



الجمهورية العربية السورية
وزارة التعليم العالي
جامعة البعث - كلية الهندسة المعلوماتية
قسم البرمجيات

مقالة بين تقنيات الويب الحديثة (AngularJS Vs ReactJS)
مشروع السنة الرابعة

إعداد الطلاب

بهار حسن معرف العبدالله علي العباس وفاء الشيخ

إشراف
د. محسن حسين

العام الجامعي 2018-2019

الفهرس

7	الفصل الأول : مقدمة
7	1-1-مقدمة
7	2-1- تقنية HTML5
8	1-2-1- دعم المتصفحات للإصدارات الجديدة
8	2-2-1- الخصائص الجديدة
9	3-2-1- الصيغ العامة لغة
9	4-2-1- من الوسوم الجديدة في html5
10	5-2-1- عناصر الوسائط المتعددة الجديدة
10	6-2-1- العناصر المحذوفة من html5
11	7-2-1- لماذا تعد الأفضل ؟
11	(CSS) cascading style sheet-3-1
11	1-3-1- مشكلة مطوري الويب قبل css
12	(Css3)-2-3-1
12	3-3-1- ميزات CSS3
14	4-1- تقنية الـ Node js
16	5-1- Bootstrap
18	6-1- Typescript
19	7-1- Gulp.js
21	الفصل الثاني : تقنيات جافا سكريبت
21	1-2- مقدمة
21	2-2- مكتبات وأطر الجافا سكريبت
22	1-2-2- Meteorjs
22	2-2-2- Jquery

23	Vue.js-3-2-2
23	Backbone.js-4-2-2
23	Breeze.js-5-2-2
24	AngularJS -6-2-2
25	AngularJS(2.1) -7-2-2
25	ReactJS-8-2-2
25	React -1-8-2-2 ميزات
26	React -2-8-2-2 قيود
26	2-3- نموذج كائنات الوثيقة DOM
27	2-1-3-2 ما هو مفهوم الـDOM ؟
27	1-1-3-2 ما هو مفهوم الـHTML DOM ؟
28	2-3- الفرق بين استخدام VirtualDOM وRealDOM
28	4-2- (MVC)Model-View-Controller
29	2-1- المكونات (Component)
29	2-4-2- MVC Use in Web Applications
30	2-3-4-2- آلية عمل الـMVC
30	2-4-4-2- محسن ومساوئ نموذج MVC
30	1-4-4-2- (MVC) Advantages MVC
30	2-4-4-2- (MVC) Disadvantages MVC
31	5-2- Single Page Application
32	5-2-1- ما هي تقنية SPA ؟
33	2-5- بني الخادم المستخدمة في SPA ؟
33	1-2-5-2- (بنية الخادم الخفيفة) Thin server architecture
33	2-2-5-2- (بنية الخادم السميكة) Thick stateful server architecture
34	3-2-5-2- (بنية الخادم الأقل سماكة) Thick stateless server architecture

35	الفصل الثالث: تفزيذ تطبيقات على Angular و React
35	1-3- تنفيذ معرض صور على React
35	1-1-3- تنصيب بيئة Node.js
35	1-2-3- تنصيب مكونات React.js على Node.js
35	1-3- إنشاء مشروع React جديد
35	1-4- برمجة معرض الصور
38	2-3- تنفيذ معرض صور على Angular
38	2-1-3- الخطوة الأولى (تصميم ملف HTML)
40	2-2-3- الخطوة الثانية (تصميم ملف CSS)
42	2-3-3- الخطوة الثالثة (تصميم ملف JavaScript)
44	3- المقارنة بين ReactJS و AngularJS
44	1-4- مقدمة
44	2-4- المنهجية المتبعة في المقارنة
45	3-4- المقارنة
45	1-3-4- الدعم والتطوير
45	2-3-4- Data Binding
45	1-2-3-4- ربط البيانات بالنسبة ل Angular
46	2-2-3-4- ربط البيانات بالنسبة ل React
47	3-3-4- نوع DOM
47	1-3-3-4- Virtual DOM
47	2-3-3-4- Regular DOM
47	3-4- الخصبة (License)
47	4-3-4- الطبيعة والبنية (Nature and Structure)
48	5-3-4- المرونة (Flexibility)
48	6-3-4- الأداء (Performance)

48	1-7-3-4- اختبار المتصفحات
50	2-7-3-4- إحصائيات عدد المخازن على موقع GitHub
51	3-7-3-4- إحصائيات عدد المستخدمين على موقع الويب
53	8-3-4- سهولة التعلم
54	3-4- الخلاصة
55	الفصل الخامس: آفاق مستقبلية
55	1-5-آفاق مستقبلية
58	A- الملحق
58	1-A- تحميل Node.js
59	2-A- تنصيب Node.js
62	المراجع

الفصل الأول

مقدمة

1-1-مقدمة :

الويب هو النظام المتنامي بسرعة ويمكّننا العثور على الأجهزة التي تمكّن الإنترنت في كل مكان. للاتصال بالإنترنت ، يلزم استخدام تطبيقات الويب. تطبيق الويب هو في الأساس واجهة سهلة الاستخدام التي طورها مطورو الويب للتفاعل مع خدمات الويب تخزينها على شبكة الإنترنت العالمية. يستخدم مطورو الويب تقنيات الإنترت مثل HTML و CSS و XML و JavaScript و jQuery وما إلى ذلك لإنشاء صفحات الويب.

جافا سكريبت هي لغة برمجة حاسوبية موجهة للકائنات تستخدم عادة لخلق تأثيرات تفاعلية داخل متصفحات الويب. يتم استخدامه لجعل صفحات الويب ديناميكية وأكثر تفاعلاً. يتم استخدامه بواسطة عدة مستعرضات مثل MS Internet Explorer و Google Chrome و Firefox وما إلى ذلك ، يستخدم ما يقرب من 88٪ من جميع مواقع الويب جافا سكريبت لتصميم صفحات الويب. لأن من الميزات الكبيرة أن مجموعة جافا سكريبت لديها شعبية كبيرة. هناك العديد من التقنيات والمنصات القادمة لمساعدة المهندسين لتطوير تطبيقات الويب.

1-2-تقنية (Hyper Text Markup Language) HTML5 :

الإصدارات الأخيرة والمحسن من لغة HTML والتي هي ليست لغة برمجة في الحقيقة بل لغة تأشير لبرمجة موقع الانترنت والتي كانت وما زالت مدمجة ضمن برنامج front page ضمن حزمة البرامج المكتبية من شركة مايكروسوفت والتي يمكن كتابة اكوادها أو التعديل عليها باستخدام أي برنامج تحرير نصوص وخصوصاً notepad حيث يتم كتابة الاكواد وحفظ الملف بامتداد .html أو .htm. و تعتبر هذه النسخة الجديدة معياراً قياسياً لنقديم و هيكلة محتويات الشبكة الدولية WWW و يتطابق تماماً مع النسخ القديمة أي ان كل التطبيقات المكتوبة بالنسخ القديمة تعمل مع هذه الاصدارات الجديدة.

ان لغة html5 هي ببساطة مجموعة مميزات جديدة تم توفيرها لتطوير تطبيقات الويب اضافة الى القدرات والامكانيات الموجودة اصلاً في الاصدار السابق html4 وقد صمممت هذه اللغة اصلاً لتسهيل عملية الاتصال بالسيرفرات و ادارة الوسائل المتعددة multimedia لكي يجعل مهمة تطوير برامج و تطبيقات الويب اسهل بكثير .

تعتبر هذه النسخة الجديدة نتيجة لتعاون بين منظومة الشبكة الدولية العالمية world wide web consortium و مجموعة عمل التطبيقات التكنولوجية للنصوص الفاصلة للويب w3c technology working group WHATWG و يتضمن الاصدار الجديد مميزات اضافية جديدة مثل دعم التشغيل المباشر للفيديو في موقع الانترنت و الاليات السحب و الافلات drug and drop و التي كانت سابقاً تحتاج الى دعم برامج الطرف الثالث مدمج مع المتصفحات مثل مشغل الفلاش من شركة ادوبي adobe flash player و google gears و Microsoft Silverlight .

1-2-1- دعم المتصفحات للإصدار الجديد:

ما يجدر الاشارة اليه هو ان هذه اللغة ما زالت في طور التطوير و الدعم لها من قبل المتصفحات المختلفة و رغم ان الاصدارات الجديدة من متصفحات سفاري و كوكل كروم و فايرفوكس و الانترنت اكسيلورر 9 و ما بعده تدعم الكثير من خصائص هذه اللغة الا انه لا يوجد لحد الان شيء اسمه html5 compliant web browser اي لا يوجد متصفح انترنت يدعم بشكل كامل و بنسبة 100% كل الخصائص الجديدة لهذا الاصدار وما زال الامر في طور التطوير و التحسين و اضافة الى متصفحات الانترنت للحواسيب المكتبية و المحمولة فإن المتصفحات المصغرة mini browser او ما تسمى متصفحات الاجهزة النقالة mobile web browsers و التي تأتي مدمجة ضمن هذه الأجهزة اللوحية و النقالة مثل ipad,iphone,android phones كل هذه المتصفحات تحتوي دعماً ممتازاً للكثير من مميزات هذا الاصدار.

1-2-2- الخصائص الجديدة :

يمتلك المقياس العالمي الجديد الكثير من الخصائص التي تهدف الى تسهيل و تطوير عملية برمجة موقع و تطبيقات الويب ومنها :

1- عناصر التحكم في شكل الموقع الجديدة مثل : <section> <footer> <header>

2- الاصدار الثاني من نماذج الويب form2 و الذي يعتبر تحسين لنماذج html للويب للإصدارات السابقة بإضافة خصائص جديدة لوسوم الـ <input> .

3- الخزن المحلي الدائم : و الذي يعمل عمل التطبيقات من الطرف الثالث التي سبق التطرق لها و يلغى الحاجة لها .

4- مقابس الشبكة العنكبوتية Web sockets وهي جيل جديد من تكنولوجيا الاتصال ثنائية الاتجاه لتسهيل الاتصال بين تطبيقات الويب المتبااعدة .

5- ارسال احداث الخادم server- sent events SSE هذا الاصدار الجديد يقدم احداث الخادم وينقلها من خادم الويب الى متصفح الويب بخدمة تسمى ارسال احداث الخادم .

6- Canvas : وهي خاصية دعم لسطح رسوم ثنائية الابعاد يمكن برمجتها بـ javascript

7- الصوتيات و الفيديو : يمكنك الان دمج الصوتيات و الفيديو في موقعك و صفحاته بدون الحاجة الى تنصيب تطبيق طرف ثالث مشغل فلاش مثلاً .

8- الموقع الجغرافي : حالياً يستطيع المستخدمون و المبرمجون لموقع الانترنت ان يشاركون موقعهم الجغرافي مع الآخرين كما في خدمة تحديد الموقع في موقع الفيس بوك مثلاً.

9- البيانات الدقيقة **micro data**: و تسمح هذه الخاصية الجديدة للمبرمج بأن يتذكر ويطرد تصاميم تحكم بالشكل **semantics** جديدة وعدم التقيد بما توفره اللغة للحصول على تحكم أكبر بشكل صفحات الموقع المطور .

10-السحب والافلات : تسمح هذه الخاصية بسحب عنصر معين من صفحة الويب و افلاته في مكان اخر من نفس الصفحة ولم تكن قابلية كهذه موجودة في الاصدارات السابقة .

1-2-3- الصيغ العامة للغة :

تأتي لغة html5 متطابقة تماماً مع html4.1 و xhtml1 من حيث كيفية كتابة الوسوم و بالشكل المأثور اي ان كل شيء مكتوب بالنسخ السابقة يمكن فهمه من قبل متصفحات النسخ الجديدة و لكنها غير متطابقة مع الاصدارات الاقدم من 4.1 ومع ذلك فهذا الاصدار جاء بمرونة عالية في كتابة الوسوم وكما يلي :

- 1- يمكن هنا استخدام الاسماء بحروف كبيرة بعد ان كانت بعض اللغات القديمة تشترط استخدام الحروف الصغيرة فقط في وسومها .

- 2- علامات الاقباس هنا اصبحت اختيارية لوصف بعض الخصائص **attributes** .
- 3- استناد قسم للخصائص اصبح اختيارياً أيضاً بعد ان كان اجبارياً سابقاً.
- اغلاق العناصر الفارغة اصبح اختيارياً ايضاً بعد ان كانت اجبارياً سابقاً .

1-2-4- من الوسوم الجديدة في html5 :

- 1 <**artical**>: يدل على المحتوى (المقالة) .
- 2 <**aside**> يدل على الشريط الجانبي .
- 3 <**details**> يدل على المزيد من المعلومات التي يمكن للمستخدم اظهارها او اخفائها .
- 4 <**figcaption**> وصف للرسوم و الصورة التوضيحية .
- 5 <**figure**> صور توضيحية .
- 6 <**footer**> ذيل الصفحة او القطاع .
- 7 <**header**> رأس الصفحة او القطاع .
- 8 <**main**> يدل على المحتوى الرئيسي للصفحة .
- 9 <**mark**> يدل على نص مميز في الصفحة .
- 10 <**nav**> يدل على شريط **navigation** أو التصفح في الموقع .
- 11 <**section**> يعرف قطاع في الصفحة .
- 12 <**summary**> يعرف الجزء الظاهر من <**details**>

- 13 **<time>** يعرف الوقت والتاريخ .
- 14 **<bdi>** تحديد جزء من النص و الذي يمكن أن يكون تم تنسيقه بشكل مختلف عن سياق النص الأصلي .
- 15 **<command>** تحديد أمر معين يمكن للمستخدم استحضاره من خلال الضغط على زر.
- 16 **<hgroup>** تجميع جميع العناوين من العديد من العناوين الفرعية ضمن عنصر واحد اذا كان العنوان الرئيسي يحتوي على **<h1>** الى **<h6>** .
- 17 **<meter>** تحديد وحدة قياس لمدى معروف.
- 18 **<progress>** اظهار الحالة لمهمة معينة .
- 19 **<ruby>** تعريف الكلمات الغريبة مثل لغات شرق آسيا.
- 20 **<rt>** تحديد مجموعة أحرف وكلمات غريبة مثل لغات شرق آسيا .
- 21 **<rp>** تعريف كلمات غريبة لم يتمكن عنصر أو المتصفح من قرائتها.
- 22 **<wbr>** تحديد عدد الخطوط الممكنة.
- 1-2-5- عناصر الوسائط المتعددة الجديدة:**
- 1 **<audio>** تعريف محتوى صوتي .
- 2 **<video>** تعريف مقطع فيديو او فيلم.
- . **<audio>,<video>** -3 تحديد مصادر وسائط متعددة مختلفة للعناصر
- . **<embed>** -4 تعريف عنصر لتطبيق خارجي او محتوى تفاعلي .
- . **<audio>,<video>** -5 تحديد مسارات نصية للعناصر
- 1-2-6- العناصر المحذوفة من html5 :**
- 1 **<acronym>**
- 2 **<applet>**
- 3 **<basefont>**
- 4 **<big>**
- 5 **<center>**
- 6 **<dir>**
- 7 ****
- 8 **<frame>**
- 9 **<frameset>**

<noframes>-10

<strike>-11

<tt>-12

<u>-13

1-2-7-لماذا تعد الأفضل ؟

1- لن يتم تخزين بيانات الصفحة لديك في المتصفح ولا حتى في مشغل الصفحة نفسه هي تخزن

للمرة الواحدة فقط وعند الإنتهاء وإغلاق الصفحة مباشرة لا يبقى لها أثر.

2- أسرع والسبب هو أنها تستقبل بيانات اخف بكثير .

3- لأن المستخدم سوف يكون المتحكم بالمحظى وليس المحظى مفروض عليه.

3-1-(CSS) cascading style sheet :

إن css هي اختصار للأحرف الأولى من cascading style sheet والتي ترجمت على انها "أوراق انماط الانسياحية" ، ففي حين أن html تهتم بوصف عناصر مستندات الويب فإن css تهتم بوصف شكل و مظهر هذه العناصر ، وقد كان ظهور أوراق الأنماط الانسياحية الماحق الرئيسي لمشكلة من أكبر المشاكل التي واجهت مطوري الويب لسنوات قبل ظهورها .

1-3-1- مشكلة مطوري الويب قبل css :

عندما تم تصميم لغة html كان التركيز الأكبر على تصميم لغة تصصف عناصر صفحات الويب فقط ولم يكن هناك اي توجه لدى مصممي html أن تهتم لغتهم بمظهر و شكل المحتوى المقدم ، و تركت هذه المهمة لمصممي مستعرضات الويب ، حيث كان -ومازال- كل مستعرض من مستعرضات الويب يعرض كل عنصر من عناصر صفحة الويب بمظهر افتراضي خاص به حسبما رأى مصممو المستعرض وبغض النظر عن رغبة مصمم الصفحة.

تم التتبه لهذه النقطة لاحقاً فأضفت وسوم جدسته تهتم بالتنسيق ومن أمثلتها الوسم الذي يستخدم لعرض نص بنوع خط وحجم ولون محدد ضمن الصفحة ، وكان هذا شيئاً جميلاً في أول الأمر لكنه سرعان ما تحول لكارثة بالنسبة لمطوري الويب ، فقد كان هناك بعض المواقع التي يصل عدد صفحاتها إلى مئات أو الآلاف الصفحات أحياناً ، مما يعني أن عملية تنسيق هذه الصفحات تحتاج لوقت وجهد كبيرين، ناهيك عن الحاجة إلى التركيز الشديد في ضبط خصائص الوسوم المذكورة سابقاً للحصول على صفحات بألوان وأحجام وأنواع خطوط موحدة و متباينة ، لكن ومن حسن الحظ فقد تلاشت هذه المشكلة مع ظهور css .

مبدأياً يمكننا القول أن css تصف مظهر وشكل كل وسم من وسوم html على حد، وهذا ما يعني بالنسبة لمطوري الويب أنه سيقوم بكتابة وصف لشكل و مظهر الوسم في مكان واحد ولمرة واحدة فقط ، ثم سيطبق هذا التنسيق على جميع مستندات الويب المرتبطة بذلك css .

: (Css3)-2-3-1

مع تطور مفهوم الويب (خصوصاً مع ظهور web 2.0) ظهرت الحاجة إلى تحسين أوراق الأنماط الاسبانية لتحقيق تصاميم أجمل في موقع الويب دون الحاجة إلى الإفراط في استخدام برامج التصميم وهذا ما دفع إلى ظهور الإصدار الجديد من أوراق الأنماط الاسبانية و المعروف بـ .css3

تضييف css3 مجموعة من المحددات والخصائص أو المحددات باسم الوحدة Module ولعل أهم الوحدات المضافة هي :

- وحدة المحددات الجديدة .
- وحدة المحدد و الخلفيات الجديدة
- وحدة تحسين نموذج الصندوق .
- وحدة مؤثرات النصوص الجديدة .
- وحدة التحويلات الهندسية ذات البعدين .
- وحدة التحويلات الهندسية ذات الثلاثة أبعاد .
- وحدة الحركات
- وحدة الصفحات متعددة الأعمدة .
- وحدة واجهة المستخدم .

المستعرضات الداعمة :

بالنسبة لـ css3 فهي مدرومة من اغلب مستعرضات الويب الحديثة وبشكل أوسع بكثير من دعم ذات المستعرضات لـ html5 .

: CSS3-3-1 ميزات الـ**الحدود الجديدة : css3 borders**

● تتيح css3 إنشاء الحدود ذات الزوايا غير القائمة round corner borders بسهولة و يسر ودون الحاجة لاستخدام أي من برامج التصميم مثل photoshop التي طالما استخدمت لتحقيق هذه المهمة عبر خاصية جديدة هي الخاصية border-radius .

● كما تتيح css3 إنشاء حدود العناصر بالاعتماد على صورة صغيرة تعبر عن أصغر شكل ممكث للإطار .
ويتم إنشاء الإطار بطريقتين :

- Repeated : بتكرار جزء من الصورة .
- Stretch : بتمديد جزء من الصورة .

● تتيح css3 أيضاً إنشاء ظل لعنصر ما عبر الخاصية box-shadow والتي يسند إليها قيمة لونية تعبر عن لون الظل و قيم بالبكسل تعبر عن إحداثيات الظل .

الخلفيات الجديدة : css3 backgrounds

- في css3 تم إضافة خاصيتين جديدتين لمساعدة مطور الويب على التحكم بالخلفيات بشكل أفضل ، الأولى هي الخاصية **background-size** والتي تسمح بتحديد أبعاد الصورة الخلفية بالبكسل مما يسمح بإعطاء الصورة كخلفية أبعاداً تختلف عن الأبعاد الأصلية للصورة وهو ما لم يكن ممكناً قبل css3 . أما الخاصية الثانية فهي **background-origin** والتي تحدد أحد أجزاء نموذج الصندوق ليتم تطبيق الخلفية على أساسه .
- من التحسينات التي أضيفت إلى خاصية الخلفية في css3 السماح بإضافة أكثر من صورة في نفس الوقت كخلفية لنفس العنصر وذلك عبر الفصل بين كل خلفية و الأخرى بالفواصل .

خصائص تنسيق النصوص الجديدة في css3 :

تم إضافة مجموعة من الخصائص الجديدة لتنسيق النصوص في css3 منها :

- **Hanging-punctuation** تحدد وضع علامة الترقيم خارج حدود العنصر <p> .
- **Punctuation-trim** تزيل علامة الترقيم الافتتاحية من بداية كل سطر من أسطر <p> .
- **text-align** تحدد طريقة تحقيق المحاذة عندما تكون قيمة الخاصية مساوية لـ .

التحولات الهندسية ذات البعدين : 2d transformation

تمكننا css3 من إجراء تحويلات هندسية ثنائية البعدين على عناصر الصفحة كالانسحاب translate و التدوير rotate و غيرها بإسناد أحد توابع التحويل إلى الخاصية transform .

التحولات الهندسية ثلاثية الأبعاد : 3d transformation

تبعد css3 القيام بتحويلات هندسية ثلاثية الأبعاد على العناصر بنفس الأسلوب السابق .

الانتقال المتحرك : transition

توفر css3 طريقة جميلة ومميزة لإنشاء تأثير على عناصر المستند يعرف هذا التأثير الخاص باسم الانتقال المتحرك والذي يتم تحقيقه عبر الخاصية transition حيث يتم تطبيق قاعدة css مختلفة عن القاعدة الأصلية للعنصر خلال زمن محدد مما يظهر العنصر بشكل متحرك .

حركات css3 :

تبعد css3 إنشاء حركات مخصصة في صفحة الويب مما يعني مطور الويب عن استخدام صور gif المتحركة أو تأثيرات flash وغيرها ضمن الصفحة .

الأعمدة المتعددة في css3 :

تسمح css3 بعرض محتوى العناصر على شكل أعمدة متعددة multi-columns على مبدأ الجرائد بطريقتين:

- عدد الأعمدة ثابت وعرض عمود متغير ويتم ذلك من خلال إسناد عدد الأعمدة إلى الخاصية column-count التي تقسم المحتوى إلى العدد المذكور من الأعمدة ، و يتغير عرض الأعمدة مع تغير حجم نافذة المستعرض .

- عرض عمود ثابت وعدد أعمدة متغير ويتم ذلك من خلال عرض العمود الواحد إلى الخاصية `column` و المسافة بين كل عمودين إلى الخاصية `column-gap` وفي هذه الطريقة تحافظ الأعمدة على عرض ثابت لكل منها و يتغير عددها مع تغير حجم نافذة المستعرض .

واجهة المستخدم : css3 user interface

بالإضافة لما قدمته css3 من ميزات لمن ينشئ صفحات الويب فقد قدمت مجموعة من التحسينات للمستخدم النهائي للصفحة عبر الخصائص التي يعرضها منها :

- Appearance تحدد لمستعرض الويب كيفية عرض العنصر ضمن المستند .
- Icon تتيح جعل عنصر ما بشكل أيقوني .
- Nav-down/nav-left/nav-right/nav-up تحدد أين سيتم الانتقال عندما يضغط المستخدم على مفاتيح الأسهم في لوحة المفاتيح .
- Nav-index يحدد العدد المستند إليها عدد ضغطات مفتاح tab اللازمة للانتقال إلى العنصر أو بشكل أدق تسلسل الانتقال بين العناصر عند الضغط على مفتاح tab
- Resize تحدد فيما إذا كان من المسموح للمستخدم تغيير حجم أحد عناصر div بشكل يدوي .

: Node js 4-1

هي بيئة تشغيل جافا سكريبت مفتوحة المصدر وعبر النظام الأساسي تقوم بتنفيذ جافا سكريبت من جانب الخادم. من الناحية التاريخية ، تم استخدام جافا سكريبت بشكل أساسي في البرمجة النصية من جانب الزبائن ، حيث يتم تضمين النصوص المكتوبة بلغة جافا سكريبت في HTML لصفحة الويب وتشغيل الزبائن من خلال محرك جافا سكريبت في متصفح الويب الخاص بالمستخدم. يتيح JavaScript للمطربين استخدام Node.js للبرمجة النصية من جانب الخادم - تشغيل البرامج النصية من جانب الخادم لإنتاج محتوى صفحة ويب ديناميكية قبل إرسال الصفحة إلى متصفح الويب الخاص بالمستخدم. وبالتالي ، يمثل Node.js نموذج "جافا سكريبت في كل مكان" ، يوحد تطوير تطبيق الويب حول لغة برمجة واحدة ، بدلاً من لغات مختلفة للنصوص الجانبيّة للخادم والنصية من جانب الزبائن.

على الرغم من أن .js هو امتداد اسم الملف التقليدي لرمز JavaScript ، لا يشير الاسم "Node.js" إلى ملف معين في هذا السياق وهو مجرد اسم المنتج. يحتوي Node.js على بنية تحرك الحدث قادرة على I / O غير متزامن. تهدف خيارات التصميم هذه إلى تحسين الإنتاجية والقابلية للتطوير في تطبيقات الويب مع العديد من عمليات الإدخال / الإخراج ، بالإضافة إلى تطبيقات الويب في الوقت الفعلي (على سبيل المثال ، برامج الاتصال في الوقت الحقيقي وألعاب المتصفح).

تم تطوير مشروع التطوير Node.js الموزع ، الذي تحكمه مؤسسة Node.js ، من خلال برنامج مشاريع التعاون التابعة لمؤسسة لينكس.

افضل مثال على الشركات التي تستخدم Node.js هي alo.yahoo.com كتب Ryan Dahl في الأصل من قبل في عام 2009 تم استلهام Dahl لإنشاء Node.js بعد رؤية شريط تقدم تحميل الملفات على Flickr. لم يكن المتصفح يعرف حجم الملف الذي تم تحميله وكان عليه الاستعلام عن خادم الويب. أراد Dahl طريقة أسهل. انتقد Dahl الإمكانيات المحدودة لخادم الويب الأكثر شعبية في عام 2009 ، Apache HTTP Server ، للتعامل مع الكثير من الاتصالات المتزامنة (ما يصل إلى 10000 وأكثر) والطريقة الأكثر شيوعاً لإنشاء التعليمية البرمجية (البرمجة المتسلسلة) ، عندما حظر الكود العملي بأكملها أو مكدسات التنفيذ المتعددة الضمنية في حالة الاتصالات المتزامنة.

في يناير 2010 ، تم تقديم مدير الحزم لبيئة Node.js المسماً npm . يجعل مدير الحزم من الأسهل للمبرمجين نشر ومشاركة شفرة المصدر لمكتبات Node.js ، وهو مصمم لتيسير عملية تركيب ، وتحديث ، وإلغاء تثبيت المكتبات. في يونيو 2011 ، نفذت MicrosoftJoyent و إصدار Windowsالأصلي من Node.js. تم إصدار أول Windows Node.js لدعم في يونيو 2011.

في فبراير 2015 ، تم الإعلان عن نية تشكييل مؤسسة Node.js محايده. بحلول يونيو 2015 ، صوت مجتمعات Node.js و io.js للعمل معاً في إطار مؤسسة Node.js.

في سبتمبر 2015 ، تم دمج v0.12 Node.js v3.3 و io.js v4.0 معاً في Node.js v4.0. هذا جلبت ميزات ES6 إلى Node.js ، ونورة إطلاق دعم على المدى الطويل. اعتباراً من عام 2016 ، يوصي موقع ويب io.js بإعادة المطوريين إلى Node.js ومن غير المقرر إصدار المزيد من الإصدارات io.js بسبب عملية الدمج. يسمح Node.js بإنشاء خوادم ويب وأدوات الشبكات باستخدام JavaScript ومجموعة من "الوحدات" التي تعامل مع الوظائف الأساسية المختلفة. يتم توفير الوحدات النموذجية لنظام الإدخال / الإخراج الخاص بالنظام ، والشبكات (DNS ، TCP ، HTTP ، TLS / SSL ، UDP) ، البيانات الثنائية (المخازن المؤقتة) ، وظائف التشفير ، تدفقات البيانات ، والوظائف الأساسية الأخرى. تستخدم وحدات Node.js واجهة برمجة التطبيقات مصممة لتنقیل تعقيد تطبيقات خادم الكتابة.

يمكن تشغيل تطبيقات Node.js على خوادم Linux و MacOS و Microsoft Windows و Unix و NonStop أو Unix. أو بدلاً من ذلك ، يمكن كتابتها مع CoffeeScript (بدليل لجافا سكريبت) أو Dart أو TypeScript (أشكال مكتوبة بقوة من JavaScript) أو أي لغة أخرى يمكن أن تحول إلى JavaScript. يتم استخدام Node.js بشكل أساسى لبناء برامج الشبكة مثل خوادم الويب. أكبر اختلاف بين Node.js و PHP هو أن معظم الوظائف في PHP حتى الاتكمال (الأوامر تنفذ فقط بعد الانتهاء من الأوامر السابقة) ، في حين أن وظائف Node.js غير قابلة للحظر (تم تنفيذ الأوامر بالتزامن أو حتى بالتوالي ، واستخدام الاسترجاعات لإكمال الإشارة أو الفشل)

يجلب Node.js البرمجة القائمة على الأحداث إلى خوادم الويب ، مما يتبع تطوير خوادم الويب السريعة في JavaScript. يمكن للمطوريين إنشاء خوادم قابلة للتجمع بشكل كبير دون استخدام خيوط ، وذلك باستخدام نموذج

مبسط للبرامج التي تعتمد على الأحداث والتي تستخدم عمليات الاستدعاء للإشارة إلى إكمال المهمة. تربط Node.js بسهولة لغة البرمجة النصية (JavaScript) مع قوة برمجة شبكة Unix.

بنيت Node.js على محرك جوجل V8 جافا سكريبت لأنّه كان مفتوح المصدر تحت رخصة BSD ، سريع للغاية ، ويتقن مع أساسيات الإنترنت مثل TCP ، DNS ، HTTP . أيضًا ، كانت JavaScript لغة مشهورة ، مما يجعل Node.js متاحًا على الفور لمجتمع تطوير الويب بأكمله.

Bootstrap-5-1 :

هي مكتبة واجهة مفتوحة المصدر مجانية لتصميم موقع الويب وتطبيقات الويب. يحتوي على قالب تصميم HTML و CSS لا ، و CSS ، typography ، buttons ، forms ، ومكونات الواجهة الأخرى ، بالإضافة إلى ملحقات JavaScript الاختيارية. على عكس العديد من أطر الويب ، فإنها تعامل مع تطوير الواجهة الأمامية فقط.

بعد Bootstrap ثانٍ أكثر المشاريع نجومًا في GitHub ، حيث يضم أكثر من 123 ألف نجم.

تم تطوير Bootstrap ، الذي كان اسمه في الأصل على تويتر Blueprint ، بواسطة Mark Otto و Jacob Thornton على تويتر كإطار لتشجيع الاتساق عبر الأدوات الداخلية. قبل Bootstrap ، تم استخدام مكتبات مختلفة لتطوير الواجهة ، مما أدى إلى تنافضات وعقبات صيانة مرتفع. وفقاً لمطور تويتر Mark Otto:

لقد تعاونت مجموعة صغيرة جدًا من المطوريين معًا لتصميم وبناء أداة داخلية جديدة ورأينا فرصة للقيام بشيء آخر. ومن خلال هذه العملية ، رأينا أنفسنا نبني شيئاً أكبر بكثير من أداة داخلية أخرى. بعد مرور أشهر ، انتهينا مع إصدار مبكر من Bootstrap كطريقة لتوثيق وتقاسم أنماط وأصول التصميم المشتركة داخل الشركة.

بعد بضعة أشهر من التطوير من قبل مجموعة صغيرة ، بدأ العديد من المطوريين في تويتر بالمساهمة في المشروع كجزء من أسبوع Hack ، وهو أسبوع على غرار Hackathon لفريق تطوير Twitter. تمت إعادة تسمية من Twitter إلى Blueprint ، وتم إصداره كمشروع مفتوح المصدر في 19 أغسطس 2011. استمر في الحفاظ عليه Jacob Thornton ، Mark Otto ، و مجموعة صغيرة من المطوريين الأساسيين ، بالإضافة إلى مجموعة كبيرة من المساهمين.

في 31 يناير 2012 ، تم إصدار 2 Bootstrap ، والذي أضاف نظام تخطيط الشبكة المتقارب ذو 12 عمودًا ، دعمًا للمكونات Glyphicons ، العديد من المكونات الجديدة ، بالإضافة إلى التغييرات في العديد من المكونات الموجودة.

في 19 أغسطس 2013 ، تم إصدار 3 Bootstrap ، والتي أعادت تصميم المكونات لاستخدام التصميم المسطح ، ونبع المحمول الأول.

في 29 أكتوبر 2014 ، أعلن مارك أوتو أن 4 Bootstrap كان قيد التطوير. تم إصدار أول نسخة لها من Bootstrap في 19 أغسطس 2015. تم إصدار النسخة التجريبية الأولى في 10 أغسطس 2017. تم إصدار

النسخة المستقرة من 4 Bootstrap في 18 يناير ، 2018 بشكل أساسي لتحسين أساليب الطباعة ، وأدوات المساعدة على الحدود وتوفير المزيد من التحكم في .flexbox يدعم 3 أحدث الإصدارات من Internet Explorer و Firefox و Google Chrome و Opera و Safari (باستثناء Windows) . بالإضافة إلى ذلك يدعم العودة إلى IE8 وأحدث إصدار دعم تمديد (ESR) firefox.

منذ الإصدار 2.0 يدعم Bootstrap تصميم الويب سريع الاستجابة. هذا يعني أن تخطيط صفحات الويب يضبط بشكل ديناميكي ، مع الأخذ في الاعتبار خصائص الجهاز المستخدم (سطح المكتب ، الكمبيوتر اللوحي ، الهاتف المحمول).

بدءاً من الإصدار 3.0 ، اعتمدت Bootstrap فلسفة التصميم المتنقلة الأولى ، مع التركيز على التصميم سريع الاستجابة بشكل افتراضي. وأضاف الإصدار ألفا الإصدار 4.0 دعم ساس و .flexbox

Bootstrap هو وحدات ويكون من سلسلة من أوراق أنماط أقل تقوم بتطبيق المكونات المختلفة لمجموعة الأدوات. عادة ما يتم تجميع أوراق الأنماط هذه في حزمة ويتم تضمينها في صفحات الويب ، ولكن يمكن تضمين المكونات الفردية أو إزالتها. يوفر Bootstrap عدداً من متغيرات التكوين التي تحكم في أشياء مثل اللون (color) وحشو (padding) (المكونات المختلفة).

منذ 2 Bootstrap ، شملت وثائق Bootstrap معالج التخصيص الذي يولد نسخة مخصصة من على أساس المكونات المطلوبة والإعدادات المختلفة.

اعتباراً من 4 Bootstrap ، يتم استخدام Sass بدلاً من Less في أوراق الأنماط. يتكون كل مكون من بنية HTML ، وإعلانات CSS ، وفي بعض الحالات مصاحبة شفرة JavaScript

نظام الشبكة والتصميم المتغاب يأتي بشكل قياسي مع تخطيط شبكة 1170 بكسل. بدلاً من ذلك ، يمكن للطور استخدام تخطيط متغير العرض. في كلتا الحالتين ، تحتوي مجموعة الأدوات على أربعة أشكال للاستفادة من مختلف أنواع الأجهزة وأنواعها: الهاتف المحمولة والعرضية والمناظر الطبيعية والأجهزة اللوحية وأجهزة الكمبيوتر ذات الدقة المنخفضة والعالية. كل تباين يضبط عرض الأعمدة.

في 29 أكتوبر 2014 ، أعلن Mark Otto أن 4 Bootstrap كان قيد التطوير. في 6 سبتمبر 2016 ، علق العامل على 3 Bootstrap من أجل توفير الوقت للعمل على 4 Bootstrap. تم إجراء أكثر من 4,000 تم الاتمام بها إلى 4 Chemebase Bootstrap حتى الآن.

يكاد يكون إعادة كتابة كاملة من 3 Bootstrap يكاد يكون إعادة كتابة كاملة من 4. التغييرات الهامة ما يلي:

- تحولت من Less إلى .Sass
- إسقاط IE8 ، ودعم iOS 6 .
- واضاف الدعم .non-flexbox ثم انخفض الدعم .root ems
- تحولت من بكسل إلى .ems
- زيادة حجم الخط من 14 بكسل إلى 16 بكسل.

- إسقاط لوحة ، صورة مصغرة ، ومكونات جيدة لصالح البطاقات.
- اسقط خط رمز Glyphicons.
- إسقاط مكون الصفحة .
- أعاد كتابة جميع المكونات تقريرًا ، ومكونات الإضافات jQuery والوثائق

٦-١ : Typescript

هي لغة برمجة مفتوحة المصدر تم تطويرها وصيانتها بواسطة Microsoft. هو مجموعة شاملة من نصوص جافا سكريبت صارمة ، ويضيف كتابة ثابتة اختيارية للغة.

تم تصميم TypeScript لتنمية التطبيقات الكبيرة وترجمة JavaScript. بما أن TypeScript عبارة عن مجموعة كبيرة من JavaScript ، فإن برامح JavaScript الموجودة هي أيضًا برامح TypeScript صالحة. يمكن استخدام TypeScript لتطوير تطبيقات مثل JavaScript للتنفيذ من جانب العميل والخادم (Node.js).

يدعم TypeScript ملفات التعريف التي يمكن أن تحتوي على معلومات الكتابة الخاصة بمكتبات JavaScript الموجودة ، مثلما يمكن أن تصف ملفات Röros C++ بنية ملفات الكائن الموجودة. وهذا يمكن البرامج الأخرى من استخدام القيم المعرفة في الملفات كما لو كانت مكتوبة بشكل ثابت كبيانات TypeScript. هناك ملفات رأس لجهة خارجية للمكتبات الشائعة مثل MongoDB و D3.js و jQuery. توجد رؤوس TypeScript للوحدات الأساسية Node.js أيضًا ، مما يسمح بتطوير برامح Node.js في TypeScript.

مكتوب بلغة البرمجة النصية لـ TypeScript نفسه في JavaScript ويتم ترجمته إلى JavaScript. مرخص بموجب ترخيص Apache 2.

يتم تضمين TypeScript لغة برمجة من الدرجة الأولى في Microsoft Visual Studio 2013 Update 2 والإصدارات الأحدث ، بجانب C # ولغات Microsoft الأخرى. يسمح ملحق رسمي لبرنامج Visual Studio 2012 بدعم TypeScript أيضًا.

قام Anders Hejlsberg ، المهندس الرئيسي لـ C # ومبدع Turbo Pascal و Delphi ، بالعمل على تطوير TypeScript.

تم نشر TypeScript لأول مرة في أكتوبر 2012 (الإصدار 0.8) ، بعد عامين من التطوير الداخلي في Microsoft IDE. بعد وقت قصير من الإعلان ، أثني ميجيل دي إيكازا على اللغة نفسها ، لكنه انتقد عدم وجود دعم الناضج بعيًّا عن Microsoft Visual Studio ، والذي لم يكن متاحًا على Linux و OS X في ذلك الوقت. اعتباراً من عام 2013 ، هناك دعم في IDEs أخرى ، لا سيما في Eclipse ، من خلال المكونات التي ساهمت بها

Palantir Technologies و Sublime و Vim و Emacs ، بما في ذلك TypeScript . Microsoft تدعم أيضًا Visual Studio Code و Atom و Webstorm . تم إصدار TypeScript 0.9 ، الذي صدر في عام 2013 ، أضاف الدعم للأدوية الجنسية. تم إصدار TypeScript 1.0 في مؤتمر Microsoft Build المطور في عام 2014. يوفر 2013 Update دعمًا مضمّنًا لنوع TypeScript .

في يوليو 2014 ، أُعلن فريق التطوير عن مترجم جديد لـ TypeScript ، مدعياً مكاسب أداء ×5. في الوقت نفسه ، تم نقل شفرة المصدر ، التي استضافت في البداية على GitHub ، إلى CodePlex . في 22 سبتمبر 2016 ، تم إصدار 2.0 TypeScript ؛ قدم العديد من الميزات ، بما في ذلك قدرة المبرمجين على منع المتغيرات من تحصيص قيم فارغة.

نشأت TypeScript من العيوب المتضورة لجافا سكريبت لتطوير تطبيقات واسعة النطاق على كل من Microsoft وبين عملائها الخارجيين. أدت التحديات في التعامل مع شفرة جافا سكريبت المعقدة إلى الطلب على الأدوات المخصصة لتسهيل تطوير المكونات في اللغة.

سعى مطورو TypeScript إلى حل لا يكسر التوافق مع المعيار ودعمه عبر الأنظمة الأساسية. مع العلم أن الاقتراح القياسي الحالي ECMAScript وعدت الدعم المستقبلي للبرمجة القائمة على الطبقة ، واستند TypeScript على هذا الاقتراح. أدى ذلك إلى مترجم جافا سكريبت مع مجموعة من الامتدادات اللغوية النحوية ، وهي مجموعة كبيرة تعتمد على الاقتراح ، والتي تحول الإضافات إلى جافا سكريبت عادية. من هذا المنطلق ، كان TypeScript بمثابة معاينة لما يمكن توقعه في ECMAScript 2015. والجانب الفريد ليس في الاقتراح ، ولكنه تمت إضافته إلى TypeScript ، هو كتابة ثابتة اختيارية تمكن تحليل اللغة الثابت ، مما يسهل الأدوات ودعم IDE.

7-1 : Gulp.js

عبارة عن مجموعة أدوات جافا سكريبت مفتوحة المصدر من Fractal Innovations ومجتمع المصدر المفتوح في GitHub ، المستخدم كنظام لإنشاء بث في تطوير الويب الأمامي.

وهو عبارة عن أداة تشغيل مبنية على Node.js و npm ، وتستخدم لأنماط المهام التي تستغرق وقتاً طويلاً والمتركرة والمضمنة في تطوير الويب مثل التضمين ، والتسلسل ، وخرق ذاكرة التخزين المؤقت ، واختبار الوحدة ، والإسراع ، والتحسين ، وما إلى ذلك.

يستخدم gulp نهج التعليمية البرمجية أكثر من اللازم لتحديد مهامها ويعتمد على المكونات الإضافية الصغيرة ذات الغرض الواحد لتنفيذها. يتضمن النظام البيئي gulp أكثر من 300 من هذه المكونات الإضافية.

gulp هي أداة بناء في JavaScript مبنية على تدفقات العقدة. هذه التدفقات تسهل ربط عمليات الملف عبر خطوط الأنابيب. يقرأ gulp نظام الملفات ويوجه البيانات في متناول اليد من البرنامج المساعد الواحد المفرد إلى آخر من خلال مشغل() pipe ، ويقوم بمهمة واحدة في كل مرة. لا تأثر الملفات الأصلية حتى تتم معالجة جميع الملحقات.

يمكن تكوينه إما لتعديل الملفات الأصلية أو لإنشاء ملفات جديدة. هذا يمنح القدرة على أداء المهام المعقدة من خلال ربط العديد من الإضافات. يمكن للمستخدمين أيضًا كتابة المكونات الإضافية الخاصة بهم لتحديد مهامهم الخاصة.

بخلاف برمج تشغيل المهام الأخرى التي تقوم بتشغيل المهام حسب التكوين ، تتطلب JavaScript gulp معرفة gulp وترميزها لتحديد مهامها. gulp هو نظام بناء يعني أنه بعيدًا عن تشغيل المهام ، فإنه قادر أيضًا على نسخ الملفات من موقع إلى آخر ، أو تجميع ، أو نشر ، أو إنشاء إشعارات ، أو اختبار الوحدة ، أو الفحص ، إلخ.

يتم إنشاء المتسابقين مثل gulp و Grunt على Node.js بدلًا من npm ، لأن البرامج النصية npm الأساسية غير فعالة عند تنفيذ مهام متعددة. على الرغم من أن بعض المطوريين يفضلون البرامج النصية npm لأنها يمكن أن تكون بسيطة وسهلة التنفيذ ، وهناك العديد من الطرق حيث يدو أن الجلب و Grunt يتمتعان بميزة فوق بعضها البعض والتصوّص الافتراضية المقدمة. يقوم Grunt بتشغيل المهام عن طريق تحويل الملفات وحفظها كحافظات جديدة في مجلدات مؤقتة ويتم أخذ إخراج إحدى المهام كمدخلات إلى آخر وهكذا حتى يصل الإخراج إلى المجلد الوجهة. يتضمن هذا الكثير من استدعاءات O / وإنشاء العديد من الملفات المؤقتة. في حين أن gulp يتدقق عبر نظام الملفات ولا يتطلب أليًا من هذه الموضع المؤقتة تقليل عدد مكالمات O / وبالتالي تحسين الأداء. يستخدم Grunt ملفات التهيئة لأداء المهام بينما تتطلب gulp ترميز ملف البناء الخاص به. في Grunt ، يجب تهيئه كل مكون إضافي ليطابق موقع إدخاله مع خرج المكون الإضافي السابق. في gulp ، يتم تلقائيًا تطبيق الإضافات.

الفصل الثاني

تقنيات الجافا سكريبت

1-2-مقدمة:

هي لغة برمجة عالية المستوى ومفقرة ، وهي لغة تميز بأنها ديناميكية سهلة الكتابة ذات نمذجة أولية ، متعددة النماذج ، تعد واحدة من التقنيات الأساسية الثلاثة لشبكة الويب العالمية إلى جانب HTML و CSS ، تضفي التفاعلية على صفحات الويب وتتوفر ببرامج عبر الأنترنت بما في ذلك ألعاب الفيديو.

توظفها غالبية الموقع، وجميع متصفحات الويب الحديثة دون الحاجة إلى مكونات إضافية عن طريق محرك (JavaScript) المدمج ، وباعتبارها لغة متعددة النماذج ، تدعم جافا سكريبت أنماط البرمجة القائمة على الأحداث والوظيفية (بما في ذلك أنماط البرمجة القائمة على الكائنات ونمذجها) .

ولديه واجهة برمجة تطبيقات للتعامل مع النص والمصفوفات والتاريخ والتعبيرات العادي ومعالجات الأساسية في DOM ، ولكن اللغة نفسها لا تتضمن أي I/O ، مثل الشبكات أو التخزين والاعتماد عليها عند البيئة المضيفة التي يتم تضمينها. في البداية فقط يتم تطبيق جانب العميل في متصفحات الويب ، يتم الآن تضمين محركات JavaScript في العديد من الأنواع الأخرى من البرامج المضيفة ، بما في ذلك الخادم في خوادم الويب وقواعد البيانات وفي البرامج غير التابعة للويب مثل معالجات النصوص وبرامج PDF ، وفي وقت التشغيل البيانات التي تجعل جافا سكريبت متاحة لكتابة تطبيقات الجوال وسطح المكتب ، بما في ذلك أدوات سطح المكتب.

على الرغم من وجود أوجه تشابه قوية بين جافا سكريبت وجافا ، بما في ذلك اسم اللغة والنحو والمكتبات القياسية المعنية ، فإن اللغتين متميزتان وتخالفان بشكل كبير في التصميم ؛ تأثرت JavaScript بلغة البرمجة مثل Self و Scheme . كانت لغة الجافاسكريبت موجهة للمبرمجين الهواة وغير المحترفين ، إلا أنه تزايد الاهتمام بها وجذب اهتمام مبرمجين محترفين بعد إضافتها لتقنيات جديدة كإنتشار تقنية AJAX التي أدت إلى سرعة في التفاعل بين الخادم والعميل.

تستخدم لغة الجافاسكريبت أسلوب البرمجة الكائنية (OPP) الذي يسمح بالتعامل مع البيانات على شكل كائنات حيث كل كائن يملك مجموعة من الطرائق (Methods) والخصائص الخاصة بها، ويمكن لها أن تستجيب للأحداث أو أن تنفذ أمر معيناً عند حدوث شيء في صفحة HTML ، مثل انتهاء تحميل الصفحة أو عند النقر على عنصر ما ، وهي قادرة على كتابة عناصر HTML ، وتغيير محتوى هذه الصفحات ، قادرة على التحقق من البيانات في النماذج قبل اعتمادها . وذلك يوفر الكثير من الوقت اللازم لمعالجتها في المخدمات Servers.

2-2-مكتبات وأطر الجافا سكريبت :

من المهم جداً أن تعلم أن المكتبات Libraries تختلف عن أطر العمل Frameworks في طريقة بناء المشاريع وهذا توضيح بسيط عنها:

المكتبات Libraries : هي مجموعة من الإجراءات البرمجية المستقلة التي يمكن استدعاؤها من قبل البرامج لتنفيذ وظائف تكمل عملها وتقوم بتوفير الوقت حيث انك تستطيع استخدامها مباشرة بصفحات مشروعك حيث ان هذه المكتبات

توفر أدوات ووظائف تم برمجتها مسبقاً أشهر على المكتبات مكتبة React ، و مكتبة jquery من تطوير شركة facebook .

أطر العمل Frameworks : هي مجموعة من المكونات الجاهزة التي يستخدمها المبرمج في برنامج، حيث أن الهدف منها هو عمل نوع من الأدوات أو البنية التحتية الضرورية للمبرمج، بحيث لا يبدأ برنامجه من الصفر تم عمل هذه الأطر لتسهيل عملية تطوير التطبيقات وعلى عكس المكتبات لاستطاع استخدام الأدوات او الوظائف المبرمجة مسبقاً مباشرة ولكن يجب ان تتبع طريقة عمل الإطار inversion of control ويعود الفضل لإستخدام Framework Work Flow . من أشهر أطر العمل في عالم الجافاسكريبت AngularJS من تطوير شركة Google .

Meteorjs-1-2-2 :

هو إطار عمل جافا سكريبت مفتوح المصدر مشابه لبقية اطارات جافاسكريبت مكتوبًا باستخدام Node.js . ويسمح برنامج Meteor للنماذج الأولية السريعة بإنتاج كود عبر النظام الأساسي (Web و iOS و Android) . يتكامل مع MongoDB ويستخدم بروتوكول البيانات الموزعة ونمط publish-subscribe لنشر التغييرات البيانات إلى الزبائن تلقائيًا دون الحاجة لكتابية أي كود تزامن . يمكن استخدام Meteor بمحرك Blaze templating الخاص به ، وكذلك مع الأطر React أو Angular .

بعد أن تم تطويره لمدة 8 أشهر ، تم إصدار Meteor في البداية في ديسمبر 2011 تحت اسم Skybreak . بحلول أبريل 2012 ، تمت إعادة تسمية إطار Meteor وتم إطلاقه رسميًا . خلال الأشهر القليلة المقبلة ، وبمساعدة استثمارات كبيرة من Andreessen Horowitz وتأييد من شخصيات بارزة في عالم بدء التشغيل ، زادت Meteor باطراد قاعدة مستخدميها وأصبحت أكثر استخدامًا في موقع الإنتاج .

Jquery-2-2-2 :

هي مكتبة جافا سكريبت مصممة لتبسيط البرمجة النصية لـ HTML من جانب الزبائن . هو برنامج مجاني مفتوح المصدر يستخدم رخصة MIT المسموح بها . يشير تحليل الويب إلى أنها مكتبة JavaScript الأكثر انتشاراً .

تم تصميم بنية jQuery لتسهيل التنقل في مستند ، وتحديد عناصر DOM ، وإنشاء رسوم متحركة ، ومعالجة الأحداث ، وتطوير تطبيقات Ajax . يوفر jQuery أيضًا إمكانيات للمطورين لإنشاء مكونات إضافية أعلى مكتبة جافا سكريبت . يتيح ذلك للمطورين إنشاء تجريدات للتفاعل على مستوى منخفض رسوم متحركة ، وتأثيرات متقدمة وأدوات سهلة الاستخدام قابلة للتطبيق . يسمح الأسلوب المعياري لمكتبة jQuery بإنشاء صفحات ويب ديناميكية قوية وتطبيقات ويب .

تتضمن Microsoft ASP.NET هذا البرنامج مع Visual Studio لاستخدامه ضمن إطار Web MVC و ASP.NET AJAX و nokia في حين قامت بدمجها في النظام الأساسي لتطوير تطبيقات Run-Time .

مبادئ التطوير مع jQuery هي:

- فضل جافا سكريبت و HTML: توفر مكتبة jQuery ببناء جملة بسيطة لإضافة معالجات الأحداث إلى DOM باستخدام جافا سكريبت ، بدلاً من إضافة سمات حدث HTML لاستدعاء وظائف جافا سكريبت .

وبالتالي ، يشجع المطورين على فضل تعليمات JavaScript البرمجية تماماً عن ترميز HTML .

- الاختصار والوضوح: يعزز jQuery الإيجاز والوضوح مع ميزات مثل الوظائف المتسلسلة وأسماء الوظائف المختصرة.

التخلص من حالات عدم التوافق بين المتصفحات: تختلف محركات جافا سكريبت لمتصفحات مختلفة قليلاً حتى لا تعمل شفرة جافا سكريبت التي تعمل في متصفح واحد مع متصفح آخر. مثل أدوات JavaScript الأخرى ، يعالج jQuery كل هذه التناقضات عبر المتصفحات ويوفّرواجهة متسقة تعمل عبر المتصفحات المختلفة.

- القابلية للتوسّع: يمكن إضافة أحداث وعناصر وطرق جديدة بسهولة ثم إعادة استخدامها كمكونات إضافي.

: Vue.js-3-2-2

إطار جافا سكريبت مفتوح المصدر لبناء واجهات المستخدم تم تصميمه لتنظيم وتبسيط تطوير الويب، أصبح التكامل مع المشاريع التي تستخدم مكتبات جافا سكريبت الأخرى أمرًا سهلاً مع Vue نظرًا لأنّه مصمم ليتم تبنيها بشكل متزامن. يمكن أن تعمل Vue أيضًا كإطار تطبيقات الويب قادر على تشغيل التطبيقات المتقدمة ذات الصفحة الواحدة.

يركز المشروع على صنع الأفكار في تطوير واجهة المستخدم على شبكة الإنترنت (المكونات ، واجهة المستخدم التقريرية ، إعادة التصحيح السريع ، تصحيح أخطاء الوقت ، إلخ). فهي تحاول أن تكون أقل بروبر وبالتالي يسهل على مطوري البرامج التقاطها.

يتميز بهندسة معمارية تدريجية ، تكرر المكتبة الأساسية على العرض التقديمي وتكون المكونات ، ويمكن تضمينها في الصفحات الحالية. توفر الميزات المتقدمة المطلوبة للتطبيقات المعقدة مثل التوجيه وإدارة الحالة وأدوات البناء من خلال المكتبات والحزام الداعمة الرسمية.

: Backbone.js-4-2-2

مكتبة جافا سكريبت مزودة بواجهة RESTful JSON وهي مبنية على نموذج تصميم تطبيق – Model-View-Presenter (MVP) . من المعروف أنه العمود الفقري لكونه خفيف الوزن ، حيث أن اعتماده الثابت الوحيد موجود على مكتبة جافا سكريبت واحدة Underscore.js ، بالإضافة إلىjQuery لاستخدام المكتبة الكاملة. وهو مصمم لتنمية تطبيقات الويب ذات الصفحة الواحدة ، وللحافظة على أجزاء مختلفة من تطبيقات الويب (مثل الزوائن المتعددين والخادم) متزامنة. تم إنشاء العمود الفقري بواسطة Jeremy Ashkenas ، المعروف أيضًا باسم CoffeeScript و Underscore.js .

عند التعامل مع DOM ، يتبع أسلوب برمجة إلزامي ، على النقيض من أسلوب البرمجة التعريفية (شائع في AngularJS باستخدام سمات البيانات).

إن محاولة توفير الحد الأدنى من تركيبة البيانات (النماذج والمجموعات) وواجهة المستخدم (طرق العرض وعناوين URL) ، يترك للمطور اختيار الإضافات للوظائف المحسنة. على سبيل المثال ، يمكن للمرء استخدام طرق العرض المتداخلة باستخدام Backbone Layout Manager أو ربط عرض النموذج مع ReSTbasis .

: Breeze.js-5-2-2

هي مكتبة جافا سكريبت تساعدنا في إدارة البيانات في تطبيقات انزيون الغنية . يعمل Breeze أساساً في كل متصفح ويدعم الاستعلام من جانب الزيون والتخزين المؤقت والرسوم البيانية للكائنات الديناميكية.

أفضل شيء في Breeze.js هو أنه لا يعكس نموذج الخادم من جانب ، ولكنها ينشئه ديناميكياً. مع Breeze ، تكون البيانات المخزنة مؤتمناً على جانب الزيون . لا تحتاج إلى الاستعلام عن الخادم لأنه يمكنه الاستعلام عن ذاكرة التخزين المؤقت بدلاً من ذلك. أنه يحفظ ذاكرة التخزين المؤقت محلياً وغير متصل. عندما يتم إعادة توصيله ، فإنه يقوم بموازنة التغييرات. اثنان من أقوى النقاط في Breeze.js هي الاستعلامات الغنية وتغيير التتبع. أربع طرق قوية للاستعلام هي الفلتر ، الفرز ، الترجيل والتوقعات. يحتاج طلب البحث إلى مساعدة لتنفيذ ذلك حيث يأتي Breeze EntityManager بشكل جيد مع الكثير من الإطارات بما في ذلك AngularJS و Knockout و Backbone.js و Node.js وغيرها الكثير .

Angular JS-6-2-2

ويشار إليه عادة باسم "Angular.js" أو "Angular 1.X" هو إطار تطبيق وبفتح المصدر يستند إلى جافا سكريبت وتحتفظ به غوغل ومجتمع من الأفراد والشركات للتصدي للعديد من التحديات التي واجهتها في تطوير تطبيقات صفحة واحدة. تكمل مكونات جافا سكريبت Apache Cordova ، إطار عمل يستخدم لتطوير تطبيقات المحمول عبر الأنظمة الأساسية. ويهدف إلى تبسيط تطوير واختبار مثل هذه التطبيقات من خلال توفير إطار للتحكم في جانب الزيون model-view-view- (MVC) وبنيات عرض نموذج الرؤية (MVVM) controller ، إلى جانب المكونات الشائعة الاستخدام في تطبيقات الإنترنت الغنية. في عام 2014 ، بدأ فريق AngularJS الأصلي العمل على Angular (منصة التطبيق).

بنيت AngularJS على الاعتقاد بأنه يجب استخدام البرمجة التعريفية لإنشاء واجهات المستخدم وتوصيل مكونات البرامج ، في حين أن البرمجة الضرورية تتلاءم بشكل أفضل مع تعريف منطق عمل التطبيق. يعمل الإطار على تكيف وتوسيع HTML التقليدي لتقديم محتوى ديناميكي من خلال ربط البيانات ثنائي الاتجاه الذي يسمح بالموازنة التلقائية للنماذج وطرق العرض. ونتيجة لذلك ، يؤكد AngularJS على التلاعب الواضح في DOM بهدف تحسين القابلية للاختبار والأداء.

تضمن أهداف تصميم AngularJS ما يلي:

- لفك مناولة DOM من منطق التطبيق. تتأثر صعوبة هذا بشكل كبير من طريقة هيكلة الكود.
- لنفصل جانب الزيون من أحد التطبيقات من جانب الخادم ، وهذا يسمح للعمل التنموي بالتنمية بالتوازي ، ويسمح بإعادة استخدام كل الجانبيين.
- لتوفير هيكل لرحلة بناء التطبيق: من تصميم واجهة المستخدم ، من خلال كتابة منطق العمل إلى الاختبار.

تطبيق MVC لنفصل العرض التقديمي والبيانات والمكونات المنطقية باستخدام dependency ، تقدم Angular injection الخدمات التقليدية للخادم ، مثل وحدات التحكم المعتمدة على العرض ، إلى تطبيقات الويب من جانب الزيون وبالتالي يمكن تقليل الكثير من العبء على الخادم.

تم تطوير AngularJS في الأصل في عام 2009 بواسطة Miško Hevery في Brat Tech LLC كبرنامج وراء خدمة تخزين JSON على الإنترنت والتي كان يمكن تسعيرها من قبل ميجابايت للتطبيقات سهلة الصنع للمشروع.

يقع هذا المشروع على نطاق الويب "GetAngular.com" ، وكان لديه عدد قليل من المشتركين ، قبل أن يقرر الاثنان التخلص عن فكرة العمل والإفراج عن Angular كمكتوب مفتوح المصدر.

أضاف الإصدار 1.6 العديد من مفاهيم AngularJS إلى Angular ، بما في ذلك مفهوم بنية التطبيقات القائمة على المكونات. قام هذا الإصدار من بين آخرين بإزالة Sandbox ، والتي يعتقد العديد من المطوروين أنها توفر حماية إضافية ، على الرغم من العديد من نقاط الضعف التي تم اكتشافها والتي تجاوزت وضع الحماية.

في يناير 2018 ، تم الإعلان عن جدول زمني للتخلص التدريجي من AngularJS بعد الإفراج عن 1.7.0 ، سيستمر التطوير النشط في AngularJS حتى 30 يونيو 2018. بعد ذلك ، سيتم دعم 1.7 حتى 30 يونيو 2021 كدعم طويل الأجل.

(angular4 : AngularJS(2.1)-7-2-2

تم الإصدار في سبتمبر 2016. لقد كان عبارة عن إعادة كتابة كاملة والتي قدمت نموذجًا مبنيًا على المكونات تم إنشاؤه، باستخدام TypeScript (والذي تم تجميعه إلى الإرتكاك ، تم إصدار الإصدار 4.0 في مارس 2017 (تم تخطي v3 لتجنب مشكلات الإصدارات الدلالية).

يختلف 2 Angular بشكل جزئي عن الإصدار v1. لا يتوافق أي منها مع الآخر - ربما كان يجب على Google منح المشروع اسمًا مختلفًا!

الأيجابيات:

- حل واحد لإنتاج تطبيقات الويب الحديثة
- لا يزال جزءًا من مجموعة أدوات MEAN (MongoDB ، ExpressJS ، AngularJS,)
- على الرغم من توفر عدد أقل من البرامج التعليمية لـ NodeJS
- يوفر TypeScript بعض المزايا لأولئك المتألفين باللغات المكتوبة بشكل ثابت مثل C # و Java.

: ReactJS-8-2-2

هي مكتبة جافا سكريبت لبناء واجهات المستخدم. تم اصداره من قبل Facebook و Instagram و مجتمع من المطوروين والشركات الفردية.

React هي مكتبة لإنشاء واجهات مستخدم قابلة للتركيب. يشجع على إنشاء مكونات واجهة مستخدم قابلة لإعادة الاستخدام ، والتي تعرض البيانات التي تتغير بمرور الوقت. يستخدم الكثير من الأشخاص React باعتباره MVC في V. وتقديم نموذج برمجة أبسط وأداء أفضل. يمكن أيضًا عرض التفاعل على الخادم باستخدام عقدة ، ويمكنه تشغيل التطبيقات الأصلية باستخدام React Native. يعمل نظام React على تنفيذ تدفق البيانات التفاعلية أحادية الاتجاه ، مما يقلل من الحد القياسي وأسهل في التفكير أكثر من ربط البيانات التقليدية.

:react-1-8-2-2

- هو امتداد لامتداد جافا سكريبت. ليس من الضروري استخدام JSX في تطوير React ، ولكن يوصى بذلك.

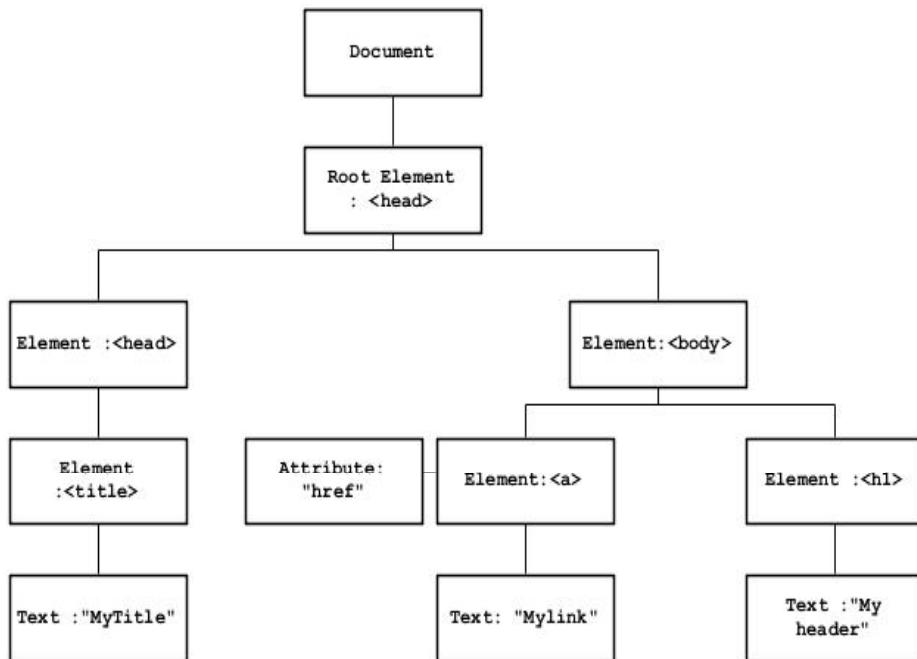
- مكونات React components : هو كل شيء عن المكونات. عليك أن تفك في كل شيء كمكون. سيساعدك هذا في الحفاظ على الرمز عند العمل على مشروع أكبر حجمًا.
- تدفق البيانات أحادي الاتجاه و Flux - يعمل React على تنفيذ تدفق البيانات في اتجاه واحد مما يجعل من السهل التفكير في تطبيقك. Flux هو نمط يساعد في الحفاظ على البيانات الخاصة بك أحادي الاتجاه.
- الترخيص - تم ترخيص React تحت شركة Facebook Inc . وتم ترخيصه بموجب CC BY 4.0.
- يستخدم DOM الظاهري الذي هو كائن جافا سكريبت. سيؤدي ذلك إلى تحسين أداء التطبيقات ، نظرًا لأن DOM Virtual DOM يكون أسرع من DOM المعتاد.
- يمكن استخدامه على جانب الزيون والخادم وكذلك مع الأطر الأخرى.
- تعمل أنماط المكونات والبيانات على تحسين إمكانية القراءة ، مما يساعد في الحفاظ على التطبيقات الأكبر حجمًا.

: react 2-8-2-2 قيود

- لا يغطي سوى طبقة العرض في التطبيق ، وبالتالي لا تزال بحاجة إلى اختيار تقنيات أخرى للحصول على مجموعة كاملة من الأدوات للتطوير.
- يستعمل تմددات مضمونة و JSX ، والتي قد تبدو مربكة لبعض المطورين.

3-نموذج كائنات الوثيقة : DOM

تعرف بـ Document Object Model عندما يتم تحميل صفحة الويب يقوم المتصفح بإنشاء نموذج كائن المستند الخاص بالصفحة .
يتم إنشاء نموذج HTML DOM كشجرة من الكائنات :



.الشكل (2) شجرة كائنات الـ DOM

باستخدام نموذج DOM ،تحصل جافا سكريبت على جميع القدرات اللازمة لإنشاء HTML ديناميكي :
 تستطيع جافاسكريبت تغيير جميع عناصر HTML وسمات هذه العناصر في الصفحة، وتمتلك القدرة على إزالة هذه العناصر والسمات الموجودة ،إضافة عناصر وسمات HTML جديدة ،ولها القدرة على تغيير جميع أنماط CSS في الصفحة.
 يمكن أن تتفاعل جافاسكريبت مع جميع أحداث HTML الموجودة في الصفحة، وتستطيع إنشاء أحداث HTML جديدة في الصفحة.

لتعرف على مفهوم الـ DOM :

؟ DOM ما هو مفهوم الـ

DOM هو معيار W3C (World Wide Web Consortium) يحدد معياراً للوصول إلى المستندات:

"يعتبر (DOM) W3C واجهة أساسية ومحايدة للغة تسمح للبرامج والبرامج النصية بالوصول والتحديث الديناميكي لمحظى وبنية ونط المستند." . يتم فصل معيار W3C DOM إلى 3 أجزاء مختلفة:

-1 Core DOM - نموذج قياسي لجميع أنواع المستندات.

-2 XML DOM - نموذج قياسي لمستندات XML

-3 HTML DOM - نموذج قياسي لمستند HTML

؟ HTML DOM ما هو مفهوم الـ

بعد HTML DOM نموذج كائن قياسي وواجهة برمجة لـ HTML يُعرف عناصر HTML كأغراض objects، وخاصّص جميع عناصر HTML، وطرق الوصول إلى جميع عناصر HTML والأحداث لجميع عناصر HTML. معنى آخر: يعد HTML DOM معيارًا لكيفية الحصول على عناصر HTML أو تغييرها أو إضافتها أو حذفها.

2-3-2- الفرق بين استخدام Virtual DOM و Real DOM :

العديد من تقنيات الويب أصبحت تستخدم Virtual DOM بدل من Real DOM وذلك لعدة ميزات تتمتع بها في البداية ما هو virtual DOM ؟

هو كائن جافا سكريبت خفيفة الوزن ، وهو في الأصل مجرد نسخة من DOM الحقيقي. وهي عبارة عن شجرة عقدة تسرد العناصر وسماتها ومحفوظاتها مثل الكائنات وخاصّصها. تقوم بإنشاء شجرة عقدة من مكونات الصفحة ثم يقوم بتحديث هذه الشجرة استجابة للطفرات في نموذج البيانات الذي يحدث بسبب الإجراءات المختلفة التي يقوم بها المستخدم أو النظام.

- يعمل في ثلاث خطوات بسيطة:
 - عند تغيير أي بيانات أساسية ، يتم إعادة عرض واجهة المستخدم بالكامل في تمثيل DOM الظاهري.
 - ثم يتم حساب الفرق بين تمثيل DOM السابق والواحد الجديد.
 - بمجرد الانتهاء من العمليات الحساسية ، سيتم تحديث DOM الحقيقي فقط بالأشياء التي تغيرت بالفعل.
- والجدول التالي يوضح مقارنة بين استخدام النمطين التاليين :

Real DOM VS Virtual DOM	
Real DOM	Virtual DOM
يتم تحريره ببطء	يتم تحريره بشكل سريع
يتم تحديثه مباشرة	لا يتم تحديثه مباشرة
يتم إنشاء DOM جديد إذا تحدث العناصر	يتم تعديل JSX إذا تحدث العناصر
معالجة DOM مكلفة جدا	معالجة DOM سهلة جدا
يسبب الهدر في الذاكرة	لا يوجد هدر في الذاكرة

الجدول (2-1) مقارنة بين أنماط الـ DOM.

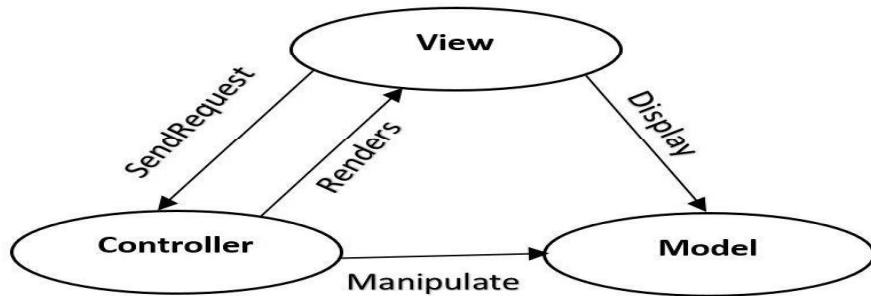
:(MVC)Model-View-Controller-4-2

هو نمط معماري يستخدم عادة لتطوير واجهات المستخدم التي تقسم التطبيق إلى ثلاثة أجزاء متراكبة . يتم ذلك لفصل التمثيلات الداخلية للمعلومات من طرق عرض المعلومات وقبوليها من المستخدم.

يفصل نمط تصميم MVC هذه المكونات الرئيسية مما يسمح بإعادة استخدام الكود بشكل فعال والتطوير المتوازي، تستخدم هذه المعمارية المستخدمة تدريجيًا لواجهات المستخدم الرسومية المكتوبة (GUIs).

تحتوي لغات البرمجة الشائعة مثل Java و C و PHP و Ruby وغيرها على إطار MVC الشائع الذي يتم استخدامها حالياً في تطوير تطبيقات الويب مباشرةً.

1-4-2 المكونات (Component):



. MVC نموذج (2-2) الشكل

Model - 1: النموذج هو المكون المركزي للنظام . يعبر عن سلوك التطبيق ، بغض النظر عن واجهة المستخدم، أي أنه يدير البيانات والمنطق وقواعد التطبيق مباشرة .

-2 **View** : العرض هو أي تمثيل خرج للمعلومات ، مثل مخطط أو رسم تخطيطي ، توفر طرق متعددة لعرض ذات المعلومات ، وكذلك تمكّنهم من تعديل البيانات.

Controller -3: وحدة التحكم ، يقبل الإدخالات ويقوم بتحويله إلى أوامر لطبقة النموذج أو طبقة العرض.

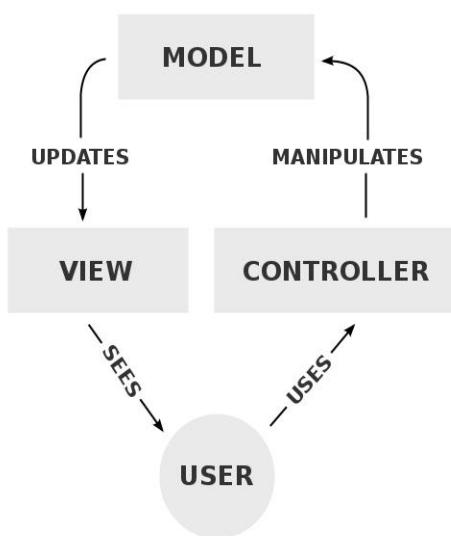
MVC Use in Web Applications-2-4-2

(استخدام الـ MVC مع تطبيقات الويب):

على الرغم من أن MVC تم تطويره في الأساس للحواسيب المكتبية ، لكنه اعتمد على نطاق واسع كمعمارية لتطبيقات الويب العالمية في لغات البرمجة الرئيسية ، تم إنشاء العديد من أطر الويب التي تفرض هذا النمط ، تختلف أطر البرامج هذه في تفسيراتها ، بشكل أساسي في الطريقة التي يتم بها تقسيم مسؤوليات MVC بين العميل والخادم .
تأخذ بعض أطر MVC للويب مقاربة للعميل مما يضع كل من Model و View و Controller على الخادم .
ويظهر هذا في أطر العمال مثل ASP.NET MVC و Rails و Django .

في هذا النهج ، يرسل العميل إما طلبات URL أو إرسال النماذج إلى وحدة التحكم ثم يتلقى صفحة ويب كاملة ومحدثة من العرض ؟ النموذج موجود بالكامل على الخادم.

أطر أخرى مثل AngularJS و Backbone.js و JavaScriptMVC و Ember.js و تسمح بتنفيذ مكونات MVC على جانب العميل .



3-4-2 : آلية عمل الـ MVC

باختصار نلاحظ أن المستخدم يقوم بإرسال بعض الأوامر إلى طبقة التحكم Controller والتي تقوم بمعالجة هذه الأوامر وإرسالها إلى طبقة النموذج Model والتي تقوم بإدارة البيانات واستدعاء الطائق المناسبة والتي تشكل العرض الجديد وترسله إلى طبقة العرض View التي بدورها تظهره إلى المستخدم .

. الشكل (3-2):آلية عمل الـ MVC .

4-4-2-محاسن ومساوي نموذج MVC : (MVC)Advantages MVC-1-4-4-2

Model التطوير المتزامن - يمكن لمطوري البرامج المستعدين العمل في الوقت نفسه على كل من طبقة Controller و View .

اقتران منخفض - إن طبيعة إطار MVC ذاته هي أن هناك اقتران منخفض بين النماذج أو العرض أو وحدات التحكم.

سهولة التعديل - بسبب الفصل بين المهام ، يكون التطوير أو التعديل المستقبلي أسهل.

طرق العرض المتعددة لطراز - يمكن أن تحتوي النماذج على طرق عرض متعددة.

2-4-4-2:(MVC)Disadvantages MVC-2-4-4-2

يمكن أن يكون التنقل في إطار العمل معقداً لأنه يقدم طبقات جديدة من التجريد ويطلب من المستخدمين التكيف مع معايير تحمل MVC .

يؤدي تفكيك إحدى السمات إلى ثلاث قطع إلى تشتت . وبالتالي ، تتطلب المطوروين للحفاظ على اتساق تمثيلات متعددة في وقت واحد.

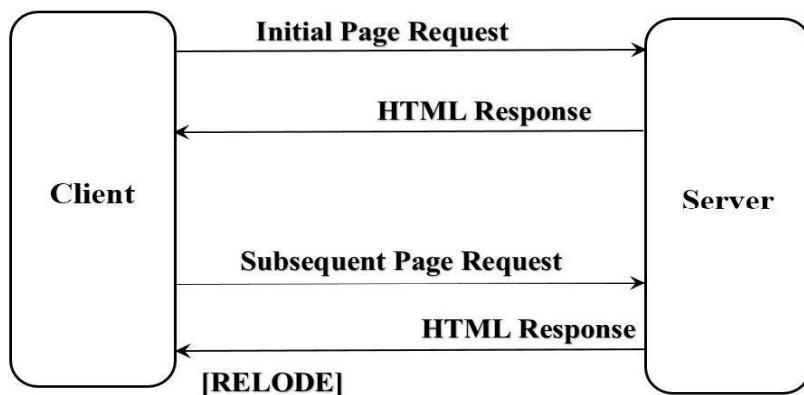
تصبح تعلم تقنيات متعددة هي القاعدة. يحتاج المطوروون الذين يستخدمون MVC إلى أن يكونوا ماهرين في تقنيات متعددة.

:Single Page Application -5-2

يتم تقسيم معظم الموقع إلى صفحات من أجل جعل المعلومات التي تحتوي عليها أسهل في الاستخدام .البني التقليدية هي إعطاء كل صفحة عنوان URL فريد . للانتقال إلى صفحة ، يرسل المتصفح طلب GET إلى عنوان URL للصفحة . سيرسل الخادم الصفحة وسيقوم المتصفح بتفريغ الصفحة الحالية وتحميل الصفحة الجديدة .

بالنسبة إلى متوسط اتصال الإنترنت ، من المحتمل أن تستغرق بضع ثوان ، حيث يجب على المستخدم الانتظار حتى يتم تحميل الصفحة الجديدة .

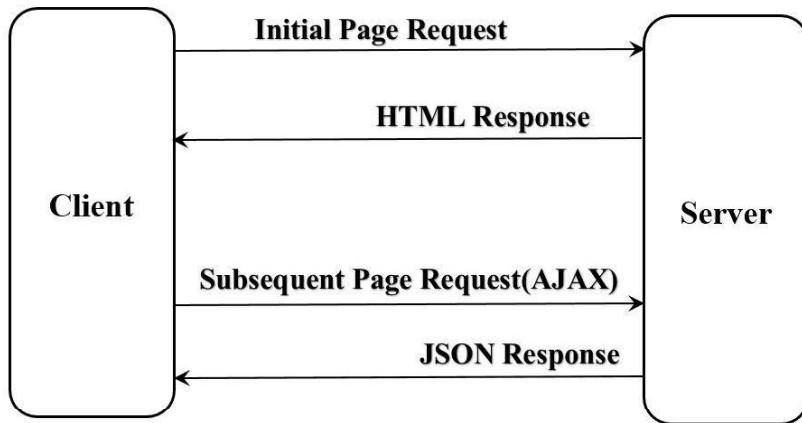
الشكل الآتي يوضح آلية طلب الصفحات من الخادم دون استخدام تقنية SPA .



الشكل (4-2) آلية طلب صفحة من خادم .

باستخدام واجهات برمجة تطبيقات جافا سكريبت والويب مثل XMLHttpRequest ، هناك نموذج مختلف ممكن: يمكن للمتصفح تحميل صفحة مبدئية ، ولكن الانتقال إلى صفحات جديدة لن يتطلب من المتصفح إلغاء تحميل الصفحة وتحميل صفحة جديدة . بدلاً من ذلك ، يمكن تحميل محتوى الصفحة من API بشكل غير متزامن مع JavaScript كتابته في الصفحة الحالية باستخدام .

من منظور المستخدم ، يبدو أن موقع الويب هذا يحتوي على صفحات تماماً مثل أي موقع ويب آخر ، ولكن من منظور تقني ، لا يحتوي هذا الموقع إلا على صفحة واحدة فقط وهو ما يعرف بـ SPA ، و الشكل الآتي يمثل آلية طلب صفحة من الخادم باستخدام الـ SPA .



. الشكل (5-2)آلية طلب صفحة من خادم باستخدام تقنية SPA .

لكي نستطيع تعريف تقنية SPA يجب علينا أن نعرف المفاهيم التالية :

تطبيق الويب هو برنامج حاسوب Client-Server يعمل عليه العميل (Client) مثل واجهة المستخدم في متصفح ويب وتشمل تطبيقات البريد الإلكتروني ومبيعات التجزئة على الإنترنت والمزادات على الإنترنت وخدمات المراسلة الفورية ومثال شهير هو موقع ويكيبيديا .

موقع الويب عبارة عن مجموعة من صفحات الويب ذات الصلة ، بما في ذلك محتوى الوسائط المتعددة وعادة ما يتم تحديده باستخدام اسم نطاق شائع (Domain name) ويتم نشره على خادم واحد على الأقل . يمكن الوصول إلى أحد مواقع الويب عبر شبكة بروتوكول إنترنت (IP) أو شبكة محلية خاصة (LAN) عن طريق الرجوع إلى محدد موقع المعلومات (URL) .

5-1-5-ما هي تقنية SPA ؟

يُعرف هذا المصطلح بأنه تطبيق ويب أو موقع ويب يتفاعل مع المستخدم عن طريق إعادة كتابة الصفحة الحالية ديناميكياً بدلاً من تحميل صفحات جديدة بالكامل من الخادم . يتجنب هذا الأسلوب مقاطعة تجربة المستخدم بين الصفحات المتتالية ، مما يجعل التطبيق أكثر شبهاً بتطبيق سطح المكتب .

في SPA، يتم جلب كل الشفرات الضرورية HTML و CSS و JavaScript اللازمة لتحميل صفحة واحدة أو الموارد اللازمة ديناميكياً وإضافتها إلى الصفحة عند الضرورة ، غالباً تكون هذه العملية استجابة لإجراء يقوم به المستخدم . الفرق الأكبر تميز بين موقع الويب العادي و SPA هو تقليل عدد تحديثات الصفحة . بمتلك SPA استخداماً أكبر لـAJAX طريقة للتواصل مع الخوادم الخلفية دون إجراء تحديث للصفحة الكاملة - لتحميل البيانات إلى التطبيق الخاص بنا . ونتيجة لذلك ، تحدث عملية تقديم الصفحات في الغالب على جانب العميل .

لقد بنت أطر جافا سكريبت مبادئ SPA، مثل ReactJS و AngularJS و EmberJS و VueJS . على الرغم من أن بناء وسائل SPA العصرية يعد أمراً عصرياً و يعتبر ممارسة تطويرية حديثة ، فمن المهم أن تكون مدرجاً لعواقبه ، بما في ذلك:

1- يقوم المتصفح بمعظم الرفع الثقيل ، مما يعني أن الأداء يمكن أن يكون مشكلة خاصة على الأجهزة المحمولة الأقل قدرة.

2- يجب التفكير بعناية في تحسين محرك البحث (SEO) بحيث يمكن اكتشاف محتواك بواسطة محركات البحث ومواقع الشبكات الاجتماعية التي توفر معاينة رابط.

قد تخفف هذه العيوب باستخدام جانب المخدم :

1- تعمل معظم أطر عمل جافا سكريبت الحديثة على طرق للتعامل مع عرض الخادم من SPA مما يعني أن المستخدم سيحصل على صفحة كاملة عند تحميل SPA للمرة الأولى ، بدلاً من رؤية مؤشر التحميل.

2- يمكن للعرض من جانب المخدم أن يخفف من بعض العبء على المتصفح الذي يجب أن يمر بها عند تقديم الصفحات ، كما أنه سيساعد في حل مشكلة تحسين محرك البحث (SEO) وقابلية اكتشاف المحتوى.

2-5-2- بني الخادم المستخدمة في SPA :Thin server architecture-1-2-5-2 (بنية الخادم الخفيفة):

ينقل SPA منطقياً من الخادم إلى العميل . هذا يؤدي إلى أن دور خادم الويب يتضور إلى واجهة برمجة تطبيقات خالصة أو خدمة ويب . هذا التحول المعماري ، في بعض الدوائر ، سلط الضوء على أن التعقيد قد تم نقله من الخادم إلى العميل ، مع الحجة بأن هذا يقلل في النهاية من تعقيد النظام بشكل عام .

2-2-5-2 Thick stateful server architecture (بنية الخادم السميكة) :

يحافظ الخادم بالحالات الضرورية في ذاكرة متصفح العميل الخاصة بصفحة معينة . وهكذا في حين طلب المستخدم إجراء معين من الخادم يرسل الخادم الرماز (CODE:HTML OR JavaScript) المناسبين مع التغييرات الملحوظة لغabar الحالة المطلوبة الجديدة

يحتاج هذا الأسلوب إلى مزيد من ذاكرة الخادم ومعالجته ، ولكن الميزة هي نموذج تطوير بسيط لأن:

- 1 عادةً ما يتم تشفير التطبيق بالكامل في الخادم .
- 2 تم مشاركة البيانات وحالة واجهة المستخدم في الخادم في نفس مساحة الذاكرة بدون تحتاج إلى جسور اتصال العميل / الخادم المخصصة.

3-2-5-2 Thick stateless server architecture (بنية الخادم الأقل سماكة):

ترسل صفحة العميل البيانات التي تمثل حالتها الحالية إلى المخدم ، عادة من خلال طلبات AJAX ، باستخدام هذه البيانات يمكن الخادم من إعادة بناء حالة العميل لجزء من الصفحة التي تحتاج إلى تعديل ويمكنها إنشاء البيانات أو الشفرة الضريبة والتي يتم إرجاعها إلى العميل لإحضار الصفحة إلى حالة جديدة ، عادةً ما يتم تعديل شجرة صفحات DOM وفقاً لإجراءات العميل عليها .

يتطلب هذا الأسلوب إرسال المزيد من البيانات إلى المخدم وقد تتطلب موارد حسابية إضافية لكل طلب لإعادة إنشاء حالة صفحة العميل بشكل جزئي أو كامل في المخدم.

ويمكن بسهولة توسيع هذا النهج نظراً لعدم وجود بيانات صفحة لكل عميل محفوظة في الخادم ، وبالتالي ، يمكن إرسال طلبات Ajax إلى عقد خادم مختلفة دون الحاجة إلى مشاركة بيانات الجلسة أو تقارب الخادم.

الفصل الثالث

تنفيذ تطبيقات على React و Angular

3-1-تنفيذ معرض صور على React

معرض صور بخطوات بسيطة تم عمله بمساعدة من بيئة node.js التي نفذت الـ React من جانب الخادم ، تقدم مكاتب الـ React الكثير من المكونات الجاهزة التي تساعدهك على بناء أي شيء وهذا مما سهل العمل كثيراً وقد يوفر جهد كبير على المبرمجين أو مصممي موقع الويب .

3-1-1-تنصيب بيئة Node.js

أولاً يجب تحميل برنامج node.js يجب أن نحدد النسخة التي تتوافق مع نظام التشغيل الخاص بالجهاز المراد العمل عليه ، قد نرى عدة نسخ ونختار الإصدار الذي نريده أو يتناسب مع مواصفات الجهاز .(انظر الملحق A لطريقة التحميل والتنصيب) .

3-1-2-تنصيب مكونات React.js على Node.js

أولاً نقوم بتشغيل موجه الأوامر **CMD** ثم ندخل الأمر **start npm** ثم ندخل الأمر **npm** الآن لكي نستطيع تشغيل الـ **React** نحتاج لتحميل بعض المكونات عن طريق **npm install -g create-react-app** الأمر التالي :

ننتظر حتى ينتهي من تنزيل المكاتب وهكذا تكون قد نصبنا مكونات **React.js** .

3-1-3-إنشاء مشروع React جديد

أولاً نقوم بتشغيل موجه الأوامر **CMD** ثم ندخل الأمر **create-react-app my-app** يبدأ موجه الأوامر بتحميل المكتبات اللازمة وينشئ مشروع جديد لتشغيله ندخل الأوامر التالية

cd my-app

npm start

ثانياً : لبدء العمل نجد داخل المجلد **my-app** عدة مجلدات نختار مجلد **Src** الذي يحتوي ملفات الجافا سكريبت **css** يمكننا أن نبدأ التعديلات على الملف **App.js** ونلاحظ تغير مباشر على صفحة الويب دون إعادة تحميلها .

3-1-4-برمجة معرض الصور

نحتاج إلى مكون هو مكون معرض الصورة وهو مكون جاهز تقدمه **React** ليس عليك سوى أن تختار الخصائص التي تريده أن يقدمها كعرض الفيديو أو الصور أو عرض الشرائح بقيم زمنية يحددها المستخدم . نقوم بضم مكتاب اللازم ثم نبني هذه المكونات ولنبع الخطوات التالية :

1- فتح الملف **App.js** الذي سنقوم بالتعديلاته عليه .

2- نضمن المكتبة التالية في ترويسة الملف على الشكل التالي :

```
import ImageGallery from './ImageGallery.jsx'
```

3- ثم نبني غرض منه في الدالة render() ونضع الصفات كما في الكود التالي :

```
class App extends React.Component {
  constructor() {
    super();
    this.state = {
      showThumbnails: true,
      showFullscreenButton: true,
      showGalleryFullscreenButton: true,
      showNav: true,
    };
  }

  _onImageClick(event) {
    console.debug('clicked on image', event.target, 'at index', this._imageGallery.getCurrentIndex());
  }

  _onImageLoad(event) {
    console.debug('loaded image', event.target.src);
  }

  _onScreenChange(fullScreenElement) {
    console.debug('isFullScreen?', !!fullScreenElement);
  }
}
```

```

render() {
  return (
    <body>

      <ImageGallery
        ref={i => this._imageGallery = i}
        items={images}
        onClick={this._onImageClick.bind(this)}
        onImageLoad={this._onImageLoad}
        onScreenChange={this._onScreenChange.bind(this)}
        showThumbnails={this.state.showThumbnails}
        showNav={this.state.showNav}
        showFullscreenButton={false}
        thumbnailPosition= 'bottom'
        additionalClass="Gallery"
      />
    </body>
  ) ;
}

export default App;

```

- نقوم بتشغيل المشروع عن طريق الأمر التالي : `npm start` ضمن مسار المجلد من موجه الأوامر

ويكون التنفيذ على الشكل التالي :



الشكل (1-3) عرض صور بتقنية الـ React

3-2-تنفيذ معرض صور على :Angular

أحد الأهداف الرئيسية لـ AngularJS هو توفير امتداد لـ HTML . هذا يتم إجراؤها باستخدام علامات ng التي تم استخدامها داخل ملف HTML لربط عرض البيانات بمساعدة وحدات التحكم . وحدات التحكم يتم كتابتها في ملف جافاسكريبت وتحفظ بكلّة الوظائف من أجل التلاعّب بالنسوج .

إنشاء شريط تمرير الصور مع angular.js و css3 . إن شريط التمرير نفسه ليس معقداً للغاية و لكنه سيكون له تأثير ثلاثي الأبعاد فريد عندما نحول الشرائط . إنشاء المعرض يستخدم AngularJs v1.2 و آلية HTML الإطار في إنشاء ترميز شرائط التدوير . سيكون لدينا زرين لتحويل الشرائط إلى الأمام و الخلف ، بالإضافة إلى لوحة صغيرة بها صور صغيرة للتبديل فوراً إلى الشريحة المطلوبة .
لدينا ثلاثة خطوات :

3-1- الخطوة الأولى (تصميم ملف HTML) :

في ملف الـ HTML . تهيئه الأساسيات التي يقوم عليها المشروع . نبدأ أولاً بـ ng-app الذي يعرّف التطبيق ويدونه لن يعمل التطبيق ، و بالإضافة إلى ذلك يمكننا من استناد قيمة ، هذه القيمة تمثل اسم الـ modules الرئيسي للمشروع .
هذا هو الرأس العادي نوعاً ما (معلومات وصفية مختلفة وأنماط و مكتبات الجافاسكريبت) .

```
<html ng-app="example366">
<head>
  <meta charset="utf-8" />
  <meta name="author" content="Script Tutorials" />
  <title>Photo Gallery with AngularJS and CSS3 | Script Tutorials</title>
  <meta name="description" content="Photo Gallery with AngularJS and CSS3 - demo page">
  <meta http-equiv="X-UA-Compatible" content="IE=Edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
  user-scalable=no" />

  <!-- add styles -->
  <link href="style.css" rel="stylesheet" type="text/css" />

  <!-- add javascripts -->
  <script src="jquery-2.0.3.min.js"></script>
  <script src="angular.js"></script>
  <script src="angular-animate.min.js"></script>
  <script src="angular-touch.min.js"></script>
  <script src="app.js"></script>
</head>
```

لعرض البيانات ضمن طريقة العرض (`ng-controller` يتم تعريف المتحكم في نفس الملف وفي جزء الـ MVC) . كما انا استخدمنا تقنية `ng-repeat="photo in photos"` . ذلك لعرض صورة من مجموعة من الصور (أي الدوران) التي تم تحميلها في البداية و الموجودة في الشريط المصغر اسفل العرض . واستخدمنا تقنية الحدث `ng-click` و `ng-swipe-left` و `ng-swipe-right`.

```
<body ng-controller="MainCtrl">
  <header>
    <a href="C:\Users\Diyar\Desktop\images" class="logo"></a>
  </header>
  <!-- slider container -->
  <div class="container slider">
    <!-- enumerate all photos -->
    
    <!-- prev / next controls -->
    <a class="arrow prev" href="#" ng-click="showPrev()"></a>
    <a class="arrow next" href="#" ng-click="showNext()"></a>
    <!-- extra navigation controls -->
    <ul class="nav">
      <li ng-repeat="photo in photos" ng-class="{'active':isActive($index)}">
        
      </li>
    </ul>
  </div>
  <!--<footer>
    <h2>Photo Gallery with AngularJS and CSS3
      (<a href="https://www.script-tutorials.com/photo-gallery-with-angularjs-and-css3/">
        Back to original tutorial</a>)</h2>
    </footer> -->
    <!-- Only used for Script Tutorial's Demo site. Please ignore and remove. -->
    <!--<script src="ads.js" async></script> -->
</body>
```

3-2-2- الخطوة الثانية (تصميم ملف CSS):

في ملف CSS يتم تقسيم جميع الأنساط إلى ثلاثة أقسام : الأنماط لشريط التمرير و شرائحة ، و أنماط أسهم التنقل و أنماط شريط التنقل المصغر .

```
.arrow {  
    cursor: pointer;  
    display: block;  
    height: 64px;  
    margin-top: -35px;  
    outline: medium none;  
    position: absolute;  
    top: 50%;  
    width: 64px;  
    z-index: 5;  
}  
.arrow.prev {  
    background-image: url("images/prev.png");  
    left: 20px;  
    opacity: 0.2;  
    transition: all 0.2s linear 0s;  
}  
.arrow.next {  
    background-image: url("images/next.png");  
    opacity: 0.2;  
    right: 20px;  
    transition: all 0.2s linear 0s;  
}
```

```
.slider {  
    border: 15px solid #FFFFFF;  
    border-radius: 5px;  
    height: 500px;  
    margin: 20px auto;  
    position: relative;  
    width: 800px;  
  
    -webkit-perspective: 1000px;  
    -moz-perspective: 1000px;  
    -ms-perspective: 1000px;  
    -o-perspective: 1000px;  
    perspective: 1000px;  
  
    -webkit-transform-style: preserve-3d;  
    -moz-transform-style: preserve-3d;  
    -ms-transform-style: preserve-3d;  
    -o-transform-style: preserve-3d;  
    transform-style: preserve-3d;  
}  
.slide {  
    position: absolute;  
    top: 0;  
    left: 0;  
}  
  
.slide.ng-hide-add {  
    opacity: 1;  
  
    -webkit-transition: 1s linear all;  
    -moz-transition: 1s linear all;  
    -o-transition: 1s linear all;  
    transition: 1s linear all;  
  
    -webkit-transform: rotateX(50deg) rotateY(30deg);  
    -moz-transform: rotateX(50deg) rotateY(30deg);  
    -ms-transform: rotateX(50deg) rotateY(30deg);  
    -o-transform: rotateX(50deg) rotateY(30deg);  
    transform: rotateX(50deg) rotateY(30deg);  
  
    -webkit-transform-origin: right top 0;  
    -moz-transform-origin: right top 0;  
    -ms-transform-origin: right top 0;  
    -o-transform-origin: right top 0;  
    transform-origin: right top 0;  
}  
.slide.ng-hide-add.ng-hide-add-active {  
    opacity: 0;  
}
```

من أجل تحقيق تأثيرات الانتقال ثلاثية الأبعاد بين الشرائح استخدمنا تأثيرات التحويل لـ `css3` مع `rotateX` و `rotateY` و `rotate-3d` و `preserve-3d` للعنصر الرئيسي .

3-2-3- الخطوة الثالثة (تصميم ملف JavaScript) :

ملف الجافاسكريبت وهو وحدة التحكم بالتطبيق.

```
angular.module('example366', ['ngAnimate', 'ngTouch'])
.controller('MainCtrl', function ($scope) {
    // Set of Photos
    $scope.photos = [
        {src: 'http://farm9.staticflickr.com/8042/7918423710_e6dd168d7c_b.jpg', desc: 'Image 01'},
        {src: 'http://farm9.staticflickr.com/8449/7918424278_4835c85c7a_b.jpg', desc: 'Image 02'},
        {src: 'http://farm9.staticflickr.com/8457/7918424412_bb641455c7_b.jpg', desc: 'Image 03'},
        {src: 'http://farm9.staticflickr.com/8179/7918424842_c79f7e345c_b.jpg', desc: 'Image 04'},
        {src: 'http://farm9.staticflickr.com/8315/7918425138_b739f0df53_b.jpg', desc: 'Image 05'},
        {src: 'http://farm9.staticflickr.com/8461/7918425364_fe6753aa75_b.jpg', desc: 'Image 06'}
    ];
    // initial image index
    $scope._Index = 0;
    // if a current image is the same as requested image
    $scope.isActive = function (index) {
        return $scope._Index === index;
    };
    // show prev image
    $scope.showPrev = function () {
        $scope._Index = ($scope._Index > 0) ? --$scope._Index : $scope.photos.length - 1;
    };
    // show next image
    $scope.showNext = function () {
        $scope._Index = ($scope._Index < $scope.photos.length - 1) ? ++$scope._Index : 0;
    };
    // show a certain image
    $scope.showPhoto = function (index) {
        $scope._Index = index;
    };
});
```

المجال `scope` هو الجزء الذي يربط بين العرض في HTML و المتحكم في Javascript في المثلث `HTML` و `controller` نمر له كوسقط الغرض `$scope` . عند اضافة الخصائص إلى الغرض `$scope` في المتحكم ، العرض (HTML) تستطيع الوصول إلى هذه الخصائص ولكن ليس باستخدام الصریح `[$scope]` وانما باستخدام `propertyName` .

`$scope.photos` تعريف مصفوفة الصور (لكل صورة المسار و الاسم) `$scope._index` هو دليل الصورة .

دليل الصورة الظاهرة حالياً على العرض (إذا كانت الصورة الحالية هي نفسها المطلوبة)
حساب \$scope.isActive دلليل الصورة السابقة لإظهارها عند الضغط على السهم المخصص للحدث.
حساب \$scope.showPrev دلليل الصورة التالية لإظهارها عند الضغط على السهم المخصص للحدث.
\$scope.showPhoto تعني الصورة الظاهرة حالياً .



الشكل (2-3) معرض صور بـتقنية الـ Angular

الفصل الرابع

مقارنة بين REACTJS و ANGULARJS

1-4 مقدمة :

قد ذكرنا سابقاً أن لغة الجافا سكريبت تم برمجتها وتطويرها لعمل بشكل رئيسي من خلال المتصفح ، ولا يوجد موقع إلا ويستخدمها ، فالبعض يستخدمها للتحكم بالنماذج أو الحماية للتحقق من البيانات المدخلة بالنماذج والبعض الآخر يستخدمها بالإضافة مؤثرات حركية بالصفحات ويمكن استخدامها لزيادة سرعة تصفح الموقع .

وفي الآونة الأخيرة تم تطوير تقنيات تسمح باستخدام JavaScript كلغة خادم يتم معالجة أوامرها بالسيرفر وعرض النتيجة بالمتصفح وأيضاً امكانية الإتصال بقواعد البيانات وتخزين وجلب البيانات بكل سهولة بواسطة node.js مما مكن جافاسكريبت ان تكون لغة متصفحات وأيضاً سيرفرات لتميز على لغات الويب الأخرى .

React مكتبة Front-End تم تطويرها بواسطة Facebook ، ويتم استخدامه لمعالجة طبقة العرض لتطبيقات الويب وأجهزة الهاتف الذكية ، تسمح لنا بإنشاء مكونات قابلة لإعادة الاستخدام ، تتميز بشعبيتها ولها أساس قوي ومجتمع كبير .

Angular هو إطار تطبيق ويب مفتوح المصدر يستند إلى جافا سكريبت ، وتحتفظ به بشكلأساسي Google ومجتمع من الأفراد والشركات للتصدي للعديد من التحديات التي تواجهها في تطوير تطبيقات أحدادية الصفحة . هناك نسخة جديدة منها 2.0 والتي تم إصدارها في 2014 .

أثناء البحث عن خدمات الويب وتقنيات الويب ، لاحظنا أن غالبية المبرمجين والمهندسين متخصصون للتعرف على الميزات والاختلافات في كل من هذه التقنيات على شبكة الإنترنت . حيث أن كلاهما مشهور بنفس القدر ويواجه العديد من الأشخاص الصعوبة في الاختيار بين احدهما لتطوير تطبيقات الصفحة الواحدة .

2-4-المنهجية المتبعة في المقارنة :

نستخدم المنهج GQM تعرف بـ Goal-Questions-Metrics طريقة متدرجة وفعالة للقياس . يتضمن ثلاث خطوات :

1. تحديد أهداف القياس .

2. تحديد الأسئلة التي ستجيب عما إذا كانت الأهداف مستوفاة .

3. تحديد وقياس المقاييس للإجابة على الأسئلة .

فيما يلي نموذج GQM لهذا المشروع :

الهدف : اختيار إطار عمل جافا سكريبت يتوافق بشكل أفضل مع تطوير تطبيقات الويب .
الأسئلة :

1- ما هو إطار JavaScript الأكثر شعبية ومتعددة الأغراض ؟

المقاييس : عدد الأشخاص الذين يستخدمون التكنولوجيا المعنية .

مجموعات البيانات ذات الصلة: عدد من مستودع github ، وعدد من الموقع التي تستخدمها ، والعلامات على Stackoverflow .

2- أي منها لديه أداء أفضل؟

المقاييس: الوقت اللازم لتطوير الشفرة وبناء التطبيق ، الوقت المطلوب إنشاؤه .

مجموعات البيانات ذات الصلة: الطبيعة والبنية والأداء والمرنة.

3- ما هي التكلفة الفعلة فيما يتعلق بالجهود المطلوبة لاستخدامها وتنفيذها؟

المقاييس: الوقت اللازم للتعلم والاستخدام.

مجموعات البيانات ذات الصلة: سهلة الاستخدام ، ومنحنى التعلم ، الطبيعة و بناء .

من كل مقاييس ، قمنا بالبحث وجمع البيانات من المعلومات الأساسية والخبراء الرأي ، واختبار الأداء على حد سواء React و Angular . كما قمنا بجمع البيانات من عدد الأشخاص الذين يصلون إلى موقع الويب وعدد المطوريين الذين يستخدمون كل مكتبة. ثم نحن نقيم كل مقاييس استناداً إلى إحصائيات البيانات المجمعة.

3-4-المقارنة :

1-3-4- الدعم والتطوير:

Angularjs هي إطار جافا سكريبت ، مطورة ومدعومة من قبل شركة Google ، تستخدم إلا HTML وإن JS في عملها ، تعمل من جانب الزبائن ولكن مع الإصدارات المطورة عنها كـ Angular 2 والتي تعرف الآن بـ MVC تكون من جانب الخادم ، تعتمد في عملها على نمط Angular 4 **React** هي مكتبة جافاسكريبت مدعومة من قبل شركة facebook و فيما بعد دعمتها Instagram ، تستخدم في عملها JSX و Redux و Flex ، وتكون من جانب المخدم .

ما هي لغة الـ **jsx** ؟

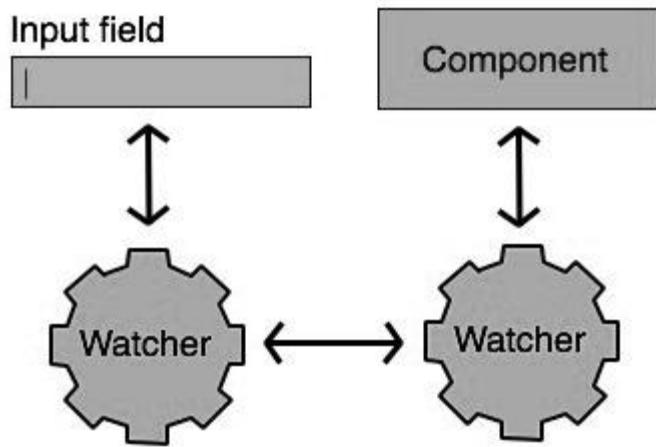
هي لغة برمجة (هي جافاسكريبت مع oop و فيما بعد خرجت عنها) مكتوبة بشكل ثابت و موجهة للكائنات و مصممة للتشغيل على المتصفحات الحديثة ومن مزاياها: easier و safer و faster سهلة و سريعة و آمنة .

2-3-4 : Data Binding

الترجمة اللغوية هو ربط البيانات و اصطلاحا هو الآلة التي يتم بها تنظيم ربط او تدفق البيانات بين نموذج البيانات وواجهة المستخدم user interface model .

1-2-3-4 : ربط البيانات بالنسبة لـ **Angular**

إن ربط البيانات بالنسبة لـ **Angular** هي bi-data binding أي ثنائي الاتجاه :
هذا النمط يتجلّى ببساطة في أن أي تغيير يطرأ على قيم نماذج البيانات data model يؤدي ضمنياً لتغيير واجهة المستخدم (تحديثها) و العكس صحيح اي ان المستخدم عندما يقوم بتغيير قيمة حقل ما input فإن ذلك يعني تغييراً لنموذج البيانات المقابل لهذا الحقل .

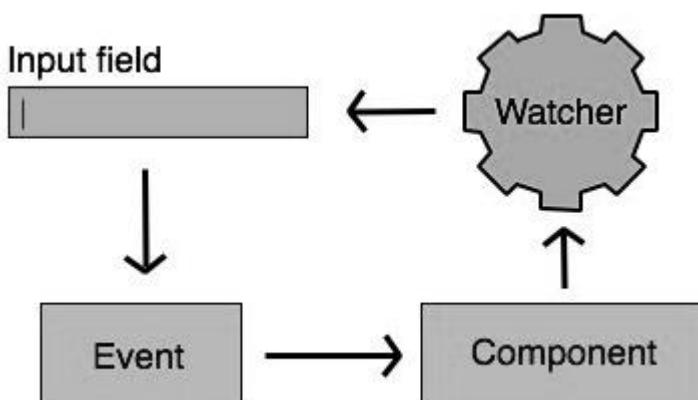


الشكل (4-1) مخطط يوضح طريقة ربط البيانات بالنسبة ل Angular

4-3-2-2- ربط البيانات بالنسبة ل React :

إن ربط البيانات ربط البيانات بالنسبة ل react هي uni-data binding في react تختلف الأمور لا تستطيع كما قلنا اجراء تحديقات على مستوى html يتم مراقبتها اوتوماتيكياً مع كود جافاسكريبت هنا لا يحدث في مكتبة react .

بكل تأكيد نستطيع اخذ قيمة الحقل وتحديث حالة المكون أو واجهة المستخدم و لكن الاختلاف ان العملية هنا ليست ضمنية أو تلقائية بل يجب أن تقوم بذلك بشكل صريح .الطريقة هي اننا نقوم بالاستماع لحدث الذي يتم اطلاقه عندما يتغير على قيمة الحقل input و بعد ذلك نقوم بتغيير حالة واجهة المستخدم بشكل يدوي عن طريق الواجهة البرمجية التي يتيحها لنا react .



الشكل (4-2) مخطط يوضح طريقة ربط البيانات بالنسبة ل React

٣-٤-٣-٤ نوع الـ DOM :

١-٣-٣-٤ Virtual DOM :

في React لكل غرض DOM يوجد غرض Virtual DOM مناظر له ، الغرض Virtual DOM هو تمثيل لغرض DOM (نسخة خفيفة الوزن) يحتوي على نفس خصائص غرض DOM الحقيقي .

التعامل مع Virtual DOM اسرع بكثير و بمجرد تحديث Virtual DOM فان React يقارن Virtual DOM مع لقطة DOM تم التقاطها مباشرة قبل التحديث .

من خلال مقارنة Virtual DOM الجديد مع اصدار ما قبل التحديث يستنتج React بالضبط اي كائنات DOM افتراضية قد تغيرت فعندئذ يقوم React بتحديث تلك الكائنات فقط على DOM الحقيقي و بالتالي يقوم React بتحديث الاجزاء الضرورية فقط من DOM ومن هنا تكمن أدائه العالي . بمعنى اخر : المنافس الأول لإطار العمل Angular في هذه الأيام هو مكتبة React.js التابعة كما ذكرنا لشركة فيسبوك . هذه المكتبة تم تصميمها لبناء واجهات

المستخدم UI وجاءت بأمر جديده ومهمة مثل طريقة كتابة أكواد HTML داخل جافاسكريبت والتي تسمى JSX ، وكذلك سرعتها الفائقة في تحديث DOM كونها تتبع فلسفة جديدة في التعاطي مع تغيرات الأخير ، حيث تعامل مع نسخة وهمة من DOM داخل الجافاسكريبت تسمى Virtual DOM تقوم React.js بتعديل عليه ثم بعد ذلك تقوم بنقل التغييرات وتطبيقها على DOM الحقيقي لضمان أداء أفضل وسرعة أكبر لواجهة المستخدم .

٢-٣-٣-٤ Regular DOM :

في Regular DOM يكون Angular في Regular DOM لا يوجد نسختين من عناصر DOM جديدة و قديمة فهناك فقط نسخة واحدة فبدلاً من ارسال اصدار جديد عنصر DOM الى اخر فقط يمكنك تحرير عنصر DOM الموجود مباشرة .

٤-٣-٤ License :

كلاهما مفتوح المصدر ولكنها ترخيصان مختلفان من حيث الاستخدام .

AngularJS تحت رخصة معهد ماساتشوستس للتكنولوجيا لذلك ، يمكن لأي شخص استخدامها لعرضه الخاص ، أو تعديل الكود ، أو تطوير كود جديد على أساس AngularJS . وعلاوة على ذلك ، يمكنهم استخدام AngularJS لأغراض تجارية في أي سوق العمل .

هناك اختلاف بسيط من ترخيص BSD الذي يستخدمه ReactJS . يمكن للمطربين أيضا الحصول على كل فئة في React أو تطوير البرنامج المساعد أو استخدامها مع إعلاناتهم التجارية الخاصة البرمجيات . ولكن هناك قاعدة واحدة تمكّنهم من استخدام هذه المكتبة كأي شيء يريدونه إذا كان كذلك ليس المنتج التنافسي لفيسبوك (صاحب المكتبة) .

٥-٣-٤ Nature and Structure :

AngularJS تمثل نفسها كإطار للتطوير تحت هيكل (MVC)Model-View-Controller . ReactJS هي المكتبة الوحيدة التي يمكن للمطور استيرادها ك "new class" واستخدامها في أي مكان .

و بالإضافة إلى ذلك تم تطوير AngularJS كحزمة كاملة من التعليمات البرمجية، تم تطوير كل فئة من جانب جوجل. من ناحية أخرى ، قام Facebook بتطوير البنية الأساسية لـ React .

ستقوم AngularJS بتحديث وإختيار إطار العمل بأكمله ولكن React ، مكتبات الطرف الثالث تقوم بتحديث مكتباتهم لدعم نسخة جديدة من مكتبة ReactJS الأساسية.

في هذه النقاط البيانات AngularJS هو هيكل أكثر تطوراً و API متوفقة.

4-3-6-المرنة (Flexibility) :

من الجزء الأخير ، الطبيعة والبنية ، AngularJS هو الإطار الكامل يجب أن يتم تنفيذها باستخدام هيكل Model-View-Controller (MVC) وهو صارم هيكل صعب. يجب على المطور اتباع القاعدة. ReactJS ليست سوى مكتبة ، المطور يقوم بإنشاء بنية خاصة لذلك فهي مرنة للغاية. ومع ذلك ، فإن Flowtype تدعم كل من ReactJS و Typescript ولكن ReactJS يدعم فقط النصي ..

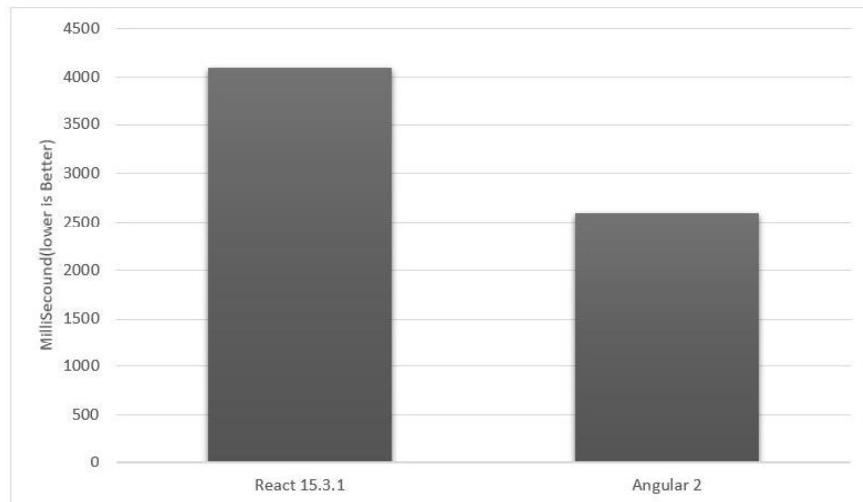
في هذه النقطة تعد React أفضل .

4-3-7-الاداء (Performance) :

لتحديد أي مكتبة لديها أداء أفضل ، نقوم بتشغيل المقياس brenchmark للعثور على سرعة تقديم تطبيق ويب قائمة المهام . مثلاً إنشاء ألف سطر و حذفها . المقياس يستخدم نفس نمط CSS ، لكن فقط الإطار Javascript المختلف . سنقوم بتنفيذ التطبيق على عدة متصفحات وتكون النتائج كالتالي :

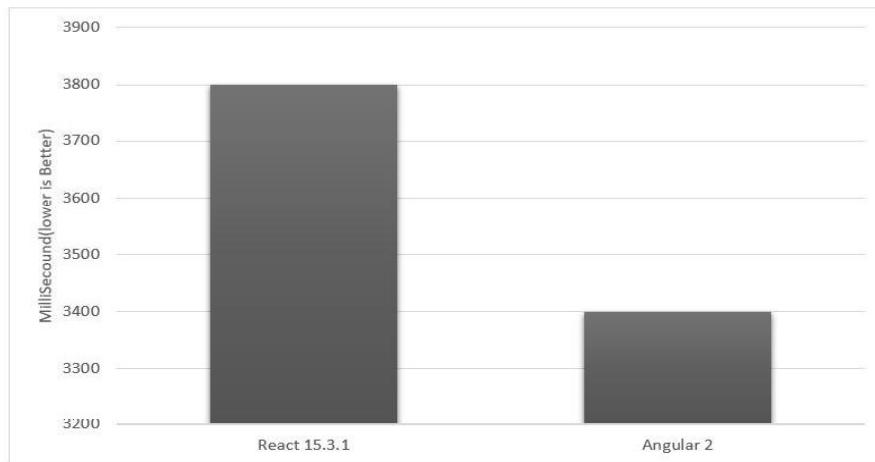
4-3-7-1-اختبار المتصفحات :

1- باستخدام المتصفح Chrome :



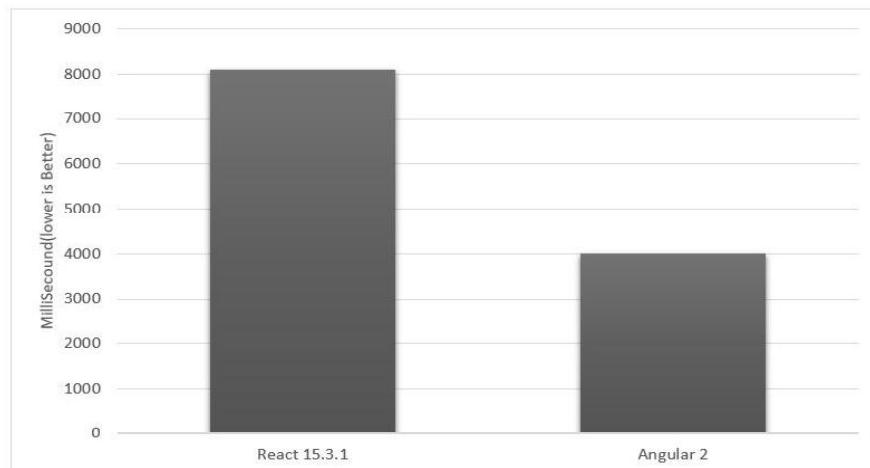
الشكل (4-3) مخطط يوضح زمن تنفيذ باستخدام متصفح Chrome

2- باستخدام متصفح : Safari



الشكل (4-4) مخطط يوضح زمن تنفيذ باستخدام متصفح Safari

3- باستخدام متصفح : Firefox



الشكل (4-5) مخطط يوضح زمن تنفيذ باستخدام متصفح Firefox

من النتيجة ، نرى أن Angular 2 أسرع في كل متصفح وتشغيله أسرع أكثر بمرتين على فايرفوكس. من بين الثلاث متصفحات .متوسط أوقات العرض :

Angular 2 is 3334.34 ms

React is 5454.67 ms

4-3-7-2- إحصائيات عدد المخازن على موقع GitHub :

يمكن أن يشير عدد مستودع البيانات على Github إلى عدد المطورين.

1. **watch** : من أجل المطور إذا كنت تريد الحصول على إشعار من هذا المخزن ، هناك معلومات جديدة مثل الإصدار الجديد.

2. **Star** : عندما يريد المطور التخزين على لوحة التحكم الخاصة به.

3. **fork** : من أجل المطورين الذين يرغبون في تنفيذ التعليمات البرمجية الأساسية على المخازن.

ولكن فقط عدد البيانات في المخازن لا يكفي لأن أعمار المخازن ليست متساوية. تمثل المخازن التي جاءت من قبل إلى الحصول على عدد أكبر من المطورين في كل البيانات.

Library Name	Date	Age
React	May 26 ,2013-May 2,2018	1437Days
Angular 2	Sep 14,2014-May 2, 2018	1326Days
Angular 1	Jan 3,2010- May 2, 2018	3041Days

الجدول(4-1) قيم كل من React و Angular و Angular

:May 2, 2018 ونرى أن بيانات كل مخزن بتاريخ

React -1 :



Angular2 -2 :



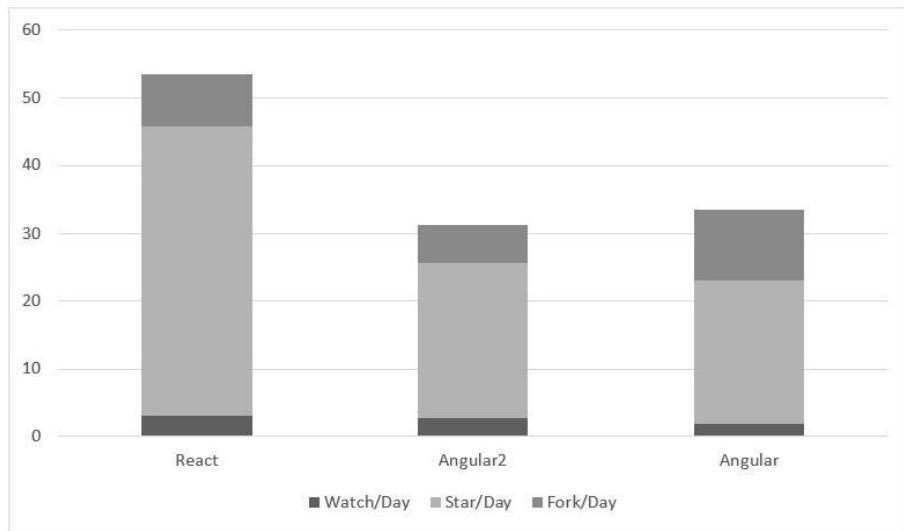
Angular -3 :



نحن تعتبر كل البيانات حسب اليوم لأنها ليست نفس العمر :

Library Name	Watch/Day	Star/Day	Fork/Day	Total
React	3.03	42.82	7.67	53.52
Angular 2	2.64	22.86	5.87	31.37
Angular 1	1.76	21.25	10.52	33.53

الجدول (4-2) إحصائيات ازدياد كل مخزن في يوم واحد



الشكل (4-4) معدل ازدياد كل من الأطر في يوم واحد

نلاحظ من الشكل (4-4) أن المعدل الوسطي لزدياد تقنية React هو أعلى قيمة في اليوم الواحد من كل إطار ، ونلاحظ أيضاً أن Angular أكثر شيوعاً من Angular 2 .

4-3-7-3-4- إحصائيات عدد المستخدمين على موقع الويب :

هذه هي أعلى موقع الويب JS استناداً إلى عدد عمليات الكشف عن طريق wappalyzer في آخر 7 أيام . (أعد بتاريخ 2, 2018) وبلغ إجمالي الوصول لأفضل 10 موقع (562,594) مرة .

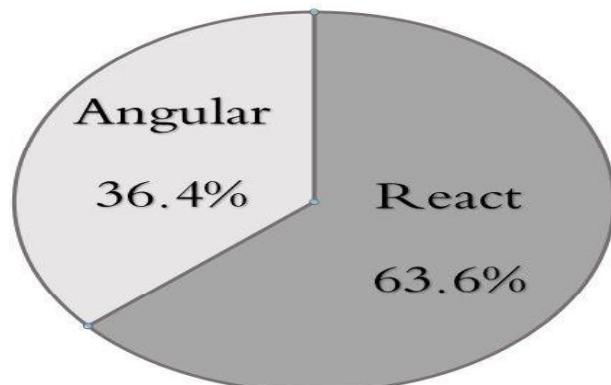
#	WEBSITE	DETECTIONS
1	<i>upwork.com</i>	197,038
2	<i>rpm.newrelic.com</i>	93,802
3	<i>freelancer.com</i>	74,174
4	<i>udemy.com</i>	44,657
5	<i>youtube.com</i>	38,009
6	<i>angular.io</i>	28,853
7	<i>web.telegram.org</i>	25,204
8	<i>ovh.com</i>	22,124
9	<i>docs.angularjs.org</i>	19,980
10	<i>istockphoto.com</i>	18,663

الجدول (3-4) أكثر عشرة موقع تستخدم تقنية الـ Angular

هذه هي أعلى مواقع الويب React JS استناداً إلى عدد عمليات الكشف عن طريق wappalyzer في آخر 7 أيام (أعد بتاريخ 2, 2018) . وبلغ إجمالي الوصول لأفضل 10 موقع (982,492) مرة.

#	WEBSITE	DETECTIONS
1	<i>web.whatsapp.com</i>	114,743
2	<i>imgur.com</i>	114,590
3	<i>imdb.com</i>	114,761
4	<i>dropbox.com</i>	114,044
5	<i>instagram.com</i>	105,252
6	<i>netflix.com</i>	86,932
7	<i>fiveww.com</i>	80,271
8	<i>facebook.com</i>	74,404
9	<i>app.hubspot.com</i>	68,753
10	<i>vimeo.com</i>	48,742

الجدول (4-4) أكثر عشرة موقع تستخدم تقنية الـ React



الشكل (4-5) معدل إجمالي الوصول لكلا التقنيتين

تظهر النتيجة أن React لديه مستخدم أكثر من 2 بمرتين ، على الرغم من أن Angular و React هي مكتبات جافا سكريبت الأكثر شعبية .

4-3-8-سهولة التعلم :

لأن Angular هو هيكل MVC هيكل صارم ويستخدم Typescript وهو يعتبر لغة جديدة وضفت لكتابه javascript لذلك هو أكثر صعوبة من React الذي هو فقط مكتبة view (واجهة) واستخدام صيغة جافا سكريبت العادية. بالإضافة إلى ذلك، عندما نحاول تصحيف بيانات في المقاييس الشعبية. وجدنا أن عدد العلامات في stackoverflow يتعارض مع عدد من البيانات في المخزن على github . لذلك ، فإننا نعتبر العدد الإجمالي وعدد العلامات في هذا الأسبوع لبحث معدل نمو كل المكتبات:

	Total Tag	Week	Growth Rate
React	28016	597	2.10%
Angular2	26731	870	3.30%
Angular	209728	1355	0.60%

الجدول (5-4) معدل نمو كل من التقنيات حسب موقع Stackoverflow

وجدنا أنه على الرغم من أن React لديه الكثير من المطوريين ولكن من الأسهل تعلمها وفهمها . لذلك إذا كان المشروع لا يحتوي على واجهة مستخدمة معقدة وللفريق هيكله الخاص ، أو أن الفريق لديه حدود الوقت لتعلم مكتبة جديدة. React هو الخيار الأفضل.

إذا كان المشروع كبير ولا يملك الفريق بنية خاصة به ، وهناك العديد من المطوريين الفريق ، واجهة المستخدم معقدة للغاية لذا فإن المشروع يتطلب بنية أكثر. في هذه الحالة Angular هو خيار أفضل.

3-4-الخلاصة :

	React	Angular
Auther	Facebook	Google
Type	JS Library	Framwork
Language	JSX	JS,HTML
Rendering	Server Side	Client Side
App Architecture	View layer only	MVC
Data-Binding	Uni- Directional	Bi-Directional
DOM	Virtual- Dom	Regular-Dom
Latest Version	15.5.4	1.6.4
Tooling	High	Low
Learning	Easy to learn	Requires more tool
Packaging	Strong	Weak
Performance	Above Avg	Faster
Size	151K	766K

الجدول (6-4) جدول يوضح المقارنة بين التقنيتين

الفصل الخامس

آفاق مستقبلية

1-5 آفاق مستقبلية:

تعد كل من Facebook و Google اثنين من أقوى الشركات في العالم ، وقد أنشأ كل منها إطار لبناء تطبيقات الويب. يبدو أن React و Angular على التوالي في معركة من أجل مستقبل الويب ، مع النقاش المستمر عبر الإنترنت واعتماد التطبيقات الكبيرة التي تواجه المستهلكين والتي يبدو أنها تميل بقوة إلى React في الوقت الحاضر. هل يسيطران بشكل جماعي على الواجهة الأمامية؟ هل React يقود حقباً؟

على الرغم من أن صفحات الويب التقليدية ، إلى حد كبير ثابتة ، لا تزال تحتل مكاناً مهماً ، إلا أن الجدول أصبح الآن نموذج الحواسيب السائد ، وقد أصبح مستخدمو الجوال يتوقعون تفاعل التطبيقات المحلية . لمحاولة مطابقة تجربة تطبيق أصلي ، لا يمكن عرض تطبيق الويب بالكامل على جانب الخادم ، يجب تغيير الصفحة ديناميكياً على الزبون . كلما زادت التغييرات كلما زادت الحاجة إلى تجريد أفضل من DOM (نموذج كائن المستند) لإدارة التعقيد . وقد أدى هذا إلى تزايد الاستخدام المتزايد لمكتبات وأطر عمل جافا سكريبت التابعة لجهات خارجية.

تاريخياً ، كانت jQuery أول مكتبة تحظى بشعبية كبيرة ، مما يتيح سهولة التعامل مع DOM على جانب الزبون . لا يزال الأكثر شعبية اليوم ، كمكتب الواجهة الأمامية الأساسية لـ 34٪ من مطوري الويب . ومع ذلك ، يتبيّن أن التلاعب في DOM يدوياً

تجريداً أفضل . بشكل عام ، لا يستخدم 12٪ من مطوري الويب أي نوع من أنواع الإطارات ، بينما 6٪ منهم كتبوا بأنفسهم . هذا يترك 48٪ من مطوري الويب يستخدمون حالياً إطار عمل خارجي غير jQuery كطريقة رئيسية للقيام بتطوير الويب الأمامي . بينما تمثل React و Angular نسبة 30٪ من جميع الاستخدام ، تاركة وراءها جميع الآخرين . في الواقع ، يعد تطوير الويب الأمامي بمثابة مساحة مجرأة لا تحتوي على أي مكتبة أو إطار عمل واحد لأكثر من 2٪ من الاستخدام الأساسي . لهذا فإن React و Angular يؤديان بالتأكيد إلى أطر أخرى ، على الرغم من أن ما يقرب من نصف مطوري الويب قد احتضناها تماماً أي إطار عمل لتطبيق صفحة واحدة حتى الان

React هو المفضل من قبل المتخصصين في الواجهة الأمامية:

لمعرفة كيف أن الواقع في السوق قد يكون مختلفاً ركة عبر الإنترنت وحتى استطلاعاً ، من المفيد أن ننظر إلى انهيار مكتبة جافا سكريبت واستخدام الإطار بواسطة لغة البرمجة الأساسية. إذا نظرنا فقط إلى مستخدميأحدث إصدارات جافا سكريبت - أولئك الذين يرغبون في البقاء في طليعة وأكثر عرضة لإيجاد خيارات إطار مناقشة على الإنترنت - نرى React هو الإطار الأساسي لـ 27٪ منهم. ومن بين أولئك الذين أجروا التحويل إلى ESNext (أي إصدار 2015 من جافا سكريبت القياسي أو أحدث) ، الذين يستخدمون أدوات لتحويل كودهم إلى جافا سكريبت التي تدعمها على نطاق واسع في المتصفحات (والمعروفة باسم ES5 ، وعرض في عام 2009) ، تستخدم React أكثر من كل الإصدارين من Angular combined. ومع ذلك ، فهذه هي المجموعة الوحيدة من المطوروين الذين يتغلب فيها React على أي إصدار من Angular وحده. إن مستخدمي جافا سكريبت الذين يتطلعون

إلى المستقبل هم أقل من نصف مستخدمي جافا سكريبت الذين يستخدمون جافا سكريبت في المقام الأول ، و 16٪ فقط من جميع مطوري الويب (الذين يستخدمون جميعا تقريبا JavaScript). لا يزال 18٪ آخرين من مطوري الويب يستخدمون ES5 بشكل أساسى. لا يزال عدد أكبر منها يستخدم حاليا Angular 1.x (21٪) كإطار عمل أساسى لها من x.2٪ (9٪) و React (8٪) مجتمعين. يعلم هؤلاء المطوريين على ما يعرفونه ويغلوونه منتجأً. قد يكونون متابعين للمعايير والأطر الجديدة ولكن معظمهم لا يرى فائدة كافية في التبديل حتى الآن. تستخدم 3٪ أخرى من جميع مطوري برمج الويب في المقام الأول TypeScript ، والتي يمكن اعتبارها أحدث إصدارات جافا سكريبت المتوفرة حالياً. ومع ذلك ، لا يرغب بعض مطوري الويب في فهم أي شيء لم يتم تطبيقه بعد في المعايير ، بينما لا يرغب آخرون في استخدام الأنواع الثابتة الاختيارية ، ولا تزال أقلية مهمة تتمنى أن شيئاً من Microsoft TypeScript قد اعتمدت. بالنظر إلى أن 2.x قد اعتمدت Microsoft .Angular 1.x تجد 41٪ من هؤلاء الذين يستخدمون اللغة في المقام الأول قد تبنوا الإطار. هناك 18٪ أخرى لا تزال تستخدم حالياً Angular 2.x والتي من المرجح أن تهاجر إلى

يفضل مطورو الويب الخلفي Angular في الواجهة الأمامية

بعد بعض النكبات من جافا سكريبت ، فإن اللغة الأكثر شعبية لمطوري الويب هي PHP ، مع أن 21٪ لا يزالون يعيرونها لغتهم الأساسية. نظر للتركيز على تقديم الصفحات على جانب الخادم في معظم أنظمة إدارة محتوى PHP الشائعة ، ليس من المفاجئ جدًا العثور على اهتمام أقل بأطر تطبيقات الصفحة الواحدة بشكل عام بين مطوري البرامج هؤلاء ، مع استمرار 52٪ في استخدام jQuery كسكناتهم الأساسية. ومن المثير للاهتمام أن 3٪ فقط من مطوري PHP يستخدمون بشكل أساسى 1.x Angular ، مع 8٪ على 2.x ، و 4٪ فقط لـ React. في الواقع ، لا يستخدم عدد كبير من مطوري PHP أي مكتبة أو إطار عمل للواجهة الأمامية (14٪) مثل استخدام React .Angular version plus

المطوريين الذين يستخدمون اللغات من جانب الخادم بشكل أساسى بخلاف JavaScript / Node.js أو PHP (يبلغ مجموعهم 42٪ من جميع مطوري الويب) أقل اهتماما بكثير لاستخدام jQuery من مطوري PHP ولكنهم أيضاً أقل اهتماماً بشكل كبير بالـ React و Angular من JavaScript (26٪ مقابل 38٪). عندما يستخدمون في المقام الأول أحد أطر الواجهة الأمامية هذه ، يختارون Angular (20٪) أكثر بكثير من React (6٪) ، وأكثر من المستخدمين لا angular في الإصدار 2.x (11٪) من الإصدار 1.x (9٪). وبالنظر إلى جميع المطوريين من جانب الخادم الذين لا يستخدمون Node.js ، والتي تمثل 63٪ من عدد مطوري الويب ، فإن Angular مفضل بشكل كبير للتفاعل في هذه المرحلة ، ربما لأنه إطار عمل كامل ، بدلاً من إجبار المطور على إجراء الكثير من خيارات المكتبة والأدوات الأخرى التي تحتاجها حالياً مع React ماذا حدث بعد ذلك؟

هناك العديد من العقود المستقبلية البديلة التي يمكن استنتاجها من هذه البيانات. ستكون أبسط قصة هي أن تفضيلات الإطار لن تتحرك كثير للمجموعات المختلفة. سيستمر المطوريين من جانب الخادم في الاهتمام قليلاً بالأطر الأمامية وسوف يتلزم مطورو Angular 1.x بـ ES5 عندما ينتقلون في النهاية إلى ESNext أو TypeScript. لا يتوافق

هذا مع الاتجاه الحالي لزيادة استخدام جافا سكريبت عبر الويب والواجهة الأمامية والخدم. كما يتجاهل أيضاًحقيقة أن Google ستنتقل إلى Angular 2.x داخلياً وأن المطوريين لن يرغون في تركهم بدون دعم في يوم واحد. يمكننا أيضاً تخيل أنه مع بدء المطوريين باستخدام TypeScript أو ESNext ، تغير تفضيلات إطار العمل وفقاً لذلك. ربما يكون هناك بعض الحقيقة في هذا ، لكنه يرتكز بشكل كبير على مطوري الواجهة الأمامية. من غير المرجح أن يجد مطورو البرامج من جانب الخادم الذين لا يستخدمون Node.js تفاعلاً جذاباً بدون منحني تعلم أبسط بكثير للأنظمة الإيكولوجية. ثم مرة أخرى ، لا يزال إطار PHP الأكثر شعبية هو WordPress ، وقد اختارت الشركة وراء WordPress.com إطار أمامي جديد لـ WordPress. ينبع Facebook برمجم كبير مع React ، ولكن من المرجح أن تظل Angular هي الأكثر شيوعاً للمشاريع الصغيرة والتطبيقات الداخلية. ما يمكن أن تتوقعه هو أنه على الرغم من الرخص الحتمي في الواجهة الأمامية ، فقد بني كل الإطارات بنجاح كتلة حرجية من المطوريين الذين ينشئون أنظمة بيئية قيمة ، وكلها تم تعينهما لتحقيق نمو كبير في السنوات القادمة. سنفاجأ إذا لم يصبح 30٪ من مطوري الويب الذين يستخدمون Angular أو React 40٪ في السنين المقبلتين. من جانب آخر بعد ظهور تقنية ReactJS و اعتماد المطوريين عليها في اغلب تطبيقاتهم . واعتماداً عليها تم ظهور إطار عمل React Native لبرمجة تطبيقات الموبايل الأصلية باستخدام لغة البرمجة جافاسكريبت ، هذا الإطار ايضاً مدعاوم من قبل شركة فيسبوك.

هو اطار يعد متوافقاً مع ECMAScript 6 الخاص بلغة البرمجة جافا سكريبت هذا أمر يعد رائعاً ، حيث يسمح لنا بكتابة كود منظم والاستفادة من أحداث إمكانيات وميزات لغة الجافاسكريبت .

لا يعتمد React Native على لغة CSS في تسيق واجهة المستخدم ، بل يقوم بذلك عن طريق الجافاسكريبت نفسها وذلك عن طريق كائنات Objects بخاصيات Properties وقيم Values .

هذه التقنية حظيت بشقة عد من كبار الفاعلين الاقتصاديين مثل سلسلة متاجر والمرت Walmart و منصة Airbnb الشهيره وطبعاً تطبيق فيسبوك وانستغرام يعتمدان في أجزاء منها على React Native. هنا يدفع للتفاؤل بمستقبل كبير لهذا الإطار وهو لا يزال قيد التطوير والتحسين.

كما أن فريق React يعمل بجد على بعض تحسينات الأداء المستقبلية ، على الرغم من وجود تركيز مختلف قليلاً. يدور العمل حول مشروع آخر في فيس بوك ، يدعى Prepack ، والذي يقع في قلب مترجم React جديد.

Prepack هي أداة تعمل على تحسين شفرة مصدر جافا سكريبت. يتم التخلص من الحسابات التي يمكن تنفيذها في وقت التحويل البرمجي بدلاً من وقت التشغيل . يحل برنامج Prepack محل الشفرة العامة لحزمة جافا سكريبت مع كود مكافئ وهو عبارة عن تسلسل بسيط من المهام.

كما انه الان يتم العمل على اصدار نسخة Angular5 بالاعتماد على تقنية 2 Angular و التي تعد منافسة React بشكل كبير.

A الملحق

تحميل وتنصيب Node.js

:Node.js - تحميل 1-A

<https://nodejs.org/en/download>

نفتح المتصفح وندخل الرابط التالي :

ثم نقوم اختيار الإصدار الذي يناسبنا هنا اختبرنا **Windows ونواة 64 ×**

Downloads

Latest LTS Version: 8.11.3 (includes npm 5.6.0)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS	Current
Recommended For Most Users	Latest Features
Windows Installer node-v8.11.3-win64.msi	macOS Installer node-v8.11.3.pkg
Source Code node-v8.11.3.tar.gz	

Windows Installer (.msi)

Windows Binary (.zip)

macOS Installer (.pkg)

macOS Binary (.tar.gz)

Linux Binaries (x86/x64)

Linux Binaries (ARM)

Source Code

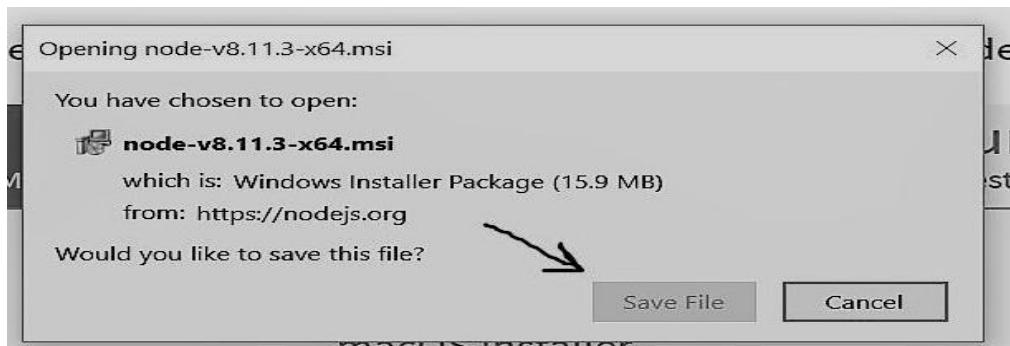
32-bit	64-bit
32-bit	64-bit ✓
64-bit	64-bit
64-bit	64-bit
32-bit	64-bit
ARMv6	ARMv7
node-v8.11.3.tar.gz	

Additional Platforms

32-bit	64-bit
Official Node.js Docker Image	
64-bit	64-bit
64-bit	64-bit

- Signed SHASUMS for release files (How to verify)
- All download options
- Installing Node.js via package manager
- Previous Releases
- Nightly builds
- Node-ChakraCore Nightly builds
- Building Node.js from source on supported platforms

تظهر لدينا النافذة التالية :



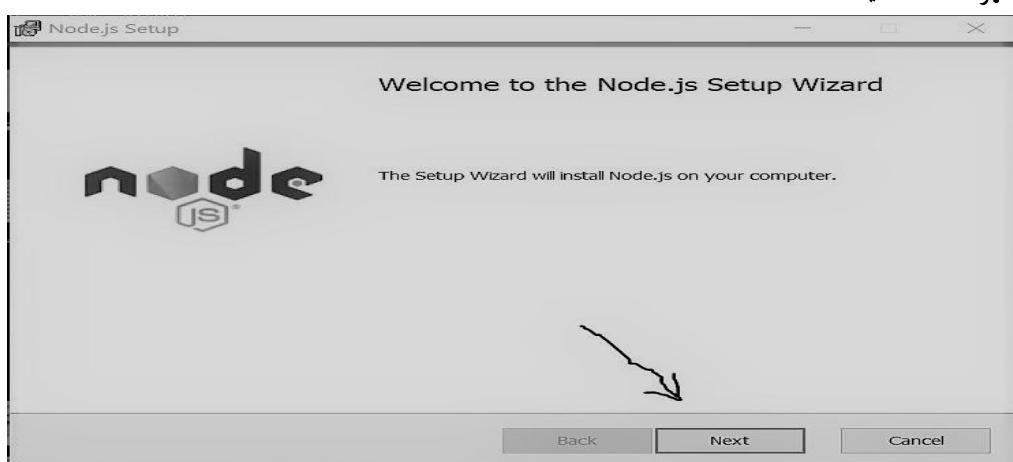
نختار **Save File** يبدأ التحميل ننتظر حتى انتهاء التحميل .

: Node.js -2-A

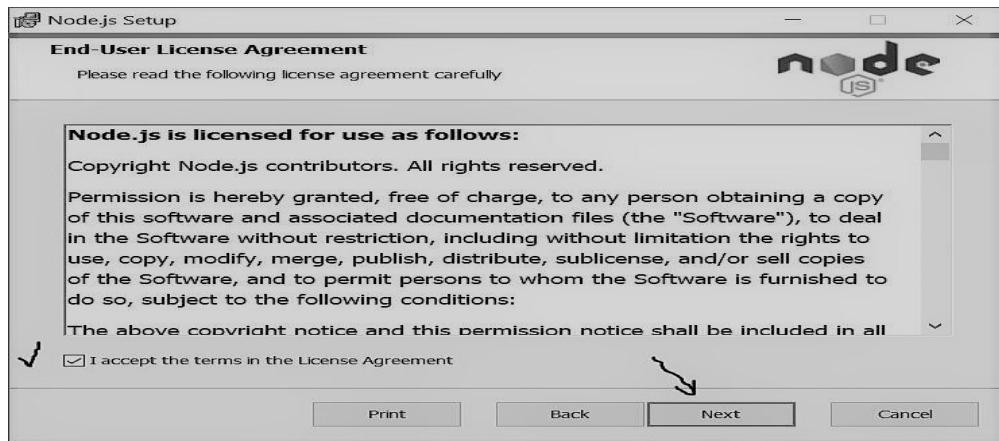
نقر على أيقونة الملف الذي حملناه



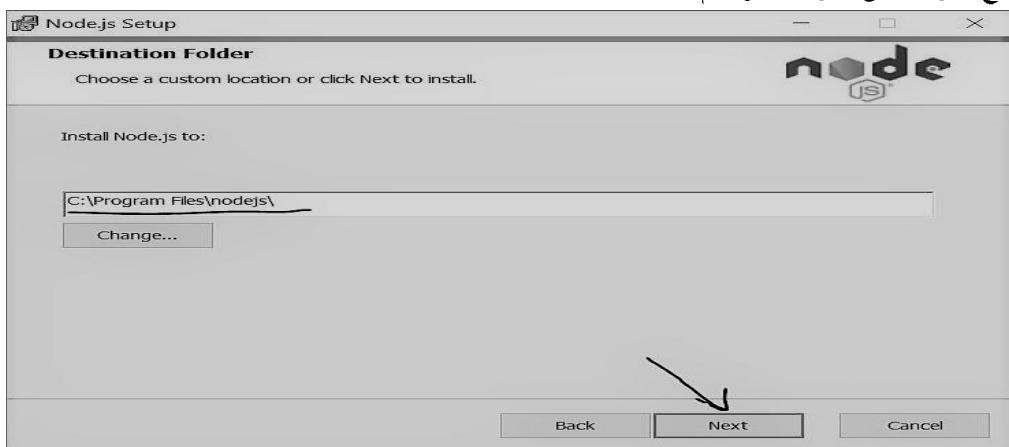
تظهر النافذة التالية :



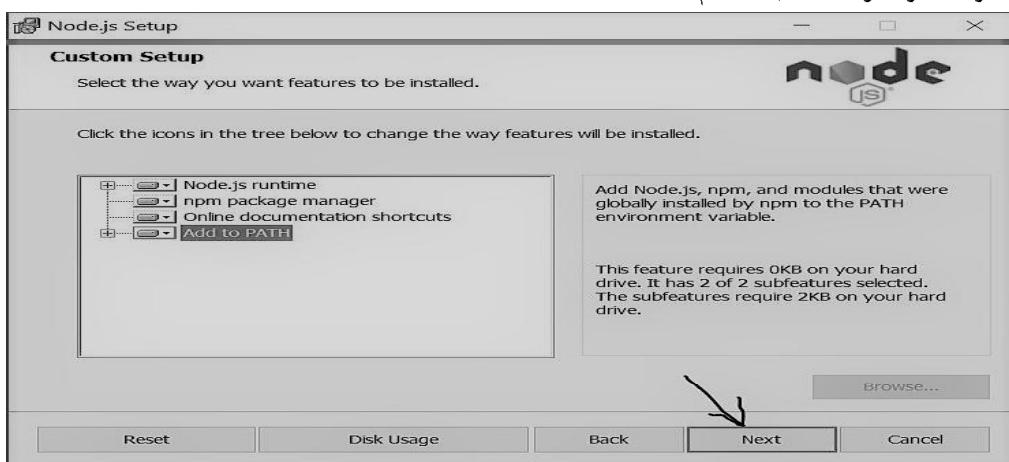
... **Next** نضغط



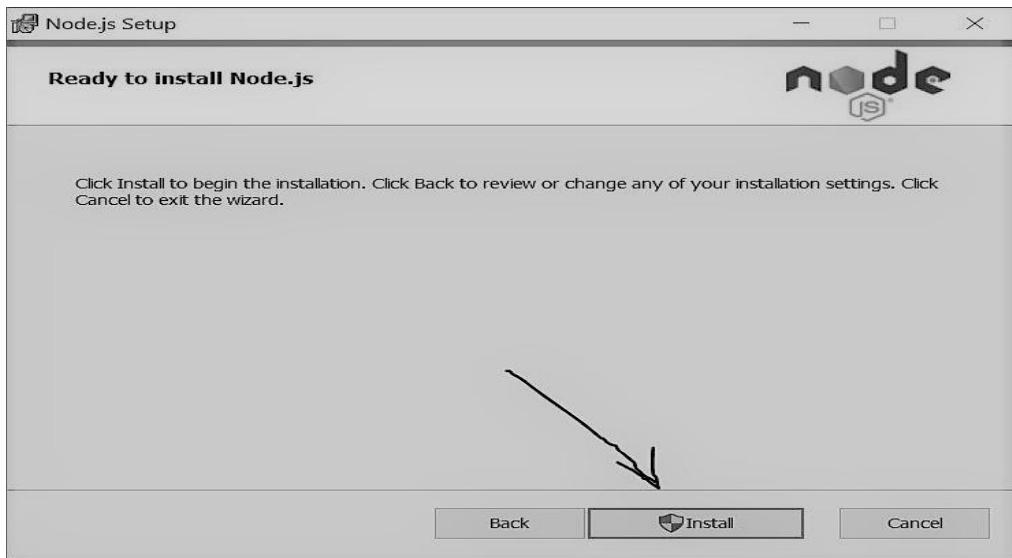
نضغط على حقوق الناشر ، ثم نضغط Next



نختار المسار المراد التنصيب فيه ، ثم نضغط Next



نختار الأدوات التي نريد تنصيبها ثم نضغط Next



الآن نضغط Install ثم يبدأ التنصيب وهكذا حتى ينتهي فيصبح جاهزا للعمل ..

المراجع

مسلسل	المادة	المراجع
1	HTML5	http://download-internet-pdf-ebooks.com/10452-free-book
		https://mustafasadiq0.com/2013/11/20/hyper-text-markup-language-version-5-html-5-
		http://download-internet-pdf-ebooks.com/12273-free-book
2	CSS3	https://en.wikipedia.org/wiki/Node.js
3	Node.js	https://getbootstrap.com/
4	Bootstrap	https://en.wikipedia.org/wiki/Bootstrap_(frontend_framework)
5	TypeScript	https://en.wikipedia.org/wiki/TypeScript
6	Gulp.js	https://en.wikipedia.org/wiki/Gulp.js
7	Javascript	https://en.wikipedia.org/wiki/JavaScript
		https://www.w3schools.com/jS/default.asp
		https://www.codecademy.com/learn/introduction-to-javascript

	libraries and frameworks	
https://onaircode.com/top-javascript-frameworks		
https://www.talentica.com/techwatch/TechWatch-Frameworks.pdf		
https://en.wikipedia.org/wiki/Meteor_(web_framework)		
https://en.wikipedia.org/wiki/JQuery		
https://vuejs.org		
https://en.wikipedia.org/wiki/Vue.js		
http://backbonejs.org		
https://en.wikipedia.org/wiki/Backbone.js		
http://www.getbreezenow.com/breezejs		
http://breeze.github.io/doc-js		
https://en.wikipedia.org/wiki/AngularJS		
https://www.c-sharpcorner.com/article/angularjs-2-0-from-beginning-introduction-of-angular-js-2-0		
https://reactjs.org		
https://en.wikipedia.org/wiki/React_(JavaScript_library)		
https://en.wikipedia.org/wiki/Document_Object_Model	DOM	9
https://www.w3schools.com/js/js_htmldom.asp		
https://en.wikipedia.org/wiki/Single-page_application	SPA	10
https://en.wikipedia.org/wiki/Model–view–controller	MVC	11
https://www.tutorialsteacher.com/mvc/mvc-architecture		