

BDW - Projet

Gestion de données autour de la danse

UCBL - Département Informatique de Lyon 1 – Printemps 2023

Table des matières

1 Informations générales sur le projet	1
2 Spécification de l'application	2
3 Mise en place de la Base de Données et Fonctionnalités	3
3.1 Construction de la base de données	3
3.2 Intégration des données fournies	3
3.3 Page d'accueil et statistiques	5
3.4 Fonctionnalités de base	6
3.5 Tableau de bord d'une école de danse	6
3.6 Tableau de bord d'une fédération	8
3.7 Fonctionnalité en plus : Compétition VIP	9
4 Déroulé du projet et jalons	10
4.1 Les jalons	10
4.2 Rendu du projet	10
4.3 Calendrier récapitulatif	10

1 Informations générales sur le projet

Le projet est à réaliser **en binôme** (les trinômes ne sont pas autorisés). La réalisation du projet en monôme nécessite l'accord du responsable de l'UE. Les participants d'un même binôme appartiennent de préférence au **même groupe de TD**.

A noter que la réalisation du projet nécessitera un travail personnel et/ou collaboratif complémentaire au travail réalisé durant les séances de TP. Il est donc important de choisir un binôme avec un emploi du temps compatible avec le votre.

A propos des fonctionnalités à développer

Vous n'avez pas toutes les fonctionnalités de l'application proposée à développer. Certaines fonctionnalités sont communes à tous les groupes, mais d'autres sont spécifiques à certains groupes! Pour cela, il est important de connaître votre nom de groupe qui est sur TOMUSS. Votre nom de groupe est de la forme $X1z$ ou $X2z$ avec X correspond à votre groupe de TD et z correspond à une lettre d'indexation.

Cluster	Lettre d'indexation
Cluster 1	a,e,i,m
Cluster 2	b,f,j,n
Cluster 3	c,g,k,o
Cluster 4	d,h,l,p

Ainsi, le groupe A2b fait partie du cluster 2, le groupe B1d fait partie du cluster 4...

Par la suite, vous verrez qu'en fonction de votre appartenance à un cluster vous aurez telle ou telle fonctionnalité à développer.

A propos de l'évaluation du projet

Le projet est évalué à travers les notes issues des différents QCM de Validation des Acquis pour le Projet, de l'évaluation des jalons et de la soutenance finale.

Gestion des plagias. Il est important de noter que toute ressemblance trop importante entre des projets sera sanctionnée par un 0 à chacun des protagonistes.

BDW est avant tout une UE d'introduction aux bases de données. La non-utilisation d'une BD relationnelle dans le projet impactera forcément la note de manière négative.

Attention à la gestion de votre temps! La mise en place de jalons réguliers est là pour vous aider dans ce sens. En effet, il a souvent été observé que des étudiants passaient trop de temps sur les aspects modélisation (avec l'objectif d'avoir un schéma EA parfait) au détriment du développement des fonctionnalités dont la compréhension tardive imposait finalement de modifier le schéma EA *a posteriori*.

2 Spécification de l'application

L'objectif du projet est de développer une application web qui permet de gérer un ensemble d'école de danses et de compétitions organisées par des fédérations.

Cette section va vous permettre de modéliser et concevoir la base de données sur laquelle s'appuiera votre application web. A partir de cette spécification, il est important de bien distinguer les données qui devront être stockées par rapport à celles qui seront calculées.

Dans un premier temps, nous considérons les écoles de danse qui sont identifiées par un entier, et on stocke leur nom et une chaîne de caractères contenant le ou les noms des fondateurs de l'école. L'école est située à une adresse. Par la suite, dès que l'on fait référence à une adresse, nous considérons un numéro de voie, une rue, un complément de rue, avec éventuellement une boîte postale et un numéro de cedex, un code postal, un nom de ville et un pays. L'école de danse dispose d'un ensemble de salles identifiées par un numéro relativement à chaque école. Une salle dispose d'un nom (qui peut être un numéro, une lettre, le nom d'un danseur, le nom d'une danse...) et sa superficie. Les salles peuvent faire référence aussi bien à des espaces de danse qu'à des vestiaires. Dans le cas d'espace de danse, on stocke les informations sur le type d'aération et le type de chauffage. Dans le cas de vestiaires, on indique s'ils sont mixtes ou pas et s'ils disposent de douches ou non.

Ensuite, on considère les personnes intervenants dans l'environnement d'une école de danse. Il y a d'une part, les employés de l'école et d'autre part, les adhérents. Concernant les employés, on dispose de leur nom et leur prénom (il peut y avoir des homonymes). A noter qu'au sein d'une école donnée, un employé ne peut avoir qu'une seule fonction (professeur, responsable accueil, entretien...) sur une période de temps. Les périodes de temps considérées ici sont de type année. Un employé peut travailler dans différentes écoles. Pour une période donnée, un professeur peut être responsable d'un ou plusieurs cours. Pour chaque cours, on dispose d'un libellé et d'une catégorie d'âge (3-6ans, 7-10ans, 11-15ans, 16-18ans, adulte). Un cours peut être un cours d'éveil à la danse, ou un cours de zumba ou un cours de danse. Pour les cours de zumba, on stocke l'ambiance principale (latino, dance...). Pour les cours de danse, on précise une catégorie (loisir, compétition) et le ou les types de danse (salsa, rock, modern'jazz...) enseignés dans le cours. Par exemple, un cours de 'Rock-Salsa' concerne deux types de danse. A noter que les types de danses ont des influences entre elles (Par exemple, le modern'jazz a pour influence les danses africaines et les danses européennes).

Concernant les adhérents, ils sont identifiés par un numéro de licence, et on stocke leur nom, prénom et date de naissance. On stocke également leur adresse. Un adhérent doit avoir un certificat médical couvrant sa période d'adhésion. Un adhérent est inscrit à un ou plusieurs cours. Les cours se déroulent en plusieurs séances. Chaque séance dispose d'un numéro qui l'identifie relativement au cours. Pour chaque séance, on précise le jour de la semaine et le créneau horaire de déroulement de la séance.

Une école de danse est rattachée à une fédération dont on stocke le nom, un sigle et une chaîne de caractères contenant le nom et le prénom du président de la fédération. Une fédération se décompose en comités. Un

comité peut être au niveau régional ou départemental. Les comités départementaux sont rattachés aux comités régionaux. Chaque comité est affilié à une fédération. La fédération propose l'organisation de compétitions qui sont gérées par les comités. Une compétition est disposée d'un code, on stocke son libellé (Championnat de France...) et son niveau (départemental, régionale, nationale, internationale). Chaque année, une édition de la compétition est organisée. Cette édition est identifiée par le couple (code de compétition, année). Pour chaque édition, on stocke la ville organisatrice et les informations sur la structure sportive où se déroule la l'édition de la compétition. Pour chaque structure sportive, on stocke son nom (ex : Raphaël de Barros), le type de structure (ex : gymnase) et son adresse. Lors d'une édition de compétition, des adhérents peuvent danser en couple ou en groupe. Dans le cas de la constitution d'un groupe de danse, on stocke un nom et l'information sur son genre, à savoir si le groupe est uniquement féminin, uniquement masculin ou mixte. Pour chaque participation d'un couple ou d'une équipe à une édition de compétition, on stocke son numéro de passage et son rang final.

3 Mise en place de la Base de Données et Fonctionnalités

3.1 Construction de la base de données

Dans un premier temps, vous allez concevoir la base de données à partir des spécifications précédentes. Votre modélisation doit respecter au mieux ces spécifications.

1. Produire un diagramme Entité/Association selon les spécifications fournies *via* l'outil Looping.
2. Produire le modèle relationnel dérivé de votre schéma Entité/Association. Attention : si vous générez automatiquement ce schéma avec l'outil de modélisation, il est fortement recommandé de le vérifier pour éventuellement le corriger ou le compléter.
3. Produire le script SQL permettant la création de votre base de données (*idem* que précédemment, il sera sûrement nécessaire de la modifier, si vous le générez automatiquement).

3.2 Intégration des données fournies

Dans un deuxième temps, vous allez migrer les données existantes fournies dans la base de données *dataset* dans votre base de données. L'utilisation de ces données est **obligatoire**. Les données fournies sont stockées dans une table qui ne correspond pas à votre modèle de données, il va donc être nécessaire de transformer les données fournies pour pouvoir les insérer dans vos tables. Ce processus, que l'on appelle *intégration de données* sera réalisé à travers des requêtes SQL que vous exécuterez dans PHPMyAdmin.

L'étape d'intégration peut se décomposer en 3 parties :

1. Réfléchir aux correspondances entre les données existantes et votre schéma (*e.g.*, à quel attribut de votre schéma correspond tel attribut fourni). Vérifier si les types de données sont cohérents entre attributs correspondants, et quelles sont les transformations nécessaires au niveau des valeurs. Il est important de comprendre que le jeu de données fourni n'est pas bien modélisé. Donc si vous modélisez votre schéma de base de données pour intégrer facilement les données fournies, votre schéma ne respectera plus les spécifications demandées.
2. Écrire les d'insertion du type `INSERT INTO maTable SELECT ... FROM donnees_fournies.instancesX;` permettant de peupler votre base.

Il est important de noter que la base *donnees_fournies* sera à terme (avant la soutenance) supprimée. Donc les données doivent être absolument récupérées dans votre propre base de données.

Remarque : il n'est pas demandé de développer une fonctionnalité permettant de gérer l'intégration depuis votre interface Web.

Principe par l'exemple

On considère d'une part, la relation initiale R définie comme suit :

$$R (idA, A1, A2, B1, B2, C1, C2, D)$$

En sachant que D est une valeur numérique représentant une capacité mémoire en Mo.

D'autre part, on dispose du schéma relationnel suivant :

$EA (idA, A1, A2), \#idB$

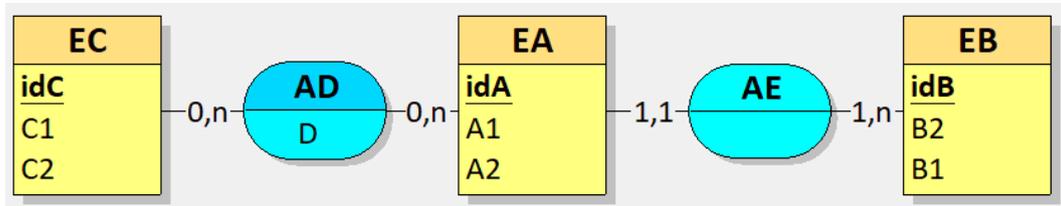
$EB (idB, B1, B2)$

$EC (idC, C1, C2)$

$ED (\#idA, \#idC, D)$

En sachant que D est exprimé en Ko.

Ce modèle relationnel répond au schéma Entité/Association suivant :



Comment intégrer les données de R dans les relation EA, EB, EC, ED ?

1. **Création de la table EB** : comme EB ne dispose d'aucune contrainte de clé étrangère, nous allons commencer par créer la table correspondante. Cependant, il n'y a pas de valeur de idB dans R. Nous allons donc définir idB de type `integer` avec un `AUTO_INCREMENT`. On peut donc créer la EB de la manière suivante :

```
CREATE TABLE EB (  
  idB integer AUTO_INCREMENT PRIMARY KEY,  
  B1 varchar(100),  
  B2 varchar(100)  
)
```

2. **Création de la table EC** : on utilise le même principe que précédemment. On peut donc créer la EC de la manière suivante :

```
CREATE TABLE EC (  
  idC integer AUTO_INCREMENT PRIMARY KEY,  
  C1 varchar(100),  
  C2 varchar(100)  
)
```

3. **Création de la table EA** : comme l'identifiant idA de 'EA' se trouve dans R, il est possible de récupérer directement sa valeur dans R. On peut donc créer la EA de la manière suivante :

```
CREATE TABLE EA (  
  idA integer PRIMARY KEY,  
  A1 varchar(100),  
  A2 varchar(100),  
  idB integer  
);  
ALTER TABLE EA ADD CONSTRAINT fk_EA_EB FOREIGN KEY (idB) REFERENCES EB (idB);
```

4. **Création de la table ED** : La table ED peut être définie de la manière suivante :

```
CREATE TABLE ED (  
  idA integer,  
  idB integer,  
  D integer );  
  
ALTER TABLE ED ADD CONSTRAINT fk_ED_EA FOREIGN KEY (idA) REFERENCES EA (idA);  
ALTER TABLE ED ADD CONSTRAINT fk_ED_EB FOREIGN KEY (idB) REFERENCES EB (idB);
```

5. **Peuplement de la table EB** : Pour cela, il est possible d'utiliser la requête suivante :

```
INSERT INTO EB (B1, B2) SELECT DISTINCT B1, B2 FROM R ;
```

Les valeurs de `idB` sont alors définies.

6. **Peuplement de la table EC** : Pour cela, il est possible d'utiliser la requête suivante :

```
INSERT INTO EC (B1, B2) SELECT DISTINCT B1, B2 FROM R ;
```

Les valeurs de `idB` sont alors définies.

7. **Peuplement de la table EA** : Pour cela, il est possible d'utiliser la requête suivante :

```
INSERT INTO EA (idA, A1, A2, idB)
SELECT DISTINCT R.idA, R.A1, R.A2, EB.idB FROM R JOIN EB ON R.B1 = EB.B1 AND
R.B2 = EB.B2 ;
```

La valeur de `idB` en clé étrangère est retrouvée grâce à la jointure sur `B1` et `B2`, que l'on sait unique dans ce cas particulier (du fait de la manière dont on a peuplé `EB`). Dans le cas général, il sera important de s'assurer que les attributs de jointures garantissent d'identifier de manière unique le n-uplet.

8. **Peuplement de la table ED** : Pour cela, il est possible d'utiliser la requête suivante en transformant les données quand cela est nécessaire :

```
INSERT INTO ED (idA, idC, D)
SELECT DISTINCT R.idA, EC.idC, (1024 * R.D) as D FROM R JOIN EC ON R.C1 = EC.C1 AND
R.C2 = EC.C2
```

3.3 Page d'accueil et statistiques

En parallèle du processus d'intégration, vous pouvez commencer à mettre en place votre site WEB à partir du code fourni lors des TP précédent. Il est important de conserver l'approche MVC que l'on vous a apprise.

A propos des statistiques

La première fonctionnalité de votre application est de mettre en place la page d'accueil qui offre un descriptif de l'application et un ensemble d'informations et de statistiques. Parmi ces informations et ces statistiques, il est demandé d'avoir au moins :

- "le nombre de fédérations, le nombre de comités régionaux et le nombre de comités départementaux stockés dans la base". Le résultat s'affichera selon le schéma (`nbFed`, `nbCR`, `nbCD`).
- "Le nombre d'écoles par code de département français". Un code de département français correspond aux deux premiers caractères du code postal s'il est différent de 97, et aux trois premiers caractères sinon.
- "la liste des comités régionaux (libellé) de la Fédération Française de Danse, triée selon l'ordre alphabétique inverse sur le libellé".
- "le top 5 des écoles (nom et ville) françaises qui ont eu le plus grand nombre d'adhérents en 2022". On précisera dans le résultat le nombre d'adhérents de l'école.

A propos du design du site

Chaque page de votre site dispose d'une structure qui vous est imposée. En effet, pour chaque page vous devez avoir les cinq zones suivantes :

- Un bloc "entête" (`<header>`), qui contient le nom du site et un logo cliquable pour toujours revenir à la page d'accueil
- Un bloc "menu" (`<nav>`) dans lequel chaque élément permet d'accéder à une page spécifique

- Un bloc "contenu de la page" ; contenu qui s'adaptera en fonction de la fonctionnalité à laquelle l'utilisateur souhaite accéder
- Un bloc "pied de page" (<footer>), qui contiendra au moins un lien vers le site de l'UCBL, le nom des auteurs du site, l'année de réalisation, un lien vers une page statique contenant le plan du site...

Concernant la mise en forme du site, elle se fera avec des styles CSS. Le projet sera évalué avec le navigateur Mozilla Firefox, donc vérifiez le rendu de votre site avec ce navigateur !

Concernant le rendu, il est possible d'embellir votre application en ajoutant des composants ou widget javascript. Il est cependant important de garder en mémoire que le projet est un projet PHP, donc la proportion de javascript doit rester raisonnable.

3.4 Fonctionnalités de base

Dans ce projet, vous allez avoir à développer des interfaces associées à des fonctionnalités :

- d'affichage d'un résultat simple d'une requête
- d'affichage d'un résultat tabulaire d'une requête
- d'ajout d'un élément dans la base
- de modification d'un n-uplet en base
- de suppression d'un n-uplet en base

Ces fonctionnalités serviront pour la gestion d'adhérents, d'écoles, de compétitions, de fédérations, de comités...

Il est important dans un premier temps de définir (ou récupérer dans les TP4-5) ces primitives de votre modèle que vous pourrez alors utiliser dans les tableaux de bord qui sont définis par la suite.

3.5 Tableau de bord d'une école de danse

***** Attention *** Il est important de se référer à votre numéro de cluster pour savoir quelles fonctionnalités vous avez à traiter**

Cette fonctionnalité consiste à permettre à une école de danse de gérer i) les données qui la concernent, ii) ses employés, iii) ses adhérents et iv) ses cours.

Dans un premier temps, l'utilisateur s'identifie en sélectionnant son nom dans une liste contenant les responsables d'école de danse trié par ordre alphabétique.

A noter qu'il n'est pas demandé ici de mettre un système d'authentification sécurisé avec mot de passe. Les aspects sécurité seront traités plus tard dans votre cursus.

En validant son choix, l'application affiche la page servant de tableau de bord de son école de danse.

Sur cette page, s'affiche :

- le nom et l'adresse de l'école,
- la liste des employés,
- le nombre d'adhérents pour l'année en cours,
- la liste des cours proposés dans l'école,
- le nombre d'adhérents ayant participé à une compétition.

Sur cette page, l'utilisateur a la possibilité d'accéder aux pages permettant :

- la gestion des informations de l'école,
- la gestion des employés,
- la gestion des adhérents,
- la gestion des cours,
- la gestion des inscriptions.

Remarque : pour permettre l'accès aux pages souhaitées dans le tableau de bord des écoles de danse, il sera nécessaire d'utiliser les paramètres dans l'URL. Par exemple, en supposant de la fonctionnalité 'Tableau de bord des écoles de danse' est nommée "*tdb_ecole*" et que les pages listées ci-dessus sont respectivement nommées "*info*", "*employe*", "*adherent*", "*cours*" et "*inscription*", vous pourrez accéder aux pages par les URL respectives :

- `.../index.php?f=tdb_ecole&p=info`
- `.../index.php?f=tdb_ecole&p=employe`
- `.../index.php?f=tdb_ecole&p=adherent`
- `.../index.php?f=tdb_ecole&p=cours`
- `.../index.php?f=tdb_ecole&p=inscription`

Il restera ensuite à traiter *via* `$_GET[]` les paramètres `f` et `p`.

Sur cette page de gestion **des informations de l'école**, l'utilisateur a la possibilité de :

- visualiser le nom de l'école, l'adresse de l'école et le nom du ou des fondateurs de l'école tels que définis actuellement,
- modifier les valeurs.

Remarque : Vous avez la possibilité de coupler la notion de visualisation avec celle de mise à jour, en utilisant un même formulaire pour les deux notions. Mais, vous pouvez également faire un affichage spécifique (sans passer par un formulaire), non modifiable mais avec une mise en forme adaptée à chaque type d'information pour la visualisation et gérer les modifications *via* un formulaire (sans mise en forme particulière).

Sur cette page de gestion **des employés de l'école**, l'utilisateur a la possibilité de :

- sélectionner un employé existant
- visualiser les informations sur l'employé, telles que définies actuellement,
- modifier les valeurs concernant l'employé.
- ajouter un nouvel employé

Sur cette page de gestion **des cours**, l'utilisateur a la possibilité de :

- sélectionner un cours existant
- visualiser les informations sur le cours, telles que définies actuellement,
- modifier les valeurs concernant le cours,
- ajouter un nouveau cours.

Sur cette page de gestion **des adhérents de l'école**, l'utilisateur a la possibilité de :

- sélectionner un adhérent existant
- visualiser les informations sur l'adhérent, telles que définies actuellement,
- modifier les valeurs concernant l'adhérent,
- ajouter un nouvel adhérent.

Sur cette page de gestion **des inscriptions de l'école**, l'utilisateur a la possibilité de :

- sélectionner un adhérent existant,
- visualiser la liste des cours suivis par l'adhérent,
- modifier une affectation de cours,
- ajouter une nouvelle affectation de cours à l'adhérent.

Attribution des fonctionnalités spécifiques

:

Cluster	Fonctionnalités
Cluster 1	gestion des employés de l'école
Cluster 2	gestion des cours
Cluster 3	gestion des adhérents de l'école
Cluster 4	gestion des inscriptions

3.6 Tableau de bord d'une fédération

Cette fonctionnalité consiste à permettre à un fédération de gérer i) les données qui la concernent, ii) ses comités, iii) ses membres et iv) ses compétitions.

Dans un premier temps, l'utilisateur s'identifie en sélectionnant son nom dans une liste contenant les membres de la fédération trié par ordre alphabétique.

En validant son choix, l'application affiche la page servant de tableau de bord de sa fédération.

Sur cette page, s'affiche :

- le nom, le sigle, le niveau de l'instance (nationale ou internationale) et l'adresse de la fédération,
- le nombre de comités,
- le nombre des membres de la fédération,
- la liste des compétitions organisées par la fédération,
- le nombre d'adhérents ayant participé à une compétition.

Sur cette page, l'utilisateur a la possibilité d'accéder aux pages permettant :

- la gestion des informations de la fédération,
- la gestion des informations des comités,
- la gestion des membres de la fédération,
- la gestion des compétitions.

Sur cette page de gestion des **fédération**, l'utilisateur a la possibilité de :

- visualiser le nom, l'adresse, le niveau de l'instance (nationale ou internationale), telles que définies actuellement,
- modifier la valeur des informations,
- ajouter une nouvelle fédération.

Sur cette page de gestion des **comités**, l'utilisateur a la possibilité de :

- visualiser le nom, le niveau de l'instance (régionale ou départementale) et le nom du ou de la présidente, telles que définies actuellement,
- modifier la valeur des informations
- ajouter un nouveau comité,
- rattacher un comité à une fédération.

Sur cette page de gestion des **membres**, l'utilisateur a la possibilité de :

- sélectionner un membre d'une fédération ou d'un comité
- visualiser le nom, le prénom et la date de naissance, tels que définis actuellement,

- modifier la valeur des informations,
- ajouter un nouveau membre,
- affecter le membre à une fédération ou un comité en précisant sa fonction dans ladite structure.

Sur cette page de gestion des **compétitions**, l'utilisateur a la possibilité de :

- sélectionner une compétition
- visualiser le nom et la liste des éditions de la compétition sélectionnée,
- modifier les informations relatives à la compétition,
- ajouter une compétition
- modifier les informations relative à une édition (y compris sur la structure sportive accueillant l'événement)
- ajouter une nouvelle édition,
- supprimer une édition
- supprimer une compétition
- inscrire un couple ou un groupe de danseurs à une édition de compétition,
- affecter un rang à un couple ou un groupe de danseurs lors d'une édition.

Attribution des fonctionnalités spécifiques

:

Cluster	Fonctionnalités
Cluster 1	gestion des fédérations
Cluster 2	gestion des comités
Cluster 3	gestion des membres
Cluster 4	gestion des membres

NB : la gestion des compétitions n'est pas une fonctionnalité spécifique.

3.7 Fonctionnalité en plus : Compétition VIP

Une fédération souhaite organiser une édition de compétition particulière dont l'objectif est d'attirer le maximum de public.

L'idée est i) d'inviter des danseurs qui ont été bien classés dans les différentes compétitions récentes et qui viennent d'écoles avec de nombreux adhérents et ii) de laisser l'organisation à un ou plusieurs comités parmi les plus expérimentés.

Pour cela, la fédération vous demande de trouver une solution pour générer automatiquement ces éditions VIP (liste des danseurs + lieu + liste de comités) tout en ayant la possibilité de choisir l'importance entre les paramètres (palmarès des danseurs, taille de l'école des danseurs, nombre d'organisation d'éditions par le ou les comités...) .

Que proposeriez-vous comme solution à la fédération ? Implémentez votre solution.

4 Déroulé du projet et jalons

Lors de la réalisation de votre projet, on vous demande de respecter le calendrier de jalons suivant.

4.1 Les jalons

JALON 1

Lors de la séance de TP du 24/03/2023, vous présenterez votre schéma entité association en l'état. Une appréciation vous sera alors attribuée.

JALON 2

Lors de la séance de TP du 31/03/2023, vous présenterez votre base de données peuplées et les requêtes statistiques. Une appréciation vous sera alors attribuée.

JALON 3

Lors de la séance de TP de la semaine du 03/04/2023, vous présenterez l'état d'avancement de votre tableau de bord d'une école de danse. Une appréciation vous sera alors attribuée.

JALON 4

Lors de la séance de TD du 24/04/2023, vous aurez un QCM Programmation Web permettant de s'assurer que vous avez acquis les compétences nécessaires grâce à la réalisation du projet. Le QCM sera sanctionné par une note.

4.2 Rendu du projet

Ce projet est à rendre avec les consignes de rendu suivantes :

- L'archive rendue (un seul par binôme) aura une extension .zip et il doit inclure votre/vos nom(s) à la fois dans le nom de fichier et en entête de chacun de vos fichiers
- Ce fichier zip contiendra au moins 4 fichiers :
 - l'archive contenant les pages de votre site (code commenté et indenté).
 - un fichier .sql qui contiendra le script SQL "exécutable" de création de votre base de données,
 - un fichier .sql qui contiendra le script SQL "exécutable" d'importation des données dans votre base,
 - un fichier contenant votre MCD (schéma E/A).

Ce fichier zip est à rendre le 27 avril à 23h59 dans la zone de dépôt TOMUSS dédiée (Rendu_projet).

4.3 Calendrier récapitulatif

Date	Type	Description
24/03/23	Jalon 1	Évaluation du schéma E/A en séance
31/03/23	Jalon 2	Évaluation état BD et statistique en séance
03/04/23	Jalon 3 (A1,B1,C1,D1)	Évaluation du tableau de bord d'une école de danse
07/04/23	Jalon 3 (A2,B2,C2,D2)	Évaluation du tableau de bord d'une école de danse
24/04/23	Jalon 4 (en TD)	QCM de validation d'acquisition des compétences en programmation Web
27/04/23	Jalon 5	Dépôt sur TOMUSS de votre rendu de projet
28/04/23	Soutenance	démonstration de l'application