

Réalisation professionnel 1^{er} année BTS SIO SLAM 2022-2023



Table des matières

Résumé du contexte.....	3
La mission globale à effectuer.....	4
Étape 1 préparer l'environnement de travail et créer la base de données :.....	4
Générer le script SQL correspondant, au format MySQL.....	4
Créer la base de données correspondante sous MySQL ou MariaDB.....	6
Créer un utilisateur (login/pwd) ayant les droits d'accès à cette base de données.....	8
Ajouter une table responsable qui contient 2 champs : 'login' et 'pwd' de type varchar longueur 64	10
Alimenter la base de données :.....	11
Étape 2 : dessiner les interfaces, structurer l'application en MVC, créer un dépôt, coder le visuel...15	
Utiliser un outil de maquettage pour dessiner les interfaces répondant aux besoins.....	15
Créer une nouvelle application sous l'IDE choisi.....	15
Créer les 3 packages/bibliothèques/dossiers correspondant au MVC.....	15
Créer un dépôt distant (sur GitHub) et faire une première sauvegarde.....	16
Dans votre compte GitHub, créer un "project" (kanban) et y insérer toutes les issues correspondant aux tâches à réaliser : à chaque fois qu'une tâche est réalisée, il faudra changer son état.....	17
Coder la partie Vue de l'application (uniquement le visuel des interfaces).....	17
Sauvegarder le travail sur le dépôt distant avec un commentaire explicite et mettre à jour le kanban.....	18
Étape 3 : coder le modèle et les outils de connexion, générer la documentation technique.....	19
Si nécessaire, configurer l'IDE pour accéder à la base de données.....	19
Créations de tous les packages nécessaire à l'application.....	19
Étape 4 : coder les fonctionnalités de l'application à partir des cas d'utilisation.....	20
Explications générale du code.....	20
Étape 5 : créer une documentation utilisateur en vidéo.....	21
Étape 6 : gérer le déploiement.....	21

Résumé du contexte



InfoTech Services 86

InfoTech Services 86 (ITS 86), est une Entreprise de Services Numériques (ESN) spécialisée dans le développement informatique (applications de bureau, web, mobile), l'hébergement de site web, l'infogérance, la gestion de parc informatique et l'ingénierie système et réseau. Elle répond régulièrement à des appels d'offres en tant que société d'infogérance et prestataire de services informatiques.

Parmi les marchés gagnés récemment par ITS 86, on trouve celui de la gestion du parc informatique du réseau des médiathèques de la Vienne, MediaTek86, ainsi que l'informatisation de plusieurs activités internes des médiathèques ou en lien avec le public.

La mission globale à effectuer

La mission consiste en la création d'une solution logiciel pour MediaTek86.
Pour se faire nous disposons d'un schéma conceptuel de donnée et d'un dossier documentaire contenant un diagramme de cas d'utilisation.

Détail de l'intégralité des étapes demandées

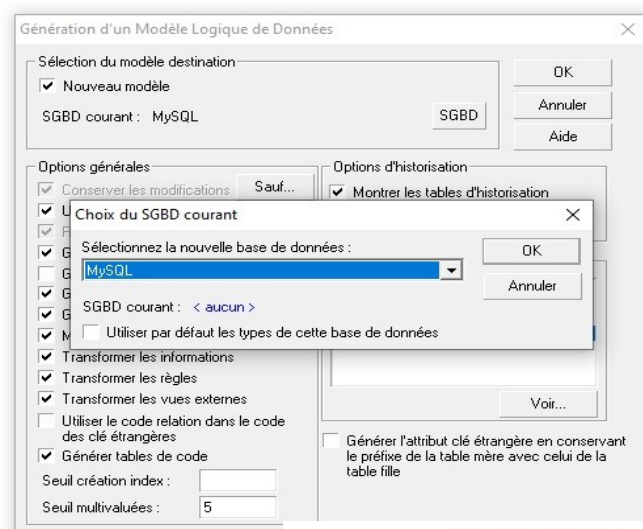
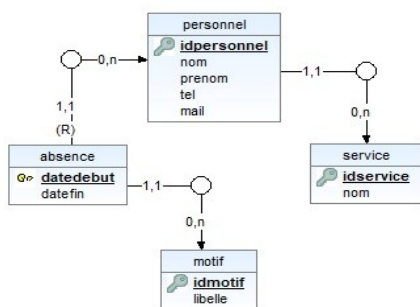
Étape 1 préparer l'environnement de travail et créer la base de données :

Après avoir installé les divers solutions logicielles requis (wampserver, Visual Studio etc...) Je récupère le schéma conceptuel de données.

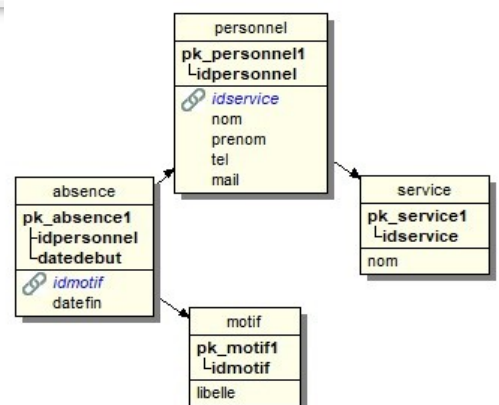
Générer le script SQL correspondant, au format MySQL

Pour se faire sous WinDesign il faut depuis le schéma conceptuel de données créer un schéma logique de données.

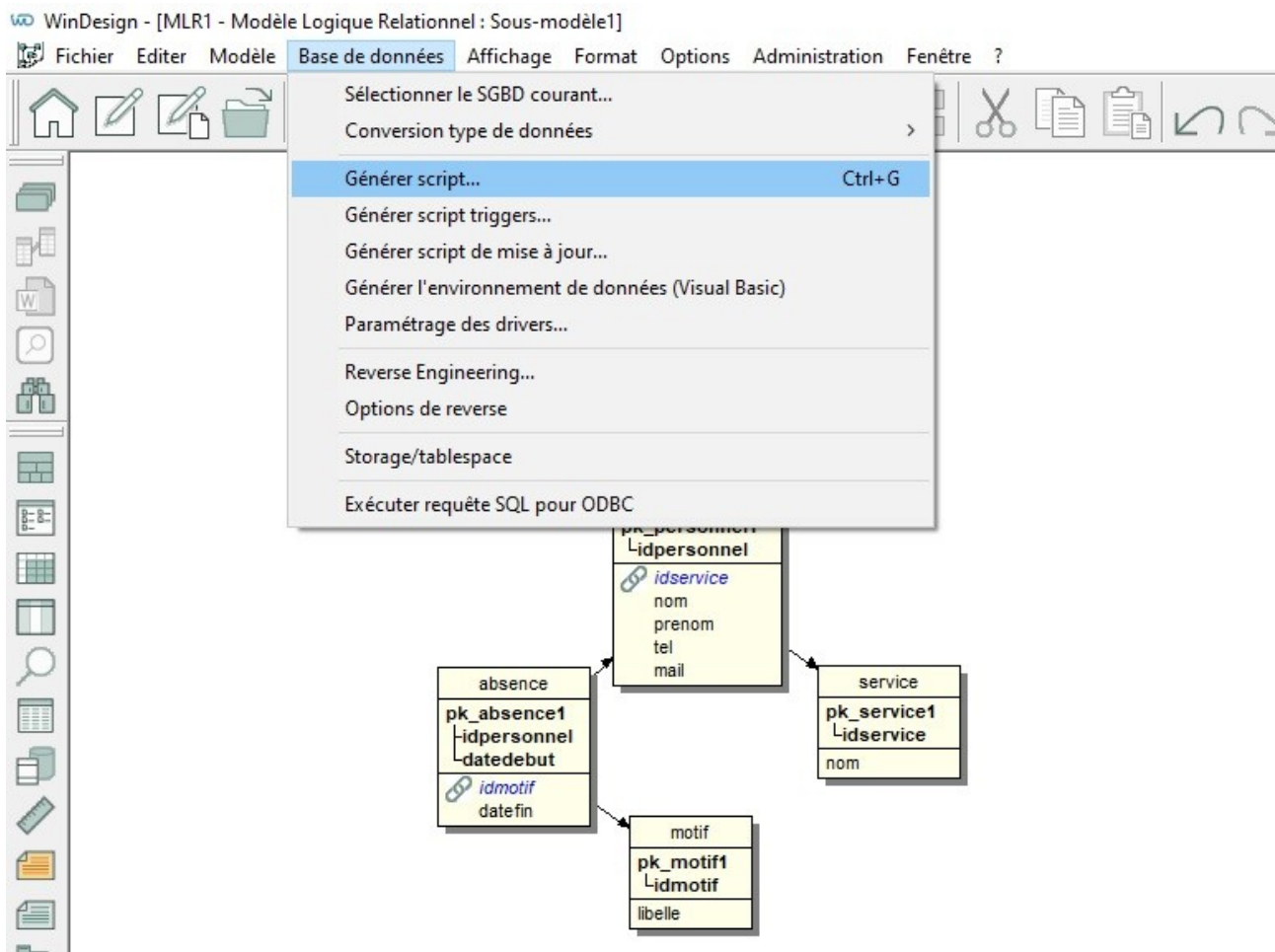
Donc depuis WinDesign en choisissant bien le format MySQL cela donne en images :



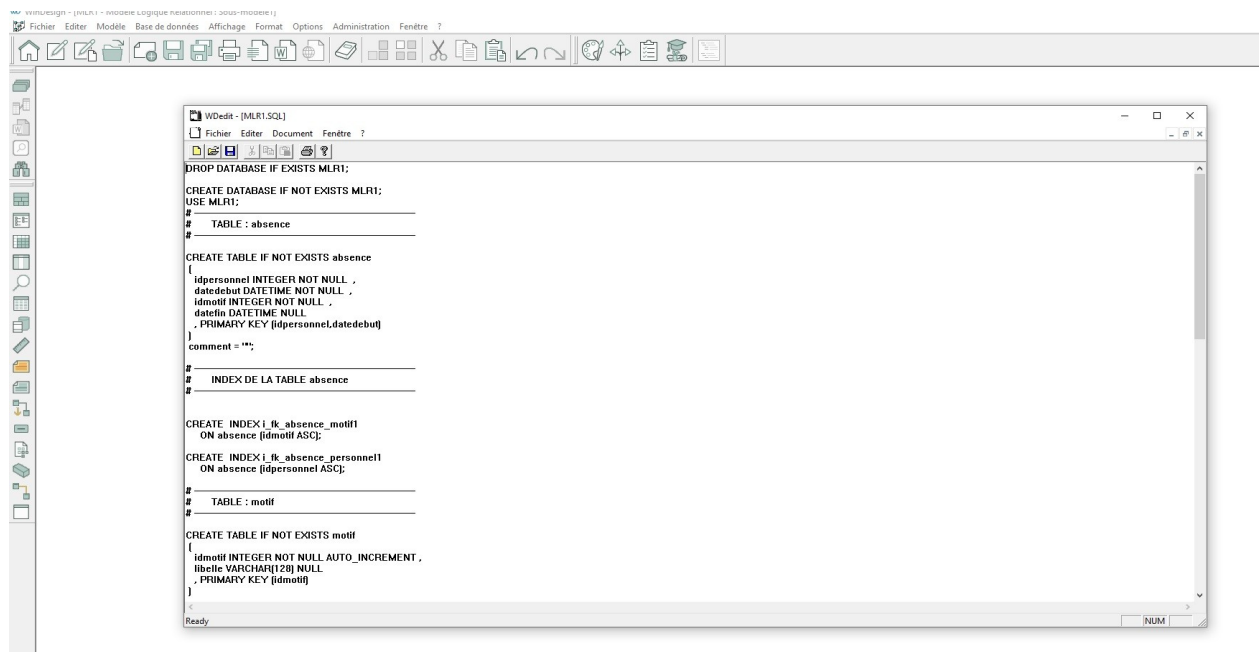
Ce qui donne ensuite le schéma logique de données suivant :



Et ensuite il devient possible de générer le scripte SQL depuis le schéma logique de données de la manière suivante :



Et on obtient donc le script SQL au format MySQL comme demandé dans l'énoncé.



Créer la base de données correspondante sous MySQL ou MariaDB.

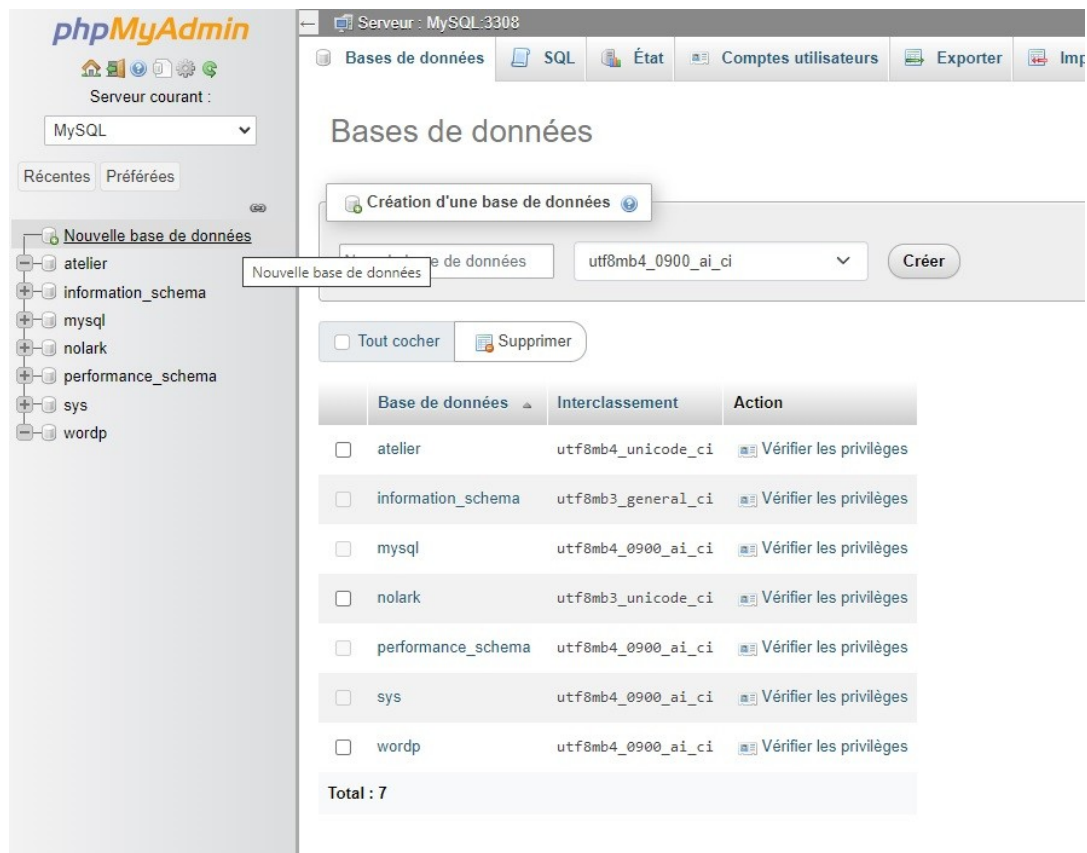
Il faut lancer phpMyAdmin depuis wampserver



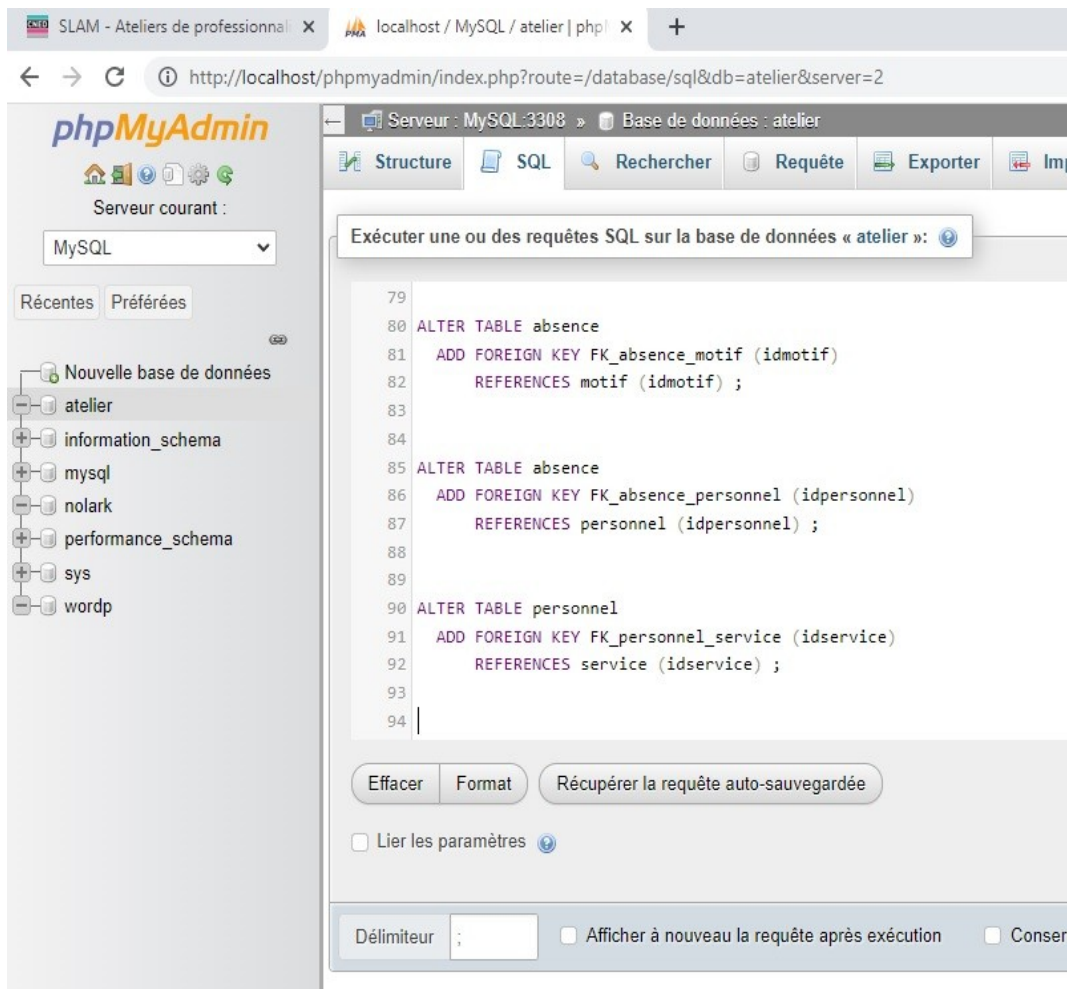
Depuis la page de connexion , par défaut le nom d'utilisateur est root sans mot de passe

A screenshot of the phpMyAdmin login page. The page has the phpMyAdmin logo and the text 'Bienvenue dans phpMyAdmin'. There is a 'Langue (Language)' dropdown menu set to 'Français - French'. Below it is a 'Connexion' section with a blue circular icon. It contains three input fields: 'Utilisateur :' with 'root' entered, 'Mot de passe :', and 'Choix du serveur :' with 'MySQL' selected in a dropdown. A 'Connexion' button is at the bottom right.

Ensuite on peut administrer les bases de données et ici il suffit de créer une nouvelle base de données que je nomme atelier.



Depuis l'onglet SQL de la base de données atelier il suffit de copier coller le fichier SQL qui a été généré précédemment mais pas les trois premières lignes.



Créer un utilisateur (login/pwd) ayant les droits d'accès à cette base de données.

Il suffit de retourner sur la page d'accueil puis l'onglet comptes utilisateur et de cliquer sur ajouter un compte utilisateur.

phpMyAdmin interface showing the 'Comptes utilisateurs' (Users) tab. The 'root' user is highlighted with a red arrow. Below the table, the 'Ajouter un compte d'utilisateur' (Add user) link is visible.

Nom d'utilisateur	Nom d'hôte	Mot de passe	Privileges globaux	« Grant »	Action
<input type="checkbox"/> comptewp	%	Oui	USAGE	Non	Éditer les privilèges Exporter Verrouiller
<input type="checkbox"/> mysql.infoschema	localhost	Oui	SELECT	Non	Éditer les privilèges Exporter Déverrouiller
<input type="checkbox"/> mysql.session	localhost	Oui	SHUTDOWN, SUPER	Non	Éditer les privilèges Exporter Déverrouiller
<input type="checkbox"/> mysql.sys	localhost	Oui	USAGE	Non	Éditer les privilèges Exporter Déverrouiller
<input type="checkbox"/> nolarkuser	localhost	Oui	USAGE	Non	Éditer les privilèges Exporter Verrouiller
<input type="checkbox"/> root	localhost	Non	ALL PRIVILEGES	Oui	Éditer les privilèges Exporter Verrouiller

↑ ☐ Tout cocher Avec la sélection : Exporter

Nouvel utilisateur

[Ajouter un compte d'utilisateur](#)

Ensuite on saisie un Nom d'utilisateur ici avasara puis sur générer un mot de passe pour ne pas avoir de problème de sécurité et combler mon manque d'imagination chronique.

Ajouter un compte d'utilisateur

Informations pour la connexion

Nom d'utilisateur : Saisir une valeur ▼ avasara ←

Nom d'hôte : Tout hôte ▼ % ⓘ

Mot de passe : Saisir une valeur ▼ Force : Fort

Saisir à nouveau :

Extension d'authentification : Mise en cache de l'authentification sha2 ▼

Générer un mot de passe : Générer ← -E2(Zly8TrtK8lfy

Ensuite depuis l'onglet base de données de l'utilisateur créé on sélectionne la base de données atelier pour pouvoir administrer les droits que l'on peut donner à l'utilisateur.

✓ L'utilisateur a ajouté un utilisateur.

```
CREATE USER 'avasarala'@'%' IDENTIFIED WITH caching_sha2_password BY '****';GRANT USAGE ON *.* TO 'avasarala'@'%;ALTER USER 'avasarala'@'%' REQUIRE NONE WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

Global Base de données Change password Informations pour la connexion

Éditer les privilèges : Compte d'utilisateur 'avasarala'@'%'

Privilèges spécifiques à une base de données

Base de données	Privilèges	« Grant »	Privilèges spécifiques à une table	Action
Aucun(e)				

Ajouter des privilèges sur ces bases de données :

- atelier
- mysql
- nolark
- sys

Exécuter

Et depuis cette page je choisis dans un premier temps de donner tout les droits à cet utilisateur car il est demandé de créer un utilisateur ayant les droits « d'accès à cette base de données » mais comme je ne sais pas pour quel usage dans un premier temps je fais comme cela, je pourrai toujours changer les droits après.

Éditer les privilèges : Compte d'utilisateur 'avasarala'@'%'

Base de données atelier

Privilèges spécifiques à une base de données ☒ Tout cocher

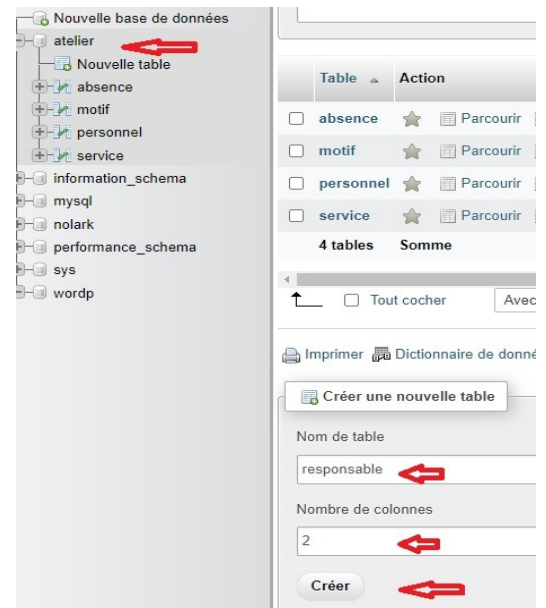
IB : les noms de privilèges sont exprimés en anglais.

Données	Structure	Administration
<input checked="" type="checkbox"/> SELECT	<input checked="" type="checkbox"/> CREATE	<input checked="" type="checkbox"/> GRANT
<input checked="" type="checkbox"/> INSERT	<input checked="" type="checkbox"/> ALTER	<input checked="" type="checkbox"/> LOCK TABLES
<input checked="" type="checkbox"/> UPDATE	<input checked="" type="checkbox"/> INDEX	<input checked="" type="checkbox"/> REFERENCES
<input checked="" type="checkbox"/> DELETE	<input checked="" type="checkbox"/> DROP	
	<input checked="" type="checkbox"/> CREATE TEMPORARY TABLES	
	<input checked="" type="checkbox"/> SHOW VIEW	
	<input checked="" type="checkbox"/> CREATE ROUTINE	
	<input checked="" type="checkbox"/> ALTER ROUTINE	
	<input checked="" type="checkbox"/> EXECUTE	
	<input checked="" type="checkbox"/> CREATE VIEW	
	<input checked="" type="checkbox"/> EVENT	
	<input checked="" type="checkbox"/> TRIGGER	

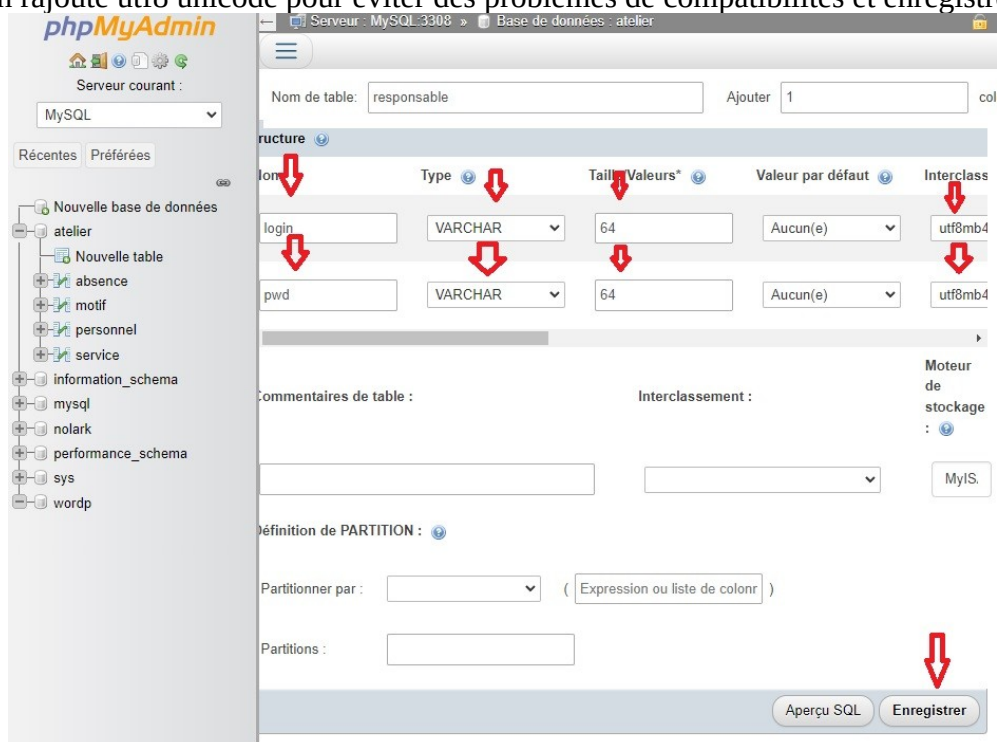
Exécuter

Ajouter une table responsable qui contient 2 champs : 'login' et 'pwd' de type varchar longueur 64

Depuis la base de données atelier on sélectionne créer une nouvelle table qui se nomme responsable et qui contient deux champs comme sur la capture suivante.



Puis dans la page suivante on précise bien login et pwd comme champ, le tout en type varchar de taille 64, on rajoute utf8 unicode pour éviter des problèmes de compatibilités et enregistrer.



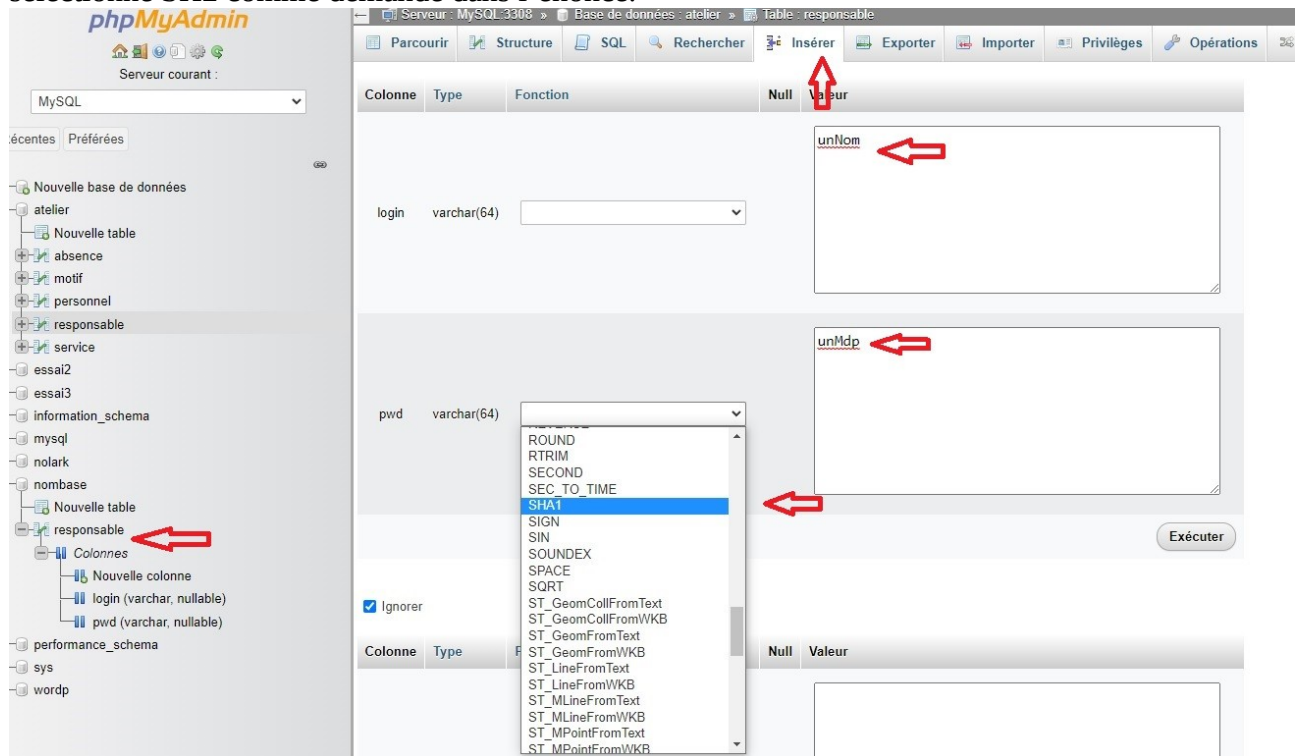
Ou plus simplement en SQL , on se positionne sur la bdd concerné puis dans l'onglet SQL on écrit :

```
CREATE TABLE responsable ( login varchar(64), pwd varchar(64));
```

Alimenter la base de données :

La table 'responsable' doit être remplie avec un login et un pwd chiffré de votre choix : pour le hashage, utiliser la fonction SHA2 : SHA2("le pwd", 256).

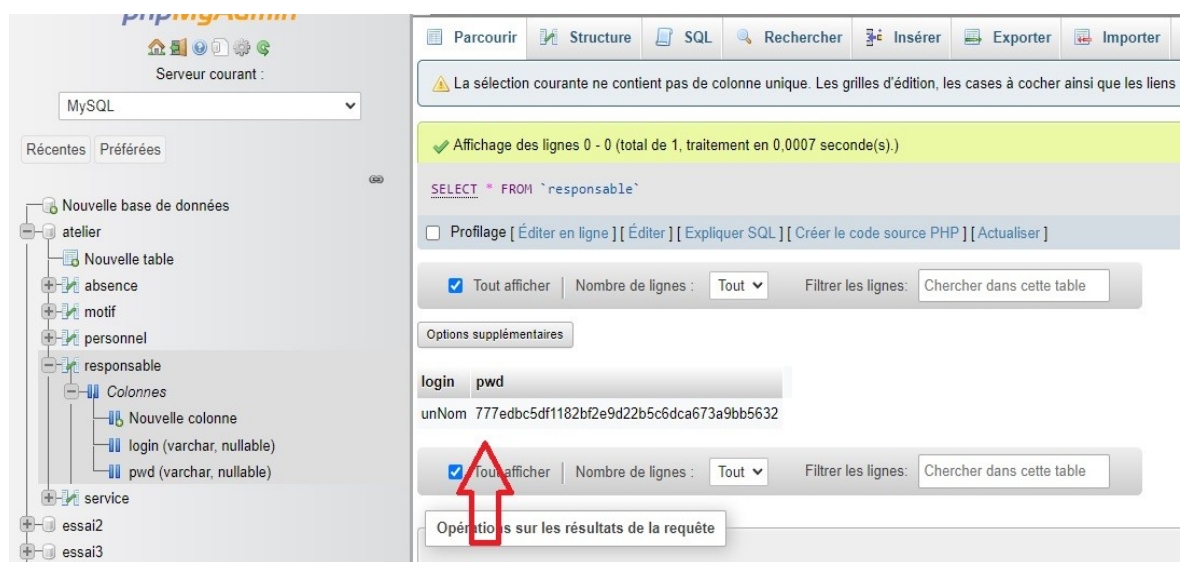
Et donc pour se faire on se positionne sur la table responsable de la base de donnée atelier. Puis on sélectionne Colonnes et l'onglet Insérer et l'on donne une valeur à login et pwd et pour pwd on sélectionne SH2 comme demandé dans l'énoncé.



phpMyAdmin nous limite au SHA1 donc toujours sur la même bdd on sélectionne l'onglet SQL et on écrit :

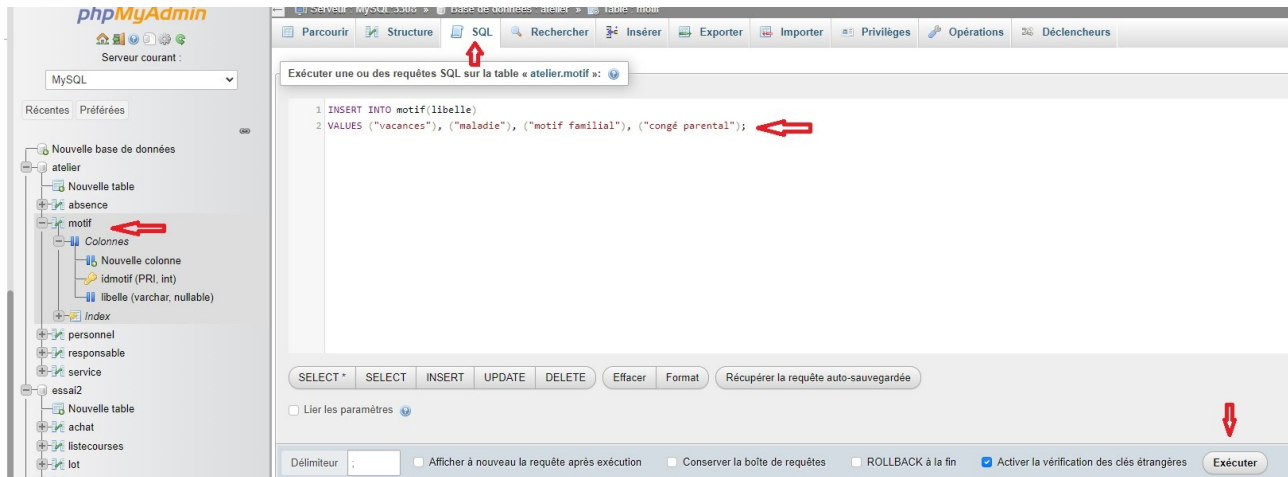
```
INSERT INTO responsable (login, pwd) VALUES ('unNom', SHA2('unMdp',256));
```

Ensuite on parcourt la table pour s'assurer que le mot de passe soit bien haché :



La table 'motif' doit être remplie avec les motifs suivants : vacances, maladie, motif familial, congé parental. Le contenu de cette table n'est pas modifié dans l'application.

Pour se faire il suffit de se positionner sur la bonne table et dans l'onglet SQL écrire :
INSERT INTO motif(libelle)
VALUES ("vacances"), ("maladie"), ("motif familial"), ("congé parental");



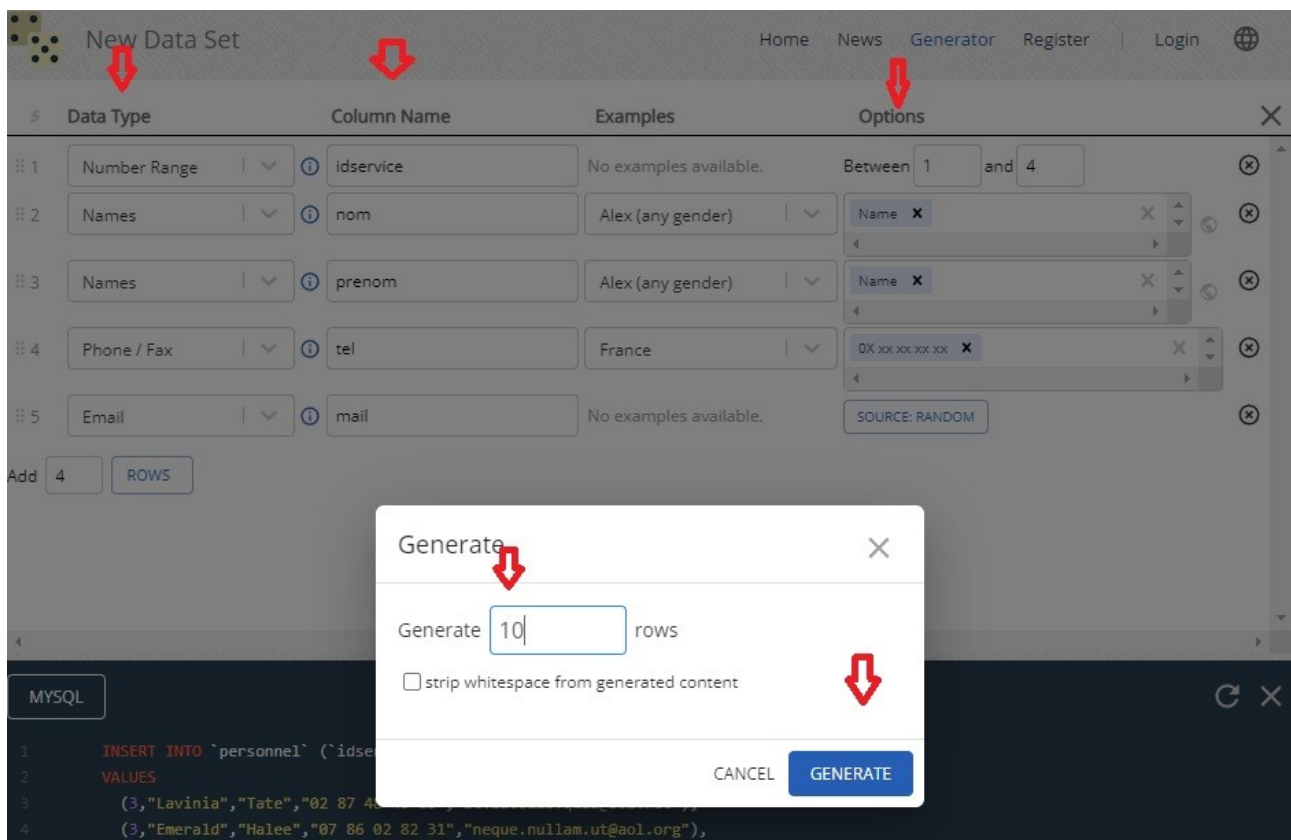
La table 'service' doit être remplie avec les services présentés dans le contexte (administratif, médiation culturelle, prêt). Le contenu de cette table n'est pas modifié dans l'application.

Même principe que précédemment, on change juste la table concernée et les valeurs entrées ce qui donne :

```
INSERT INTO service (nom)
VALUES ("administratif"), ("médiation"), ("culturelle"), ("prêt");
```

Les tables 'personnel' et 'absence' doivent être remplies d'exemples aléatoires pour réaliser des tests (utilisez un site de génération automatique d'insert, comme <https://www.generatedata.com/>). Mettez une dizaine d'exemples dans 'personnel' et une cinquantaine dans 'absence'.

Comme les consignes l'exigent, je me rends sur [generatedata.com](https://www.generatedata.com/) et je configure les paramètres pour les adapter à ma base de donnée et ma table. Les bons noms de bdd et champs ainsi que les bonnes informations, pour l'id service, je comprend de la question précédente qu'il a quatre services. Puis on intègre le fichier SQL ainsi généré dans la base de donnée, soit via l'onglet importer dans phpMyadmin soit en copiant collant le contenu du fichier dans l'onglet SQL.



Et on parcourt la table pour s'assurer que l'on a bien ce que l'on attend

Options supplémentaires

	idpersonnel	idservice	nom	prenom	tel	mail
<input type="checkbox"/> Éditer Copier Supprimer	1	3	Harding	Angelica	03 41 40 76 23	at.velit.pellentesque@protonmail.com
<input type="checkbox"/> Éditer Copier Supprimer	2	4	Eve	Hop	02 51 78 38 56	rhoncus@protonmail.edu
<input type="checkbox"/> Éditer Copier Supprimer	3	2	Harding	Cecilia	05 45 12 97 27	sociis.natoque.penatibus@protonmail.net
<input type="checkbox"/> Éditer Copier Supprimer	4	1	Brandon	Jordan	01 69 52 18 21	dignissim@hotmail.org
<input type="checkbox"/> Éditer Copier Supprimer	5	3	Harding	Hall	03 24 73 59 92	amet@yahoo.edu
<input type="checkbox"/> Éditer Copier Supprimer	6	2	Kyla	Hope	08 27 52 46 41	aliquet.magna@aol.ca
<input type="checkbox"/> Éditer Copier Supprimer	7	3	Tara	Vera	02 12 53 12 90	ut.dolor.dapibus@yahoo.edu
<input type="checkbox"/> Éditer Copier Supprimer	8	3	Todd	Wang	07 82 50 46 13	mus@aol.org
<input type="checkbox"/> Éditer Copier Supprimer	9	3	Buckminster	Lyle	03 13 48 70 14	arcu.morbi.sit@outlook.com
<input type="checkbox"/> Éditer Copier Supprimer	10	4	Alfonso	Amber	06 55 98 89 21	dolor.fusce@google.couk

(Petite anecdote, les personnel id 1,3 et 6 portent le même nom. Cela m'a causé beaucoup d'incompréhension quand je programmais, je n'avais pas fait attentions à ce détail et ne comprenais pas pourquoi je n'avais pas les bons personnels avant de me rendre compte...;)

C'est le même procédé pour la table absence, la seule différence est qu'il faut gérer une date de début et une date de fin et m'assurer qu'aucun personnel n'a commencé une absence avant d'avoir fini une autre absence pour que cela soit cohérent ce qui va nécessiter quelques corrections manuelles. Le plus simple que j'ai pu faire sous generatedata est de générer 5 fois 10 absence avec des mois décalés et de corriger les dates si besoin à chaque fois. Ce qui me donne au final :

← Serveur : MySQL:3308 » Base de données : atelier » Table : absence

Parcourir Structure SQL Rechercher Insérer Exporter Importer

✓ Affichage des lignes 0 - 49 (total de 50, traitement en 0,0006 seconde(s).)

`SELECT * FROM `absence``

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 100 ▼ Filtrer les lignes: Chercher dans cette table Trier par cl

Options supplémentaires

← T →	idpersonnel	datedebut	idmotif	datefin
<input type="checkbox"/> Éditer Copier Supprimer	1	2022-06-27 03:55:20	3	2022-07-25 04:05:08
<input type="checkbox"/> Éditer Copier Supprimer	6	2022-06-07 00:23:48	4	2022-07-27 22:43:03
<input type="checkbox"/> Éditer Copier Supprimer	10	2022-06-15 08:18:00	3	2022-07-23 20:39:32
<input type="checkbox"/> Éditer Copier Supprimer	5	2022-06-18 15:13:27	3	2022-07-23 06:44:11
<input type="checkbox"/> Éditer Copier Supprimer	3	2022-05-03 03:25:26	2	2022-06-18 06:20:50
<input type="checkbox"/> Éditer Copier Supprimer	10	2022-05-05 08:18:42	1	2022-06-06 10:32:54
<input type="checkbox"/> Éditer Copier Supprimer	1	2022-05-17 03:38:39	3	2022-06-19 10:44:25
<input type="checkbox"/> Éditer Copier Supprimer	5	2022-05-04 11:03:39	1	2022-06-09 18:33:20
<input type="checkbox"/> Éditer Copier Supprimer	4	2022-05-24 16:29:53	1	2022-06-12 04:48:30
<input type="checkbox"/> Éditer Copier Supprimer	2	2022-05-30 15:56:55	3	2022-06-11 04:16:49
<input type="checkbox"/> Éditer Copier Supprimer	6	2022-05-24 08:02:20	2	2022-06-10 11:48:26
<input type="checkbox"/> Éditer Copier Supprimer	7	2022-05-12 04:32:13	2	2022-06-27 22:22:39
<input type="checkbox"/> Éditer Copier Supprimer	8	2022-05-02 03:53:17	1	2022-06-10 03:07:30
<input type="checkbox"/> Éditer Copier Supprimer	9	2022-05-15 06:38:22	3	2022-06-14 18:28:38
<input type="checkbox"/> Éditer Copier Supprimer	8	2022-04-12 03:00:03	1	2022-05-09 14:08:35
<input type="checkbox"/> Éditer Copier Supprimer	5	2022-04-28 20:02:22	3	2022-05-01 23:02:19
<input type="checkbox"/> Éditer Copier Supprimer	10	2022-04-11 15:15:23	3	2022-05-06 01:05:52
<input type="checkbox"/> Éditer Copier Supprimer	9	2022-04-21 05:37:52	3	2022-05-28 16:45:53

☐ Copier le code source SQL

Étape 2 : dessiner les interfaces, structurer l'application en MVC, créer un dépôt, coder le visuel

Utiliser un outil de maquettage pour dessiner les interfaces répondant aux besoins.

Pour se faire j'utilise Pencil et avec pour leitmotive, une application rustique et ludique ! Donc je conçois trois pages, une page de connexion, une page pour la gestion du personnel et une page pour la gestion des absences,

(toute ressemblance avec la page de connexion du cned est purement fortuite)

Media Tek86 gestion du personnel

* Champ obligatoire

Nom d'utilisateur *

Saisissez votre nom d'utilisateur

Mot de passe *

Saisissez votre mot de passe

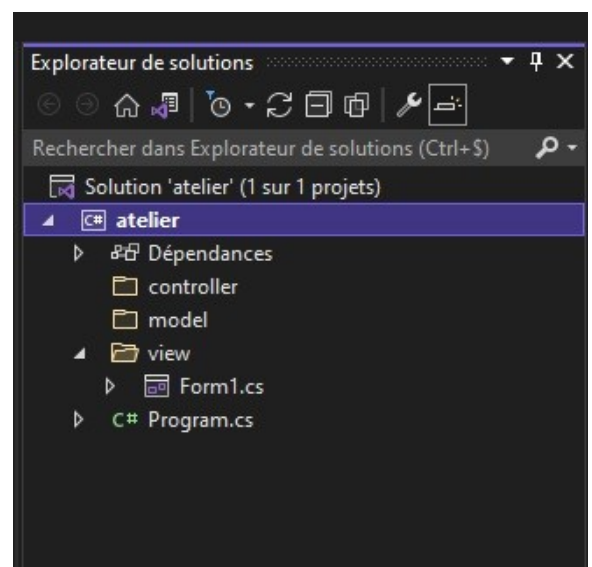
Se connecter

Créer une nouvelle application sous l'IDE choisi.

Je choisis de faire l'application en C# sous Visual Studio

Créer les 3 packages/bibliothèques/dossiers correspondant au MVC.

Je crée mes dossiers model, view et controller (explication du code plus loin)



Créer un dépôt distant (sur GitHub) et faire une première sauvegarde.

Je crée un dépôt que je nomme atelier sur mon compte gitHub

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *  hedi-k / Repository name * atelier

✓ atelier is available.

Great repository names are short and memorable. Need inspiration? How about [cuddly-octo-tribble](#)?

Description (optional)

SLAM - Ateliers de professionnalisation – 1re année

- ☐  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☒  **Private**
You choose who can see and commit to this repository.

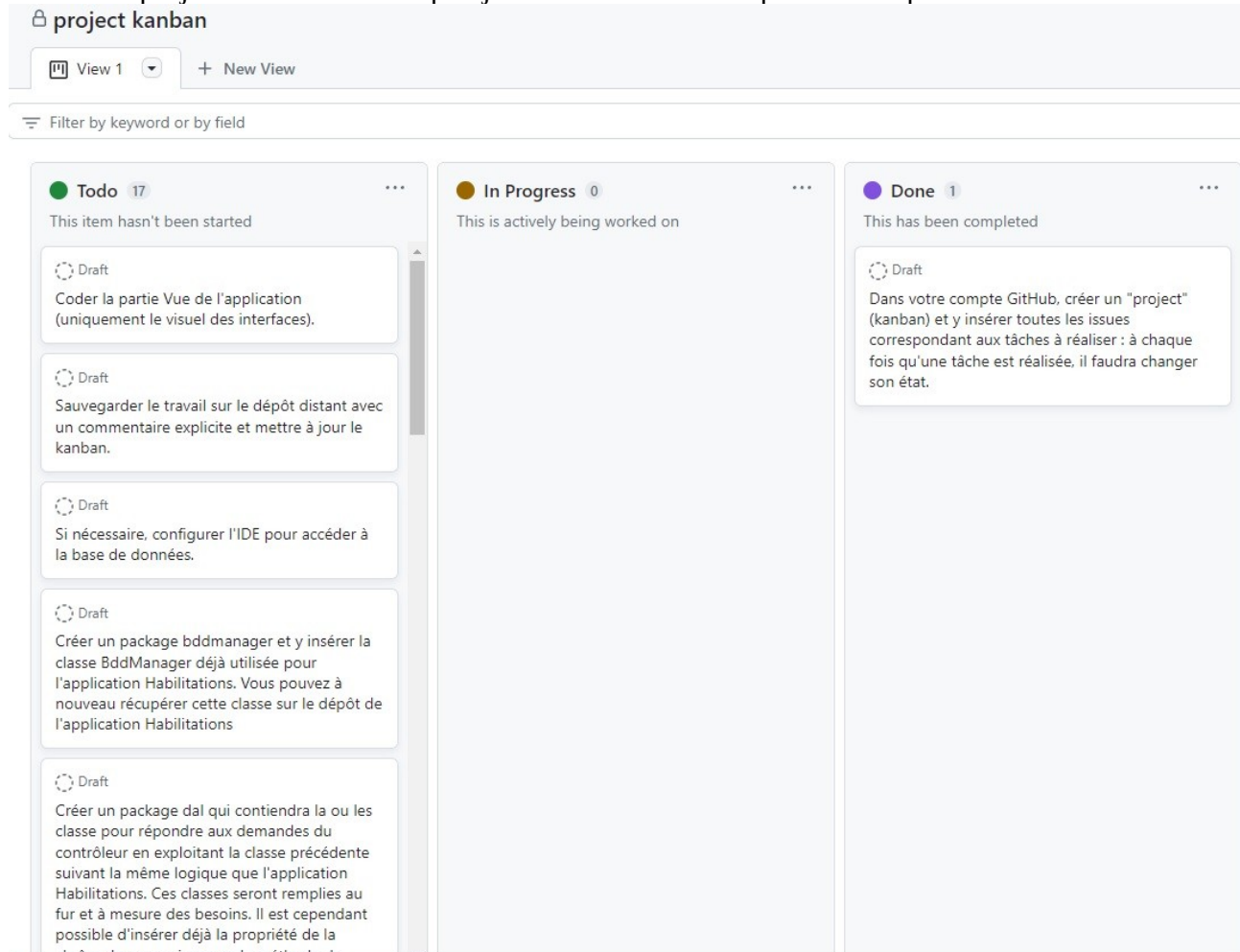
Et une fois le dépôt créé, il suffit de se connecter à son gitHub via son ide et de « pousser » son travail sur le dépôt de son git.

```
PowerShell développeur
+ PowerShell développeur |  |  | 
PS C:\Users\960m\Desktop\test\programmes c#\atelier> git remote add origin https://github.com/hedi-k/atelier.git
error: remote origin already exists.
PS C:\Users\960m\Desktop\test\programmes c#\atelier> git branch -M main
PS C:\Users\960m\Desktop\test\programmes c#\atelier> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 41, done.
Counting objects: 100% (41/41), done.
Delta compression using up to 4 threads
Compressing objects: 100% (30/30), done.
Writing objects: 100% (41/41), 86.81 KiB | 3.21 MiB/s, done.
Total 41 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/hedi-k/atelier.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\960m\Desktop\test\programmes c#\atelier>
```

(Je n'ai pas paramétré mon ide avec le même compte github donc tout les commit portent un nom différent qui correspond à un compte essai de quand je découvrais github pendant les cours.)

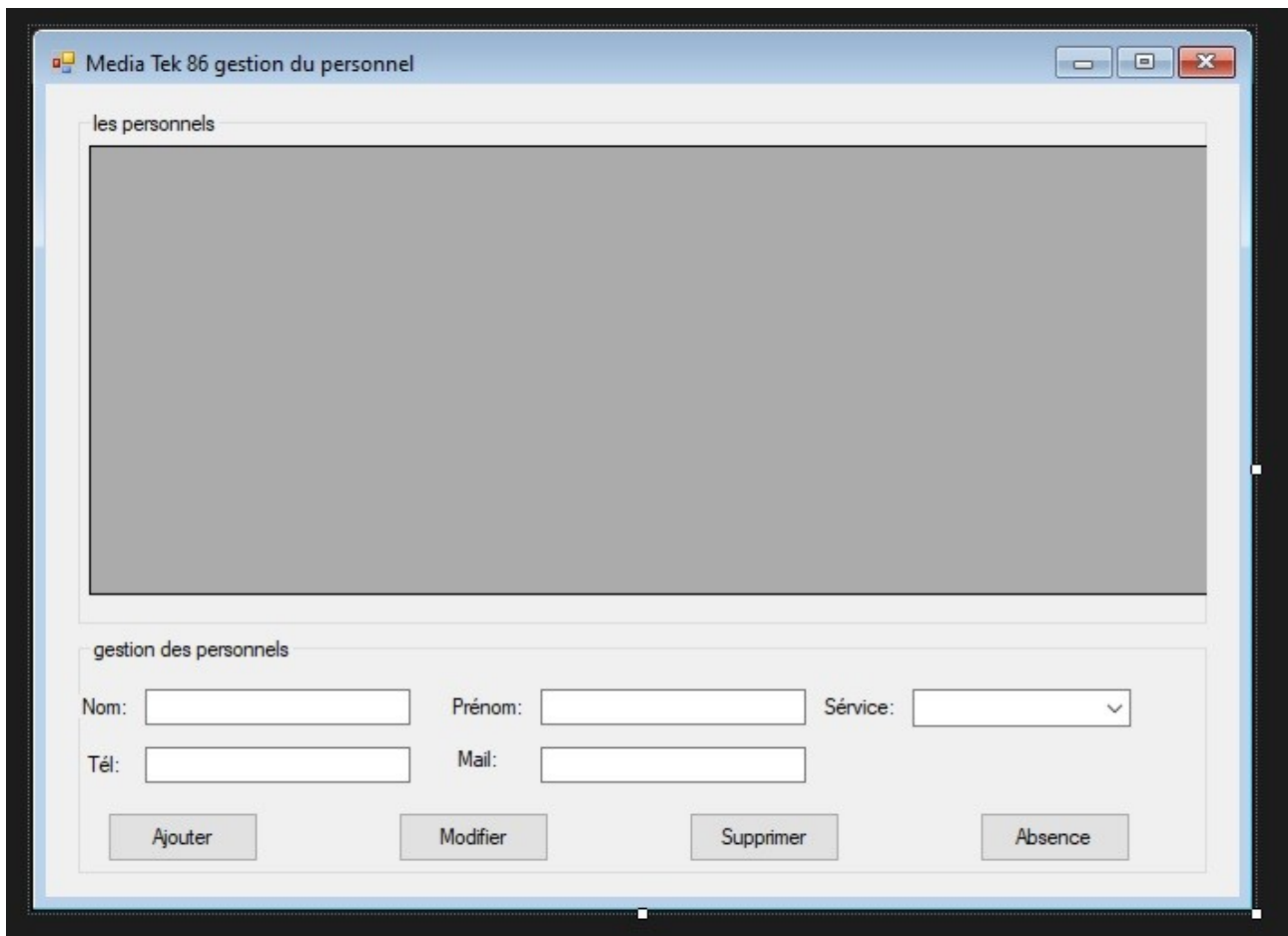
Dans votre compte GitHub, créer un "project" (kanban) et y insérer toutes les issues correspondant aux tâches à réaliser : à chaque fois qu'une tâche est réalisée, il faudra changer son état.

Je crée un project kanban dans le quel je mets tous les descriptifs des étapes en todo.



Coder la partie Vue de l'application (uniquement le visuel des interfaces).

Je code donc le visuel de l'application à l'image de ma maquette avec quelques modifications mineur :



Sauvegarder le travail sur le dépôt distant avec un commentaire explicite et mettre à jour le kanban.

```
no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\960m\Desktop\test\programmes c#\atelier> git add .
PS C:\Users\960m\Desktop\test\programmes c#\atelier> git commit -m "Visuel de l'application terminé"
[main 4fadcf] Visuel de l'application terminé
create mode 100644 obj/Debug/net6.0-windows/atelier.FrmConnexion.resources
create mode 100644 obj/Debug/net6.0-windows/atelier.csproj.GenerateResource.cache
create mode 100644 obj/Debug/net6.0-windows/atelier.designer.deps.json
create mode 100644 obj/Debug/net6.0-windows/atelier.designer.runtimeconfig.json
create mode 100644 obj/Debug/net6.0-windows/atelier.view.Form2.resources
create mode 100644 view/Form1.resx
create mode 100644 view/Form2.Designer.cs
create mode 100644 view/Form2.cs
create mode 100644 view/Form2.resx
create mode 100644 view/Form3.Designer.cs
create mode 100644 view/Form3.cs
create mode 100644 view/Form3.resx
PS C:\Users\960m\Desktop\test\programmes c#\atelier> git push
Enumerating objects: 53, done.
Counting objects: 100% (53/53), done.
Delta compression using up to 4 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (32/32), 14.62 KiB | 1.12 MiB/s, done.
Total 32 (delta 11), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (11/11), completed with 8 local objects.
To https://github.com/hedi-k/atelier.git
 2a6cfa6..4fadcf main -> main
PS C:\Users\960m\Desktop\test\programmes c#\atelier> 
```

Pour sauvegarder le travail, un CTRL+S pour l'enregistrer sur le pc.
Puis avec la console dans l'IDE, j'utilise les commandes suivantes :

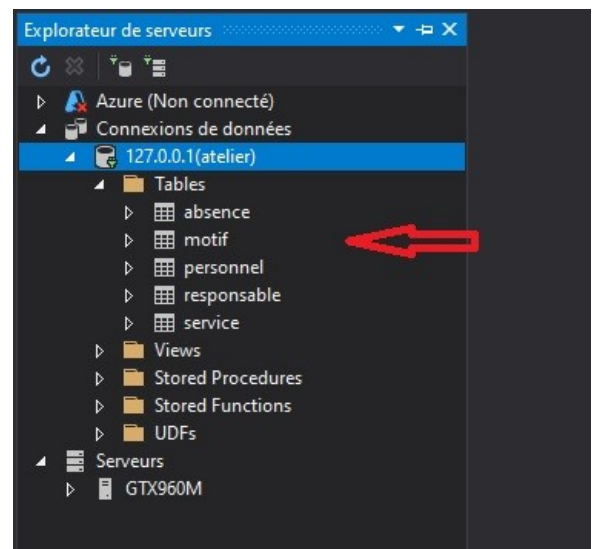
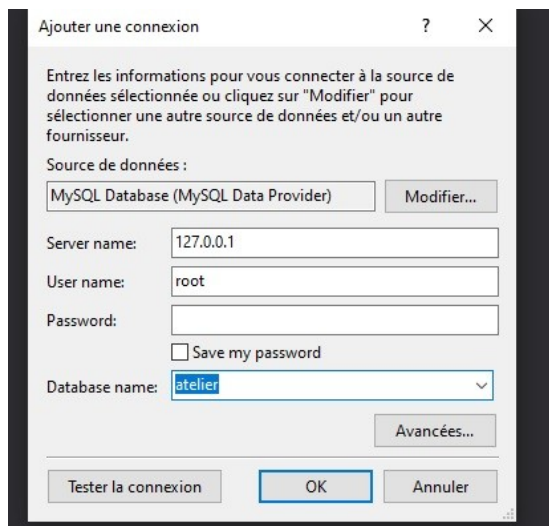
git add . Pour ajouter toutes les modifications (le point veut dire tout)
git commit -m + commentaire Pour sauvegarder réellement les modification que j'ai « git add »
Et pour finir git push pour envoyer le commit sur mon dépôt github en ligne.

Et sur le kanban je glisse la vignette de in progress à done et je passe à la suivante.

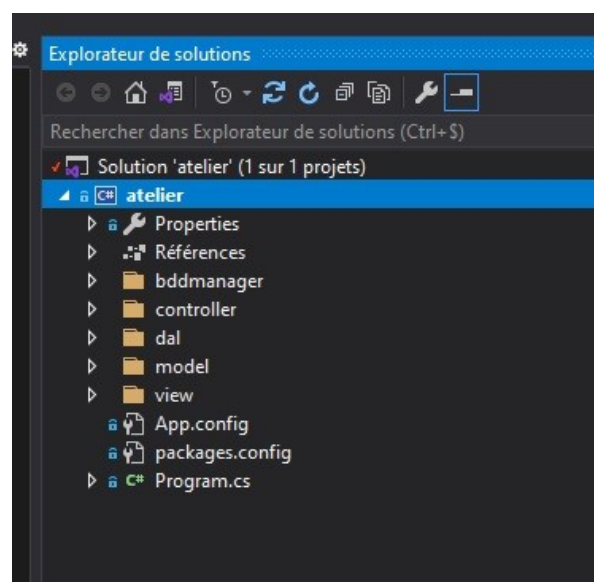
Étape 3 : coder le modèle et les outils de connexion, générer la documentation technique

Si nécessaire, configurer l'IDE pour accéder à la base de données.

Après paramètre de l'ide on à accès à la bdd atelier

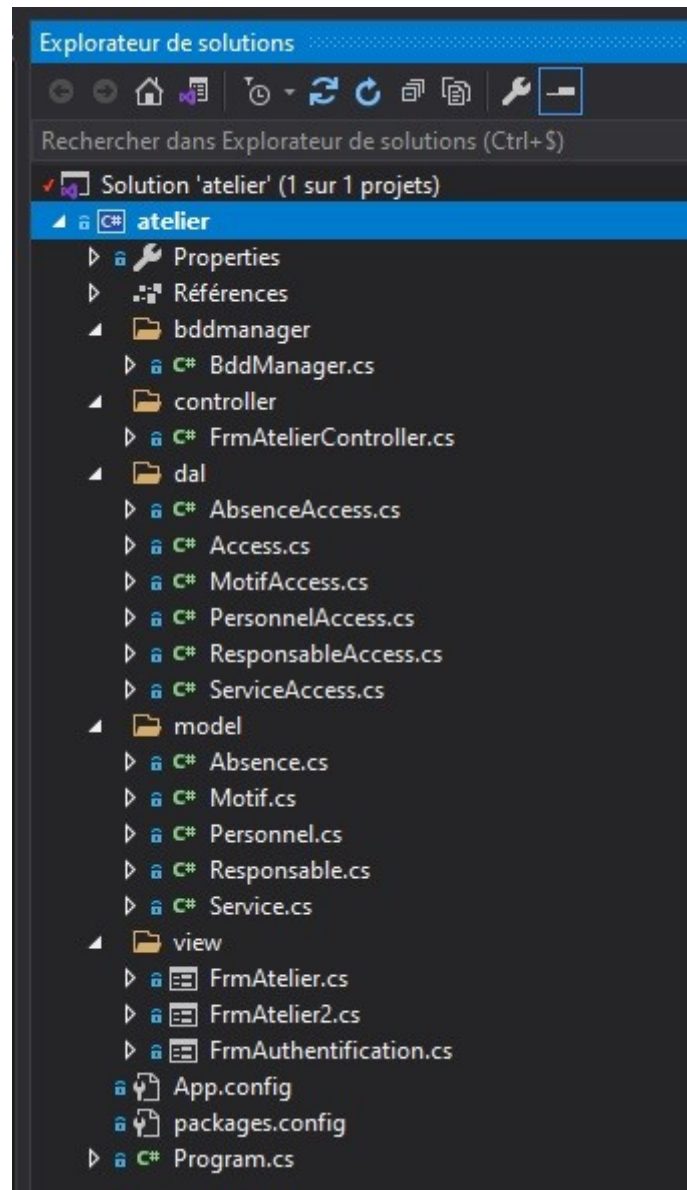


Créations de tous les packages nécessaire à l'application



Étape 4 : coder les fonctionnalités de l'application à partir des cas d'utilisation

Explications générale du code



BddManager permet de gérer la connexion à une base de données MySQL et de gérer l'exécution de requêtes SQL.

FrmAtelierController fait le lien entre model, view et dal.

Dans le pattern MVC, dal est la couche d'accès aux données.

Access est donc le moyen de se connecter à la base de donnée on le paramètre avec le login et mot de passe que l'on a crée durant l'étape 1.

AbsenceAccess, contient toutes les requêtes SQL qui concerne la classe absence. Il en est de même pour MotifAccess, PersonnelAccess, ResponsableAccess et ServiceAccess elles contiennent les requêtes de leur classes respectives.

Model contient la création des classes Absence, Motif, Personnel, Responsable et Service.

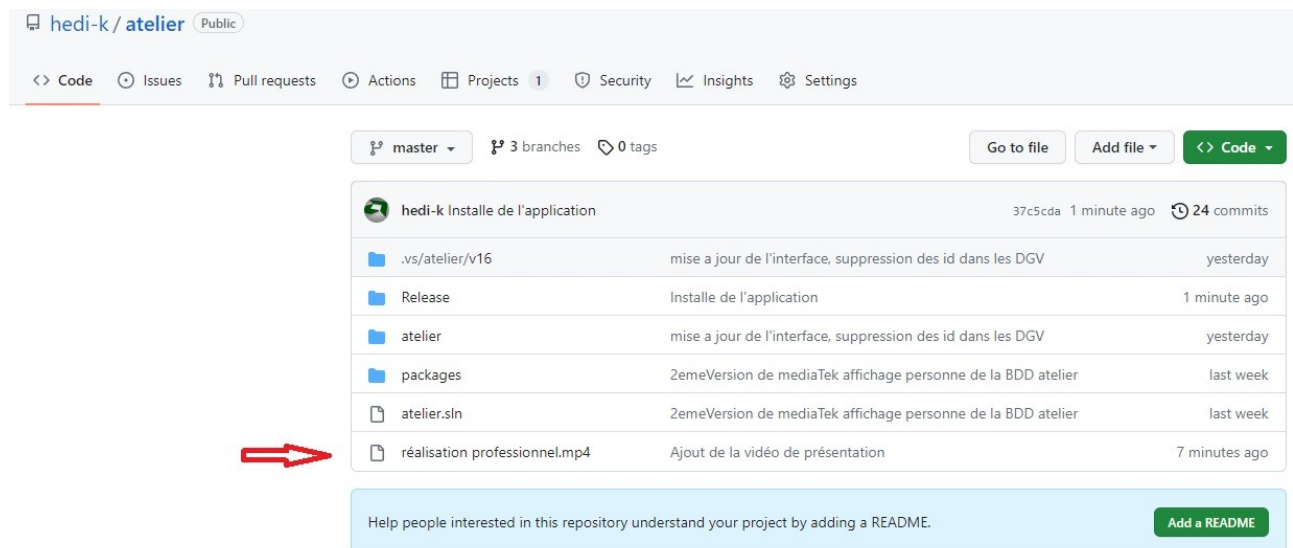
La view contient les forms et composant graphiques.

Étape 5 : créer une documentation utilisateur en vidéo

La documentation video est réalisé via OBS studio

Elle est hébergée sur mon github :

<https://github.com/hedi-k/atelier>



Étape 6 : gérer le déploiement

L'installateur et le script de la BDD se trouvent aussi sur le github :

<https://github.com/hedi-k/atelier>

master ▾

3 branches

0 tags

[Go to file](#)[Add file ▾](#)[Code ▾](#)

hedi-k script de la BDD associé à l'application

48e0d2e · 11 minutes ago

25 commits



.vs/atelier/v16

mise a jour de l'interface, suppression des id dans les DGV

yesterday



Release

Installe de l'application

19 minutes ago



atelier

mise a jour de l'interface, suppression des id dans les DGV

yesterday



packages

2emeVersion de mediaTek affichage personne de la BDD atelier

last week



atelier.sln

2emeVersion de mediaTek affichage personne de la BDD atelier

last week



atelier.sql

script de la BDD associé à l'application

11 minutes ago



réalisation professionnel.mp4

Ajout de la vidéo de présentation

25 minutes ago

Help people interested in this repository understand your project by adding a README.

[Add a README](#)