

DOSSIER PROJET

LOUAANABI ABDESLEM

Formation DWM 2020-2021

1) EXPRESSION DES BESOINS ET CAHIER DES CHARGES

Projet:

-Projet réalisé en symfony 5

-Ecran:

-Création de compte

-Affichage des anecdotes successives

-Affichage d'1 seule anecdote avec

illustration et formulaire de commentaire:

Contenu :

-BDD MySql


-Authentification

-Création de compte et de commentaires des anecdotes

2) OUTILS UTILISÉS:

- SYMFONY VERSION 5.2.9
- HTML5 ET CSS3
- JAVASCRIPT
- LIBRAIRIE BOOTSTRAPS
- BASE DE DONNÉES MYSQL

3) ZONING PAGE D'ACCEUIL

	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<div><div><input type="text"/></div><div><input type="text"/></div><div><input type="text"/></div><div><input type="text"/></div><div><input type="text"/></div></div>						

4) WIREFRAME PAGE D'ACCEUIL

LOGO	Anecdotes	Proposer une anecdote	Connexion	Inscription	Recherche..	Search
------	-----------	-----------------------	-----------	-------------	-------------	--------

Titre de l'anecdote

Anecdote: ntkoznfdsvJFKLEZ\$JFKLMZhgilmsghlfem hgilremqghlrmqhglmfq nhgilrmehgilkdf hgmrehilmgrez

Lire la suite

Titre de l'anecdote

Anecdote: jfkjknfdmqhvgkfdmqjklgvdnfilkemqhlidhqilmfnfzemqhiglmfejqgfnfmqhilmreq

Lire la suite

Titre de l'anecdote

Anecdote: hikglmrfqgh bhgjrkehjgrfhugrflqhbqj nhfglmsqngrfqjglfeq hgujrlshjkgts e

Lire la suite

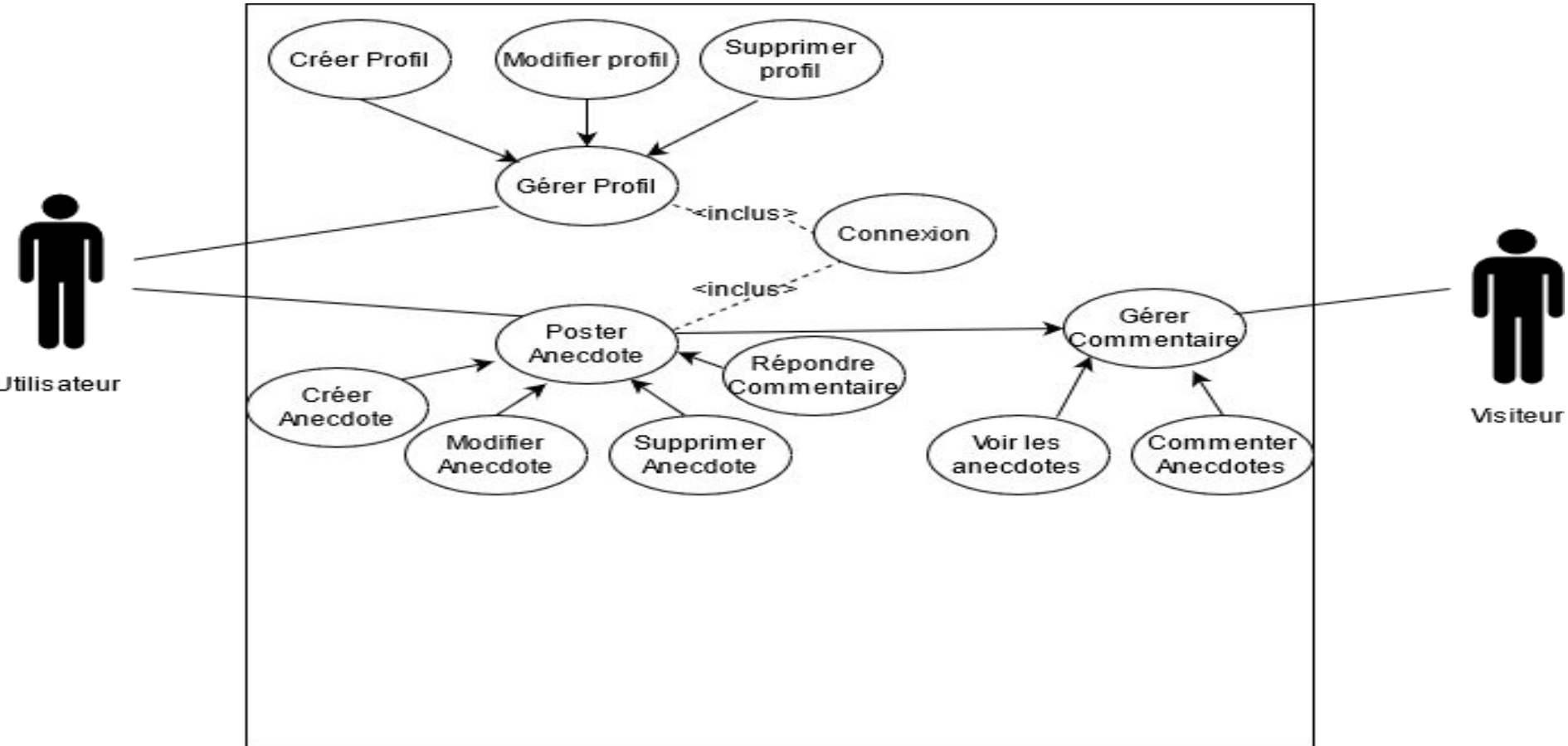
Titre de l'anecdote

Anecdote : hgjrmeghj gjfds nhgjkfezhjgkkrz gjezgdzmlhgjerkzhgrhgjrezngjkrz

Lire la suite

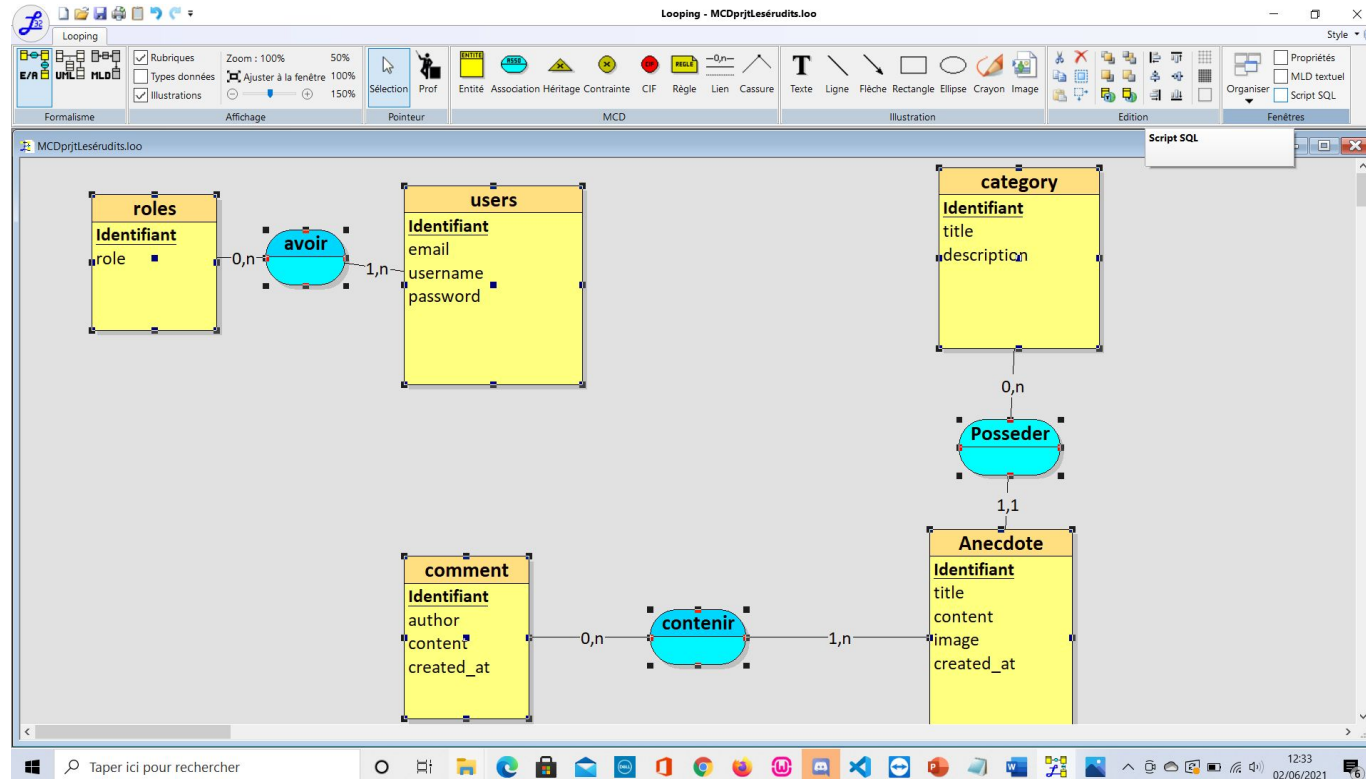
Text

5) USECASE DIAGRAM



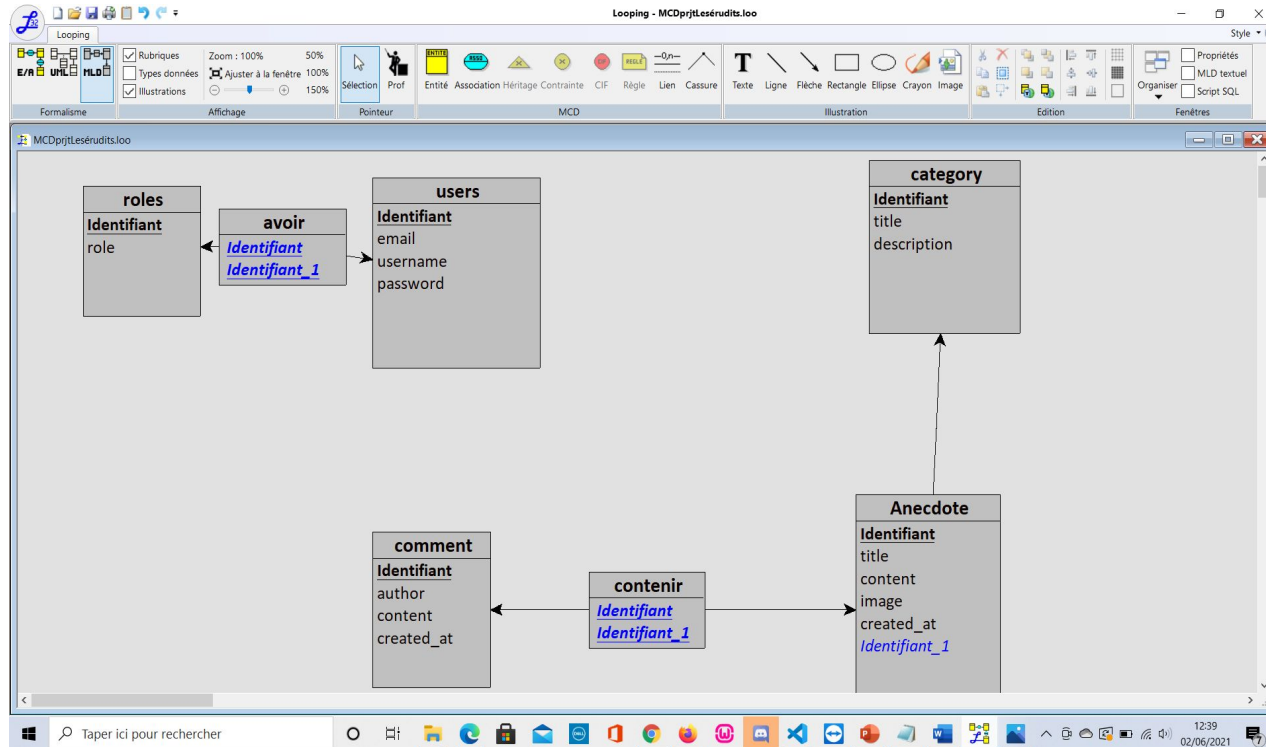
6) MODÈLE CONCEPTUEL DE DONNÉES:

Le **modèle conceptuel des données (MCD)** a pour but d'écrire de façon formelle les **données** qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des **données**, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités.

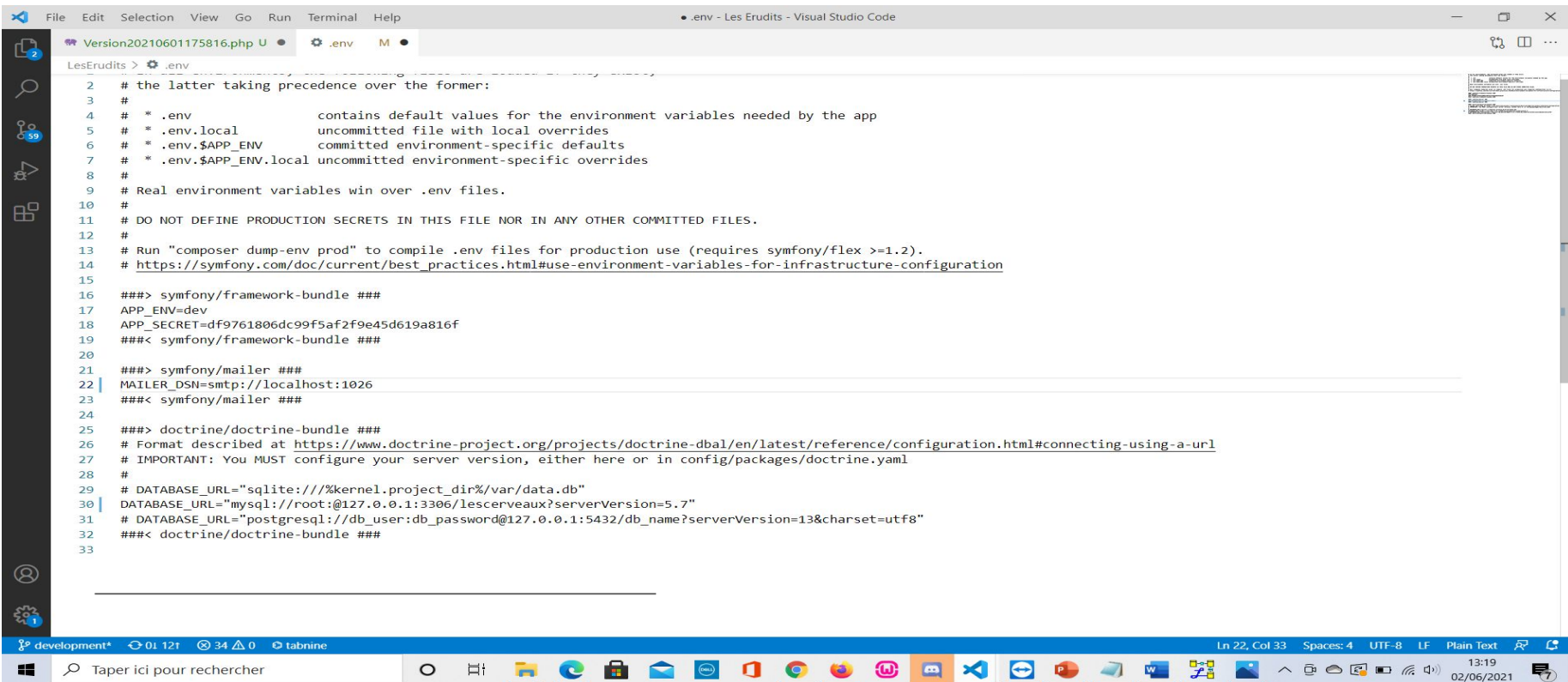


7) MODÈLE LOGIQUE DE DONNÉES:

Modélisation logique des données (MLD) Le modèle logique de données est une étape de la conception qui consiste à décrire la structure des données utilisées sans faire référence à un langage de programmation. Il s'agit de préciser le type de données utilisées lors des traitements



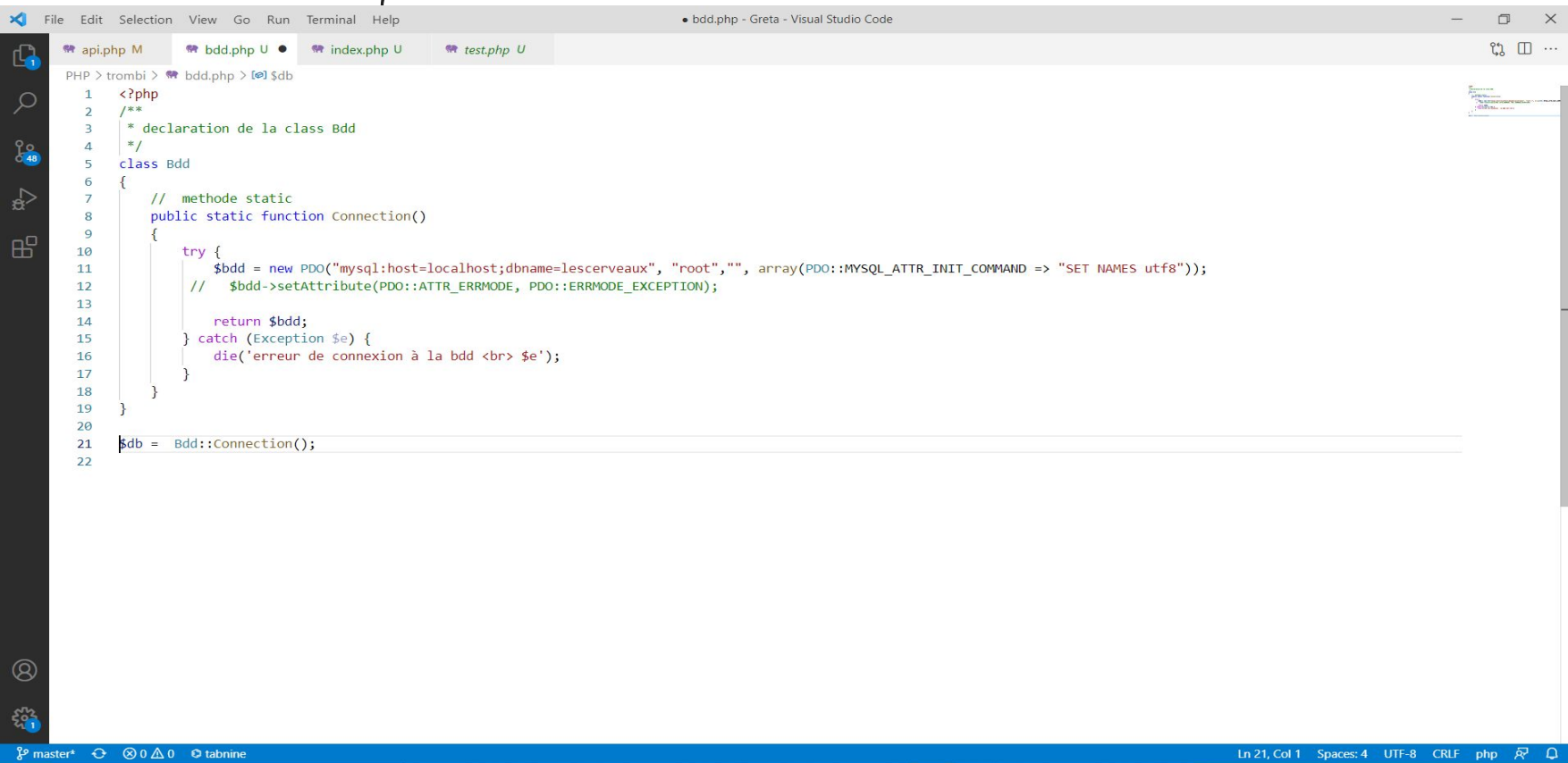
8) MISE EN RELATION AVEC LA BASE DE DONNÉE:



```
2 # the latter taking precedence over the former:
3 #
4 # * .env contains default values for the environment variables needed by the app
5 # * .env.local uncommitted file with local overrides
6 # * .env.$APP_ENV committed environment-specific defaults
7 # * .env.$APP_ENV.local uncommitted environment-specific overrides
8 #
9 # Real environment variables win over .env files.
10 #
11 # DO NOT DEFINE PRODUCTION SECRETS IN THIS FILE NOR IN ANY OTHER COMMITTED FILES.
12 #
13 # Run "composer dump-env prod" to compile .env files for production use (requires symfony/flex >=1.2).
14 # https://symfony.com/doc/current/best\_practices.html#use-environment-variables-for-infrastructure-configuration
15
16 ###> symfony/framework-bundle ###
17 APP_ENV=dev
18 APP_SECRET=df9761806dc99f5af2f9e45d619a816f
19 ###< symfony/framework-bundle ###
20
21 ###> symfony/mailer ###
22 MAILER_DSN=smtp://localhost:1026
23 ###< symfony/mailer ###
24
25 ###> doctrine/doctrine-bundle ###
26 # Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url
27 # IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
28 #
29 # DATABASE_URL="sqlite:///kernel.project_dir%/var/data.db"
30 DATABASE_URL="mysql://root:@127.0.0.1:3306/lescerveaux?serverVersion=5.7"
31 # DATABASE_URL="postgresql://db_user:db_password@127.0.0.1:5432/db_name?serverVersion=13&charset=utf8"
32 ###< doctrine/doctrine-bundle ###
33
```

9) MISE EN RELATION BDD ET PROJET EN PHP :

En PHP, la mise en relation avec la bdd s'établit par le biais d'une méthode statique comme ci-dessous :



```
File Edit Selection View Go Run Terminal Help • bdd.php - Greta - Visual Studio Code
api.php M bdd.php U index.php U test.php U
PHP > trombi > bdd.php > $db
1 <?php
2 /**
3  * declaration de la class Bdd
4  */
5 class Bdd
6 {
7     // methode static
8     public static function Connection()
9     {
10         try {
11             $bdd = new PDO("mysql:host=localhost;dbname=leserveaux", "root", "", array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"));
12             // $bdd->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
13
14             return $bdd;
15         } catch (Exception $e) {
16             die('erreur de connexion à la bdd <br> $e');
17         }
18     }
19 }
20
21 $db = Bdd::Connection();
22
```

Ln 21, Col 1 Spaces: 4 UTF-8 CRLF php

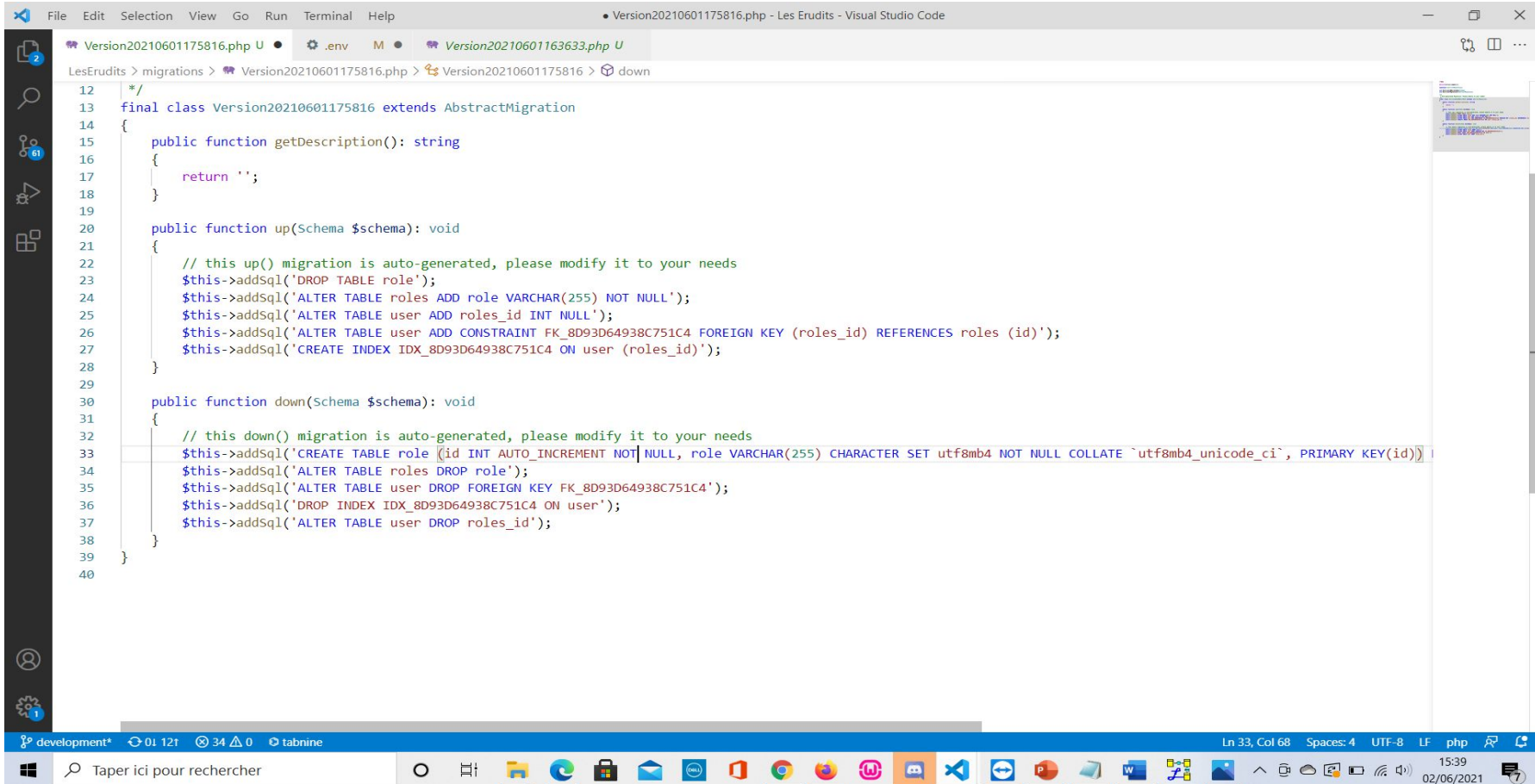
10) INTERACTIONS ENTRE LE LANGAGE ET LA BDD SUR SYMFONY:

Sur Symfony les interactions entre le langage et la BDD se font à travers la CLI. Symfony utilise un ORM qui est une brique de logiciel qui fait la liaison entre le langage et la BDD par l'utilisation de classes et d'objets.

L'ORM de Symfony s'appelle Doctrine qui fournit les outils "Entity" représentant les tables de notre BDD, "Manager" nous permettant la manipulation des lignes qui la composent entre autres l'insertion, la mise à jour ou encore la suppression de lignes, puis de "Repository" nous permettant la sélection d'éléments de la Database.

Une fois nos Tables nommées et leurs champs complétés, Symfony utilise les migrations qui sont des fichiers regroupant l'ensemble des requêtes SQL qui vont créer nos tables sur l'interface PHPMyAdmin.

11) EXEMPLE DE FICHER MIGRATION

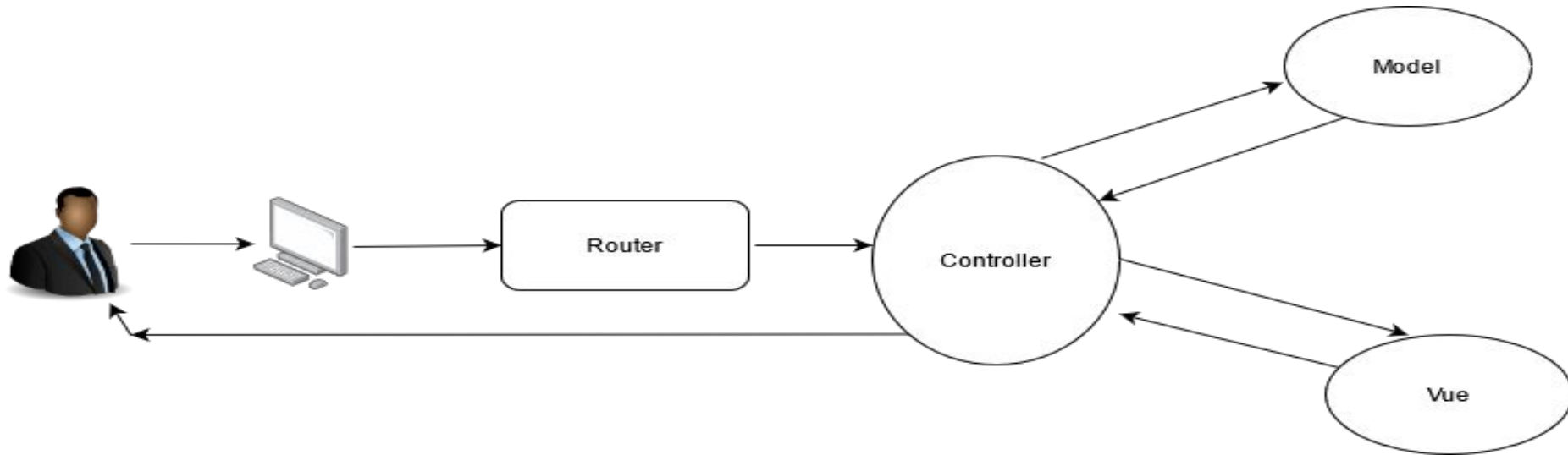


```
12  */
13  final class Version20210601175816 extends AbstractMigration
14  {
15      public function getDescription(): string
16      {
17          return '';
18      }
19
20      public function up(Schema $schema): void
21      {
22          // this up() migration is auto-generated, please modify it to your needs
23          $this->addSql('DROP TABLE role');
24          $this->addSql('ALTER TABLE roles ADD role VARCHAR(255) NOT NULL');
25          $this->addSql('ALTER TABLE user ADD roles_id INT NULL');
26          $this->addSql('ALTER TABLE user ADD CONSTRAINT FK_8D93D64938C751C4 FOREIGN KEY (roles_id) REFERENCES roles (id)');
27          $this->addSql('CREATE INDEX IDX_8D93D64938C751C4 ON user (roles_id)');
28      }
29
30      public function down(Schema $schema): void
31      {
32          // this down() migration is auto-generated, please modify it to your needs
33          $this->addSql('CREATE TABLE role (id INT AUTO_INCREMENT NOT NULL, role VARCHAR(255) CHARACTER SET utf8mb4 NOT NULL COLLATE `utf8mb4_unicode_ci`, PRIMARY KEY(id))');
34          $this->addSql('ALTER TABLE roles DROP role');
35          $this->addSql('ALTER TABLE user DROP FOREIGN KEY FK_8D93D64938C751C4');
36          $this->addSql('DROP INDEX IDX_8D93D64938C751C4 ON user');
37          $this->addSql('ALTER TABLE user DROP roles_id');
38      }
39  }
40
```

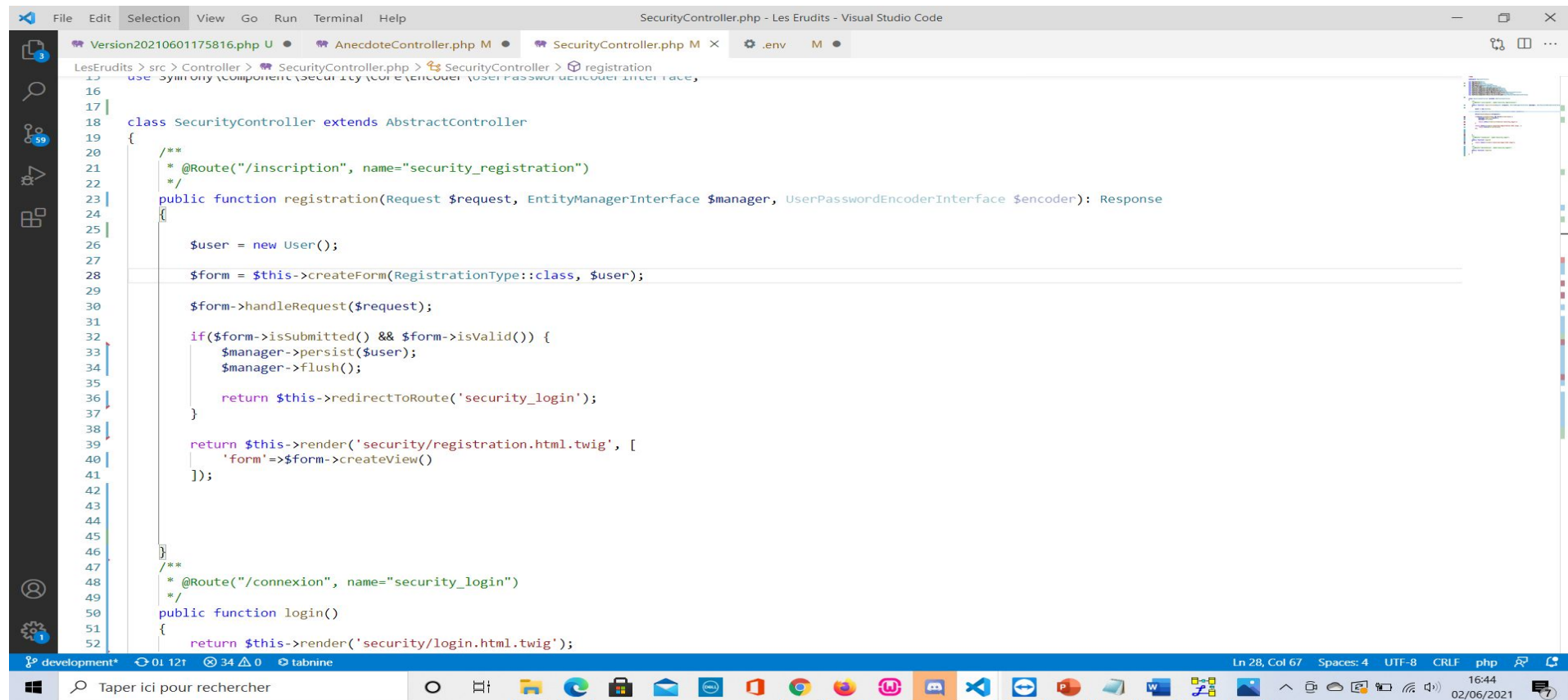
12) L'ARCHITECTURE MVC:

Modèle-vue-contrôleur ou MVC est un motif d'architecture logicielle (Design Pattern). Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

- Un modèle (Model) contient les données à afficher.
- Une vue (View) contient la présentation de l'interface graphique.
- Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.



13) EXTRAIT DE CODE BACK-END:



```
File Edit Selection View Go Run Terminal Help
SecurityController.php - Les Erudits - Visual Studio Code

Version20210601175816.php U • AnecdoteController.php M • SecurityController.php M X .env M •

LesErudits > src > Controller > SecurityController.php > SecurityController > registration
use Symfony\Component\Security\Core\Encoder\UserPasswordEncoderInterface;

16
17
18 class SecurityController extends AbstractController
19 {
20
21     /**
22      * @Route("/inscription", name="security_registration")
23      */
24     public function registration(Request $request, EntityManagerInterface $manager, UserPasswordEncoderInterface $encoder): Response
25     {
26         $user = new User();
27
28         $form = $this->createForm(RegistrationType::class, $user);
29
30         $form->handleRequest($request);
31
32         if($form->isSubmitted() && $form->isValid()) {
33             $manager->persist($user);
34             $manager->flush();
35
36             return $this->redirectToRoute('security_login');
37         }
38
39         return $this->render('security/registration.html.twig', [
40             'form'=>$form->createView()
41         ]);
42
43
44
45
46     }
47     /**
48      * @Route("/connexion", name="security_login")
49      */
50     public function login()
51     {
52         return $this->render('security/login.html.twig');
```

development* 01 12 34 0 tabnine Ln 28, Col 67 Spaces: 4 UTF-8 CRLF php 16:44 02/06/2021

Taper ici pour rechercher

RENDU AFFICHAGE:

The screenshot shows a web browser window with a dark theme. The browser's address bar displays the URL `127.0.0.1:8000/inscription`. The page has a purple header with navigation links: [Les Erudits](#), [Anecdotes](#), [Proposer une anecdote](#), [Connexion](#), and [Inscription](#). A search bar is located on the right side of the header. The main content area is dark purple and features the title

Inscription

 in a light blue font. Below the title are four input fields:
1. **Nom d'utilisateur**: A text input field with the placeholder text "Nom d'utilisateur...".
2. **Adresse email**: A text input field containing the email address "doutez3@gmail.com".
3. **Mot de Passe**: A password input field represented by a series of dots.
4. **Confirmation mot de passe**: A text input field with the placeholder text "Confirmez votre mot de passe".
At the bottom of the form is a green button labeled "Inscription". The browser's taskbar at the bottom shows several open applications, including a terminal window with the command `@ security_registration` and a status bar indicating the time as 10:03 on 03/06/2021.

14) EXTRAIT DE CODE FRONT-END:

```
10 | {% endblock %}
11 | </head>
12 | <body>
13 |   <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
14 |     <div class="container-fluid">
15 |       <a class="navbar-brand" href="{{ path ('home') }}">Les Erudits</a>
16 |       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarColor01" aria-controls="navbarColor01" aria-expanded="false" aria-label="
17 |         <span class="navbar-toggler-icon"></span>
18 |       </button>
19 |
20 |       <div class="collapse navbar-collapse" id="navbarColor01">
21 |         <ul class="navbar-nav me-auto">
22 |           <li class="nav-item">
23 |             <a class="nav-link active" href="{{ path ('anecdote') }}">Anecdotes
24 |               <span class="visually-hidden">(current)</span>
25 |             </a>
26 |           </li>
27 |           <li class="nav-item">
28 |             <a class="nav-link" href="{{ path ('anecdote_create') }}">Proposer une anecdote</a>
29 |           </li>
30 |           {% if not app.user %}
31 |           <li class="nav-item">
32 |             <a class="nav-link" href="{{ path ('security_login') }}">Connexion</a>
33 |           </li>
34 |
35 |           {% else %}
36 |           <li class="nav-item">
37 |             <a class="nav-link" href="{{ path ('security_logout') }}">Deconnexion</a>
38 |           </li>
39 |         {% endif %}
40 |
41 |       </ul>
42 |       <form class="d-flex">
43 |         <input class="form-control me-sm-2" type="text" placeholder="Search">
44 |         <button class="btn btn-secondary my-2 my-sm-0" type="submit">Search</button>
45 |       </form>
46 |     </div>
```


15) CONCLUSION:

Pour ce projet, je me suis basé sur les applications de blogs classiques et leur systèmes de fonctionnement. J'ai donc essayé de dégager l'essentiel de leurs fonctionnalités et choix adoptés, ce qui m'a permis de déterminer les grands axes de la conception de mon application.

Cette dernière permettra entre autres, de se cultiver à travers diverses anecdotes sur tout les sujets et de créer une communauté liée par la soif de de savoir.

Beati Paupers Spiritu (Bien heureux sont les pauvres d'esprits).

16) REMERCIEMENTS:

Je tiens à remercier nos formateurs de cette année qui nous enseignent les bases et fondements de la programmation, en particulier ESAT et MARC.

Aussi DANY et MARIE pour leurs aides et le personnel du Greta de Noisy le Grand pour nous avoir permis de continuer à suivre les cours sur place malgré le contexte.