Facebook وذلك لتضمين وظائف ضمن صفحة الويب. ففي مشروعنا قمنا بالاعتماد على هذه المكتبة لإضافة العديد من الأحداث إلى عناصر الواجهة وللتواصل بين المخدم وجهاز المستخدم باستخدام تقنية Ajax.

حيث تم الاعتماد عليها ضمن المشروع 95% في كتابة الـ JavaScript

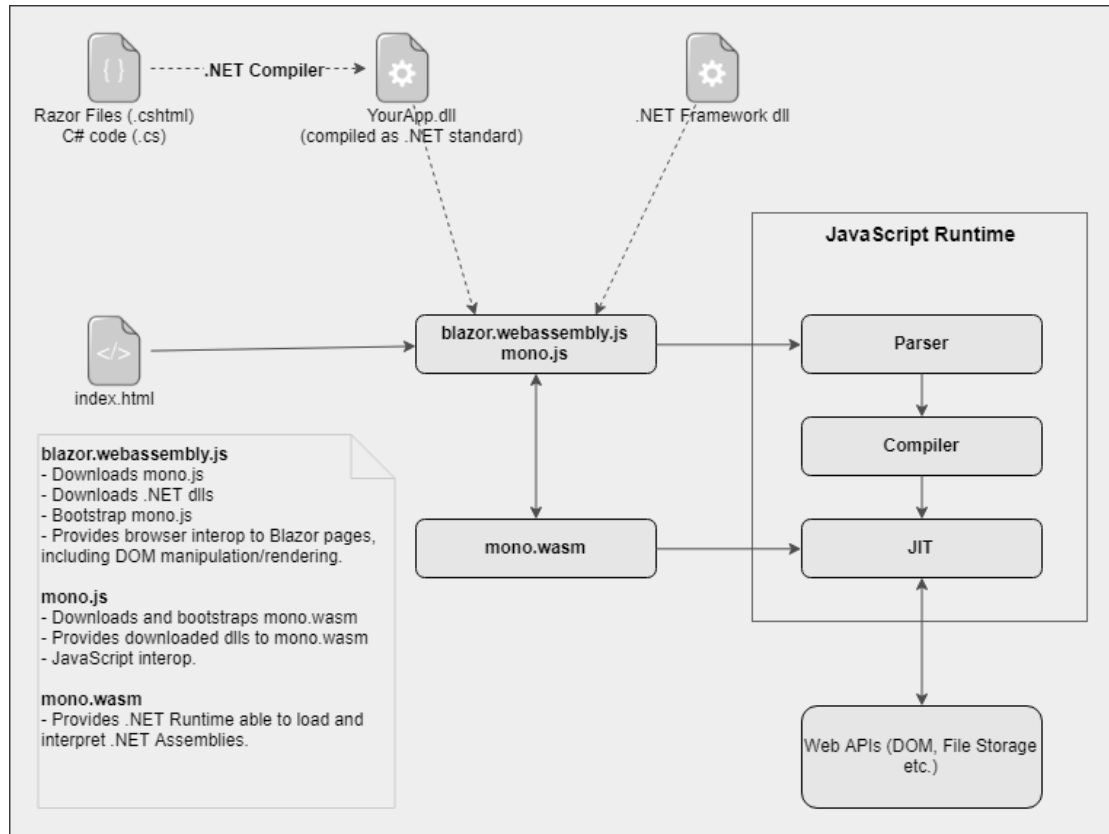
12-4 طلبات Ajax

بنية ضمنية لـ JQuery حيث تقوم بطلبات من Http تشمل كلا من الـ Get , Post , Delete.. ز وغيرهم والاستقبال. بشكل عام ضمن المشروع لم يتم استخدام اجاكس كقيام بطلبات واحضار البيانات وانما تتم عملها في حال مشاكل الاتصال SignalR فلا حاجة لها ضمن الزمن الحقيقي فهي تعد ابطئ نوعاً ما من الطلبات التي تعتمد على بنية الـ HttpClient أو الـ TCP وهي أقل أماناً.

13-4 ماهي الـ WebAssembly

تقنية ومفهوم جديد في عالم البرمجة سيتم اقتصار الكلام عنه ضمن المشروع على انه ضمن الـ Blazor وبالأخص الـ C#.

ضمن البيئة الموضحة بالشكل رقم (13-4) :



الشكل (13-4)

14-4 طريقة Hosting Models

تمكين طبقة الـ Api Controller للإتصال بين الـ Server-Side و الـ Client-Side .. مما يعطي سماحية عزل الـ Host وتنقل بين طرفي الـ Side ضمن الـ Components بشكل مريح.

WorkWise 15-4 ثيمة

مجرد Template Front-end تم سحب منه العناصر لسرعة إيضاح الفكرة والمشروع والتركيز على البنية الداخلية Back-end .

الفصل الخامس:

البنية والتصميم

مقدمة

سنتناول في هذا الفصل البنية المرتكز عليها المشروع من أنماط تصميم Design Pattern تم تطبيقها،

يعد هذا الفصل من أهم الفصول لأن أي مشروع يتطلب عمل ضخم وكتابة اكواد ضمن لغات وبيئات مختلفة لابد من تنظيمه وبناء قاعدة مرجعية تمكن المبرمج من التعديل واستكشاف الأخطاء والتطوير على المشروع

بالإضافة لقابلية إعادة الاستخدام والفصل بين الاعتماديات وضمان السرعة والأداء وهذا هو فحوة هندسة البرمجيات.

Model-View-Controller 1-5

هذا النسق يستخدم ضمن إطارات عمل برمجة مواقع الويبمثالية لتطوير تطبيقات معقدة ولكن خفيفة الوزن. يوفر إطار عمل قابل للتوسيع ويمكن توصيله، والذي يمكن استبداله وتخصيصه بسهولة.

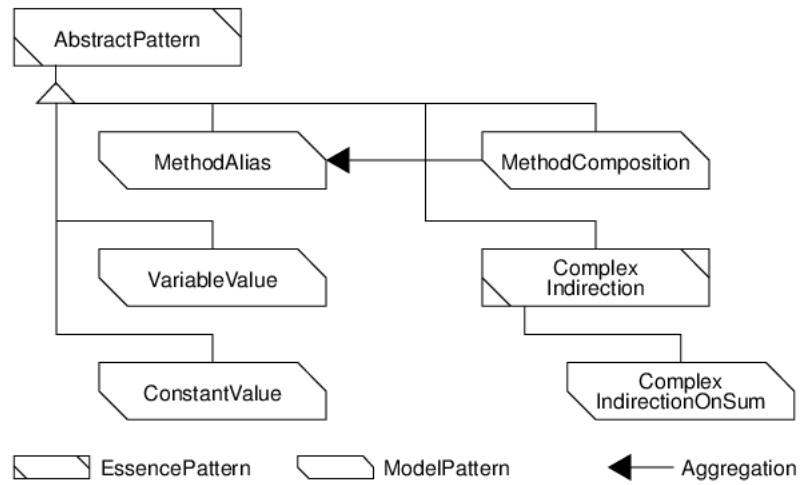
يستخدم التصميم المستند إلى المكون للتطبيق عن طريق تقسيمه منطقياً إلى مكونات Model و View و Controller. وهذا يمكن المطورين من إدارة تعقيد المشاريع الكبيرة الحجم والعمل على المكونات الفردية.

يعزز هيكل MVC التطوير الاختباري وقابلية الاختبار للتطبيق، حيث يمكن تصميم كافة المكونات المستندة إلى الواجهة واختبارها باستخدام كائنات وهمية. وبالتالي ، ASP.NET Core إطار MVC مثالية للمشاريع مع فريق كبير من مطوري الويب. يدعم كافة وظائف واسعة موجودة، مثل التحويل والمصادقة، الصفحات الرئيسية، ربط البيانات، عناصر تحكم المستخدم، العضوية، التوجيه، الخ. يساعد في بناء التطبيقات، والتي هي خفيفة الوزن ويعطي السيطرة الكاملة للمطورين.

Hierarchy Pattern Composite DLL 2-5

يعد الهرمية التي تشمل باقي أنماط التصميم

الشكل موضح رقم (2-5)



الشكل (2-5)

Dependencies Layer 3-5

هذا النمط هو جوهر الهندسة المعمارية

بحيث تكون الاعتماديات منفصلة ومستقلة وبذات الوقت مترابطة ومرنة للعمل ضمنها

وتم تقسيمها ضمن المشروع إلى كتل تتميز:

وحدة الـ UI

وحدة الـ Bounded Context

وحدة الـ Infrastructure

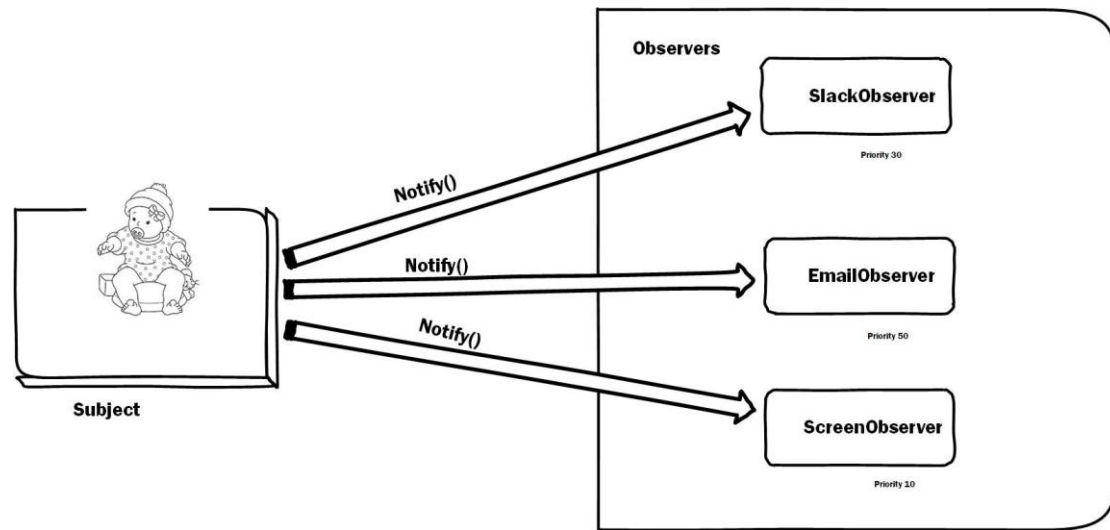
وحدة الـ Shared Kernel

وحدة الـ Unit Test

Observer Pattern 4-5

من النمط المستخدم ضمن الـ Blazor حيث يعمل بشكل وثيق مع الـ MVVM مهمة الرئيسية عدم تراكم البيانات والتعطيل الانمي والاشعار Notify بشكل لا يؤثر على توليد البيانات السابقة من إضافة وحذف وتعديل والعمل ضمن الموقتات الزمنية وتراكم المنشورات ضمن الـ Components

نمط تصميم المراقب مناسب للإشعارات الموزعة المستندة إلى التعديل والحفظ، لأنه يدعم فصل نظيف بين مكونين مختلفين أو طبقات تطبيق مختلفة، مثل طبقة مصدر البيانات (منطق العمل) وطبقة واجهة المستخدم (عرض). يمكن تطبيق النمط كلما استخدم موفر عمليات الاسترجاع لتوفير العملاء مع المعلومات الحالية. الشكل رقم (4-5) يبين عملية الـ Notify



الشكل (4-5)

حيث يستخدم ضمن الـ Net Core. يعتمد الـ system.IObservable والكثير من الـ Data Structure التي توفر هذا النمط.

Factory Pattern 5-5

من الأنماط التي توفره السي شارب ذاتيا فهو

طريقة المصنع هو نمط تصميم الإنشاء، أي، المتعلقة خلق الكائن. في نمط المصنع، نخلق كائن دون تعريف منطق إنشاء العميل واستخدام العميل نفس الواجهة الشائعة لإنشاء نوع جديد من الكائن.

والفكرة هي استخدام وظيفة عضو ثابت (أسلوب مصنع ثابت) الذي يخلق إرجاع الحالات، وإخفاء تفاصيل وحدات فئة من المستخدم.

نمط المصنع هو واحد من مبادئ التصميم الأساسية لإنشاء كائن ، مما يسمح للعملاء بإنشاء كائنات من المكتبة (موضح أدناه) بطريقة لا تحتوي على اقتران ضيق مع التسلسل الهرمي للفئات من المكتبة.

6-5 ModelViewViewModel (MVVM)

عندما نتحدث عن عناصر UI فمن الضروري التطرق إلى كيفية نقل ووصف وتحويل الشيء المصمم إلى لغة أو فلسفة تعرف بالتمثيل المكونات الظاهرية إلى أغراض وصفوف تطابقها بنويا حيث استخدمت بشكل واسع بالمشروع ضمن الـ Blazor في اعتمادية على الـ Components وتقسم إلى:

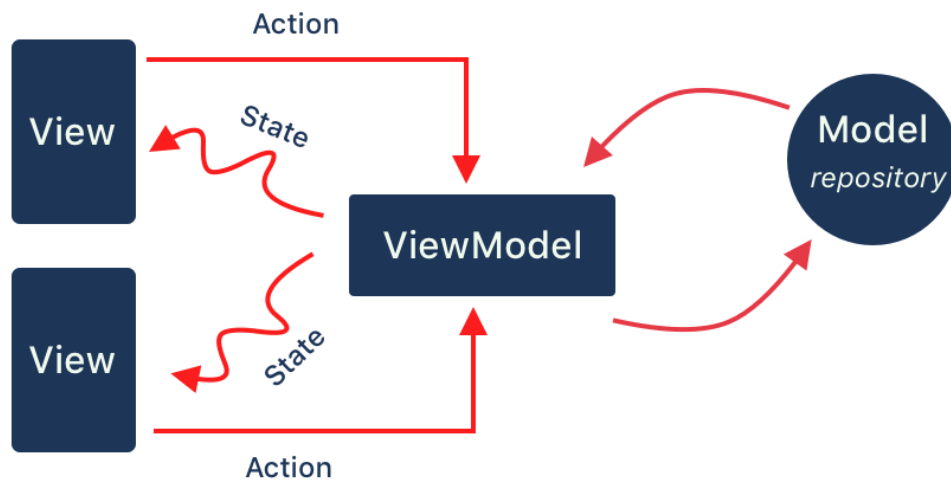
Model : تعد الطبقة الحساسة لحتوائها على المعلومات والبيانات التي يتم عجب ارسالها او نوعها إلى الطرف الاخر.

View : وهي الطبقة المرئية للمستخدم .. تجهل تماما أي شيء عن ال Model وتمنع المستخدم معرفة البنية الأساسية وتجريده بالخواص اللازمة فقط.

ViewModel : وهو طبقة الربط التي تتيح الفصل بين ال Model و ال View

عن طريق تمثيل وصفي لعناصر ال View بما يماثلة من نوع بيانات مقابلة

الشكل رقم (5-6) يبين الطبقات وكيفية التعاون فيما بينها



الشكل (5-6)

Repository Pattern 7-5

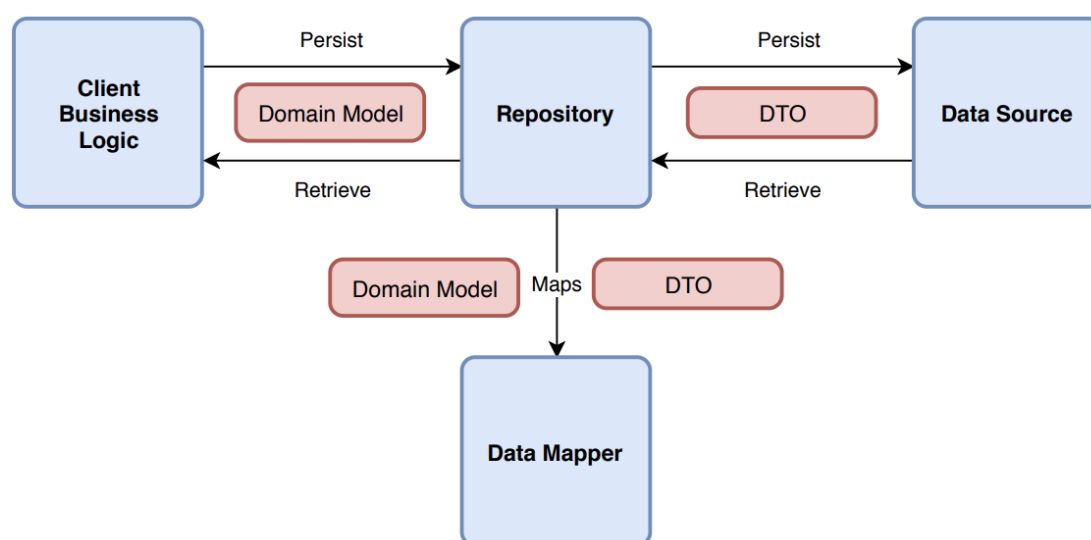
يسمى نمط المستودع

حيث هو يتم نقل الانشاء ضمن ال DI الحقن بعد ان تم العزل باستخدام **Dependencies** و **Hierarchy** .

ضمن المشروع تم العمل على معالجة الارتباطات وحل العقد وفصل قاعدة البيانات وتوفير عملية اصلاح المشاكل .. عن طريق المستودعات والاعتماديات

حيث: توفر مكونات ثبات البيانات إمكانية الوصول إلى البيانات المستضافة ضمن حدود الخدمات الجزئية (أي قاعدة بيانات الخدمات الجزئية). وهي تحتوي على التنفيذ الفعلي للمكونات مثل المستودعات وفئات وحدة العمل، مثل كائنات DbContext الخاصة بإطار عمل الكيانات (EF). EF DbContext ينفذ كلا من المستودع ووحدة أنماط العمل

المستودعات هي فئات أو مكونات تغلف المنطق المطلوب للوصول إلى مصادر البيانات. فهي تقوم بتمركز وظائف الوصول إلى البيانات الشائعة، مما يوفر إمكانية صيانة أفضل وفصل البنية التحتية أو التكنولوجيا المستخدمة للوصول إلى قواعد البيانات من طبقة طراز المجال. إذا كنت تستخدم مخطط كائن العلائقية (ORM) مثل إطار عمل الوحدة، التعليمات البرمجية التي يجب تطبيقها يتم تبسيطها، وذلك بفضل LINQ والكتابة القوية. وهذا يتيح لك التركيز على منطق استمرار البيانات بدلاً من الوصول إلى البيانات الحساسة



الشكل (7-5)

يقوم المستودع بتنفيذ مهام الوسيط بين طبقات طراز المجال وتعيين البيانات، ويعمل بطريقة مشابهة لمجموعة من كائنات المجال في الذاكرة. كائنات العمل بشكل تعريفي بناء الاستعلامات وإرسالها إلى المستودعات للحصول على إجابات الشكل أعلى رقم (7-5) يمثل البنية المتكاملة للعبور بنمط الـ Repo.

من الناحية النظرية، يقوم المستودع بتغليف مجموعة من الكائنات المخزنة في قاعدة البيانات والعمليات التي يمكن تنفيذها عليها، مما يوفر طريقة أقرب إلى طبقة الاستمرار. المستودعات،

أيضاً، تدعم الغرض من فصل، بوضوح وفي اتجاه واحد، التبعية بين مجال العمل وتخصيص البيانات أو تعيين.

الفصل السادس:
الملفات والملحقات

مقدمة

سنتناول في هذا الفصل الملفات والملحقات الرئيسية من صيغ وامتدادات.

6-1 صيغة Mdf

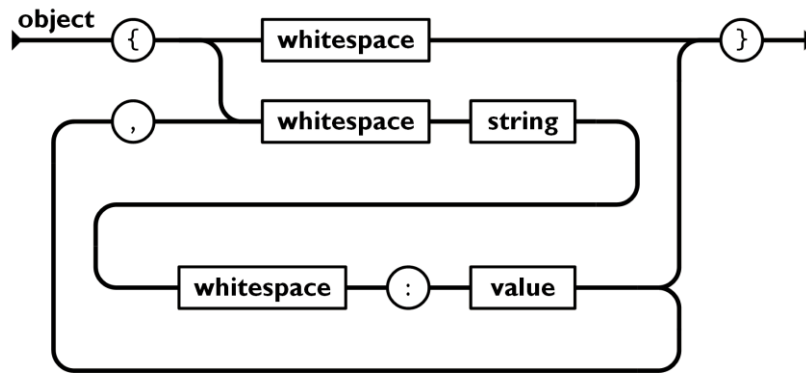
ملف يتم بتخزين البيانات في التنسيق الخاص به استناداً إلى مشغل قاعدة بيانات "RDBMS". كما يمكن أن تقوم هذه باستيراد البيانات المخزنة في التطبيقات وقواعد البيانات الأخرى أو ربطها مباشرة.

6-2 Xml

لغة التوصيف القابلة للتوسيع وهي لغة ترميز تحدد مجموعة من القواعد لترميز المستندات بتنسيق يمكن قراءته بواسطة الإنسان وقراءته آلياً.

6-3 JSON

تبادل البيانات عند تبادل البيانات بين المتصفح والخادم، يمكن أن تكون البيانات نصية فقط. JSON لإرسال JSON إلى JavaScript عبارة عن نص ، ويمكننا تحويل أي كائن JSON JavaScript. لتم استلامه من الخادم إلى كائنات JSON إلى الخادم. يمكننا أيضاً تحويل أي ، بدون تحليل وترجمات معقدة JavaScript بهذه الطريقة يمكننا العمل مع البيانات ككائنات ، فيمكنك تحويل الكائن JavaScript إرسال البيانات إذا كانت لديك بيانات مخزنة في كائن وإرساله إلى خادم كما موضح بالشكل رقم (6-3) JSON إلى نوع مفهوم



الشكل (3-6)

حيث تم استخدامها بشكل رئيسي للاحتفاظ بالبيانات ضمن ال MongoDB

الباب الثاني:
عرض المشروع

الفصل الأول:

المخططات التحليلية وقواعد العمل المتبعة

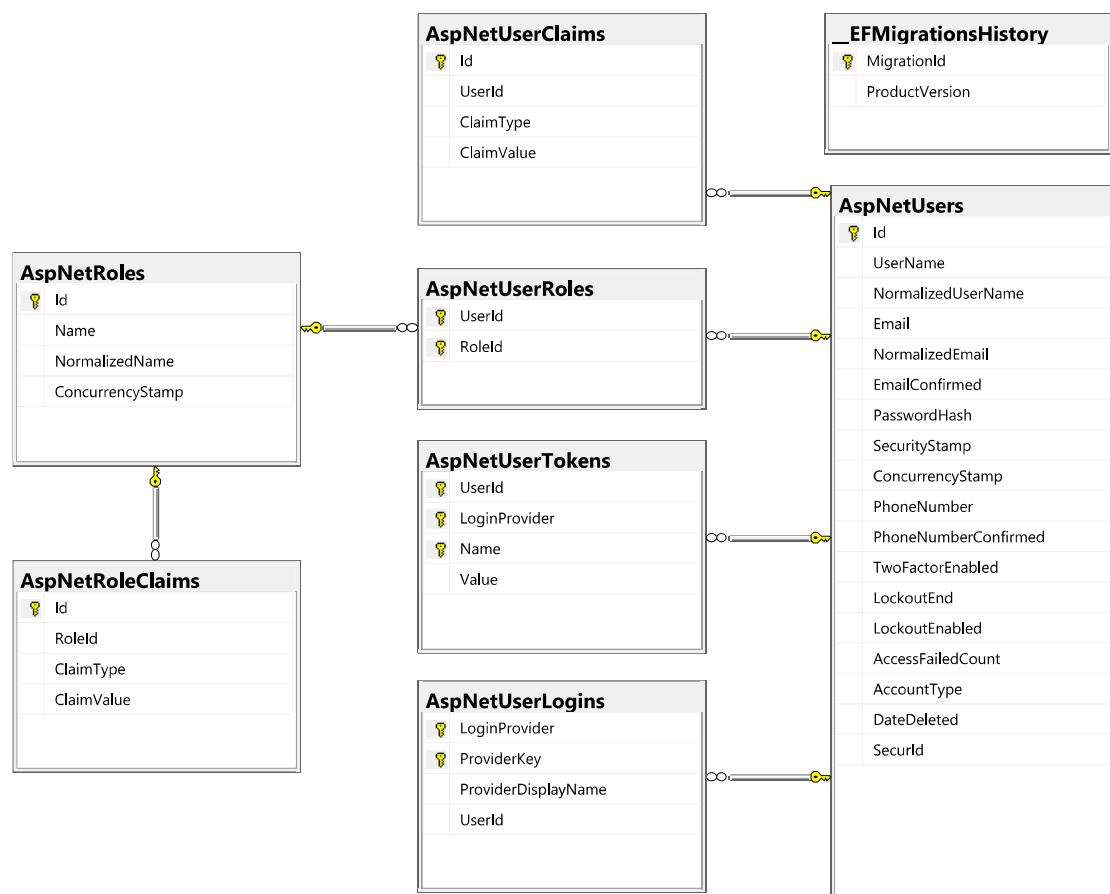
مقدمة

عند البدء بتطوير أية مشروع برمجي، نحتاج في مراحله الأولية إلى إنشاء مخططات برمجية لتحديد كيفية تفاعل المستخدم مع المشروع بالإضافة إلى الوظائف والخدمات والمزايا التي يقوم بتقديمها، لذلك سنقوم باستعراض بعض المخططات الرئيسية ضمن المشروع والتي تساعد في فهم الية عمله بشكل عام قبل البدء بشرح تفاصيله كاملة.

1-1 مخطط ERD لقاعدة البيانات (Security)

يضم الجداول الرئيسية ضمن طبقة الـ SQL وهي المسؤولة ضمن الموقع على الحسابات والذاتيات الرئيسية للمستخدمين، بالإضافة لتخزين المعلومات المتعلقة بعدد الزيارات ومحاولات الدخول وما إلى هنالك من ملفات الارتباط الخاصة بالمتصفحات وحفظ كل ما هو متعلق بالأمان وصلاحيات المستخدمين.

ويشمل الجداول الآتية شكل رقم (1-1-2):



الشكل (1-1-2)


جدول بيانات المستخدمين :AspNetUsers

يعبر البيانات الحساسة للحسابات ضمن المنصة فهو المسؤول عن تسجيل الدخول من خلاله والتأكد من صلاحية الحساب.

خصائص هذا الجدول:

- **UserName**: اسم المستخدم ضمن المنصة لتسجيل الدخول.
- **Email**: حساب المستخدم (عادةً يستخدم بديل عن اسم المستخدم للولوج للمنصة).
- **PasswordHash**: كلمة المرور يتم تشفيرها والاحتفاظ بها بـ (Hash).
- **PhoneNumber**: رقم المستخدم لضمان درجة الأمان.
- **AccountType**: يحدد نوع الحساب ان كان ينتمي لشركة أو مستخدم عادي.
- **SecurId**: يعد الحقل الموافق لحقل الـ Identity مع المرجع لـ MongoDB .
- **DateDeleted**: تاريخ حذف الحساب بشكل كامل (soft delete) .
- خصائص أخرى متعلقة بالمزامنة والتأكيد.

AspNetUsers

	Id
	UserName
	NormalizedUserName
	Email
	NormalizedEmail
	EmailConfirmed
	PasswordHash
	SecurityStamp
	ConcurrencyStamp
	PhoneNumber
	PhoneNumberConfirmed
	TwoFactorEnabled
	LockoutEnd
	LockoutEnabled
	AccessFailedCount
	AccountType
	DateDeleted
	SecurId

جدول صلاحيات الولوج AspNetRoles:

يتم حفظ الصلاحيات المتاحة لكلا الشركات والمستخدمين بالإضافة الى الإدارة الداخلية للموقع وقفل الـ Http منعا من الوصول الغير مسموح

وهذا الجدول يحقق عزل روابط Route الخاصة بين الأنواع المستخدمين للمنصة. وعند تأكيد الحساب وتسجيل الدخول يتم الاحتفاظ ضمن الـ Cookies للمتصفح .

خصائص هذا الجدول:

■ **Name**: يعبر عن اسم الصلاحية أو مجموع الصلاحيات عبر جدول الكسر للمستخدم.

AspNetRoles	
🔑	Id
	Name
	NormalizedName
	ConcurrencyStamp

جداول الاحتفاظ بالمصافحة Login، Claim،Token :

حيث تشمل كل من الشيفرات المولدة والاحتفاظ بها ضمن المتصفح لضمان امان عالي وتخزين بعض المعلومات والتحقق الدائم عندك كل طلب وتوليد بنية وشيفرة جديدة في كل طلب تضمن عدم الاختراق للحساب.

AspNetUserClaims	
🔑	Id
	UserId
	ClaimType
	ClaimValue

AspNetUserLogins	
🔑	LoginProvider
🔑	ProviderKey
	ProviderDisplayName
	UserId

AspNetUserTokens	
🔑	UserId
🔑	LoginProvider
🔑	Name
	Value

AspNetRoleClaims	
🔑	Id
	RoleId
	ClaimType
	ClaimValue

من أهم الخصائص تلك الجداول:

■ **ClaimType** : نوع القيمة الداخلة للمتطلبات لمعرفة السيرفر القيم المشفرة من اسم مستخدم وال Id الخاصة للمستخدم الحالي مع الاحتفاظ بكافة القيم الخاصة بال Cookies للمستخدم

■ **Value-Token**: نص مشفر بطول معين يتم توليده بشكل متكرر وفريد للمستخدم.. يستخدم للمصادقة وفتح الوصول إلى باقي الجداول.

حيث يعبر في كافة جداول ال ERD Security توليد ID من نوع GUID مما لها عدد غير محدود وكبير بالنسبة لـ Int .

1-2 مخطط Class Model-ER لقاعدة البيانات الرئيسية:

تشمل كافة الجداول التي يتم التخزين البيانات ضمن MongoDB

ويتألف من ملفات Json و Collection ضمن مفهوم No Sql.

حيث تعدل قاعدة بيانات ديناميكية الشكل Horizontal لا يهم عدد الاعمدة حيث يتم تخزين بشكل مباشر والتعريف عن الحقول وإمكانية انشاء ملفات ضمن الـ Run Time للتخزين مع إعطاء مرجع لكل حقل فهي تنمو بما يتناسب مع الموقع.

حيث يتم اتباع طريقة ان معظم البيانات توضع ضمن مجموعة واحدة وجدول واحد وكل حقل يمكن ان يعبر عن بيئة ثانية داخلية.

Identity
_Id
FirstName
SureName
Country
SecurId
Type
HeadLine
Overview
CoverImagePath
ProfileImagePath
Follwing
Followers
Skills
Specification
Field
Area
Qualification
other...

Post
_Id
Date
Title
Description
Tags
Specification
Hours
Sallaries
KindPay
Skills
Categories
Likes
Comments
Views
AccountType
AccountId
PostType
ReactApplyPost
Note
Age
JobNum
CompanyArea
other...

```
> { "_id": ObjectId("5f4c18402c33a96087b25ec0"),
  "Date": "2020-08-30T21:21:04.624+00:00",
  "Title": "مطور سي هارب",
  "Description": "نحن شركة نبحث عن مطور سي هارب ذو خبرة لاتقل عن تسعين",
  "Tags": Array(
    0: "C#"
    1: "wpf"
  ),
  "Specifications": Object(
    one: "programer"
    two: "div"
  ),
  "Hours": Object(
    Min: 4
    Max: 6
  ),
  "Sallaries": Object(
    Min: 600
    Max: 800
  ),
  "KindPay": "ساعي",
  "Skills": Array(
    0: "يتقن سي هارب اخر اصدار"
    1: "لديه خبرة سابقة فتعامل مع تطبيقات سطح المكتب"
  ),
  "Categories": Array(
    0: "حواسيب"
  ),
  "Likes": "16k",
  "Comments": "100",
  "Views": "17k",
  "AccountType": 0,
  "AccountId": null
}
```

حيث يتم التعامل مع ملفات JSON في التخزين والادارة من قبل الـ MongoDB

1-3 قواعد العمل:

يقوم المستخدمون بتسجيل حسابات ضمن المنصة بالإضافة للشركات للتعاقد مع المنصة. يتيح الحساب بروفایل شخصي كامل من متابعة أشخاص وعرض المهارات والاختصاصات وأرشفة المفضلات وإمكانية البحث والتواصل الدائم من اشعارات واخر الاعمال. حيث يمكن من نشر عمل لمشروع شخص ما أو التقدم لفرص العمل عبر المنشورات التوظيف والتعليق على الوظائف ومراجعة والتواصل مع الشركات

وبناء شبكة اجتماعية من الخبرات وريادة الاعمال

وإتاحة حفظ المنشورات ومشاركتها مع كافة الأصدقاء

ويأتي دور المنصة من تحليل وتجميع تلك البيانات للحصول على نتيجة أفضل لفرص التوظيف بما يتناسب مع المستخدم والبيئة التفاعلية من حوله.

أما الشركات تقوم بطرح فرص العمل ضمن تصنيفات وشروط مطلوبة من مهارات والرواتب المقدمة وشرح عن تلك الفرصة وعدد الموظفين الراغبة الشركة لتوظيفهم والمنطقة المحيطة بهم حيث يشمل الافرع للشركات وادارتهم والتواصل مع المقدمين على تلك الفرص.

كما يمكن للشركات نشر المقالات والمنشورات الريادية التي تعكس توجهات الشركة ومشاهدتها من قبل المتابعين.

■ حيث يمكن ان يعبر علائقياً:

■ يمكن للشركة الواحدة أن تملك مجال واحد فقط

■ الشركة الواحدة تملك عجة اختصاصات

■ الشركة تحوي عدة أفرع بمناطق مختلفة

■ الشركة الواحدة تنشر فرصة عمل بفرع أو أكثر باختصاص محدد

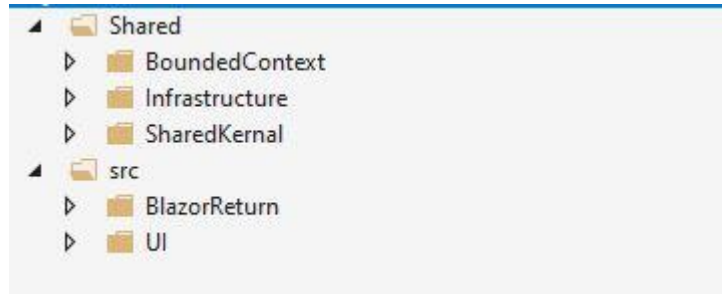
- كل بوست أكثر من فئة مستهدفة
- المستخدم له أكثر من مؤهل
- المستخدم يتواجد بمنطقة واحدة ويمكن للمنطقة أن يتواجد عدة مستخدمين
- يمكن أن يتفاعل على بوست أكثر من مستخدم
- يمكن للمستخدم أن يقوم عدة طلبات للعمل لبوست أو أكثر لشركة معينة أو كثر بمنطقة أو أكثر
- يمكن للمستخدم أن يقدم طلب توظيف باختصاص محدد
- يمكن للشركات أن تتفاعل مع بوستات المستخدمين وقد تقبله أو لا

الفصل الثاني:

البنية الداخلة وخدمات المنصة

1-2 الهرمية الداخلية للكود:

تم توصيف الهرمية والعمل عليها بما يتناسب ما تم طرحه ضمن الباب الأول حيث تم تقسم إلى اعتماديات وطبقات بالشكل:



حيث تصنف إلى طبقتين أساسيتين هما الـ

Front-end التي تشكل الـ UI والـ Blazor

Back-end التي تشكل البنية الـ MVC الداخلية مع طبقة الـ Shared

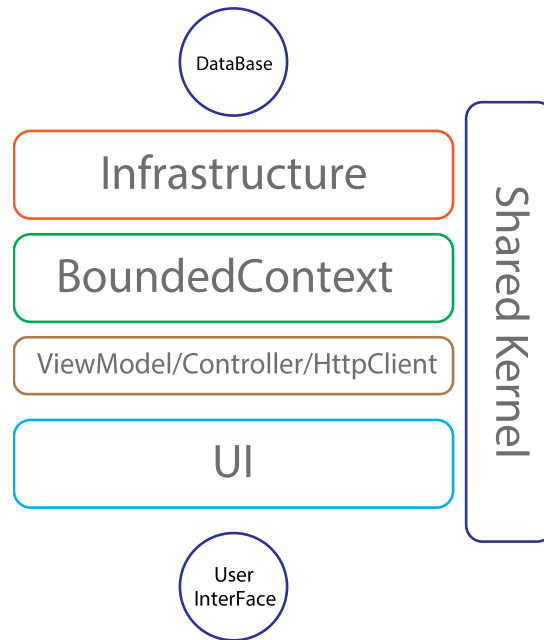
التي تتألق من:

BoundedContext : منطقة الربط الأساسية مع الـ MVC و الجلب البيانات والاستعلامات وكل ما يتعلق بعمل الريبوزتري , حيث يتم عبور البيانات عبر مايسمى بأوبجكتات الانتقال DTO.

Infrastructure : منطقة الاتصال مع قواعد البيانات

SharedKernel : تعد المنطقة المشتركة بين كلا الجوانب من حيث الاعتمادية بين الـ UI و منطقة العمل Repositories.

كما موضحة بالشكل رقم (1-2-2) :

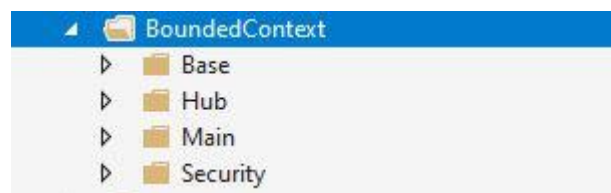


الشكل (1-2-2)

2-2 الاعتماديات والتقسيمات البنيوية الداخلية:

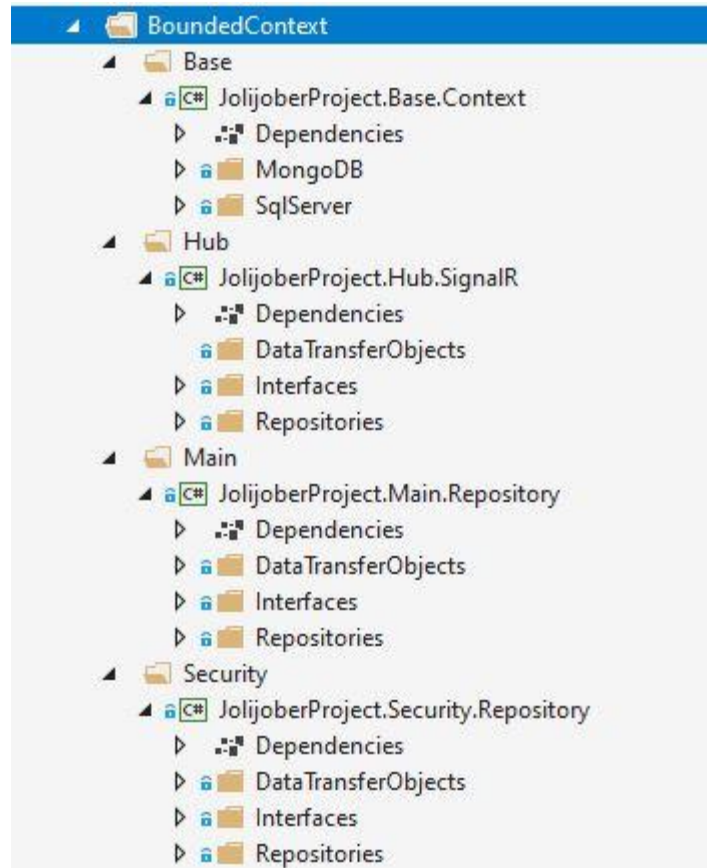
ضمن أي طبقة تتكون من عدة طبقات داخلية توصف انتقال البيانات والعزل الداخلي.

1 - BoundedContext :



تمثل ال Schema لل DB من بنيات رئيسية وبنيات الأمان بالإضافة للاتصال ال RealTime عن طريق موزعات Hubs ... هذه الطبقة مهمتها الرئيسية تأمين الاستخدام للاتصال مع ال DB ثم عمل استعلامات وفلترة بما يناسب المحتوى UI

حيث كل مجال يتألف من نمط متماثل لل DI كما تم شرحه في الباب الأول

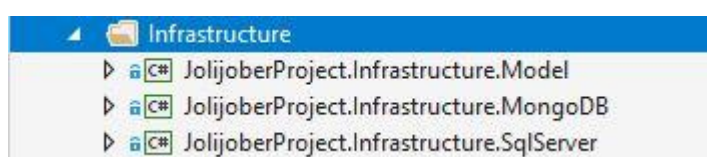


حيث تم اعتماد ال Repository بالإضافة مع هرمية الأصناف DLL بشكل معزول حيث يتم التبادل عن طريق DTO (Data Transfer Object) وحقن ال Interface ضمن ال Repository ومنها يوصف عمليات ال CURD على قواعد البيانات و جلب البيانات وتحويله بشكل المراد .

2- Infrastructure :

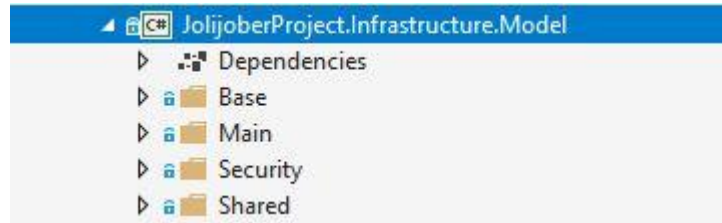
تمثل بنية التخاطب والاتصال مع قاعدة البيانات حيث يتم فيها انشاء الجداول عن طريق CodeFisrt وتمثيلها بما يعرف بال Model

وعزل المجالات كالأمان والجداول الغير مرتبطة و الأساسية



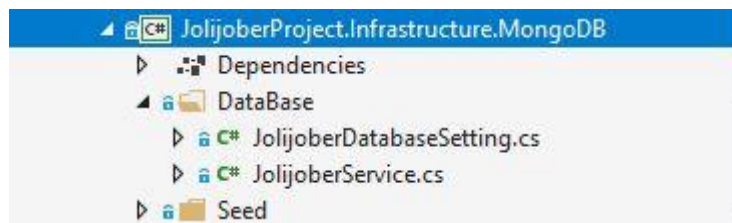
Model : الأصناف المعبرة عن الجداول ضمن مجالات موصفة للداتا

كما هو موضح بالشكل:



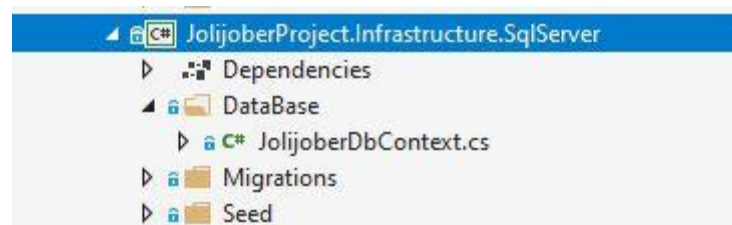
MongoDB : تمثل الاتصال مع المخدم عن طريق ConnectionString وموصف لقاعدة البيانات

كما موضح بالشكل:



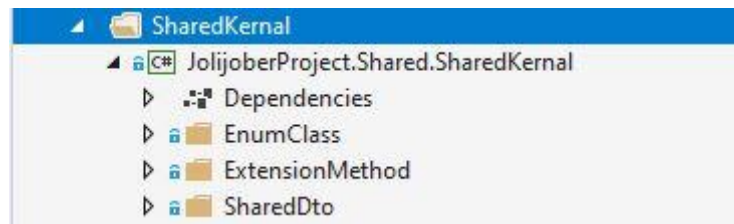
Sql Server : تمثل الطبقة الوسيطة بين EF Core و ال DB حيث يتم الاتصال مع قاعدة البياناتات ConnectionString وتحويل بما يناسب ال Model وحفظ التغييرات من انشاء CodeFirst بما يعرف بالهجرة Migration

كما موضح بالشكل:



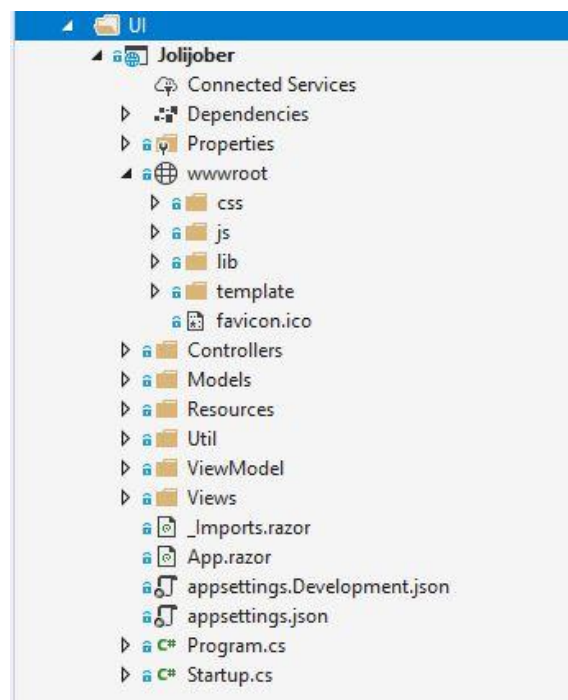
Shared Kernel – 3

هي طبقة وسيطة مشتركة تحوي ال Enum وتوابع الأكثر استخداما ومفيدة للتغليف كال Extension وغيرها



3-2 الطبقة الخارجية (User Interface) UI:

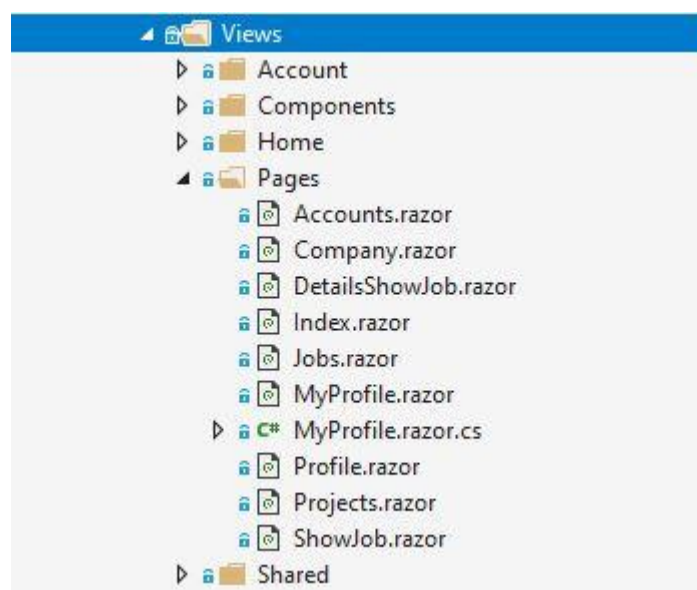
تمثل كلا من ال Host والنقل البيانات ونقطة الالتقاء MVC وعرض الواجهات وتخدیم ال Middleware وضبط الاعدادات.



حيث تشمل المكاتب التصميم من JS , CSS وحفظ الملفات والصور بما يعرف بـ محدد الوصول كال FTTP الذي يسمى wwwroot فهو المصدر الذي يستر الموقع منه ويقوم المتصفح بتحميل وإمكانية الوصول.

View –1

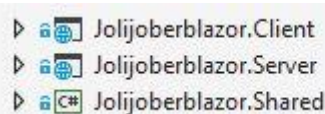
وهي اللتي تمثل بالمصفحات وملفات ال razor ال chtml الحاوية على ال Html وتعد الطبقة الأخير التي يتم بها مخاطبة أفعال المستخدم



4-2 بنية العميل Client:

تمثل التقنيات الفصل بين السيرفر والإطار العمل Blazor

كما موضح بالشكل الاتي:



5-2 التصميم والمرئيات (الصفحات):

1- صفحة تسجيل الدخول وانشاء حساب

تسجيل الدخول

انشاء حساب

تسجيل دخول

اسم المستخدم

كلمة المرور

☐ تذكرني

هل نسيت كلمة المرور؟

تسجيل الدخول

تسجيل الدخول عن طريق منصات اخرى

Login Via Facebook

Login Via Twitter

منصة توظيف

المنصة، هي عبارة عن منصة عالمية للعمل الحر وشبكات اجتماعية حيث تتواصل الشركات والمهنيون المستقلون ويتعاونون عن بُعد

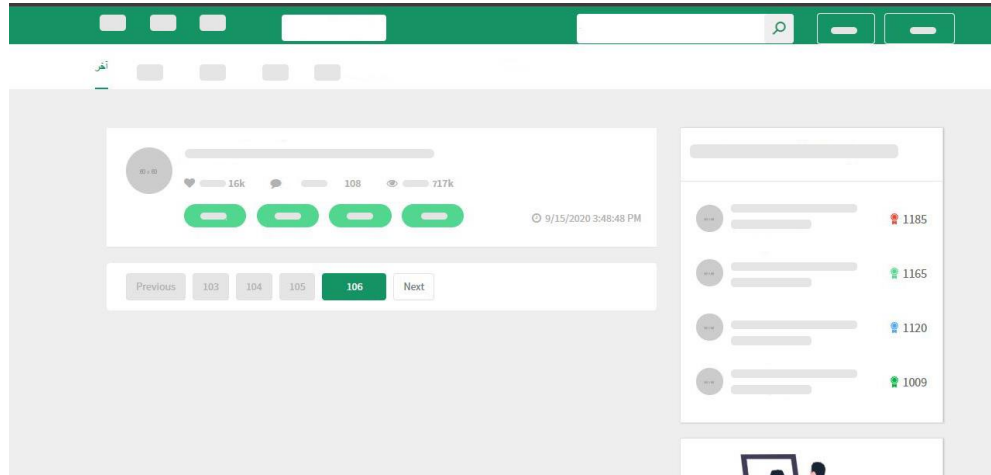
احصل على عمل الآن

(واجهة-1)

ادخال معلومات مبدئية لتمكين الدخول والمتابعة

بالإمكان البحث على آخر الاعمال دون انشاء حساب عن طريق الضغط على "احصل على عمل الان"

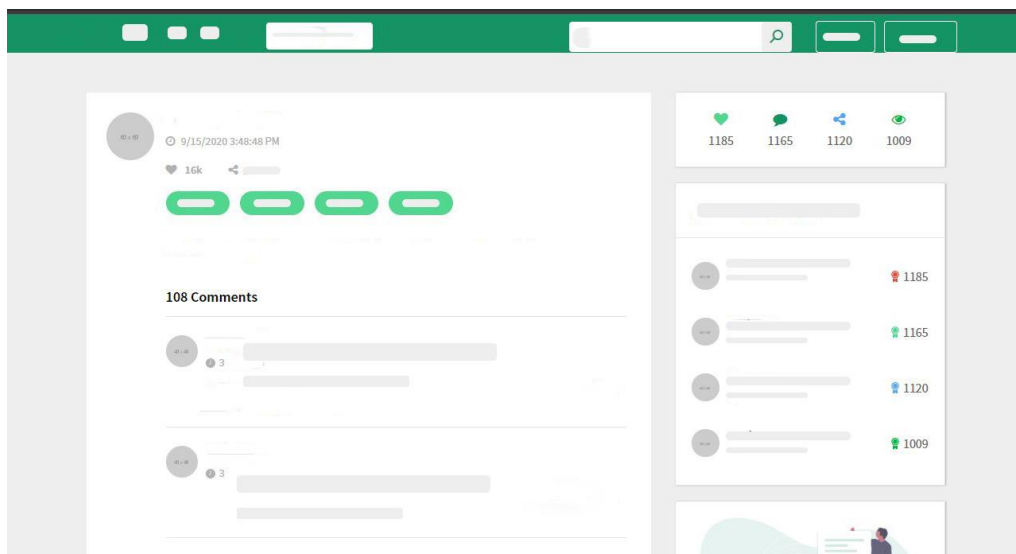
2- صفحة اخر الاعمال والاسئلة



(واجهة-2)

تشمل البحث الغير ذكي (يتطلب تسجيل دخول) والاطلاع على اخر عروض الشركات
ضمن تصنيفات وأنواع

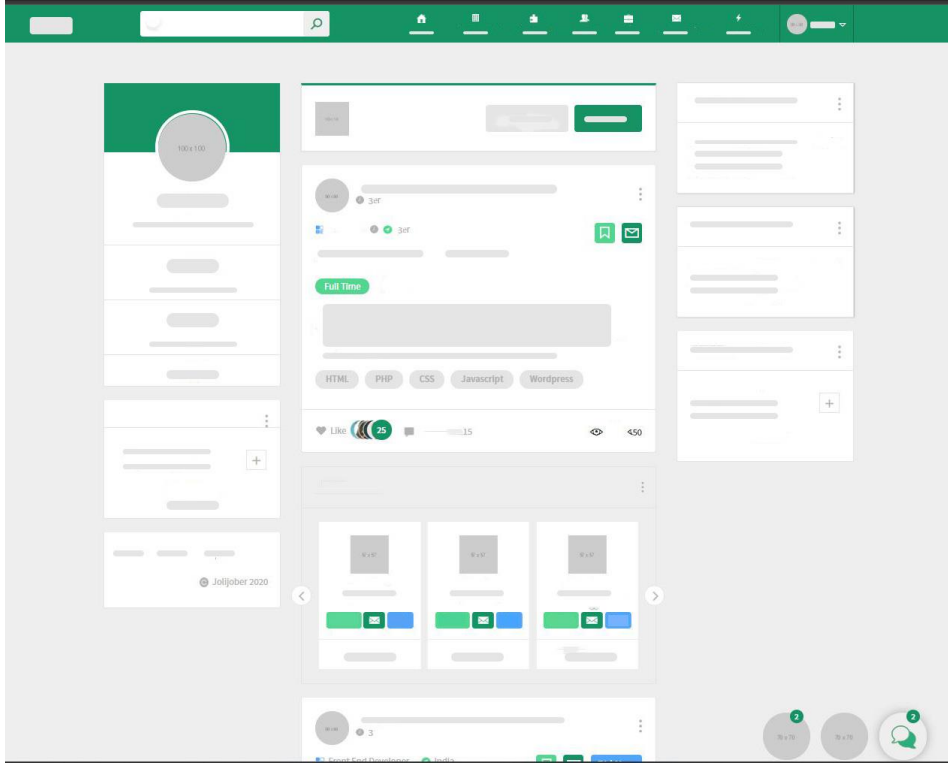
3- صفحة معلومات الطالب



(واجهة-3)

بيانات على الطلبات التوظيف التي لا تشترط تسجيل دخول

4- الصفحة الرئيسية

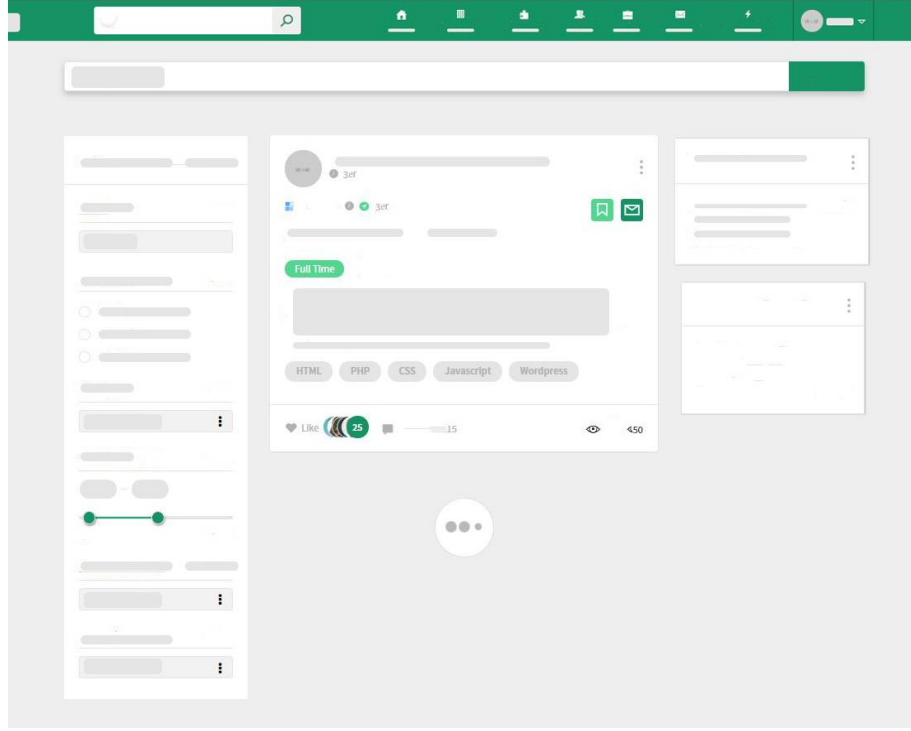


(واجهة-4)

تمثل (المنزل) تحوي على اخر المنشورات والتفاعلات بالإضافة لإمكانية متابعة الأصدقاء والتواصل عن طريق محادثات وأعلى التقييمات للشركات .. مع إمكانية انشاء منشور جديد ومشاركته مع الآخرين من اعمال ريادية أو مشاريع خاصة

5- صفحة البحث والفلتر

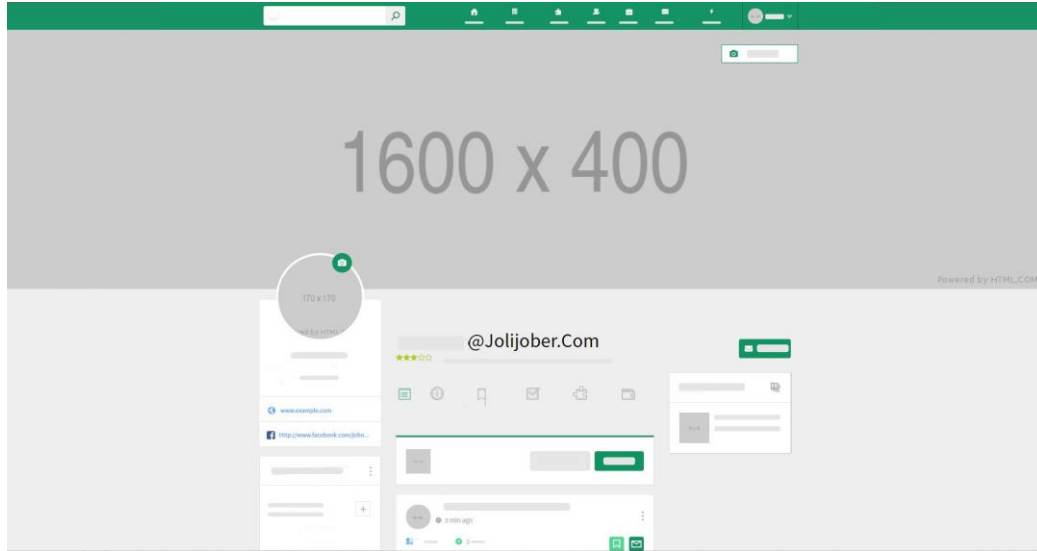
حيث تقسم لصفحتين تمتلكان نفس البنية ولكن أحدهما تهتم بالبحث عن الاعمال والوظائف المطروحة والأخرى تهتم بالمشاريع الذاتية المطروح للعمل ضمنها من جه غير رسمية و عمل غير ثابت.



(واجهة-5)

لا يقتصر البحث على المعطيات المدخلة من (مجال الراتب , نوع ساعات العمل , عدد ساعات العمل , طبيعة العمل , الاختصاص , المهارات , المتطلبات , وخيرها من الأعلى تقيما) وإنما على البيئة الأقرب للمستخدم.

6- صفحة الحساب الشخصي



(واجهة-6)

كافة المعلومات الشخصية التي تأمن بروفایل و CV ملائمة مع طبيعة المنصة والعمل فهي تهدف لمجتمع متواصل . وتقدم إمكانية التحكم بالمنشورات والأصدقاء والدخال الخبرات مع ربط بالحسابات البنكية وإتاحة إمكانية توظيف مع ضمان حق كل الشركة والمستخدم.

7- صفحة الشركة

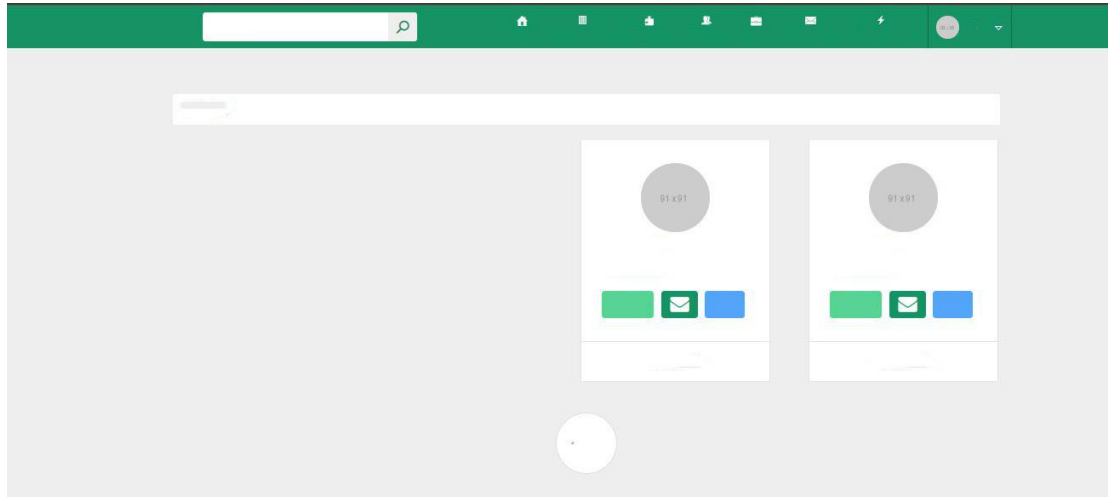
مماثلة لصفحات المستخدمين مع الاعدادات الإضافية للتحكم بالعقود ومراقبة البريد الإلكتروني والمراسلات وبعض الخصائص المحدود مثل قبول وظيفي او نشر عمل حر لا ينتمي إلى الاعمال انما للمشاريع المستقلة.

8- صفحة الشركات

صفحة بسيطة يعرض بها حسب البحث عن اسم او منطقة والاقرب إليك

9- صفحة الأشخاص

صفحة بسيطة يعرض بها حسب البحث عن اسم او منطقة والاقرب إليك



(واجهة-9)

والعديد من الصفحات الفرعية كالمحادثة و ضبط كلمة المرور والاشعارات ...

المراجع

تعد المراجع المقتبس منها هي مواقع لشركات وأصحاب الملكية التقنية والفكرية منها
Microsoft , MongoDB إلى بعض المبرمجين الذين يعملون على التوثيق:

Daniel Roth, Luke Lathan, ajcvicker كمطورين

تاريخ نشر المدونات حيث التقنيات هي وليدة العصر بتاريخ 1-2-2020

ومنها تقنيات قديمة مثل ال HTML وبعض المدونات التي تقتصر على الشرح الوافر
والمفيد

الأدوات المستخدمة:

[1] <https://github.com/>

[2] <https://code.visualstudio.com/>

[5] <https://www.mongodb.com/try/download/compass>

[4] <https://www.microsoft.com/en-us/sql-server/sql-server-downloads?SilentAuth=1&wa=wsignin1.0>

[5] <https://visualstudio.microsoft.com/vs/>

التقنيات والتضمينات الرئيسية:

[1] <https://docs.mongodb.com/drivers/>

[2] <https://docs.microsoft.com/en-us/aspnet/core/blazor/progressive-web-app?view=aspnetcore-3.1&tabs=visual-studio>

[3] <https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/choose-between-traditional-web-and-single-page-apps>

- [4] <https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-3.1>
- [5] <https://docs.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.identity.entityframeworkcore.identitydbcontext?view=aspnetcore-3.0&viewFallbackFrom=aspnetcore-3.1>
- [6] <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/>
- [7] <https://getbootstrap.com/>
- [8] <https://dotnet.microsoft.com/apps/aspnet/signalr>
- [9] <https://jquery.com/>
- [10] <https://api.jquery.com/jquery.ajax/>
- [11] <https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor>
- [12] <https://docs.microsoft.com/en-us/aspnet/core/blazor/hosting-models?view=aspnetcore-3.1>
- [13] <https://docs.microsoft.com/en-us/ef/>
- [14] <https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-3.1>

البنية والتصميم:

[1]

https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

[2] <https://medium.com/grandcentrix/software-architecture-about-dependencies-4b7258323dac>

[3] <https://docs.microsoft.com/en-us/dotnet/standard/events/observer-design-pattern>

[4] <https://www.geeksforgeeks.org/design-patterns-set-2-factory-method/>

الملفات والملحقات:

[1] https://en.wikipedia.org/wiki/Microsoft_Access

[2] <https://en.wikipedia.org/wiki/TypeScript>

[3] https://www.w3schools.com/js/js_json_intro.asp

التقنيات الأساسية:

[1] https://en.wikipedia.org/wiki/Web_application

[2] <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-3.1>

[3] <https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor>

[4] <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15>

اللغات البرمجية:

[1] <https://docs.mongodb.com/manual/tutorial/query-documents/>

[2] https://www.tutorialspoint.com/css/css3_tutorial.htm

[3] <https://www.tutorialspoint.com/sql/index.htm>

[4] <https://www.tutorialspoint.com/javascript/index.htm>

الكتب:

[1] SignalR Programming in Microsoft ASP.NET (Developer Reference) 1st Edition , By Jose M. Aguilar (Author).

[2] Pro ASP.NET Core 3 (Develop Cloud-Ready Web Applications Using MVC 3, Blazor, and Razor Pages) 8th ed. Edition

By Adam Freeman (Author).

[3] Microsoft Blazor: Building Web Applications in .NET Kindle Edition.

By Peter Himschoot (Author).

للاستفادة

تم رفع كود المشروع المقدم على الـ GitHub وهو مجاني بشكل كامل و OpenSource لمن أحب الاطلاع

راجين استخدامه لغايات مفيدة وعدم التشبث به واقتباس منه دون التعمق والفهم ومجرد كود مفتوح المصدر.. وانما لإيضاح وقابلية التطوير والتسهيل في المشاريع المماثلة.

الرابط: <https://github.com/MhozaifaA/jolijober>

The screenshot displays the GitHub repository page for MhozaifaA/jolijober. The repository is a public project with 28 commits and 0 forks. The file list shows a directory structure with folders for Jolijober, JolijoberProject.Base.Context, JolijoberProject.Hub.SignalR, JolijoberProject.Infrastructure.Model, JolijoberProject.Infrastructure.Mongo..., JolijoberProject.Infrastructure.SqlServer, JolijoberProject.Main.Repository, JolijoberProject.Security.Repository, JolijoberProject.Shared.SharedKernel, Jolijoberblazor, .gitattributes, .gitignore, JolijoberProject.sln, and README.md. The README.md file is open, showing the project name 'jolijober' and the description 'Project for the fourth year Aleppo University'.

الشكر والتقدير

بعد الشكر لله عز وجل أن أعاننا على انجاز هذا المشروع المتواضع لنتقدم بجزيل الشكر والامتنان إلى الدكتور أحمد حاج درويش على فضله بقبوله الاشراف على مشروعنا هذا وعلى ما اسداه لنا من نصائح وارشادات كانت بمثابة النبرس في كل خطواتنا.

الملحق:

لابد من سرد كيفية رفع الموقع على الاستضافة. فكما هو معروف أي موقع أو أي تطبيق دون إمكانية الوصول وإن يبقى ضمن جهاز المبرمج فهو بلا قيمة إن كان تطبيق أو برنامج ذو تواصل اجتماعي واساسة المشاركة والمستخدمين.

هذه العملية تسمى **Deploy**.

من أشهر المخدمات التي تدعم وتتعامل مع الـ RealTime فهي خدمات Azure Microsoft حيث تدعم الاتصال الغير محدود كـ Socket وتأمين بيانات العمل ضمن برمجيات التي تنتجها مايكروسوفت ... ولكن سلبيتها تعد مكلفة وغير مجانية بعض الشيء

(1)

<https://docs.microsoft.com/en-us/aspnet/core/blazor/host-and-deploy/?view=aspnetcore-3.1&tabs=visual-studio>

(2)

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/publish-to-iis?view=aspnetcore-3.1&tabs=visual-studio>

(3)

<https://www.vps.net/>

