Front-end & Back-end Structure

Name: Noora Najeeb Al-tamimi

❖ ماذا نقصد بال Back end and Front end:

1) Front end:

ال Front end هو تصميم الواجهات التي تظهر للمستخدم عند زيارته لموقع أو تطبيق ويب وما تحويه من أزرار وصور وتصوص وغيرها وتحديد كيف سيبدو كل جزء من موقع الويب وكيف سيتفاعل المستخدم معه، تُستخدم لغات معينة لتصميم هذي الواجهات وهي:

- لغة HTML: هي اختصار لعبارة HyperText Markup Languag وهي لغة ترميزية مسؤولة عن بناء الهيكل العام للصفحة وما فيها من عناصر باستخدام الوسوم وهي علامات موضوعة ضمن قوسي زاوية <> تحدد نوع المحتوى أو هيكليته.
- لغة CSS: هي اختصار ل Cascading Style Sheets وهي اللغة المسؤولة عن تنسيق عناصر الصفحة كأختيار الألوان والخطوط والخلفيات وما إلى ذلك كما أنها تستخدم في جعل مواقع الويب متجاوبة مع مختلف الشاشات من خلال كتابة شيفرات تغير تنسيق وتخطيط العناصر بحسب أبعاد الشاشة.
- لغة Javascript: وهي اللغة المسؤولة عن التفاعلات والحركة داخل الصفحة، تعد واحدة من أشهر لغات البرمجة وأكثرها استخدامًا فهي تمكنك من تطوير كل من الواجهة الأمامية والخلفية.

2) Back end:

تطوير الواجهة الخلفية للويب أوما يعرف بتطوير النظم الخلفية backend development أو التطوير من طرف الخادم server-side هو عملية كتابة الشيفرات البرمجية التي تعمل على خوادم الويب web servers والتي تجعل موقع الويب يعمل خلف الكواليس ويتفاعل مع المستخدمين ويلبي كافة طلباتهم، لا غنى عن تطوير الواجهات الخلفية في مواقع الويب الديناميكية التي تتفاعل مع المستخدمين وتحتاج إلى تخزين البيانات وفهرستها وحفظها واستردادها وتعديلها وحذفه مثل مواقع التجارة الإلكترونية والمواقع التي تتطلب ملء نماذج وتخزين البيانات المتغيرة مثل وصف المنتجات والمنشورات وملفات تعريف المستخدمين وما إلى ذلك، أما مواقع الويب الثابتة أو الساكنة فهي لا تتطلب تطوير الواجهة الخلفية وتكون

مجرد مواقع بسيطة بمحتوى ثابت ولا تتفاعل مع المستخدمين ولا تستطيع تخزين أي شيء وتسعمل خوادم شبه جاهزة لتخديم البيانات.ومن أشهر اللغات المستخدمة في عملية تطوير الواجهة الخلفية:

- JavaScript (Node.js): هي نفس لغة الواجهة الأمامية ولكن تستخدم Node.js تعمل على السيرفرات وتعد مشهورة جداً في تطبيقات الويب الحديثة.
- لغة python: لغة بسيطة وسهلة مناسبة للمبتدئين ، أشهر إطارات العمل التي تستخدمها: Django، Flask، FastAPI ، تُستخدم في مواقع الويب، الذكاء الاصطناعي ، تحليل البيانات.
- Java: لغة قوية ومستقرة تستخدم غالباً في المشاريع الكبيرة.

❖ هيكلية عمل الواجهة الأمامية مع الواجعة الخلفية (Backend & Frontend structure):

كما قلنا سابقاً أن المستخدم لا يظهر له شي من الواجهة الخلفية وإنما ما يظهر للمستخدم وما يتفاعل معه هو الواجهة الأمامية وهذه لواجهة الأمامية ماهي الا عبارة عن وسيط او مستقبل للبيانات والمدخلات المختلفة من المستخدم ومرسل لهذه البيانات للواجهة الخلفية ولكن هذا الارسال او الطلب يتم تحليله وارجاع النتائج عن طريق عمليات واليات يمكن توضيحها بمثال :

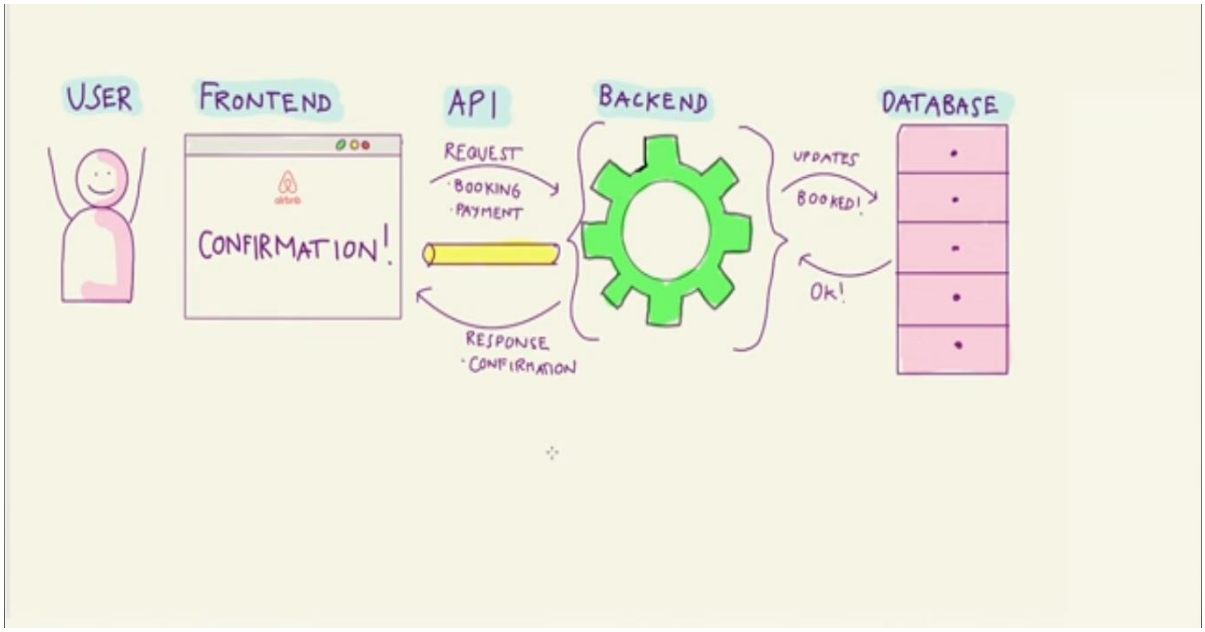
لنفترض أن المستخدم في صفحة تسجيل الدخول ووجد حقل لإدخال الاسم الخاص به فأدخل المستخدم اسمه بعد ذلك تأخذ الواجهة هذه البيانات المدخلة وتقدم طلب للواجهة الخلفية عن طريق ما يُعرف بال API اختصار Application programming Interface وهو عبارة عن وسيط ينقل الطلبات والبيانات بين ال frontend وال backend او بين نظامين مختلفين، بعدها تستقبل الواجهة الخلفية الطلب والبيانات من الواجهة الأمامية ثم تعمل استعلام(Query) على قاعدة البيانات التي يتم تخزين جميع البيانات المطلوبة لاستمرارية عمل التطبيق ، بعد ذلك ترجع قاعدة البيانات البيانات المطلوبة التي استعلمت عنها الواجهة الخلفية وبدور الواجهة الخلفية تعيد ارسال الرد او الاستجابة للواجهة الأمامية عبر ال API بالتأكيد كونها الوسيط بينهما ولكن الواجهة الخلفية لا ترجع الرد مباشرة بل تطبق عليه بعض العمليات المنطقية أو تحليل لهذه البيانات بناءاً على ما صممه المطور من خوارزميات وشروط خاصة بعدها كما قلنا تعود النتيجة مرة أخرى للواجهة الأمامية وفي هذا المثال اذا كان اسم المستخدم مخزن في قاعدة البيانات فانها ستنعرض له رسالة نجاح وان لم يكن فتعرض له رسالة خطأ ولن يستطيع الولوج للصفحة التالية للموقع.

كما يمكن للواجهة الخلفية التعامل مع مصدر خارجي للبيانات Server او موقع خارجي عن طريق API خاص مثل تطبيق معرفة حالة الطقس.

أما عن سبب عدم ارتباط ال frontend بقاعدة البيانات مباشرة تأتي بعض الأسباب:

1- الأمان: لو الواجهة اتصلت مباشرة بقاعدة البيانات هذا يعني انه أي شخص ممكن يسرق هذه البيانات أو يعدل عليها.

2- التحقق من المنطق: الBackend يتحقق من صحة البيانات ، يحدد من يملك الصلاحية، يطبق القوانين.



(هذا الشكل شرح مختصر لما سبق)