



DOCUMENTATION PUBLICOM

Idirakou, Mateo, Lewis



SEMESTRE 1

BTS SIO

AP3

Table des matières

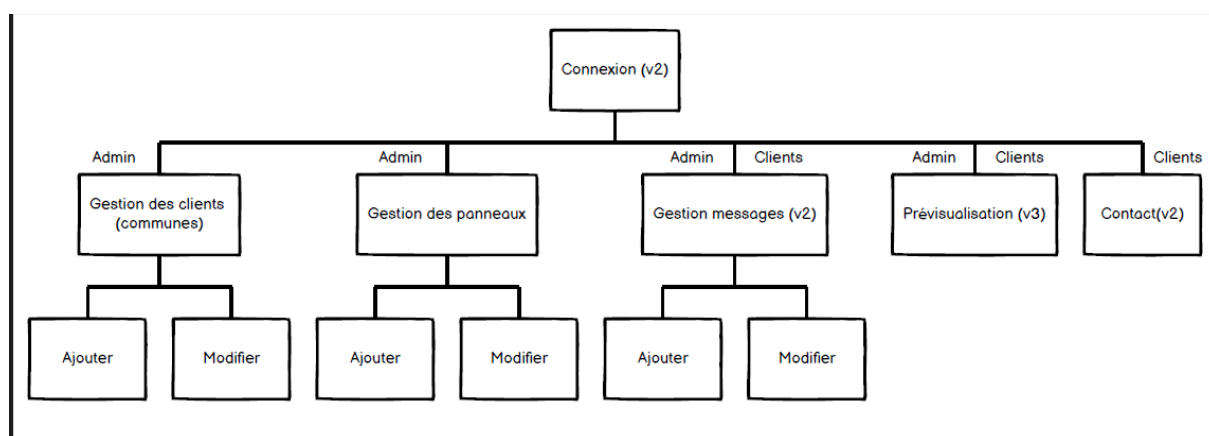
Présentation générale	2
Plan et maquette du site	2
Le dictionnaire de données et le modèle conceptuel de données	8
Le descriptif des techniques de l'application	10
Technologies et langages utilisés :.....	10
Organisation du code :	11
Les controllers :.....	11
Les models :.....	12
Le guide d'utilisation.....	14
Quelques écrans de l'application :	14
Procédures d'installation	20
Déroulement du projet.....	21
Diagramme de Gantt :.....	21
Les difficultés :.....	22
Conclusion	22

Présentation générale

PubliCom est un 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

Ajouter

Nom commune

Responsable

Localisation

Description

Modifier

Supprimer

Nom commune

Responsable

Localisation

Description

Modifier

Supprimer

Nom commune

Responsable

Localisation

Description

Modifier

Supprimer

Nom commune

Responsable

Localisation

Description

Modifier

Supprimer

Ajouter des clients

Deconnexion

Ajouter des clients

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

Gestion des panneaux

Deconnexion

Liste des panneaux

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

Albi
Toulouse
Coutade
Montauban

Messages

Etat message

Disponibilité

Modifier panneaux

Deconnexion

Modifier panneaux

Choisir la commune

Albi
Toulouse
Coutade
Montauban

Messages

Etat message

Disponibilité



Liste des messages

Q

Ajouter

commune	Responsable	Description	Message	Modifier	Supprimer
commune	Responsable	Description	Message	Modifier	Supprimer
commune	Responsable	Description	Message	Modifier	Supprimer
commune	Responsable	Description	Message	Modifier	Supprimer



Liste des messages

Ajouter

Responsable	Description	Message	Modifier	Supprimer
Responsable	Description	Message	Modifier	Supprimer
Responsable	Description	Message	Modifier	Supprimer
Responsable	Description	Message	Modifier	Supprimer

Ajouter messages(v2)

Deconnexion

Ajouter messages

Messages

A[°] A[°] Aa A₀
A₀ A₀ A₀ A₀

Valider

Modifier panneaux (v2)

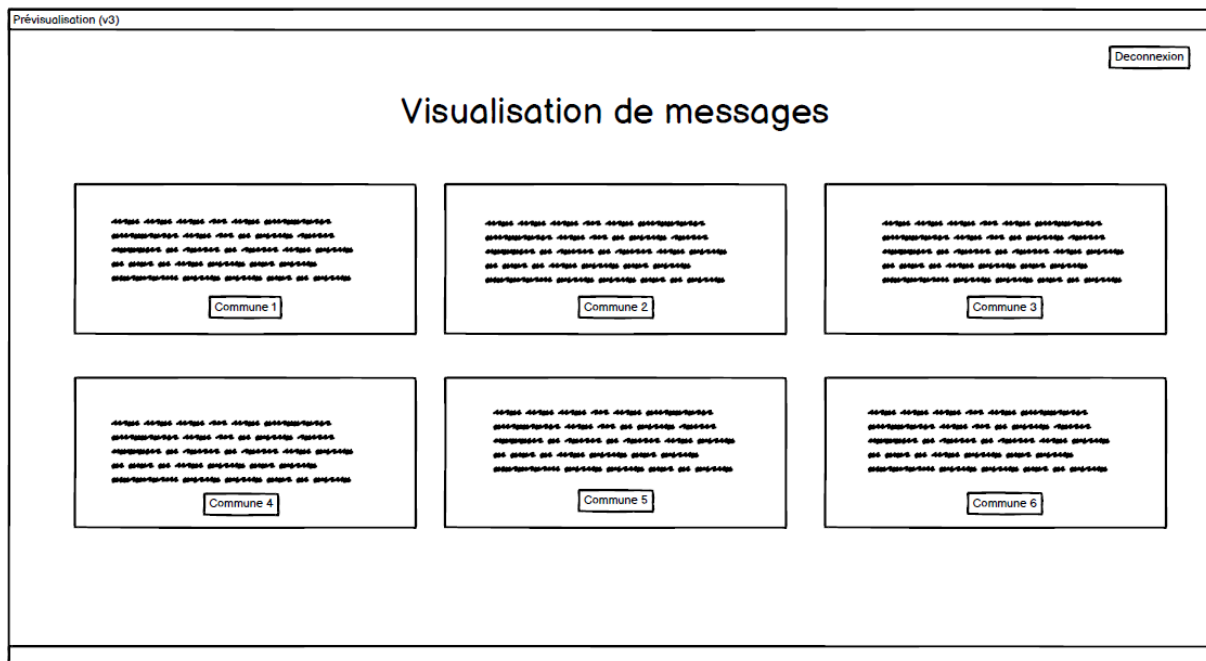
Deconnexion

Modifier messages

Messages

A[°] A[°] Aa A₀
A₀ A₀ A₀ A₀

Modifier

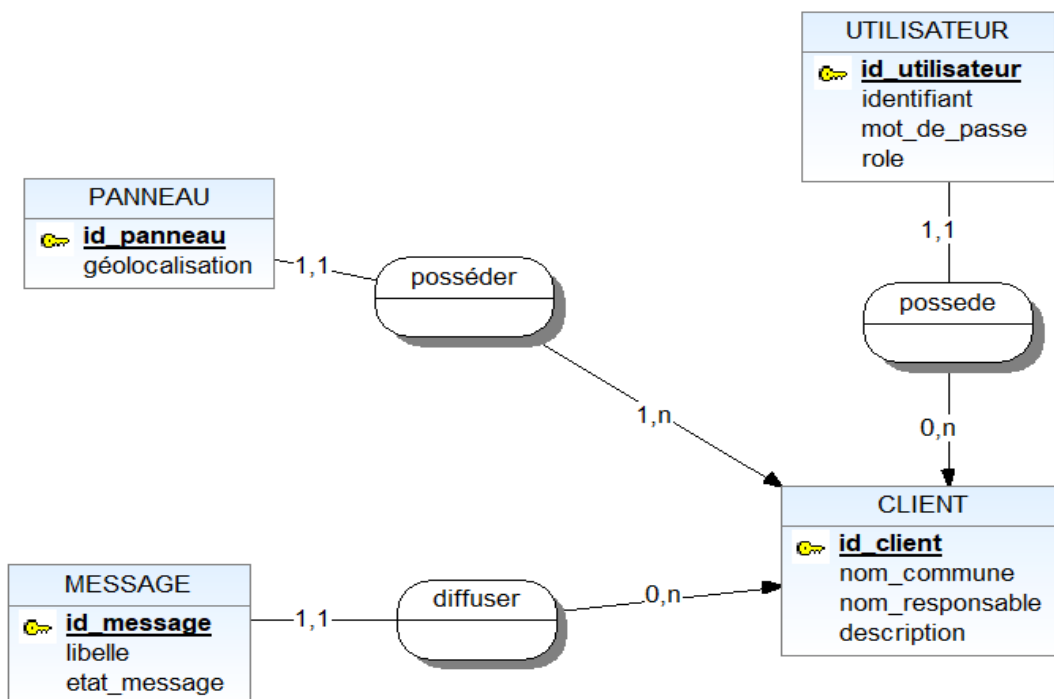


Le dictionnaire de données et le modèle conceptuel de données

Dictionnaire de données :

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 :



Modèle relationnel de données :

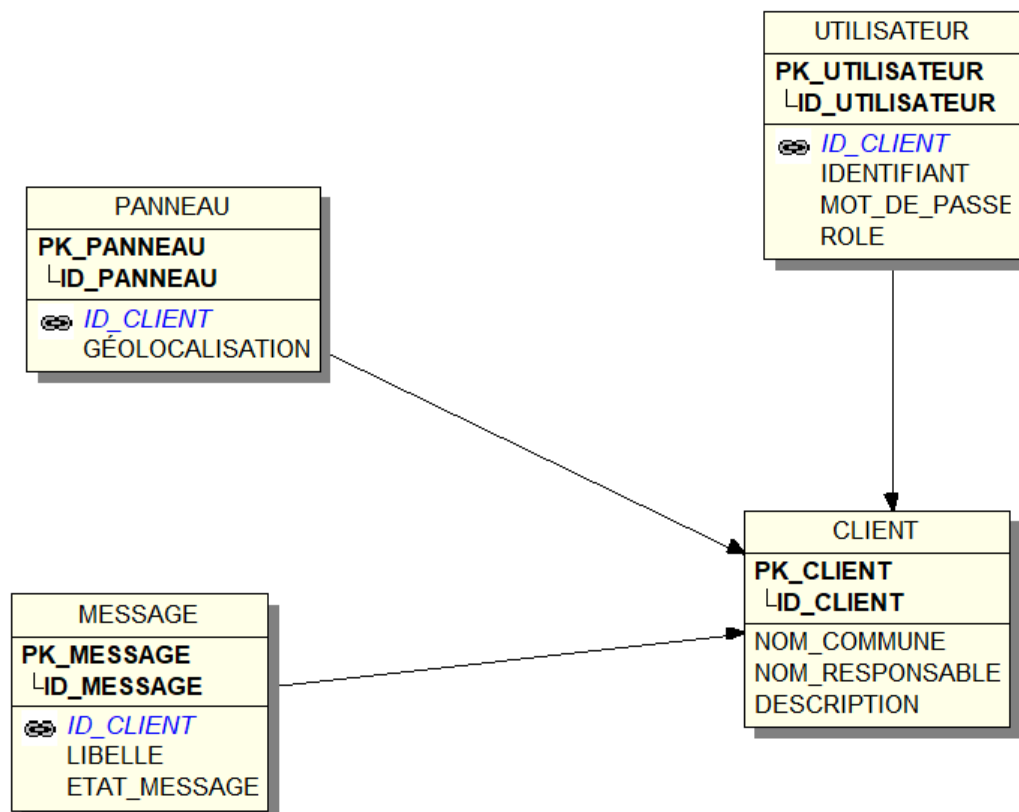
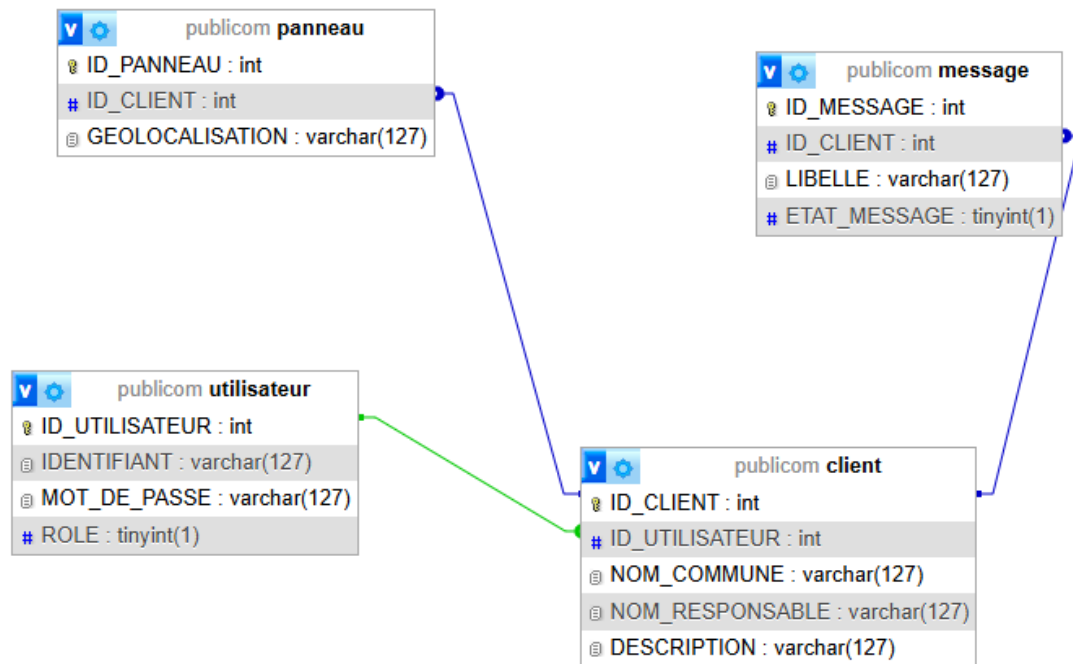


Schéma concepteur de la base de données :



Le descriptif des techniques de l'application

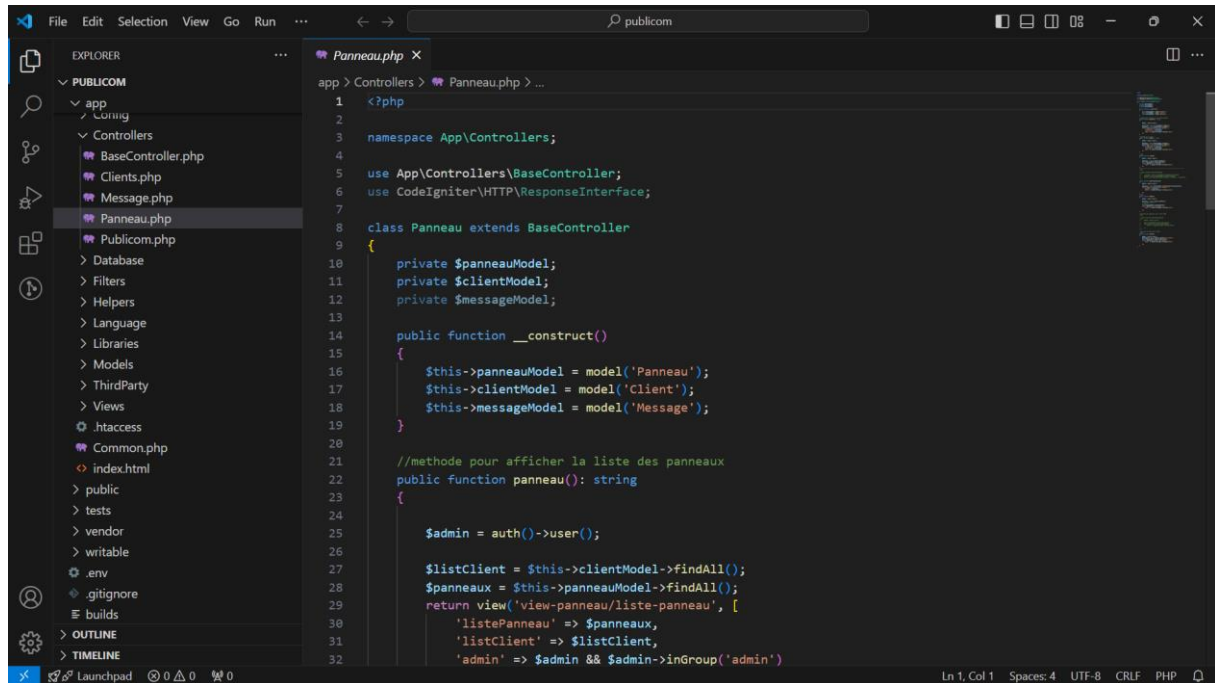
Technologies et langages utilisés :

Outils	Langages
<ul style="list-style-type: none"> - Wampserver64 (serveur web local) - Visual Studio Code (développement & programmation) - PhpMyAdmin (base de données) - CodeIgniter 4 (FrameWork Web) - Firefox (Navigateur Web) - GanttProject (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 :

Panneau :



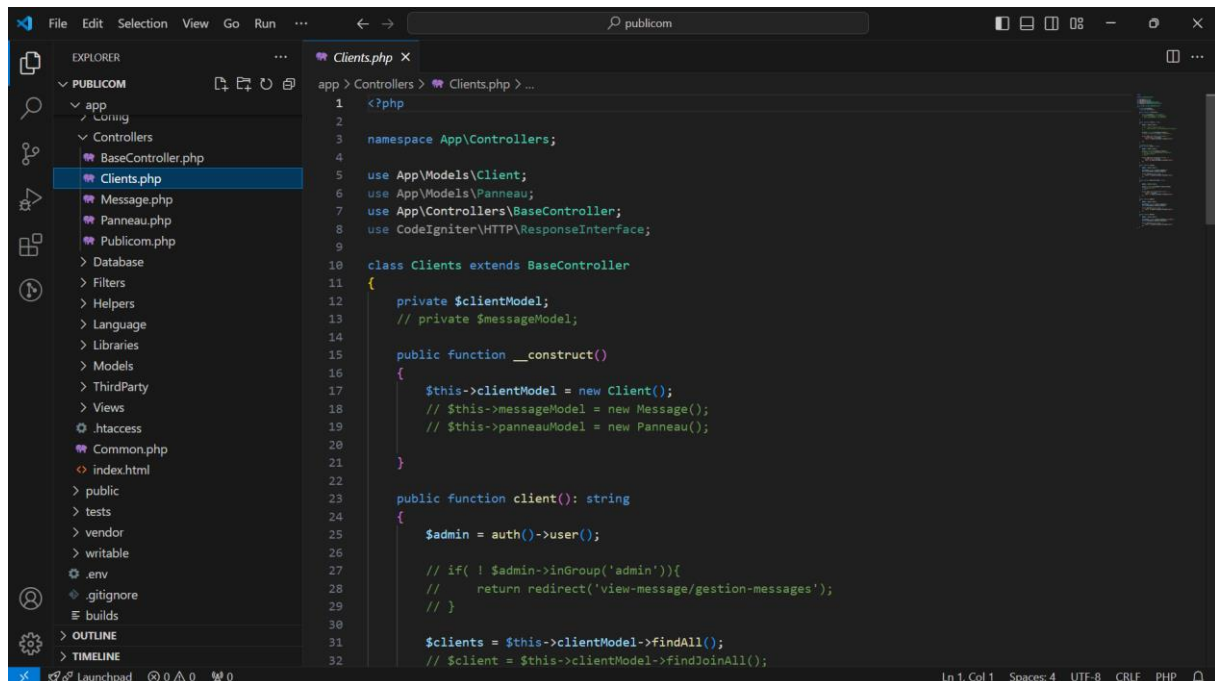
The screenshot shows a code editor with the file explorer on the left and the code editor on the right. The file explorer shows the project structure with the following folders and files:

- PUBLICCOM
 - app
 - Controllers
 - Panneau.php (selected)
 - Publicom.php
 - Database
 - Filters
 - Helpers
 - Language
 - Libraries
 - Models
 - ThirdParty
 - Views
 - .htaccess
 - Common.php
 - index.html
 - public
 - tests
 - vendor
 - writable
 - .env
 - .gitignore
 - builds
 - OUTLINE
 - TIMELINE

The code editor shows the content of Panneau.php:

```
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         $admin = auth()->user();
25
26         $listClient = $this->clientModel->findAll();
27         $panneaux = $this->panneauModel->findAll();
28         return view('view-panneau/liste-panneau', [
29             'listePanneau' => $panneaux,
30             'listClient' => $listClient,
31             'admin' => $admin && $admin->inGroup('admin')
32         ]);
33     }
34 }
```

Clients :



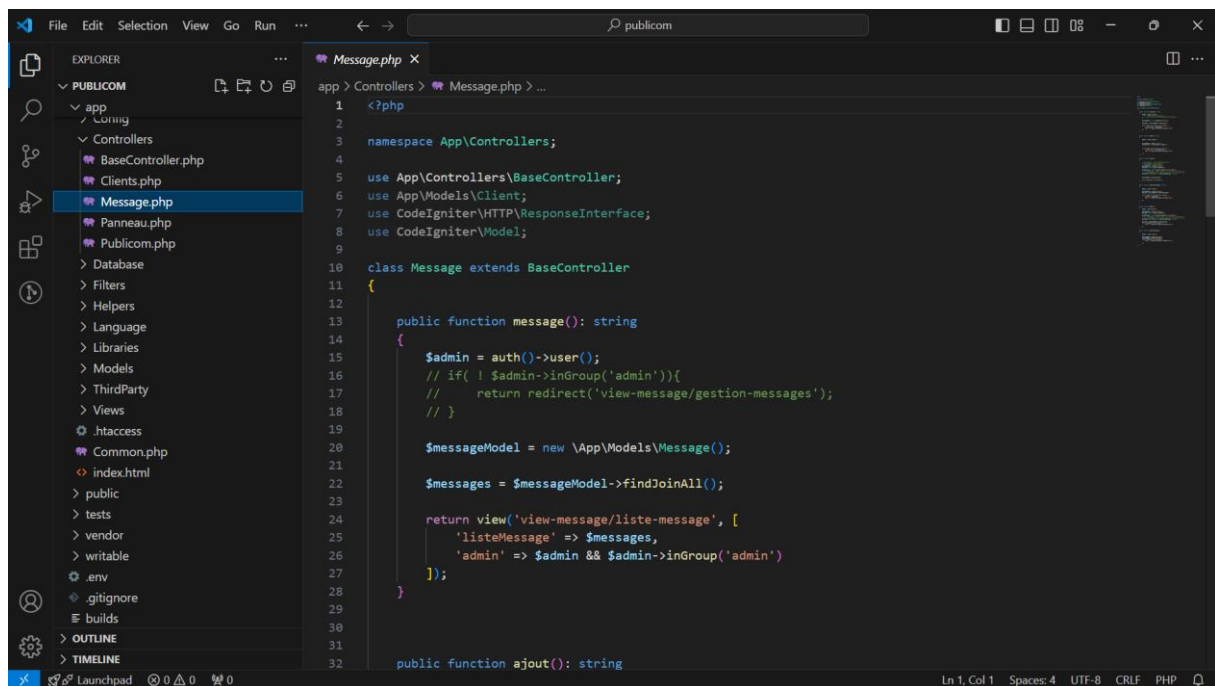
The screenshot shows a code editor with the file explorer on the left and the code editor on the right. The file explorer shows the project structure with the following folders and files:

- PUBLICCOM
 - app
 - Controllers
 - Clients.php (selected)
 - Message.php
 - Panneau.php
 - Publicom.php
 - Database
 - Filters
 - Helpers
 - Language
 - Libraries
 - Models
 - ThirdParty
 - Views
 - .htaccess
 - Common.php
 - index.html
 - public
 - tests
 - vendor
 - writable
 - .env
 - .gitignore
 - builds
 - OUTLINE
 - TIMELINE

The code editor shows the content of Clients.php:

```
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 }
```

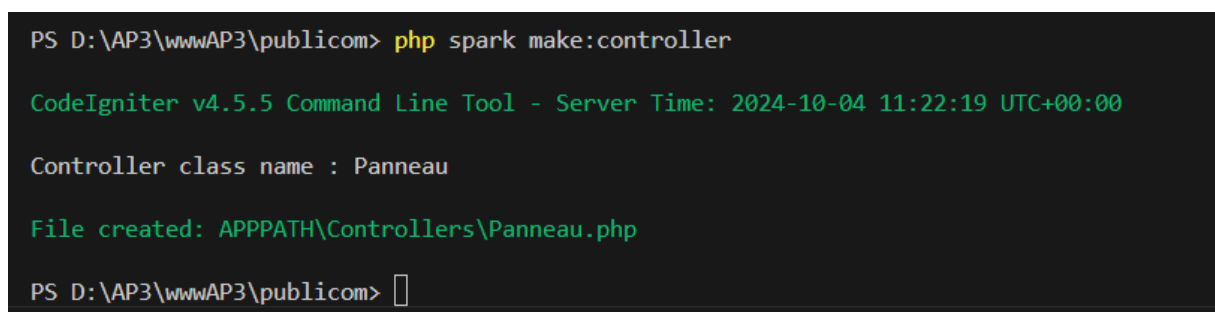
Message :



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 'Controllers'. The main editor displays the following PHP code:

```
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 commands and output:

```
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

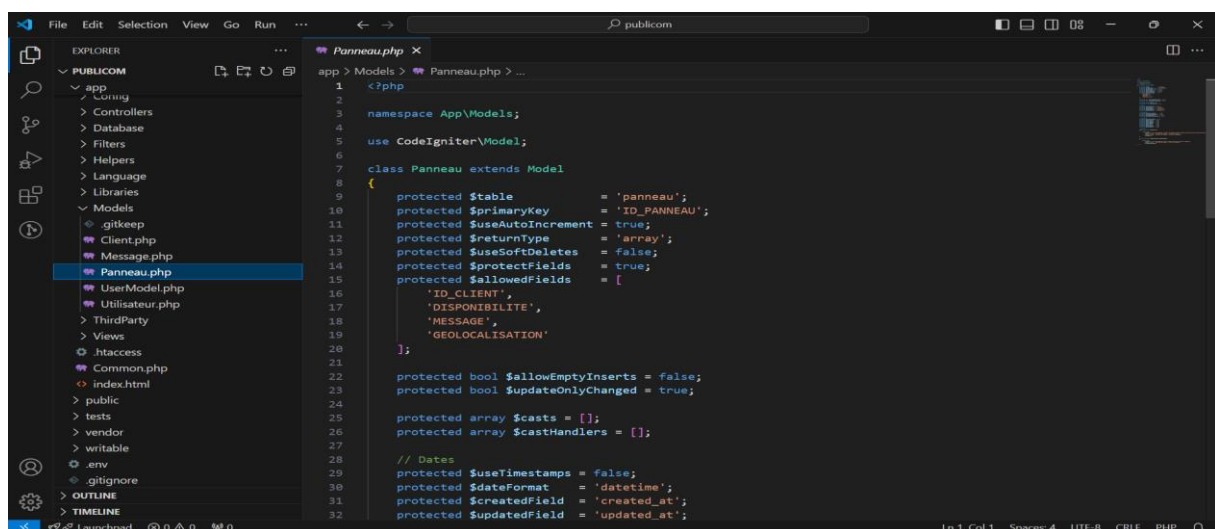
Controller class name : Panneau

File created: APPPATH\Controllers\Panneau.php

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

Les models :

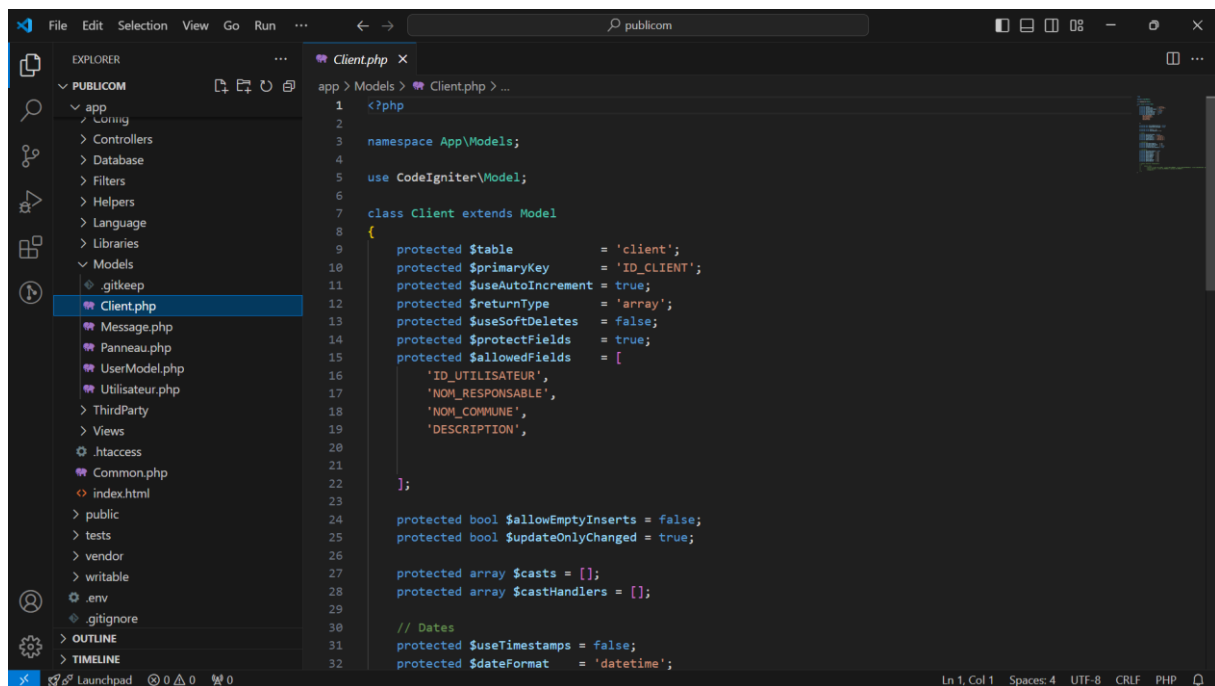
Panneau :



The screenshot shows the Visual Studio Code editor with the 'Panneau.php' file open. The Explorer sidebar on the left shows the project structure with 'Panneau.php' selected under 'Models'. The main editor displays the following PHP code:

```
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';
```

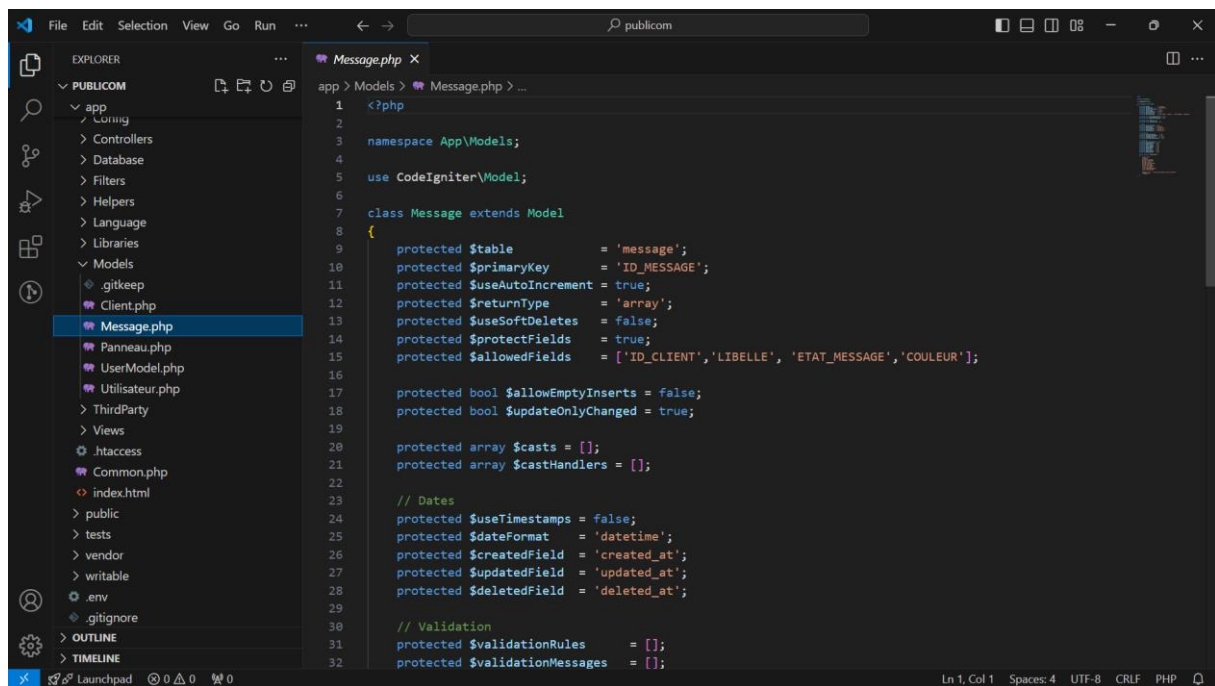
Client :



The screenshot shows an IDE window with the Explorer on the left and the Client.php file open in the editor. The Explorer shows the project structure with the Models directory expanded. The Client.php file contains the following code:

```
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';
```

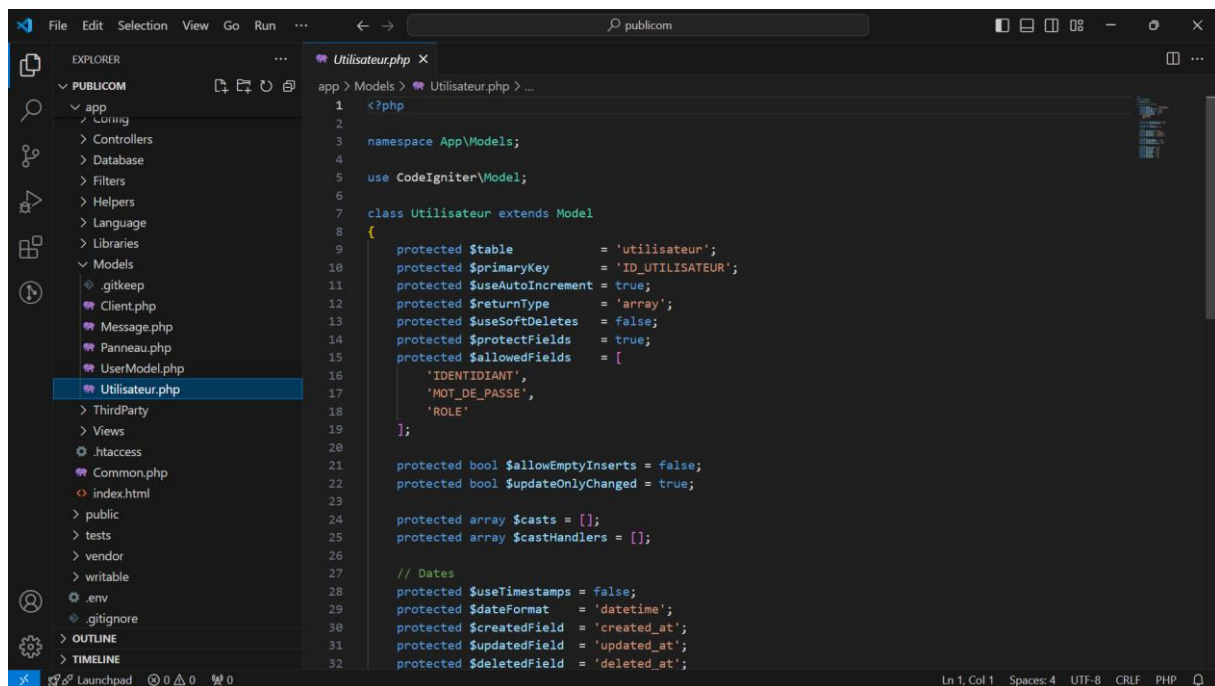
Message :



The screenshot shows an IDE window with the Explorer on the left and the Message.php file open in the editor. The Explorer shows the project structure with the Models directory expanded. The Message.php file contains the following code:

```
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    protected bool $allowEmptyInserts = false;
18    protected bool $updateOnlyChanged = true;
19
20    protected array $casts = [];
21    protected array $castHandlers = [];
22
23    // Dates
24    protected $useTimestamps = false;
25    protected $dateFormat    = 'datetime';
26    protected $createdField  = 'created_at';
27    protected $updatedField  = 'updated_at';
28    protected $deletedField  = 'deleted_at';
29
30    // Validation
31    protected $validationRules = [];
32    protected $validationMessages = [];
```

Utilisateur :

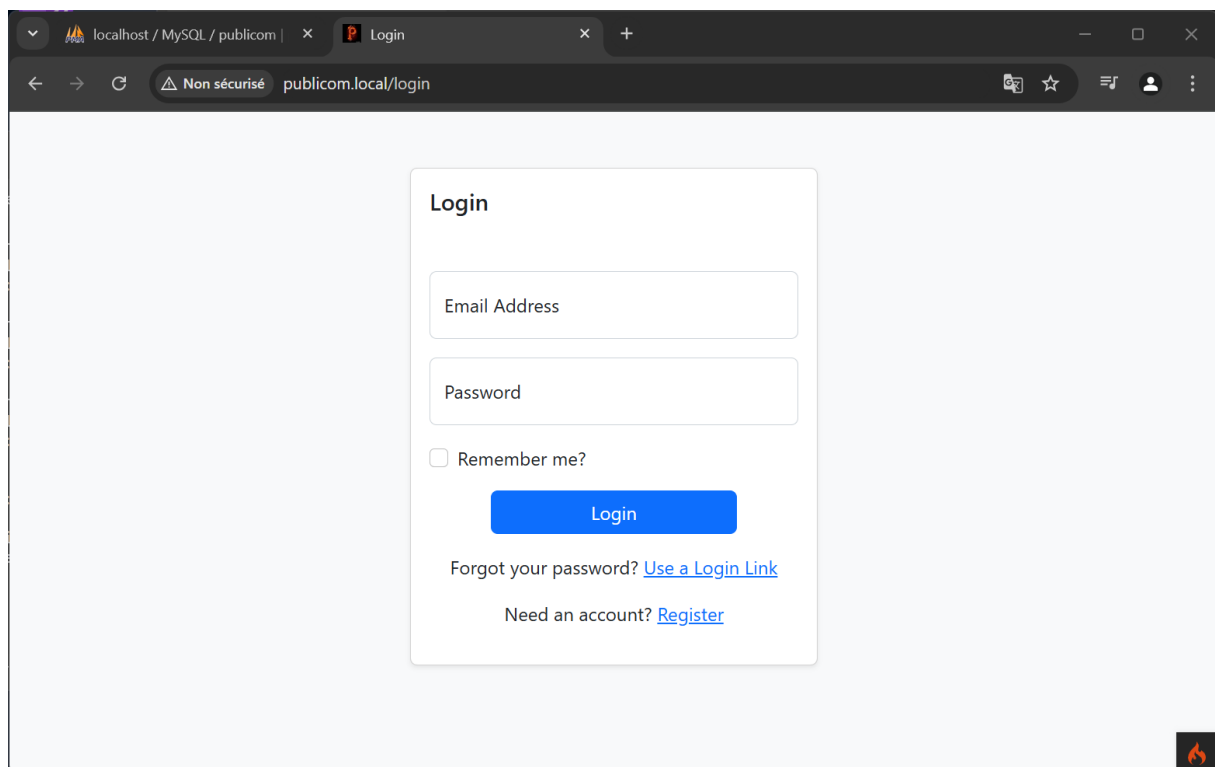


```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Utilisateur extends Model
8 {
9     protected $table = 'utilisateur';
10    protected $primaryKey = 'ID_UTILISATEUR';
11    protected $useAutoIncrement = true;
12    protected $returnType = 'array';
13    protected $useSoftDeletes = false;
14    protected $protectedFields = 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 :



localhost / MySQL / publicom | x Login x +

Non sécurisé publicom.local/login

Login

Email Address

Password

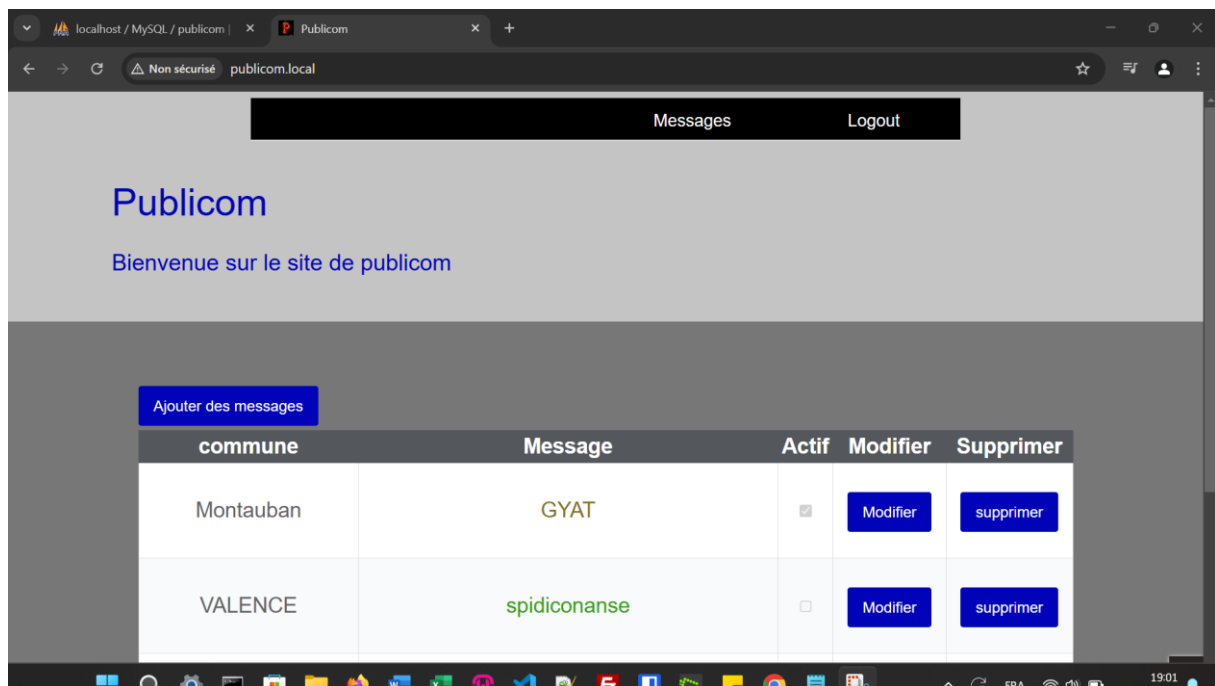
☐ Remember me?

Login

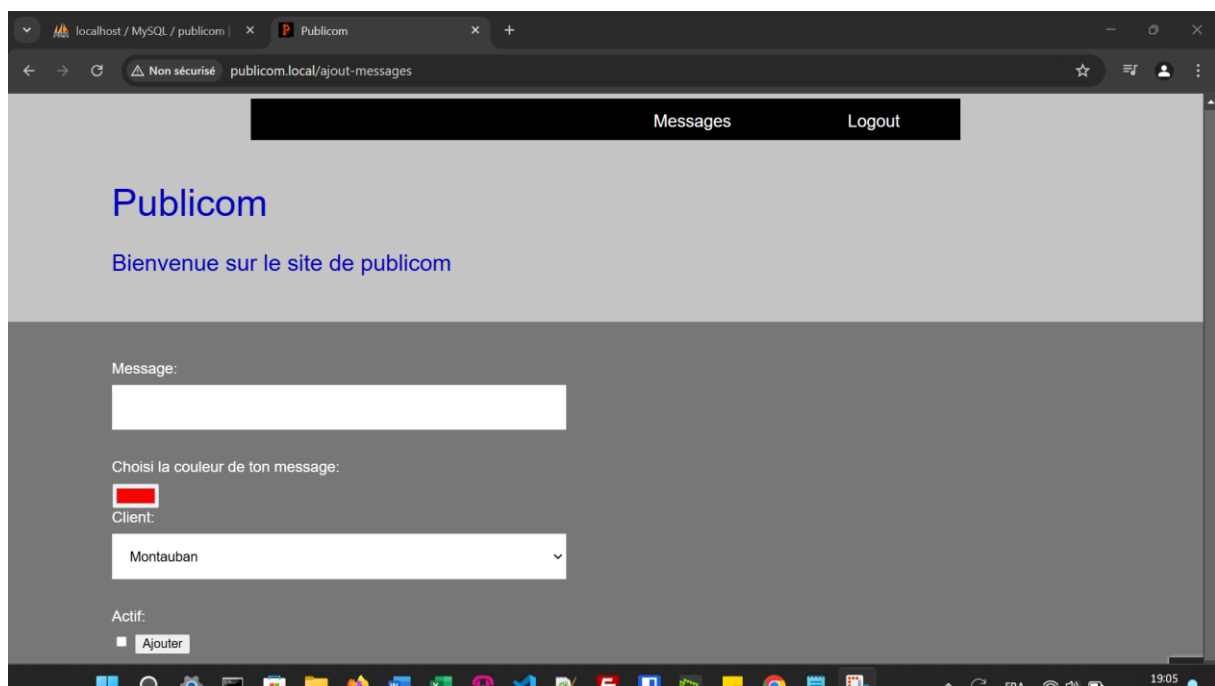
Forgot your password? [Use a Login Link](#)

Need an account? [Register](#)

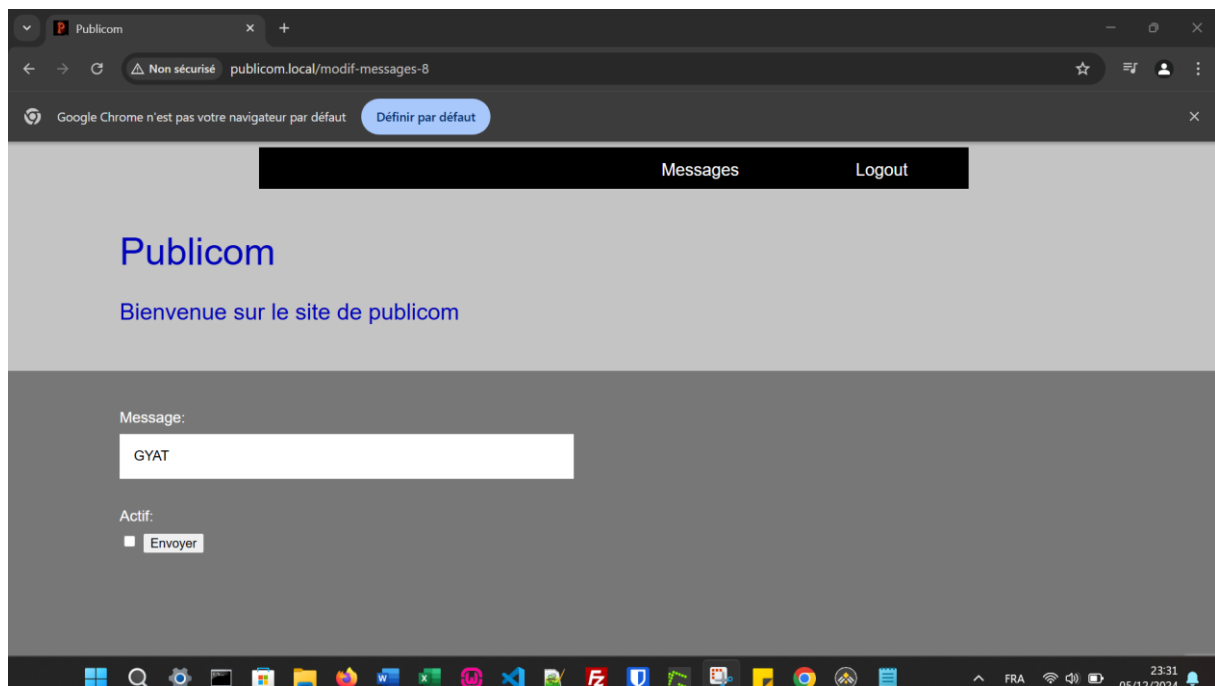
Ecran liste des messages quand on est connecté en tant qu'utilisateur :



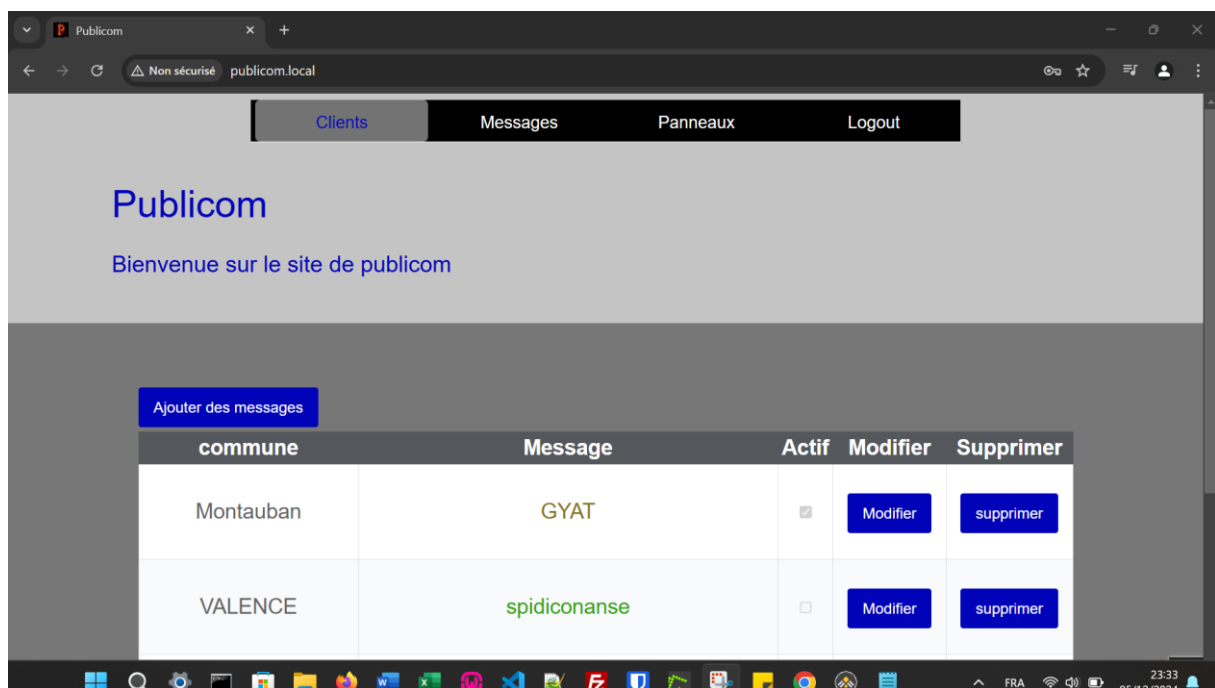
Ecran d'ajout de message en tant qu'utilisateur :



Ecran modifier de message en tant qu'utilisateur :



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



Ecran listes des clients :

The screenshot shows a web browser window with the URL `publicom.local/gestion-clients`. The page has a navigation bar with links for **Clients**, **Messages**, **Panneaux**, and **Logout**. Below the navigation bar, the page title is **Publicom** with a subtitle **Bienvenue sur le site de publicom**. The main content area is titled **GESTIONS CLIENTS** and features a button **Ajouts Clients**. Below this is a table with the following data:

NOM COMMUNE	RESPONSABLE	DESCRIPTION	MODIFIER	SUPPRIMER
Montauban	Teddy	efesfezfef	Modifier	Supprimer
VALENCE	MATHIS	BURGIR	Modifier	Supprimer

The Windows taskbar at the bottom shows the time as 23:43 on 05/12/2024.

Ecran ajout de clients :

The screenshot shows a web browser window with the URL `publicom.local/ajout-clients`. The page title is **AJOUTS CLIENTS**. The main content area contains a form with the following fields:

- Nom commune :** A text input field with a dotted line indicating the start of the input.
- Nom responsable :** A text input field with a dotted line indicating the start of the input.
- Description :** A larger text input field.
- Valider** button.

The Windows taskbar at the bottom shows the time as 23:45 on 05/12/2024.

Ecran de modification des clients :

MODIFICATION CLIENT

Nom commune:
VALENCE

Nom responsable:
MATHIS

Description:
BURGIR

Valider

Ecran liste des panneaux :

Publicom
Bienvenue sur le site de publicom

Gestion des Panneaux

Ajouts Panneaux

Geolocalisation	Nom_Commune	Modifier	Supprimer
88;97	VALENCE	modifier	supprimer
12;22	CASTELSARRASIN	modifier	supprimer

Ecran de modification des panneaux :

The screenshot shows a web browser window with the URL `publicom.local/modif-panneau-1`. The application has a dark navigation bar with links for 'Clients', 'Messages', 'Panneaux', and 'Logout'. The main content area has a light gray header with the 'Publicom' logo and the text 'Bienvenue sur le site de publicom'. Below this, there is a form for editing a panel. The form contains a text input field with the value 'VALENCE', a 'Geolocalisation' label, and another text input field with the value '88,97'. At the bottom of the form, there is a 'Valider' label and a 'Valider' button. The Windows taskbar is visible at the bottom of the screen, showing various application icons and the system clock indicating 23:52 on 05/12/2024.

Publicom

Bienvenue sur le site de publicom

VALENCE

Geolocalisation

88,97

Valider

Valider

Ecran d'ajout de panneaux :

The screenshot shows a web browser window with the URL `publicom.local/ajout-panneaux`. The application has a dark navigation bar with links for 'Clients', 'Messages', 'Panneaux', and 'Logout'. The main content area has a light gray header with the 'Publicom' logo and the text 'Bienvenue sur le site de publicom'. Below this, there is a form for adding a new panel. The form contains a label 'Choisir un nom de commune :', a dropdown menu with the value '--Choisissez un client--', a 'Geolocalisation' label, and a text input field. At the bottom of the form, there is a 'Valider' label and a 'Valider' button. The Windows taskbar is visible at the bottom of the screen, showing various application icons and the system clock indicating 23:53 on 05/12/2024.

Publicom

Bienvenue sur le site de publicom

Choisir un nom de commune :

--Choisissez un client--

Geolocalisation

Valider

Valider

Procédures d'installation

Mise en ligne sur le serveur Proxmox de Mateo :

The screenshot shows the Proxmox VE web interface. On the left, a tree view lists various virtual machines and containers. The main panel displays the console of container 'cesar'. The terminal output shows the installation of Composer version 2.5.3. The user runs the command `php -r "eval('?'>.file_get_contents('https://getcomposer.org/installer'));"` to download the installer, then `mv composer.phar /usr/local/bin/composer` to move it to the system path, and finally `composer --version` to verify the installation. The output confirms that Composer version 2.5.3 is installed and ready to use.

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 ProjetAP1
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 prenomns
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 ProjetAP1
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 prenomns
drwxr-xr-x 7 www-data www-data 4096 Dec 6 15:13 publicom
www-data@SrvWeb-msage:~$
```

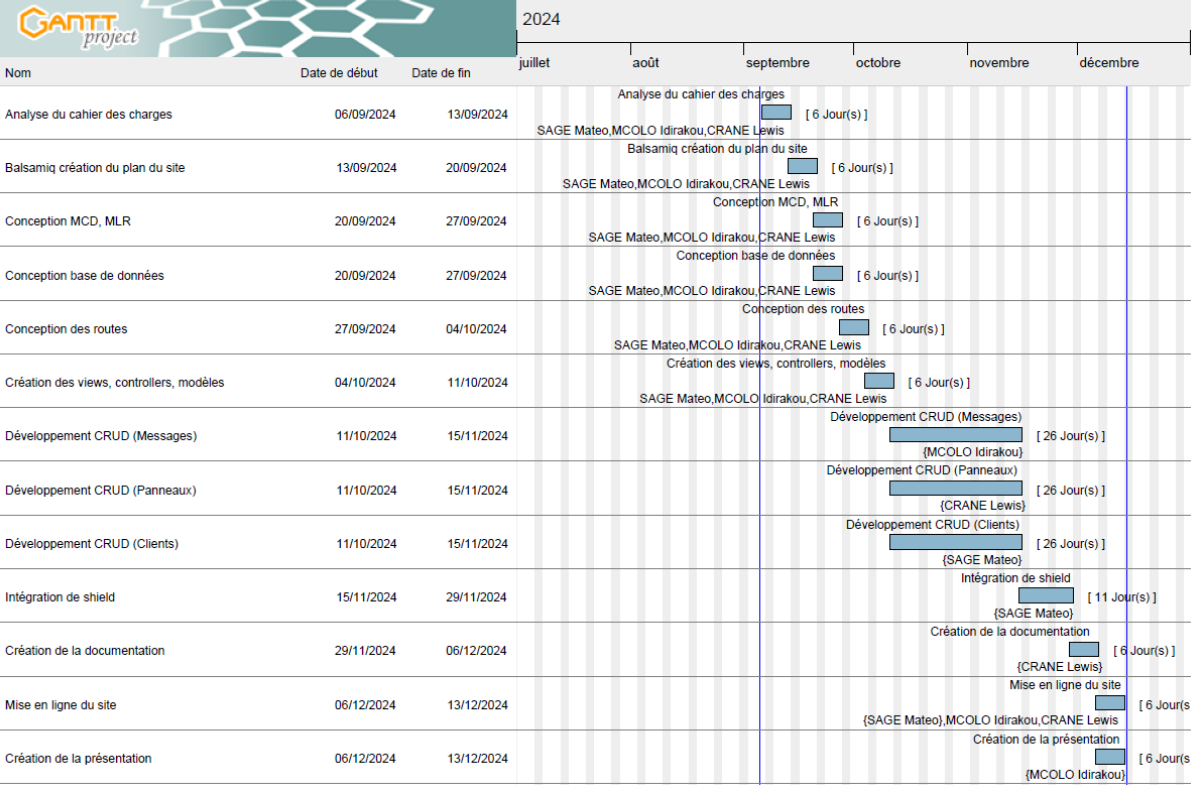
```
root@srvWeb-masage:~# apt install php-intl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php8.2 php8.2 php8.2-bz2 php8.2-cli php8.2-common php8.2-curl php8.2-gd php8.2-intl php8.2-mbstring 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/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 [697 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.