

Sommaire

Schydloowsky Djoulita

5/06/2025

Introduction

I – Contexte

II – Mission globale

III – Présentation des étapes

IV – Liens GitHub et Canva

Introduction

Ce compte rendu présente les différentes étapes de réalisation de l'**Atelier 2** du BTS SIO SLAM, consacré à la conception et au développement d'une application web de gestion des absences du personnel.

Ce projet a été mené dans un contexte pédagogique visant à mettre en pratique les compétences en **développement d'applications**, **gestion de bases de données**, **programmation en architecture MVC**, et **documentation technique**.

Réalisé à l'aide de technologies comme **ASP.NET**, **Visual Studio** et **MySQL**, le projet a nécessité une organisation rigoureuse, la création d'interfaces adaptées, la manipulation de données, et la mise en œuvre d'un dépôt GitHub pour assurer le suivi du développement.

Malgré des difficultés d'organisation personnelles (travail en plusieurs fois, oublis, blocages techniques), ce projet m'a permis de progresser de façon significative dans la maîtrise des outils de développement web et dans la gestion complète d'un projet informatique.

Contexte

L'atelier 2 du **BTS SIO SLAM** s'inscrit dans une démarche de mise en situation professionnelle. Il a pour objectif principal la création d'une **application web de gestion des absences du personnel**, destinée à une **structure culturelle ou administrative**, telle qu'un centre culturel, une bibliothèque, une médiathèque ou un service municipal.

Ce projet permet de mobiliser des compétences transversales : analyse de besoins, modélisation de base de données, conception d'interfaces, développement orienté objet, programmation en architecture MVC, mise en place d'un environnement de développement, et documentation technique.

Il s'agit d'un cas concret dans lequel un **responsable** doit pouvoir se connecter de manière sécurisée à l'application, puis **gérer le personnel** et leurs **absences** (ajout, modification, suppression, consultation). L'atelier est découpé en étapes progressives allant de la préparation technique jusqu'à la livraison finale et la présentation fonctionnelle.

Environnement de développement et outils utilisés

<i>Outil / Technologie</i>	<i>Utilisation</i>
<i>Visual Studio 2022</i>	Développement de l'application en ASP.NET MVC
<i>ASP.NET MVC 5</i>	Architecture du projet (Modèle - Vue - Contrôleur)
<i>MySQL / phpMyAdmin</i>	Création et gestion de la base de données relationnelle
<i>Looping</i>	Modélisation du schéma relationnel selon la méthode Merise
<i>GitHub + Kanban</i>	Suivi de projet, gestion du code source, organisation des tâches

Mission globale

L'objectif principal de cet atelier était de **concevoir une application web complète**, fonctionnelle et sécurisée, permettant à un **responsable** d'assurer la gestion des absences du personnel au sein d'une organisation. Il s'agissait d'un projet réaliste, mené de bout en bout, avec une structuration rigoureuse et des livrables professionnels attendus à chaque étape.

L'application devait répondre aux besoins suivants :

Authentification sécurisée:

Le responsable devait pouvoir se connecter à l'application à l'aide d'un **identifiant et d'un mot de passe**.

Pour garantir la sécurité, le mot de passe devait être **chiffré à l'aide de l'algorithme SHA2 (256 bits)**, directement au niveau de la base de données MySQL. Cela permettait de simuler une connexion sécurisée respectant les bonnes pratiques en matière de protection des données personnelles.

Gestion des personnels:

Une fois connecté, le responsable devait accéder à une interface lui permettant de **consulter la liste du personnel**, avec la possibilité d'effectuer les actions suivantes :

- Ajouter** un membre du personnel via un formulaire dédié
- Modifier** les informations d'un personnel (nom, prénom, email, service...)
- Supprimer** un personnel en cas de départ

Ces fonctionnalités représentent les opérations essentielles du **CRUD (Create, Read, Update, Delete)**, et devaient être réalisées dans le respect du modèle MVC.

Gestion des absences:

L'application devait également permettre au responsable de :

- Afficher la liste des absences** enregistrées dans la base de données

-Ajouter, modifier et supprimer une absence, en précisant notamment le **motif**, la **date**, et le **personnel concerné**

Une attention particulière devait être portée sur l'ergonomie, afin que ces manipulations soient simples et efficaces pour l'utilisateur.

Structuration du projet (GitHub):

Le développement devait être organisé avec rigueur :

- Dépôt GitHub** contenant tout le code source du projet
- Création d'un **Project Kanban** avec des "issues" correspondant à chaque tâche (ex. : créer la table responsable, développer le formulaire d'ajout, générer la vidéo, etc.)
- Sauvegardes régulières avec des **commits clairs et commentés**

Cette démarche permettait de suivre l'avancement et de travailler de manière professionnelle.

Vidéo de démonstration :

L'une des productions attendues était la réalisation d'une **vidéo de 5 minutes**, servant de **documentation utilisateur**, présentant :

- L'application en fonctionnement
- La navigation entre les pages
- L'exécution des principales fonctionnalités avec explications vocales

L'objectif était de simuler une **prise en main réelle de l'application**.

Documentation technique et fonctionnelle:

Enfin, une **documentation complète** devait être produite, comportant :

- Un **compte rendu d'activité** détaillant toutes les étapes du projet
- La **documentation XML générée dans Visual Studio**
- L'organisation du code en MVC
- Les extraits de code et les explications nécessaires à la compréhension du fonctionnement global

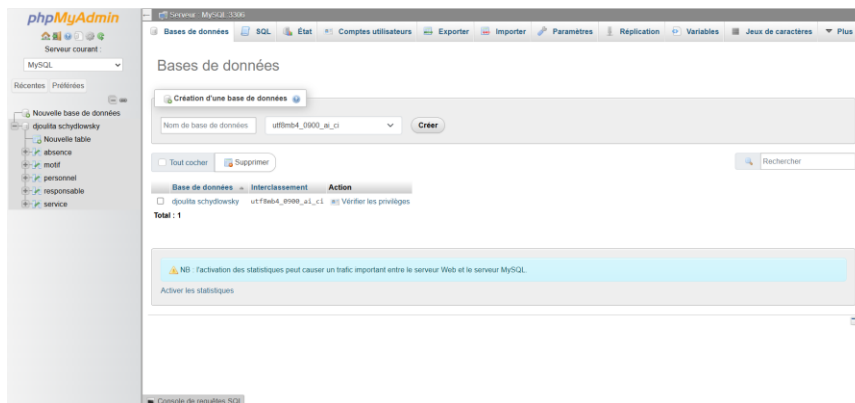
Cette documentation permettait de **valoriser le travail accompli** et de garantir sa **pérennité technique**.

Présentation des étapes

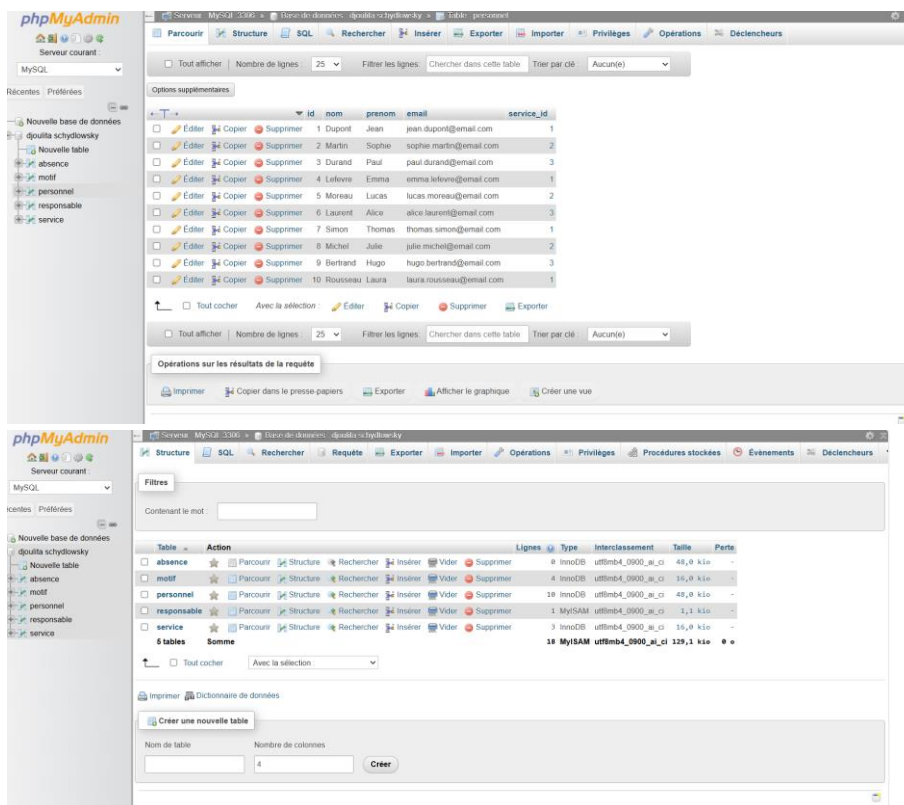
Étape 1 – Environnement de travail:

- Installation de Wampserver
- Création de la BDD djoulita avec tables responsable, personnel, absence, motif, service
- Remplissage via generatedata.com
- Création de l'utilisateur admin avec mot de passe hashé :

Base de Données:



Personnel:



Étape 2 – Interfaces et structure MVC:

Création des maquettes sur Canva :

- Page de connexion
- Tableau de bord personnel

-Formulaire d'ajout

Page de connexion, d'inscription et changement de Mdp :

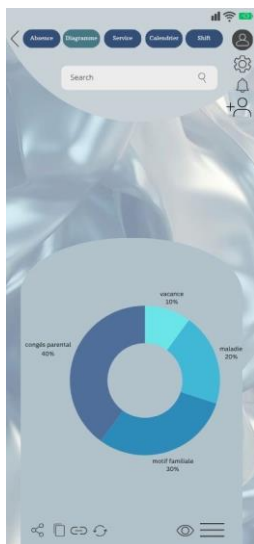
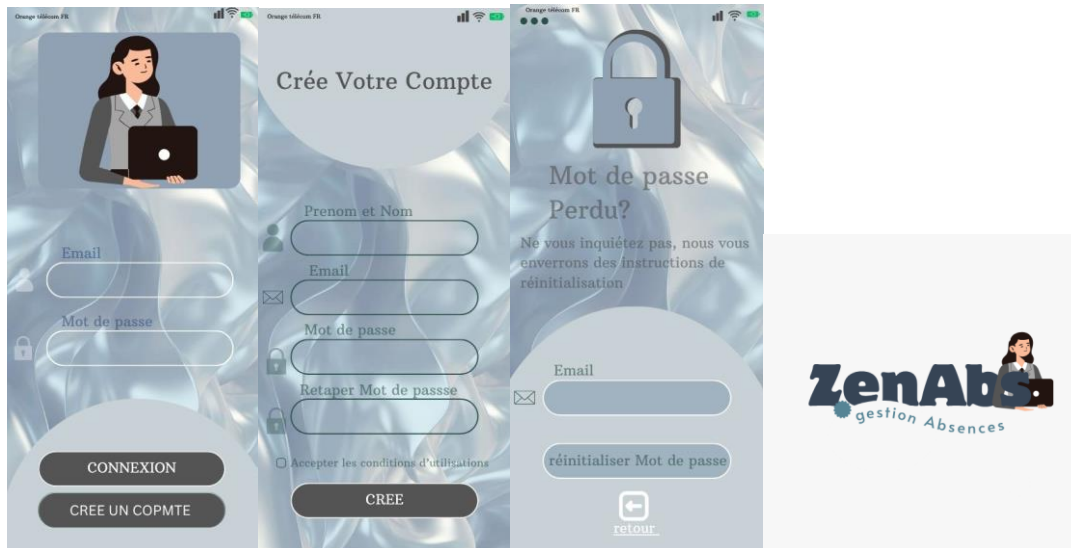


Tableau de bord Personnel

The form screen for adding a new absence. It features a table with columns for 'ID', 'NOM', 'PRENOM', 'EMAIL', and 'SERVICE'. The table is currently empty. Below the table, there are icons for adding, editing, deleting, and saving records. The screen also includes a search bar and a '+ Ajouter une Absence' button.

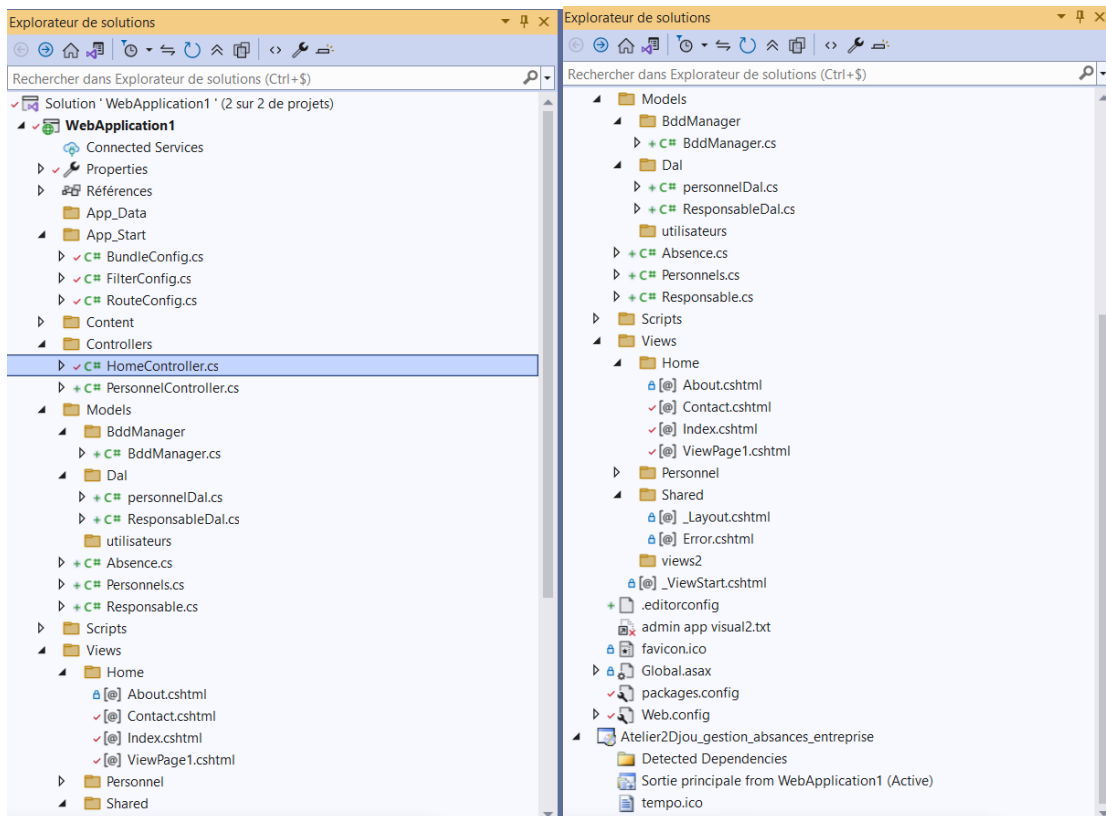
ID	NOM	PRENOM	EMAIL	SERVICE

Formulaire Ajout

Mise en place du projet ASP.NET MVC :

- Dossiers Models, Views, Controllers
- Ajout de GitHub + Project Kanban

master	1 Branch 0 Tags	Go to file	Add file	Code
Admin and Admin Ajoutez des fichiers projet. 069f501 · 2 days ago 2 Commits				
App_Start	Ajoutez des fichiers projet.	2 days ago		
Content	Ajoutez des fichiers projet.	2 days ago		
Controllers	Ajoutez des fichiers projet.	2 days ago		
Properties	Ajoutez des fichiers projet.	2 days ago		
Scripts	Ajoutez des fichiers projet.	2 days ago		
Views	Ajoutez des fichiers projet.	2 days ago		



Étape 3 – Connexion BDD et modèle:

Classe `BddManager.cs` (singleton)


```

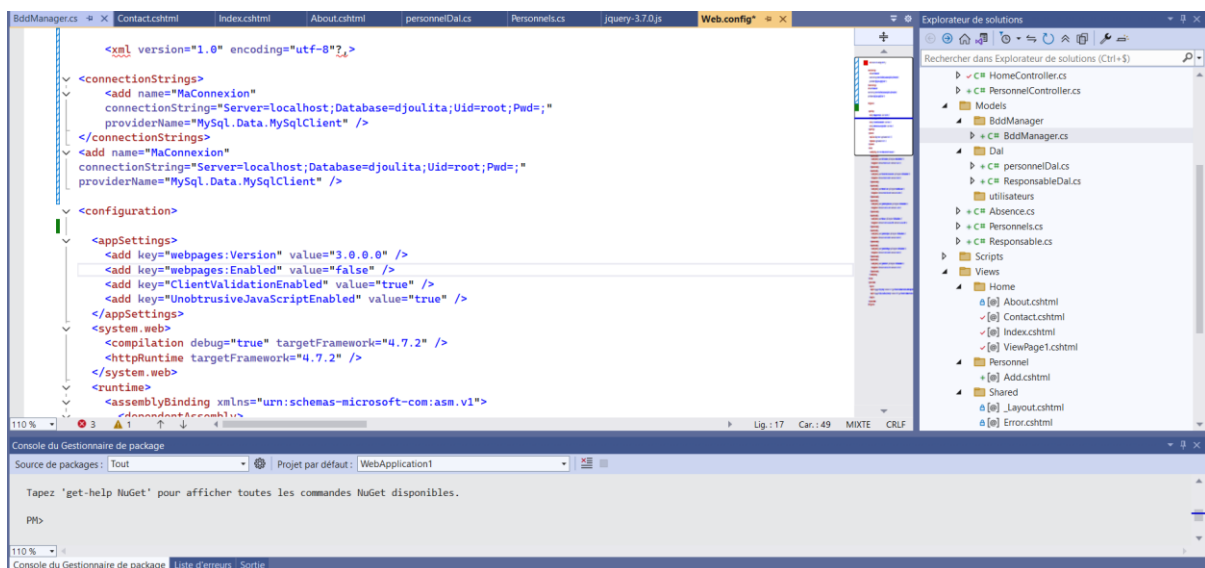
4 namespace WebApplication1.BddManager
5 {
6     0 références
7     public class BddManager
8     {
9         private static MySqlConnection connexion;
10
11     0 références
12     public static MySqlConnection GetInstance()
13     {
14         if (connexion == null)
15         {
16             string chaineConnexion = ConfigurationManager.ConnectionStrings["MaConnexion"].ConnectionString;
17             connexion = new MySqlConnection(chaineConnexion);
18         }
19         return connexion;
20     }
21 }

```

Modèles métiers créés :

- Personnel. Cs
- Responsable. Cs
- Absence.cs

Web.Config:



Étape 4 – Fonctionnalités :

Fichiers créés :

- Login.cshtml (vue de connexion)

- Index.cshtml (liste du personnel)
- Add.cshtml, Edit.cshtml

DAL pour la table `personnel` :

```

1  using System.Collections.Generic;
2  using WebApplication1.Models;
3  using WebApplication1.BddManager;
4  using MySql.Data.MySqlClient;
5
6  namespace WebApplication1.Dal
7  {
8      public class PersonnelDal
9      {
10         public static List<Personnel> GetAll()
11         {
12             List<Personnel> liste = new List<Personnel>();
13             MySqlConnection conn = BddManager.GetInstance();
14             MySqlCommand cmd = new MySqlCommand("SELECT * FROM personnel", conn);
15             MySqlDataReader reader = cmd.ExecuteReader();
16
17             while (reader.Read())
18             {
19                 Personnel p = new Personnel
20                 {
21                     Id = reader.GetInt32("id"),
22                     Nom = reader.GetString("nom"),
23                     Prenom = reader.GetString("prenom"),
24                     Email = reader.GetString("email")
25                 };
26                 liste.Add(p);
27             }
28             reader.Close();
29             return liste;
30         }
31     }
32 }

```

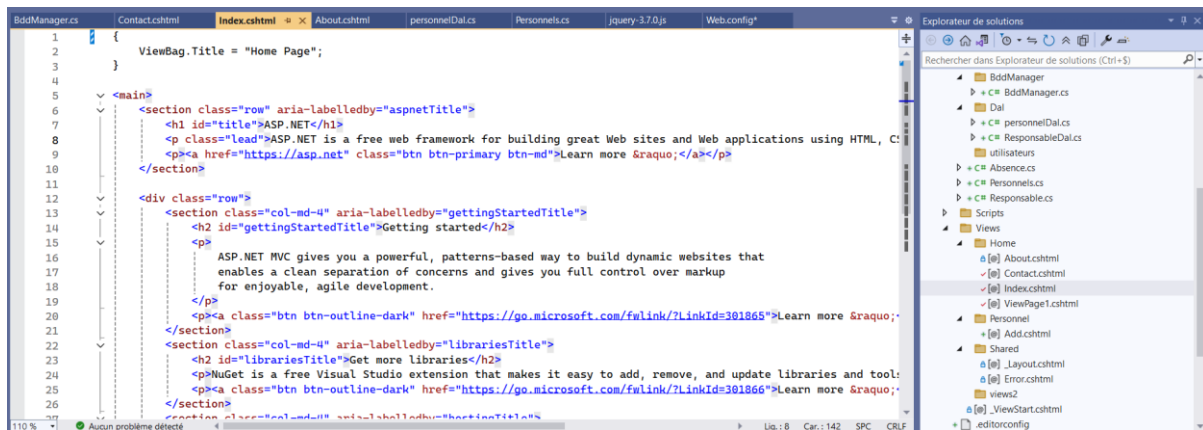
```

24         Prenom = reader.GetString("prenom"),
25         Email = reader.GetString("email"),
26         ServiceId = reader.GetInt32("service_id")
27     };
28     liste.Add(p);
29     reader.Close();
30     return liste;
31 }
32
33 public static void Add(Personnel p)
34 {
35     MySqlConnection conn = BddManager.GetInstance();
36     string sql = "INSERT INTO personnel(nom, prenom, email, service_id) VALUES (@nom, @prenom, @email, @service_id)";
37     MySqlCommand cmd = new MySqlCommand(sql, conn);
38
39     cmd.Parameters.AddWithValue("@nom", p.Nom);
40     cmd.Parameters.AddWithValue("@prenom", p.Prenom);
41     cmd.Parameters.AddWithValue("@email", p.Email);
42     cmd.Parameters.AddWithValue("@service_id", p.ServiceId);
43
44     cmd.ExecuteNonQuery();
45 }
46
47 }

```

Liste dynamique depuis la BDD :

Index.cshtml:



Difficultés rencontrées :

La réalisation de cet atelier n'a pas été linéaire. Le projet a été mené **en plusieurs sessions espacées**, ce qui a causé un **manque de continuité** dans le suivi des étapes. À plusieurs reprises, des **pauses prolongées** ont rendu difficile la reprise du travail, car certaines parties du code ou de l'organisation générale n'étaient plus claires.

Cette désorganisation a entraîné :

- La **multiplication d'erreurs techniques** (oubli de déclarations, noms de classes incohérents, erreurs dans la chaîne de connexion...)
- Des **dossiers ou fichiers mal positionnés** dans le projet Visual Studio
- Des **difficultés à retrouver les étapes déjà réalisées** ou celles encore à faire

Au lieu de suivre une progression étape par étape, certaines fonctionnalités ont été abordées dans le désordre, ce qui a demandé **plusieurs corrections, restructurations** et **allers-retours** entre le code, la base de données et la documentation.

- Retrouver une logique de travail claire et efficace

Cette difficulté m'a beaucoup appris : sur l'importance de l'organisation, de la rigueur dans les sauvegardes, et de la documentation au fil du projet.

Étape 5 – Documentation vidéo:

(Je n'ai pas eu le temps de la faire)

Une vidéo de démonstration a été faite avec :

- Présentation de l'application
- Navigation dans l'interface
- Explication de chaque fonctionnalité

Étape 6 – Finalisation & Portfolio

Page dans Canva avec :

- Le lien GitHub (code source + SQL complet)
- Le lien PDF du compte rendu

Liens GitHub et Canva :

Dépôt GitHub du projet ASP.NET – Gestion des absences :

<https://github.com/djoulita/WebApplication1>

Dépôt Canva du projet ASP.NET

https://www.canva.com/design/DAGjU6MlCeY/2qGoGLi3RxMS4bjqUdZAIQ/edit?utm_content=DAGjU6MlCeY&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton