اعداد: م. محمد أسامة بيطار

دورة تطوير الويب

الجلسة الأولى

المحتويات

[ما هو الويب؟ 2](#_Toc159778554)

[كيف تعمل صفحات الويب؟ 2](#_Toc159778555)

[عملية تطوير الويب: 3](#_Toc159778556)

[كيف نحصل على رباط بين الجوانب السابقة؟ 4](#_Toc159778557)

[بعض الأسئلة المهمة: 5](#_Toc159778558)

[هل يجب تعلم كل الجوانب السابقة لكي أستطيع ان أقوم بتطوير موقع الكتروني؟ 5](#_Toc159778559)

[ماهي المهارات الأساسية التي يجب تعلمها؟ 5](#_Toc159778560)

[هل يمكن ان أصبح خبيرا في كل الجوانب الخاصة بعملية تطوير الويب؟ 6](#_Toc159778561)

[من أين يجب أن أبدأ في تعلم تطوير الويب؟ 6](#_Toc159778562)

[المسار المقترح لهذا الكورس: 7](#_Toc159778563)

[تقسم الجلسات الى: 7](#_Toc159778564)

[المهارات التي سوف يحصل عليها الطالب مع انتهاء الدورة: 8](#_Toc159778565)

[انشاء البيئة المناسبة للتطوير: 9](#_Toc159778566)

[أولا: تحميل برنامج VS Code 9](#_Toc159778567)

[ثانيا: بعض الإضافات التي تسهل العمل: 9](#_Toc159778568)

[ثالثا: تحميل Node JS 9](#_Toc159778569)

# ما هو الويب؟

الويب هو نظام عالمي للمعلومات المتصلة عبر شبكة الإنترنت. يعتبر الويب بمثابة مجموعة من الصفحات والمواقع والتطبيقات التي يمكن الوصول إليها عبر المتصفحات الإلكترونية.

# كيف تعمل صفحات الويب؟

صفحات الويب تعمل بناءً على نموذج يسمى نموذج العميل-الخادم (Client-Server Model). وهذا يعني أن هناك اتصال وتفاعل بين متصفح الويب الذي يعمل على جهاز العميل (مثل الكمبيوتر أو الهاتف الذكي) والخادم الذي يحتوي على الموقع الويب.

عندما يقوم المستخدم بفتح صفحة ويب في متصفحه، يتم إرسال طلب من المتصفح إلى الخادم للحصول على المحتوى المطلوب (مثل صفحة HTML). يستخدم المتصفح بروتوكول ( Hypertext Transfer Protocol : HTTP) لإرسال الطلب إلى الخادم.

عند استلام الخادم للطلب، يبحث عن الملف المطلوب ويقوم بإرساله إلى المتصفح كجزء من الاستجابة. يستخدم الخادم لغات الويب مثل HTML وCSS وJavaScript لبناء صفحات الويب وتنسيقها وإضافة وظائف تفاعلية.

عندما يتلقى المتصفح الملفات، يقوم بتفسيرها وعرض المحتوى للمستخدم. يترجم المتصفح العناصر في صفحة HTML إلى عرض مرئي يمكن للمستخدم رؤيته والتفاعل معه، مع تطبيق التنسيقات المحددة في CSS. كما يستخدم المتصفح لغة JavaScript لتنفيذ البرامج النصية التفاعلية والتحكم في سلوك الصفحة.

بالإضافة إلى ذلك، قد يتم استرداد بيانات إضافية من الخادم بواسطة المتصفح بواسطة طلبات إضافية، مثل الصور أو مقاطع الفيديو أو البيانات المرتبطة.

# عملية تطوير الويب:

تطوير الويب يشير إلى عملية بناء وتطوير مواقع الويب وتطبيقاتها. يشمل هذا العملية العديد من الأنشطة والمهارات التقنية التي تهدف إلى إنشاء وتحسين وصيانة المواقع الويب. فيما يلي نظرة عامة على بعض جوانب تطوير الويب:

1. تصميم الواجهة الأمامية (Front-end Development): يتعلق بتطوير جزء المستخدم الظاهري للموقع الذي يتفاعل المستخدم معه مباشرة. يشمل ذلك استخدام لغات برمجة مثل HTML وCSS وJavaScript لبناء صفحات الويب وتنسيقها وتوفير تجربة مستخدم مريحة وجذابة.

2. تطوير الجانب الخلفي (Back-end Development): يتعلق ببناء الأجزاء التي تدير عمليات الموقع وتتعامل مع قواعد البيانات والمنطق البرمجي. يشمل ذلك استخدام لغات برمجة مثل PHP، Python، Ruby، Java أو .NET وإطارات العمل المرتبطة بها لتطوير وتشغيل الخوادم والمنطق الخلفي.

3. قواعد البيانات (Database): يشمل تصميم وإنشاء وإدارة قواعد البيانات التي تخزن وتنظم المعلومات المتعلقة بالموقع، مثل المستخدمين والمحتوى والتفاصيل الأخرى. يمكن استخدام نظم إدارة قواعد البيانات مثل MySQL، PostgreSQL، MongoDB وغيرها.

4. تجربة المستخدم (User Experience - UX): يتعلق بتحسين تفاعل المستخدم مع الموقع أو التطبيق، بحيث يكون التصميم سهل الاستخدام ومريح للمستخدمين. يشمل ذلك تحليل استجابة المستخدمين وتصميم واجهات مستخدم فعالة وعملية.

5. الأمان والأداء: يشمل تطوير الويب أيضًا الاهتمام بأمان الموقع وحماية البيانات الحساسة، بالإضافة إلى تحسين أداء الموقع وسرعته في التحميل والاستجابة.

ملاحظة: يجب أن نشير إلى أن كل جانب من الجوانب المذكورة في الفقرة السابقة يشكل بحد ذاته عالمًا متعلقًا بالعلوم المختلفة. على سبيل المثال، تطوير الواجهة الأمامية يتطلب معرفة عميقة بتقنيات HTML وCSS وJavaScript، بينما تطوير الجانب الخلفي يستدعي فهماً متقدماً للغات البرمجة وإطارات العمل. علاوة على ذلك، تصميم قواعد البيانات يتضمن معرفة بنماذج البيانات والاستعلامات، وتحسين تجربة المستخدم يستلزم دراسة عميقة في علم النفس وتصميم الواجهات.

لذا، يمكن اعتبار تطوير الويب بأكملها كبحر من العلوم المترابطة، حيث يجب على المطورين أن يكتسبوا معرفة متعددة الجوانب للعمل بكفاءة في هذا المجال. لذلك، قد يحتاج المطورون إلى الاستفادة من مختلف المصادر التعليمية والتدريبية المتاحة لتعلم وتطوير المهارات المطلوبة في كل جانب من جوانب تطوير الويب.

# كيف نحصل على رباط بين الجوانب السابقة؟

عندما يتعلق الأمر بترابط عملية برمجة الواجهة الأمامية (Front-end) والجانب الخلفي (Backend) في تطوير الويب، هناك بعض الأساليب التي يمكن اتباعها للتحقيق بنجاح تلك الربط. إليك خطوات عامة لتوضيح العملية:

1. تحديد وتوثيق واجهة البرمجة (Application Programming Interface: API): الهدف منها هو عملية تقوم بالربط بين الواجهة الأمامية front-end والجانب الخلفي back-end. يجب تحديد النقاط النهائية (Endpoints) والبيانات المتوقعة للتبادل بين الجانبين.

2. بناء الواجهة الأمامية: يبدأ الفريق في بناء الواجهة الأمامية باستخدام لغات الويب مثل HTML وCSS وJavaScript. يتم استخدام هذه اللغات لتصميم وتنسيق صفحات الويب وتحقيق وظائف تفاعلية. في هذه المرحلة، الAPI من الج الاتصال بالنقاط النهاية المحددة والحصول على البيانات المطلوبة من الخادم.

3. تطوير الجانب الخلفي: يتم بناء الجانب الخلفي من التطبيق باستخدام لغات البرمجة وإطارات العمل المناسبة. يتم استخدام وثائق واجهة البرمجة لتعريف نقاط النهاية وتطبيق الوظائف المطلوبة للتفاعل مع قواعد البيانات وتنفيذ العمليات المنطقية.

4. الاتصال بين الجانبين: بمجرد الانتهاء من بناء الواجهة الأمامية والجانب الخلفي، يتم استخدام طرق الاتصال المتاحة مثل استدعاءات API (API calls) أو AJAX للتواصل بين الجانبين.

5. اختبار وتحسين: يتم إجراء اختبار شامل للتأكد من أن الواجهة الأمامية والجانب الخلفي يعملان بشكل صحيح ويتواصلان بشكل صحيح. يتم تحسين الأداء وإصلاح أي أخطاء محتملة في هذه المرحلة.

6. النشر والصيانة: بعد اجتياز مرحلة الاختبار والتحسين، يتم نشر التطبيق بشكل عام على الخوادم المناسبة. وبعد النشر، يتم تتبع وصيانة التطبيق لضمان استمراريته وتحسين أدائه وحل المشكلات التي تنشأ.

# بعض الأسئلة المهمة:

## هل يجب تعلم كل الجوانب السابقة لكي أستطيع ان أقوم بتطوير موقع الكتروني؟

لا، لا يجب عليك تعلم كل الجوانب السابقة لتطوير موقع إلكتروني.

## ماهي المهارات الأساسية التي يجب تعلمها؟

لتطوير موقع إلكتروني، قد تحتاج إلى مهارات برمجية وتقنية أساسية مثل:

1. لغات الويب الاساسية: مثل HTML وCSS لتصميم وتنسيق الصفحات، وJavaScript لإضافة الوظائف التفاعلية والديناميكية.

2. قواعد البيانات: مثل SQL للتعامل مع قواعد البيانات، وتخزين واسترجاع البيانات المرتبطة بالموقع.

3. تطوير الواجهة الأمامية (Front-end Development): يتطلب فهمًا لتصميم المستخدم وتجربة المستخدم، واستخدام أدوات وإطارات عمل مثل React.js أو Angular.js او Vue.js

4. تطوير الواجهة الخلفية (Back-end Development): يتطلب إتقان لغات البرمجة مثل Python أو PHP أو Ruby أو Node.js، وإطارات العمل مثل Django أو Laravel أو Ruby on Rails.

5. أمان الموقع وحماية البيانات: يجب أن تكون على دراية بمفاهيم أمان الويب وحماية البيانات وتنفيذ إجراءات أمان مناسبة.

## هل يمكن ان أصبح خبيرا في كل الجوانب الخاصة بعملية تطوير الويب؟

يمكن أن تصبح خبيرًا في مجالات مختلفة من عملية تطوير الويب، ولكن من الصعب أن تصبح خبيرًا في كل الجوانب الخاصة بها. عملية تطوير الويب تشمل نطاقًا واسعًا من المهارات والتقنيات، وكل مجال يتطلب تعلم وتجربة مستقلة.

## من أين يجب أن أبدأ في تعلم تطوير الويب؟

بعض الخطوات التي يمكنك اتخاذها:

1. تعلم اللغات الأساسية: يمكنك البدء بتعلم لغات الويب الأساسية مثل HTML تستخدم لبناء هيكل وتنسيق المحتوى، CSS تستخدم لتصميم وتنسيق الواجهة الأمامية، وJavaScript يتيح لك إضافة التفاعل والوظائف الديناميكية لموقعك.

2. تجربة الأدوات والإطارات: قم بتجربة الأدوات والإطارات الشائعة في تطوير الويب مثل Bootstrap وReact وAngular وVue.js. هذه الأدوات تساعدك في بناء وتطوير المشاريع بشكل أكثر فعالية وسرعة.

3. تعلم الواجهة الخلفية: بمجرد أن تتقن الواجهة الأمامية، يمكنك التعمق في تطوير الواجهة الخلفية باستخدام لغات البرمجة مثل Python أو Node.js أو PHP. هذه اللغات تستخدم لبناء الخوادم والتفاعل مع قواعد البيانات.

4. بناء مشاريع تطبيقية: قم ببناء مشاريع تطبيقية صغيرة لتطبيق المفاهيم والمهارات التي تعلمتها. يمكنك بدء المشاريع البسيطة مثل موقع شخصي أو مدونة ومن ثم التحرك إلى مشاريع أكثر تعقيدًا تتطلب تفاعلات أكثر.

5.من اهم الخطوات هي استكشاف المصادر التعليمية عبر الإنترنت: هناك العديد من المصادر التعليمية المجانية والمدفوعة عبر الإنترنت التي تقدم دروسًا ومواد تعليمية حول تطوير الويب. يمكنك الاستفادة من الدروس المباشرة والفيديوهات والمقالات والمنتديات النقاشية.

6.الخطوة الأهم هي الممارسة والتطبيق: لا تنسى أهمية الممارسة والتطبيق العملي. قم ببناء مشاريع وحاول حل التحديات التي تواجهك. كلما قمت بمزيد من التطبيق العملي، ستتعلم وتتقن المزيد.

تذكر أن تعلم تطوير الويب يستغرق الوقت والجهد، فالممارسة المنتظمة والاستمرارية في التعلم هي المفتاح لتحقيق التقدم وتطوير مهاراتك.

# المسار المقترح لهذا الكورس:

## تقسم الجلسات الى:

**جلسة 1: مقدمة**

- مفهوم تطوير الويب وأهميته.

- لمحة سريعة عن محتويات الدورة.

**جلسة 2-4: HTML**

- مفاهيم أساسية في HTML

- تنسيق النصوص والعناوين

- إضافة الصور والروابط

- الجداول والنماذج

**جلسة 5-8: CSS**

- أساسيات CSS وتنسيق العناصر

- التحكم في الألوان والخطوط

- تخطيط الصفحات والعناصر المرنة

**جلسة 9-12 (16 ساعة): JavaScript**

- أساسيات البرمجة بلغة JavaScript

- المتغيرات والعمليات الحسابية

- الشروط والحلقات

- إنشاء واستخدام الدوال

**جلسة 13-16 (16 ساعة): Node.js**

- مقدمة في Node.js ومفهوم السيرفر

- إنشاء سيرفر بسيط باستخدام Node.js و Express.js

- التعامل مع قواعد البيانات باستخدام MongoDB

- إنشاء API والتفاعل معها باستخدام Node.js

**جلسة 17-20 (16 ساعة): المشاريع**

- تطوير مشروع تطبيقي كامل باستخدام HTML وCSS وJavaScript وNode.js.

## المهارات التي سوف يحصل عليها الطالب مع انتهاء الدورة:

عند انتهاء الدورة واستكمال المسار الذي تم ذكره، سيحصل الطلاب على مجموعة من المهارات في تطوير الويب. ستشمل المهارات التي يكتسبها الطلاب ما يلي:

1. إنشاء صفحات ويب قائمة على HTML: ستكون لدى الطلاب القدرة على إنشاء صفحات ويب بناءً على لغة HTML، والتي تمكنهم من هيكلة المحتوى وإضافة الروابط والصور والجداول والنماذج.

2. تنسيق وتصميم المواقع باستخدام CSS: سيتعلم الطلاب كيفية استخدام لغة CSS لتنسيق وتصميم المواقع الإلكترونية، بما في ذلك تحكمهم في الألوان والخطوط والتخطيطات والانتقالات بالإضافة الى تعلم التعامل مع بعض الأدوات مثل: Bootstrap أو Tailwind.

3. إضافة التفاعل والديناميكية باستخدام JavaScript: سيكتسب الطلاب مهارات في استخدام لغة JavaScript لإضافة التفاعل والوظائف الديناميكية إلى صفحات الويب، مثل التحقق من صحة البيانات المدخلة وتغيير المحتوى بناءً على إجراءات المستخدم.

4. بناء الخوادم والتفاعل مع قواعد البيانات باستخدام Node.js: سيتعلم الطلاب كيفية استخدام Node.js لبناء الخوادم وإنشاء API والتفاعل مع قواعد البيانات، مما يمكنهم من إنشاء تطبيقات ويب قابلة للتوسع والتفاعلية.

5. تطوير مشروع موقع لإدارة المكتبات: سيعمل الطلاب على مشروع موقع لإدارة المكتبات كجزء من الدورة، وهذا سيسمح لهم بتطبيق المفاهيم والمهارات التي تعلموها في المحاضرات السابقة وتحويلها إلى تطبيق عملي قابل للعرض.

باستكمال هذا المسار والمشروع، ستكون لدى الطلاب قاعدة قوية في تطوير الويب وإدارة المشاريع، ويمكنهم استخدام هذه المهارات لبناء مواقع وتطبيقات ويب متقدمة واحترافية.

# انشاء البيئة المناسبة للتطوير:

## أولا: تحميل برنامج VS Code

<https://code.visualstudio.com/>

## ثانيا: بعض الإضافات التي تسهل العمل:

Live server

Auto Close Tag

Prettier - Code formatter

Comment Divider

Bootstrap IntelliSense

node-snippets

## ثالثا: تحميل Node JS

<https://nodejs.org/en>

# Hypertext Markup Language

HTML هي اختصار لـ HyperText Markup Language، وهي لغة توصيف تستخدم لبناء وتنسيق صفحات الويب. تعتبر HTML لغة أساسية في تطوير الويب، حيث يتم استخدامها لتحديد بنية وتنظيم المحتوى على صفحة الويب وتعريف العناصر والعناوين والفقرات والصور والروابط والجداول وغيرها من العناصر التفاعلية. تستخدم HTML عادة بالاشتراك مع لغات أخرى مثل CSS و JavaScript لإضفاء الشكل والتفاعل على صفحات الويب. تستخدم العناصر في HTML علامات معينة تحيط بالنص لتحديد نوع العنصر وتنسيقه وترتيبه في الصفحة.