

Equipe 2 - Artaud Jules / Vadier Mathis - Projet Karate



Présentation et objectifs

Nous sommes l'**équipe 2**, composée de **Artaud Jules** et **Vadier Mathis**. Nous avons choisi la **mission 4**.

Afin de réaliser notre mission, nous avons deux tâches à accomplir :

- **Gestion des compétitions** : Cette tâche doit permettre **la création, la modification et la suppression** d'une **compétition** à partir d'un formulaire.
- **Saisie des notes par un entraîneur** : Cette tâche doit permettre **la saisie, la modification et la suppression** des notes que met un entraîneur aux candidats participant à une compétition. Le formulaire doit permettre **la connexion** de l'entraîneur puis **la saisie** de toutes ses notes.

Environnement de travail

Avant de découvrir notre travail, laissez nous vous présenter notre environnement. Nous avons travaillé à l'aide d'une plateforme de

développement Web de type **WAMP**  **WampServer** et d'une **base de**

donnée  **MySQL** qui à été fournit au début de notre projet.

Par la suite, pour réaliser notre application nous avons utilisé l'IDE Visual

Studio 2019  sous un projet de type **Application Windows**

Forms(.NET Framework) qui utilise le Langage de programmation :  .

Connexion à la BD

Pour réaliser la connexion à la **base de donnée** nous avons du ajouter une extension nommé **MySql.Data** provenant du connecteur **Mysql** que l'on à du installer auparavant.

La connexion se fait à l'aide de la variable **connectionString** qui comporte les information sur **Le Serveur, la base de donnée, nom d'utilisateur et du mot de passe** :

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using MySql.Data.MySqlClient;
7
8  namespace PROJET_PPE2._1_KARATE
9  {
10     6 références | Jules ARTAUD, il y a 41 jours | 1 auteur, 1 modification
11     class bdd
12     {
13         6 références | Jules ARTAUD, il y a 41 jours | 1 auteur, 1 modification
14         public static MySqlConnection ConnectionBD()
15         {
16             // Création de la chaîne de connexion
17             string connectionString = "SERVER=localhost; DATABASE=karate; UID=root; PASSWORD=";
18             MySqlConnection conn = new MySqlConnection(connectionString);
19             return conn;
20         }
21     }
```

Apparence

L'onglet d'Accueil de notre application se présente ainsi :



L'onglet d'Ajout des compétition de notre application se présente ainsi:

The screenshot shows a window titled "Ajouter compétition" with a dark blue header. The main area has a grey background. At the top center, the text "AJOUT D'UNE COMPETITION" is displayed in a large, bold, black, sans-serif font. Below this, there are two input fields. The first is labeled "Numéro du club:" and is a simple white text box. The second is labeled "Date de la compétition:" and is a date picker showing "mercredi 9 mars" with a dropdown arrow. At the bottom, there are two buttons: an orange button with the text "AJOUTER" and a red button with the text "FERMER".

L'onglet de suppression et de modification des compétitions de notre application se présente ainsi:

Editor compétition

SUPPRIMER UNE COMPETITION

	NUM_COMPETITION	NUM_CLUB	DATE_COMPETITION
▶	1	3	03/03/2022 00:00:00
	2	1	02/03/2022 00:00:00
*			

SUPPRIMER

MODIFIER UNE COMPETITION

	NUM_CLUB	NOM_CLUB
▶	1	Karaté-Club Nancéien
	2	SCK Forbach
	3	Karaté-Club de Metz
*		

N° CLUB **MODIFIER** Date de la Compétition



L'onglet de saisi des notes de notre application se présente ainsi :

Ajouter note

AJOUT DE NOTES

	NUM_COMPETITION	NUM_LICENCE
▶	1	1234567890
	1	5678901234
	2	1234567890
	2	5678901234
*		

Numéro de compétition :

Numéro de licence :

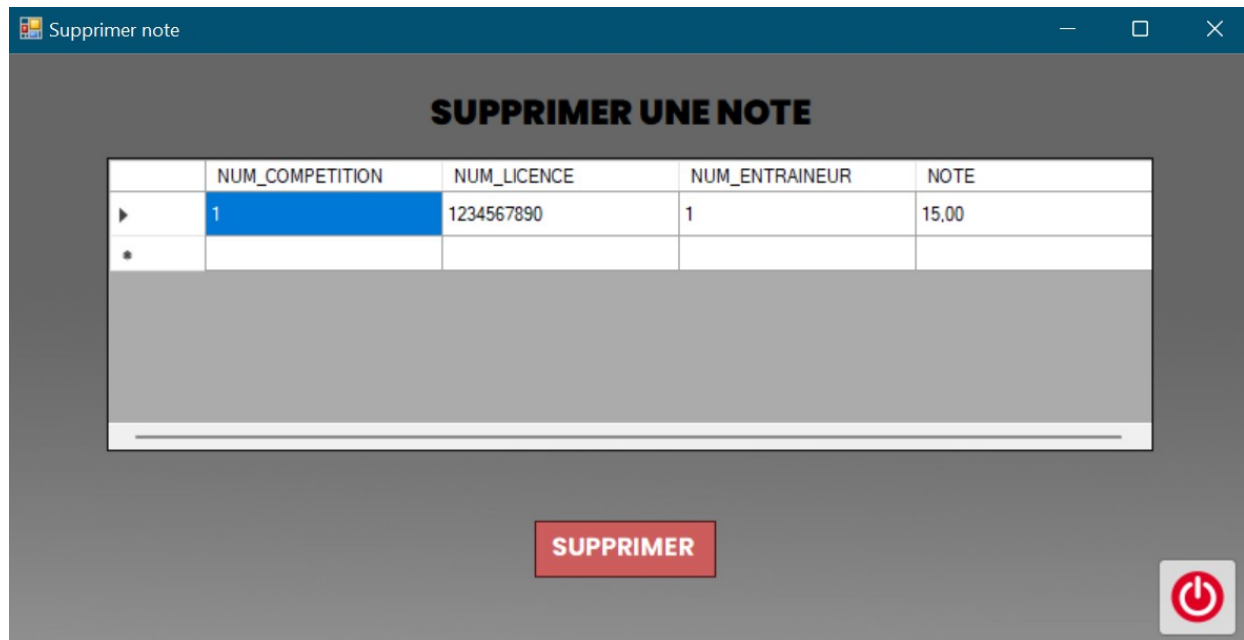
Note du jury 1 :

Note du jury 2 :

INSERER



Et enfin L'onglet de suppression des notes de notre application se présente ainsi :



Détails de notre travail

- [Ajouter une Compétition](#)
- [Supprimer ou modifier une compétition](#)
- [Saisir des notes](#)
- [Supprimer des notes](#)

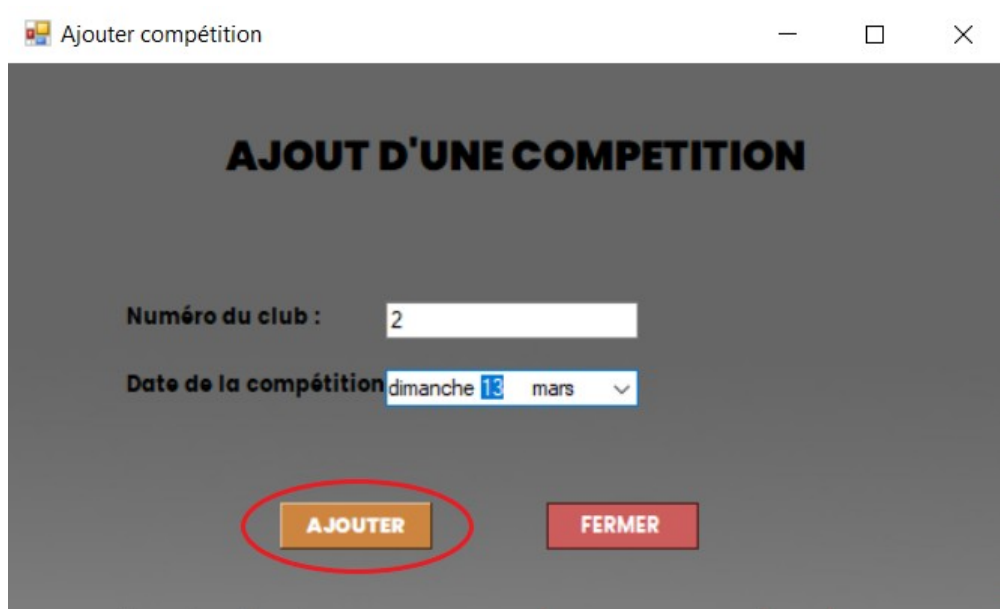
Conclusion

Pour la réalisation de notre AP, nous avons utilisé plusieurs outils informatique comme WampServer, Visual studio 2019 ou encore le langage de programmation C# que nous maîtrisons un peu ce qui nous a permis d'améliorer notre niveau en programmation et en analyse de base de donnée.

De plus, elle nous a permis de découvrir la conception d'application avec Windows Forms(.NET Framework) qui est plutôt intuitif et nous a permis d'apprendre beaucoup de choses.

Ajout des Compétitions

Bouton Ajouter



Lorsque l'on souhaite **ajouter** une compétition il nous suffit de rentrer le numéro du club et la date puis d'appuyer sur le bouton **AJOUTER** ce bouton possède un événement cliquable, lorsqu'on clique sur le bouton le code commence par une connexion à la base de données puis récupère avec une requête SQL le **NUM_COMPETITION** puisqu'il n'est pas en auto incrément afin de ne pas avoir de problème d'insertion :

```
private void Cmd_Ajouter_Click(object sender, EventArgs e)
{
    MySqlConnection conn = bdd.ConnectionBD();
    conn.Open();

    // Selection du plus grand NUM_COMPETITION de la table pour auto
    // incrémenter
    string SqlNumCompet = "SELECT MAX(NUM_COMPETITION) as
    Max_Compet FROM competition";

    MySqlCommand cmd = new MySqlCommand(SqlNumCompet,
    conn);

    MySqlDataReader reader = cmd.ExecuteReader();

    reader.Read();
}
```

```
// conversion en entier Le numéro obtenu
int nb = Int32.Parse(reader["Max_Compet"].ToString());

reader.Close();
```

Ici, nous gérons le cas d'erreur lorsqu'on saisit un numéro de club inexistant dans la base de donnée, pour cela on a créé un **READER** qui permet de lire une table suivant la requête donnée. La variable **@numClub** récupère le numéro saisi dans la TextBox **Txt_Num_Club** du formulaire.

```
// Vérification numéro de club existant

        string sqlNumClub = "SELECT NUM_CLUB FROM club where
num_club=@numClub";

        MySqlCommand cmdClub = new MySqlCommand(sqlNumClub,
conn);

        cmdClub.Parameters.AddWithValue("@numClub",
Txt_Num_Club.Text);
        // Ce reader permet de vérifier si le numéro de club saisi existe
dans
        //La table club
        MySqlDataReader readerClub = cmdClub.ExecuteReader();

        if (readerClub.Read())
        {

            conn.Close();
```

Ici nous gérons l'insertion des données saisies par l'utilisateur à l'aide d'une requête SQL de type **INSERT**, nous gérons également le cas d'erreur si l'utilisateur décide de saisir une date inférieure à la date du jour.

La variable **@numCompet** récupère le numéro saisi par l'utilisateur dans la TextBox plus haut dans le code et on lui ajoute 1 pour gérer l'auto incrémentation.

La variable **@numClubSaisie** récupère le numéro de club saisi par l'utilisateur dans la TextBox **Txt_Num_Club**

La variable **@DateCompet** récupère la date saisi par l'utilisateur dans le DatePicker **Date_Competition**

```
conn.Open();

        string sql = "INSERT INTO competition (NUM_COMPETITION,
NUM_CLUB,DATE_COMPETITION)" +
        " VALUES(@numCompet,@numClubSaisie,@DateCompet)";
```

```

        MySqlCommand cmdInsert = new MySqlCommand(sql, conn);

        cmdInsert.Parameters.AddWithValue("@numCompet", nb + 1);
        cmdInsert.Parameters.AddWithValue("@numClubSaisie",
Txt_Num_Club.Text);
        cmdInsert.Parameters.AddWithValue("@DateCompet",
Date_Competition.Value);

        if (Date_Competition.Value.Date >= DateTime.Today &&
Txt_Num_Club.Text!="")
        {
            cmdInsert.ExecuteNonQuery();
            MessageBox.Show("Compétition ajoutée");
            Txt_Num_Club.Clear();
        }
        else
        {
            MessageBox.Show("Date incorrect");
        }

    }
    else
    {
        MessageBox.Show("Numéro de club inexistant");
    }

    conn.Close();
}

```

Bouton Fermer

Ajouter compétition

AJOUT D'UNE COMPETITION

Numéro du club :

Date de la compétition : jeudi 10 mars ▼

AJOUTER **FERMER**

Ici nous avons un évènement qui permet de fermer le formulaire lorsqu'on clique sur le bouton.

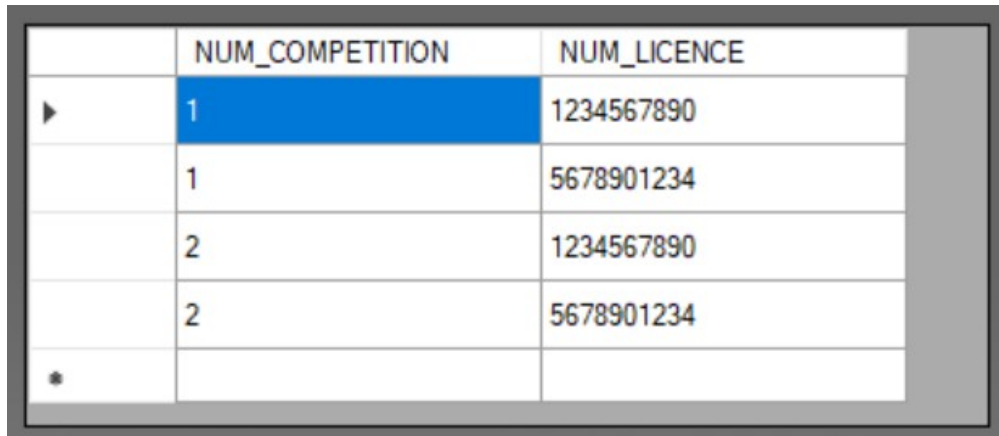
```
private void Cmd_Annuler_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Revenir à l'Accueil : [Accueil](#)

Saisir des Notes

Reader

Lors du démarrage du formulaire, il y a 1 DataGridView récupérant une partie de la table **INSCRIPTION** de la base de donnée à l'aide d'un **READER** :



	NUM_COMPETITION	NUM_LICENCE
▶	1	1234567890
	1	5678901234
	2	1234567890
	2	5678901234
*		

```
public Frm_Notes()
{
    InitializeComponent();

    MySqlConnection conn = bdd.ConnectionBD();

    conn.Open();

    string sqlInscription = "SELECT * FROM inscription";

    MySqlCommand cmdNote1 = new
    MySqlCommand(sqlInscription, conn);

    MySqlDataReader readerInscription =
    cmdNote1.ExecuteReader();

    while (readerInscription.Read())
    {
        string NUM_COMPETITION =
        (readerInscription["NUM_COMPETITION"].ToString());
        string NUM_LICENCE =
        (readerInscription["NUM_LICENCE"].ToString());

        Grid_Inscription.Rows.Add(NUM_COMPETITION,
        NUM_LICENCE);
    }
}
```

```

        conn.Close();
    }

```

Bouton Supprimer

Lorsque l'on souhaite **saisir** une note pour une compétition il nous suffit de saisir une compétition, un numéro de licence ainsi que 2 notes puis d'appuyer sur le bouton **INSERER**, l'événement du bouton **INSERER** démarre une connexion à la base de donnée.

Nous avons fait 2 requêtes de types **INSERT**, une pour insérer les données saisie par l'utilisateur avec **1** pour le *NUM_ENTRAINEUR* et **2** pour la seconde requête. Les erreurs d'insertion de type **MySqlException** sont gérés à l'aide d'un **try catch**

```

private void Cmd_SaisirNote_Click(object sender, EventArgs e)
{
    MySqlConnection conn = bdd.ConnectionBD();

    conn.Open();

    string noteSQL = "INSERT INTO note
(NUM_COMPETITION,NUM_LICENCE,NUM_ENTRAINEUR,NOTE) VALUES
(@numCompet,@numLicence,1,@note)";

    MySqlCommand cmdNote1 = new MySqlCommand(noteSQL,

```

```

conn);

        if (txt_noteJury1.Text != "" && txt_noteJury2.Text != "" &&
txt_numCompet.Text != "" && txt_numLicence.Text != "")
        {
            try
            {
                cmdNote1.Parameters.AddWithValue("@numCompet",
int.Parse(txt_numCompet.Text));
                cmdNote1.Parameters.AddWithValue("@numLicence",
txt_numLicence.Text);
                cmdNote1.Parameters.AddWithValue("@note",
int.Parse(txt_noteJury1.Text));

                string note2SQL = "INSERT INTO note
(NUM_COMPETITION,NUM_LICENCE,NUM_ENTRAINEUR,NOTE) VALUES
(@numCompet,@numLicence,2,@note2)";

                MySqlCommand cmdNote2 = new
MySqlCommand(note2SQL, conn);

                cmdNote2.Parameters.AddWithValue("@numCompet",
int.Parse(txt_numCompet.Text));
                cmdNote2.Parameters.AddWithValue("@numLicence",
txt_numLicence.Text);
                cmdNote2.Parameters.AddWithValue("@note2",
int.Parse(txt_noteJury2.Text));

                cmdNote1.ExecuteNonQuery();
                cmdNote2.ExecuteNonQuery();
                MessageBox.Show("Ajout des notes : SUCCES");
            }
            catch(MySqlException p)
            {
                MessageBox.Show("Une note à déjà été inséré");
            }
        }
        else
        {
            MessageBox.Show("Erreur veuillez recommencer");
        }
        this.Close();
    }
}

```

Bouton Quitter



Lorsque l'on souhaite **Quitter** cette onglet il nous suffit d'appuyer sur le bouton **QUITTER**, suite à ça l'onglet se ferme grâce au code suivant :

```
private void cmd_Quitter_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Revenir à l'Accueil : [Accueil](#)

Supprimer des Notes

Reader

Lors du démarrage du formulaire, il y a 1 DataGridView récupérant une partie de la table **NOTE** de la base de donnée à l'aide d'un **READER** :

	NUM_COMPETITION	NUM_LICENCE	NUM_ENTRAINEUR	NOTE
▶	1	1234567890	1	15,00
•				

```
public Frm_SupprNotes()
{
    InitializeComponent();

    MySqlConnection conn = bdd.ConnectionBD();

    conn.Open();
    string sqlnote= "SELECT NUM_COMPETITION, NUM_LICENCE,
NUM_ENTRAINEUR, NOTE FROM note";

    MySqlCommand cmdNote = new MySqlCommand(sqlnote, conn);
    MySqlDataReader readerNote = cmdNote.ExecuteReader();

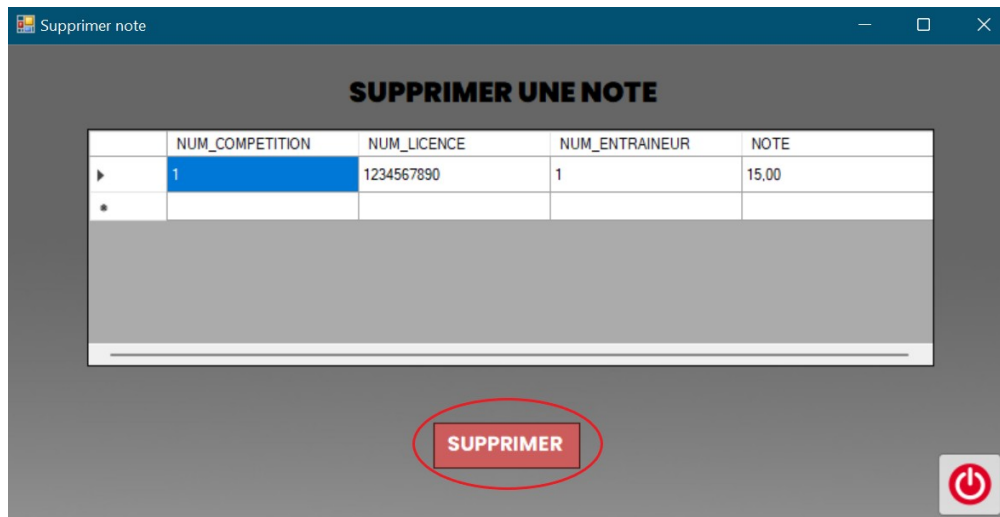
    while (readerNote.Read())
    {
        string NUM_LICENCE =
(readerNote["NUM_LICENCE"].ToString());
        string NUM_ENTRAINEUR =
(readerNote["NUM_ENTRAINEUR"].ToString());
        string NOTE = (readerNote["NOTE"].ToString());
        string NUM_COMPETITION =
(readerNote["NUM_COMPETITION"].ToString());

        GridView_Note.Rows.Add(NUM_COMPETITION, NUM_LICENCE,
NUM_ENTRAINEUR, NOTE);
    }

    readerNote.Close();
    conn.Close();
}
```

```
}
```

Bouton Ajouter



Lorsque l'on souhaite **supprimer** une note il nous suffit de sélectionner la note avec le numéro de licence et le numéro de compétition puis d'appuyer sur le bouton **SUPPRIMER** ce bouton possède un événement cliquable, lorsqu'on clique sur le bouton, la variable **ligne** récupère le numéro de la ligne de la DataGridView sélectionné par l'utilisateur, on utilise une requête de type **DELETE** afin de supprimer des enregistrements dans la base de donnée qui a pour condition les valeurs correspondant à la ligne choisit par l'utilisateur.

```
private void Cmd_delete_Click(object sender, EventArgs e)
{
    MySqlConnection conn = bdd.ConnectionBD();

    conn.Open();

    int ligne = GridView_Note.CurrentCell.RowIndex;

    decimal note = Convert.ToDecimal(GridView_Note[3,
ligne].Value);
    int numEntraîneur = Convert.ToInt32(GridView_Note[2,
ligne].Value);
    string numLicence = GridView_Note[1,
ligne].Value.ToString();
    int numCompet = Convert.ToInt32(GridView_Note[0,
ligne].Value);

    string sqlDelete = "DELETE FROM note WHERE NOTE = @note AND
NUM_LICENCE =@numLicence AND NUM_ENTRAINEUR=@numEntraîneur AND
NUM_COMPETITION=@numCompet";
```

```

        MySqlCommand cmdDelete = new MySqlCommand(sqlDelete,
conn);

        cmdDelete.Parameters.AddWithValue("@note", note);
        cmdDelete.Parameters.AddWithValue("@numCompet", numCompet);
        cmdDelete.Parameters.AddWithValue("@numEntraîneur",
numEntraîneur);
        cmdDelete.Parameters.AddWithValue("@numLicence", numLicence);

        cmdDelete.ExecuteNonQuery();
        conn.Close();

        this.Close();
    }

```

Bouton Quitter



Lorsque l'on souhaite **Quitter** cette onglet il nous suffit d'appuyer sur le bouton **QUITTER**, suite à ça l'onglet se ferme grâce au code suivant :

```

private void cmd_Quitter_Click(object sender, EventArgs e)
{
    this.Close();
}

```

Revenir à l'Accueil : [Accueil](#)