



الجمهورية العربية السورية
كلية الهندسة الكهربائية والميكانيكية
قسم الحاسبات والتحكم الآلي

توثيق

لبناء متجر إلكتروني على الإنترنت

إعداد الطلاب:

ندى أحمد وسوف عبد الكريم محمد ملا بهاء الدين يحيى اسكيف

تحت إشراف:

الدكتور عصام أسعد

**Syrian Arabic Republic
Faculty of mechanical and
electrical engineering
Department of computers
engineering**



Documentation

**To build An E-commerce Website on the
Internet**

Prepared by:

Nada Wassouf

Abdul Karim Mulla

Bahaa Eskif

Supervised by:

Dr. Issam Asaad

1	الفصل الأول	1
1	المقدمة:	1.1
3	أهمية البحث وأهدافه:	1.2
3	الدراسة المرجعية:	1.3
3	الكلمات المفتاحية المعبرة عن المشروع:	1.4
4	الفصل الثاني	2
4	اللغات والأدوات المستخدمة:	2.1
4	HTML :	2.1.1
6	CSS (Cascading Style Sheet) :	2.1.2
6	Tailwind:	2.1.3
7	javascript :	2.1.4
8	مكتبة React js :	2.1.5
8	مكتبة Formik :	2.1.6
9	مكتبة Redux :	2.1.7
9	بيئة العمل Node js :	2.1.8
10	إطار العمل Express.js :	2.1.9
11	TypeScript :	2.1.10
12	MongoDB :	2.1.11
13	محرك الأكواد Vscode :	2.1.12
14	الفصل الثالث	3
14	واجهة المستخدم (تطبيق العميل):	3.1
15	شريط الإشعارات (Navigation bar) :	3.1.1
16	قسم المنتجات:	3.1.2
18	صفحة التحميل:	3.1.3

جدول الرسوم التوضيحية:

Figure1-2 HTML Tag.....	5
Figure 2-2 CSS SYNTAX.....	6
Figure3-2 VS code IDE.....	13
Figure1-3 use case diagram.....	14
Figure2-3 landing page.....	15
Figure 4-3 Navigation bar.....	16
Figure 3-5 Link is provided from React.....	16
Figure6-3 products section.....	16
Figure 3-7 product card.....	17
Figure8-3 increment and delete product.....	17
Figure9-3 reducers.....	18
Figure10-3 Actions in redux.....	18

Abstract الملخص

يهدف المشروع إلى بناء موقع على الأنترنت يمثل متجر إلكتروني الذي يعد صلة الوصل بين العميل وصاحب العمل. والذي من خلاله يقوم بتقديم المنفعة على كلا الطرفين. والمشروع قائم على لغات برمجية أساسية خاصة بالويب (**Cascading ,HyperText Markup Language** , **Style Sheet, JavaScript**) بالإضافة إلى مكتبات وأدوات حديثة سرعت من وتيرة بناء المشروع.

الفصل الأول

المقدمة Introduction

1.1 المقدمة:

يعتبر المتجر الإلكتروني موقع إلكتروني يتم فيه عمليات البيع والشراء للمنتجات والخدمات عبر الإنترنت وتتوافر به خصائص مختلفة عن المواقع، مثل سلة التسوق التي يجمع بها العميل المنتجات التي يرغب بها، كما يتوفر إمكانية الدفع أون لاين. فهو شبيه بالسوق التجاري على أرض الواقع، ولكن عمليات البيع والشراء تتم من خلاله عن بعد لكلا الطرفين.

هناك العديد من الفوائد التي تجعل العملاء يرغبون في التسوق من خلال المتجر الإلكتروني:

- **سهولة التسوق**، فيستطيع العملاء المفاضلة والمقارنة بين المنتجات والأسعار والعروض المختلفة واختيار الأفضل دون أدنى مجهود.
- **المتجر الإلكتروني يكون متاح للعملاء على مدار الوقت** دون النظر لعطلات أو مسافات بعيدة كما لا توجد الحاجة للنزول من المنزل فيمكنهم تصفحه بسهولة ويسر وطلب ما يرغبون من سلع دون أدنى مجهود.
- **انخفاض سعر المنتجات** على المتاجر الإلكترونية، وذلك نتيجة انخفاض تكاليف ومصروفات المتجر مقارنة بالنوع الآخر.

هناك العديد من المميزات التي ساهمت في ازدياد أعداد المتاجر الإلكترونية، وهي:

1. توفير الوقت والجهد

المتاجر الإلكترونية من شأنها توفير الوقت والجهد الخاص بالعميل لأنه لن يضطر إلى الخروج من المنزل للتسوق كما أنك تستطيع التحكم بالمتجر عن بعد.

2. طرق دفع متعددة

من أهم مميزات المتاجر هي توفير طرق دفع متعددة يمكن لعملائك اختيار الأنسب منها.

3. معرفة آراء العملاء

تستطيع معرفة آراء العملاء بما تقدمه لهم ومعرفة أي المنتجات تلقى إعجابهم وأي المنتجات لا تناسبهم.

4. معرفة رغبات العميل وتلبيتها

عندما يقوم العميل بالتسوق داخل المتجر تستوقفه بعض المنتجات التي تلقى استحسانه ولكنه لا يريدّها في الوقت الحالي، هذا يسجل في السجل الشرائي للعميل بالمتجر، ويُمكن من تقديم اقتراحات وعروض على هذه المنتجات.

5. التواصل الدائم مع العملاء

يمكن التواصل بشكل دائم مع العملاء من خلال طلب اشتراكهم في الرسائل البريدية أو تسجيل الإعجاب بصفحات السوشيال ميديا الأمر الذي يجعلك أقرب للعملاء، وتقوم بإرسال عروض وتخفيضات وأحدث منتجات سواء من خلال الصفحات أو البريد الإلكتروني.

6. سهولة الوصول للعملاء

من المميزات المهمة للمتجر الإلكتروني هو سهولة الوصول للعملاء الذين يهتمون بمنتج من خلال الطرق التسويقية سواء عن طريق محركات البحث أو البريد الإلكتروني أو منصات السوشيال ميديا.

المتجر الإلكتروني هو بمثابة موقع مخصص لبيع للمنتجات والخدمات التي تقدمها على الويب، ففي ظل التطورات التكنولوجية الواردة في السوق التجاري، أصبح متجر الكتروني من أفضل السبل للحصول على مزيد من الأرباح وتحقيق المزيد من المبيعات.

1.2 أهمية البحث وأهدافه:

شهد القرن الـ 21 تغيراً جذرياً من ناحية التقدم التكنولوجي، حيث أضحت كل شيء تقريباً أن يصبح رقمياً حتى شمل مجال التجارة، فظهر العديد من المصطلحات مثل التجارة الإلكترونية، التسويق الرقمي ... إلخ. وتم الوضع مجموعة من الأسس والمبادئ في هذا المجال ليتم جذب أكبر عدد من الزبائن وزيادة المبيعات والكثير من الفوائد الأخرى وكان لبناء متجر الإلكتروني ليقوم البائع بتسويق بضاعته ومنتجاته عليه الركيزة الأساسية في هذا المجال.

1.3 الدراسة المرجعية:

المشروع يقوم على فكرة ليست جديدة بل تم بناء العديد من المتاجر الإلكترونية الموجودة حالياً على الإنترنت مثل متجر أمازون الذي يعتبر أول متجر إلكتروني على إنترنت. وتم استلهام الكثير من أمور مثل الإضافة إلى السلة والدفع إلكترونياً ومعاينة المنتج والبحث عنه ضمن مشروعنا. لكن تم تخصيص بعض الأمور لتناسب مع حجم مشروعنا والموقع الجغرافي الذي سيتم تخدم ضمنه المستخدمين.

1.4 الكلمات المفتاحية المعبرة عن المشروع:

متجر الكتروني، تجارة الكترونية، تسويق على الانترنت، متجر بيع أطعمة، متجر بيع حلويات وقهوة، إلخ...

الفصل الثاني

2.1 اللغات والأدوات المستخدمة:

إن بناء التطبيقات البرمجية تحتاج إلى لغة برمجية كي يستطيع الحاسب فهم المبرمج، أي أنها ببساطة اللغة التي يتخاطب بها المبرمج مع الحاسوب.

عادة ما تتم بناء التطبيقات الويب باستخدام لغتين **توصيف** هما **HTML, CSS** ولغة **برمجية** تدعى **JAVASCRIPT** ولكن عند بناء تطبيقات ضخمة ومعقدة من صعب جداً مداركة الأخطاء وصيانة الشيفرة المصدرية عند استخدام فقط هذه اللغات بذلك كما سنرى لاحقاً أن لدينا مجموعة من المكتبات والأدوات البرمجية التي تجعل بناء التطبيقات البرمجية من دون دموع. وبالطبع من الأدوات الهامة التي نحتاجها هي محرر النصوص الذي من خلاله سنكتب شيفرة البرنامج المصدر.

-تطوير واجهات المستخدم:

إن **تطوير واجهات المستخدم** أو الـ **Front-end** الخاص بالموقع هو ما تراه وتتفاعل معه في المتصفح، فهي تمثل تطوير واجهة المستخدم من نصوص أو ألوان أو أزرار أو صور. حيث أن كل العناصر من أزرار وأماكن للإدخال وتنسيقهم من ألوان وخطوط كل هذا يتم باستخدام تقنيات الـ **Front-end**.

-لغات الـ **Front-end**:

2.1.1 HTML :

الحروف هي إختصار **HyperText Markup Language**، وهي ليست لغة برمجية بالمعنى المتعارف عليه للغات البرمجة، ولكنها لغة أو طريقة لوصف محتويات صفحة الويب لبرامج التصفح، حيث أن أي برنامج متصفح انترنت يحتاج لمعرفة محتويات الصفحة من عناوين ونصوص وصور وروابط وغير ذلك من مكونات صفحة الويب، حتى تتمكن من عرضها بالشكل الصحيح، ولغة الترميز **HTML**، هي اللغة الرئيسية التي يتكون منها أي موقع أو صفحة على شبكة الإنترنت.

وتتكون صفحة الويب من مجموعة من العناصر (Tags) أو تعرف ك الوسوم، يكتب كل وسم بين (angle brackets) أو كما تسمى علامة أصغر من < وعلامة أكبر من >. ولا يقوم المتصفح بعرض هذه الوسوم، ولكن يقوم بعرض محتوياته. وهناك نوعين من الوسوم:

النوع الأول: وهو عبارة عن وسم يتم عرض محتواه مباشرة في صفحة الويب، مثال و <input />.

النوع الثاني: وهو عبارة عن وسم يصف ما بداخله من نصوص وقد يحتوي هذا الوسم بداخله على وسم آخر كعنصر فرعي لهذا الوسم وسوف نتلاحظ في هذا النوع أن كل وسم يتكون من جزئين (وسم البداية) و (وسم النهاية) وبينهما المحتوى الذي يتم عرضه في المتصفح



Figure1-2 HTML Tag

سوف نتلاحظ في الشكل الأعلى أن كل عنصر (من النوع الثاني) من عناصر لغة الـ HTML يتكون من وسم البداية و وسم النهاية ويجب أن يكون إسم وسم النهاية هو نفسه إسم وسم البداية ولكن مسبوق بعلامة / (أو المعروفة بإسم forward slash ، وبينهما محتويات العنصر والتي تظهر في متصفح الإنترنت، وسوف نتلاحظ أيضا أننا إستخدمنا وصف (attribute) للوسم، وكل وصف يحتوي على جزئين إسم وقيمة وبينهما علامة (=) ومن المثال السابق فإسم الوصف هو href والقيمة عنوان موقع إلكتروني، وكل ما سبق من أوامر (بإستثناء محتوى الوسم) (لا يظهر في متصفح الإنترنت) ولكن هي أوامر لمتصفح الإنترنت بإنشاء رابط تشعبي (Link) بالقيمة الموجودة في محتوى الوسم، وعند الضغط عليه يقوم المتصفح بتحويل الصفحة إلى عنوان الموقع الموجود في قيمة الوصف (بين علامات التنصيص)، ويمكن إستبدال عنوان الموقع بإسم صفحة أخرى.

2.1.2 CSS (Cascading Style Sheet):

تعتبر **CSS** -اختصار لـ **Cascading Style Sheet**) لغة توصيفية تقوم بمنح موقع الويب شكله الجميل وتصميمه الفريد الذي سيميزه عن غيره من المواقع . تعد لغة ال **CSS** صديقة للغة ال **HTML** فهي دائماً مرافقة لها وبجانبها في تصميم وإنشاء صفحات الويب. تشكل كل من هاتان اللغتان الأساس لكل موقع، فال **HTML** هي أساس كل عناصر صفحة الويب أما ال **CSS** فهي أساس تصميم وشكل الموقع فبدونها ستظل مواقع الويب نصاً عادياً على خلفيات بيضاء. وفي لغة ال **CSS** يتم تطبيق الخصائص على كل

العناصر التي تحمل ال **Class or Id** المختار في ال **CSS**، مثلاً في ال **HTML** لو تم تعيين **Selector** واحد لأكثر من عنصر في الصفحة وتم تعيين شكل وخصائص لهذا ال **Selector** فكل العناصر الحاملة له ستأخذ نفس الخصائص. في الشكل الأدنى يوضح لنا **Syntax** الخاص ب **CSS**.



```
1 body{
2     background-color: var(--gray);
3     position: relative;
4     height: 100vh;
5 }
```

Figure 2-2 CSS SYNTAX

2.1.3 Tailwind:

TailwindCSS هي عبارة عن اطار عمل للـ **CSS** لا تعتمد على المكونات **Component Based** كباقي أطر العمل الأخرى كـ **Bootstrap** و **Foundation** تعتمد عليه بشكل كبير. والجميل في هذا اطار العمل أنه يوفر لك أساسيات أولية لما تحتاجه لتصميم صفحات الويب أو بالأحرى يوفر لك نظام تصميم خاص بك مثل الهوامش ، والأحجام ، والألوان ...الخ، وهذا النوع من الأشياء. لن تجد مكونات جاهزة مثل الأزرار وعلامات التنقل - الأمر متروك لك في استخدام **Tailwind** لإنشاء مكوناتك الخاصة. هذا يعطي حرية هائلة حيث العديد من الأطر الحديثة تقيد فقط ما تنبئه ؛ ليست هناك حاجة لتجاوز أي شيء.

2.1.4: javascript

فهي لغة برمجية عالية المستوى تستخدم أساساً في متصفحات الويب ولها استخدامات واسعة أخرى في المجالات الأخرى , تضيف أساساً تفاعلية عالية الى صفحة الويب الخاصة بك , ظهرت أساساً لغة جافاسكريبت للمبرمجين الهواة ! وتم ملاحظتها من عموم المبرمجين المحترفين وتطويرها واصدار نسخ أعلى قوة منها لاحقاً لتصبح اللغة الأكثر اثارة للاهتمام من عام 2015 وحتى الآن, يتم انشاء ملفات جافاسكريبت بطريقة طبيعية , وبإضافة اللاحقة .js الى الملف لتعريفه على أنه ملف جافاسكريبت. تؤثر لغة جافاسكريبت في برمجة الويب بشكل كبير جداً , وتعتبر من قواعد وإساسيات أي مبرمج ويب فلن تجد أي مبرمج ويب بحق الا ويستطيع التعامل مع لغة جافاسكريبت . تدخل جافاسكريبت بشكل مباشر في تصميم وبرمجة الويب على حد سواء فلها ميزات التصميمية كميزاتها البرمجية بنفس القدر وخصوصاً بظهور مكتبات جافاسكريبت الحديثة ك **Vue.js** و **Angular** و **jQuery**.

مميزات جافا سكريبت :

1. تقدم جافاسكريبت ميزة البرمجة كائنية التوجه , يمكنك ذلك من انشاء كائنات (**Objects**) او كلاسات (**Classes**) لتسريع عملية البرمجة بشكل كبير وتنظيمه.
2. ميزة أخرى كبيرة للجافاسكريبت , قائمة مطولة جداً من الخدمات المدعومة من قبل جافاسكريبت ! يمكنك التعرف عليها من هنا الكثير من الخدمات يمكن لجافاسكريبت الحصول عليها عن طريق **api** الخدمة المناسب.
3. لغة جافاسكريبت هي لغة متعددة المنصات , ما معنى هذا ؟ معناه ببساطة أن هذه اللغة تستطيع التعامل مع جميع منصات البرمجة (**Web , Smartphones , Desktop**) , وتقدم خدمات مميزة في كل قسم منهم , فمثلاً , نجد أن محرر الأكواد الشهير **Atom** تم برمجته باستخدام جافاسكريبت , كما الكثير من أقسام هذا الموقع تم استخدام جافاسكريبت فيه. وموَقَّر للوقت والمجهود لكل المطورين.

2.1.5 مكتبة React js:

1. **React** (والتي تُعرف أيضًا باسم **React.js** أو **ReactJS**) هي مكتبة **JavaScript** تُستخدم لبناء واجهات المستخدم. تُدار **React** من شركة **Facebook** بالإضافة إلى مجتمع كبير من المطورين، فهي مشروع مفتوح المصدر.
2. تُسهّل **React** عملية إنشاء واجهات مستخدم تفاعلية. عليك فقط تصميم عروض (**views**) لكل حالة في تطبيقك، وستُحدّث **React** المكونات الصحيحة بكفاءة وتقوم بتُصييرها عندما تتغير بياناتك.
3. تعتمد **React** بشكل أساسي على مفهوم المكونات (**Components**). حيث يجب عليك بناء مكونات مُغلّفة تُدير حالتها الخاصة، ومن ثم تُركّب هذه المكونات مع بعضها لإنشاء واجهات مستخدم مُعقّدة. ولما كان منطق المكونات مكتوبًا باستخدام **JavaScript** بدلاً من صيغة القوالب، فإمكانك تمرير الكثير من البيانات عبر تطبيقك بسهولة وإبقاء الحالة بعيدة عن **DOM**.
4. تدمج **React** بين التصميم والبرمجة الخاصة بالواجهات ، ويكون ذلك عن طريق كتابة كود **JSX** وهو ليس كود **JavaScript** وليس كود **HTML**، بل هو ناتج عن دمج هاتين اللغتين ظاهريًا لنتج طريقة جديدة في الكتابة تسمى **JSX**، وهو اختياري أثناء استخدام الرياكت وهنا مثال عنها:
`const element = <h1>Hello World!</h1>`

وكما نرى فهذا الكود يحوي تعريف للعنصر `element` كما في **JavaScript**، ويحوي أيضًا عنصرًا يظهر كأنه **HTML**، حيث أن تافات تشبه تافات **HTML** وتملك أسماء وخصائص وتافات أبناء.

2.1.6 مكتبة Formik :

مكتبة خاصة بـ **Reactjs** تهتم بتعامل مع **Form** وتعتني **Formik** بالأشياء المتكررة والمزعجة - تتبع القيم / الأخطاء / الحقول التي تمت زيارتها ، وتنظيم التحقق من الصحة ، والتعامل مع الإرسال - حتى لا تضطر إلى ذلك. هذا يعني أنك تقضي وقتًا أقل في توصيل الأسلاك وتغيير المعالجات والمزيد من الوقت في التركيز على منطق عملك.

2.1.7 مكتبة Redux:

قد نستطيع إنشاء تطبيقات كاملة بواسطة **React.js** ولكن سرعان ما سنكتشف محدودية خياراتنا خصوصا عندما تزداد درجة تعقيد التطبيق وتشابك العلاقات بين مختلف مكوناته (**Components**). هذه المكتبة هي بالأساس مسؤولة عن إنشاء المشاهد (**Views**) في تطبيقات الويب وليست إطار عمل متكامل مثل أنجولار على سبيل المثال، ولهذا يتم تمثيلها عادة على أنها الحرف V في نظام **MVC** مكتبة **React.js** تعتمد نظام تدفق بيانات أحادي الاتجاه **Unidirectional Data Flow**، حيث أن المصدر الوحيد للحقيقة هو **State**، وبالتالي فكل تغيير يطرأ على حالة مكونات التطبيق يجب أن يمر أولا عبر تحديث هذا الكائن، ولهذا السبب أصلا سمي **State** أو حالة. وقد سبق لنا كتابة مقال خاص عن هذا الموضوع بيننا فيه وجه الاختلاف بين نظام **Unidirectional Data Flow** الذي تنتهجه **React.js** ونظام **Bidirectional Data Flow** الذي تعتمد عليه أطر عمل جافاسكربت الأخرى وعلى رأسها **Angular**.

2.1.8 بيئة العمل Node.js:

1. ال **Node.js** هو اطار عمل برمج العديد من وحداته **Modules** الأساسية بلغه **JavaScript** تم تطويره من قبل مبرمج امريكي يعيش في المانيا يدعى ريان دال و السوبر فايز جوينت، و كان اول إطلاق لهذا الباترن في عام 2009.
2. من المعروف بأن لغة الجافا سكربت هي لغة ال **Client Side**، بمعنى ذلك انها لغة تعمل بجانب المستخدم وليس في السيرفر ولكن بعد تطوير ريان اصبح الامر اكثر تعقيدا من ذي قبل و تحول هذا الإطار بالعمل في السيرفر كمثال ال **PHP**، بمعنى اخر، **Node Js** اصبح إطار عمل **Functionality** حركية و ديناميكية بشكل تام بإمكانك التعامل مع قواعد البيانات وما الى ذلك من خلاله بكل سهولة.
3. تستخدم **Node Js** لبرمجة تطبيقات الويب بالتحديد و المواقع بشكل عام، يعتمد في عمله على ال **events** او باللغة العربية تدعى (المناسبات)
4. عمليا، هذا يعني ان الموقع او التطبيق لا يمكن ان يتاخر بتنفيذ مناسبة **event** حتى لو دخله او استخدمه آلاف المستخدمين في نفس الوقت لان إطار ال **Node.js** برمج ليكون **Non-**

Blocking بذلك الأوامر يتم تنفيذها بالتوازي بخلاف ال PHP التي هي لغة **block until completion** أي ان الاوامر لا يتم تنفيذها الا بتنفيذ ما سبق من اوامر وهذا هو الفرق الاكبر ما بين PHP & Node.js وهذا هو سبب تميز إطار عمل النود وسبب السرعة الرهيبة في تنفيذ الاوامر.

2.1.9 إطار العمل Express.js:

Express.js هو أحد هذه الحلول التي جاءت لمساعدة مطورين **Node.js** على تطوير تطبيقات ويب على أساس متين ومنظم، إنه إطار عمل صغير (**Micro framework**) جلب معه عدد من المزايا التي يحتاجها قطعاً أي مطور في مشاريعه.

مزايا إطار العمل Express.js:

1. Routing:

من المزايا المهمة والأساسية التي يجب على كل إطار عمل توفيرها للمطورين هي إتاحة نظام لقراءة العناوين (**routes**) وهذه من أجمل الميزات التي نجدها في **Express**، نظام **Routing** الخاص به رائع وسهل الفهم والإستيعاب.

2. نظام القوالب **Templates**

3. **Middlewares**:

ال **Middleware** (أو الوسيط) هو عبارة عن برنامج صغير له وظيفة معينة ومحددة. وإطار العمل **Express** مبني على مفهوم ال **Middlewares** و يوفّر على عدة أنواع منها، نذكر منها ما يلي:

- **cookie-parser**: للتعامل مع الكوكيز.
- **Cookie-session**: للتعامل مع الجلسات (**sessions**).
- **CSssssssRF**: للحماية من ثغرات **CSRF**.

TypeScript هي لغة برمجة صممها شركة مايكروسوفت عام 2012 بغرض وهدف تحسين ورفع انتاجية مطوري الويب (جافا سكريبت). هذه اللغة تعتبر بمثابة توسعة للغة البرمجة جافا سكريبت بحيث توفر عددا من المزايا التي ليست مدعومة بعد بشكل رسمي من طرف جافا سكريبت أو أن دعمها محدود من طرف المتصفحات وكذلك Node.js.

-أهم مزايا تايب سكريبت:

1. دعم الأنواع:

تنقسم لغات البرمجة عموما إلى صنفين اثنين:

- لغات تتحقق من أنواع المتغيرات خلال مرحلة الترجمة أو **Compilation** ، ولذلك يسمى دعمها للأنواع بالدعم الستاتيكي أو الصريح (**Static Type Checking**) من هذه اللغات نجد جافا، **C++** و **Go**. عندما تصادف هذه اللغات خطأ متعلق بنوع المتغير في الكود فإن الترجمة تفشل ولا تتم بنجاح.
- وصنف آخر من اللغات يقوم بالتحقق من أنواع المتغيرات بشكل ديناميكي (**Dynamic Type Checking**) أثناء تشغيل البرنامج، ومن هذا الصنف نجد جافا سكريبت، بايثون و **PHP**.

مشكلة **Dynamic type Checking** أنه يؤدي في كثير من الحالات إلى برامج يصعب التأكد من اشتغالها بشكل طبيعي في وقت التشغيل (**Runtime**)، لهذا نجد أن نسبة كبيرة من أكثر الأخطاء التي تظهر في برامج جافا سكريبت هي أخطاء متعلقة بالأنواع.

2. الدوال في تايب سكريبت:

تضيف لغة **TypeScript** مجموعة من المميزات للدوال، فإلى جانب تحديد نوع البارامترات يمكننا كذلك تحديد نوع القيمة المرجعة من طرف الدالة. وإذا كانت الدالة لا ترجع أي قيمة فإننا نستعمل النوع **void** كما هو معمول به في العديد من لغات البرمجة المعروفة (**PHP** كمثال).

3. الواجهات (**Interfaces**): تتيح لنا الواجهات تعيين الخصائص والوظائف الواجب توفرها في كائن (**Object**) أو صنف (**Class**) محدد.

جافا سكريبت لغه برمجة لديها خصوصياتها التي تجعلها مختلفة عن غيرها من اللغات، ودور TypeScript هو عصرنة أو تحديث هذه اللغة وجعلها شبيهة بلغات البرمجة الأخرى التي تتبع نمط OOP. هذا يجعلها كذلك في متناول مطوري النظم الخلفية الذي يتوجسون . أغلبهم . من جافا سكريبت.

:MongoDB 2.1.11

قاعدة بيانات **MongoDB** عبارة عن نظام إدارة قاعدة بيانات **SQL** مفتوح المصدر ومتعدد المنصات. إنه مصمم لتخزين وإدارة كميات كبيرة من البيانات بطريقة مرنة وقابلة للتطوير. تتمثل إحدى الميزات الرئيسية لـ **MongoDB** في نموذج البيانات الموجه نحو المستندات ، والذي يخزن البيانات في شكل مستندات شبيهة بـ JSON مع مخططات اختيارية. يسمح ذلك بمرونة أكبر وأوقات تطوير أسرع ، حيث يمكن تعديل المخطط بسهولة دون الحاجة إلى عمليات ترحيل بيانات باهظة الثمن. من حيث التفاصيل الفنية ، تستخدم **MongoDB** بنية موزعة يتم فيها تقسيم البيانات عبر خوادم متعددة أو أجزاء. يسمح هذا بالتوسيع الأفقي ، مما يعني أن قاعدة البيانات يمكنها التعامل مع كميات متزايدة من البيانات وقراءة وكتابة أعباء العمل دون الحاجة إلى ترقية الأجهزة باهظة الثمن. يتضمن **MongoDB** أيضًا العديد من الميزات المضمنة للإتاحة العالية ، مثل تجاوز الفشل التلقائي ومجموعات النسخ المتماثلة. هذا يضمن أن تظل قاعدة البيانات متاحة ويمكن الوصول إليها أثناء أعطال الأجهزة أو الاضطرابات الأخرى.

MongoDB هي أداة قوية ومرنة لتخزين وإدارة كميات كبيرة من البيانات في التطبيقات الحديثة التي تعتمد على البيانات. إن نموذج البيانات الموجه نحو المستندات والبنية الموزعة يجعلها مناسبة تمامًا للعديد من حالات الاستخدام ، من التحليلات في الوقت الفعلي وإدارة المحتوى إلى تطبيقات إنترنت الأشياء والتجارة الإلكترونية .

الأدوات Tools :

2.1.12 محرر الأكواد Vscode :

هو محرر شيفرات مجاني و مفتوح المصدر "MIT License" مطور من قبل **Microsoft** موجه للعديد من الانظمة "Windows, Linux and macOS". يدعم هذا المحرر العديد من اللغات ويحتوي على عدد كبير من الاضافات التي ستسهل عليك العمل من كتابة و هيئة المحرر و تحليل الشفرة "Code".

يدعم **Visual Studio Code** الكثير من اللغات مثل **HTML, CSS, TypeScript, JavaScript** ولغات اخرى عديدة ويمكنك دعم لغات اكثر عن طريق **VS Code Marketplace** وكما يوفر توثيق للتعامل مع بعض اللغات يمكن ان تستكشفها في التوثيق الرسمي للمحرر

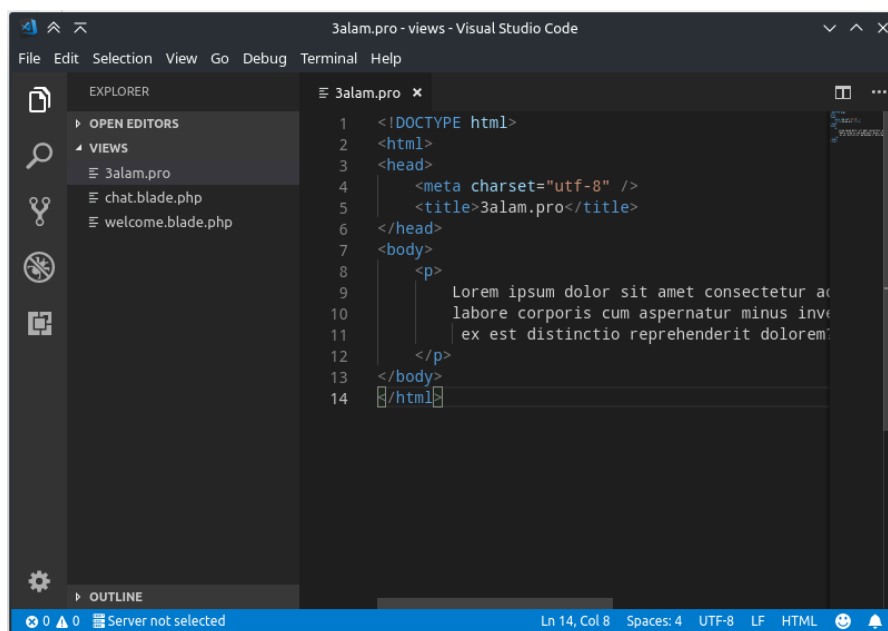


Figure3-2 VS code IDE

الفصل الثالث

3.1 واجهة المستخدم (تطبيق العميل):

يكن الهدف من واجهة المستخدم الفعالة في جعل تجربة المستخدم سهلة ومرنة، مما يعني الحد الأدنى من الجهد من جانب المستخدم مقابل الحد الأقصى من النتائج المرجوة من جانب المنتج. يتفاعل المستخدم مع الواجهة ويستطيع القيام بالكثير من الأفعال (Actions) التي من شأنها استجابة لمتطلبات المستخدم، يوضح لنا مخطط الحالة التالي (رقم الشكل) عن الأفعال التي تجري ضمن تطبيقنا حيث نلاحظ أن المستخدم يستطيع من خلال الواجهة إضافة منتج إلى السلة وتعديل أو حذف السلة وطلب وحساب الضريبة وعند الموافقة على الفاتورة يقوم الواتس اب API بإرسال إلى صاحب الموقع رسالة بتفاصيل الطلب. أما بالنسبة لمدير الموقع فيستطيع تسجيل الدخول إلى اللوحة التحكم وإعطاءه صلاحيات على حسب نوع الحساب ويستطيع إضافة وحذف المنتجات التي يتم عرضها ضمن المتجر وإضافة وتعديل الأصناف وإضافة الأسعار وتعديلها.

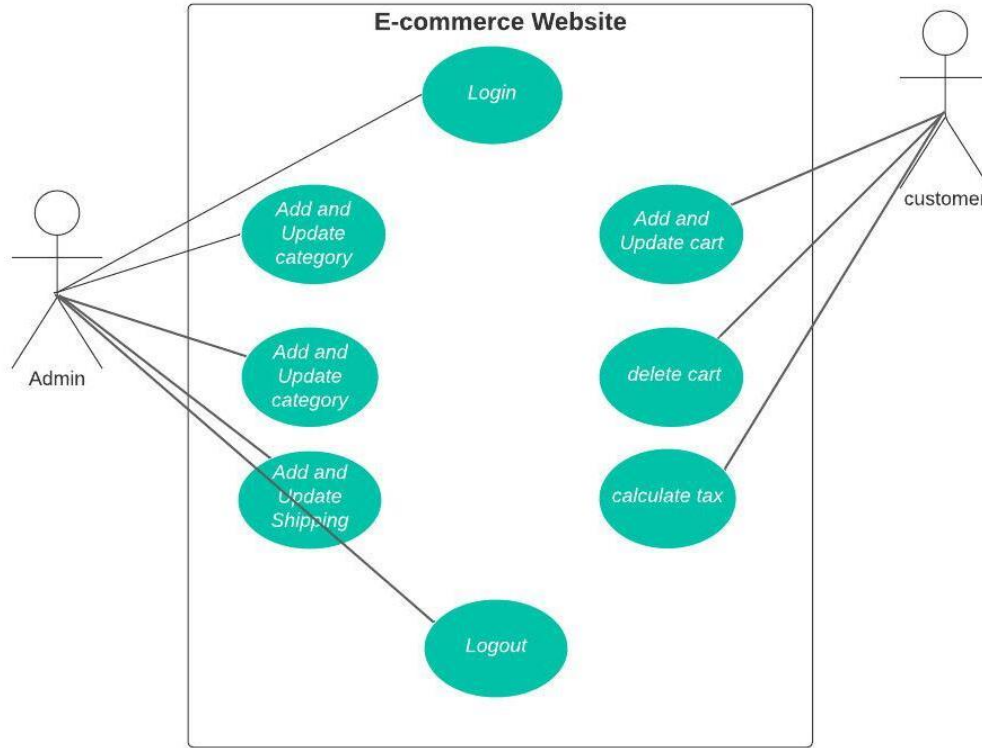


Figure1-3 use case diagram

تتألف الواجهة الرئيسية من شريط إشعارات (Navigation bar) الذي بدوره يحتوي على أزرار يمكننا من التنقل بين صفحات الموقع، وقسم المنتجات على شكل بطاقات تحتوي على زر إضافة المنتج إلى السلة وشريط سفلي لحساب السعر الإجمالي.

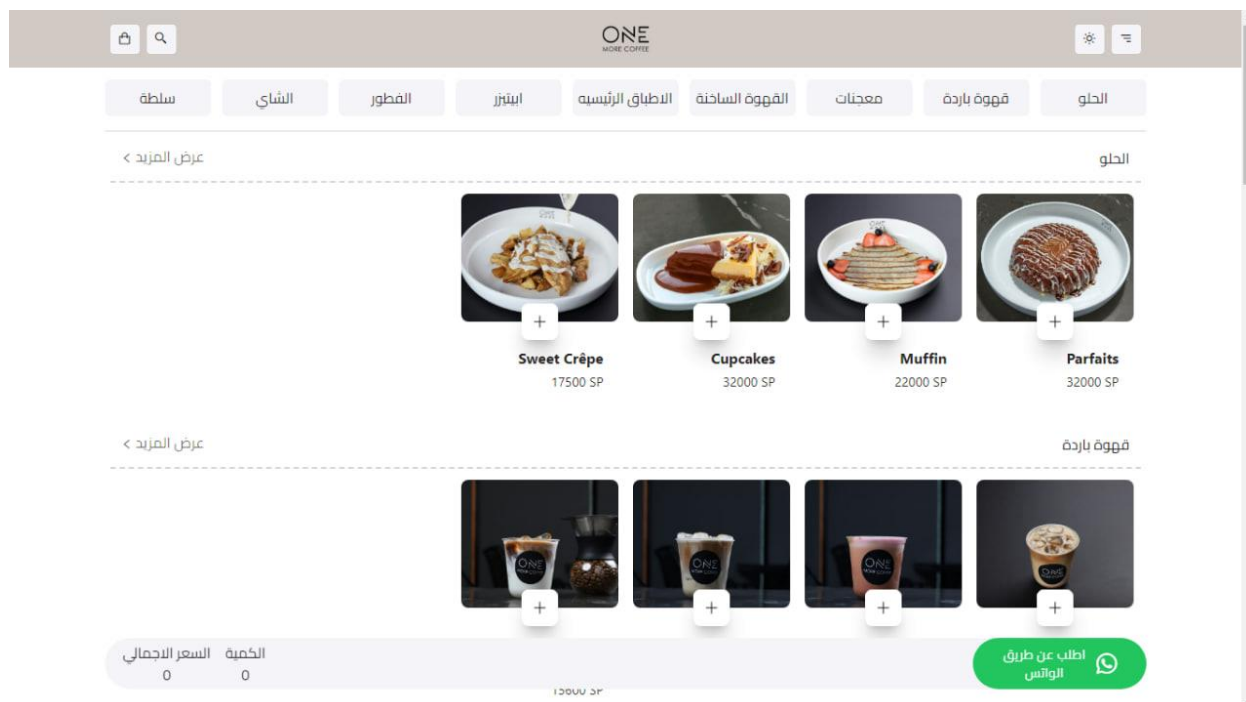


Figure2-3 landing page

3.1.1 شريط الإشعارات (Navigation bar):

شريط الإشعارات (Navigation bar) هو شريط مخصص لتقديم الإشعارات والتنبيهات يظهر في الجزء العلوي الخاص بالموقع ويستخدم بشكل كبير في المتاجر الإلكترونية لترويج العروض والخصومات وتحديثات المنتجات. يحتوي زر الشريط الإشعارات على أربع أزرار من اليمين الى اليسار (زر القائمة، زر الوضع الداكن، زر صفحة البحث، زر صفحة السلة). حيث ان زر القائمة الجانبية عند الضغط عليه تظهر قائمة تحتوي معلومات عن المتجر مثل أيام العمل وتواقيت الدوام والعنوان المتجر في سوق الفعلي. أما زر الوضع الداكن فيعطي للموقع ألوان داكنة مختلفة عن الألوان الأصلية. بالنسبة للزر الثالث فهو عبارة عن رابط لصفحة البحث عن المنتجات توفرها مكتبة **Reactjs** للتنقل بين صفحات دون الحاجة لإعادة تحميل الصفحة

(Figure 3-3). اما بالنسبة للزر الأخير فوظيفته مشابه للزر السابق إلا انه يقوم بربط مع صفحة السلة الالكترونية.



Navigation bar 4-3 Figure

```
<div className="flex items-center gap-2">
  <Link to={RouteGlobal.search} className={` ${darkMode ? ' bg-tertiary
    <i className={`bx bx-search-alt-2 text-md ${darkMode ? ' text-teal-500
  </Link>
  <Link to={RouteGlobal.cart} className={` ${darkMode ? ' bg-tertiaryDa
    <i className={`bx bx-shopping-bag text-md ${darkMode ? ' text-teal-500
      {amount} > 0 ? <div className={`absolute left-[-8px] top-[-5px] w-4
    </Link>
```

Figure 3-5 Link is provided from React

3.1.2 قسم المنتجات:

وهنا يستطيع صاحب المتجر عرض جميع منتجاته حسب الأصناف التي تنتمي إليها. ولعرض هذه المنتجات تكون بالأساس تمت إضافتها من قبل اللوحة التحكم إلى قاعدة البيانات نقوم باستعلام عن هذه البيانات بالاتصال ب **Endpoints** الذي يوفرها لنا ال **Back-End** وتعود البيانات بشكل غرض (**Array**) نقوم بالتفاف على المصفوفة ونعيد عناصرها على شكل أوسمة ب **JSX** وهكذا يكون قد جعلنا الواجهة ديناميكية بحيث عند كل تحديث بقاعدة البيانات يتم التحديث البيانات عند المستخدم بشكل تلقائي.

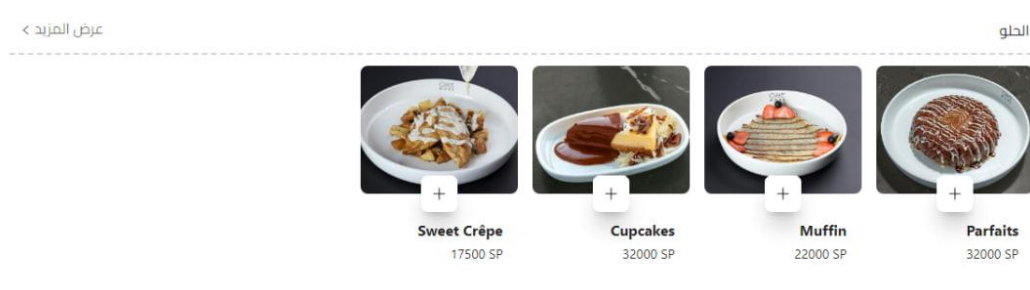


Figure 6-3 products section

كل منتج في هذا القسم يتمثل ببطاقة تحتوي على اسم المنتج وسعره وزر لإضافته إلى السلة المشتريات، لكن ما يحدث بالفعل هو عند ضغط على هذا الزر يقوم الزر باستدعاء تابع هذا التابع يقوم بدفع فعل (الشكل 3-9) مبرمج مسبقا لنفترض أنه إضافة منتج إلى سلة فأنه يقوم بدفعه إلى المخزن الذي بدوره يستدعي تابعا خاصا يسمى ب **Reducer** (الشكل 3-10) وظيفته تنفيذ أمر برمجي الذي يتمثل بإضافة منتج الذي يشكل غرض **object** إلى مصفوفة خاصة بسلة المشتريات عندها يصبح لدينا نسخة عن كل المنتجات التي ضغط المستخدم الزر عندها. هذه الطريقة هي طريقة ومنطقية عمل خاصة فقط ب **Redux** قد تختلف من مكتبة إلى أخرى.



Figure 3-7 product card

بعد أن تتم إضافة المنتج إلى السلة تتم زيادة كمية المنتج بواحد ويظهر لدينا زر جديد هو حذف المنتج بحالة كان المنتج كميته فقط 1 أما إذا كانت بخلاف ذلك يظهر زر بإنقاص كميته بواحد.



Figure 8-3 increment and delete product

```
export const {
  minusQuantity,
  addQuantity,
  productsRequest,
  productsReceive, clearProduct } = slice.actions
```

Figure10-3 Actions in redux

```
reducers: {
  addQuantity: (state, { payload }) => {
    const product = state.listItems.find((item) => {
      return item.id === payload.id;
    });
    product.quantity = product.quantity + 1;
  },
  minusQuantity: (state, { payload }) => {
    const product = state.listItems.find((item) => {
      return item.id === payload.id;
    });
    product.quantity = product.quantity - 1
  }
}
```

Figure9-3 reducers

3.1.3 صفحة التحميل: