

مشروع الحجز الفندقي



إشراف :
م.وليد العليوي



إعداد الطلاب :

محمد علي الإبراهيم

محمد ضياء خميس

عبد القادر أبو علي

Project Hotel reservation



Supervisor :
Eng Walid Aliwia



paring students :
Mohamed Ali Ibrahim
Mohamed Diaa Khamis
Abdulqader Abu Ali

{ يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ }

سورة المجادلة - آية ١١

الإهداء

إلى الذي علمني ولم يبخل عليّ بشيء وسعى لأجل راحتي ونجاحي وأعطاني من وقته وجهده ما اضاء لي طريقاً نحو النجاح: **والدي العزيز**.

إلى من ساندتني في صلاتها ودعائها إلى من سهرت الليالي تنير دربي إلى نبع العطف والحنان: **أمي الغالية**.

إلى من شدني الله بهم إلى من بوجودهم أقوى وبحنانهم أسعد وبنجاحهم أفخر: **إخوتي وأخواتي**.

إلى من مرّ الحياة بقربهم يطيب وبوجودهم أسعد إلى من ساندني ودعمني في مسيرتي إلى من لهم في قلبي كل الحب والاحترام: **أقربائي وعائلتي**.

إلى من جمعتني بهم الحياة وبادلوني المودة والإخاء إلى من أنست الروح لوجودهم معها الذين أشهد لهم بأنهم نعم الرفقاء: **أصدقائي**.

إلى من شاء الله أن أكمل دربي معهم وأكرمني الله بصحبتهم إلى الذين رسموا في وجهي البهجة والسرور وكانوا في الخير والعطاء والمنفعة مثل رائحة الزهور: **زملائي**.

إلى من أشعل الطريق نوراً وضياء والذين غرسوا فينا حب العلم والمعرفة وإلى من صححوا عثراتي وأسعدوا قلبي بكلماتهم وإرشاداتهم: **أساتذتي الكرام جميعاً**.

وفي الختام:

إلى كل من دعا لي دعوة وإلى كل من ساعدني ودعمني وكان لي عوناً.

كلمة شكر

أحمد الله سبحانه وتعالى الذي منّ علينا بنعمة العقل والدين، وهو القائل في محكم التنزيل:

“فَاذْكُرُونِي أَذْكُرْكُمْ وَاشْكُرُوا لِي وَلَا تَكْفُرُونِ” (البقرة ١٥٢)

وقد قال رسول الله صلى الله عليه وسلم:

“مَنْ صَنَعَ إِلَيْكُمْ مَعْرُوفًا فَكَافِئُوهُ، فَإِنْ لَمْ تَجِدُوا مَا تُكَافِئُونَهُ فَادْعُوا لَهُ حَتَّى تَرَوْا أَنَّكُمْ قَدْ كَافَأْتُمُوهُ”

وأيضاً وفاءً وتقديراً واعترافاً بالجميل والفضل الجزيل أتقدم بجزيل الشكر والتقدير إلى أستاذي الفاضل المهندس وليد العليوي الذي تفضل بإشرافه على هذا المشروع ولكل ما قدمه من دعم وتوجيه وإرشاد.

كما أن شكري موجه للقائمين على جامعة إدلب وأخص الأساتذة والمهندسين الأفاضل في كلية الهندسة المعلوماتية ومعهد التقاني للحاسوب وإدارة الكلية والمعهد الذين لم يبخلوا علينا بأي جهد وفي دعمنا للوصول إلى نجاحنا، وهم أصحاب الفضل فجزاهم الله كل خير.

وأخيراً أتقدم بجزيل الشكر إلى كل من مدّ لنا يد العون والمساعدة والحمد لله رب العالمين.

الفهرس

٩	الفصل الأول.....
٩	الفكرة العامة:
٩	فكرة المشروع:
١٠	أهداف المشروع:
١١	دراسة الجدوى:
١١	الجدوى الاقتصادية:
١٢	الجدوى الفنية:
١٢	الجدوى التشغيلية:
١٢	خطة المشروع:
١٣	المخاطر:
١٣	وثيقة مقترح للمشروع:
١٤	المخططات:
١٤	١. مخطط ERD:
١٥	٢. مخطط Use Case:
١٦	٣. مخطط Activity:
١٧	٤. مخطط Sequence:
١٨	٥. المخطط المعماري:
١٨	٦. مخطط غانت:
١٩	٧. مخطط Diagram:
٢٠	التقنيات المستخدمة:
٢٠	تعريف C#:

٢٠	أهداف التصميم:
٢٠	التسمية وتاريخ اللغة:
٢١	تاريخ اللغة:
٢١	المزايا الفارقة:
٢٥	فئات أنماط المعطيات:
٢٦	لمحة عن SQL:
٢٦	مقدمة عن SQL:
٢٦	SQL SERVER:
٢٨	أجزاء Microsoft SQL Server (وبعض المعلومات عنها):
٣١	أدوات تحليل قاعدة البيانات:
٣١	• SQL Server
٣١	• SQL Server Enterprise Manager
٣١	لمحة عن لغات تصميم المواقع:
٣١	HTML:
٣٢	تاريخ لغة HTML:
٣٣	بنية مستندات HTML:
٣٣	CSS:
٣٤	مميزات تقنية CSS:
٣٤	JAVASCRIPT:

٣٤	تاريخ لغة JS:
٣٥	بنية لغة JS:
٣٦	PHP:
٣٦	تاريخ PHP:
٣٦	١. PHP/FI:
٣٦	٢. PHP (3.0):
٣٧	٣. PHP (4.0):
٣٨	سبب الاختيار:
٣٨	العيوب:
٣٩	الواجهات:
٣٩	واجهة تسجيل الدخول للتطبيق
٣٩	لوحة التحكم الخاصة بالمدير
٤٠	واجهة إضافة موظف
٤٠	واجهة تخصيص
٤١	واجهة الخصوصية للمدير
٤١	واجهة إدخال بيانات النزير
٤٢	واجهة إضافة نزير
٤٢	واجهة الغرف
٤٣	واجهة إضافة غرفة
٤٤	المراجع:

الفصل الأول

فكرة عن المشروع

الفكرة العامة:

المشروع عبارة عن فندق، مجموعة من الموظفين كل موظف له عمله الخاص المكلف به، لتسهيل عملية الحجز وتنسيق الزبائن.

فكرة المشروع:

تهدف فكرة مشروع حجز فندقى إلى تسهيل عملية الحجز للعملاء وتوفير تجربة مميزة لهم عند حجزهم للأماكن المناسبة للإقامة. سيتم تطوير منصة إلكترونية تعمل على توفير قائمة واسعة من الفنادق والمنتجات المتاحة في الوجهات المختلفة.

ميزة هذا المشروع هي توفير واجهة مستخدم سهلة الاستخدام وتجربة تفاعلية ممتعة للعملاء. ستتيح المنصة للمستخدمين البحث عن الفنادق والمنتجات وفقاً لمجموعة متنوعة من المعايير مثل التاريخ، والموقع، وعدد الضيوف، والميزانية.

سيتم توفير معلومات واضحة حول خدمات الفنادق، والغرف المتاحة، والتسهيلات المقدمة، والأسعار وغيرها من التفاصيل الهامة.

كما ستقدم المنصة خدمة عملاء متاحة على مدار الساعة للرد على استفسارات العملاء وتقديم المساعدة في حالة حدوث أي مشكلة أثناء الحجز أو الإقامة.

بالتالي، سيتمكن العملاء من الاستمتاع بتجربة سلسة وموثوقة عند حجز فنادقهم والحصول على أفضل العروض والخدمات.

وايضاً عند الحجز عن طريق الموقع سيذهب العميل الى الفندق لتثبيت حجه.

أهداف المشروع:

١. إتاحة حجز الفنادق عبر الإنترنت: يجب أن يكون لديك نظام يسمح للمستخدمين بحجز الفنادق عبر الإنترنت بسهولة وفعالية.
 ٢. قاعدة بيانات الفنادق: يجب أن تتضمن قاعدة البيانات الخاصة بك معلومات شاملة عن الفندق مثل الموقع والأسعار والغرف المتاحة وتقييمات العملاء.
 ٣. نظام إدارة الحجوزات: يجب أن تحتوي على أدوات إدارة الحجوزات التي تسمح للموظفين بإدارة وتتبع الحجوزات بسهولة، بما في ذلك إلغاء الحجوزات وتغييرها.
 ٤. توفير معلومات مفصلة عن الفنادق: يجب أن يكون لديك نظام يوفر معلومات مفصلة عن الفنادق المتاحة، بما في ذلك الخدمات المقدمة والصور والتقييمات والمرافق المجاورة.
 ٥. دعم العملاء: يجب أن يتوفر نظام دعم العملاء الذي يوفر مساعدة فورية ومتاحة على مدار الساعة للعملاء الذين يحتاجون إلى مساعدة في عملية الحجز أو لديهم أي استفسارات أخرى.
 ٦. تقارير وإحصائيات: يجب أن يتم تجميع بيانات الحجوزات لإنشاء تقارير وإحصائيات تفصيلية تساعد في تحليل أداء الفنادق واتخاذ قرارات استراتيجية لتحسين العمل.
- هذه أهداف عامة يمكن أن تكون نقطة انطلاق لتطوير مشروع حجز فندق. يمكنك أيضًا إضافة أهداف إضافية تتناسب مع رؤيتك واحتياجات العملاء المحتملين .

الفصل الثاني

دراسة الجدوى:

الجدوى الاقتصادية:

- يهدف المشروع على تحسين الخدمة وإنجاز العمل بدقة وسرعة ودون تعب وإدارة الفندق بشكل أفضل وتحسين صورة الفندق وعدم تأخير النزيل بعملية الحجز
- (وهذا يؤدي لزيادة عدد النزلاء).
- يقدر الزمن لبدء الحصول على عائد من المشروع في غضون شهر.

السعر	احتياجات المشروع	
800 \$	حاسب ذو مواصفات (4GB RAM – HARD SSD) عدد (٢) وجميع مستلزماته – جهاز للدفع عن طريق البطاقة .	العتاد والمعدات
-----	-----	البرمجيات
150 \$	الموظفين عدد (1) .	القوة العاملة
10 \$	تدريب الموظفين على عمل النظام وطريقة استخدامه .	برامج التدريب
20 \$	أسلاك ومكان مناسب .	التجهيزات
20 \$	مصاريف أخرى	
1000 \$	المجموع الكلي	

الجدوى الفنية:

- يملك الفندق جزء كبير من مستلزمات المشروع والجزء الآخر يمكن توفيره بسهولة.
- يملك الموظفين خبرة فنية كاملة للعمل على النظام ولكن هم بحاجة الى بعض التوضيحات لـ عمل النظام بدون أخطاء ومشاكل.
- سيكون النظام قادر على معالجة حجم الحجوزات وتنظيمها مهما كان الضغط عالي على النظام فهو يعتبر ايضا ذو جدوى فنية.

الجدوى التشغيلية:

- نقوم بإعداد واجهة سهلة التعامل والفهم وتعمل بكفاءة عالية.
- المنظومة ستكون سهلة الاستخدام نظراً لأن الموظفين لديهم الخبرة الكافية للعمل بهذا النظام.
- المنظومة الجديدة ستكون أفضل بالنسبة للموظفين والمدراء لأنها ستوفر لديهم الوقت والجهد وستحقق احتياجاتهم.
- ستكون المنظومة سهلة الاستخدام ولكن سنوفر التدريب الكافي للعمل عليها قبل تسليمها.

خطة المشروع:

تطوير برنامج يسهل عملية الحجز عبر هذا البرنامج وتحديد تاريخ ووقت الزيارة.
المهام المطلوبة:

- تسجيل بيانات النزيل.
- حجز الغرفة المناسبة حسب متطلبات النزيل.
- تسريع عمليات الحجز.
- حساب فاتورة النزيل.
- الدفع (كاش - بطاقة).
- تحديد عدد النزلاء.
- حذف الحجز بعد مغادرة النزيل.

المخاطر:

الجدول الآتي يوضح بعض المخاطر التي قد تهدد النظام مع بعض الحلول لها:

المخاطر	الاحتمال	التأثير	الحلول
حدوث عطل بالنظام	20%	توقف عمليات الحجز لفترة معينة	إصلاح العطل بأسرع وقت ممكن
الاختراق السيبراني	5 %	الوصول لبيانات الزبائن أو تعديل الحجوزات	تشفير النظام واستخدام نظم الوصول المعقدة
انقطاع الخدمة	10%	تعطل النظام وإيقافه	خطة احتياطية وأنظمة احتياطية للتأكد من استمرارية الخدمة
انتهاك الأمن في الشبكة	15 %	خطر ع معلومات الحجز	تطبيق تدابير حماية الشبكة وتحديثها بشكل منتظم
الأخطاء في المعلومات والحجوزات	10 %	إلغاء حجوزات أو توقف سير العمل	الاعتماد على نظام موثوق ودقيق للتحقق من صحة المعلومات

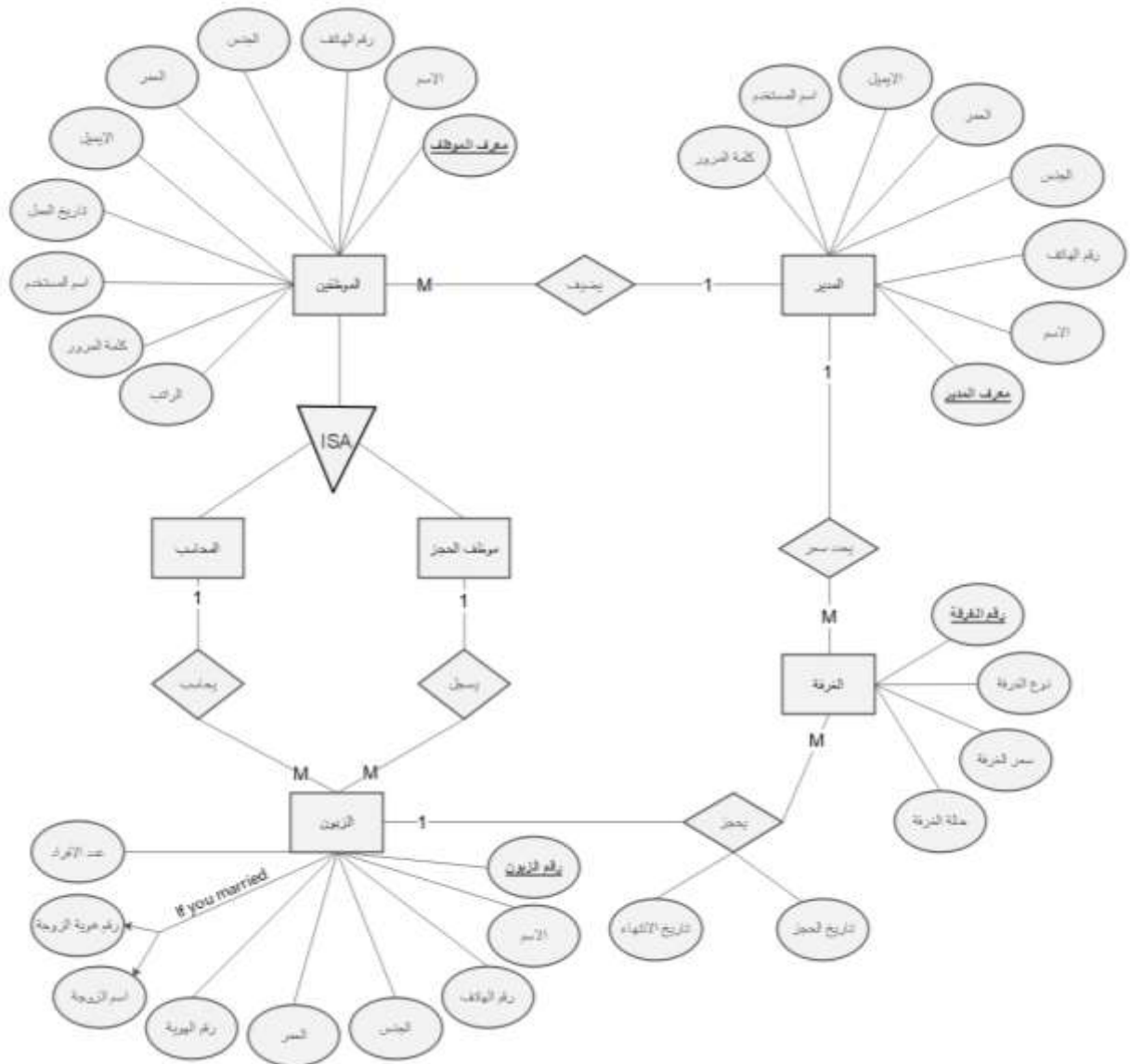
وثيقة مقترح للمشروع:

- وصف المشكلة: الازدحام والانتظار على الدور من اجل الحجز وتسجيل البيانات وإدارة الف.
- حل المشكلة: انشاء نظام يدعم الحجز عن بعد وذلك بموقع **WEP** للمستخدم و **Desktop** لإدارة الفندق.
- النظام الحالي: نظام لا يدعم الحجز عن بعد.
- النظام المقترح: نظام قادر على الحجز للنزلاء وحساب الفواتير لهم، وتوفير طرق دفع مختلفة.
- المستفيدين: موظفين الحجوزات - المدير - النزلاء.

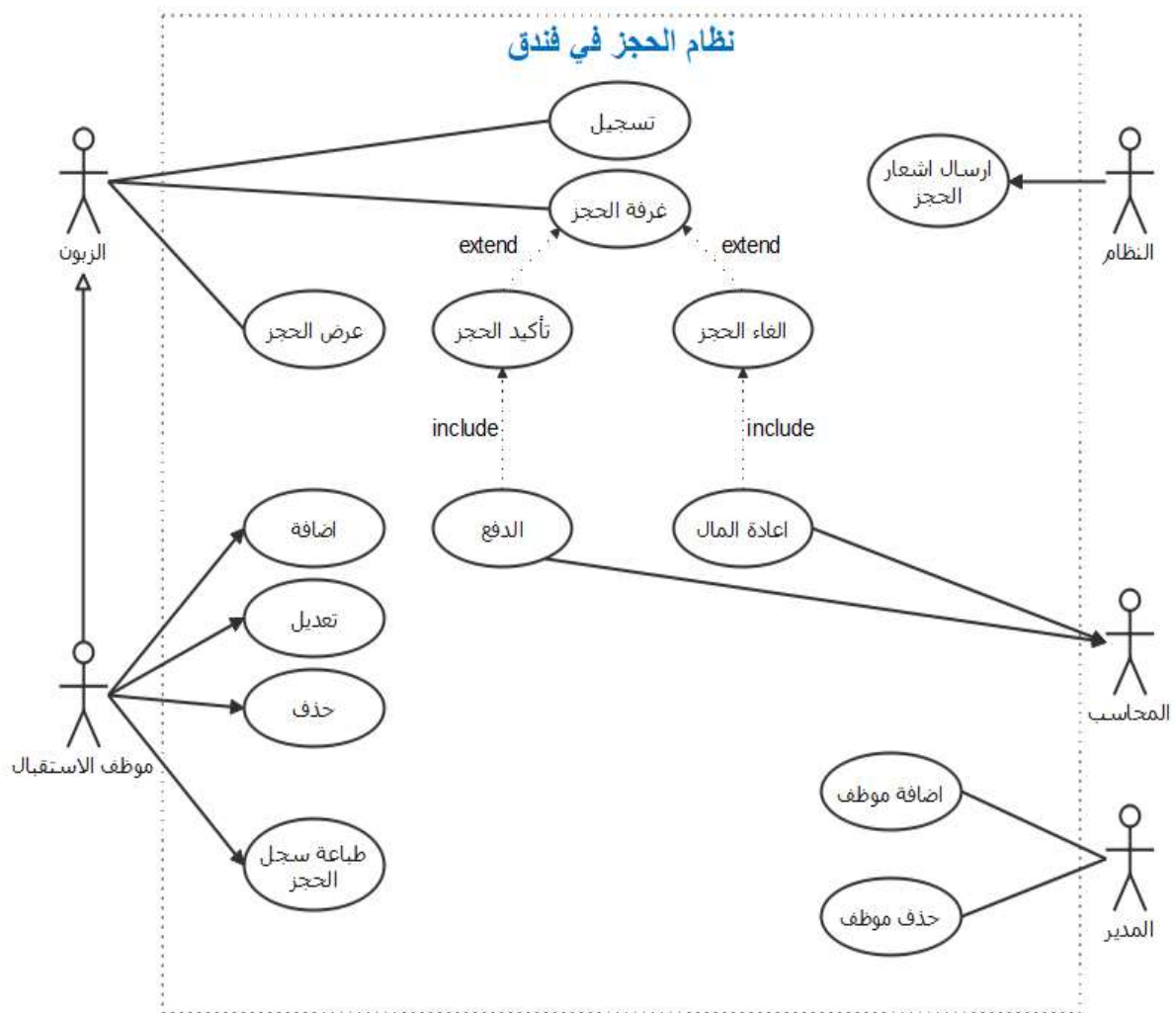
الفصل الثالث

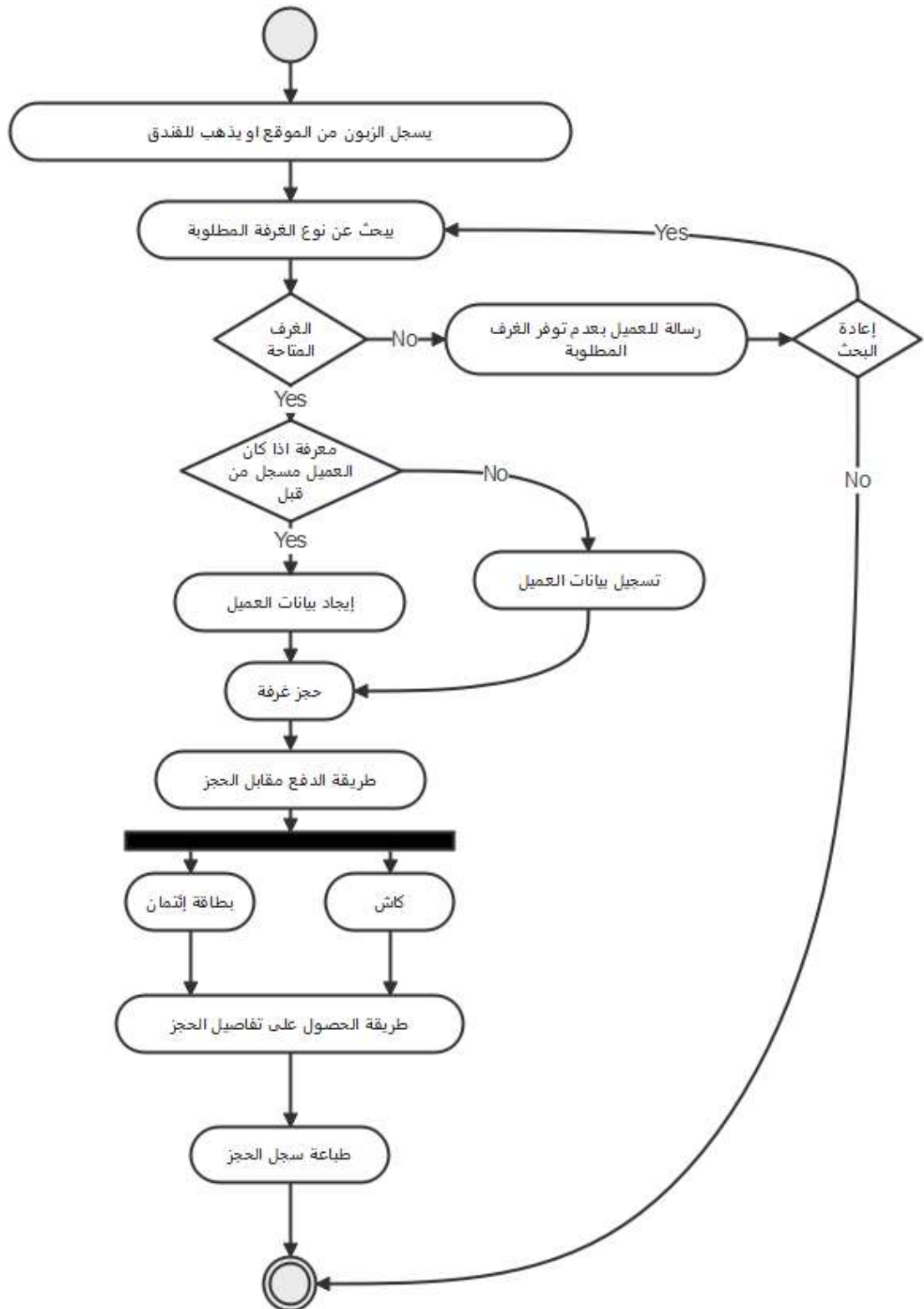
المخططات:

١. مخطط ERD:

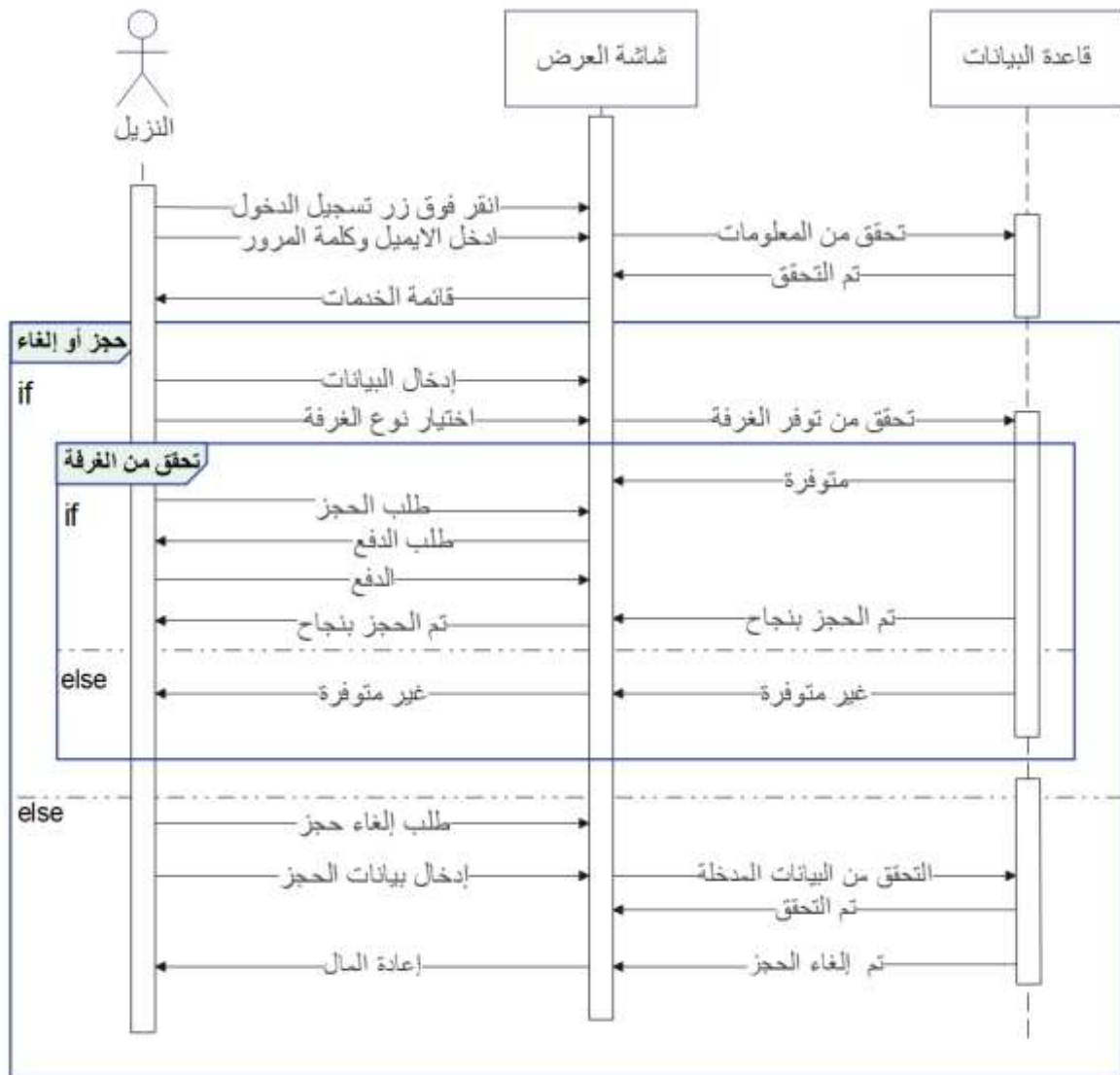


٢. مخطط Use Case:

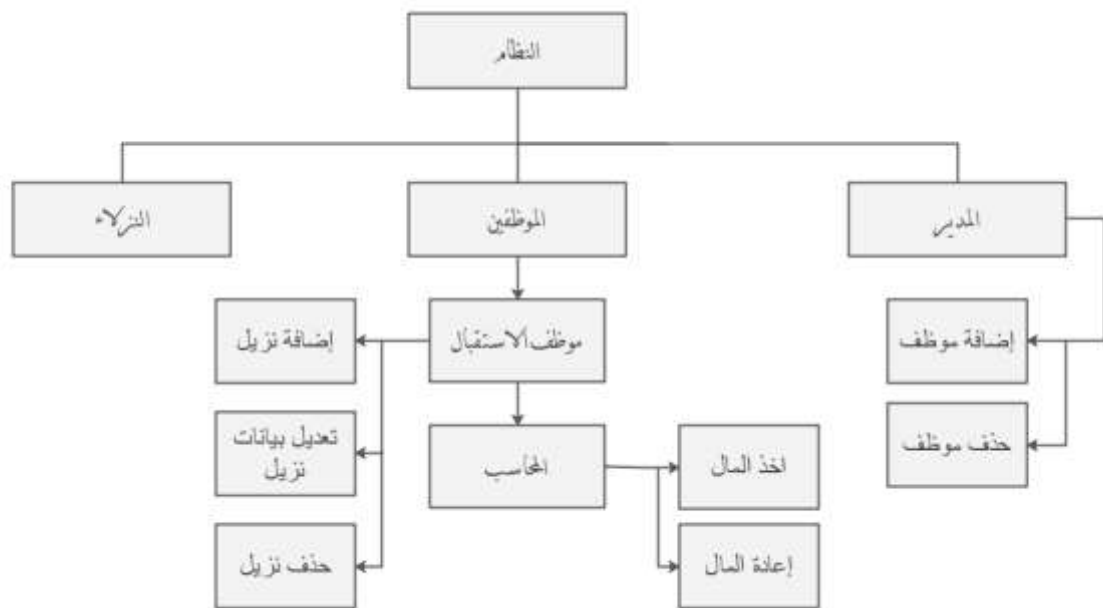




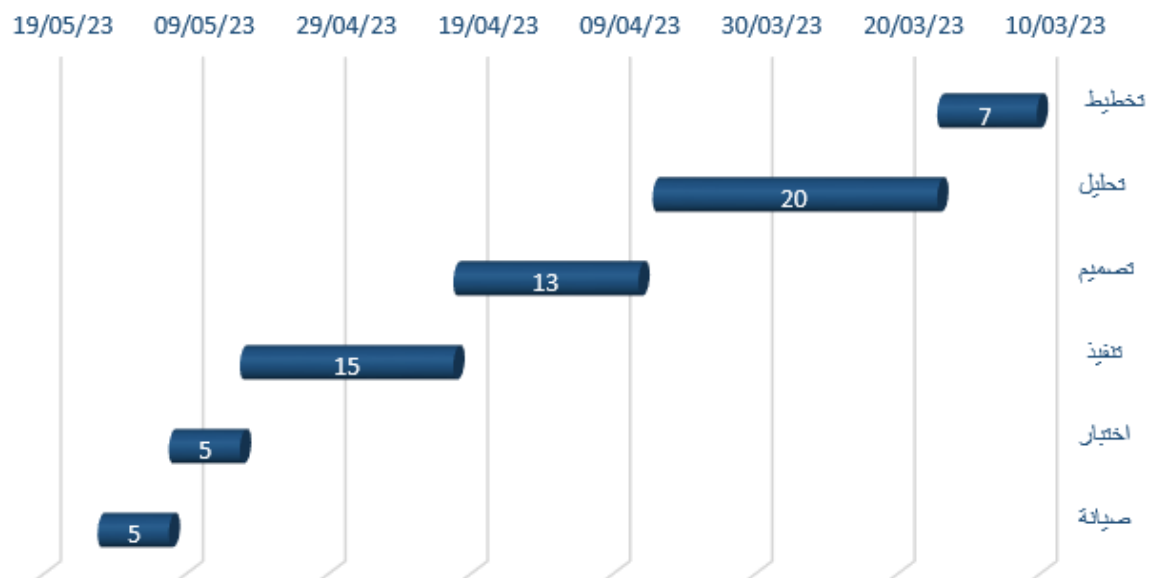
٤. مخطط Sequence:

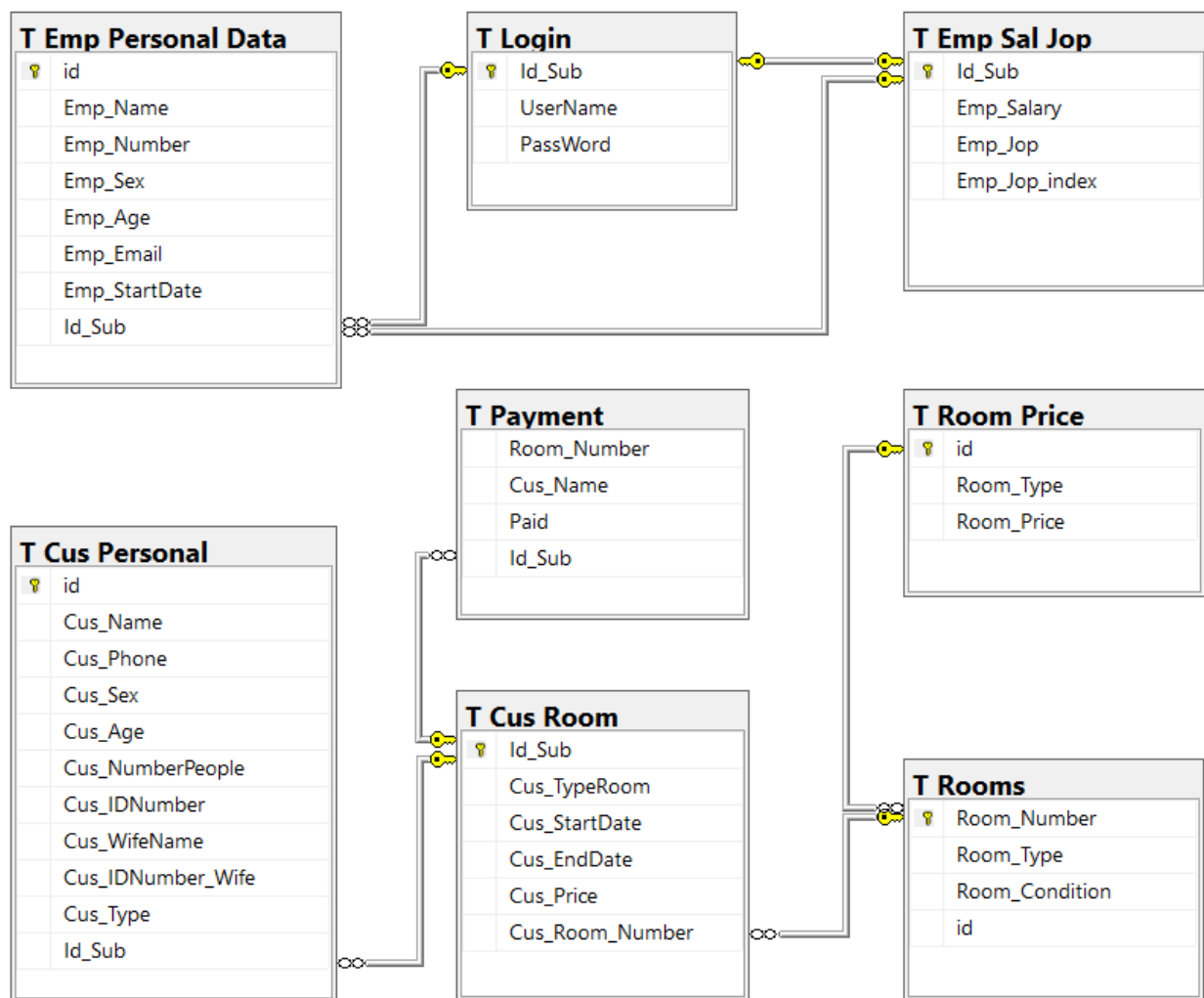


٥. المخطط المعماري:



٦. مخطط غانت:





الفصل الرابع

التقنيات المستخدمة:

تعريف C#:

هي لغة برمجة متعددة الأنماط تتمتع بكونها سكونية التتميط وأمرية وتعريفية ووظيفية واجرائية وعمومية وشيئية المنحى غرضيه التوجه باستخدام الصفوف كما تخضع لمبادئ البرمجة تركيبية المنحى. قامت مايكروسوفت بتطوير هذه اللغة في إطار عملها على تطوير وتمت الموافقة على تعبيرها من منظمة ECMA

أهداف التصميم:

- يُدرج المعيار ECMA الأهداف التصميمية التالية للغة سي شارب:
1. ينبغي أن تكون لغة سي شارب بسيطة وحديثة وعامة الاستخدام وشيئية المنحى.
 2. ينبغي أن توفر اللغة والتحقق أيضاً دعماً لمبادئ هندسة البرمجيات مثل التحقق القوي من الأنماط أو التحقق الإستاتيكي والتحقق من حدود المصفوفات واكتشاف محاولات استخدام المتحولات غير المهيئة وجمع القمامة الأوتوماتيكي. كذلك التأكيد على أهمية متانة وديمومة البرمجيات وإنتاجية المبرمج.
 3. يجب أن يتيح التصميم إمكانية استخدام اللغة لتطوير مكونات برمجية قابلة للاستخدام في البيئات الموزعة.
 4. إن محمولية الكود المصدري هدف ذو أهمية عالية ، كذلك محمولية المبرمج، خاصة بالنسبة لأولئك ذوي الخبرة بلغة سي بلس بلس ولغة سي.
 5. ينبغي أن تكون لغة سي شارب ملائمة لبرمجة تطبيقات خاصة بالنظم المضمنة والأنظمة أكانت تطبيقات ضخمة تستخدم المضيفة سواءً أنظمة تشغيل معقدة أو تطبيقات بسيطة لديها وظيفة محددة.

التسمية وتاريخ اللغة:

الاسم "سي شارب" مستوحى من الرمز الموسيقي حيث يشير الرمز # إلى أن النوتة المكتوبة مرتفعة أكثر بمقدار نصف درجة موسيقية [3]. تتشابه هذه التسمية مع اسم لغة سي بلس بلس حيث تشير "++" إلى وجوب زيادة المتحول بمقدار 1، يشبه الرمز # شكلاً من أربع يشبه الرمز # أن هذه اللغة هي زيادة بمقدار 1 على شكلاً من أربع إشارات "+" في شبكة 22 بما يعني ضمناً لغة سي بلس بلس.

تاريخ اللغة:

بدأ تطوير منصة دوت نت بكتابة مجموعة من مكتبات الصفوف، وقد استخدم نظام تصريف مدار اسمه Simple Managed C أو اختصاراً SMC للقيام بذلك.

لاحقاً وبالتحديد في كانون الثاني 1999 شكل أندرس هيلسبرغ فريقاً من المطورين بهدف بناء لغة جديدة اسمها كول بالإنجليزية: COOL يشكل الاسم اختصاراً لعبارة "لغة غرضية التوجه شبيهة بلغة السي شارب" بالإنجليزية: C-like Object Oriented Language

قررت مايكروسوفت الإبقاء على هذا الاسم إلى أنها تخلت عن ذلك لاحقاً لأسباب قانونية لها عاقبة بحقوق العلامات المسجلة.

على التوازي مع ذلك أعلن مشروع دوت نت رسمياً في مؤتمر للمطورين المحترفين PDC وأعيد تسمية اللغة إلى سي # كما تم تصدير وقت التنفيذ الخاص بلغة إيه إس بي دوت نت بالإضافة إلى مكتبات الصفوف إلى هذه اللغة.

اعتبر مصمم لغة جافا جيمس جوسلينج وبيل جوي وهو أحد مؤسسي شركة صن مايكروسيستمز التي أتت بلغة جافا أن لغة سي # ليست سوى "تقليداً" للغة جافا؛ وقد قال جوسلينج معقّباً "إنها المقصود سي # كجافا نوعاً ما ولكن بعد التخلي عن الاعتمادية والإنتاجية والأمان".

كتب كل من كالوس كرفت وأنجلكا لانجر في مقال لهما في مدونة "إن جافا وسي # لغتا برمجة متطابقتان تقريباً، هذا تكرار مضجر يفتقر للإبداع"، "من الصعب جداً الادعاء أن جافا أو سي # لغة برمجة ثورية غيرت الأسلوب الذي نكتب فيه البرامج"، "لقد استعارت سي # كثيراً من جافا والعكس صحيح، حيث تدعم سي # ميزة التعليب وفك التعليب الآن وقريباً سنجد ميزة شبيهة في جافا".

قال أندرس هيلسبرغ في تموز عام 2000 أن سي # ليست "نسخة من جافا" بل أنها "أقرب إلى لغة سي ++" من ناحية التصميم.

المزايا الفارقة:

تتفرد لغة سي # عن غيرها من لغات برمجة إطار عمل دوت نت في كونها مرتبطة بشكل وثيق بمزايا البنية التحتية المشتركة للغات البرمجة CLI فعلى سبيل المثال تنسب معظم الأنماط الحقيقية إلى قرائنها من أنماط القيمة بالإنجليزية types-value: الموجودة في CLI .

على الرغم من ذلك فإن توصيف اللغة لا يتطرق إلى متطلبات توليد الكود الخاصة بالمصرف، بمعنى أن التوصيف لا يرغب مصرف سي # على توليد كود متوافق مع وقت التنفيذ المشترك للغات CLR أو توليد

تعليمات متوافقة مع اللغة المشتركة الوسيطة بالإنجليزية : Common Intermediate Language
أو أي تنسيق آخر.

وهكذا يمكن نظرياً على الأقل مصرف سي# من توليد كود لغة الآلة مباشرة على غرار باقي المصروفات كمصرف سي ++ مصرف فورت ارن.

فيما يلي بعض من المزايا الجديرة بالاهتمام كونها تميز لغة سي# عن لغتي سي وسي++ بالإضافة إلى جافا لدى ذكر ذلك صراحة :

١. تدعم سي# التصريح الضمني عن المتحولات قوية التتميط عن طريق الكلمة المفتاحية var بالإضافة إلى التصريح الضمني عن المصفوفات المنمطة عن طريق الكلمة المفتاحية متبوعة ب مهيئ المجموعة.

٢. إن إمكانية الميتا-برمجة بالإنجليزية: Programming Meta من خلال استخدام السمات بالإنجليزية: Attributes هي جزء من توصيف اللغة، توفر العديد من هذه السمات وظيفية مطابقة لتلك التي توفرها موجهات قبل التصريف المعتمدة على المنصة المدعومة من قبل جي.سي.سي وفيجوال سي++ .

٣. يجب على مبرمجي سي# استخدام الكلمة المفتاحية virtual من أجل إتاحة تجاوز الطرق في الصفوف المشتقة على غرار للغة سي++ وخلافاً للغة جافا.

٤. تتيح الطرق الملحقة بالإنجليزية: Methods Extension لمبرمجي سي# إمكانية استخدام الطرق الستاتيكية كما لو أنها طرق خاصة بالصف، وهذا ما يمكن المبرمج من إضافة طرق جديدة إلى الصفوف عن الإحساس بضرورة وجودها في الصف نفسه وكافة الصفوف المشتقة منه.

٥. يتيح النمط dynamic إمكانية الربط مع الطرق أثناء التنفيذ بما يمكن من استدعاء الطرق وتركيب الكائنات وقت التنفيذ مثلما تفعل لغة جافا سكربت.

٦. لدى سي# الكلمة المفتاحية delegate التي تتيح تعريف مؤشرات قوية التتميط إلى توابع.

٧. لدى سي# وظيفية الإعلان عن الأحداث والتسجيل عليها من خلال استخدام delegate وغيرها بشكل يشابه وظيفية الإشارات والمقابس بالإنجليزية: slots and signals التي تضيفها بيئة تطوير كيوت إلى لغة سي++.

١. توفر سي# استدعاءات الطرق المتزامنة بما يشابه لغة جافا عن طريق استخدام السمة MethodImplOptions.MethodImplSynchronized كما أنها تدعم أقفال استبعاد التشارك عبر الكلمة المفتاحية lock .

٢. لا تسمح لغة سي# باستخدام المتحولات العامة ولا التوابع الشاملة إذ يجب التصريح عن كافة الطرق والأعضاء ضمن الصفوف، تعد الطرق الستاتيكية الخاصة بالصفوف العامة بديلاً للتوابع والمتحولات الشاملة.

٣. خلافاً للغتي سي وسي++ لا تستطيع المتحولات المحلية إخفاء المتحولات الموجودة في الكتلة البرمجية المحيطة.

٤. يوفر فضاء التسمية namespace في لغة سي# نفس المستوى من العزل الذي توفره رزمة package جافا وفضاء التسمية namespace سي++، كما أنه يتمتع بالعديد من القواعد والميزات الشبيهة برزمة جافا.

٥. تدعم سي# نمط المعطيات البوليني bool بشكل صارم، إذ تتطلب كافة العبارات البرمجية التي تتضمن شروطاً مثل عبارة while وعبارة if تعبيراً من نمط يحقق المعامل true كنمط البوليني مثلاً، بالمقارنة مع لغة سي++ التي تحوي أيضاً النمط البوليني نجد أن سي++ تتيح تحويل هذا النمط من وإلى أرقام صحيحة، نتيجةً لذلك يمكن تتطلب بعض التعبيرات مثل ifa فقط أن يكون a قابل للتحويل إلى bool مفسحة المجال بذلك لأن يكون a من النمط int أو حتى يكون مؤشراً، لا تتيح سي# مقارنة "الأعداد الصحيحة تعني صحيح أو خطأ" على اعتبار أن إجبار المبرمج على استخدام تعابير تعيد النمط bool تماماً قد يمنع حدوث من الأخطاء البرمجية الشائعة في لغتي سي و سي++ مثل العبارة if a = b حيث استبدل معامل المقارنة == بمعامل الإسناد =.

٦. تتيح سي# إمكانية استخدام المؤشرات إلى عناوين الذاكرة فقط ضمن كتل مشار إليها بالكلمة المفتاحية unsafe غير آمن وتتطلب البرامج التي تحتوي كوداً غير آمن سمحايات ملائمة كي تعمل، غالباً ما تستخدم الدلائل بالإنجليزية : References الأمانة من أجل الولوج إلى الكائنات، حيث تشير هذه الدلائل دائماً لي كائنات فعالة في الذاكرة كما أن لديها قيمة فارغة null معرفة بشكل ملائم؛ من المستحيل الحصول على دليل إن كائن تم تدميره من خلال جامع القمامة أو الولوج إلى كتلة عشوائية في الذاكرة، يمكن للمؤشر غير الآمن الإشارة إلى متحول من نمط-قيمة أو مصفوفة أو سلسلة محرفية أو كتلة من الذاكرة محجوزة على المكس، يمكن للكود غير الآمن أن يخزن ويتلاعب بالمؤشرات من خلال النمط System.IntPtr ولكن لا يمكنه الاطلاع على محتوياتها.

١. لا يمكن تحرير الذاكرة المُدارة بشكل مباشر إذ يتم تحريرها بشكل أوتوماتيكي من قبل جامع القمامة، إن جمع القمامة يعالج معضلة تسرب الذاكرة من خلال إعفاء المبرمج من مسؤولية تحرير الذاكرة.

٢. توفر لغة سي# بالإضافة إلى بنية try...catch الخاصة بمعالجة الاستثناءات البنية finally...try التي تضمن تنفيذ الكود الموجود في كتلة finally سواء حصل استثناء أم لم يحصل.

٣. لا تدعم سي# الوراثة المتعددة على الرغم من أنه من الممكن لصف أن يحقق عدداً من الواجهات، يعود السبب في ذلك إلى قرار اتخذه المعمارى الرائد في تصميم اللغة بهدف الابتعاد عن التعقيد وتبسيط المتطلبات المعمارية الخاصة بالبنية التحتية المشتركة للغات البرمجة CLI، في حال تحقيق عدة واجهات تحتوي الطريقة ذاتها تتيح سي# للمبرمج إمكانية تحقيق الطريقة عدة مرات بشكل يتناسب مع

- الواجهة التي سيتم استدعاؤها من خلالها، أو كما في لغة جافا يمكن للمبرمج أن يحقق الطريقة مرة واحدة فقط بحيث يتم استدعاء هذا التحقيق من قبل كافة الواجهات التي تحتوي تلك الطريقة.
٤. خلافاً للغة جافا تدعم لغة سي# التحويل الزائد للمعاملات، إلا أنها تدعم تحميل المعاملات الأكثر شيوعاً فقط بالمقارنة مع لغة سي++ .
٥. تتمتع سي# بكونها ذات تنميط أكثر أمناً من لغة سي++، إن التحويلات الضمنية المتاحة بشكل افتراضي في اللغة هي تلك المعدة الآمنة بشكل حصري، مثل توسيع الأعداد الصحيحة من Int16 إلى Int32 مثلاً يتم تطبيق هذه السياسة أثناء التصرف وخلال الترجمة في الوقت المناسب JIT وفي بعض الأحيان أثناء التنفيذ، لا يوجد تحويل ضمني بين النمط البوليني والنمط الصحيح وال بين أعضاء الأنماط التعدادية والنمط الصحيح باستثناء المحرف 2 والذي يمكن تحويله ضمناً إلى أي نمط تعدادي يجب الإشارة إلى أي تحويل معرف من قبل المستخدم بشكل صريح أو ضمني بشكل يخالف البواني الناسخة ومعاملات التحويل في لغة سي++ التي تعمل بشكل ضمني افتراضياً.
٦. تدعم سي# بشكل كامل التباين والتباين المعاكس للأنماط العامة على خالف لغة سي++ التي توفر دعماً محدوداً للتباين المعاكس من خلال التحكم بمعاني الأنماط المعادة في الطرق الافتراضية.
٧. لأعضاء التعدادات ضمن مجال الرؤية الخاص بها.
٨. توفر لغة سي# الخصائص كتحميل لغوي لنمط برمجي شائع يتم فيه تعريف زوج من الطرق: طريقة للحصول على القيمة بالإنجليزية: getter وأخرى لتعديل القيم بالإنجليزية: setter بهدف تغليف عمليات الوصول إلى عضو من أعضاء الصف، نتيجة لذلك لم تعد هناك حاجة لكتابة العديد من هذه الطرق بشكل تكراري، بالإضافة إلى ذلك يمكن الولوج إلى الخصائص بشكل موجز بنفس طريقة الولوج إلى الأعضاء ولا حاجة لاستخدام استدعاء الطريقة المعتاد للقيام بذلك.
٩. خلافاً للغة جافا لا تدعم لغة سي# الاستثناءات المفحوصة الاستثناءات التي يمكن أن تحصل ضمن طريقة ما، يمكن للطريقة أن تصرح عن هذه الاستثناءات كجزء من توقيعها وقد اتخذ قرار بالإحجام عن هذه الميزة نظراً لأنها قد تسبب معضلات فيما يتعلق بقابلية التوسع والتعامل مع الإصدارات المختلفة.
١٠. ابتداءً من الإصدار 2.3 تدعم لغة سي# تقنيات البرمجة الوظيفية من خلال الصفوف التابعة على الرغم من كونها لغة أمرية أصلاً.

فئات أنماط المعطيات:

يصنف نظام الأنماط المشترك أنماط المعطيات ضمن فئتين:

١. أنماط المرجع

٢. أنماط القيمة : تتصف الكائنات من نمط القيمة بأنها لا تملك محدداً مرجعياً ولا تملك خصائص المقارنة المرجعية .

فمعاملات المساواة وانعدام المساواة بين أنماط القيمة تقوم بمقارنة القيمة الحقيقية للبيانات الموجودة ضمن الكائنات مالم تكن معاملات المقارنة هذه محملةً بشكل زائد، تشتق أنماط القيمة من النمط System.ValueType ولديها دائماً قيمة افتراضية كما يمكن انشاؤها ونسخها في أي وقت، من أهم القيود المفروضة على أنماط القيمة هو عدم المقدرة على الاشتقاق من بعضها البعض إلا أنها تستطيع تحقيق الواجهات كما أنها لا يمكن أن تحوي بانياً افتراضياً بدون بارامترات إن الأنماط البسيطة مثل النمط Int عدد صحيح ذو إشارة بطول 32 بتاً (والنمط float) عدد فاصلة عائمة بطول 32 بت معرف من أي تربل إي (والنمط char) وحدة كود من نظام يونيكود بطول 16 بت (والنمط DateTime System (يعرف لحظة زمنية ما بدقة نانو ثانية هي أمثلة على أنماط القيمة، كذلك النمط enum نمط تعدادي والنمط struct بنى معرفة من قبل المستخدم .

بمقابل ذلك تمتلك أنماط المرجع مفهوم المحدد المرجعي - أي أن كل كائن من النمط المرجعي يمتلك محدداً متميزاً عن محددات بقية الكائنات حتى وإن كان البيانات الموجودة ضمن كائنين هي ذاتها، ينعكس هذا المفهوم على عمليات مساواة أو عدم مساواة أنماط المرجع حيث تقوم باختبار المساواة المرجعية ولا تختبر مساواة القيم بينها إلا في حال تحميل المعاملات المقابلة بشكل زائد كما هو الحال بالنسبة للنمط String.System ، بشكل عام لا يمكن إنشاء كائنات من النمط المرجعي ولا نسخ كائن موجود ولا إجراء مقارنات بين قيم كائنين مرجعيين ،على الرغم من ذلك توفر بين أنماط المرجع خدمات كهذه عبر التصريح عن بانٍ عام أو عبر تحقيق الواجهة الموافقة مثل ICloneable و IComparable بعض الأمثلة على أنماط المرجع النمط Object الصف الأساسي الذي تشتق منه كافة الصفوف الأخرى والنمط System. String سلسلة محرفية من النمط يونيكود و النمط System. Array الصف الأساسي الذي تشتق منه كافة المصفوفات كلا الفئتين قابلتان للتوسيع بأنماط جديدة معرفة من قبل المستخدم.

لمحة عن SQL:

مقدمة عن SQL:

تمكنك لغة SQL من إدارة قواعد البيانات بشكل كامل وإجراء جميع العمليات القياسية لإنشاء الجداول وتعبئتها بالبيانات، أو إجراء الاستعلامات عليها وكذلك الربط بين الجداول المختلفة. وقاعدة البيانات هي عبارة عن مكان أو مستودع كبير لتخزين البيانات المختلفة.

تكون البيانات في قاعدة البيانات مخزنة في عدة جداول Tables وكل جدول يتكون من صفوف وأعمدة Rows و Columns.

وفي قواعد البيانات فإننا نسمي الصفوف بالسجلات Records ونسمي الأعمدة بالحقول. الحقل: هو الذي تخزن فيه البيانات المتشابهة من حيث النوع مثل حقل الاسم أو حقل الوظيفة. السجلات: هو الذي تخزن فيه بيانات شخص معين.

قواعد اللغة:

- لا تفرق بين الحروف الكبيرة والصغيرة لغة SQL
- لا تهتم بالمسافات البيضاء لغة SQL
- تنتهي جميع الاستعلامات بالفاصلة المنقوطة

SQL Server:

❖ هو عبارة عن قاعدة بيانات مركزية تقوم بإدارة قواعد البيانات وتوزيعها عبر شبكة الكمبيوتر لقد بدء ظهور قواعد البيانات المركزية بشكل مكثف في نهاية الثمانينات، طبعاً هي موجودة بطريقة أو بأخرى منذ فترة أطول ولكن استخدامها اقتصر إلى بعض المؤسسات الضخمة فقط لقد كان الجميع قبل ذلك يستخدم الب ارمج التي صنعت بلغات البرمجة العادية والتي كانت تخزن بياناتها في ملفات خاصة بها، والمشكلة الأساسية في تلك البرامج كانت محدودية استخدام البيانات، أعني أنك ال تستطيع الاستعلام عن البيانات بطريقة أخرى غير الطريقة التي صمم بها البرنامج

❖ كمثال: لنفترض بأنك تستخدم برنامج صمم بواسطة لغة برمجة ما ليخزن بياناته في ملفات خاصة به وليس في قواعد البيانات، وكان أحد فروع برنامجك هو الاستعلام عن البضائع، مهما تتعب في تصميم هذا الاستعلام، فقد تجد دوماً من يأتي لك بفكرة جديدة لاستعلام ما فقد يطلب الشخص كل البضائع التي تم استلامها في تاريخ معين، من مورد معين وتم بيعها في تاريخ معين إلى زبون معين، قد يكون برنامج قادر على القيام بذلك، ولكنه قد يعقد الاستعلام ويضاف إلى الاستعلام السابق ترتيب البضائع

حسب تاريخ الاستلام مثال فقد ال تك ونن وضعت تلك الميزة في برنامجك عند تصميمه، ولذلك فلن يتمكن المستخدم النهائي الحصول على طلبه بالضبط .

❖ ومن هنا بدأت فكرة قواعد البيانات عامة، وكانت الفكرة إيجاد طريقة موحدة لحفظ البيانات، وإيجاد برنامج قادر على جميع أنواع الاستعلامات من قاعدة البيانات، فلو فشل برنامج السابق، فبإمكان المستخدم فتح ملف البيانات عن طريق برنامج الاستعلام الرئيسي الذي يأتي مع قاعدة البيانات وأجراء كل الاستعلامات التي يحلم بها ومن هنا ظهرت قواعد البيانات وظهرت أيضا لغة SQL المخصصة للاستعلام في قواعد البيانات، وبدأت تتطور وانتقلت العديد من الشركات لاستخدامها، نظرا لسهولة التعامل معها وسرعة برمجتها ولكن مع زيادة حجم المؤسسات وبداية ظهور شبكات الكمبيوتر، أصبحت قواعد البيانات بحاجة إلى أن تعمل على أكثر من جهاز في نفس الوقت، فتطورت برامج إدارة قواعد البيانات وأصبحت قادرة على فتح نفس الملفات المخزنة في الجهاز المركزي من عدة أجهزة كمبيوتر في نفس الوقت ومن أمثلتها Dbase و Paradox وغيرهما ولقد حل السابق بعض المشاكل ولكن ليس كلها ، لان مع زيادة حجم البيانات وزيادة عدد الأجهزة المتصلين بالشبكة أصبح صعب إدارة ملفات قاعدة البيانات المخزنة على الجهاز المركزي ، كما أن أمنها كان معرض للخطر دائما ، ف بإمكان الجميع الوصول إلى الملف المركزي الذي يحتوي على البيانات ويعبث به ، أو حتى أن يصل إلى بيانات لا صالحة له باستخدامها.

❖ كما أن الاستعلامات المتزايدة على قواعد البيانات ازادت من الضغط على الشبكة فكما تعلم يتطلب الاستعلام عن شخص ما، يتطلب البحث في كل قاعدة البيانات حتى يجدها بها.

❖ فعلى سبيل المثال: لو كان لديك جدول في قاعدة البيانات وبه معلومات عن عشرة آلاف نوع من البضائع وطلبت أن من جهازك كل البضائع التي استلمت في تاريخ معين، فعليا سيقوم جهازك بإحضار العشرة آلاف سجل في قاعدة البيانات إلى جهاز وذلك عبر الشبكة وسيفترهم ويفرز البيانات المطلوبة ولكنه ضغط على الشبكة بالحصول على كل تلك المعلومات معا، طبعا الشبكة قد تتحمل طلب أو طلبين معا.

❖ ولكن ماذا بالنسبة للبنوك مثالا: هناك آلاف السجلات وعشرات العمليات في نفس الوقت و لذلك بدئت الحاجة إلى تطوير قواعد البيانات العادية ، ومن هنا ظهرت قواعد البيانات المركزية ، فهي عبارة عن برنامج يعمل بداخل الجهاز المركزي ويخزن البيانات فيه أيضا ولكن الاختلاف بينها وبين السابق ذكره أن طلب البيانات لا يتم إلا من خلال البرنامج الذي يعمل في الجهاز المركزي والذي يسمى محرك

قواعد البيانات المركزية ، وبذلك يكون قد تم فصل المستخدم النهائي عن الملف الرئيسي لقواعد البيانات فلو كنت بحاجة إلى استعلام معين، فسيقوم برنامجك بطلب ذلك الاستعلام من محرك قواعد البيانات المركزية الموجود في الجهاز المركزي، حيث بدوره سيقوم هو بالاستعلام ومن ثم يعطي النتيجة فقط للجهاز العادي الذي طلب الاستعلام وبذلك يكون قد أنهى كابوس إغراق الشبكة بالبيانات كما أن محرك قواعد البيانات الرئيسية مسؤول عن حماية البيانات، قبل كان الجميع يستطيع الوصول إلى كل البيانات المخزنة، ولكن باستخدام النظام الجديد، أصبح فقط من لديهم صالحة الوصول قادرين على ذلك، كما أصبح بإمكان مدير الشبكة إعطاء صلاحيات مختلفة للمستخدمين، فقد يمنع مستخدم من إضافة بضاعة جديدة على جدول البضاعة ويسمح له بالنظر عليها فقط، وقد يمنعه نهائياً من الوصول إلى جدول الموردين مثال، أو فقط على جزء من جدول الموردين مثل عناوينهم أو أرقام هواتفهم مثلاً.

❖ كما أن محرك قواعد البيانات المركزية أصبح يقوم بعمليات النسخ الاحتياطي والحفاظ على البيانات من التلف أتماتيكي، وذلك بفحصها باستمرار ونسخها على أشرطة النسخ الاحتياطي وإعلام مدير النظام بأي مشاكل صغيرة بداخلها.

وهكذا استمر تطور قواعد البيانات المركزية إلى يومنا هذا وظهر العديد منها في السوق مثل:

Borland, Informix, Sybase, IBM DB2, SQL Server 7, Oracle 8, Intranasal

أجزاء Microsoft SQL Server (وبعض المعلومات عنها):

يتكون برنامج Microsoft SQL Server 2000 من عدة أجزاء مهمة مترابطة مع بعضها البعض، مثل محرك قواعد البيانات، ملفات التصليح، شاشة الإدارة والمراقبة وغيرها من الأجزاء والتي سأحاول أن أشرح معظمها في مقالي هذا والأجزاء هي:

- **SQL Server Enterprise Manager**: وهو البرنامج الأساسي الذي يقوم بإدارة النظام بالكامل، فمنه تستطيع إنشاء قواعد البيانات، وإنشاء الجداول والاستعلامات، وإعطاء الصلاحيات وغيرها من الأمور ونسخها احتياطياً ويقوم ذلك البرنامج بتصنيف أجزاء النظام إلى كائنات وخصائص لتلك الكائنات وذلك في شكل مشابه لشكل مستكشف الويندوز، وذلك لتسهيل استخدامه من معظم المستخدمين.
- **SQL Server Client Network Utility**: وهو برنامج صغير يساعدك في توصيف أجهزة تحمل محرك قواعد بيانات لـ SQL Server لإدارتها عن بعد عن طريق الـ Enterprise Manager ويستخدم في حالة الحاجة إلى إدارة جهاز مركزي بعيد في شبكة أخرى كإنترنت مثال: Utility SQL Server Network يحدد البروتوكولات التي يمكن لمحرك قواعد البيانات من استخدامها لتوزيع البيانات على الشبكة الداخلية وعلى إنترنت.

- **SQL Server Performance Monitor**: وهي إضافة لنظام المراقبة الرئيسي الخاص ب NT Windows حيث تسمح لك تلك الإضافة بمراقبة محرك قواعد البيانات المركزية عن طريق البرنامج السابق، فعن طريقه تستطيع معرفة تفاصيل دقيقة عن عدد المتصلين بقاعدة البيانات في أوقات مختلفة، أو عن حجم الضغط عليها، عدد الطلبات، حجمها في القرص الصلب، الخ.
 - **SQL Server Profiler**: برنامج رائع لمراقبة كافة أوامر ال SQL Transact المرسل من المستخدمين إلى النظام المركزي، فهو يعرض لك الأوامر المرسل و وقت التنفيذ لحظة بلحظة، وهو رائع لاستخدام المبرمجين، يمكن استخدامه لوحده أو كجزء من برنامج الإدارة الرئيسي.
 - **SQL Server Query Analyzer**: إن رغبت بإصدار الأوامر مباشرة إلى محرك قواعد البيانات المركزية عن طريق لغة Transact SQL ، فهذا هو البرنامج الذي سيساعدك في القيام بذلك، ولكن ال يقتصر عمله على هذا فقط، فهو قادر على احتساب الوقت المخصص لإجراء تلك الأوامر، وإن كانت أوامر معقدة فهو قادر على تفكيكها ورسم خريطة رسومية لطريقة تنفيذها والوقت الذي ستأخذه كل خطوة بها، وبذلك سيساعدك بإعادة كتابة الأوامر لتجعلها أفضل وأسرع
 - **SQL Server Service Manager**: وهو برنامج صغير يعمل مع تشغيل الجهاز ويبقى في شريط الأدوات بجانب الساعة ويخبرك عن حالة محرك قواعد البيانات المركزية، أن كانت تعمل أم لا، كما تستطيع بواسطته تشغيل وإيقاف محرك قواعد البيانات أو البرامج المترابطة معه.
 - **SQL Server Books Online**: وهي مجموعة ضخمة من الكتب والمراجع حول كل ما يتعلق بذلك النظام بالتفصيل، حوالي عشرة آلاف صفحة إن حاولت طباعتها، أن كنت لا تصدق ذلك فحاول، ما عليك إلا وان تضغط على زر الطباعة أعلى البرنامج ومن ثم تجلس بجانب الطباعة وتنتظر.
 - **SQL Server OLAP**: هذا نظام آخر ضخم مرفق مع Microsoft SQL Server 2000 ويسمح لك بناء قواعد بيانات ثلاثية الأبعاد (كما اسميها أنا) وذلك باستخدام الجداول والبيانات من قواعد البيانات مسبقا وذلك بهدف تحليلها والحصول على تقارير منها، ذلك البرنامج هو من التكنولوجيا الجديدة والتي ترفق مجاناً مع Servers SQL وهو مكلف للغاية إن حاولت شركات أخرى مثل أوراكل أو غيرها، وسأقوم بشرح ذلك النظام وطريقة برمجته في المرحلة المتقدمة.
- ✚ هناك مجموعة أخرى من البرمجيات الصغيرة والمخصصة لمحبي استخدام شاشات DOS القديمة، حيث تسمح لهم تلك البرمجيات بالقيام بعدة عمليات على النظام كما أن هناك ثلاثة كائنات مخصصة لبرمجة النظام، وتستخدم بواسطتهم القيام بكل ما يقوم به برنامج الإدارة المركزي وذلك بكل سهولة، حيث تستطيع صناعة برنامج إدارة خاص بك لو رغبت بذلك

كما هناك أيضا برنامج Language Query English والذي سأشرح طريقة التعامل معه مستقبلاً وكما ذكرت من قبل فأنت تستطيع بذلك البرنامج إصدار الأوامر بلغة إنجليزية عادية إلى محرك قاعدة البيانات، حيث سيقوم بتحليلها والإجابة عليك

تلك كانت البرامج التي يتكون منها: Microsoft SQL Server 2000

من الضروري التحدث قليلاً عن محرك قاعدة البيانات قبل بدئك باستخدام برنامج Microsoft SQL Server 2000 حيث سأحاول أن اشرح طريقة عمل ذلك المحرك وأجزائه وحدوده القصوى، اعتقد بأن ذلك معلومات مهمة على الجميع أن يعرفها، أو على الأقل يعرف القليل عنها

يتكون المحرك الخاص بقاعدة البيانات من ثلاثة برامج معتمدة على بعضها البعض، مع العلم أيضاً انه قادر على الاستغناء عنها والعمل لوحده، ولكن ذلك سيقلل من مميزاته قليلاً، والأجزاء هي:

- MS SQL Server
- محرك قاعدة البيانات، وهو المسؤول عن كل العمليات على البيانات
- SQL Server Agent

برنامج إدارة محرك قاعدة البيانات، وهو يختلف عن البرنامج المخصص لإدارة كل النظام، فصراحة أنت لن تتعامل مع ذلك البرنامج مباشرةً، فهو يتميز ببعض الذكاء الاصطناعي وهو مخصص لتنفيذ العمليات المجدولة حسب توقيت معين بداخل محرك قاعدة البيانات ، مثل إصلاحها وفحصها وإجراء النسخ الاحتياطية منها ، وكتابة التقارير حول قاعدة البيانات وغيرها بالنسبة لمحرك قاعدة البيانات المركزية فهو يستخدم ثلاثة قواعد بيانات في عمله أهمها:

قاعدة البيانات Master والتي يخزن بها المعلومات الأساسية عن كل النظام ، مثل أماكن قواعد البيانات الأخرى ، أسماء الجداول التي تحتوي عليها ، وأمور أخرى عديدة ، قاعدة البيانات تلك هي قاعدة بيانات صغيرة ، ولكن فقدانها يؤدي الى فقدان النظام كله ، أي فقدان كل قواعد البيانات وكل البيانات التي بها، لذلك يجب نسخها احتياطياً دوماً

ثاني قاعدة بيانات مستخدمة من المحرك هي Tempdb ويستخدمها المحرك كمخزن تخزين مؤقت، وذلك في حال قيامه بعمليات طويلة، وهي غير مهمة بالنسبة لنا، فهو يقوم بمسح محتوياتها مئات المرات يومياً

ثالث قاعدة بيانات هي Model ويستخدمها النظام عند إنشاء قاعدة بيانات جديدة، حيث يقوم بنسخ محتوياتها إلى قاعدة البيانات الجديدة، وهي لا تهمنا أيضاً ولا يجب نسخها احتياطياً، إلا إذا كنا عدلنا

عليها، حيث نستطيع اعتماد مواصفات قياسية لكل قاعدة بيانات جديدة نصنعها، ونستطيع تخزين تلك المواصفات في قاعدة البيانات تلك ومن ثم نصنع قواعد البيانات الخاصة بنا.

أدوات تحليل قاعدة البيانات:

• SQL Server:

هو عبارة عن قاعدة بيانات مركزية تقوم بإدارة قواعد البيانات وتوزيعها عبر شبكة الحاسب، يتكون برنامج Microsoft SQL Server 2008 من عدة أجزاء مهمة مترابطة مع بعضها البعض، مثل محرك قواعد البيانات، ملفات التصليح، شاشة الإدارة والمراقبة وغيرها من الأجزاء

• SQL Server Enterprise Manager:

وهو البرنامج الأساسي الذي يقوم بإدارة النظام بالكامل، فمنه تستطيع إنشاء قواعد البيانات، وإنشاء الجداول والاستعلامات، وإعطاء الصلاحيات وغيرها من الأمور، ونسخها احتياطياً ويقوم ذلك البرنامج بتصنيف أجزاء النظام إلى كائنات وخصائص لتلك الكائنات، وذلك لتسهيل استخدامه من قبل معظم المستخدمين.

لمحة عن لغات تصميم المواقع:

:HTML

هي لغة توصيف النص الفائق بالإنجليزية: Hyper Text Markup Language اختصار لها HTML هي لغة توصيف تستخدم لإنشاء وتصميم مواقع الويب، وتعتبر هذه اللغة من أقدم اللغات وأوسعها استخداماً في تصميم، HTML هيكل صفحة الويب وتعطي متصفح الإنترنت وصفاً لكيفية عرضه لمحتوياتها، يمكن أن تساعده تقنيات مثل أوراق الأنماط المتتالية (CSS) ولغات البرمجة النصية مثل جافا سكريبت تستقبل متصفحات الويب مستندات HTML من خادم الويب أو من نظام الملفات وتعرضها، ووظيفة لغة HTML هي وصف بنية صفحات الويب هيكلياً.

العناصر في HTML هي اللبنة الأساسية لبناء مستندات HTML، إذ نستطيع عبرها إضافة الصور والكائنات التفاعلية مثل النماذج أو ملفات الفيديو والصوت؛ وتستطيع أيضاً إنشاء مستندات منظمة عبر استخدام وسوم للتصريح عن الفقرات والعناوين والروابط والاقتباسات والجداول وغيرها.

يمكن للغة HTML أن تُصمّن برامجَ مكتوبةٍ بلغات مثل جافا سكريبت لتعديل سلوك ومحتوى صفحات الويب؛ وإضافة شيفرات أوراق الأنماط المتتالية CSS تؤدي إلى تعريف شكل وتخطيط المحتوى.

تاريخ لغة HTML:

في عام 1980، قام الفيزيائي تيم بيرنرزلي والذي كان عاملا في المؤسسة الأوروبية للأبحاث النووية سيرن باقتراح واعداد نموذج بدئي لنظام يمكن باحثي سيرن من استخدام ومشاركة المستندات، وفي عام 1989 قام بكتابة مذكرة يقترح فيها نظام نص فائق hypertext مبني على الإنترنت، وقام بوصف لغة HTML وبكتابة برامج المزود والمتصفح في أواخر عام 1990.

كان أول وصف للجمهور من الاتش تي ام ال وثيقة تسمى علامات الاتش تي ام ال ذكر لأول مرة على شبكة الانترنت عن طريق بيرنرز لي في أواخر عام 1991، فهو يصف 18 من العناصر الأولى التي تتألف منها ، نسبيا التصميم بسيط في الاتش تي ام ال بإستثناء علامة الارتباط التشعبي ، هذه تأثرت بقوة في (الاس جي ام ال كويد) ، اسست ال (الاس جي ام ال) على شكل وثائق في منزل سيرن ، أحد عشر من هذه العناصر لا تزال موجودة في الاتش تي ام ال.

لغة توصيف النص التشعبي هي لغة العلامات التي تستخدم متصفحات الويب لتفسير وتأليف النص والصور وغيرها من المواد في صفحات الويب المرئية أو المسموعة.

يتم تعريف وتوصيف الخصائص الافتراضية لكل بند من الاتش تي ام ال في المتصفح، وهذه الخصائص يمكن تغييرها أو تحسينها بواسطة استخدام مصمم صفحة ويب اضافية من الاسي اس اس، تم العثور على العديد من عناصر النص في عام 1988 (أي اس أو) تقرير التقنية (تي ار 9537) تقنيات لاستخدام (الاس جي ام ال) الذي يغطي بدوره ملامح اللغات تنسيق النص في وقت مبكر مثل تلك المستخدمة من قبل الأمر الجريان السطحي وضعت في 1960 في وقت مبكر ل CTSS (التوقيت متوافق نظام تقاسم) نظام التشغيل: وقد استمدت هذه الأوامر التنسيق من الأوامر المستخدمة من قبل عمال التجميع على تنسيق المستندات يدويا.

ومع ذلك، يستند مفهوم SGML من معمم العلامات على عناصر (نطاقات متداخلة مع سمات المشروح) بدلا من مجرد آثار الطباعة، مع أيضا الفصل بين هيكل و العلامات، وقد تم HTML انتقلت تدريجيا في هذا الاتجاه مع CSS بيرنرزلي يعتبر تطبيق HTML من SGML تم تعريفه رسميا على هذا النحو من قبل فريق عمل هندسة الإنترنت (IETF) مع منتصف عام 1993 نشر أول اقتراح لمواصفات HTML : لغة توصيف النص التشعبي (HTML) إنترنت مشروع من قبل بيرنرزلي و دان كونولي، الذي تضمنت نوع الوثيقة SGML تعريف لتعريف النحوي.

بنية مستندات HTML:

تتألف صفحات HTML من عناصر، والتي تتألف عادةً من وسم بداية ووسم نهاية، ويكون بينهما محتوى نصي عادةً؛ ويمكن أن نضع في وسم البداية بعض الخصائص التي تُغيّر من سلوك العنصر أو تؤدي إلى ضبطه. لاحظ أن بعض العناصر يمكن أن تتشعب داخل بعضها بعضًا. الأقسام الأساسية للعناصر هي:

- وسم البداية: وهو يحتوي على اسم العنصر، موضوعًا ضمن قوسين على شكل زاوية، وقد يلي الاسم الخصائص التي تؤثر عليه مثلًا `<p>`.
- وسم النهاية: وهو يحتوي على اسم العنصر أيضًا مسبقًا بخط مائل قبله للإشارة إلى نهاية العنصر؛ لاحظ أن نسيان وسم النهاية قد يسبب أخطاءً في بعض الأحيان، لذا خذ حذرك وتذكره. مثلًا `<p/>`.
- المحتوى: وهو موجود بين وسمَي البداية والنهاية، ويُملأ في معظم الأحيان محتوى العنصر.
- العنصر: هو وسم البداية ووسم النهاية إضافةً إلى المحتوى.

قد تحتوي العناصر على خصائص، وهذه الخصائص بعضها اختياري وبعضها إجباري، وهناك خصائص تسمى بالخصائص العامة والتي يمكن استخدامها على جميع العناصر.

تتألف الخاصية في أغلب الحالات من اسم الخاصية ثم قيمة تليها.

CSS:

صفحات التنسيق النمطية بالإنجليزية: Cascading Style Sheets اختصار لها CSS

هي لغة تنسيق لصفحات الويب تهتم بشكل وتصميم المواقع، صممت خصيصًا لفصل التنسيق (الألوان - الخطوط - الأزرار) عن محتوى المستند المكتوب بلغة HTML وينطبق ذلك على الألوان والخطوط والصور والخلفيات التي تستخدم في الصفحات، بمرونة وسهولة تامة.

هذه التقنية تعنى بالمظهر الكلي لصفحات مواقع الويب من ألوان وصور وغيره. ويمكن إضافته للصفحة بعدة طرق أفضلها التضمين الخارجي بكتابة شفرة السي إس إس في ملف منفصل، وقد تم تطويرها لتصل حاليا إلى سي إس إس 3 والذي أضيف إليه عدة إضافات لم تكن متاحة في الإصدار السابق وكان إنجازها يتم فقط بجي كويري.

أطلقها مجلس ويب / رابطة الشبكة العالمية، كمواصفة قياسية لتوصيف مظهر وثائق ويب من محددات للخطوط والألوان والتنسيق.

مميزات تقنية CSS:

تهدف تقنية سي إس إس إلى فصل محتوى الصفحات عن مظهرها بحيث يعطي عدة مكاسب:

١. جعل الصفحة أبسط وتفيد فقط ما عملت من أجله بحيث تشمل المحتوى، أما الشكل فيكون في ملف الأنماط السي إس إس.
٢. جعل الصفحة قابلة للتعامل مع عدة متصفحات أو شاشات عرض مختلفة ويمكن معالجة كل جهاز (حاسوب أو حتى هاتف نقال) أو متصفح على حدة وبالتالي قابلية استخدام أكبر.
٣. يمكن وضع عدة مظاهر كل مظهر يلبي حاجة كل قارئ فمثلا يمكن ضمان قابلية الاستخدام لذوي الاحتياجات الخاصة أو وضع عدة ألوان لتلبية أذواق شتى.
٤. يمكن لملف الأنماط أن يتم تضمينه في عدة صفحات وبالتالي عند تغيير شكل العرض فإنك تحتاج لتعديل ملف واحد فقط وهذا يعني وقت أقل وتعديل أكبر وأشمل.

JavaScript:

جافا سكريبت بالإنجليزية: JavaScript واختصار لها JS

هي لغة برمجة عالية المستوى، تستخدم أساساً في متصفحات الويب لإنشاء صفحات أكثر تفاعلية تطورها حالياً لجنة TC39 التي تديرها منظمة EACMA للمعايير.

بدأ استخدام الجافا سكريبت كلغة برمجة موجهة للمبرمجين الهواة وغير المحترفين. إلا أنه تزايد الاهتمام بها وجذبت اهتمام مبرمجين محترفين بعد إضافتها لتقنيات جديدة كانتشار تقنية أجاكس التي أدت إلى سرعة في التفاعل بين الخادم والعميل.

تُستخدَم لغة الجافا سكريبت في تطوير صفحات ويب تفاعلية، وتطبيقات الويب، بما في ذلك الألعاب، وهي مُستعمَلة من أغلبية المواقع، وتدعمها جميع المتصفحات تقريباً دون الحاجة إلى إضافات خارجية.

تاريخ لغة JS:

في عام 1993، أصدر المركز الوطني لتطبيقات الحوسبة الفائقة (NCSA)، وهي وحدة تابعة لجامعة إلينوي في إربانا-شامبين، NCSA Mosaic، أول متصفح ويب رسومي شائع، والذي لعب دوراً مهماً في توسيع نمو الشبكة العالمية الناشئة الويب خارج مكانة NeXTSTEP حيث تشكل الشبكة العنكبوتية قبل ثلاث سنوات. في عام 1994، تأسست شركة تدعى Mosaic Communications في ماونتن فيو، كاليفورنيا ووظفت العديد من مؤلفي NCSA Mosaic الأصليين لإنشاء Mosaic Netscape ومع ذلك، فقد تعمدت عدم مشاركة رمز مع NCSA Mosaic الاسم الرمزي الداخلي لمتصفح الشركة هو Mozilla،

وهو رمز لـ "Mosaic and Godzilla" تم إصدار الإصدار الأول من مستعرض الويب، Mosaic Netscape 0.9، في أواخر عام 1994 خلال أربعة أشهر، استحوذت بالفعل على ثلاثة أرباع سوق المتصفح وأصبحت متصفح الويب الرئيسي للتسعينات. لتجنب مشاكل ملكية العلامة التجارية مع NCSA ، تمت إعادة تسمية المستعرض لاحقًا في Netscape Navigator في نفس العام، وحصلت الشركة على اسم Netscape Communications أدركت Netscape Communications أن الويب يحتاج إلى أن يصبح أكثر ديناميكية، يعتقد مارك أندريسن، مؤسس الشركة، أن HTML يحتاج إلى «لغة لاصقة» كان من السهل استخدامها من قبل مصممي الويب والمبرمجين غير المتفرغين لتجميع مكونات مثل الصور والإضافات، حيث يمكن كتابة الكود مباشرة في الويب ترميز الصفحة.

بنية لغة JS:

في الماضي كانت لغة JavaScript محدودةً إلى متصفحات الويب فيما قد سلف، لكن توسّع مجال JavaScript في الفترة الأخيرة لتضم إنشاء تطبيقات ويب من جهة الخادم (server-side) مع التعامل مع قواعد البيانات، إضافةً إلى إمكانية استخدامها في بيئات ليست متعلقة بالويب مثل معالجات النصوص وبرمجيات PDF ، ومن الممكن استعمالها أيضًا لكتابة تطبيقات سطح المكتب أو تطبيقات الهواتف، وحتى تطوير الألعاب!

هنالك الكثير من محركات JavaScript ، وكل واحدٍ منها يخضع لمواصفة (ECMAScript مع أنَّها تختلف فيما بينها بالإصدارات التي تدعمها).

يوجد في نهاية كل صفحة من صفحات هذا التوثيق قسمٌ بعنوان «دعم المتصفحات» الذي نذكر فيه ما هي المتصفحات التي تدعم الميزة المذكورة.

يتألف أساس لغة JavaScript من الميزات الشائعة للغات البرمجة، التي تسمح لنا:

- بتخزين القيم داخل متغيرات
- إجراء عمليات معالجة على السلاسل النصية
- تنفيذ إجراءات معيّنة اعتمادًا على وقوع الأحداث

ما يجب أن يثير اهتمامنا هو الواجهات البرمجية التي تُضيف وظائف على لغة JavaScript ، وتسمى هذه الواجهات البرمجية بالمصطلح APIs أي (Application Programming Interfaces) والتي تسمح للمطورين باستخدام لغة JavaScript للتعامل مع كثيرٍ من الأشياء، في مجال متصفحات الويب يمكننا استخدام JavaScript للتعامل مع شجرة DOM ، ولإستخدام الواجهة البرمجية للمواقع الجغرافية (Geolocation API)، ولإنشاء رسومات ثنائية وثلاثية الأبعاد عبر Canvas و Web GL، وللتحكم بالوسائط مثل الفيديو والصوت، وغير ذلك.

:PHP

بي إتش بي بالإنجليزية: Hypertext Preprocessor «المعالج المسبق للنصوص الفائقة»

كانت مجموعة من التطبيقات التي كتبت باستخدام لغة بيرل أطلق راسموس اسم Personal Home Page Tools الصفحة الرئيسية الشخصية هي لغة برمجة نصية صممت أساساً من أجل استخدامها لتطوير وبرمجة تطبيقات الويب، كما يمكن استخدامها لإنتاج برامج قائمة بذاتها وليس لها علاقة بالويب فقط بي إتش بي لغة مفتوحة المصدر ويطورها فريق من المتطوعين تحت رخصة بي إتش بي، تدعم البرمجة كائنية التوجه وتركيبها البنوي يشبه كثيراً التركيب البنوي للغة السي، هذا بالإضافة إلى أنها تعمل على أنظمة تشغيل متعددة مثل لينكس وويندوز.

تاريخ PHP:

١. PHP/FI:

ظهرت PHP أولاً في سنة 1995 على يد راسموس ليردورف كانت تسمى وقتها ب PHP/FI وفي الحقيقة لم تكن لغة برمجة وقتها وإنما كانت مجموعة من التطبيقات التي كتبت باستخدام لغة بيرل أطلق راسموس اسم Personal Home Page Tools على هذه التطبيقات، لأنه احتاج فيما بعد إلى تطبيقات أكثر فائدة قام راسموس بكتابة تطبيق أكبر باستخدام لغة سي حيث أصبحت قادرة على الاتصال بقواعد البيانات كما أنها كانت تسمح للمستخدمين بتطوير تطبيقات مواقع ديناميكية بسيطة، اختار راسموس أن تكون الشيفرة المصدرية الخاصة ب PHP/FI متوفرة للجميع لذا كان يمكن لأي شخص أن يستخدمها ويقوم بتحسينها والمشاركة في حل أخطاءها ومشاكلها.

كانت PHP/FI وقتها تحوي على بعض الوظائف المتوفرة بالإصدارات الحالية من اللغة، كما أن المتغيرات كانت تشبه متغيرات بيرل، وكانت تركيبها النحوي يشبه بيرل بالرغم من بساطتها ومحدوديتها. في عام 1997 تم إطلاق الإصدار 2.0 من PHP/FI ، حيث بلغ عدد مستخدميها آنذاك 50000 نطاق، وكان هناك مجموعة من الأشخاص الذين يشاركون في التطوير، وتم إطلاق الإصدار الرسمي من 2.0 في شهر نوفمبر من نفس العام بعد العديد من الإصدارات التجريبية بيتا.

٢. PHP (3.0):

في عام 1997 تم إعادة كتابة PHP/FI على يد زيف سوراسكي، أندري جتمانز بعدما وجدوا أن PHP/FI 2.0 ليست قوية بما فيه الكفاية من أجل كتابة تطبيق تجارة إلكترونية والذي كانوا يعملون عليه كمشروع تخرج لجامعتهم، كان هناك تعاون بينهم وبين مؤسس اللغة راسموس ليردورف على أن تكون PHP 3.0 هي النسخة الرسمية بعد PHP/FI.

أحد أهم الميزات التي تميزت بها بي إتش بي 3 عن سابقتها إنها أصبحت قابلة للتوسع وتوفر مع هذا الإصدار العديد من المكتبات والدوال، وأدت قابلية التوسع إلى إقبال العديد من المطورين على تطوير المكتبات الجديدة وإضافتها مع اللغة، ويقال أن هذا هو السبب الأساسي للنجاح الذي حققه هذا الإصدار، ومن الميزات الأخرى التي تمت إضافتها في هذا الإصدار هي البرمجة كائنية التوجه.

وفي هذا الإصدار تم تغيير اسم اللغة ووضعها تحت اسم جديد وهو بي إتش بي (بدون FI) والذي كان يحمل اختصارا لمعنى جديد مختلف عن الإصدار السابق والمعنى هو

"PHP: Hypertext Preprocessor".

تم إطلاق بي إتش بي 3.0 في يونيو 1998 بعد 9 أشهر من الاختبارات.

٣. (4.0): PHP

في عام 1998 ، وبعد الانطلاق الرسمي لبي إتش بي 3.0 بقليل بدأ زيف سوراسكي وأندي جتمانز بإعادة كتابة أساس لغة بي إتش بي وكان الهدف من ذلك هو تحسين الأداء للبرامج المعقدة والضخمة وتحسين قابلية اللغة للتوسع.

المحرك الجديد الذي ظهر بعد إعادة الكتاب تم تسميته بمحرك زند واسم «زند» مأخوذ من أوائل حروف أسماء مطوريه، ونجحوا في تحقيق اهدافهم عن طريق هذا المحرك، وتم الإعلان عنه في عام 1999 .

تعتمد بي إتش بي 4.0 على هذا المحرك وتم الإعلان عن هذا الإصدار من بي إتش بي في مايو 2000، بالإضافة إلى تحسين الأداء في هذا الإصدار احتوى كذلك على مجموعة جديدة من الميزات مثل دعمه لعدد أكبر من خوادم الويب، الجلسات، طرق آمنة جديدة لمعالجة دخل المستخدم وغيرها.

وقد تم إعلان إيقاف تطوير ودعم بي إتش بي 4 في: 13 يوليو 2007.

سبب الاختيار:

- تعتبر لغة C# من اللغات المستخدمة في مجال الذكاء الاصطناعي تتعامل مع الذاكرة بشكل سلس ولا يوجد بها مشاكل تستخدم لغة السي شارب على جميع المنصات مثل منصة أندرويد وأنظمة التشغيل.
- SQL تتيح امكانية إضافة ملفات ومراجعتها، وتعديلها، وتحديث البيانات بها ومراجعتها · تخزين البيانات وتعجيلها بالإضافة والحذف والتعديل
- باستخدام HTML، ستمكن من إنشاء هيكل موقع الويب الخاص بك.
- يمنحك CSS القدرة على جعل موقع الويب يبدو أكثر جاذبية من الناحية المرئية.
- بالنسبة إلى JavaScript، فهذه لغة برمجة قوية تسمح لك بتغيير مكونات HTML و CSS لموقع الويب الخاص بك بشكل فعال لتناسب مع مواصفاتك بدقة.
- PHP لغة بسيطة وسهلة ولا تأخذ وقتاً طويلاً في تعلمها وفهماً مقارنةً باللغات الأخرى مثل ASP و SP لوغيرها.

العيوب:

- C# تعد من لغات البرمجة الأقل مرونة، فهي تعتمد على إطار عمل Microsoft .Net واجهة المستخدم الرسومية الخاصة بها تعتبر ضعيفة.
- في SQL التحكم الجزئي، تبعاً لقواعد العمل المخفية؛ فإنه لا يمتلك المبرمجين السيطرة الكاملة على قاعدة البيانات.
- HTML لا يمكن أن تنتج مخرجات ديناميكية وحدها، لأنها لغة ثابتة، يكون من الصعب فهم بنية مستندات HTML.
- تقتصر CSS على العمل بشكل أساسي فقط مع ما هو موجود في ملف الترميز Markup ، ويمكن تغيير ترتيب العرض إلى حد ما، كما يمكن إدراج قدر صغير من محتوى النص ولكن لإحداث تغييرات كبيرة في مصدر HTML أو XML تحتاج إلى استخدام طريقة أخرى مثل XSL وتحولات XSLT.
- JS مشكلات الأمان، بما أن معظم التعليمات البرمجية يتم تنفيذها على الحاسب الخاص بالمستخدم، فيمكن استغلالها في بعض الأحيان لأغراض ضارة،
- PHP غير مناسبة لتطبيقات الويب العملاقة القائمة على المحتوى، كما أنها غير قادرة على التعامل مع عدد كبير من التطبيقات.

الواجهات:

واجهة تسجيل الدخول للتطبيق

لوحة التحكم الخاصة بالمدير



واجهة إضافة موظف

الموظفين

الاسم

الجنس

اسم المستخدم

الراتب

تاريخ العمل

العنايت

الحجم

كلمة المرور

نوع الوظيفة

البريد الإلكتروني

إضافة

Mohamed ibrahem

لوحة التحكم

حسابات الموظفين

حسابات الزبائن

تسجيل الخروج

الإعدادات

واجهة تخصيص

اللغة العربية
لغة التطبيق

فاتح
داش
مظهر التعليل

أسعار الفوركس

مرددة	مردودة
\$	٥٩
\$	٣٠

أسعار الفوركس

مرددة	مردودة
\$	٦٩
\$	١٢

تسجيل الخروج
الإعدادات

واجهة الخصوصية للمدير

الاسم

رقم الهاتف

العمر

البريد الإلكتروني

الخصومة



Mohamed ibrahem

- لوحة التحكم
- حسابات الموفلين
- حسابات الربانين



تسجيل الخروج

الإعدادات

كلمة المرور القديمة

اسم المستخدم القديم

كلمة المرور الجديدة

كلمة المرور الجديدة

اسم المستخدم الجديد

إعادة كتابة كلمة المرور

التالي
لغو كلمة المرور

واجهة إدخال بيانات النزيل

<input type="text"/>	الاسم الأول	<input type="text"/>	الاسم الأخير
<input type="text"/>	الجنسية	<input type="text"/>	العمر
<input type="text"/>	رقم الهوية	<input type="text"/>	رقم الهاتف
<input type="text"/>	تاريخ الدخول	<input type="text"/>	تاريخ الخروج
<input type="button" value="التالي"/>		<input type="button" value="الغاء التسجيل"/>	

واجهة إضافة نزيل

نوع الغرفة : ☒ مفردة ☐ مزدوجة ☐ ثلاثية ☐ سويت

رقم الغرفة :

عدد الافراد :

السعر الإجمالي :

واجهة الغرف

قائمة الغرف					
حالة الغرفة	سعر الغرفة	نوع الغرفة	رقم الغرفة	الرقم الافتراضي	
محمّولة	٩٩	سويت	١	١	
محمّولة	٧٩	ثلاثية	٢	٢	
محمّولة	٥٩	مزدوجة	٣	٣	
محمّولة	٣٩	مفردة	٤	٤	
فارغة	٣٩	مفردة	٥	٢١	

واجهة إضافة غرفة



مفردة

٢

ثلاثية

١

رقم الغرفة

عدد الغرف

٥

ثلاثية

١

سويت

١

نوع الغرفة

▼

سعر الغرفة

الغاء

إضافة

الخصوصية

الغرف

تخصيص

حول

Home

Users

Home

Key

Hand

Settings

المراجع :

- <https://github.com/>
- <https://www.noor-book.com/>
- <https://www.youtube.com/@ElzeroWebSchool/videos>
- https://www.youtube.com/channel/UCjKpb_hadZ3BQkHcTy81MRA
- <https://www.youtube.com/c/RJCodeAdvanceEN>
- <https://www.youtube.com/c/RJCodeAdvance>
- <https://www.programmer-tech.com/>
- <https://www.w3schools.com/>
- <https://wiki.hsoub.com/>



The End