



DOCUMENTATION PUBLICOM

Idirakou, Mateo, Lewis



SEMESTRE 1

BT S SIO
AP3

Table des matières

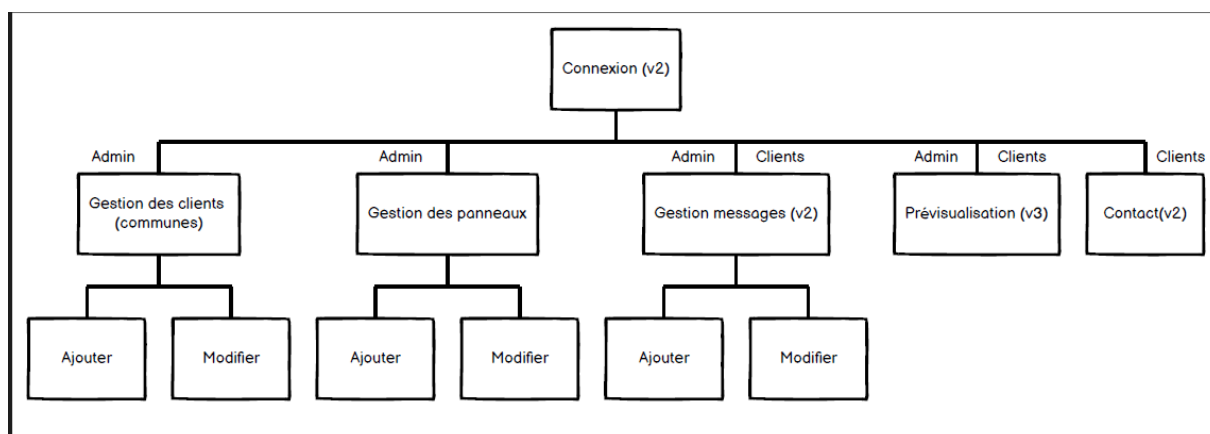
Présentation générale	3
Plan et maquette du site	3
Le Dictionnaire de Données et le Modèle Conceptuel de Données	9
Le descriptif des techniques de l'application	11
Technologies et langages utilisés :	11
Organisation du code :	12
Les Controllers :	12
Les Models :	13
Le guide d'utilisation	15
Quelques écrans de l'application :	15
Procédures d'installation	21
Déroulement du projet	22
Diagramme de Gantt :	22
Les difficultés :	23
Conclusion	23

Présentation générale

Publicom acteur majeur dans la conception et la fabrication de supports d’affichage pour les collectivités locales. Le groupe se distingue par son savoir-faire en innovation et son engagement envers les communes de France et d’Europe. Afin de moderniser ses services et renforcer l’autonomie de ses clients, Publicom souhaite développer une application en ligne de gestion des messages destinés à être diffusés sur ses panneaux d’affichage lumineux.

Cette solution permettra aux communes clientes de gérer directement et facilement les messages affichés sur leurs panneaux, tout en garantissant une expérience utilisateur optimale sur différents supports (ordinateurs, tablettes et smartphones).

Plan et maquette du site



Connexion

Nom utilisateur

Mot de passe

☐ Condition d'utilisation

Valider

Gestion clients

Gestion clients ...

Deconnexion

Liste de clients

Image commune

Nom commune

Responsable

Localisation

Description

Modifier

Supprimer

Image commune

Nom commune

Responsable

Localisation

Description

Modifier

Supprimer

Image commune

Nom commune

Responsable

Localisation

Description

Modifier

Supprimer

Image commune

Nom commune

Responsable

Localisation

Description

Modifier

Supprimer

Ajouter

Ajouter des clients

Deconnexion

Nom Commune

Nom responsable

Description

Localisation

[Un lien](#)

Image

Valider

Modifier des clients

Deconnexion

Modifier des clients

Nom Commune

Nom responsable

Description

Localisation

[Un lien](#)

Image

Valider

Gestion panneaux

Deconnexion

Gestion des panneaux ...

Liste des panneaux

Q

Ajouter

panneau 1

commune

Responsable

Description

Disponibilité

Messages

Etat message

Modifier

Supprimer

panneau 2

commune

Responsable

Description

Disponibilité

Messages

Etat message

Modifier

Supprimer

panneau 3

commune

Responsable

Description

Disponibilité

Messages

Etat message

Modifier

Supprimer

panneau 4

commune

Responsable

Description

Disponibilité

Messages

Etat message

Modifier

Supprimer

Ajouter des panneaux

Deconnexion

Ajouter des panneaux

Choisir la commune

Albias
Toulouse
Caussade
Montauban

Messages

Etat message

Disponibilité

Modifier panneaux

Deconnexion

Modifier panneaux

Choisir la commune

Albias
Toulouse
Caussade
Montauban

Messages

Etat message

Disponibilité

Gestion panneaux (admin)

Gestion des messages ...

Deconnexion

Liste des messages

Ajouter

commune	Responsable	Description	
			<div>Modifier</div> <div>Supprimer</div>
			<div>Modifier</div> <div>Supprimer</div>
			<div>Modifier</div> <div>Supprimer</div>
			<div>Modifier</div> <div>Supprimer</div>

Gestion panneaux (clients)

Gestion des messages ...

Deconnexion

Liste des messages

Ajouter

Responsable	Description	
		<div>Modifier</div> <div>Supprimer</div>
		<div>Modifier</div> <div>Supprimer</div>
		<div>Modifier</div> <div>Supprimer</div>
		<div>Modifier</div> <div>Supprimer</div>

Ajouter messages(v2)

Deconnexion

Ajouter messages

Messages

A[°] A[°] Aa A_p
A B C A

Valider

Modifier panneaux (v2)

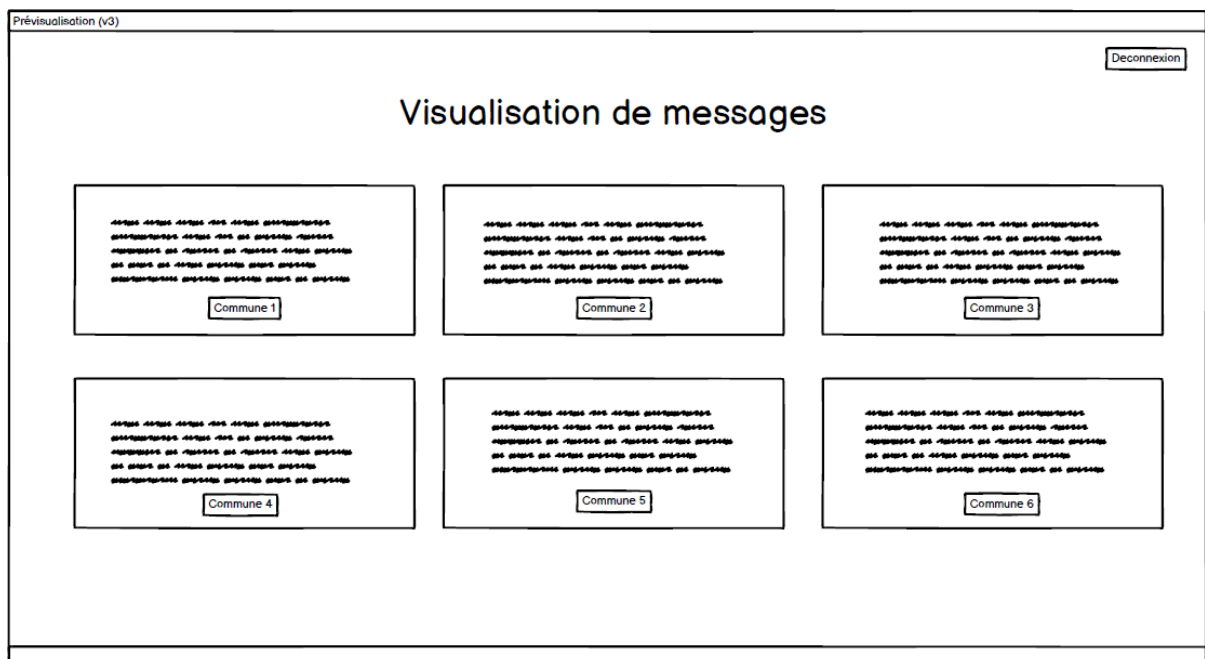
Deconnexion

Modifier messages

Messages

A[°] A[°] Aa A_p
A B C A

Modifier

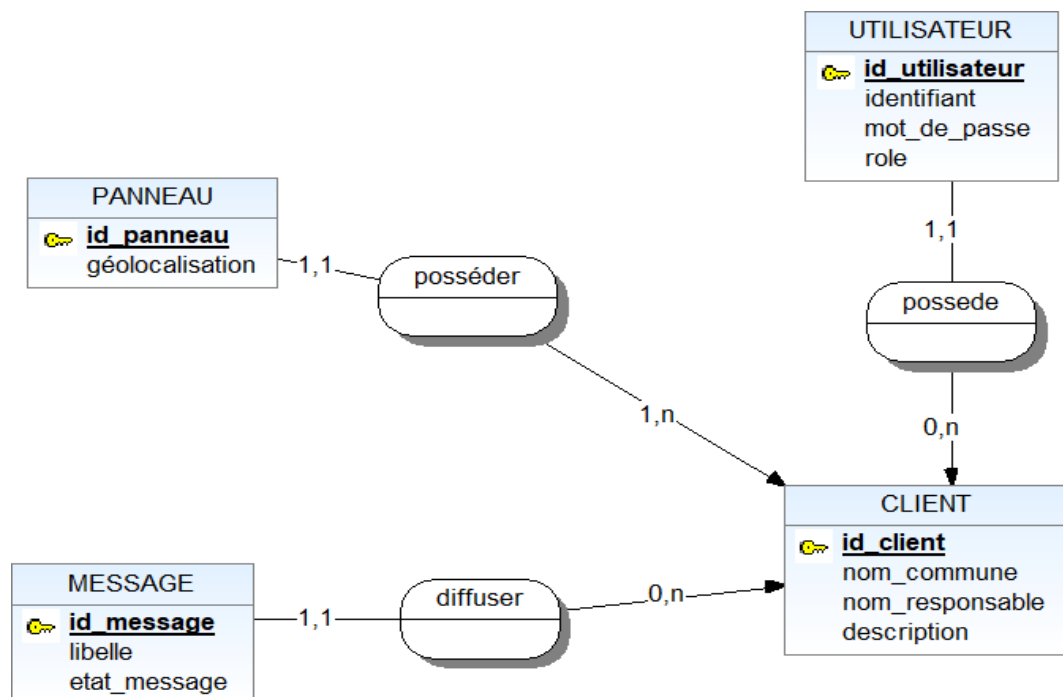


Le Dictionnaire de Données et le Modèle Conceptuel de Données

Visualisation du Dictionnaire de Données (DD) :

Nom	Format	Longueur	Type		Règle de calcul	Règle de gestion	Document
			E	C			
id_client	integer	5	X				//
nom_commune	Alphabétique	30	X				//
nom_responsable	Alphabétique	127	X				//
description	Alphanumérique	127	X				//
id_utilisateur	integer	5	X				//
identifiant	Alphanumérique	127	X				//
mot_de_passe	Alphanumérique	127	X				//
rôle	Booléen	1	X				//
id_panneau	integer	5	X				//
géolocalisation	Alphanumérique	30	X				//
id_message	integer	5	X				//
libelle	Alphanumérique	127	X				//
etat_message	Booléen	1	X				//

Modèle Conceptuel de Données (MCD) :



Modèle Relationnel de Données (MRD) :

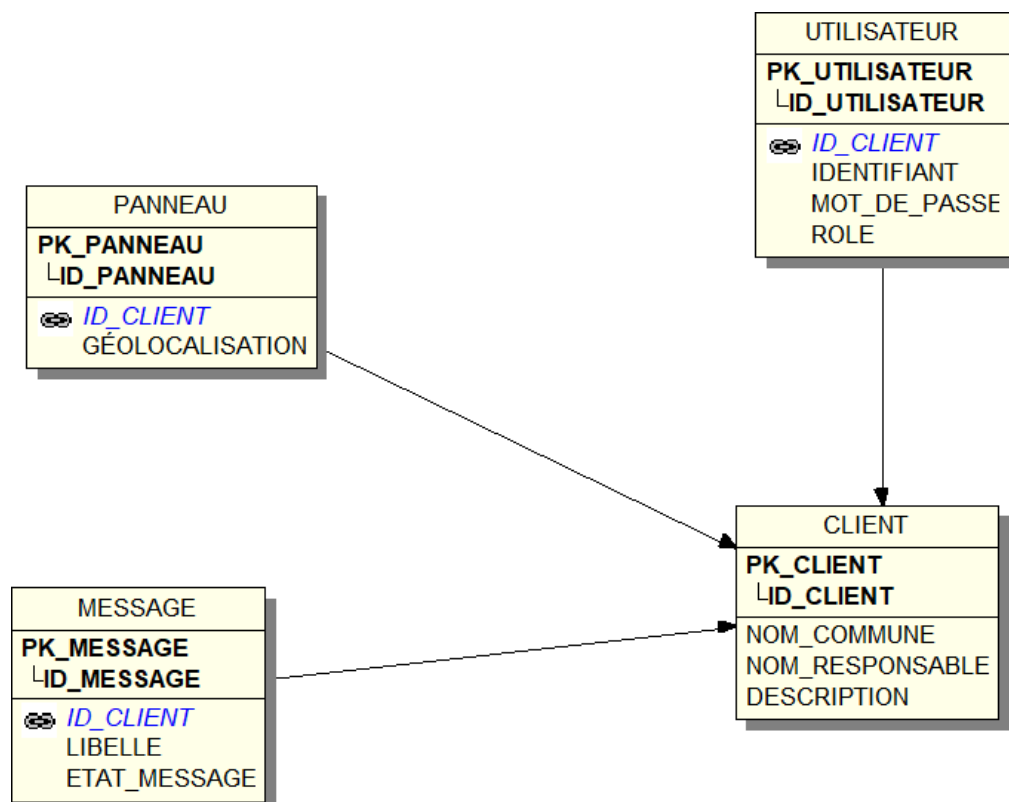
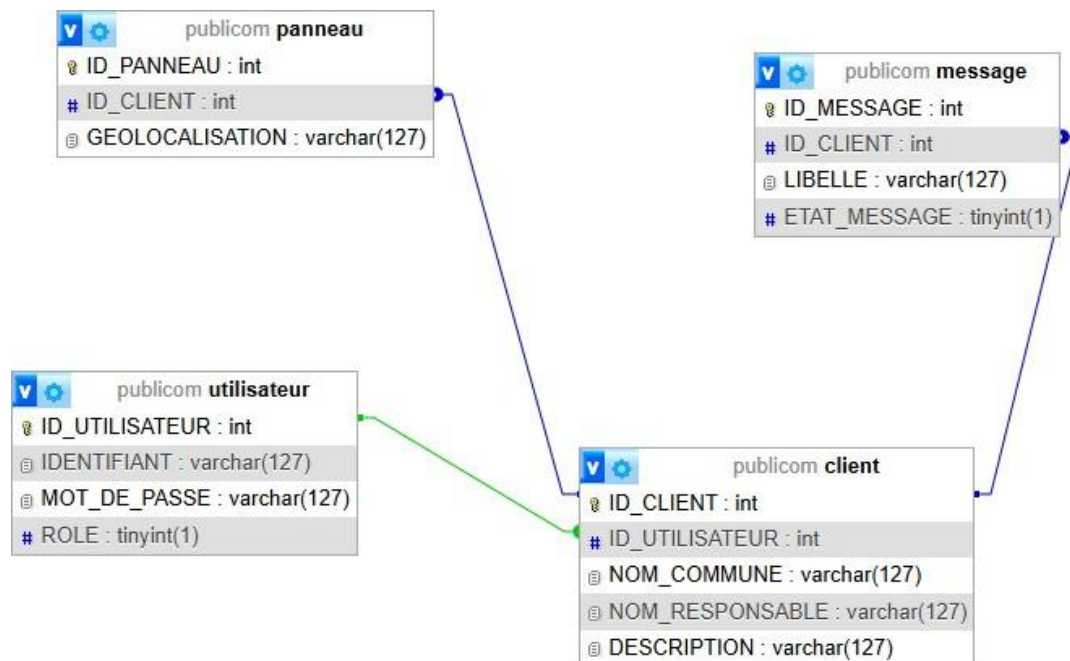


Schéma concepteur de la base de données (phpmyadmin) :



Le descriptif des techniques de l'application

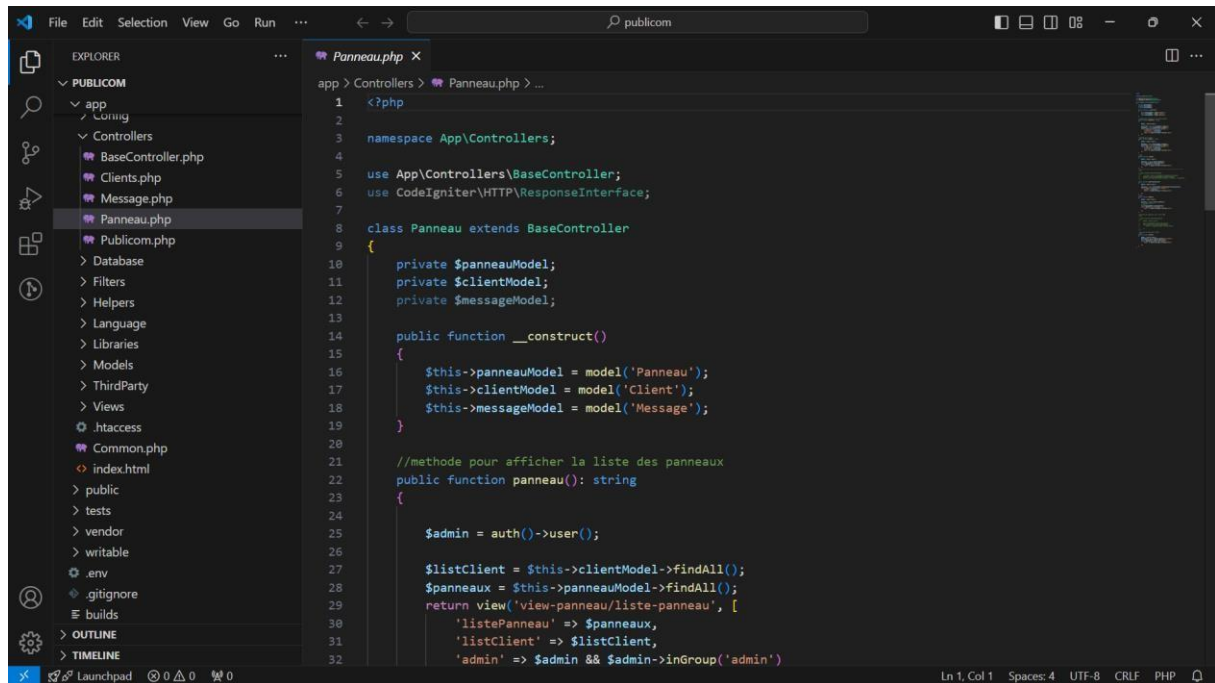
Technologies et langages utilisés :

Outils	Langages
<ul style="list-style-type: none"> - Wampserver64 (serveur web local) - Visual Studio Code (développement C programmation) - PhpMyAdmin (base de données) - CodeIgniter 4 (FrameWork Web) - Firefox (Navigateur Web) - Gantt Project (Diagramme de Gantt) - Proxmox (serveur web) - Canva pour le diaporama 	<ul style="list-style-type: none"> - HTML - PHP - CSS - SQL - JavaScript - Bash

Organisation du code :

Les Controllers :

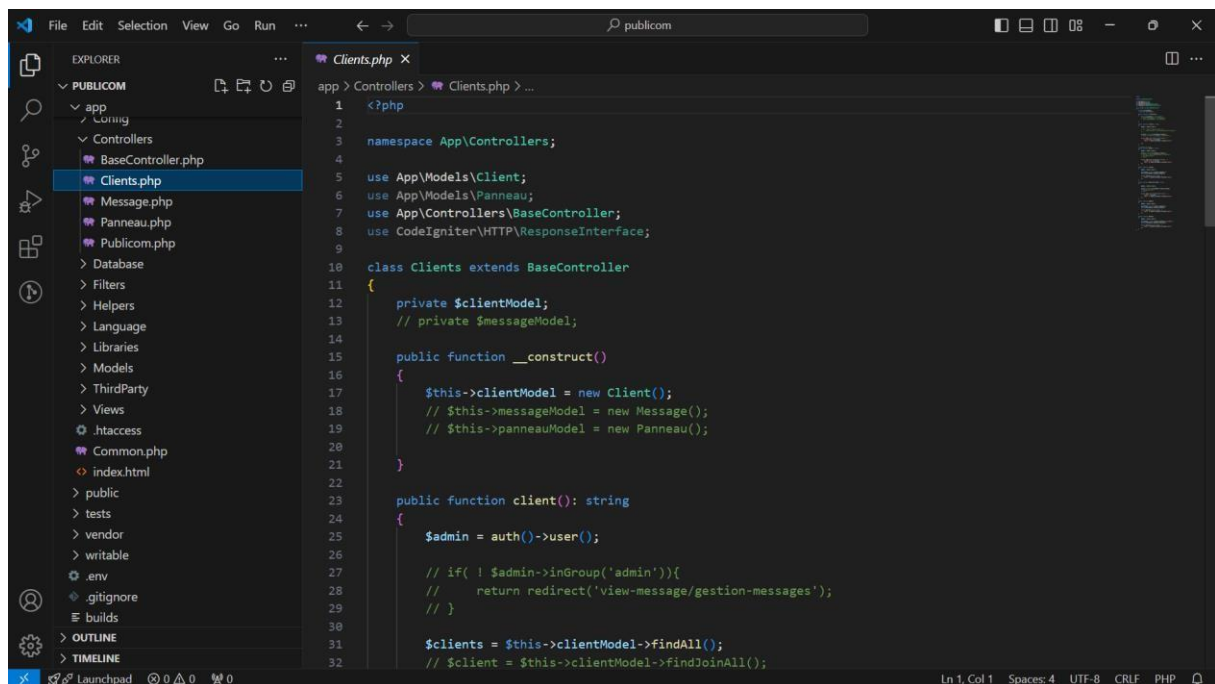
Controller Panneau :



The screenshot shows a code editor with the file explorer on the left. The file explorer shows the project structure: PUBLICOM > app > Controllers > Panneau.php. The main editor displays the code for Panneau.php. The code is as follows:

```
1 <?php
2
3 namespace App\Controllers;
4
5 use App\Controllers\BaseController;
6 use CodeIgniter\HTTP\ResponseInterface;
7
8 class Panneau extends BaseController
9 {
10     private $panneauModel;
11     private $clientModel;
12     private $messageModel;
13
14     public function __construct()
15     {
16         $this->panneauModel = model('Panneau');
17         $this->clientModel = model('Client');
18         $this->messageModel = model('Message');
19     }
20
21     //methode pour afficher la liste des panneaux
22     public function panneau(): string
23     {
24
25         $admin = auth()->user();
26
27         $listClient = $this->clientModel->findAll();
28         $panneaux = $this->panneauModel->findAll();
29         return view('view-panneau/liste-panneau', [
30             'listePanneau' => $panneaux,
31             'listClient' => $listClient,
32             'admin' => $admin && $admin->inGroup('admin')
33         ]);
34     }
35 }
```

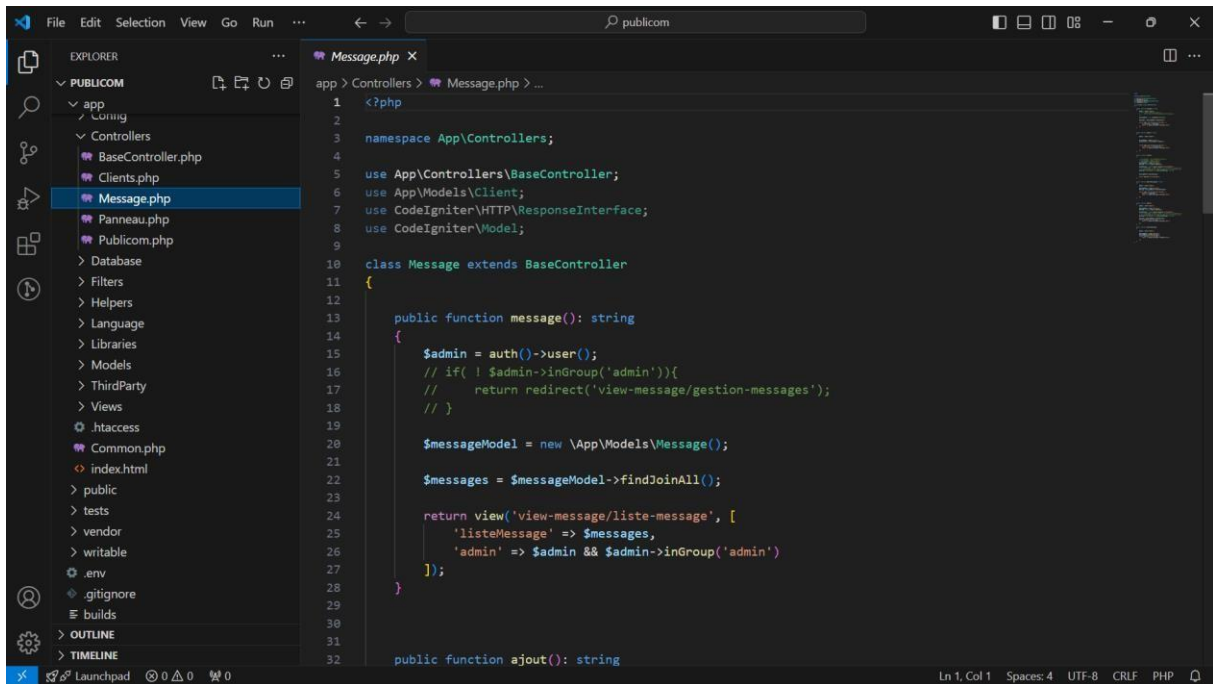
Controller Clients :



The screenshot shows a code editor with the file explorer on the left. The file explorer shows the project structure: PUBLICOM > app > Controllers > Clients.php. The main editor displays the code for Clients.php. The code is as follows:

```
1 <?php
2
3 namespace App\Controllers;
4
5 use App\Models\Client;
6 use App\Models\Panneau;
7 use App\Controllers\BaseController;
8 use CodeIgniter\HTTP\ResponseInterface;
9
10 class Clients extends BaseController
11 {
12     private $clientModel;
13     // private $messageModel;
14
15     public function __construct()
16     {
17         $this->clientModel = new Client();
18         // $this->messageModel = new Message();
19         // $this->panneauModel = new Panneau();
20     }
21
22     public function client(): string
23     {
24         $admin = auth()->user();
25
26         // if( ! $admin->inGroup('admin')){
27         //     return redirect('view-message/gestion-messages');
28         // }
29
30         $clients = $this->clientModel->findAll();
31         // $client = $this->clientModel->findJoinAll();
32     }
33 }
```

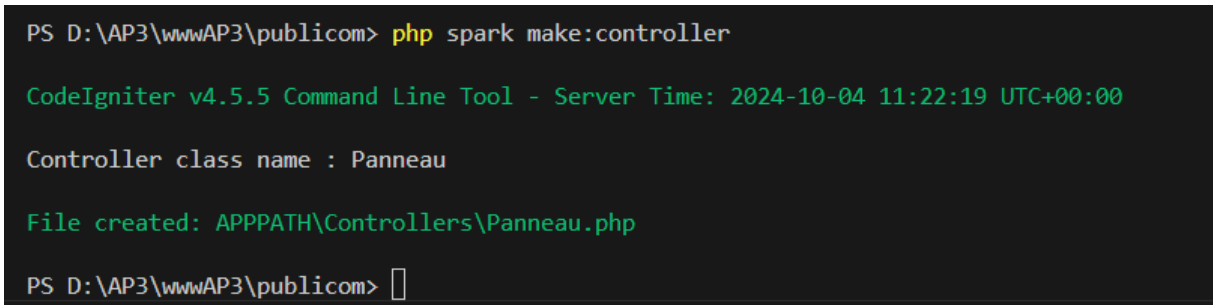
Controller Message :



The screenshot shows a code editor with the file explorer on the left. The file explorer shows the project structure: PUBLICCOM > app > Controllers > Message.php. The main editor displays the content of Message.php, which is a PHP class extending BaseController. The code includes namespace declarations, use statements for App\Controllers\BaseController, App\Models\Client, CodeIgniter\HTTP\ResponseInterface, and CodeIgniter\Model. The class Message has two public methods: message() and ajout(). The message() method performs authentication, checks if the user is in the 'admin' group, and if so, it fetches messages from the database and returns a view. The ajout() method is currently empty.

```
1 <?php
2
3 namespace App\Controllers;
4
5 use App\Controllers\BaseController;
6 use App\Models\Client;
7 use CodeIgniter\HTTP\ResponseInterface;
8 use CodeIgniter\Model;
9
10 class Message extends BaseController
11 {
12
13     public function message(): string
14     {
15         $admin = auth()->user();
16         // if( ! $admin->inGroup('admin')){
17         //     return redirect('view-message/gestion-messages');
18         // }
19
20         $messageModel = new \App\Models\Message();
21
22         $messages = $messageModel->findJoinAll();
23
24         return view('view-message/liste-message', [
25             'listeMessage' => $messages,
26             'admin' => $admin && $admin->inGroup('admin')
27         ]);
28     }
29
30
31
32     public function ajout(): string
```

Exemple de commandes pour création des controllers :



The screenshot shows a terminal window with the following text:

```
PS D:\AP3\wwwAP3\publicom> php spark make:controller

CodeIgniter v4.5.5 Command Line Tool - Server Time: 2024-10-04 11:22:19 UTC+00:00

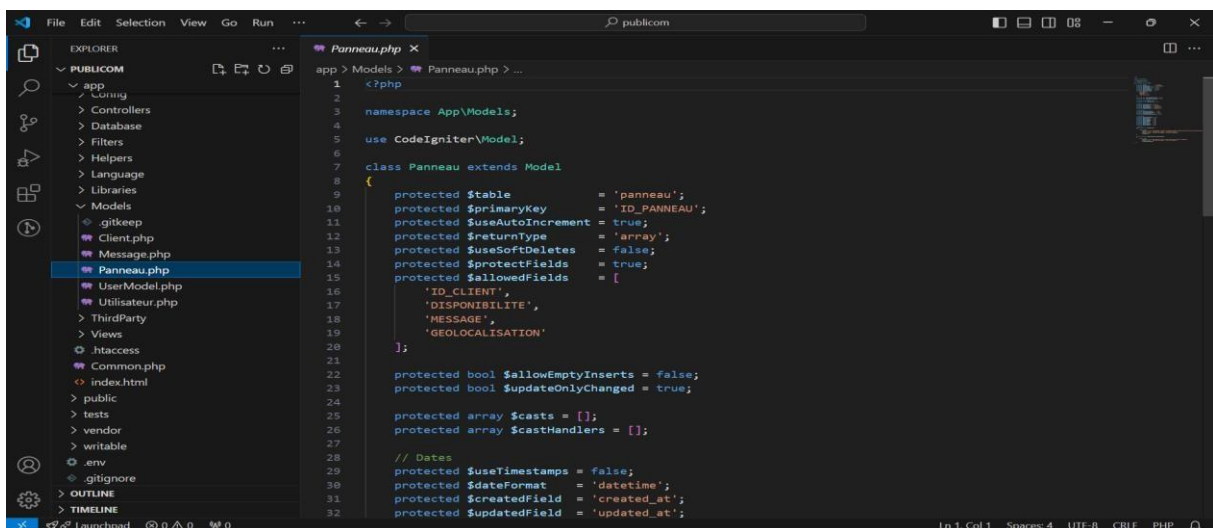
Controller class name : Panneau

File created: APPPATH\Controllers\Panneau.php

PS D:\AP3\wwwAP3\publicom> 
```

Les Models :

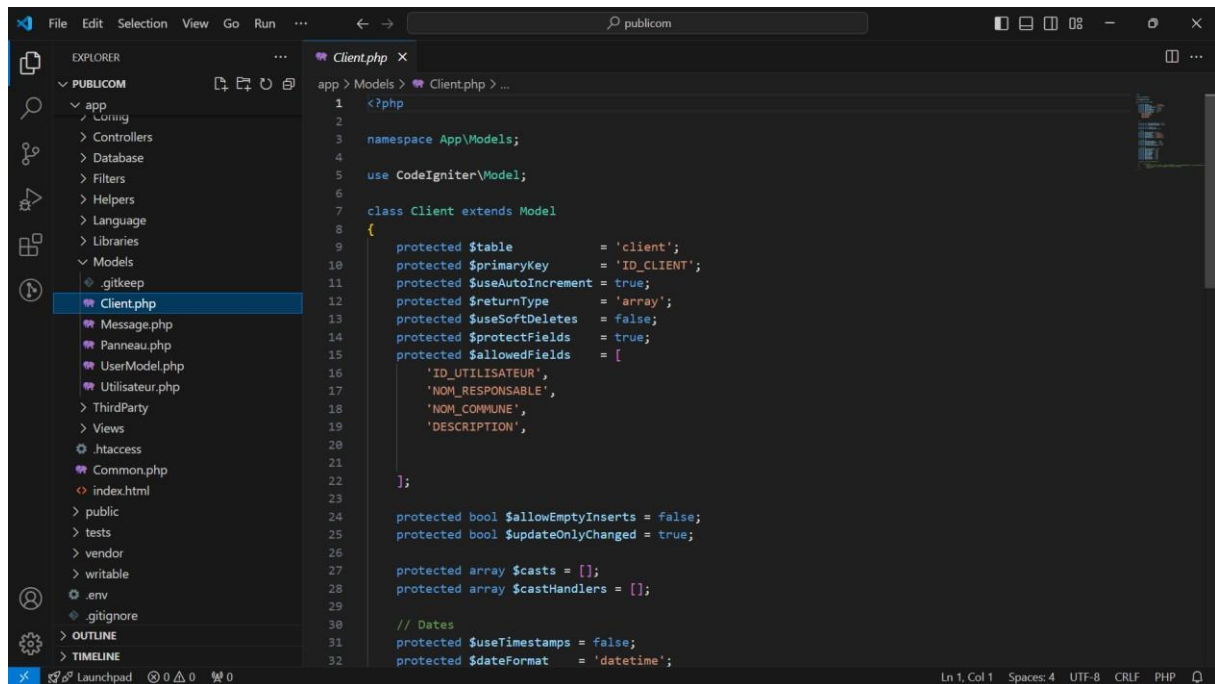
Model Panneau :



The screenshot shows a code editor with the file explorer on the left. The file explorer shows the project structure: PUBLICCOM > app > Models > Panneau.php. The main editor displays the content of Panneau.php, which is a PHP class extending Model. The code includes namespace declarations, use statements for CodeIgniter\Model, and class Panneau. The class Panneau has several protected properties: \$table (panneau), \$primaryKey (ID_PANNEAU), \$useAutoIncrement (true), \$returnType (array), \$useSoftDeletes (false), \$protectFields (true), and \$allowedFields (ID_CLIENT, DISPONIBILITE, MESSAGE, GEOLOCALISATION). It also has protected boolean properties \$allowEmptyInserts (false) and \$updateOnlyChanged (true), and protected array properties \$casts (empty array) and \$castHandlers (empty array). The class also has protected boolean properties \$useTimestamps (false), \$dateFormat (datetime), \$createdField (created_at), and \$updatedField (updated_at).

```
1 <?php
2
3 namespace App\Models;
4
5 use CodeIgniter\Model;
6
7 class Panneau extends Model
8 {
9
10     protected $table = 'panneau';
11     protected $primaryKey = 'ID_PANNEAU';
12     protected $useAutoIncrement = true;
13     protected $returnType = 'array';
14     protected $useSoftDeletes = false;
15     protected $protectFields = true;
16     protected $allowedFields = [
17         'ID_CLIENT',
18         'DISPONIBILITE',
19         'MESSAGE',
20         'GEOLOCALISATION'
21     ];
22
23     protected bool $allowEmptyInserts = false;
24     protected bool $updateOnlyChanged = true;
25
26     protected array $casts = [];
27     protected array $castHandlers = [];
28
29     // Dates
30     protected $useTimestamps = false;
31     protected $dateFormat = 'datetime';
32     protected $createdField = 'created_at';
33     protected $updatedField = 'updated_at';
```

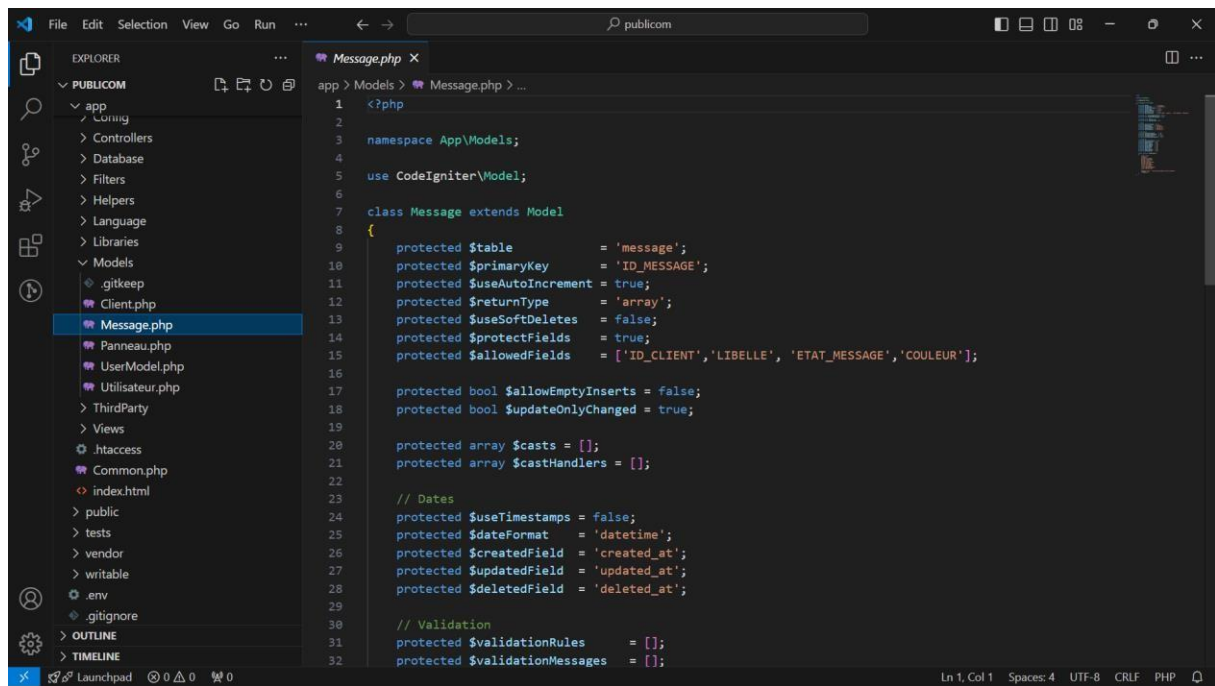
Model Client :



The screenshot shows the Visual Studio Code editor with the 'Client.php' file open. The Explorer sidebar on the left shows the project structure, with 'Client.php' selected under the 'Models' folder. The main editor displays the PHP code for the 'Client' model, which extends the 'Model' class from 'CodeIgniter'. The code includes properties for table name, primary key, auto-increment, return type, soft deletes, protected fields, allowed fields, empty inserts, update only changed, casts, cast handlers, timestamps, and date format.

```
1 <?php
2
3 namespace App\Models;
4
5 use CodeIgniter\Model;
6
7 class Client extends Model
8 {
9     protected $table      = 'client';
10    protected $primaryKey = 'ID_CLIENT';
11    protected $useAutoIncrement = true;
12    protected $returnType  = 'array';
13    protected $useSoftDeletes = false;
14    protected $protectFields = true;
15    protected $allowedFields = [
16        'ID_UTILISATEUR',
17        'NOM_RESPONSABLE',
18        'NOM_COMMUNE',
19        'DESCRIPTION',
20    ];
21
22
23
24    protected bool $allowEmptyInserts = false;
25    protected bool $updateOnlyChanged = true;
26
27    protected array $casts = [];
28    protected array $castHandlers = [];
29
30    // Dates
31    protected $useTimestamps = false;
32    protected $dateFormat    = 'datetime';
```

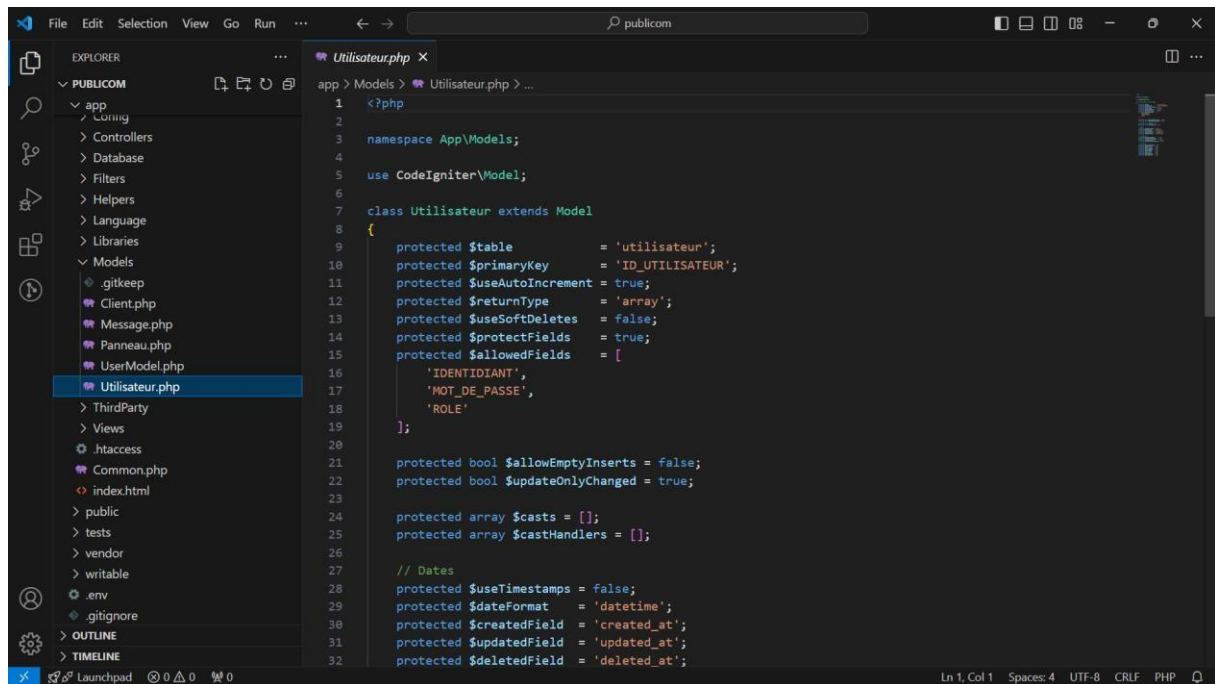
Model Message :



The screenshot shows the Visual Studio Code editor with the 'Message.php' file open. The Explorer sidebar on the left shows the project structure, with 'Message.php' selected under the 'Models' folder. The main editor displays the PHP code for the 'Message' model, which extends the 'Model' class from 'CodeIgniter'. The code includes properties for table name, primary key, auto-increment, return type, soft deletes, protected fields, allowed fields, empty inserts, update only changed, casts, cast handlers, timestamps, date format, created/updated/deleted fields, validation rules, and validation messages.

```
1 <?php
2
3 namespace App\Models;
4
5 use CodeIgniter\Model;
6
7 class Message extends Model
8 {
9     protected $table      = 'message';
10    protected $primaryKey = 'ID_MESSAGE';
11    protected $useAutoIncrement = true;
12    protected $returnType  = 'array';
13    protected $useSoftDeletes = false;
14    protected $protectFields = true;
15    protected $allowedFields = ['ID_CLIENT', 'LIBELLE', 'ETAT_MESSAGE', 'COULEUR'];
16
17
18    protected bool $allowEmptyInserts = false;
19    protected bool $updateOnlyChanged = true;
20
21    protected array $casts = [];
22    protected array $castHandlers = [];
23
24    // Dates
25    protected $useTimestamps = false;
26    protected $dateFormat    = 'datetime';
27    protected $createdField   = 'created_at';
28    protected $updatedField   = 'updated_at';
29    protected $deletedField   = 'deleted_at';
30
31    // Validation
32    protected $validationRules = [];
33    protected $validationMessages = [];
```

Model Utilisateur (Shield) :



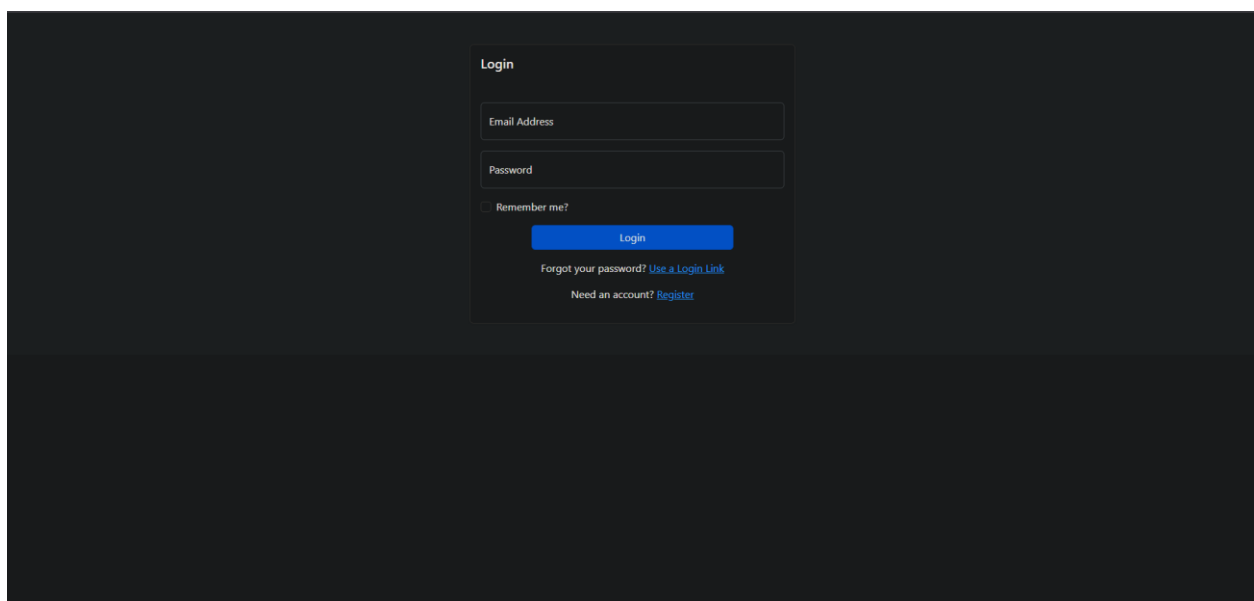
The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'app', 'controllers', 'database', 'filters', 'helpers', 'language', 'libraries', 'models', 'public', 'tests', 'vendor', 'writable', and 'env'. The 'models' folder is expanded, showing files like 'Client.php', 'Message.php', 'Panneau.php', 'UserModel.php', and 'Utilisateur.php'. The 'Utilisateur.php' file is selected and its code is displayed in the editor. The code is a PHP class named 'Utilisateur' that extends 'Model'. It defines various properties like '\$table', '\$primaryKey', '\$useAutoIncrement', '\$returnType', '\$useSoftDeletes', '\$protectFields', '\$allowedFields', '\$allowEmptyInserts', '\$updateOnlyChanged', '\$casts', '\$castHandlers', '\$useTimestamps', '\$dateFormat', '\$createdField', '\$updatedField', and '\$deletedField'.

```
1 <?php
2
3 namespace App\Models;
4
5 use CodeIgniter\Model;
6
7 class Utilisateur extends Model
8 {
9     protected $table = 'utilisateur';
10    protected $primaryKey = 'ID_UTILISATEUR';
11    protected $useAutoIncrement = true;
12    protected $returnType = 'array';
13    protected $useSoftDeletes = false;
14    protected $protectFields = true;
15    protected $allowedFields = [
16        'IDENTIFIANT',
17        'MOT_DE_PASSE',
18        'ROLE'
19    ];
20
21    protected bool $allowEmptyInserts = false;
22    protected bool $updateOnlyChanged = true;
23
24    protected array $casts = [];
25    protected array $castHandlers = [];
26
27    // Dates
28    protected $useTimestamps = false;
29    protected $dateFormat = 'datetime';
30    protected $createdField = 'created_at';
31    protected $updatedField = 'updated_at';
32    protected $deletedField = 'deleted_at';
```

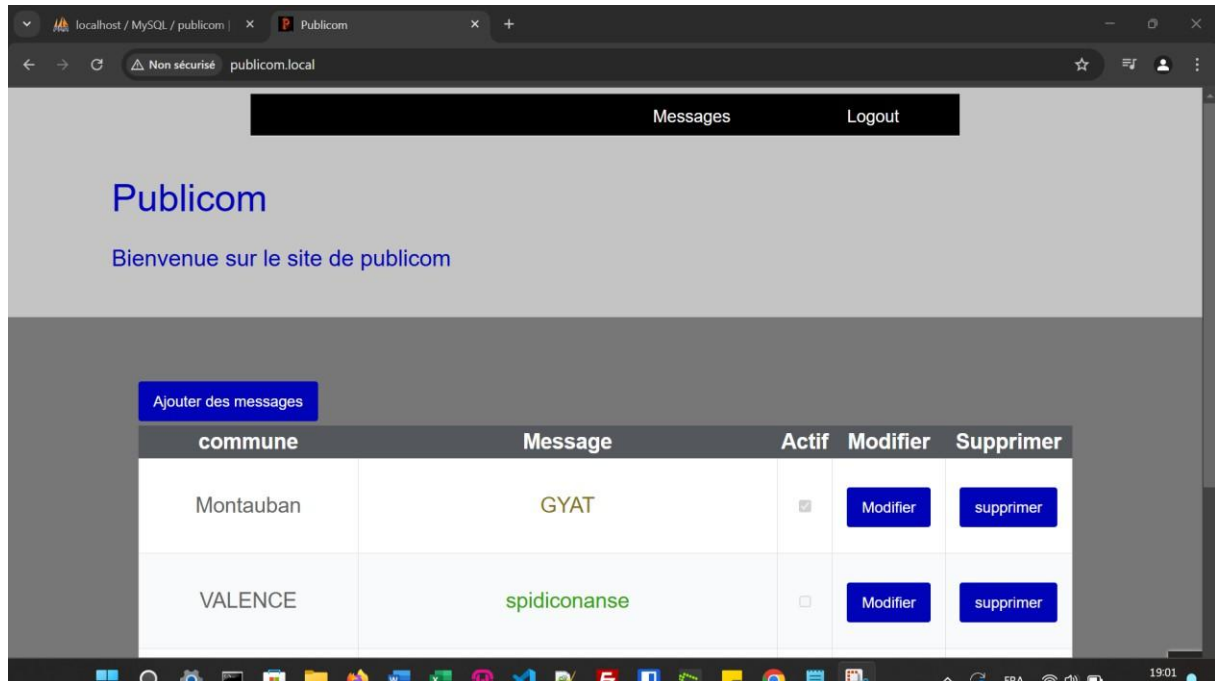
Le guide d'utilisation

Quelques écrans de l'application :

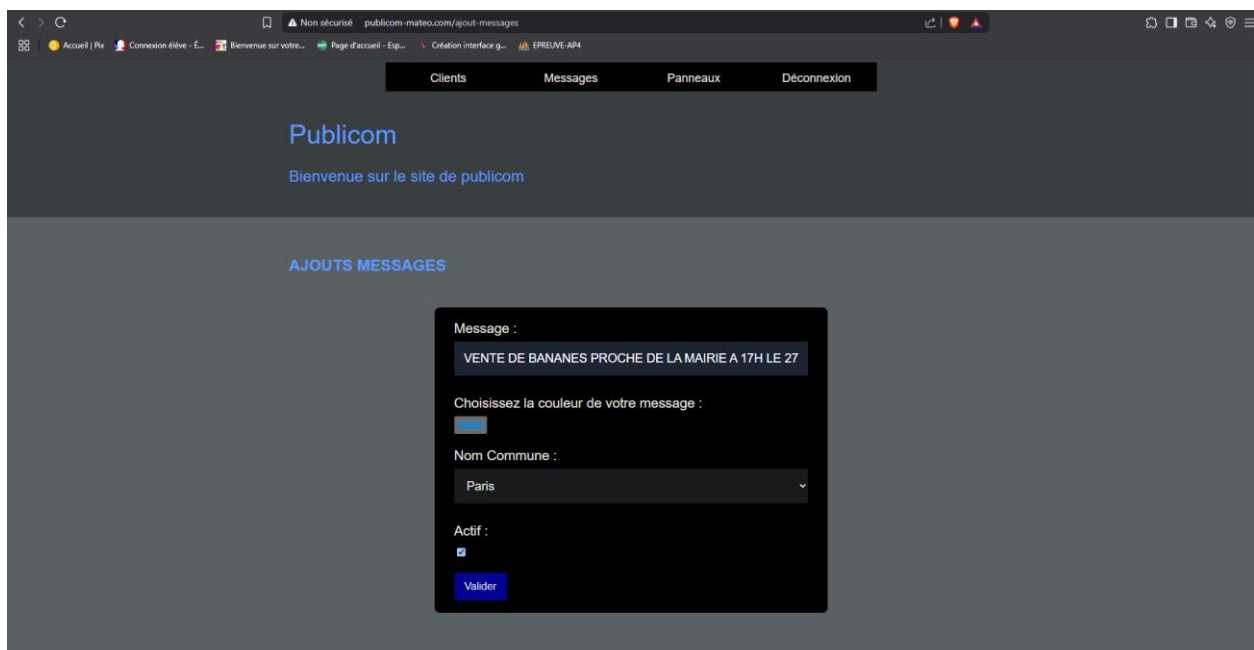
Ecran de connexion :



Ecran liste des messages quand on est connecté en tant qu'utilisateur (commune) :



Ecran d'ajout de message en tant qu'utilisateur (commune) :



Ecran modifier de message en tant qu'utilisateur :

The screenshot shows a web browser window with the URL `publicom-matéo.com/modif-messages-14`. The page has a dark header with navigation links: **Clients**, **Messages**, **Panneaux**, and **Déconnexion**. Below the header, the **Publicom** logo and the text "Bienvenue sur le site de publicom" are displayed. The main content area features a dark modal box for editing a message. Inside the modal, the text "Message :" is followed by a text input field containing "VENTE DE BANANES PROCHE DE LA MAIRIE A 17H LE 27". Below this, the label "Choisi la couleur de ton message:" is followed by a color selection interface with a blue square and the text "Actif:". At the bottom of the modal is a blue button labeled "Valider".

Ecran d'accueil correspondant à la liste des messages de l'ensemble des clients :

The screenshot shows the home page of the Publicom application. The browser window has the URL `publicom.local`. The header includes navigation links: **Clients**, **Messages**, **Panneaux**, and **Logout**. The **Publicom** logo and the text "Bienvenue sur le site de publicom" are present. Below the header, there is a blue button labeled "Ajouter des messages". The main content area displays a table of messages.

commune	Message	Actif	Modifier	Supprimer
Montauban	GYAT	<input checked="" type="checkbox"/>	Modifier	supprimer
VALENCE	spidiconanse	<input type="checkbox"/>	Modifier	supprimer

Ecran listes des clients :

The screenshot shows the 'Gestions Clients' page in the Publicom application. The page has a dark theme with a navigation bar at the top containing 'Clients', 'Messages', 'Panneaux', and 'Déconnexion'. Below the navigation bar, the 'Publicom' logo and the text 'Bienvenue sur le site de publicom' are displayed. The main content area is titled 'GESTIONS CLIENTS' and features a button 'Ajouts Clients'. Below this button is a table with the following data:

NOM COMMUNE	RESPONSABLE	DESCRIPTION	MODIFIER	SUPPRIMER
Paris	Jean Dupont	Client premium basé à Paris avec une longue collaboration.	Modifier	Supprimer
Lyon	Sophie Martin	Intéressée par les solutions éco-responsables.	Modifier	Supprimer
Marseille	Paul Bernard	Client fidèle depuis 2015.	Modifier	Supprimer
Toulouse	Claire Durand	Expansion rapide dans le sud-ouest.	Modifier	Supprimer
Nice	Luc Moreau	Nouvel acteur du marché azuréen.	Modifier	Supprimer

Ecran ajout de clients :

The screenshot shows the 'Ajouts Clients' page in the Publicom application. The page has a dark theme with a navigation bar at the top containing 'Clients', 'Messages', 'Panneaux', and 'Déconnexion'. Below the navigation bar, the 'Publicom' logo and the text 'Bienvenue sur le site de publicom' are displayed. The main content area is titled 'AJOUTS CLIENTS' and features a form for adding new clients. The form has the following fields:

- Nom commune :
- Nom responsable :
- Description :

Below the form is a 'Valider' button.

Ecran de modification des clients :

The screenshot shows a web browser window with the URL `publicom-mateo.com/modif-clients-1`. The page has a dark header with the 'Publicom' logo and a navigation bar with links: 'Clients', 'Messages', 'Panneaux', and 'Déconnexion'. Below the header, the text 'Bienvenue sur le site de publicom' is displayed. The main content area is titled 'MODIFICATION CLIENT' and contains a form with the following fields:

- Nom commune:** A text input field containing 'Paris'.
- Nom responsable:** A text input field containing 'Jean Dupont'.
- Description:** A text area containing 'Client premium basé à Paris avec une longue collaboration.'
- Valider:** A blue button at the bottom of the form.

Ecran liste des panneaux :

The screenshot shows a web browser window with the URL `publicom-mateo.com/gestion-panneaux`. The page has a dark header with the 'Publicom' logo and a navigation bar with links: 'Clients', 'Messages', 'Panneaux', and 'Déconnexion'. Below the header, the text 'Bienvenue sur le site de publicom' is displayed. The main content area is titled 'GESTIONS PANNEAUX' and contains a table with the following data:

Ajouts Panneaux			
NOM COMMUNE	GEOLOCALISATION	MODIFIER	SUPPRIMER
Paris	48.8566, 12.0000	Modifier	Supprimer
Lyon	45.7640, 4.8357	Modifier	Supprimer
Marseille	43.2965, 5.3698	Modifier	Supprimer
Toulouse	43.6047, 1.4442	Modifier	Supprimer
Nice	43.7102, 7.2620	Modifier	Supprimer
Nantes	47.2184, -1.5536	Modifier	Supprimer

Ecran de modification des panneaux :

The screenshot shows a web browser window with the URL `publicom-mateo.com/modif-panneau-7`. The page has a dark theme and a navigation bar with links: `Clients`, `Messages`, `Panneaux` (active), and `Déconnexion`. The main content area displays the `Publicom` logo and the text `Bienvenue sur le site de publicom`. A central form is used for editing a panel, with the following fields:

- `Nom commune :` A text input field containing `Strasbourg`.
- `Geolocalisation :` A text input field containing `48.5734,`.
- A `Valider` button.

The footer of the page contains the copyright notice: `© 2024 - 2025 - BOURGIR`.

Ecran d'ajout de panneaux :

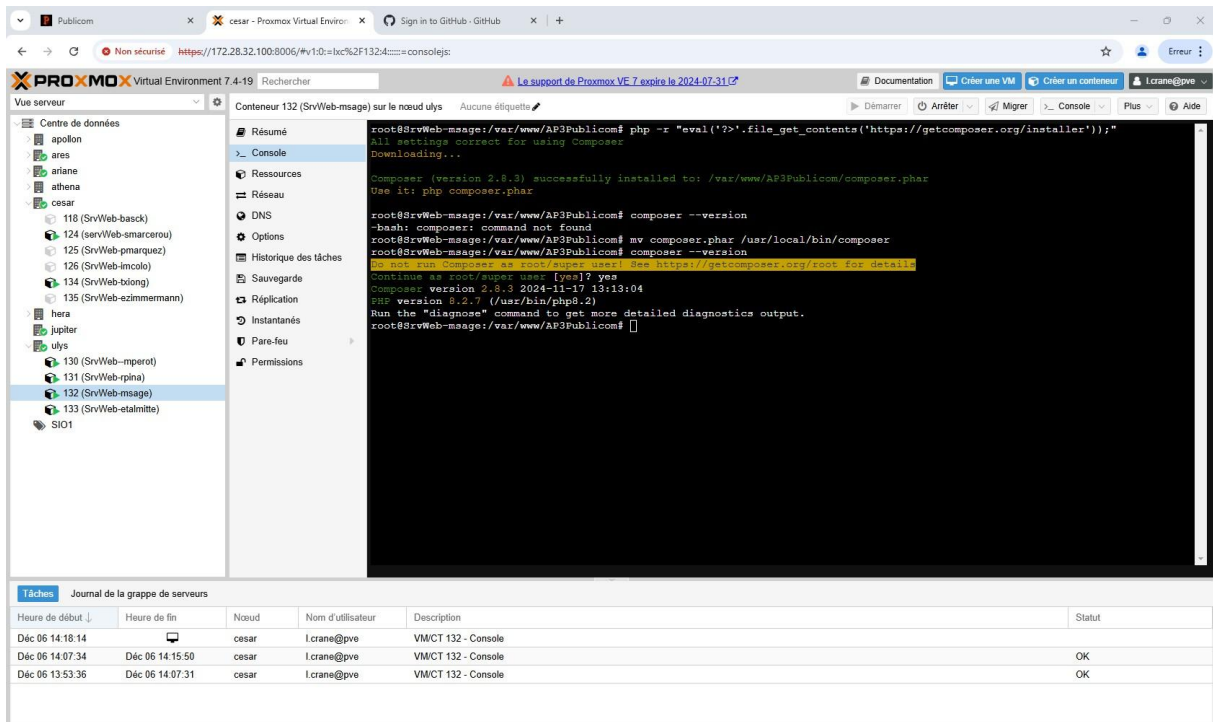
The screenshot shows the 'Ajout de panneaux' screen in the Publicom application. The navigation bar is the same as in the previous screen. The main content area displays the `Publicom` logo and the text `Bienvenue sur le site de`. A central form is used for adding a new panel, with the following fields:

- `Cliet`: A dropdown menu with a list of cities. The list includes: `Paris`, `Lyon`, `Marseille`, `Toulouse`, `Nice`, `Nantes`, `Strasbourg` (highlighted), `Montpellier`, `Bordeaux`, `Lille`, `Rennes`, `Reims`, `Le Havre`, `Saint-Etienne`, `Grenoble`, `Niamey`, and `--Choisissez un client--`.
- `Geolocalisation :` A text input field.
- A `Valider` button.

The footer of the page contains the copyright notice: `© 2024 - 2025 - BOURGIR`.

Procédures d'installation

Mise en ligne sur le serveur Proxmox de Mateo :



The screenshot displays the Proxmox VE web interface. The main window shows the console of VM 132 (SrvWeb-msage). The console output indicates the successful installation of Composer 2.8.3. The left sidebar shows a list of VMs, and the bottom table shows the task log.

Heure de début	Heure de fin	Nœud	Nom d'utilisateur	Description	Statut
Déc 06 14:18:14		cesar	lcrane@pve	VMCT 132 - Console	
Déc 06 14:07:34	Déc 06 14:15:50	cesar	lcrane@pve	VMCT 132 - Console	OK
Déc 06 13:53:36	Déc 06 14:07:31	cesar	lcrane@pve	VMCT 132 - Console	OK

```
www-data@SrvWeb-msage:~$ ls -l
total 36
drwxr-xr-x 2 www-data www-data 4096 Jan 31 2024 Ethane
drwxr-xr-x 2 www-data www-data 4096 Jan 31 2024 Mathis
drwxr-xr-x 2 www-data www-data 4096 Jan 31 2024 Paul
drwxrwxrwx 6 www-data www-data 4096 Apr 24 2024 ProjetAPI
drwxr-xr-x 2 www-data www-data 4096 Jan 31 2024 Ronald
drwxr-xr-x 2 www-data www-data 4096 Jan 31 2024 Teddy
drwxr-xr-x 2 www-data www-data 4096 May 6 2024 html
drwxrwxrwx 6 www-data www-data 4096 May 6 2024 monpanier
drwxr-xr-x 2 www-data www-data 4096 Feb 7 2024 prenom
www-data@SrvWeb-msage:~$ git clone https://github.com/idrakMcl/publicom.git
Cloning into 'publicom'...
Username for 'https://github.com': idrakMcl
Password for 'https://idrakMcl@github.com':
remote: Enumerating objects: 895, done.
remote: Counting objects: 100% (158/158), done.
remote: Compressing objects: 100% (101/101), done.
remote: Total 895 (delta 86), reused 110 (delta 57), pack-reused 737 (from 1)
Receiving objects: 100% (895/895), 195.97 KiB | 1.13 MiB/s, done.
Resolving deltas: 100% (562/562), done.
www-data@SrvWeb-msage:~$ ls -l
total 40
drwxr-xr-x 2 www-data www-data 4096 Jan 31 2024 Ethane
drwxr-xr-x 2 www-data www-data 4096 Jan 31 2024 Mathis
drwxr-xr-x 2 www-data www-data 4096 Jan 31 2024 Paul
drwxrwxrwx 6 www-data www-data 4096 Apr 24 2024 ProjetAPI
drwxr-xr-x 2 www-data www-data 4096 Jan 31 2024 Ronald
drwxr-xr-x 2 www-data www-data 4096 Jan 31 2024 Teddy
drwxr-xr-x 2 www-data www-data 4096 May 6 2024 html
drwxrwxrwx 6 www-data www-data 4096 May 6 2024 monpanier
drwxr-xr-x 2 www-data www-data 4096 Feb 7 2024 prenom
drwxr-xr-x 7 www-data www-data 4096 Dec 6 15:13 publicom
www-data@SrvWeb-msage:~$
```

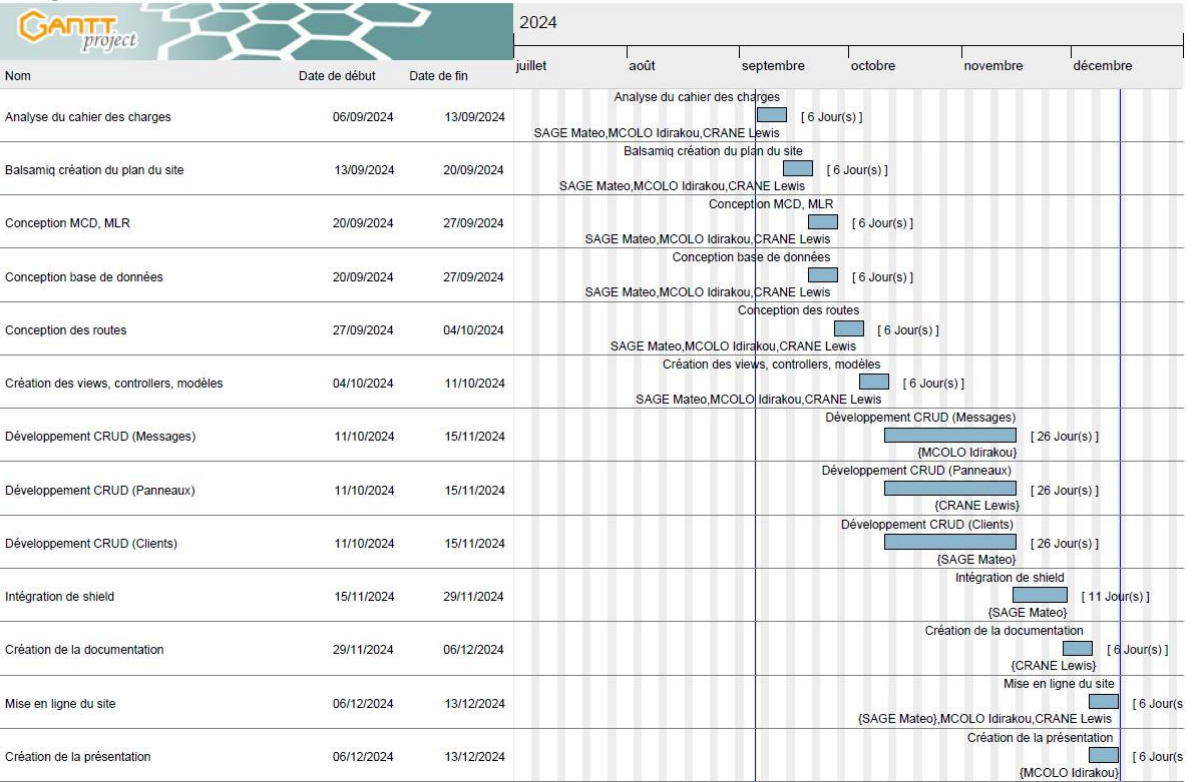
```
root@srvweb-mage:/# apt install php-intl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php8.2 php8.2 php8.2-bz2 php8.2-cli php8.2-common php8.2-curl php8.2-gd php8.2-intl php8.2-mbstring php8.2-mysql php8.2-openssl php8.2-readline php8.2-xml
  php8.2-zip
Suggested packages:
  php-pear
The following NEW packages will be installed:
  php-intl php8.2-intl
The following packages will be upgraded:
  libapache2-mod-php8.2 php8.2 php8.2-bz2 php8.2-cli php8.2-common php8.2-curl php8.2-gd php8.2-mbstring php8.2-mysql php8.2-openssl php8.2-readline php8.2-xml php8.2-zip
13 upgraded, 2 newly installed, 0 to remove and 101 not upgraded.
Need to get 5421 kB of archives.
After this operation, 672 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bookworm-security/main amd64 php-intl all 2:8.2+93 [3664 B]
Get:2 http://security.debian.org bookworm-security/main amd64 php8.2-zip amd64 8.2.26-1-deb12u1 [26.1 kB]
Get:3 http://security.debian.org bookworm-security/main amd64 php8.2-xml amd64 8.2.26-1-deb12u1 [113 kB]
Get:4 http://security.debian.org bookworm-security/main amd64 php8.2-readline amd64 8.2.26-1-deb12u1 [12.4 kB]
Get:5 http://security.debian.org bookworm-security/main amd64 php8.2-openssl amd64 8.2.26-1-deb12u1 [346 kB]
Get:6 http://security.debian.org bookworm-security/main amd64 php8.2-mysql amd64 8.2.26-1-deb12u1 [117 kB]
Get:7 http://security.debian.org bookworm-security/main amd64 php8.2-mbstring amd64 8.2.26-1-deb12u1 [443 kB]
Get:8 http://security.debian.org bookworm-security/main amd64 php8.2-gd amd64 8.2.26-1-deb12u1 [28.8 kB]
Get:9 http://security.debian.org bookworm-security/main amd64 php8.2-curl amd64 8.2.26-1-deb12u1 [35.9 kB]
Get:10 http://security.debian.org bookworm-security/main amd64 php8.2-bz2 amd64 8.2.26-1-deb12u1 [10.6 kB]
Get:11 http://security.debian.org bookworm-security/main amd64 libapache2-mod-php8.2 amd64 8.2.26-1-deb12u1 [1677 kB]
Get:12 http://security.debian.org bookworm-security/main amd64 php8.2-cli amd64 8.2.26-1-deb12u1 [1737 kB]
Get:13 http://security.debian.org bookworm-security/main amd64 php8.2-common amd64 8.2.26-1-deb12u1 [587 kB]
Get:14 http://security.debian.org bookworm-security/main amd64 php8.2-intl amd64 8.2.26-1-deb12u1 [138 kB]
Get:15 http://security.debian.org bookworm-security/main amd64 php8.2 all 8.2.26-1-deb12u1 [45.6 kB]
Fetched 5421 kB in 5s (1066 kB/s)
Reading changelogs... Done
(Reading database ... 35733 files and directories currently installed.)
Preparing to unpack .../00-php8.2-zip_8.2.26-1-deb12u1_amd64.deb ...
Unpacking php8.2-zip (8.2.26-1-deb12u1) over (8.2.7-1-deb12u1) ...
Preparing to unpack .../01-php8.2-xml_8.2.26-1-deb12u1_amd64.deb ...
Unpacking php8.2-xml (8.2.26-1-deb12u1) over (8.2.7-1-deb12u1) ...
Preparing to unpack .../02-php8.2-readline_8.2.26-1-deb12u1_amd64.deb ...
Unpacking php8.2-readline (8.2.26-1-deb12u1) over (8.2.7-1-deb12u1) ...
Preparing to unpack .../03-php8.2-openssl_8.2.26-1-deb12u1_amd64.deb ...
Unpacking php8.2-openssl (8.2.26-1-deb12u1) over (8.2.7-1-deb12u1) ...
Preparing to unpack .../04-php8.2-mysql_8.2.26-1-deb12u1_amd64.deb ...
Unpacking php8.2-mysql (8.2.26-1-deb12u1) over (8.2.7-1-deb12u1) ...
Preparing to unpack .../05-php8.2-mbstring_8.2.26-1-deb12u1_amd64.deb ...
Unpacking php8.2-mbstring (8.2.26-1-deb12u1) over (8.2.7-1-deb12u1) ...
Preparing to unpack .../06-php8.2-gd_8.2.26-1-deb12u1_amd64.deb ...
Unpacking php8.2-gd (8.2.26-1-deb12u1) over (8.2.7-1-deb12u1) ...
Preparing to unpack .../07-php8.2-curl_8.2.26-1-deb12u1_amd64.deb ...
Unpacking php8.2-curl (8.2.26-1-deb12u1) over (8.2.7-1-deb12u1) ...
Preparing to unpack .../08-php8.2-bz2_8.2.26-1-deb12u1_amd64.deb ...
```

Déroulement du projet

Diagramme de Gantt :

Publicom		6 déc. 2024	
Tâches		2	
Nom		Date de début	Date de fin
Analyse du cahier des charges		06/09/2024	13/09/2024
Balsamiq création du plan du site		13/09/2024	20/09/2024
Conception MCD, MLR		20/09/2024	27/09/2024
Conception base de données		20/09/2024	27/09/2024
Conception des routes		27/09/2024	04/10/2024
Création des views, controllers, modèles		04/10/2024	11/10/2024
Développement CRUD (Messages)		11/10/2024	15/11/2024
Développement CRUD (Panneaux)		11/10/2024	15/11/2024
Développement CRUD (Clients)		11/10/2024	15/11/2024
Intégration de shield		15/11/2024	29/11/2024
Création de la documentation		29/11/2024	06/12/2024
Mise en ligne du site		06/12/2024	13/12/2024
Création de la présentation		06/12/2024	13/12/2024

Diagramme de Gantt



Les difficultés :

Problèmes / Difficultés	Solutions
<ul style="list-style-type: none">Maitrise du frameWork WebCRUD à réaliserRésolution et lecture des messages d’erreurs	<ul style="list-style-type: none">Documentation techniqueConsole outils des navigateur web

Conclusion

Ce projet est une belle occasion de créer une application utile pour les communes clientes de Publicom. On peut rencontrer des difficultés comme apprendre à bien utiliser un Framework web qui était dans notre cas CodeIgniter 4, faire les fonctions CRUD ou comprendre et résoudre les messages d’erreurs.

Mais avec la documentation technique et les outils de console des navigateurs, ces problèmes ont pu être réglés, et on a pu livrer une application pratique et facile à utiliser.