**جامعة تشرين**

**كلية الهندسة المعلوماتية**

**السنة الخامسة**

**قسم البرمجيات ونظم المعلومات**

**تصميم موقع ويب لإدارة علاقات العملاء**

**Customer Relationship Management (CRM)**

**مشروع تخرج**

**إعداد الطلاب**

مايا أحمد ناصر احمد غسان تركمان

**إشراف**

د. باسل حسن

العام الدراسي 2022-2023

المحتويات

[فهرس الأشكال: 4](#_Toc142289270)

[فهرس الجداول: 5](#_Toc142289271)

[ملخص 6](#_Toc142289272)

[1. مقدمة 7](#_Toc142289273)

[1.1. الهدف من المشروع 7](#_Toc142289274)

[2.1. نطاق المشروع 7](#_Toc142289275)

[2. توصيف النظام: 8](#_Toc142289276)

[1.2 متطلبات النظام: 12](#_Toc142289277)

[1.1.2 المتطلبات الوظيفية: 12](#_Toc142289278)

[2.1.2 المتطلبات غير الوظيفية: 13](#_Toc142289279)

[2.2 مخططات UML: 14](#_Toc142289280)

[1.2.2 مخطط حالة الاستخدام Use Case Diagram: 14](#_Toc142289281)

[2.2.2 مخطط الصفوف class diagram: 18](#_Toc142289282)

[3. التقنيات والبرامج المستخدمة: 19](#_Toc142289283)

[1.3 تقنيات الـ Frontend: 19](#_Toc142289284)

[1.1.3 لغة HTML: 19](#_Toc142289285)

[2.1.3 CSS 19](#_Toc142289286)

[3.1.3 JavaScript: 19](#_Toc142289287)

[4.1.3 React JS: 20](#_Toc142289288)

[5.1.3 Visual studio code: 21](#_Toc142289289)

[2.3 تقنيات الـ Backend: 22](#_Toc142289290)

[1.2.3 قاعدة بيانات MySQL: 22](#_Toc142289291)

[2.2.3 Java: 22](#_Toc142289292)

[3.2.3 Spring: 22](#_Toc142289293)

[4.2.3 Spring Boot: 23](#_Toc142289294)

[5.2.3 MySQL Workbench: 23](#_Toc142289295)

[6.2.3 Spring Tool Suit (STS) 24](#_Toc142289296)

[4. تصميم قاعدة البيانات Database Design: 25](#_Toc142289297)

[5. صفحات الموقع الأساسية: 27](#_Toc142289298)

[1.5 صفحة تسجيل الدخول login: 27](#_Toc142289299)

[2.5 الصفحة الرئيسية للموقع Home Page: 27](#_Toc142289300)

[3.5 صفحة المنتجات: 28](#_Toc142289301)

[4.5 صفحة خطوات البيع sales steps: 29](#_Toc142289302)

[5.5 صفحة الـ opportunities: 29](#_Toc142289303)

[6.5 صفحة مخطط القمع Funnel Chart: 30](#_Toc142289304)

[7.5 صفحة خطط العمل التجارية: 31](#_Toc142289305)

[8.5 صفحة تفاصيل خطة العمل التجارية: 32](#_Toc142289306)

[6. الخاتمة: 34](#_Toc142289307)

[7. مقترحات لتطوير المشروع: 35](#_Toc142289308)

[المراجع: 36](#_Toc142289309)

# فهرس الأشكال:

|  |  |
| --- | --- |
| الشكل 1. **قمع المبيعات** | 9 |
| الشكل 2. **تحليل SWOT** | 9 |
| الشكل 3. **مخطط 20/80** | 11 |
| الشكل 4. **مراحل BCrypt** | 13 |
| الشكل 5. **مخطط حالة الاستخدام** | 14 |
| الشكل 6. **مخطط الصفوف** | 18 |
| الشكل 7. **مخطط ERD** | 25 |
| الشكل 8. **صفحة تسجيل الدخول** | 27 |
| الشكل 9. **الصفحة الرئيسية** | 72 |
| الشكل 10. **صفحة المنتجات** | 28 |
| الشكل 11. **إضافة منتج** | 28 |
| الشكل 12. **خطوات البيع** | 29 |
| الشكل 13. **Opportunities** | 29 |
| الشكل 14. **تفاصيل الـ opportunity** | 30 |
| الشكل 15. **مخطط قمع المبيعات** | 30 |
| الشكل 16. **خطط العمل التجارية** | 31 |
| الشكل 17. **إضافة خطة عمل تجارية جديدة** | 31 |
| الشكل 18. **رؤية خطة العمل** | 32 |
| الشكل 19. **تحليل swot** | 32 |
| الشكل 20. **مخطط ABC** | 33 |
| الشكل 21. **الزبائن مرتبين حسب تقييم RFM** | 33 |

# فهرس الجداول:

|  |  |
| --- | --- |
| جدول 1. **توصيف حالة الاستخدام login** | 15 |
| جدول 2. **توصيف حالة الاستخدام show sales funnel chart** | 16 |
| جدول 3. **توصيف حالة الاستخدام show RFM analysis** | 17 |

# ملخص

إدارة علاقات العملاء (CRM) عبارة عن مزيج من الأشخاص والعمليات والتكنولوجيا التي تسعى إلى فهم عملاء الشركة. إنه نهج متكامل لإدارة العلاقات من خلال التركيز على الاحتفاظ بالعملاء وتطوير العلاقات. تطورت إدارة علاقات العملاء نتيجة التقدم في تكنولوجيا المعلومات والتغييرات التنظيمية في العمليات التي تتمحور حول العملاء. الشركات التي تنفذ إدارة علاقات العملاء بنجاح ستجني ثمار ولاء العملاء والربحية على المدى الطويل. ومع ذلك، فإن التنفيذ الناجح بعيد المنال للعديد من الشركات، ويرجع ذلك في الغالب إلى أنهم لا يفهمون أن إدارة علاقات العملاء تتطلب إعادة هندسة عمليات الأعمال على مستوى الشركة.

# مقدمة

يعتبر نظام إدارة علاقات العملاء (CRM) نظاماً معروفاً لإدارة اتصالات المؤسسة مع العملاء الحاليين والعملاء المحتملين. يُستخدم هذا النظام لتنظيم ومزامنة وأتمتة أنشطة الأعمال والمبيعات. إنه نظام افتراضي ولا يتطلب أي مكاتب أو بيئة فعلية. لذلك يمكن للجميع استخدامه بسهولة.

## 1.1. الهدف من المشروع

* تبسيط عملية التسويق ونظام المبيعات.
* تقديم خدمة أفضل للعملاء.
* اكتشاف عملاء جدد وزيادة الأرباح من العميل.
* زيادة بيع المنتجات بشكل أكثر فعالية.
* الحفاظ على علاقة جيدة مع العملاء.
* تتبع الفرص التجارية وتطويرها.
* مراقبة العملاء.

تتمثل الأهداف العامة في البحث عن عملاء جدد وجذبهم والحصول عليهم، ورعاية العملاء الذين تمتلكهم المؤسسة بالفعل، حيث يجب جذب العملاء السابقين مرة أخرى. بالإضافة إلى خفض تكلفة التسويق وخدمة العملاء.

## 2.1. نطاق المشروع

يستهدف المشروع أصحاب الشركات الصغيرة إلى متوسطة الحجم المهتمين بإدارة زبائنهم. حيث إن تكلفة شراء واستخدام نظام CRM شهيراً تعتبر باهظةً جداً بالنسبة للشركات الناشئة، بالتالي أصبح استخدامها حكراً على الشركات الكبيرة. وكما نعلم، إن النسبة الأكبر من الشركات في السوق الآن هي شركات صغيرة ومتوسطة. فكرة مشروعنا كانت لإنشاء نظام يقدم ميزات تشبه ميزات أنظمة الـ CRM الشهيرة ولكن بسعر مناسب.

# توصيف النظام:

يهدف نظامنا بشكل أساسي إلى إدارة العلاقة مع الزبون وفهم احتياجاته وتحسينها مع الوقت، حيث يستهدف المؤسسات والشركات.

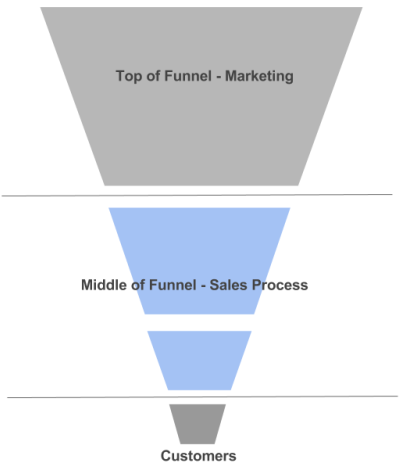
يجب على المستخدمين أن يسجلوا دخولهم إلى النظام من أجل أن يقوموا بوظائف النظام. بعد تسجيل الدخول، يمكن لجميع المستخدمين أن يقوموا بإدارة الزبائن customers. حيث يمكن إضافة زبون جديد أو تعديل زبون أو حذف زبون. يتم تعريف الزبون باسمه وعنوانه المفصّل والرمز البريدي والمدينة والدولة ورقم الهاتف الثابت ورقم الجوال والإيميل. وفي حال كان الزبون شركة، اسم المدير ورقم أعمال الشركة والقوة العاملة (عدد الموظفين) وتاريخ الإنشاء ورقم السجل التجاري بالإضافة إلى الموقع الإلكتروني.

كما تتم إدارة الزبائن المحتملين Leads من إضافة وحذف وتعديل ومراقبة، حيث لديهم نفس معلومات الزبون.

يمكن أن يضيف المستخدم المنتجات الخاصة به إلى النظام. ويقوم كل مستخدم بتحديد خطوات البيع sales steps الخاصة بخطة مبيعاته.

يتيح النظام للمستخدمين إنشاء opportunities وإدارتها، حيث ترتبط كل opportunity بمنتج واحد وزبون واحد. حيث عند إنشاء الـ opportunity تصبح في حالة open، ثم يتم المرور بخطوات البيع إلى حين انتهائها فتصبح في حالة closed، حيث إما أن تكون won أو lost.

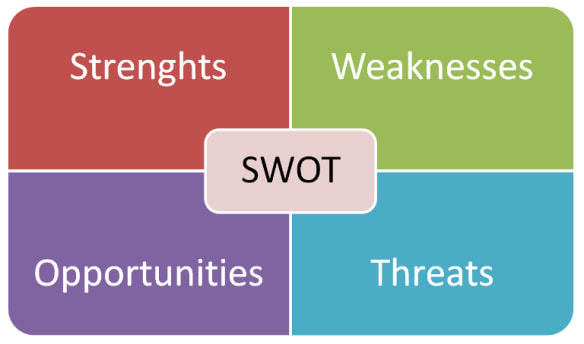
يتم إعطاء حساب لكل شركة تستخدم النظام، حيث يمكن للشركة إضافة حسابات فرعية لموظفيها. كما يمكن أن تعرض إحصائيات وتقارير لعملها، حيث يوفر النظام عرض مخطط القمع لمبيعات (Funnel chart) الشركة. حيث قمع المبيعات هو مصطلح التسويق للرحلة التي يمر بها العملاء المحتملون في طريقهم للشراء. هناك عدة خطوات لمسار المبيعات، والمعروف عادةً باسم الجزء العلوي والمتوسط والجزء السفلي من مسار التحويل، على الرغم من أن هذه الخطوات قد تختلف اعتماداً على نموذج مبيعات الشركة.



**الشكل 1.** قمع المبيعات

كما يمكن للشركة أن تدير خطط عملها التجارية. حيث يمكن أن تُنشئ خطط عمل تجارية تتضمن دراسة للخطة وإحصائيات لفهم مدى جدواها.

يتم أولاً تحديد الهدف من الخطة أو الرؤية الخاصة بالخطة، ثم تطبيق تحليل SWOT عليها لتحديد نقاط القوة والضعف وفهم السوق ولتحديد الإجراءات اللازم اتباعها لتحقيق أفضل نتيجة.



**الشكل 2.** تحليل SWOT

ثم يتم إجراء عدة دراسات وإحصائيات لتحديد أفضل الزبائن الذين لديهم أكبر احتمالية للشراء.

يتم تطبيق تحليل Pareto لتحديد الزبائن الذين حققوا أكبر ربح للشركة. ويتم عرض مخطط ABC لهم. كما يسمح النظام بوضع أرباح متوقعة من الزبائن في المستقبل وتصنيف المستخدمين وفقاً للأرباح السابقة والأرباح المتوقعة.

**ما هو تحليل ABC في إدارة المخزون؟**

تحليل ABC هو تقنية لإدارة المخزون تحدد قيمة عناصر المخزون بناءً على أهميتها بالنسبة للأعمال. تصنف ABC العناصر حسب الطلب وبيانات التكلفة والمخاطر، ويقوم مديرو المخزون بتجميع العناصر في فئات بناءً على تلك المعايير. يساعد هذا قادة الأعمال على فهم المنتجات أو الخدمات الأكثر أهمية للنجاح المالي لمنظمتهم. [1]

أهم الوحدات، بناءً على حجم المبيعات أو الربحية، هي عناصر "الفئة A"، تليها العناصر الأكثر أهمية هي "الفئة B" والأقل أهمية هي "الفئة C". قد تختار بعض الشركات نظام تصنيف يقسم المنتجات إلى أكثر من مجرد تلك المجموعات الثلاث (A-F، على سبيل المثال). [1]

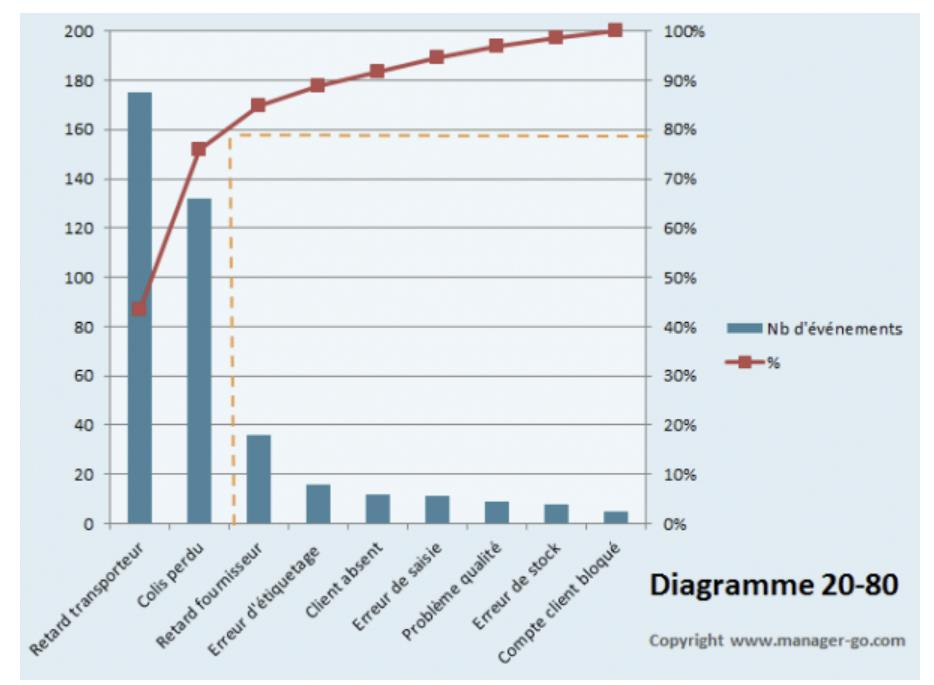
قمنا باستخدام هذا التحليل في نظامنا لتصنيف الزبائن، حيث أن الزبائن الذين حققوا أكبر ربح للشركة يتم تصنيفهم في الفئة A، والزبائن الذين حققوا ربح متوسط يتم تصنيفهم في الفئة B، وأخيراً الزبائن الذين حققوا أقل ربح يتم تصنيفهم في الفئة C.

**كيف يرتبط تحليل ABC بمبدأ Pareto؟ [1]**

يقول مبدأ باريتو أن معظم النتائج تأتي من 20٪ فقط من الجهود أو الأسباب في أي نظام. استناداً إلى قاعدة باريتو 80/20، يحدد تحليل ABC أن 20٪ من الزبائن يقدمون حوالي 80٪ من الربح.

لذلك، تمتلك معظم الشركات عدداً صغيراً من الزبائن في فئة "A"، ومجموعة أكبر قليلاً في "B" ومجموعة كبيرة في "C"، وهي فئة تحدد غالبية الزبائن.

في الشكل 3، الخط المنحني يعبر عن نسبة الأرباح، كما نلاحظ إن 80٪ من الأرباح تم تحقيقها عن طريق نسبة قليلة فقط من الزبائن.



**الشكل 3.** مخطط 20/80

كما يتم تطبيق تحليل RFM، الذي يصنف الزبائن وفقاً للمدة الزمنية منذ آخر شراء، وتكرارات الشراء، والربح التراكمي منه خلال فترة زمنية محددة.

وأخيراً يتم تصنيف الزبائن وفقاً لعاملين، العامل الأول هو نسبة الأرباح المتوقعة والعامل الثاني هو مدى إمكانية الوصول إلى الزبون.

يمكن أن يقوم المستخدمون بتسجيل الخروج logout.

## 1.2 متطلبات النظام:

### 1.1.2 المتطلبات الوظيفية:

1. تسجيل دخول وتسجيل خروج login and logout.
2. إدارة الزبائن من إضافة وتعديل وحذف.
3. إدارة المستخدمين المحتملين Leads.
4. إدارة المنتجات.
5. تحديد خطوات البيع sales steps وتعديلها وحذفها.
6. إنشاء opportunity وإدارته وتتبع حالته.
7. عرض مخطط القمع (Funnel chart).
8. إضافة الموظفين وإنشاء حسابات لهم.
9. إدارة خطط العمل التجارية، حيث في كل خطة يتم تحديد الهدف منها وتطبيق تحليل SOWT، وتحليلات pareto و RFM و IPO/IPA.

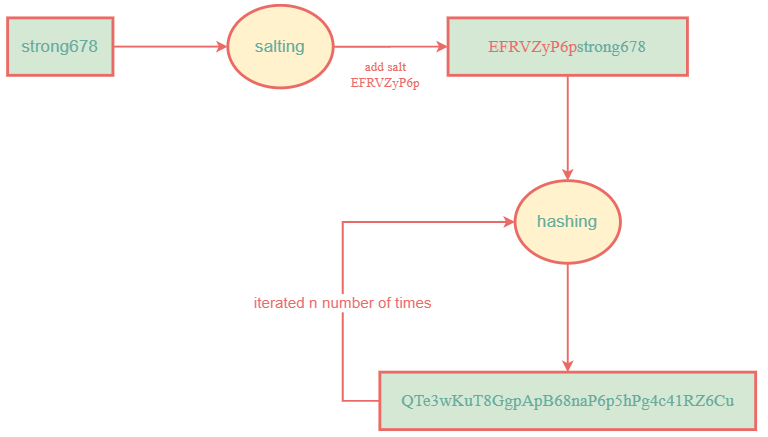
### 2.1.2 المتطلبات غير الوظيفية:

* **الأمان:**

**يتم تشفير كلمات المرور والبيانات الحساسة باستخدام خوارزمية BCrypt، حيث تمر عملية التشفير بالمراحل التالية:**

* **يتم أولاً إضافة تسلسل عشوائي من الأحرف إلى كلمة المرور.**
* **ثم يتم تجزئة كلمة المرور.**
* **تحافظ عملية التجزئة على تكرار نفسها لعدد محدد من الجولات، يسمى عامل التكلفة.**

**يبين الشكل التالي مراحل الخوارزمية:**

****

الشكل 4. **مراحل BCrypt**

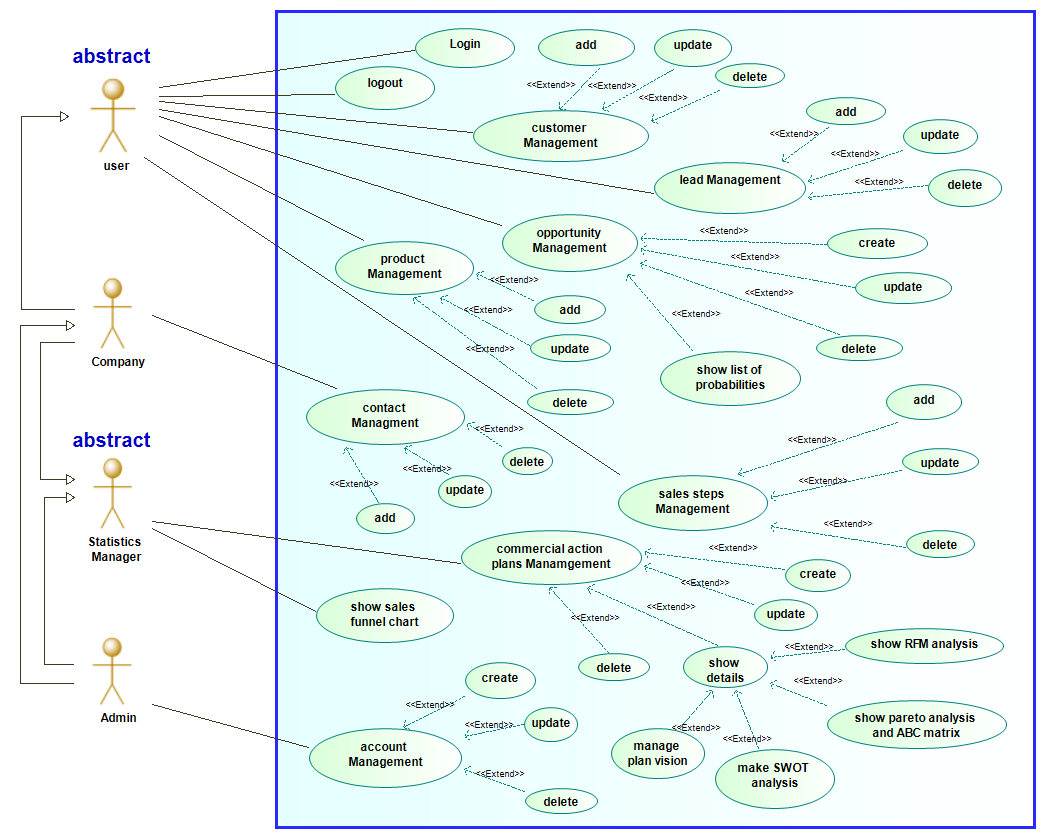
* **قابلية التوسيع والصيانة:**

**حيث النظام مبني على معمارية الـ MVC. إن المفهوم الأشهر لنمط MVC هو أنه يعتمد على مبدأ فصل الاهتمامات SoC (Separation of concerns)، حيث تعد تطبيقات الويب الحديثة معقدة للغاية، وقد يكون إجراء تغيير في بعض الأحيان مشكلة كبيرة.**

**الاعتماد على هذا المبدأ يتيح للتطبيق أن يكون قابلاً للصيانة والتوسيع.**

## 2.2 مخططات UML:

### 1.2.2 مخطط حالة الاستخدام Use Case Diagram:



الشكل 5. **مخطط حالة الاستخدام**

**جداول توصيف حالات الاستخدام:**

**يبين الجدول 1 توصيف حالة الاستخدام login:**

|  |  |
| --- | --- |
| Name: | Login |
| Short description: | The user logs into the system using his user name and password. |
| Precondition: | The user has an account. |
| Postcondition: | The user can perform all the functionalities that he is allowed to. |
| Error situations: | 1. Invalid email. 2. Wrong password. |
| System state in the event of an error: | The system show an error message and the user wouldn’t access to his account. |
| Actors: | user |
| Trigger: | The user clicks Login button. |
| Standard process: | 1. User enters his user name and password and hits login button. 2. System confirms that user name and password are valid. 3. User successfully logs in. |
| Alternative processes: | (2`) user name or password is not valid.  (3`) System shows an error message. |

جدول 1. **توصيف حالة الاستخدام login**

**يبين الجدول 2 توصيف حالة الاستخدام show sales funnel chart:**

|  |  |
| --- | --- |
| Name: | show sales funnel chart |
| Short description: | User can show the funnel chart of a year that he chooses. |
| Precondition: | The user is logged in. |
| Postcondition: | The chart is drawn alongside with the details on a grid. |
| Error situations: | 1. User lost internet connection. |
| System state in the event of an error: | The chart is not shown |
| Actors: | Statistics Manager |
| Trigger: | The user navigates to funnel page. |
| Standard process: | 1. User chooses the year. 2. The chart is drawn. |
| Alternative processes: | (**2`**) The chart is not drawn. |

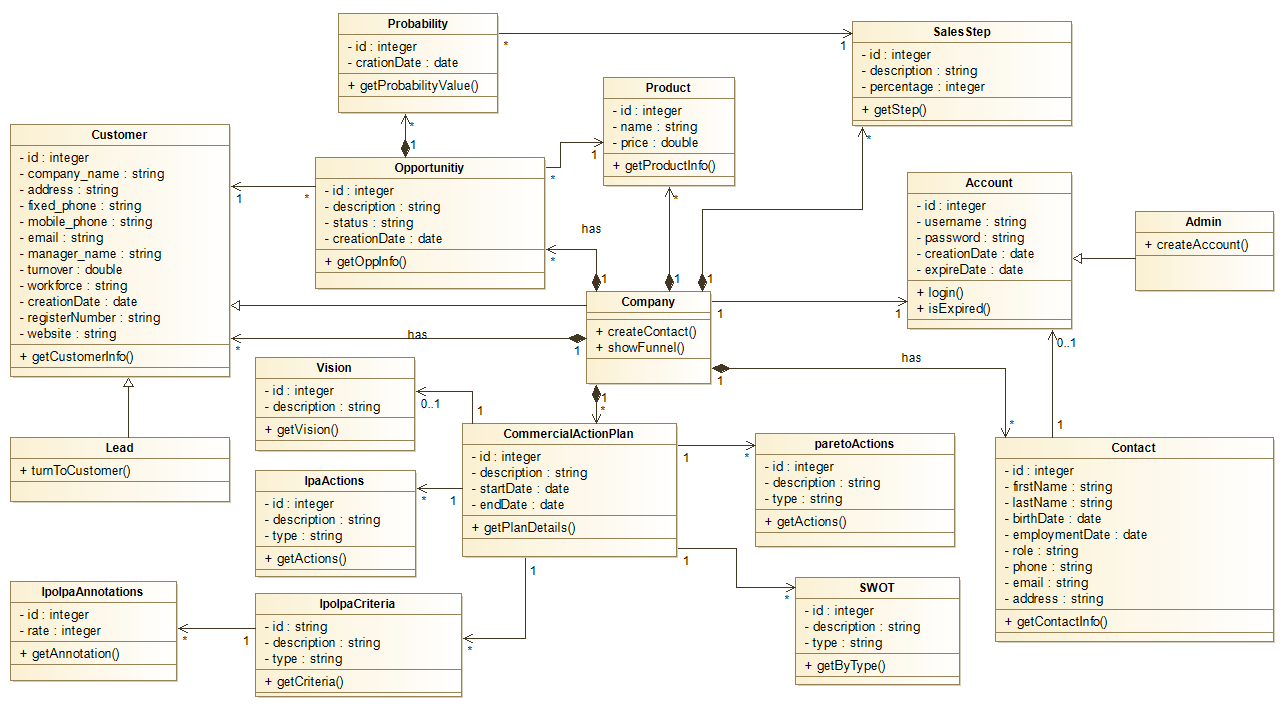
جدول 2. **توصيف حالة الاستخدام show sales funnel chart**

**يبين الجدول 3 توصيف حالة الاستخدام show RFM analysis:**

|  |  |
| --- | --- |
| Name: | Show RFM analysis |
| Short description: | User can show the RFM of customers based on some criteria that the user specifies. |
| Precondition: | The user is logged in. |
| Postcondition: | The customers are listed sorted by the score. |
| Error situations: | 1. User didn’t specify all the criteria. |
| System state in the event of an error: | The result is not shown. |
| Actors: | Statistics Manager. |
| Trigger: | The user clicks Apply button. |
| Standard process: | 1. User specifies all the criteria. 2. The result is shown. |
| Alternative processes: | (2`) error message shown. |

جدول 3. **توصيف حالة الاستخدام show RFM analysis**

### 2.2.2 مخطط الصفوف class diagram:

****

الشكل 6. **مخطط الصفوف**

# التقنيات والبرامج المستخدمة:

## 1.3 تقنيات الـ Frontend:

### 1.1.3 لغة HTML:

**HTML هي اختصار لـ Hyper Text Markup Language (لغة ترميز النص التشعبي) هي اللغة المستخدمة لإنشاء صفحات الويب. HTML ليست لغة برمجة. إنها لغة ترميز، مما يعني أنها نظام لتحديد ووصف المكونات المختلفة لصفحة الويب مثل العناوين والفقرات والقوائم. يشير الترميز إلى الهيكل الأساسي للصفحة (يمكنك التفكير فيه على أنه مخطط تفصيلي مفصل يمكن قراءته آلياً). لا تحتاج إلى مهارات البرمجة -فقط الصبر والحس السليم- لكتابة HTML. [2]**

### 2.1.3 CSS

**بينما يتم استخدام HTML لوصف المحتوى في صفحة الويب، تصف CSS (cascading style sheets) كيف يجب أن يبدو المحتوى. يتم التحكم في جميع الخطوط والألوان وصور الخلفية وتباعد الأسطر وتخطيط الصفحة وما إلى ذلك باستخدام CSS. يمكنك أيضاً إضافة مؤثرات خاصة ورسوم متحركة بسيطة إلى صفحتك. [2]**

### 3.1.3 JavaScript:

**JavaScript هي لغة برمجة تضيف التفاعل والسلوكيات إلى صفحات الويب، بما في ذلك (على سبيل المثال لا الحصر):**

* **التحقق من إدخالات النموذج لإدخالات صالحة.**
* **تبديل تصميم عنصر أو موقع بأكمله.**
* **تحميل يغذي التمرير بالمزيد من المحتوى تلقائياً.**
* **جعل المتصفح يتذكر المعلومات المتعلقة بالمستخدمين.**
* **إنشاء عناصر واجهة مستخدم، مثل مشغلات الفيديو المضمنة أو مدخلات النماذج الخاصة.**

**قد تسمع أيضاً مصطلح DOM المستخدمة فيما يتعلق بجافا سكريبت. يرمز DOM إلى Document Object Model )نموذج كائن المستند(، ويشير إلى القائمة الموحدة لعناصر صفحة الويب التي يمكن الوصول إليها ومعالجتها باستخدام JavaScript (أو لغة برمجة نصية أخرى). [2]**

### 4.1.3 React JS:

* **ما هيReact ؟ [3]**

هي مكتبة JavaScript تعريفية وفعالة ومرنة لبناء واجهات المستخدم. تتيح إنشاء واجهات مستخدم معقدة انطلاقاً من أجزاء صغيرة ومعزولة من التعليمات البرمجية تسمى المكونات "components". نستخدم المكونات لإخبار React بما نريد عرضه على الشاشة. عندما تتغير البيانات، ستقوم React بتحديث المكونات وإعادة عرضها بكفاءة.

يأخذ المكوّن وسطاء تسمى props (اختصاراً لـ "Properties")، ويعيد تسلسلاً هرمياً من المكونات الجزئية لعرضها باستخدام طريقة تسمى render(). تعرض الطريقة render() وصفاً لما سيُعرَض على الشاشة. تأخذ React الوصف وتعرض النتيجة. على وجه الخصوص، تُعيد عنصر React، وهو وصف بسيط لما سيتم عرضه.

يستخدم معظم مطوري React صيغة خاصة تسمى "JSX" مما يجعل كتابة هذه البنى أسهل. تدعم JSX القوة الكاملة لجافا سكريبت. يمكن وضع أي تعبيرات JavaScript ضمن أقواس داخل JSX. كل عنصر من عناصر React هو كائن JavaScript يمكن تخزينه في متغير أو تمريره في البرنامج.

* **ميزات React:**

تعتمد على البرمجة التصريحية Declarative Programming، حيث تسهّل إنشاء واجهات مستخدم تفاعلية. تسمح أيضاً بتصميم طرق عرض بسيطة لكل حالة في التطبيق، وتقوم React بتحديث المكونات الصحيحة فقط وعرضها عندما تتغير البيانات. تجعل المكونات التصريحية الكود أكثر قابلية للتنبؤ وأسهل في التصحيح. [4]

كما تسمح بإنشاء تطبيقات من صفحة واحدة single-page websites. التطبيق أحادي الصفحة هو تطبيق يقوم بتحميل صفحة HTML واحدة وجميع الملفات الضرورية (مثل JavaScript و CSS) المطلوبة لتشغيل التطبيق. لا تتطلب أي تفاعلات مع الصفحة أو الصفحات اللاحقة ذهاباً وإياباً إلى الخادم، مما يعني عدم إعادة تحميل الصفحة. [5]

* **إنشاء مشروع React جديد وتشغيله: [6]**

تعد تقنية Create React App بيئة مريحة لتعلم React، وهي أفضل طريقة لبدء إنشاء تطبيق جديد من صفحة واحدة في React.

تقوم بإعداد بيئة التطوير الخاصة بك بحيث يمكنك استخدام أحدث ميزات JavaScript، وتوفر تجربة مطور رائعة، وتحسن تطبيقك للإنتاج. يجب أن يكون لديك Node> = 14.0.0 و npm> = 5.6 على جهازك. لإنشاء مشروع، قم بتنفيذ الأمر التالي على موجه الأوامر Terminal:

npx create-react-app my\_app

الآن لتشغيل المشروع، ننتقل أولاَ إلى مجلد المشروع الذي تم إنشاءه:

cd my\_app

ثم ننفذ الأمر التالي:

npm start

### 5.1.3 Visual studio code:

Visual Studio Code هو محرر أكواد خفيف ولكنه قوي يعمل على الحاسب ومتاح لأنظمة Windows وmacOS وLinux. يدعم JavaScript و TypeScript و Node.js بشكل افتراضي ولديه نظام غني من الإضافات للغات الأخرى (مثل C++ وC# وJava وPython وPHP وGo و.NET). [7]

## 2.3 تقنيات الـ Backend:

### 1.2.3 قاعدة بيانات MySQL:

**يستخدم نظام قاعدة بيانات** MySQL **بنية خادم-عميل. الخادم،** mysqld**، هو البرنامج الذي يتعامل بالفعل مع قواعد البيانات. لإخبار الخادم بما يجب القيام به، يتم استخدام برنامج عميل ينقل المطلوب عن طريق عبارات مكتوبة بلغة الاستعلام الهيكلية (**SQL**). تتم كتابة برامج العميل لأغراض متنوعة، ولكن كل منها يتفاعل مع الخادم من خلال الاتصال به، وإرسال عبارات** SQL **لإجراء عمليات قاعدة البيانات، وتلقي النتائج. [8]**

### 2.2.3 Java:

**إن** Java **ليست مجرد لغة برمجة. هناك الكثير من لغات البرمجة المتاحة، لكن القليل منها له تأثير كبير.** Java **عبارة عن منصة كاملة، بها مكتبة ضخمة، تحتوي على الكثير من التعليمات البرمجية القابلة لإعادة الاستخدام، وبيئة تنفيذ توفر خدمات مثل الأمان، وقابلية النقل عبر أنظمة التشغيل، وجمع البيانات المهملة تلقائياً. كمبرمج، ستحتاج إلى لغة ذات قواعد بسيطة ودلالات مفهومة (أي ليس** C++). **تمنحك بعض اللغات إمكانية النقل وجمع البيانات المهملة وما شابه، ولكنها لا تحتوي على الكثير من المكتبات، مما يفرض عليك إنشاء مكتبة خاصة بك إذا كنت تريد رسومات رائعة أو الوصول إلى الشبكات أو قاعدة البيانات. حسناً، تحتوي** Java **على كل شيء — لغة جيدة، وبيئة تنفيذ عالية الجودة، ومكتبة ضخمة. هذا المزيج هو ما يجعل** Java **اقتراحاً لا يقاوم للعديد من المبرمجين. [9]**

### 3.2.3 Spring:

* **ما هو Spring؟ [10]**

Spring **يجعل برمجة** Java **أسرع وأسهل وأكثر أماناً للجميع. جعلَ تركيز** Spring **على السرعة والبساطة والإنتاجية إطار عمل** Java **الأكثر شهرة في العالم. يثق المطورون في جميع أنحاء العالم في مكتبات** Spring **المرنة. يقدم** Spring **تجارب مبهجة لملايين المستخدمين النهائيين كل يوم - سواء كان ذلك عبر البث التلفزيوني أو التسوق عبر الإنترنت أو عدداً لا يحصى من الحلول المبتكرة الأخرى. يحتوي** Spring **أيضاً على مساهمات من جميع الأسماء الكبيرة في مجال التكنولوجيا، بما في ذلك** AlibabaوAmazonوGoogleوMicrosoft **والمزيد.**

* **ماذا تفعل Spring؟ [10]**
* Microservices
* Reactive
* Cloud
* Web apps
* Serverless
* Event Driven
* Batch

### 4.2.3 Spring Boot:

**يعد** Spring Boot **أحد أكثر أطر العمل شيوعاً المستخدمة لتطوير البرامج لأي تطبيق مؤسسي. من المفيد تطوير نظام موزع حديث قائم على السحابة مثل تطبيق قائم على الخدمات المصغرة. كان** Spring Boot **أحد المشاريع الرئيسية لفريق** Pivotal **وتم إطلاقه في عام 2013. يُسهل** Spring Boot **إنشاء تطبيقات** Spring **قائمة بذاتها يمكنك تشغيلها. ميزتها هي إمكانية البدء بأقل قدر من الإعدادات. تحتاج معظم تطبيقات** SpringBoot **إلى القليل جداً من إعدادات** Spring**. [11]**

### 5.2.3 MySQL Workbench:

**يعد** MySQL Workbench **أداة مرئية موحدة لمهندسي قواعد البيانات والمطورين وموظفي قواعد البيانات. يوفر** MySQL Workbench **نمذجة البيانات وتطوير** SQL **وأدوات الإدارة الشاملة لتكوين الخادم وإدارة المستخدم والنسخ الاحتياطي وغير ذلك الكثير. يتوفر** MySQL Workbench **على أنظمة التشغيل** Windows **و** Linux **و** Mac OS X**. [12]**

**يمكن تحمليه من الرابط التالي:** [**https://dev.mysql.com/downloads/workbench/**](https://dev.mysql.com/downloads/workbench/) **.**

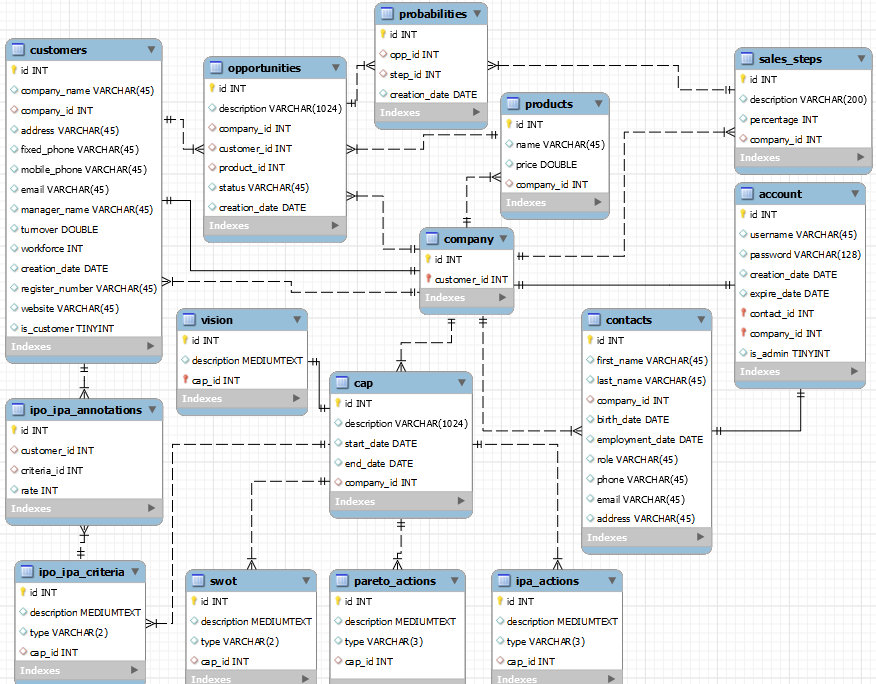
### 6.2.3 Spring Tool Suit (STS)

**يجعل** Spring Tool Suite 4 **من السهل البدء. يتيح لك التكامل المباشر والسهل الاستخدام للانتقال من لا شيء إلى تطبيق** Spring Boot **قيد التشغيل في ثوانٍ. [13]**

**يمكن تحمليه من الرابط التالي:** [**https://spring.io/tools**](https://spring.io/tools) **.**

# تصميم قاعدة البيانات Database Design:

**تم إنشاء قاعدة البيانات باستخدام نظام إدارة قاعدة البيانات MySQL. يبين الشكل 6 مخطط كيان-علاقة لقاعدة البيانات (ERD).**

****

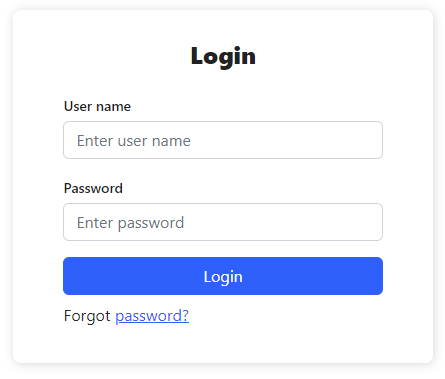
الشكل 7. **مخطط ERD**

الجداول هي:

* **accounts:** id, username, password, creation\_date, expire\_date, contact\_id, company\_id, is\_admin
* **customers:** id, company\_name, company\_id, address, fixed\_phone, mobile\_phone, email, manager\_name, turnover, workforce, creation\_date, register\_number, website, is\_customer
* **company:** id, customer\_id
* **contacts:** id, first\_name, last\_name, company\_id, birth\_date, employment\_date, role, phone, email, address
* **products:** id, name, price, company\_id
* **sales\_steps:** id, description, percentage, company\_id
* **opportunities:** id, description, company\_id, customer\_id, product\_id, status, creation\_date
* **probabilities:** id, opp\_id, step\_id, creation\_date
* **cap:** id, description, start\_date, end\_date, company\_id
* **vision:** id, description, cap\_id
* **swot:** id, description, type, cap\_id
* **pareto\_actions:** id, description, type, cap\_id
* **ipa\_actions:** id, description, type, cap\_id
* **ipo\_ipa\_criteria:** id, description, type, cap\_id
* **ipo\_ipa\_annotations:** id, customer\_id, criteria\_id, rate

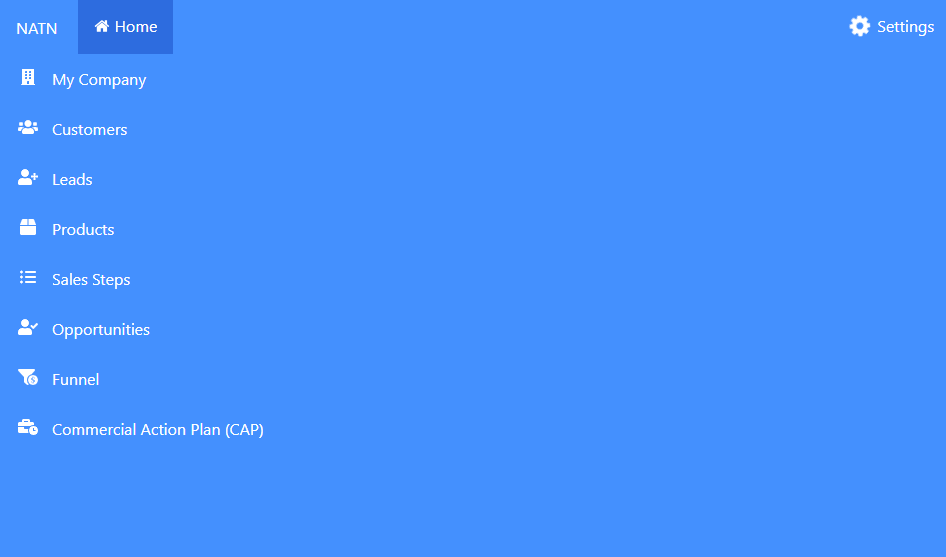
# صفحات الموقع الأساسية:

## 1.5 صفحة تسجيل الدخول login:



الشكل 8. **صفحة تسجيل الدخول**

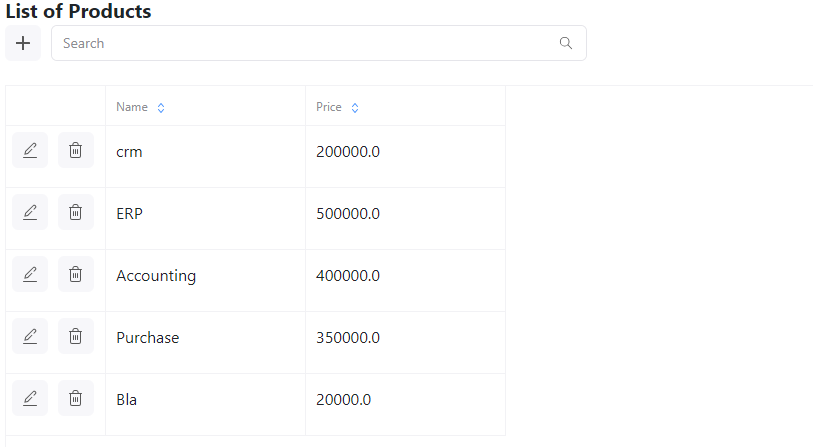
## 2.5 الصفحة الرئيسية للموقع Home Page:

****

الشكل 9. **الصفحة الرئيسية**

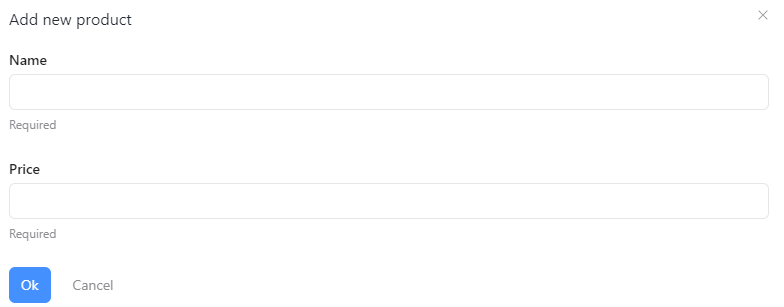
## 3.5 صفحة المنتجات:

**الشكل 10 يبين المنتجات الخاصة بالمستخدم، حيث يمكنه إضافة أو حذف أو تعديل منتج أو البحث عن منتجات.**

****

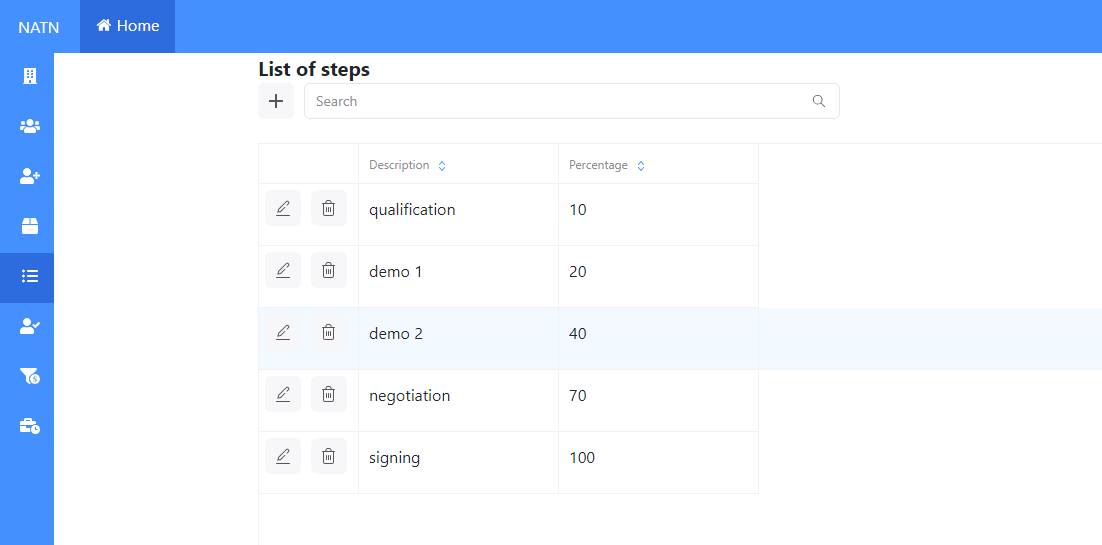
الشكل 10. **صفحة المنتجات**

**الشكل 11 يبين فورم إضافة منتج جديد:**

****

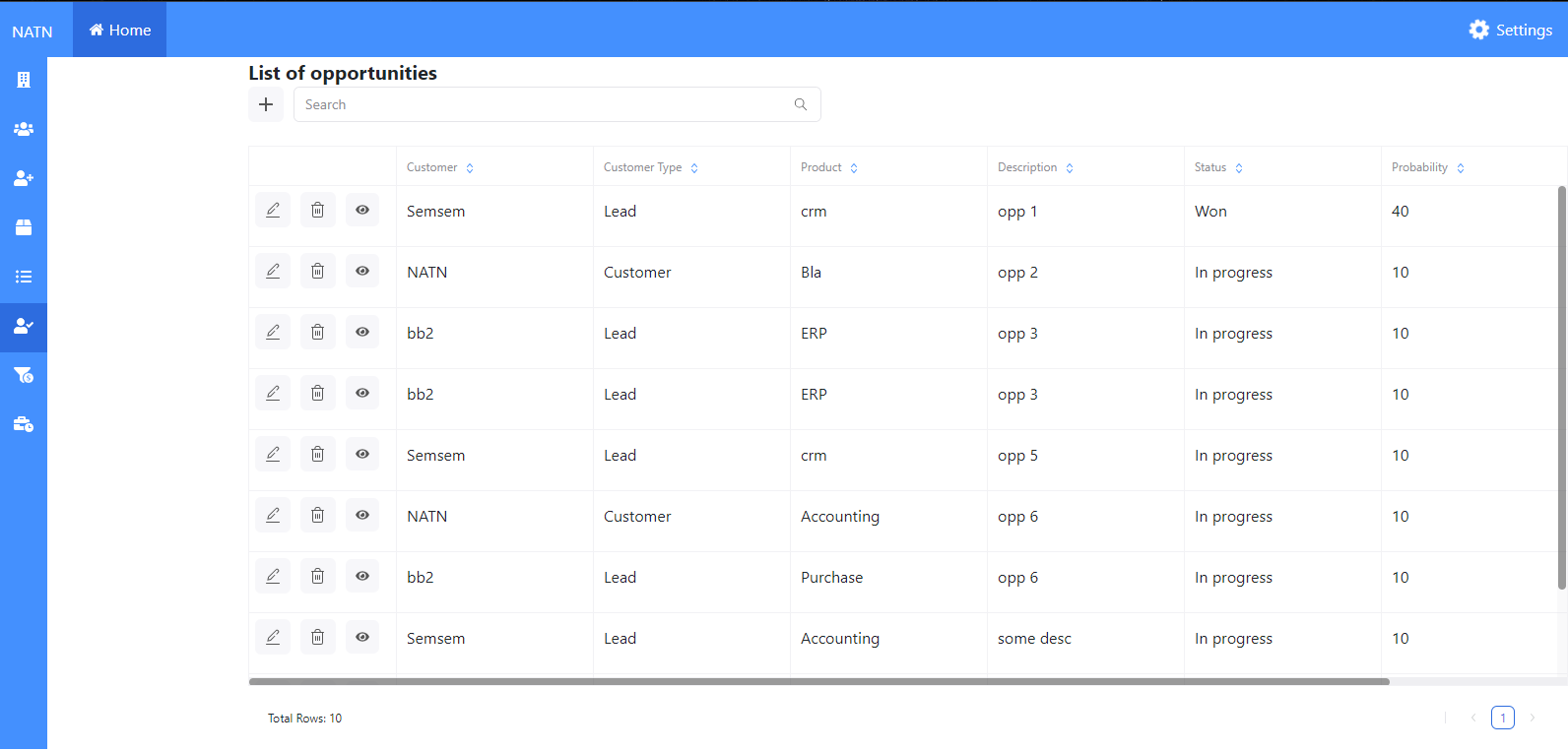
**الشكل 11. إضافة منتج**

## 4.5 صفحة خطوات البيع sales steps:

****

**الشكل 12. خطوات البيع**

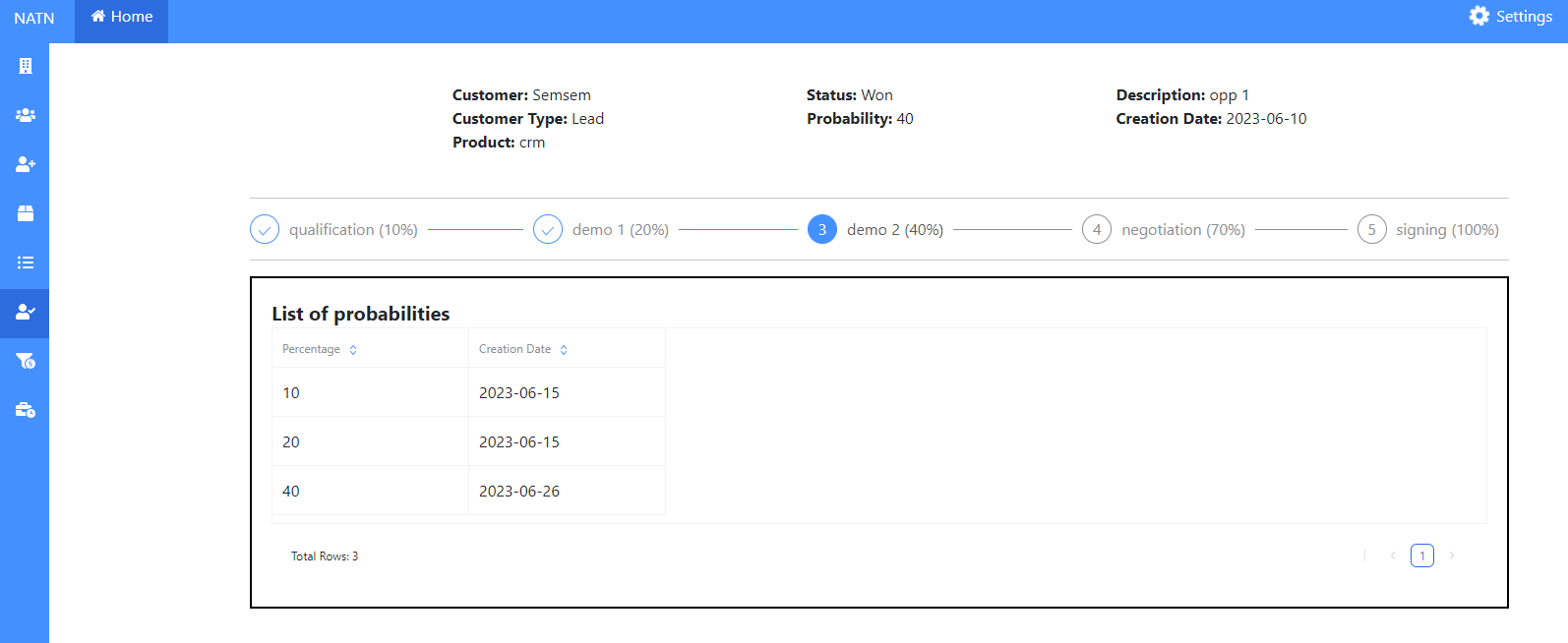
## 5.5 صفحة الـ opportunities:

****

**الشكل 13. Opportunities**

**حيث يمكن عرض تفاصيل كل opportunity بالضغط على الزر  .**

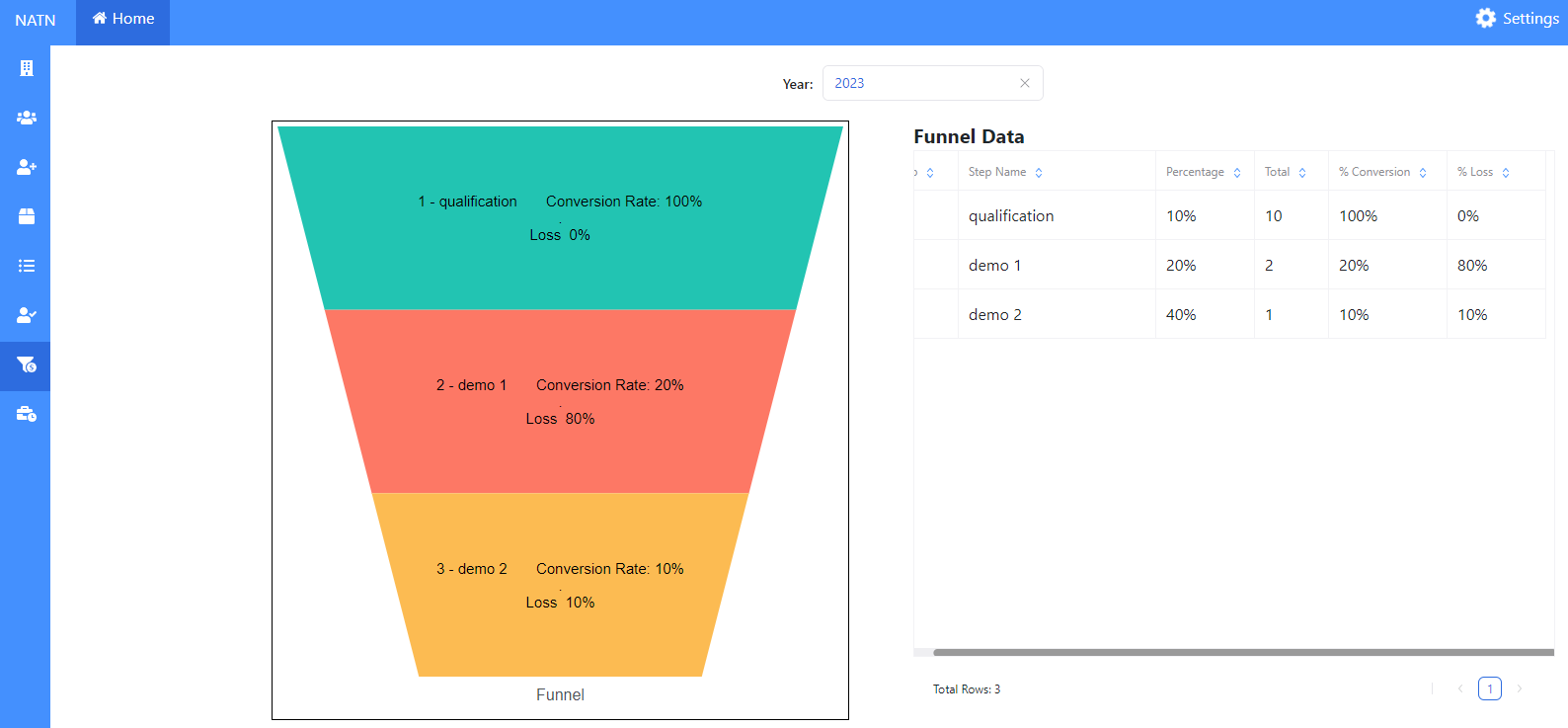
**عند الضغط على الزر السابق تظهر صفحة تفاصيل الـ opportunity، حيث يتم عرض المعلومات الخاصة بها و مسار الخطوات، بالإضافة إلى جميع الـخطوات التي مر بها سابقاً. كما هو مبين في الشكل 14.**

****

الشكل 14. **تفاصيل الـ opportunity**

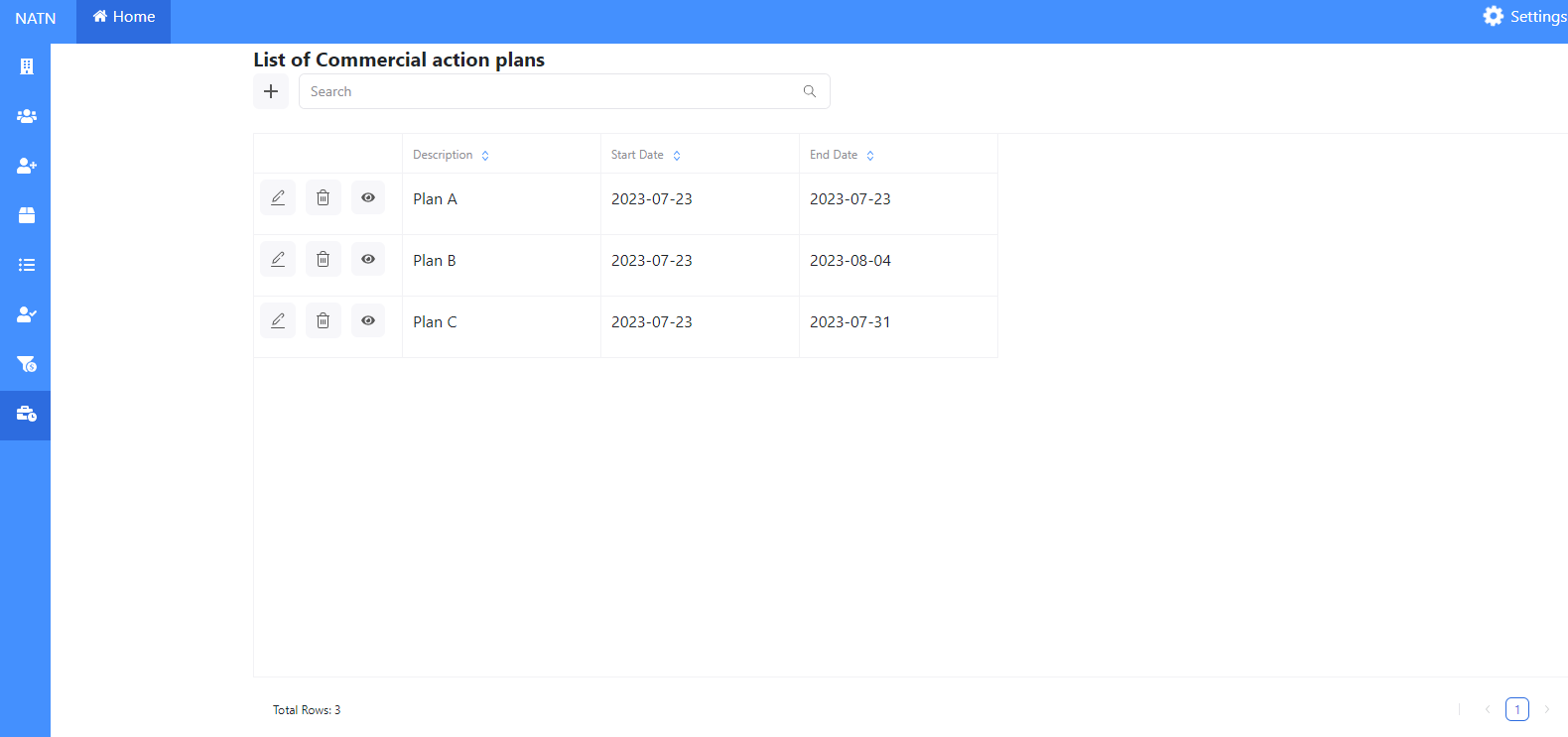
## 6.5 صفحة مخطط القمع Funnel Chart:

**يتم هنا عرض قمع المبيعات مع تفاصيله، ويمكن اختيار السنة المطلوب عرض المخطط لها.**

****

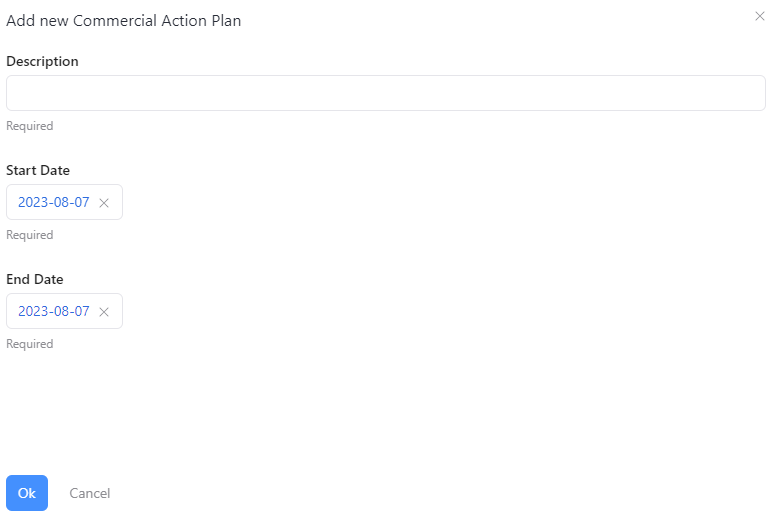
الشكل 15. **مخطط قمع المبيعات**

## 7.5 صفحة خطط العمل التجارية:

****

الشكل 16. **خطط العمل التجارية**

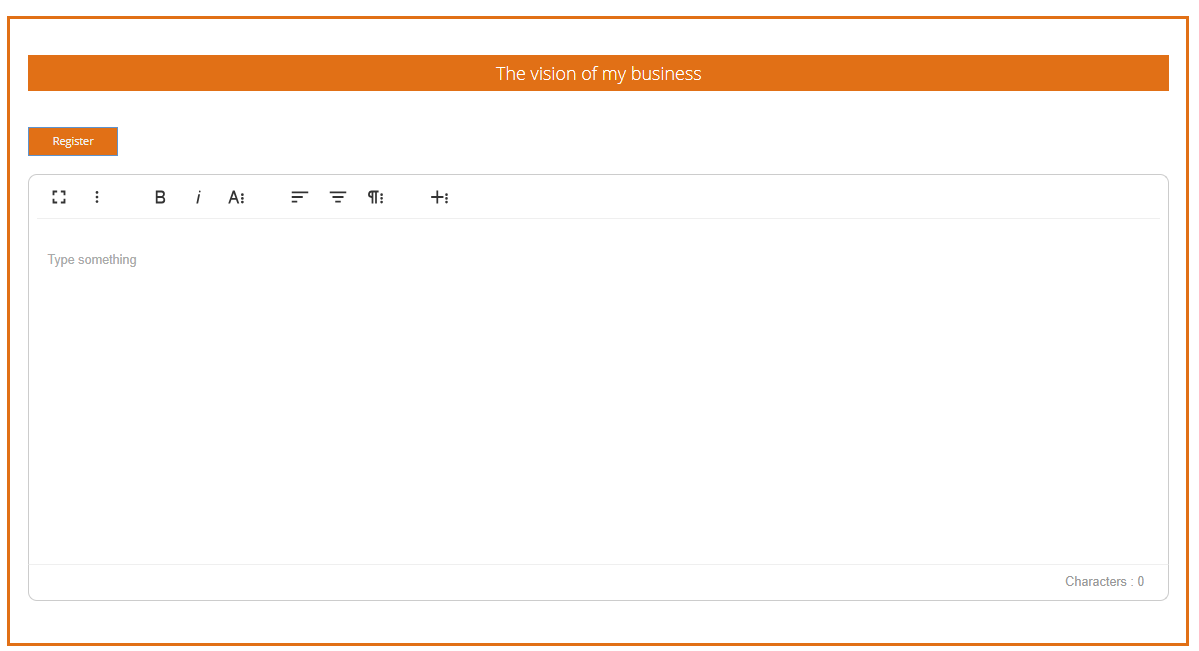
**لإضافة خطة عمل تجارية، يجب تحديد تاريخ البداية والنهاية التي تريد القيام بالقيام بالتحليلات عليها، حيث يتم عرض التحليلات ضمن هذه الفترة الزمنية.**

****

الشكل 17. **إضافة خطة عم لتجارية جديدة**

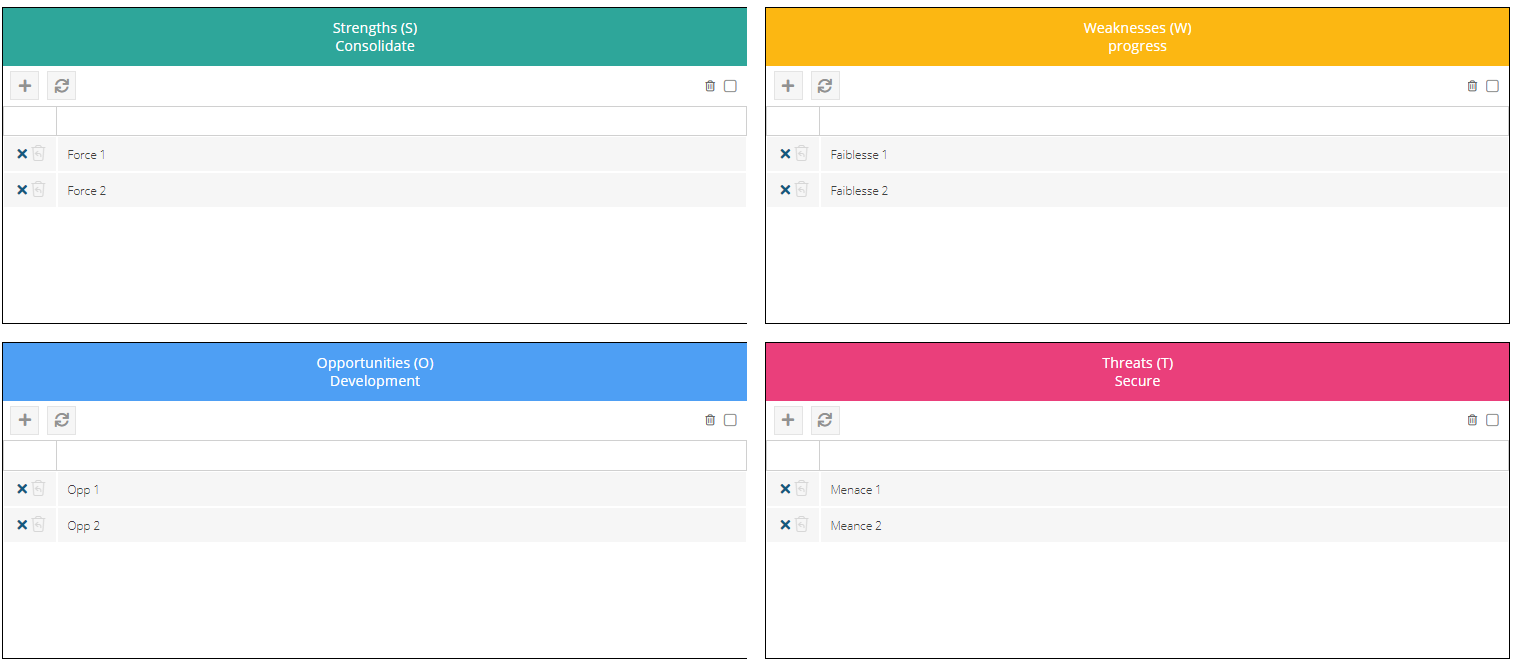
## 8.5 صفحة تفاصيل خطة العمل التجارية:

**هنا لدينا 5 تبويبات. أولاً يقوم المستخدم بتحديد رؤية خطة العمل. كما في الشكل 18.**

****

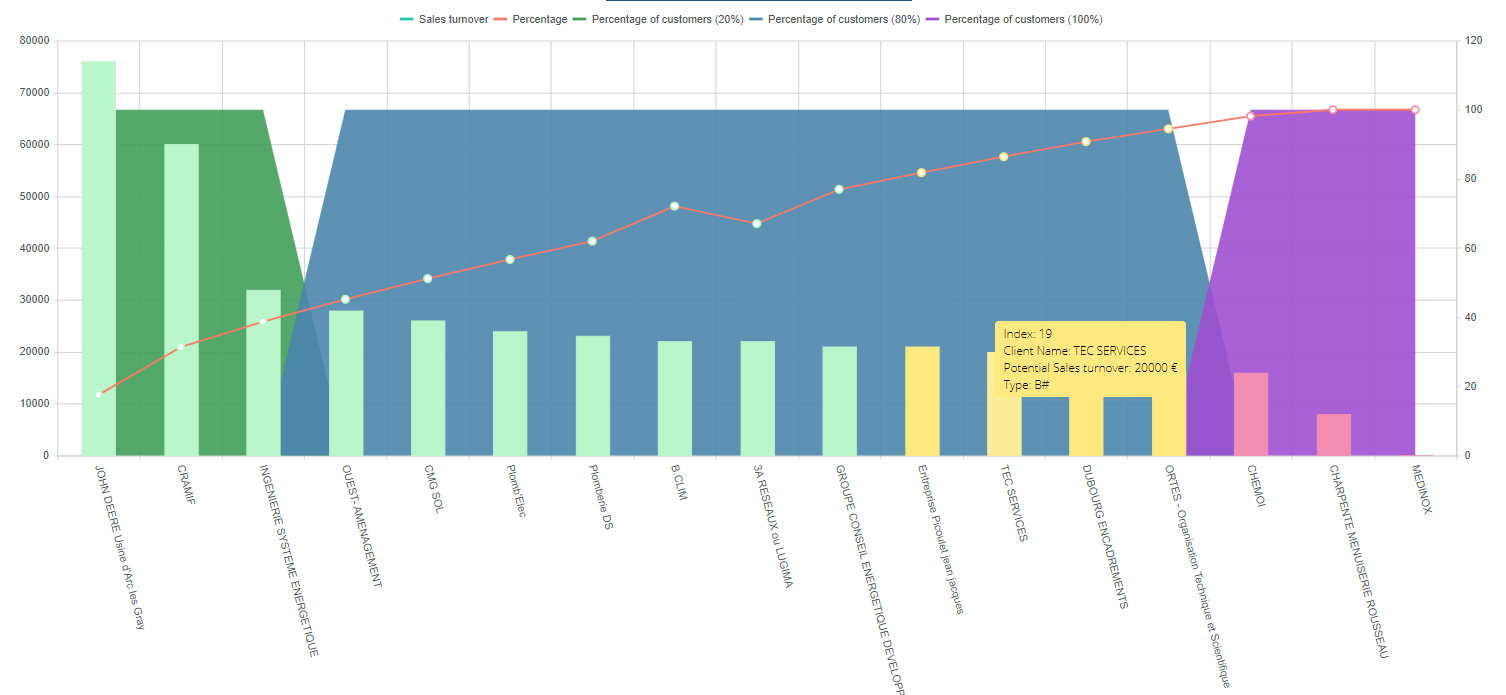
الشكل 18. **رؤية خطة العمل**

**ثانياً يمكنه القيام بتحليل swot وذلك لدراسة الوضع الراهن للبيئة الداخلية والخارجية.**

****

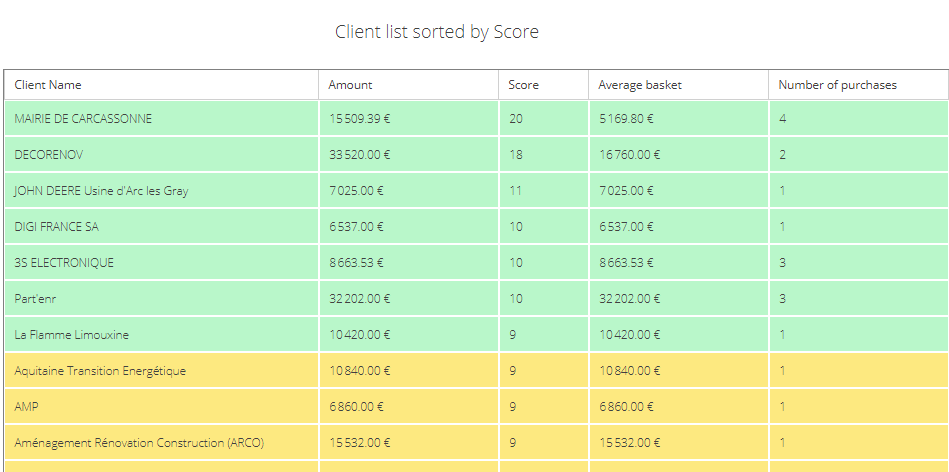
الشكل 19. **تحليل swot**

**ثالثاً يتم عرض نتائج تحليل pareto بالإضافة إلى مخطط ABC. الشكل 20 يبين مخطط ABC.**

****

الشكل 20. **مخطط ABC**

**رابعاً يتم تطبيق تحليل RFM حيث يقوم المستخدم بتحديد الفترة الزمنية المراد الدراسة على أساسها (كل شهر، كل ثلاثة أشهر، كل نصف سنة، كل سنة). يختار أيضاً قيمة كل من الحد الأدنى لمتوسط عدد مرات الشراء الذي يجعل الزبون جيداً، بالإضافة إلى الحد الأدنى من القيمة المالية التي حصلنا عليها من الزبون. حيث يتم ترتيب الزبائن من الأفضل للأسوأ حسب المعايير المحددة سابقاً.**

****

الشكل 21. **الزبائن مرتبين حسب تقييم RFM**

# الخاتمة:

كان هذا المشروع يهدف بشكل أساسي إلى تطوير موقع ويب لإدارة علاقات العملاء CRM موجه للشركات الصغيرة ومتوسطة الحجم. تشمل الوظائف الرئيسية تسجيل الدخول، وإدارة الزبائن الحاليين والمحتملين والموظفين والمنتجات وتحديد خطوات البيع. وأيضاً إدارة الـ opportunities وتتبعها. وعرض مخطط قمع المبيعات. وأخيراً دراسة خطط العمل التجارية من تحليل الوضع الراهن باستخدام swot إلى تحليل باريتو و RFM لتحديد أفضل العملاء.

بعد اكتمال تطوير المشروع، تم تحقيق الوظائف المتضمنة في المتطلبات الأساسية للنظام. تم اختبار التطبيق على متصفحات مختلفة. أثناء إعداد المشروع، تم تلقي فهم عميق لتعلم Java و JavaScript. في الوقت نفسه، علمنا أننا بحاجة إلى مزيد من الدراسات والتحسينات المتقدمة. نظراً لقيود الوقت والتعلم، لا تزال العديد من جوانب التطبيق ووظائفه بحاجة إلى التحسين. في الدراسة المستقبلية، سيتم تحسين هذه الجوانب والوظائف وتحسينها لجعل التطبيق أكثر احترافاً واكتمالاً.

# مقترحات لتطوير المشروع:

على الرغم من أنه تمت تلبية المتطلبات الأساسية للنظام، إلا أنه يمكن تحسين العديد من جوانبه. يمكن إضافة الوظائف التالية لتحسين النظام:

* إدارة المهام tasks واستخدام Calendar لعرضها بشكل زمني.
* التسويق باستخدام الإيميل Email Marketing.

# المراجع:

[1] ABC Analysis in Inventory Management: Benefits & Best Practices.

Accessed 6.8.2023

<https://www.netsuite.com/portal/resource/articles/inventory-management/abc-inventory-analysis.shtml>

[2] Robbins, J. N. (2018). *Learning Web Design: A Beginner’s Guide To Html, CSS, JavaScript, And Web Graphics* (Fifth Edition). ISBN: 978-1-491-96020-2. O’Reilly Media.

[3] What is React? Accessed 3.8.2023

<https://reactjs.org/tutorial/tutorial.html#what-is-react>

[4] React: A JavaScript library for building user interfaces. Accessed 3.8.2023

<https://reactjs.org/>

[5] Single Page Application. Accessed 3.8.2023

<https://reactjs.org/docs/glossary.html#single-page-application>

[6] Create React App. Accessed 3.8.2023

<https://reactjs.org/docs/create-a-new-react-app.html#create-react-app>

[7] Visual Studio Code. Accessed 3.8.2023

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

[8] DuBois, P. (2014). *MySQL Cookbook* (Third Edition). ISBN: 978-1-449-37402-0. O’Reilly Media.

[9] Horstmann, C. S. (2019). *Core Java: Volume I-Fundamentals* (Eleventh Edition). ISBN: 978-0-13-516630-7. Pearson Education.

[10] Why Spring? Accessed 3.8.2023

<https://spring.io/why-spring>

[11] Rajput, D. (2019). *Designing Applications with Spring Boot 2.2 and React JS: Step-by-step guide to design and develop intuitive full stack web applications* (First Edition). ISBN: 978-93-88511-643. BPB Publications.

[12] MySQL Workbench. Accessed 3.8.2023

<https://www.mysql.com/products/workbench/>

[13] Spring Tools 4. Accessed 3.8.2023

<https://spring.io/tools>