

Rapport de stage

TITRE DE MÉMOIRE :

CRM (GESTION DE LA RELATION CLIENT)

RÉALISÉ PAR:

HAMZA GHOUNBAZ

SOUKAINA FILALE

NOM DE ENTREPRISE :

DEVOSOFT

DATE DE STAGE DE FIN D'ETUDES :

01/07/2020- 30/08/2020

1

ENCADRÉ PAR:

HICHAM RGUINA

- **Introduction générale CRM.....3**
- **Définition de société DevoSOFT.....4**
- **Cahier des charges Projet CRM.....5**
- **Étude conceptuel : diagramme de classe.....7**
- **Problématique et solution.....8**
- **Definition de compétence: html - css - composer - mysql – PHP.....9**
- **Definition de MVC Laravel : Controller – Models – View – Route et Migration.....11**
- **Les capture d’écran.....12**
- **Canclusion.....33**

La Gestion de la Relation Client - GRC / CRM

Un logiciel de Gestion Relation Client (GRC / CRM) – Customer Relationship Management en anglais ou CRM – est une base de données clients qui permet à une entreprise d’avoir une vision claire et constante de ses clients et futurs clients (prospects).

Il représente donc l’ensemble des outils qui permettent d’identifier les prospects, puis traiter, analyser et fidéliser les clients.

Le but étant d’assurer par la suite un suivi personnalisé, en offrant la meilleure qualité service possible.

Cet outil permet donc de placer le client au cœur de l’entreprise, en étant à son écoute et en ayant une meilleure connaissance sur ses attentes et ses besoins.

Il permet également de mieux cibler sa clientèle, optimiser son temps de travail tout en augmentant la réactivité de l’entreprise et en dynamisant sa productivité.

DEVOSOFT

Devosoft est une agence web basée au Maroc à Safi, Spécialisée dans la création, le suivi et la maintenance de sites vitrine, sites e-commerce, et des sites sur-mesure, notre entreprise assure, dans son intégralité, le pilotage de projets divers confiés par nos clients.

Faire des sites de ses partenaires des références sur internet est devenu la spécialité de Devosoft et nous y attachons tous nos efforts. Parce que travailler en équipe permet d'être encore plus efficace pour vous fournir les solutions les plus innovantes, nous avons réuni une équipe très compétente au professionnalisme reconnu. Forces de proposition, efficaces et créatifs, nous vous accompagnons dans la définition de votre projet, la conception de votre stratégie et jusqu'à l'atteinte de vos objectifs. Grâce à notre personnel qualifié en développement web, webdesign, référencement naturel, Google Adwords et rédaction de contenus optimisés, nous regroupons tous les compétences nécessaires pour la réalisation de vos ambitions.

Cahier des charges

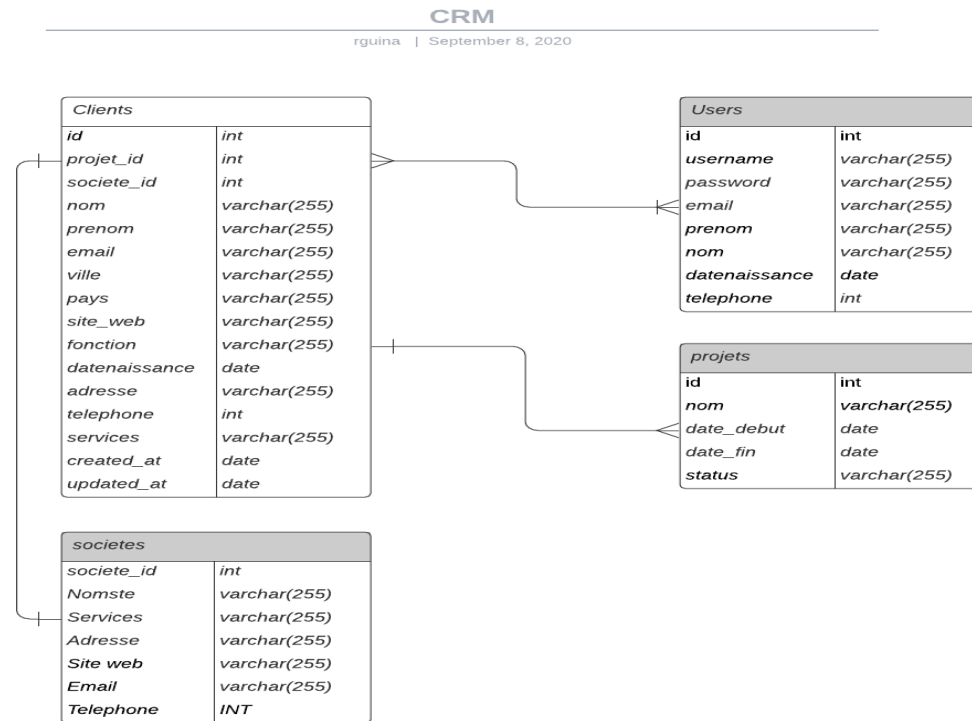
Les Technologie Utilisé a Projet CRM :

- UI / UX (Figma)
- UML de Classe Projet CRM
- HTML / CSS
- Bootstrap 4
- RESPONSIVE
- Laragon Server avec HeidiSQL Type MYSQL MariaDB
- Composer (Un gestionnaire de dépendances pour PHP)
- Framework Laravel 7 PHP Version 7

Les Pages include :

- Page Login avec Email et option auth avec Gmail
- Page Cree un nouveau compte
- Page Pour Récupération le mot de pass via Email
- Page Controle Panel Pour Voir et Gere Information Client
- Page Pour Ajouté et modifier et listé Les Clients
- Page Pour Ajouté et modifier et listé Projet
- Page Pour Ajouté et modifier et liste Societé

Diagramme de classe



Problématique et Solution

concernant les problèmes qu'on a rencontré avec cette entreprise , on a trouvé que cette dernière avait des obstacles qui l'empêcher d'augmenter ses ventes , ainsi qu'une mauvaise liaison avec ses clients et la gestion de service après-vente . pour cela on a décidé de faire un CRM ou GRC pour faire face à ces problèmes que rencontre cette société . la création de ce programme CRM va permettre à l'entreprise d'automatiser la gestion de vente à travers des outils tels que l'entreprise marketing autonome , aussi à travers l'instauration de centres d'appels chargés de l'assistance à distance des clients , le CRM permet à l'entreprise d'être à l'écoute de ses clients et de répondre à leurs questionnements , et finalement la GRC va permettre à l'entreprise d'augmenter ces ventes à travers de nouveaux outils qui permettent d'améliorer l'efficacité des vendeurs

le site web CRM facilite la gestion de relations clients complexes. où toutes les informations disponibles sur les contacts d'une entreprise sont collectées et gérées dans des bases séparées. Les utilisateurs ne sont pas obligés de se limiter aux clients. aussi gérer le temps , faciliter la recherche de client de chaque projet , Ils peuvent également saisir tous les contacts importants pour l'entreprise, par exemple des fournisseurs, des partenaires ou des influenceurs.

Definition de compétence

HTML

L'HyperText Markup Language, HTML, désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web. Il permet, entre autres, d'écrire de l'hypertexte, mais aussi d'introduire des ressources multimédias dans un contenu.

L'HTML est ce qui permet à un créateur de sites Web de gérer la manière dont le contenu de ses pages Web va s'afficher sur un écran, via le navigateur. Il repose sur un système de balises permettant de titrer, sous-titrer, mettre en gras, etc., du texte et d'introduire des éléments interactifs comme des images, des liens, des vidéos...

L'HTML est plus facilement compris des robots de crawl des moteurs de recherche que le langage JavaScript, aussi utilisé pour rendre les pages plus interactives.

CSS

CSS est l'acronyme de « Cascading Style Sheets » ce qui signifie « feuille de style en cascade ».

Le CSS correspond à un langage informatique permettant de mettre en forme des pages web (HTML ou XML).

Ce langage est donc composé des fameuses « feuilles de style en cascade » également appelées fichiers CSS (.css) et contient des éléments de codage.

Composer

Composer est un logiciel gestionnaire de dépendances libre écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin. Le développement a débuté en avril 2011 et a donné lieu à une première version sortie le 1er mars 2012. Développé au début par Nils Adermann et Jordi Boggiano (qui continuent encore aujourd'hui à le maintenir), le projet est maintenant disponible sur la plateforme GitHub. Il est ainsi développé par toute une communauté.

MYSQL

MySQL est un serveur de bases de données.

Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données.

PHP

Le langage PHP est un langage de programmation qui signifie "Hypertext Preprocessor". Le langage PHP permet de produire des pages web dynamiques via un serveur HTTP.

Definition « Laravel MVC »

Laravel

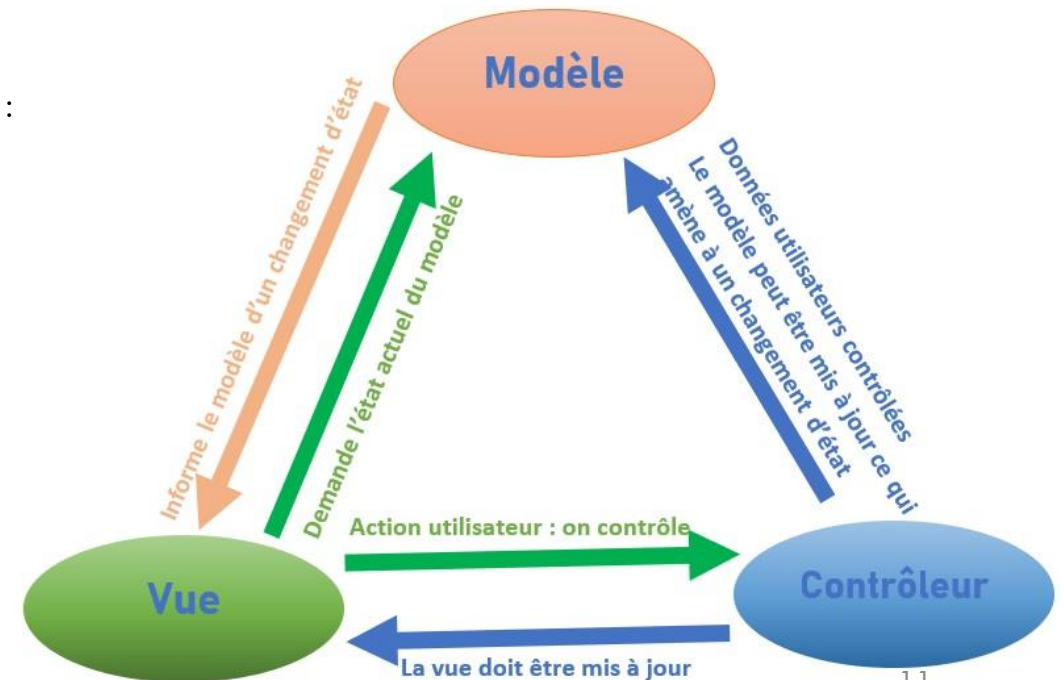
Laravel est un framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub.

Laravel MVC

Le MVC (Model View Controller) est une méthode d'organisation du développement d'applications Web permettant de séparer les différents concepts résultant de nos pages PHP.

En effet, trois grandes « actions » peuvent être identifiées lorsque nous développons en PHP :

- Les requêtes en base de données (Model)
- Le traitement des données (Controller)
- L'affichage de pages HTML (View)



Laravel « Install Laravel »

Comment installer Laravel ?

```
C:\laragon\www  
^ composer create-project --prefer-dist laravel/laravel crm-projet|
```

- composer : lance l'outil composer
- create-project : demande de créer un projet
- --prefer-dist : demande d'installer la dernière version stable
- laravel/laravel : packagist framework laravel
- crm-projet : chemin vers le dossier du projet (par rapport au dossier en cours). ici Laravel sera installé dans un dossier crm-projet créé directement dans le dossier laragon\www

Laravel « Laravel fichiers »

Structure de dossiers de Laravel

```
> app
> bootstrap
> config
> database
> node_modules
> public
> resources
> routes
> storage
> tests
> vendor
> vendor1
⚙ .editorconfig
⚙ .env
💎 .gitattributes
💎 .gitignore
! .styleci.yml
≡ artisan
{} composer.json
{} composer.lock
{} package.json
{} package-lock.json
📄 phpunit.xml
📖 readme.md
🐘 server.php
JS webpack.mix.js
```

Env fichier avec la connection DB

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:+riBS2YXBJo38MaEYPlnSAiSe4T+NiwXSmd1qKp3Ig8=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

FILESYSTEM_DRIVER=public

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=crm
DB_USERNAME=root
DB_PASSWORD=

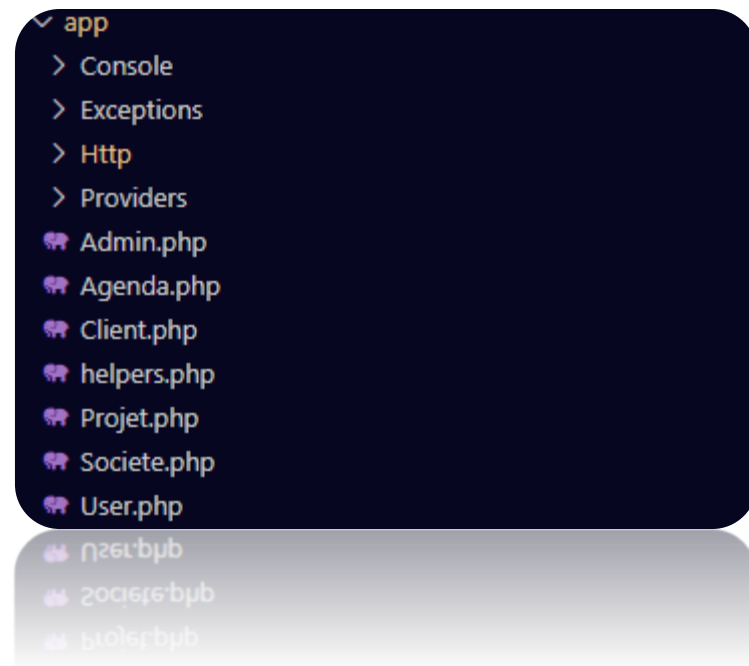
BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
```

Les Modèles (Models)

Ce que nous appelons un Modèle est en réalité un fichier PHP qui ne fait que gérer les échanges avec la base de données. Lorsque nous avons besoin de lire ou écrire dans la base de données, nous faisons appel au Modèle.

Nous avons créé 7 Models :

Admin , Agenda , Client , Helper , Projet , Societe , User



User Model

```
class User extends Authenticatable
{
    use Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    /**
     * The attributes that should be cast to native types.
     *
     * @var array
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}
```

Les Contrôleurs (Controllers)

Véritable tour de contrôle de notre application, le contrôleur a pour fonction de faire l'interface entre les modèles et les vues. Il est chargé de demander les données par l'intermédiaire des modèles, de traiter ces données et de les transmettre aux vues, prêtes à être utilisées.

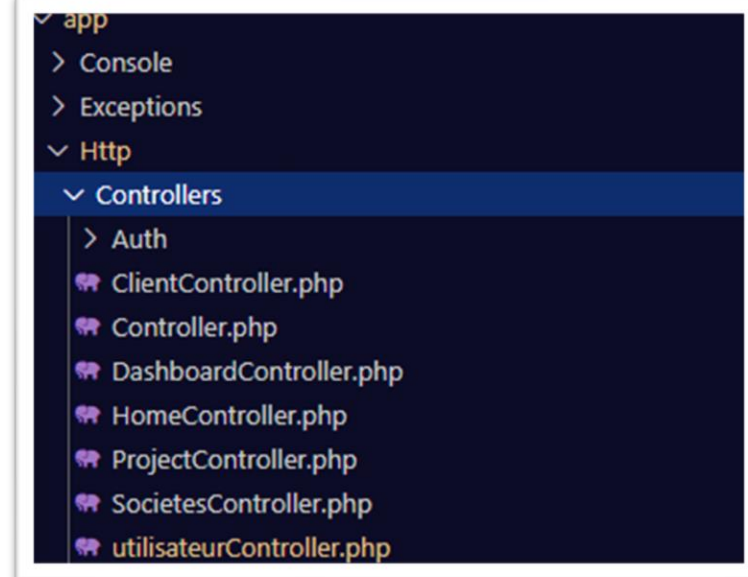
Pour Cree un Contrôleur

```
C:\laragon\www
λ php artisan make:controller Client|
```

Nous avons créé 6 Models :

Client , Dashboard , Home , Project , Societe , Utilisateur

App/http/Controller



Il y a 7 Function pour « CRUD »

- Index : Liste des Clients
- Create : Affichage du formulaire d'ajout
- Edit : l'affichage du formulaire de modification
- Show : l'affichage d'un seule enregistrement
- Store : l'ajout dans la DB
- Update : la modification dans la base de donnée
- Destroy : la suppression

Nous commencerons par le premier Contrôleur

ClientController

Function Authentication

```
public function __construct()
{
    $this->middleware('auth');
}
```

Cette fonction Permet login pour Accéder la Page

Fuction Recupere Data via DB et Affiche

```
public function index()
{
    $clients=Client::all();
    $projets=Projet::all();
    $societes=Societe::all();
    return view('pages.clients.index',compact('clients','projets','societes'));
}
```

Function Create

```
public function create()
{
    $projets=Projet::all();
    $societes=Societe::all();
    return view('pages.clients.add',compact('projets','societes'));
}
```

Cette fonction Permet affiche pour create client

Laravel « Contrôleurs »

Client Information

- Nom
- Prenom
- Email
- La date naissance
- Telephone
- Fonction
- Ville
- Services
- Adresse
- Site Web
- Project
- Societe
- Pays

Function Store

```
public function store(Request $request)
{
    $client = new Client();

    if($request->hasFile('image')){
        $path = $request->file('image')->store('clients');
        $client->image = $path;
    }

    $client->nom = $request['nom'];
    $client->prenom = $request['prenom'];
    $client->email = $request['email'];
    $client->datenaissance = $request['date'];
    $client->telephone = $request['tel'];

    $client->fonction = $request['function'];
    $client->ville = $request['ville'];
    $client->services = $request['services'];
    $client->adresse = $request['adresse'];
    $client->site_web = $request['site_web'];
    $client->projet_id = $request['projet_id'];
    $client->societe_id = $request['societe_id'];
    $client->pays = $request['pays'];

    $client->save();
    session()->flash('succes', 'Client ajouter avec succès');
    return redirect()->back();
}
```

Cette fonction Permet Ajoute Information sur DB

Laravel « Contrôleurs »

Function Recuper et Affiche Page Edit

```
public function edit($id)
{
    $client=Client::find($id);
    $projets=Projet::all();
    $societes=Societe::all();
    return view('pages.clients.edit',compact('client','projets','societes'));
}
```

Function Pour Supprimé

```
public function destroy($id)
{
    $client = Client::find($id);
    $client->delete();
    session()->flash('succes','Client supprimer avec succès');
    return redirect()->back();
}
```

Function modifier information

```
public function update(Request $request, $id)
{
    $client =Client::find($id);

    if($request->hasFile('image')){
        $path = $request->file('image')->store('clients');
        $client->image = $path;
    }

    $client->nom = $request['nom'];
    $client->prenom = $request['prenom'];
    $client->email = $request['email'];
    $client->datenaissance = $request['date'];
    $client->telephone = $request['tel'];

    $client->fonction = $request['fonction'];
    $client->ville = $request['ville'];
    $client->services = $request['services'];
    $client->adresse = $request['adresse'];
    $client->site_web = $request['site_web'];
    $client->projet_id = $request['projet_id'];
    $client->societe_id = $request['societe_id'];
    $client->pays = $request['pays'];

    $client->save();
    session()->flash('succes','Client modifier avec succès');
    return redirect()->back();
}
```

Authentication

Action d'authentifier, procédure visant à certifier l'identité de quelqu'un ou d'un ordinateur afin d'autoriser le sujet d'accéder à des ressources.

Function Login

```
protected $redirectTo = '/home';

/**
 * Create a new controller instance.
 *
 * @return void
 */
public function __construct()
{
    $this->middleware('guest')->except('logout');
}
```

Function Register

```
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);
}
```

Laravel « Migration »

Migration

Les migrations. Une migration permet de créer et de mettre à jour un schéma de base de données.

Autrement dit vous pouvez créer des tables, des colonnes dans ces tables, en supprimer, créer des index...

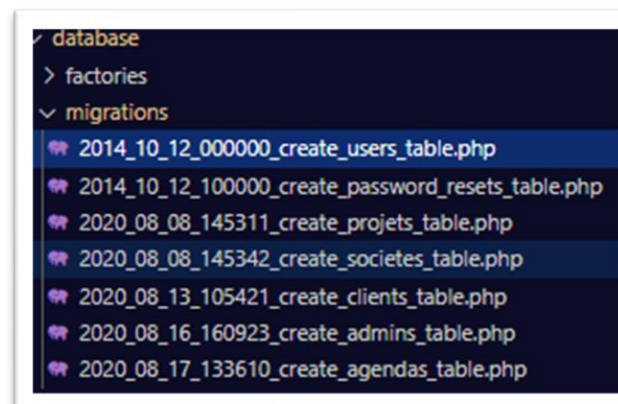
Tout ce qui concerne la maintenance de vos tables peut être prise en charge par cet outil.

Bash :

```
C:\laragon\www
λ php artisan make:migration create_users_table
```

Pour Cree database utilisé cmd :

```
C:\laragon\www
λ php artisan make:migrate
```



Crée les table de user

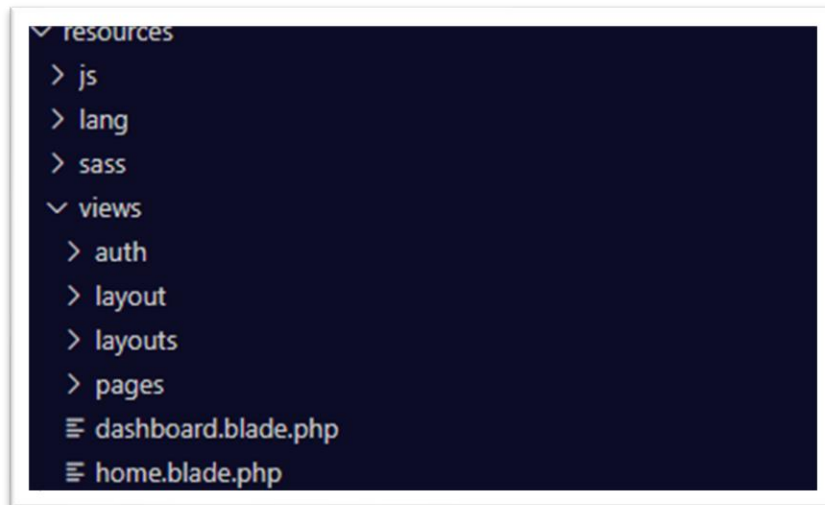
```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->bigIncrements('id');
        $table->string('name')->nullable();
        $table->string('password');
        $table->string('email')->unique();
        $table->string('prenom')->nullable();
        $table->string('nom')->nullable();
        $table->date('datenaissance')->nullable();
        $table->bigInteger('telephone')->nullable();
        $table->string('image')->nullable();
        $table->string('role')->default('User');
        $table->rememberToken();
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('users');
}
```


Les Vues (Views)

Les vues sont principalement des fichiers HTML contenant le code destiné à être transmis au navigateur de l'utilisateur. Nous disposerons des données transmises par le contrôleur afin de les intégrer dynamiquement dans nos pages.

/Ressource/Views



Ce dossier pour les fichiers laravel a ext .blade.php

/public



Ce dossier pour les fichiers bootstrap et css & images

Les Vues (Dashboard)

Page

```

<div class="float-right">
  <p class="mb-0 text-right">Utilisateurs</p>
  <div class="fluid-container">
    <h3 class="font-weight-medium text-right mb-0">{{count(App\User::all())}}</h3>
  </div>
</div>
</div>
<!--p class="text-muted mt-3 mb-0 text-left text-md-center text-xl-left">
  <i class="mdi mdi-reload mr-1" aria-hidden="true"></i> Product-wise sales </p-->
</div>
</div>
</div>

<div class="row">
  <div class="col-lg-12 grid-margin">
    <div class="card">
      <div class="card-body">
        <h4 class="card-title">Projets</h4>
        <div class="table-responsive">
          <table id="example" class="table table-striped table-bordered" style="width:100%">
            <thead>
              <tr>
                <th>Projet </th>
                <th> Date_debut </th>
                <th> Date_fin </th>
                <th> Status </th>
              </tr>
            </thead>
            <tbody>
              @foreach($projets as $projet)
                <tr>
                  <td>{{ $projet->nom }}</td>
                  <td>{{ $projet->date_debut }}</td>
                  <td>{{ $projet->date_fin }}</td>
                  <td>{{ $projet->status }}</td>
                </tr>
              @endforeach
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </div>
</div>

```

DEVOSOFT

Hamza

Nouveau Client +

Dashboard

Clients

Sociétés

Projets

Hamza

Projets 0

Sociétés 0

Clients 0

Utilisateurs 1

Projets

Show 10 entries

Search:

Projet	Date_debut	Date_fin	Status
No data available in table			

Showing 0 to 0 of 0 entries

PreviousNext

foreach (littéralement "pour chaque", c'est-à-dire une exécution de la boucle pour chaque élément d'un tableau)
une boucle spécialement dédiée aux tableaux!

Les Vues (Liste Societe)

Lien : /societe

Capture Page

List Sociétés

Show 10 entries

Search:

Societe ↑↓	Service ↑↓	SiteWeb ↑↓	Telephone ↑↓	Email ↑↓	Action ↑↓
Société Générale	Bank	societebank.ma	524652070	ste@societebank.ma	Modifier Supprimer

Showing 1 to 1 of 1 entries

Previous 1 Next

Les Vues (Liste Societe)

```
<table id="example" class="table table-striped table-bordered" style="width:100%">
  <thead>
    <tr>
      <th>Societe</th>
      <th>Service</th>
      <th>SiteWeb</th>
      <th>Telephone</th>
      <th>Email</th>
      <th>Action</th>
    </tr>
  </thead>
  <tbody>
    @foreach($societes as $sentr)
      <tr>
        <td>{{ $sentr->nomste }}</td>
        <td>{{ $sentr->services }}</td>
        <td>{{ $sentr->siteweb }}</td>
        <td>{{ $sentr->telephone }}</td>
        <td>{{ $sentr->email }}</td>
        <td>
          <form action="{{ url('societe/'.$sentr->id) }}" method="post">
            <a href="{{ url('societe/'.$sentr->id.'/edit') }}" class="btn btn-warning">Modifier</a>
            {{ csrf_field() }}
            {{ method_field('DELETE') }}
            <button type="submit" class="btn btn-danger">Supprimer</button>
          </form>
        </td>
      </tr>
    @endforeach
```

Laravel « Vues »

Ajoute Societe Code HTML

```
<form action="{{ url('societe') }}" method="post" enctype="multipart/form-data">
  <div class="row">
    <div class="form-group col-md-6">
      <label for="nomste">Societe</label>
      <input type="text" class="form-control" name="nomste" id="nomste"
        placeholder="Ex:">
    </div>
    <div class="form-group col-md-6">
      <label for="adresse">Adresse</label>
      <input type="text" class="form-control" name="adresse" id="adresse"
        placeholder="Ex:">
    </div>
    <div class="form-group col-md-6">
      <label for="services">Services</label>
      <input type="text" class="form-control" name="services" id="services"
        placeholder="Ex:">
    </div>
    <div class="form-group col-md-6">
      <label for="siteweb">site_web</label>
      <input type="text" class="form-control" name="siteweb" id="siteweb"
        placeholder="Ex:">
    </div>
    <div class="form-group col-md-6">
      <label for="lat">Maps</label>
      <input type="text" class="form-control" name="lat" id="lat"
        placeholder="Ex:3321453">
      <input type="text" class="form-control" name="lng" id="lng"
        placeholder="Ex:123232">
    </div>
    <div class="form-group col-md-6">
      <label for="email">Email</label>
      <input type="email" class="form-control" name="email" id="email"
        placeholder="Ex:">
    </div>
    <div class="form-group col-md-6">
      <label for="tel">Telephone</label>
      <input type="text" class="form-control" name="tel" id="tel"
        placeholder="Ex:">
    </div>
    <input type="hidden" name="_token" value="{{ csrf_token() }}">
    <div class="form-group col-md-12">
      <button type="submit"
        class="btn btn-outline-primary btn-fw w-25">Ajouter</button>
    </div>
  </div>
</form>
```

Les Vues (Ajout Societe)

Lien : /societe/create

Page Ajoute Societe

Formulaire d'ajouter

Societe

Ex:

Adresse

Ex:

Services

Ex:

site_web

Ex:

Maps

Ex:3321453

Ex:123232

Email

Ex:

Telephone

Ex:

Ajouter

Laravel « Vues »

Les Vues (Societe Modifier)

Code Pour Modifier Information Societe

```
</div>
@if(Session::has('message'))
<div class="alert alert-success" role="alert">{{ Session::get('message') }}</div>
@endif
<form action="{{ url('societe',$societe->id) }}" method="post" enctype="multipart/form-data">
<input type="hidden" name="_method" value="PUT" >
<div class="row">
<div class="form-group col-md-6">
<label for="nomste">Societe</label>
<input type="text" class="form-control" name="nomste" id="nomste"
value="{{ $societe->nomste }}">
</div>
<div class="form-group col-md-6">
<label for="adresse">Adresse</label>
<input type="text" class="form-control" name="adresse" id="adresse"
value="{{ $societe->adresse }}">
</div>
<div class="form-group col-md-6">
<label for="services">Services</label>
<input type="text" class="form-control" name="services" id="services"
value="{{ $societe->services }}">
</div>
<div class="form-group col-md-6">
<label for="siteweb">site_web</label>
<input type="text" class="form-control" name="siteweb" id="siteweb"
value="{{ $societe->siteweb }}">
</div>
<div class="form-group col-md-6">
<label for="lat">Maps</label>
<input type="text" class="form-control" name="lat" id="lat"
value="{{ $societe->lat }}">
<input type="text" class="form-control" name="lng" id="lng"
value="{{ $societe->lng }}">
</div>
<div class="form-group col-md-6">
<label for="email">Email</label>
<input type="email" class="form-control" name="email" id="email"
value="{{ $societe->email }}">
</div>
<div class="form-group col-md-6">
<label for="tel">Telephone</label>
<input type="text" class="form-control" name="tel" id="tel"
value="{{ $societe->telephone }}">
</div>
</div>
```

Page de modification Societe

Formulaire de modification

Societe

Société Générale

Adresse

Route Du Djorf El Youdi, Safi 46000

Services

Bank

site_web

societebank.ma

Maps

Email

ste@societebank.ma

Telephone

524652070

Modifier

Les Vues (Projets)

List Projets

Show 10 entries Search:

Projet ↑↓	date_debut ↑↓	date_fin ↑↓	status ↑↓	Action ↑↓
Salaf	2020-09-23	2025-09-30	Active	Modifier Supprimer

Showing 1 to 1 of 1 entries

Previous 1 Next

Formulaire Ajoute Projet

Formulaire de projet

Projet

Ex:

Date_debut

jj / mm / aaaa

Date_fin

jj / mm / aaaa

Status

Ex:

[Ajouter](#)


Laravel « Vues »

Les Vues (Clients)

List Clients

Show entries

Search:

Foto <small>↑↓</small>	Nom <small>↑↓</small>	Prenom <small>↑↓</small>	Email <small>↑↓</small>	Téléphone <small>↑↓</small>	adresse <small>↑↓</small>	ville <small>↑↓</small>	Payé <small>↑↓</small>	Site_web <small>↑↓</small>	Function <small>↑↓</small>	Date_né <small>↑↓</small>	Services <small>↑↓</small>	Action <small>↑↓</small>
	Ghounbaz	Hamza	hamza1ghounbaz@gmail.com	524652070	Lot Mouna rue ifrane	Safi	Maroc	Hamzaghounbaz.com		1994-06-05	Bank	Modifier Supprimer

Showing 1 to 1 of 1 entries

Previous **1** Next

Formulaire Ajoute Ckient

Formulaire d'ajouter

Nom

Adresse

Date de naissance

Function

Services

ville

Societe

Société Générale

Prénom

Email

Téléphone

Site_Web

Télécharger photo

Aucun fichier sélectionné.

Projet

Payé

Le Routing

Bien qu'indépendant de l'architecture MVC, le routing fait partie intégrante de tous les frameworks PHP.

Dans une architecture MVC, nous allons pointer vers des dossiers virtuels appelés routes

- `http://localhost/login` pour access a login page
- `http://localhost/register` pour access a register page
- `http://localhost/` pour access a dashboard
- `http://localhost/societe` pour access page societes
- `http://localhost/client` pour access page clients
- `http://localhost/projet` pour access page projet

Le Routing

Cette architecture offre de nombreux avantages :

- Protection des fichiers, ceux-ci n'étant plus affichés par le navigateur
- Des URLs plus simples à mémoriser pour les utilisateurs
- Amélioration du référencement si les routes contiennent des mots-clé contenus dans la page correspondante

Laravel « Routing »

Le Routing

Pour config le route : routes/web.php



Route::resource Variable de POST et GET avec Les Function « Index create show
store edit update destroy » Controller ensemble

```
Route::post('/registre', 'utilisateurController@registre');

Route::get('/', 'DashboardController@index');
Route::get('/home', 'DashboardController@index');

Route::resource('/utilisateur', 'utilisateurController');

Route::get('/profile', 'utilisateurController@profile');
Route::post('/regester', 'utilisateurController@register');
Route::get('/regester', 'DashboardController@index');

Route::resource('societe', 'SocietesController');
Route::resource('projet', 'ProjectController');
Route::resource('client', 'ClientController');

Auth::routes();
```


Conclusion

Ce projet nous a permis de revoir et de suivre avec beaucoup d'attention, les méthodes fondamentales de la modélisation et de la conception des systèmes d'informations grâce auxquelles l'application a été réalisée. Nous sommes d'autant plus satisfait que la réalisation de cette application ait utilisé l'éditeur de bases de données relationnel SQL serveur, car l'apprentissage de ces base de données nous a permis d'approfondir nos connaissances antérieures qui étaient mal maîtrisées.

Cette expérience nous a profité dans la complémentarité des efforts et surtout la synergie des compétences et des qualités.

La richesse de ce sujet nous a profité tant dans l'acquisition des connaissances que dans l'initiation du travail de groupe.