



TÉLÉCOM NANCY

PROJET P2I

Réalisation d'une base de données et d'une application pour les consulter

Membres du projet :

Benjamin LAVANDIER
Lucas GOUIN
Hamza ABDOULHOUSSEN

Responsables du projet :

Sébastien DA SILVA
Gérald OSTER



Table des matières

1 Introduction au sujet	5
1.1 Contexte du projet	5
1.2 Organisation	5
1.3 Précisions légales	5
2 État de l'art	7
2.1 Concours Mines-Télécom	7
2.2 Base de données	7
2.3 Bootstrap	8
2.4 HTML / CSS / JavaScript	9
2.4.1 HTML	9
2.4.2 CSS	9
2.4.3 JavaScript	10
3 Gestion de projet	11
3.1 Équipe projet	11
3.2 Analyse du projet	11
3.2.1 Définition des objectifs	11
3.2.2 Analyse des risques : Matrice SWOT	11
3.3 Organisation du projet	12
3.4 Outils de travail	12
3.4.1 Réunions	12
3.4.2 Ressources et environnements utilisés	12
3.4.3 Partage du travail	14
3.4.4 Schéma relationnel	14
3.5 Comptes rendus des réunions	14
4 Conception de la base de données	15
4.1 Analyse des données	15
4.1.1 Lecture des fichiers	15
4.1.2 Unicité des candidats	15
4.1.3 Lien admis-admissible	15

4.1.4	Candidat en normal et en spé	15
4.1.5	Type admissible	16
4.1.6	Bonification et puissance	16
4.1.7	Lien entre les données	16
4.2	Schéma relationnel	19
4.2.1	Contraintes d'intégrités	20
4.2.2	Troisième forme normale	20
4.3	Indication colonne-table	20
4.4	Création de la base de données	20
5	Conception du site web	21
5.1	Présentation générale du site et des choix	21
5.1.1	Choix de conception	21
5.1.2	Fonctionnement	21
5.2	Présentation des différentes pages et de leur fonctionnement	22
5.2.1	Page d'accueil	22
5.2.2	Page des résultats d'un candidat	23
5.2.3	Page des résultats globaux	25
5.2.4	Page Statistiques	25
5.2.5	Page A propos	26
6	Conclusion	27
6.1	Bilan global du projet	27
6.2	Bilan du projet membre par membre	28
6.2.1	Hamza Abdoulhoussen	28
6.2.2	Benjamin Lavandier	28
6.2.3	Lucas Gouin	28
6.3	Travail réalisé	28
A	Comptes-Rendus	29
A.1	Compte-Rendu du 04/04/2021	29
A.2	Compte-Rendu du 06/04/2021	30
A.3	Compte-Rendu du 13/04/2021	30
A.4	Compte-Rendu du 27/04/2021	32
A.5	Compte-Rendu du 05/05/2021	32
A.6	Compte-Rendu du 13/05/2021	33
A.7	Compte-Rendu du 22/05/2021	34
A.8	Compte-Rendu du 30/05/2021	35
A.9	Compte-Rendu du 04/06/2021	35

Chapitre 1

Introduction au sujet

1.1 Contexte du projet

Ce projet a été réalisé dans le cadre du module PPII (Projet Pluridisciplinaire d'Informatique Intégrative) de la première année du cycle ingénieur sous statut étudiant de TELECOM Nancy.

L'objectif est de concevoir une base de données relationnelle à partir des données fournies sous formats .xlsx et .csv, pour ensuite créer une application permettant d'afficher les données de la base.

Ce projet a été décomposé en 3 parties : l'étude des données et la réalisation de la base de données, la conception du site web et la gestion du projet en équipe.

1.2 Organisation

Le rapport s'organise de la sorte :

- Dans le chapitre 2, nous présenterons le concours mines télécom et les différents concepts de la base de données
- Dans le chapitre 3, nous aborderons les éléments de gestion de projet mis en place
- Dans le chapitre 4, nous étudierons les données fournies, le schéma relationnel de la base de données et l'insertion des données
- Dans le chapitre 5, nous nous pencherons sur le site web que nous proposons
- Dans le chapitre 6, le bilan personnel et global du projet

1.3 Précisions légales

Ce projet n'est pas destiné à un usage commercial, ainsi, les images présentes, notamment les images de test, ne sont pas destinées à la publication et ne sont pas toutes libres de droits.

Cependant, le caractère strictement scolaire de ce projet nous autorise à les inclure en accord avec :

1. Code civil : articles 7 à 15, article 9 : respect de la vie privée
2. Code pénal : articles 226-1 à 226-7 : atteinte à la vie privée
3. Code de procédure civil : articles 484 à 492-1 : procédure de réfééré
4. Loi n°78-17 du 6 janvier 1978 : Informatique et libertés, Article 38

Chapitre 2

État de l'art

2.1 Concours Mines-Télécom

Ce concours provient de l’Institut de Mines-Télécom créé en 1996.

C'est un concours d'écoles d'ingénieurs qui ouvre les portes à 17 grandes écoles publiques (IMT Mines Albi, IMT Mines Alès, Mines Saint-Etienne – diplôme ISMIN, IMT Lille-Douai, Institut Mines-Télécom Business School, Télécom SudParis) et partenaires (EIVP, ENM- Météo France, ENSG Géologie Nancy, ENSG Géomatique, ENSTA Bretagne, ENTPE, ENSIIE, ENSSAT Lannion, Télécom Nancy, Télécom Physique Strasbourg et Télécom Saint-Etienne) et à 8 domaines d'avenir.

Il peut-être passé par des élèves de classes préparatoires des filières MP, PC, PSI, PT, TSI, ATS et BCPST.

Le Concours Mines-Télécom est la fusion du concours commun des ÉCOLES des MINES, du concours TÉLÉCOM INT et du concours TPE-EIVP. [2] [4]

*



FIGURE 2.1 – Logo de l’Institut Mines-Télécom

2.2 Base de données

Une base de données ou encore BDD sert à stocker des informations de manière organisée afin de pouvoir les consulter et/ou modifier simplement. Elle est notamment utilisée par des organismes ayant un grand nombre d’informations à stocker comme par exemple une banque avec tous ces utilisateurs et leurs comptes.

Une base de données optimale consiste à être de 3ème forme normale. Pour être de 3ème forme normale, il faut que la base respecte la 1ère et la 2ème forme normale également.

La 1ère forme normale (1FN) consiste à ce que chaque relation possède au moins une clé et que tous ses attributs soient atomiques. Un attribut est atomique s'il ne contient qu'une seule valeur.[9]

Par exemple dans la table ci-dessous, la première ligne ne respecte pas la 1FN car on a deux valeurs dans la case prénom. Alors que la deuxième relation respecte la 1FN.

ID	Prénom
1	Jacques, Rick
2	Pascal

TABLE 2.1 – Exemple d’illustration pour la 1FN

La 2ème forme normale (2FN) est respectée si chaque relation est en 1FN et si tout attribut n’appartenant à aucune clé primaire ne dépend pas d’une partie seulement de la clé primaire.[8]

La 3ème forme normale (3FN) signifie que la base est en 2FN et si tout attribut n’appartenant à aucune clé primaire ne dépend directement que des clés primaires de sa relation.[10]

2.3 Bootstrap

Bootstrap est une collection gratuite et open source de code CSS et JavaScript/jQuery utilisée pour créer des sites Web et des applications Web de mise en page dynamique. Cet outil de création se compose d’une série de modèles de conception basés sur HTML et CSS pour différents composants d’un site Web ou d’une application tels que des formulaires, des boutons, la navigation, les modaux, la typographie et d’autres composants d’interface avec JavaScript. Bootstrap est un outil puissant pour tout type de site Web et d’application Web que nous voulons créer.

De plus, Bootstrap fournit une solution prête à l’emploi avec des centaines de composants tiers que nous pouvons intégrer, ce qui nous permet de créer rapidement un prototype pour matérialiser notre site Web idéal sans perdre trop de temps. Ensuite nous pouvons juste apporter des modifications au template choisi afin d’arriver à notre vision du site Web ou application final.[3]



FIGURE 2.2 – Logo de Bootstrap

2.4 HTML / CSS / JavaScript

2.4.1 HTML

HTML signifie « HyperText Markup Language » qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure. D'autres technologies sont utilisées avec HTML pour décrire la présentation d'une page (CSS) et/ou ses fonctionnalités interactives (JavaScript).

L'« hypertexte » désigne les liens qui relient les pages web entre elles, que ce soit au sein d'un même site web ou entre différents sites web. Les liens sont un aspect fondamental du Web. Ce sont eux qui forment cette « toile » (ce mot est traduit par web en anglais). En téléchargeant du contenu sur l'Internet et en le reliant à des pages créées par d'autres personnes.

Le langage HTML utilise des « balises » pour annoter du texte, des images et d'autres contenus afin de les afficher dans un navigateur web. Le balisage HTML comprend des « éléments » spéciaux tels que <head>, <title>, <body> et bien d'autres.[6]



FIGURE 2.3 – Logo d'HTML

2.4.2 CSS

CSS est l'un des langages principaux du Web ouvert et a été standardisé par le W3C. Ce standard évolue sous forme de niveaux (levels), CSS1 est désormais considéré comme obsolète, CSS2.1 correspond à la recommandation et CSS3, qui est découpé en modules plus petits, est en voie de standardisation.

Il permet d'apporter plus de possibilités graphiques et des animations pour des pages HTML. Il fonctionne sous forme d'attributs donnés à un type d'éléments ayant un identifiant précis donné dans la page HTML ou non.[5]



FIGURE 2.4 – Logo de CSS

2.4.3 JavaScript

JavaScript (souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web. Mais il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs web tels que Node.js, Apache CouchDB voire Adobe Acrobat. Le code JavaScript est interprété ou compilé à la volée (JIT). C'est un langage à objets utilisant le concept de prototype, disposant d'un typage faible et dynamique qui permet de programmer suivant plusieurs paradigmes de programmation : fonctionnelle, impérative et orientée objet.

Ce langage permet d'apporter encore d'autres fonctionnalités aux pages Web que le CSS ou HTML de base ne peut faire. Le langage ressemble davantage à du code "classique" proche de python ou java par exemple.[7]



FIGURE 2.5 – Logo de javascript

Chapitre 3

Gestion de projet

3.1 Équipe projet

L'équipe se compose de trois étudiants en première année : Lucas GOUIN, Benjamin LAVANDIER et Hamza ABDOULHOUSSEN.

L'équipe projet s'est organisée ainsi :
Lucas est désigné chef de projet, il sera responsable de l'avancement du projet et de l'organisation des réunions. Benjamin est désigné secrétaire de projet, il sera responsable des comptes-rendus de réunions et les enverra sur la plateforme Gitlab pour les rendre accessibles aux autres membres du groupe.

3.2 Analyse du projet

3.2.1 Définition des objectifs

Nous avons défini les objectifs avec la méthode SMART :

	Critère	Indicateur
S	Spécifique	L'objectif est défini clairement.
M	Mesurable	L'objectif est mesurable, par des indicateurs chiffrés ou livrables.
A	Atteignable	L'objectif doit être motivant sans être décourageant et doit apporter un plus par rapport au début du projet.
R	Réaliste	L'objectif doit être réaliste au regard des compétences et de l'investissement de l'équipe du projet.
T	Temporellement défini	L'objectif doit être inscrit dans le temps, avec une date de fin et des jalons.

3.2.2 Analyse des risques : Matrice SWOT

Les risques, mais aussi les qualités du projet ont été examinés par l'équipe, à l'aide d'une matrice SWOT :

	Positif	Négatif
Interne	Compétence de Benjamin sur les interfaces graphiques en java Compétences obtenues avec le premier projet	Perte de motivation Potentielle mauvaise gestion du travail individuel
Externe	Possibilité de faire les réunions à l'école plutôt que sur Teams Possibilité de voir les professeurs en présentiel	Covid : quelqu'un tombe malade Problème informatique (ordinateur en panne, etc...)

TABLE 3.1 – Matrice SWOT

3.3 Organisation du projet

3.4 Outils de travail

3.4.1 Réunions

Les réunions se sont déroulées sur la plateforme Teams pour les réunions en distanciel et à l'école pour les réunions en présentiel. Des séances de travail collectif ont été organisées à l'école.

3.4.2 Ressources et environnements utilisés

Les ressources utilisées pour charger la base données sont les fichiers `xlsx` et `csv` fournies dans l'énoncé du projet.

Ces fichiers sont organisés comme suit :

- `fichiers_admis_normal` :
 - `ADMIS_ATS.xlsx`
 - `ADMIS_MP.xlsx`
 - `ADMIS_PC.xlsx`
 - `ADMIS_PSI.xlsx`
 - `ADMIS_PT.xlsx`
 - `ADMIS_TSI.xlsx`
- `fichiers_admissible_normal` :
 - `ADMISSIBLE_ATS.xlsx`
 - `ADMISSIBLE_MP.xlsx`
 - `ADMISSIBLE_PC.xlsx`
 - `ADMISSIBLE_PSI.xlsx`
 - `ADMISSIBLE_PT.xlsx`
 - `ADMISSIBLE_TSI.xlsx`
- `fichiers_admis_spe` :
 - `ADMIS_MP-SPE.xlsx`
 - `ADMIS_PC-SPE.xlsx`
 - `ADMIS_PSI-SPE.xlsx`
 - `ADMIS_PT-SPE.xlsx`
 - `ADMIS_TSI-SPE.xlsx`
- `fichiers_admissible_spe` :
 - `ADMISSIBLE_MP-SPE.xlsx`
 - `ADMISSIBLE_PC-SPE.xlsx`

```

    — ADMISSIBLE_PSI-SPE.xlsx
    — ADMISSIBLE_PT-SPE.xlsx
    — ADMISSIBLE_TSI-SPE.xlsx
— fichiers_Classes_xlsx :
    — Classes_MP_CMT_spe_XXXX.xlsx
    — Classes_PC_CMT_spe_XXXX.xlsx
    — Classes_PSI_CMT_spe_XXXX.xlsx
    — Classes_PT_CMT_spe_XXXX.xlsx
    — Classes_TSI_CMT_spe_XXXX.xlsx
— fichiers_Classes_csv :
    — Classes_MP_CMT_spe_XXXX_SCEI.csv
    — Classes_PC_CMT_spe_XXXX_SCEI.csv
    — Classes_PSI_CMT_spe_XXXX_SCEI.csv
    — Classes_PT_CMT_spe_XXXX_SCEI.csv
    — Classes_TSI_CMT_spe_XXXX_SCEI.csv

— fichiers_CMT :
    — CMT_Oraux_YYYY_MP.xlsx
    — CMT_Oraux_YYYY_PC.xlsx
    — CMT_Oraux_YYYY_PSI.xlsx
    — CMT_Oraux_YYYY_PT.xlsx
— fichiers_Ecrit :
    — Ecrit_MP.xlsx
    — Ecrit_PC.xlsx
    — Ecrit_PSI.xlsx
    — Ecrit_PT.xlsx
    — Ecrit_TSI.xlsx
— fichier_inscription :
    — Inscription.xlsx
— fichiers_liste :
    — listeEcoles.xlsx
    — listeEtablissements.xlsx
    — listeEtatsReponsesAppel.xlsx
— fichiers_Voeux :
    — listeVoeux_ATS.xlsx
    — listeVoeux_MP.xlsx
    — listeVoeux_PC.xlsx
    — listeVoeux_PSI.xlsx
    — listeVoeux_PT.xlsx
    — listeVoeux_TSI.xlsx
— fichiers_Oral :
    — Oral_MP.xlsx
    — Oral_PC.xlsx
    — Oral_PSI.xlsx
    — Oral_PT.xlsx
    — Oral_TSI.xlsx

```

Le langage de programmation utilisé pour réaliser ce projet est Python muni de la librairie **pandas**, librairie permettant de lire les fichiers **xlsx** et **csv**; la librairie **sqlite3** permettant de générer la base de données SQL et la librairie **flask** permettant de lancer le site web.

Au niveau de l'environnement de travail, certains ont utilisé Pycharm, d'autres Visual Studio Code.

L'initialisation du site internet a été automatisée. Il faut d'abord lancer le script *init.sh*. Cependant, il est possible que le script ne se lance pas en fonction de l'OS. Dans nos tests, le script fonctionne sous

MacOS, WSL (avec bash), et évidemment sous Linux. Cependant, il est possible que des dépendances s'installent mal.

3.4.3 Partage du travail

L'ensemble du travail au long du projet a été mis sur le répertoire Gitlab fourni par l'école. Chaque membre de l'équipe a poussé son travail pour le partager.

3.4.4 Schéma relationnel

Nous avons réalisé le schéma relationnel sur le site <https://dbdiagram.io/>. Il s'agit d'un outil simple permettant de dessiner un diagramme en écrivant du code. Le code pour le diagramme a aussi été ajouté au dépôt Git et a été mis à jour régulièrement.

3.5 Comptes rendus des réunions

Tous les comptes-rendus sont en annexe de ce rapport. Cependant, toutes les réunions n'en ont pas eu car on se retrouvait souvent dans le hall de l'école après les cours pour discuter de la suite du projet et de son avancement. C'était très pratique pour avoir un rapide aperçu du travail des tout le monde et prendre note des remarques ou améliorations à faire.

Chapitre 4

Conception de la base de données

4.1 Analyse des données

4.1.1 Lecture des fichiers

Nous avons réalisé une analyse des données fournies à l'aide de fonctions python (placées dans le dossier python, une description rapide est donnée dans le fichier `fonction vérification données.xlsx`).

Pour lire les fichiers `xlsx` et `csv` à partir du module `pandas`, Nous utilisons les fonctions `lecture` et `read_all` dans le fichier `main.py`.

Les différents groupes de fichiers (3.4.2) sont indiqués. Pour chaque groupe, une liste est initialisée. Elle porte le nom du groupe et contient les fichiers de ce groupe.

```
Exemple : fichiers_CMT = ['CMT_Oraux_YYYY_MP.xlsx', 'CMT_Oraux_YYYY_PC.xlsx  
'CMT_Oraux_YYYY_PSI.xlsx', 'CMT_Oraux_YYYY_PT.xlsx']
```

4.1.2 Unicité des candidats

Nous avons d'abord vérifié l'unicité des identifiants des candidats. Ainsi, l'identifiant pourra devenir l'élément permettant de distinguer les candidats. Le fichier `idunique.py` permet de vérifier l'unicité des candidats dans les fichiers `xlsx` et `csv` contenant des candidats sauf dans les fichiers `fichiers_Voeux` où l'identifiant n'est pas nécessairement unique.

Le script python peut se lancer avec l'environnement depuis la racine, avec la commande
`python3 python/idunique.py`

4.1.3 Lien admis-admissible

Ensuite, nous avons vérifié que tous les candidats admis sont bien admissibles. Le fichier `admis_admissible.py` vérifie pour chaque filières que les candidats des fichiers `fichiers_admis_normal` et `fichiers_admis_spe` sont bien compris dans les fichiers `fichiers_admissible_normal` et `fichiers_admissible_spe` correspondants. Tous les candidats admis sont bien admissibles.

4.1.4 Candidat en normal et en spé

Nous avons cherché à déterminer les candidats décrits par les fichiers `fichiers_admis_spe` et `fichiers_admissible_spe`. Pour cela, nous avons vérifié que tous les candidats en normal (`fichiers_admis_normal` et `fichiers_admissible_normal`) sont bien en spé (`fichiers_admis_spe` et `fichiers_admissible_spe`). Le script `classeSPE.py` a permis de déduire que tous les candidats de classes normales sont en classe spé.

4.1.5 Type admissible

Toujours pour décrire les candidats en spé, nous avons cherché un lien entre le type admissible donné dans les fichiers `fichiers_Classe_xlsx` et les candidats normaux ou spé. Le script `typeadmissible` compare pour chaque candidat, son type admissible et le fait qu'il est seulement en classe normale ou dans les deux classes normale et spé. Les candidats de type admissible A correspondent aux candidats seulement en classe normale. Les candidats de type admissible B correspondent aux candidats de classe normale et spé.

Les candidats de type admissible A passent la 1ère série épreuves orales, ceux de type B passent la 2de série d'épreuves orales.

4.1.6 Bonification et puissance

A l'aide du script `bonification.py` nous avons vérifié que tous les candidats en 3/2 possède une bonification. Ainsi, seule l'information de la puissance sera conservée dans la base de données.

4.1.7 Lien entre les données

Pour comparer les informations des diverses fichiers `xlsx` et `csv`. Nous avons réalisé les fonctions `contient` et `compare`.

Pour la fonction `contient` :

Paramètre d'entrée : deux couples de chaînes de caractère. Le premier couple `grand` contient le nom du premier fichier et le nom de la colonne. Le deuxième couple `petit` contient le nom du second fichier et le nom de la colonne.

Sortie : Booléen qui vaut vrai si la colonne de `petit` est inclus dans la colonne de `grand`.

Exemple :

```
grand = ('listeEtablissements.xlsx', 'Etab')
petit = ('Inscription.xlsx', 'ETABLISSEMENT')
contient(grand, petit)
```

Ici, `contient(grand, petit)` renvoie vrai si les établissements du fichier `Inscription.xlsx` sont tous dans `listeEtablissements.xlsx`.

Pour la fonction `compare` :

Paramètre d'entrée : deux triplets de chaînes de caractère. Le premier triplet `t1` contient le nom du premier fichier, la colonne qui permet de comparer et la colonne à comparer. Le deuxième triplet `t2` contient les mêmes informations pour le second fichier.

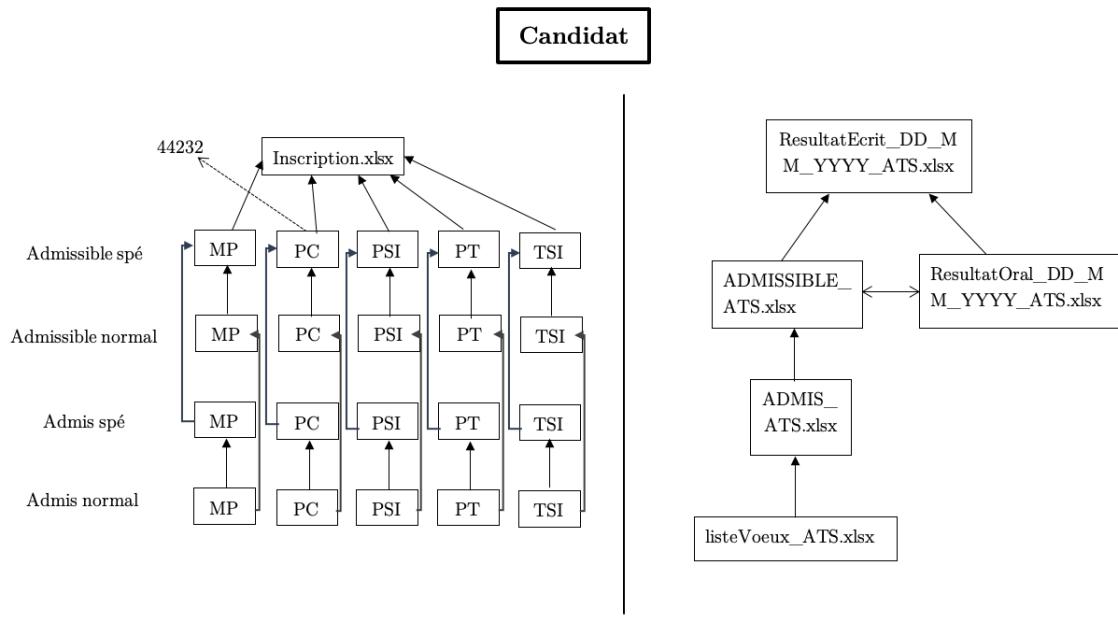
Sortie : Booléen qui vaut vrai si la deuxième colonne de `t1` est la même que celle de `t2` lorsque les premières sont identiques.

Exemple :

```
file1 = 'listeEtablissements.xlsx'
id1 = 'Rne'
col1 = 'Etab'
t1 = (file1, id1, col1)
file2 = 'Inscription.xlsx'
id2 = 'CODE_ETABLISSEMENT'
col2 = 'ETABLISSEMENT'
t2 = (file2, id2, col2)
compare(t1, t2)
```

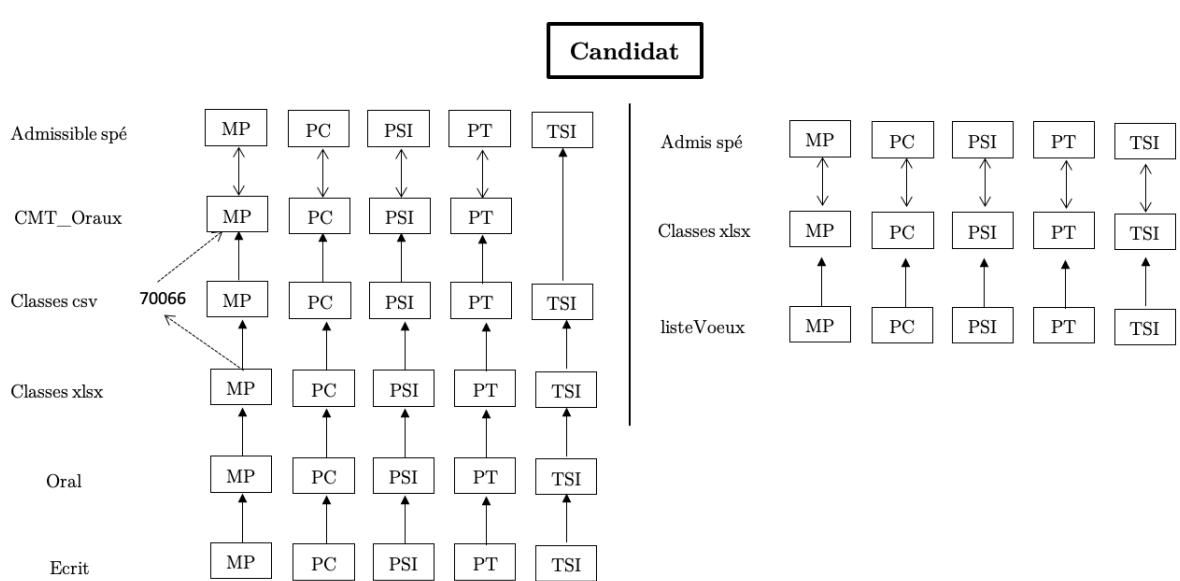
Ici, compare(t1, t2) renvoie vrai si dans les fichiers `listeEtablissements.xlsx` et `Inscription.xlsx`, chaque établissement possède le même code candidat.

Nous avons confronté les candidats et les notes, à l'aide de `contient` et `compare`. Les schémas ci-dessous 4.1, résument l'inclusion des informations dans les fichiers.



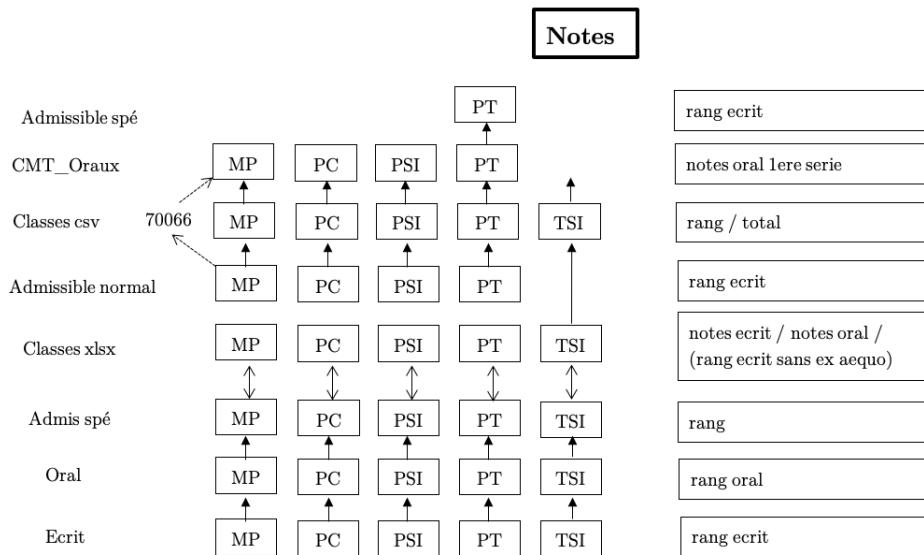
Candidat 1/2

FIGURE 4.1 – Inclusions des candidats dans les filières 1/2



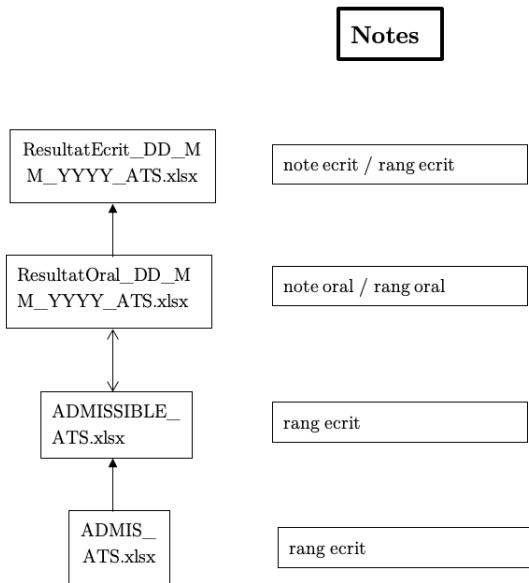
Candidat 2/2

FIGURE 4.2 – Inclusions des candidats dans les filières 2/2



Notes 1/2

FIGURE 4.3 – Inclusions des notes dans les filières 1/2



Notes 2/2

FIGURE 4.4 – Inclusions des candidats dans les filières 2/2

Les flèches indiquent les inclusions des candidats. Dans la figure 4.1, les candidats du fichier `admis_normal` sont tous en `admis_spé` et `admissible_normal`. De même, les candidats du fichier `admis_spé` sont tous en `admissible_spé`. Tous les candidats hors ATS sont dans le fichier `inscription.xlsx` sauf le candidat 44232.

Cette étude nous permettra de déterminer les fichiers à parcourir pour obtenir l'information sur les

candidats.

4.2 Schéma relationnel

Nous avons décidé de mettre un maximum d'informations dans la base de données.

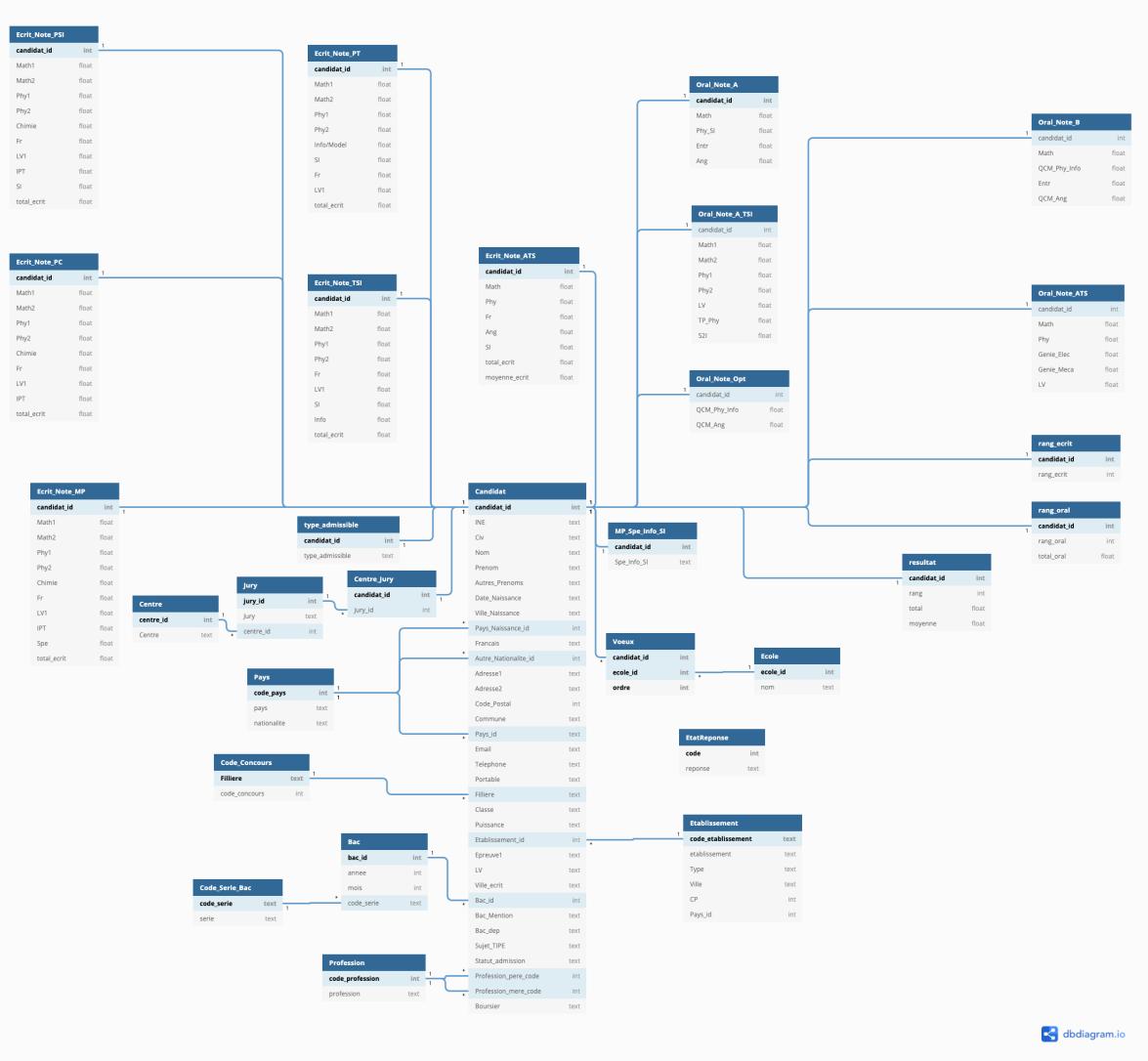


FIGURE 4.5 – Schéma relationnel de la base de données

Nous avons choisi de mettre une table pour les épreuves de chaque filière. En effet, les candidats passent des épreuves différentes en fonction de leur filière. De plus, les épreuves de mêmes matières sont différentes entre filières. L'épreuve Math1 en MP est différente de l'épreuve Math1 en PSI.

Ensuite, nous avons opté pour différentes tables par série d'épreuves orales. Les épreuves orales sont différentes pour les MP, PC, PSI, PT de type admissible A (**Oral_Note_A**) et B (**Oral_Note_B**). Les TSI de type admissible A passent les épreuves de la table **Oral_Note_A_TSI** et les type admissible B passent les mêmes épreuves que les filières précédentes (**Oral_Note_B**). Enfin, les ATS passent les épreuves de

la table `Oral_Note_ATS`. D'autre part, certains candidats de type admissible A ont passé deux épreuves supplémentaires. Les notes obtenues par ces candidats sont placées dans la table `Oral_Note_Opt`.

Nous avons aussi construit de nouvelles tables pour les renseignements souvent répétés. Ainsi, les informations pour le centre d'examen, le jury, les pays, les professions et le baccalauréat ne sont indiquées qu'une fois puis repérées par un code.

4.2.1 Contraintes d'intégrités

Nous avons indiqué différentes contraintes d'intégrités, visibles dans le fichier `database.sql`. Ces contraintes se composent des clés primaires (`PRIMARY KEY`, des clés étrangères (`FOREIGN KEY`) liant les tables et non nul (`NOT NULL`) pour imposer une valeur à l'attribut.

4.2.2 Troisième forme normale

La base de données doit être à la troisième forme normale. C'est à dire que l'on ne doit pas pouvoir déduire la valeur d'un attribut à partir des autres. C'est ce qui a été fait dans un premier temps. Cependant, il est très pratique d'avoir un attribut qui puisse être déduit des autres. C'est le cas dans les tables `Ecrit_Note_XXX`. Ces tables comprennent deux attributs, `Total`, `Moyenne`. Il serait beaucoup trop long de devoir retrouver ces valeurs là en prenant en compte les coefficients qui ne sont pas les mêmes pour toutes les épreuves des différentes filières.

4.3 Indication colonne-table

Nous avons réalisé un fichier `txt` indiquant le nom de toutes les colonnes de chaque fichier `xlsx` et `csv`. Ce fichier est créé à l'aide du script `ecriture_file.py`. Ensuite, pour chaque colonne, l'équipe a indiqué à côté l'attribut de la base de données permettant de retrouver l'information de la colonne.

4.4 Crédit de la base de données

La structure de la base de données est écrites dans le fichier `database.sql`. La base de données est ensuite générée à l'aide du script `DB_generator.py`. L'importation des données se fait grâce aux scripts `DB_insertion.py` et `DB_insertion_particulier.py`. Le script `create_table.sh` permet de générer et remplir la base de données.

Chapitre 5

Conception du site web

5.1 Présentation générale du site et des choix

5.1.1 Choix de conception

Pour ce site internet, nous voulions une interface qui allait sur tous les appareils car, de nos jours, un site internet doit obligatoirement avoir une interface pour téléphone. Ce qui est encore plus vrai pour un site qui devrait être consulté par des étudiants. Selon médiamétrie [11], un internaute sur trois se connecte à un site exclusivement depuis son téléphone, et 60% des Français utilisent leur téléphone au moins un fois par jour pour aller sur un site internet.

Ainsi, nous voulions faire un site internet à l'allure moderne adapté pour tous les appareils. Dans cette optique, il était nécessaire que le site internet soit *responsive*. C'est-à-dire que l'affichage du site internet puisse se faire automatiquement peu importe la taille et le format de l'écran. De plus, le site internet devait avoir un style épuré.

Pour faire ce site de manière *responsive*, nous avons décidé d'utiliser *Bootstrap*[1], c'est un framework CSS développé par Twitter. Il est parmi les plus utilisés au monde. Il permet de faire des sites avec un affichage épuré et facilement modulable. Alors, nous avons choisi de prendre un template de site internet qui utilisait *Bootstrap* et de l'adapter à nos besoins. Nous avons juste gardé le header du site et le footer.

Il est possible que l'affichage ait un peu changé entre l'écriture du rapport et la présentation du site internet lors de la soutenance.

Pour lancer le site internet, il faut lancer ces trois scripts dans cet ordre :

1. `init.sh`
2. `create_table.sh`
3. `run.sh`

Pour l'exécution du site internet, il faut lancer le script `run.sh`. Ce script se charge d'ouvrir l'environnement virtuel créé par le premier script. Puis, il lance le site internet.

5.1.2 Fonctionnement

Toute la partie serveur du site internet, partie dite de *back-end* a été écrit en python en utilisant le framework *Flask*. C'est un framework simple d'utilisation et facilement modulable. Nous l'avons choisi, même si la question ne s'est pas posée longtemps, car nous avions déjà de l'expérience avec cet outil grâce aux cours de Web BD.

Nous avons séparé les fichiers du serveur en trois parties, le fichier principal, `app.py` gère directement tout le serveur, c'est-à-dire les requêtes faites pour une page web. Un autre fichier, `formulaire.py`, gère les formulaires du site, il contient encore les anciennes versions des formulaires car on pensait qu'ils pourraient encore servir. Enfin le dernier fichier, `requetes.py` gère toutes les requêtes vers la base de données dont la présentation se trouve dans la partie précédente.

Cette répartition permet de mieux gérer le code, et de mieux l'entretenir. C'est toujours mieux que d'avoir un fichier de plusieurs centaines de lignes qui gère tout le serveur, dans lequel on ne pourrait plus retrouver l'utilité de telle ou telle partie du code.

5.2 Présentation des différentes pages et de leur fonctionnement

Toutes les pages du site internet sont *responsive*. L'affichage de celles-ci se fera aussi bien sur ordinateur que sur téléphone. Nous ne montrons pas l'affichage des pages sur les deux appareils pour toutes les pages du site internet, car ça n'a pas beaucoup d'intérêt. L'affichage est généralement le même en fonction de l'appareil, ce sont juste les différents éléments de la page qui ne sont pas placés pareil en fonction de la taille de l'écran.

5.2.1 Page d'accueil

La page d'accueil est une page assez classique que l'on retrouve sur beaucoup de sites web. En effet, c'est la première qu'un nouvel utilisateur va voir. Elle lui donne directement des informations sur le concours Mines-Telecom. De plus, il faut que le site internet soit rapide à prendre en main. C'est pourquoi on a choisi des images simples avec une barre de navigation claire. Cette page comme toutes les pages peut s'afficher sur les ordinateurs comme les smartphones. Pour l'affichage sur téléphone, nous avons utilisé les outils de débogage de Google Chrome. Les deux images montrent bien le fait que le site est *responsive*. Cependant, il y a un petit problème d'alignement qu'on n'a pas réussi à résoudre. Le « Bienvenue » du téléphone n'est pas centré.



FIGURE 5.1 – Accueil sur ordinateur

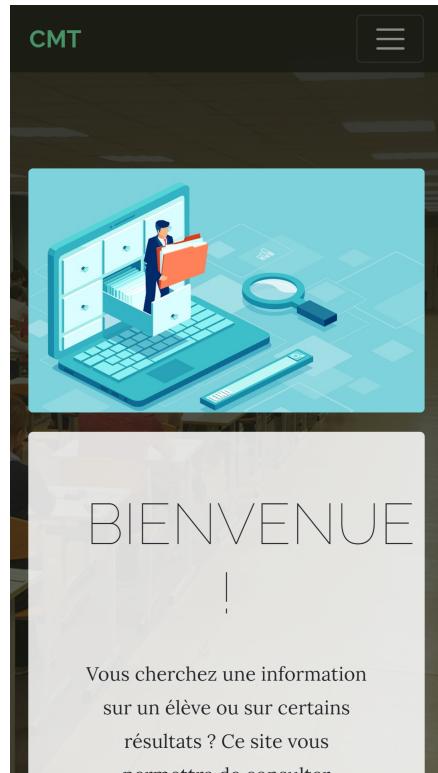


FIGURE 5.2 – Accueil sur téléphone

5.2.2 Page des résultats d'un candidat

Front-end C'est une page cruciale sur ce site internet. En effet, c'est à partir de cette page, que les candidats peuvent faire des requêtes dans la base de données et ainsi obtenir leurs résultats. Nous voulions une interface à la fois compréhensible et simple d'utilisation. Toutes les informations sont groupées par thème. On peut ainsi obtenir toutes les informations relatives à un candidat lors des épreuves, ainsi que des informations personnelles renseignées par le candidat lui-même.

À la base, nous étions partis sur une interface en trois parties. Cela permettait de faire des requêtes différentes, soit par l'ID, soit l'INE, soit avec le nom et prénom. Mais ça ne collait pas vraiment avec ce qu'on attend d'un site de consultation de données privées. Il était possible de récupérer les informations d'une autre personne uniquement avec son nom et son prénom. De plus, l'esthétique ne convenait pas à nos attentes. Nous avons donc choisi de mettre un formulaire unique, qui demanderait des informations supposées uniques. On pourrait alors afficher un seul candidat sur la page.

FIGURE 5.3 – Ancien formulaire de candidat

Nous avons décidé de le remplacer par un formulaire moins permisif et plus esthétique.

The screenshot shows a search interface titled "Recherche". It contains four input fields: "INE du candidat", "Nom du candidat", and "Prénom du candidat", followed by a "Rechercher" button.

FIGURE 5.4 – Nouveau formulaire de candidat

Nous avons ajouté un système de vérification de entrées de l'utilisateur, pour éviter par exemple de tomber sur un candidat qui s'appelle Robert') ;DROP TABLE Candidat;--.

Cette page affiche toutes les informations que le candidat a renseigné lors de son inscription. Ces données sont très diverses, elles vont de l'identité du candidat à son lieu d'examen et son ancien lycée. Enfin, la partie la plus importante est la partie des notes qui se trouve à la fin de la page. Cette partie montre de manière exhaustive toutes les notes des écrits et oraux. Enfin, on peut voir le rang de la personne ainsi que son admissibilité. Il est aussi possible de consulter les voeux qui ont été formulé. L'affichage se fait dans l'ordre dans lequel les informations ont été rentrées par le candidat puis par les personnes chargées des notes. On a d'abord les informations privées, c'est-à-dire, ses coordonnées. Ensuite, la scolarité de la personne, ses voeux, son centre d'examen, et toutes ses notes.

L'affichage se fait avec une classe de **Bootstrap**. Ce sont des *cards*, elles permettent un affichage simple des informations, tout en faisant contraste avec le reste de la page.

The screenshot shows a card titled "Coordonnées" containing a list of personal information items:

- ID : 53621
- INE : 0833465573C
- Mr. DIAS Gérard
- Né le 05/11/1999 à Calais, Maroc
- Français : non
- Autre nationalité : Tunisie
- Adresse n°1 : 806, chemin Maurice, 45569, Jacquotboeuf, Maroc
- Email : dorotheebuisson@charles.com
- Téléphone fixe : +33 (0)8 06 39 06 32
- Téléphone portable : 05 94 40 87 21
- Catégorie socio-professionnelle du père : Ingénieurs et cadres techniques d'entreprise
- Catégorie socio-professionnelle de la mère : Professions libérales

FIGURE 5.5 – Carte de coordonnées d'un candidat

Une fois les notes consultées, il est possible pour le candidat de télécharger le rapport. Pour ça, le site web ne fait aucune requête au serveur. Un code JavaScript va créer un fichier que l'utilisateur va pouvoir recevoir. Ce n'est pas du téléchargement, car toutes les informations sont déjà présentes sur la page web. Ce code va créer un bouton de téléchargement avec un tag et simuler un click sur celui-ci et le supprimer instantanément. On reçoit alors le fichier.

Lors de nos tests, nous avons remarqué quelques bugs d'affichage lorsqu'il y avait des données manquantes. Par exemple, si la base ne comprenait pas de voeux pour un candidat. Dans ce cas le site internet affiche une erreur. On a supposé que cette erreur du fait que certains candidats n'avait pas fait de voeux pour quelque raison , voire que ce soit à cause de l'anonymisation de la base de données.

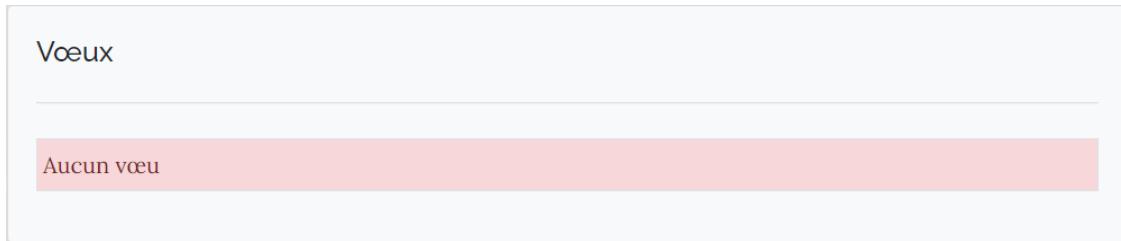


FIGURE 5.6 – Aucun voeux

Back-end Les informations du candidat sont passées par une méthode `GET` au serveur grâce à des paramètres dans l’URL. Ces informations sont ensuite reçues par le serveur qui va les traiter. Il faut que toutes les informations soient présentes pour que le serveur fasse les requêtes sur la base. Pour les requêtes, nous avons décidé d’utiliser `SQLite3`. Le serveur va faire plusieurs requêtes distinctes. Il en fait pour les coordonnées, la scolarité, etc. Enfin, il va tout retourner sous forme de listes qui vont être transmises à `Jinja2` via le template.

Le template de cette page était originellement conçu pour pouvoir afficher plusieurs candidats. Puisque l’on pouvait faire des requêtes uniquement pour le nom et prénom, mais cette fonctionnalité n’est plus utilisable, car le formulaire demande maintenant trois informations plutôt que deux.

5.2.3 Page des résultats globaux

Front-end Cette page affiche simplement tous les résultats par filière. Il n’y a aucune interaction possible de l’utilisateur à part la génération d’un rapport de ces informations et le changement de la page. Le design de cette page est comme la page sur les candidats. Elle fonctionne avec des *cards*.

Admis par filière		
Filière	Nombre de candidat	Pourcentage d’admis
ATS	139	92.05
MP	3169	62.18
PC	2087	60.26
PSI	2584	60.1
PT	619	35.37
TSI	217	30.86
Total	8815	57.01

FIGURE 5.7 – Admissibilité par filière

Back-end Le serveur va générer tous les pourcentages que l’on peut voir sur le site internet, sinon il n’y a pas beaucoup de différences par rapport à la page précédente.

5.2.4 Page Statistiques

Front-end Cette page affiche une liste déroulante contenant tous les examens du concours Mines-Télécom et un bouton "Rechercher" pour valider noter choix dans la liste déroulante.

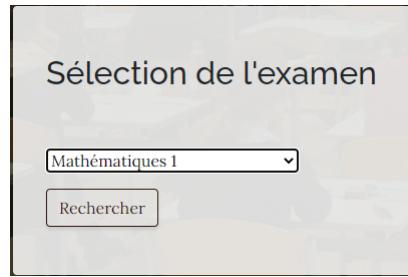


FIGURE 5.8 – Recherche d’informations sur un examen

Back-end Le serveur va générer les informations à partir des notes de la base de données et de la matière sélectionnée dans le menu déroulant.

5.2.5 Page A propos

Front-end Cette page affiche simplement des informations pour la réalisation du site internet. C'est-à-dire les membres du groupe et l'enjeu du site internet.

Back-end Il n'y a rien de spécial ici, le serveur renvoie simplement une page HTML à partir d'un template, mais sans rien modifier.

Chapitre 6

Conclusion

6.1 Bilan global du projet

Travail attendu	Travail réalisé
<ul style="list-style-type: none">- Concevoir un schéma de base de données relationnelle stockant des données du concours Mines-Telecom- Implémenter ce schéma relationnel dans un système de gestion de base de données (SGBD)- Réaliser plusieurs composants logiciels informatiques dans les langages de programmation C, Java et/ou Python- Réaliser une étude bibliographique- Développer des scripts d'importation de fichiers de données de différents formats (.csv, .xlsx).- réaliser un certain nombre de requêtes SQL démontrant la viabilité du modèle proposé- Concevoir une application permettant de consulter et d'interroger les données de la base- Construire des visualisations- Générer des rapports au format HTML	<ul style="list-style-type: none">- Concevoir un schéma de base de données relationnelle stockant des données du concours Mines-Telecom- Implémenter ce schéma relationnel dans un système de gestion de base de données (SGBD)- Réaliser plusieurs composants logiciels informatiques dans les langages de programmation C, Java et/ou Python- Réaliser une étude bibliographique- Développer des scripts d'importation de fichiers de données de différents formats (.csv, .xlsx).- réaliser un certain nombre de requêtes SQL démontrant la viabilité du modèle proposé- Concevoir une application permettant de consulter et d'interroger les données de la base- Construire des visualisations- Générer des rapports au format HTML

	Points positifs	Points négatifs	Expérience
Gestion de projet	<ul style="list-style-type: none">- Bonne gestion du Git (commit réguliers)- Bonne entente du groupe	<ul style="list-style-type: none">- Objectifs parfois trop vague	<ul style="list-style-type: none">- Mise en place d'une méthode agile
Écriture du code	<ul style="list-style-type: none">- Le code fonctionne- Bonne organisation des fichiers	<ul style="list-style-type: none">- Parfois peu clair	<ul style="list-style-type: none">- Découverte du module pandas- Première réalisation de site complet avec visualisations

6.2 Bilan du projet membre par membre

6.2.1 Hamza Abdoulhoussen

Points positifs	<ul style="list-style-type: none"> - Réactivité du groupe - Mise en pratique de la gestion de projet - Bonne organisation du dépôt Git
Difficultés rencontrées	<ul style="list-style-type: none"> - Objectifs souvent trop vague
Expérience personnelle	<ul style="list-style-type: none"> - Renseignement sur l'utilisation de pandas pour analyser des données - Découverte de visuels pour un site
Axes d'amélioration	<ul style="list-style-type: none"> - Tester le travail de chacun pour déceler des omissions - Passer plus de temps à exposer les résultats lors des réunions

6.2.2 Benjamin Lavandier

Points positifs	<ul style="list-style-type: none"> - Bonne ambiance dans le groupe - Pas de conflits dans l'utilisation de Git - Répartition du travail équitable - Séance de travail en présentiel ensemble qui permet un travail plus efficace
Difficultés rencontrées	<ul style="list-style-type: none"> - Lancement du projet un peu lent car nous ne savions pas vraiment ce que signifiaient toutes les informations
Expérience personnelle	<ul style="list-style-type: none"> - Approfondissement des connaissances en Flask et html/css - Création d'une autre base de données
Axes d'amélioration	<ul style="list-style-type: none"> - Tester le travail de chacun pour déceler des omissions

6.2.3 Lucas Gouin

Points positifs	<ul style="list-style-type: none"> - On s'entend bien donc on a pu travailler facilement ensemble - Personne n'a cassé le dépôt Git ! - J'ai bien aimé travailler sur un site internet.
Difficultés rencontrées	<ul style="list-style-type: none"> - Gestion de bugs parfois incompréhensible avec SQLite ou Flask - Etablissement d'objectifs clairs
Expérience personnelle	<ul style="list-style-type: none"> - Amélioration des connaissances en développement Web - Amélioration de l'aisance à travailler en groupe
Axes d'amélioration	<ul style="list-style-type: none"> - Mieux exposer son travail au reste du groupe - Fixer des objectifs clairs

6.3 Travail réalisé

Etapes	Hamza	Lucas	Benjamin
Gestion de Projet	3h	5h	3h
Réunions	4h	4h	4h
Code	25h	25h	20h
Rapport	8h	6h	10h
Etat de l'art	4h	2h	6h
Total	44h	42h	43h

Annexe A

Comptes-Rendus

A.1 Compte-Rendu du 04/04/2021

Durée : 34min

Ordre du jour :

- Présentation du groupe
- Attribution des rôles
- Discussion sur le sujet
- Répartition des tâches

Présentation du groupe

Lucas Gouin, prépa PT

Hamza Abdoulhoussen, Prépa MP

Benjamin Lavandier, DUT Informatique

Attribution des rôles

Lucas Gouin, chef de groupe

Benjamin Lavandier, secrétaire

Discussion sur le sujet

Requis pour la création de la BDD :

1. La conception d'un schéma relationnel pour intégrer les données mentionnées précédemment
 - ce schéma devra dans un premier temps être en 3ème forme normale, vous pourrez dénormaliser ce schéma si nécessaire en argumentant sur ce point
 - vous identifierez les contraintes d'intégrité du modèle de données
2. La transposition de ce modèle dans une base de données relationnelle sur un SGBD (SQLite, PostgreSQL, MySQL, MariaDB)
3. L'importation (automatisée) des données dans la base proposée (un nettoyage des données sera peut-être nécessaire, une vérification de la cohérence des données sera obligatoire).
4. La réalisation d'un certain nombre de requêtes SQL démontrant la viabilité du modèle proposé.

5. Documenterez la procédure pour installer/exécuter/initialiser votre base de données

Répartition des tâches

Chacun doit faire son brouillon du schéma relationnel pour Mardi 06/04/2021 à 16h (prochaine réunion)

A.2 Compte-Rendu du 06/04/2021

Durée : 55min

Ordre du jour :

- Discussion du schéma relationnel
- Gestion de projet
- Répartition des tâches suivantes

Discussion du schéma relationnel

Questions à poser :

- à quoi sert la colonne voe_ran dans le fichier de la liste des vœux exactement ?
- les notes max_physique, max_anglais, bonification des résultats représentent quoi ?
- le fichier listeEtatsReponsesAppels se relie avec quoi ?

Après avoir échanger les idées de nos brouillons, nous avons décidé de nous baser sur celui de Hamza qui l'a fait sur le site dbdiagram.io puis de compléter les parties manquantes.

Gestion de projet

Matrice SWOT à faire

Potentiellement une todolist plus tard

Répartition des tâches suivantes

En gestion de projet :

- Lucas fait la matrice SWOT

Pour le schéma relationnel :

- Benjamin s'occupe de la partie Admissions et Admis
- Lucas s'occupe de la partie Vœux (clé primaire avec codecandidat et codeecole)
- Hamza s'occupe de la partie Etablissements et Ecole

Tout le monde commence à réfléchir aux fonctions de traitements des données.

Date de la prochaine réunion :

Mardi 13/04 à 16h

A.3 Compte-Rendu du 13/04/2021

Durée : 35min

Ordre du jour :

- Discussion de l'avancement de chacun

- Retour sur les questions
- Répartition des tâches suivantes
- Choix de la date de la prochaine réunion

Discussion de l'avancement de chacun

Lucas :

- Réalisation de la matrice SWOT
- Ajout des voeux dans le schéma
- A commencé à avoir des réponses de Sébastien Da Silva

Hamza :

- A rajouté les écoles et les notes des ATS sur le schéma

Benjamin :

- A ajouté toute la partie admission, admis et admis_spe

Tout le monde :

- idées de fonctions de traitement trouvées (voir fichier dans le dossier GDP)

Après discussion avec le groupe, l'idée de Benjamin de regrouper les différentes filières dans une tables pour la partie sur laquelle il a travaillé semble bonne et sera faite par la suite.

Questions à poser restantes :

- à quoi sert la colonne voe_ran dans le fichier de la liste des voeux exactement ?
- les notes max_physique, max_anglais, bonification des résultats représentent quoi ?
- le fichier listeEtatsReponsesAppels se relie avec quoi ?
- que représente les fichiers admis_spe par rapport aux fichiers admis ?

Retour sur les questions

Lucas a demandé à Sébastien Da Silva mais il n'a répondu qu'à une question et encore il n'est pas sûr. Il lui demandera demain en TD pour le reste car avant SDS était pressé.

Répartition des tâches suivantes

Benjamin :

- appliquer son idée pour simplifier une partie du schéma relationnel

Lucas :

- poser les questions

Tout le monde :

- commencer à concevoir les fonctions de traitement

Date de la prochaine réunion :

Lundi 26/04 à 16h

A.4 Compte-Rendu du 27/04/2021

Durée : 25min

Ordre du jour :

- Discussion de l'avancement de chacun
- Retour sur les questions
- Répartition des tâches suivantes
- Choix de la date de la prochaine réunion

Discussion de l'avancement de chacun

Lucas :

- Travail sur plusieurs fonctions et optimisation du schéma relationnel

Hamza :

- Travail sur plusieurs fonctions

Benjamin :

- Simplification du schéma relationnel

Questions à poser restantes :

- à quoi sert la colonne voe_ran dans le fichier de la liste des vœux exactement ?
- les notes max_physique, max_anglais, bonification des résultats représentent quoi ?
- le fichier listeEtatsReponsesAppels se relie avec quoi ?
- que représente les fichiers admis_spe par rapport aux fichiers admis ?

Retour sur les questions

Lucas a demandé à Sébastien Da Silva en cours. Ce dernier lui a dit de renvoyer un mail, ce qu'il vient de faire aujourd'hui.

Répartition des tâches suivantes

Hamza :

- Commencer à intégrer les données dans une base

Benjamin et Lucas :

- Commencer le développement du site web

Date de la prochaine réunion :

Mardi 04/05 à 16h

A.5 Compte-Rendu du 05/05/2021

Durée : 9min

Ordre du jour :

- Discussion de l'avancement de chacun

- Retour sur les questions
- Répartition des tâches suivantes
- Choix de la date de la prochaine réunion

Discussion de l'avancement de chacun

Lucas :

- Début du back-end

Hamza :

- Travail sur les données pour les rentrer dans une base et les lier au site

Benjamin :

- Interface du site commencée (page d'accueil, des candidats, des filières avec admis et admissions)

Tout le monde :

Questions à poser restantes :

- à quoi sert la colonne voe_ran dans le fichier de la liste des vœux exactement ?
- les notes max_physique, max_anglais, bonification des résultats représentent quoi ?
- le fichier listeEtatsReponsesAppels se relie avec quoi ?
- que représente les fichiers admis_spe par rapport aux fichiers admis ?

Retour sur les questions

Toujours rien

Répartition des tâches suivantes

On se retrouve à TN pour avancer ensemble sur le projet dès demain à 14h.

Date de la prochaine réunion :

Mardi 12/05/2021 à 10h

A.6 Compte-Rendu du 13/05/2021

Benjamin était absent

Durée : 9min

Ordre du jour :

- Discussion de l'avancement de chacun
- Retour sur les questions
- Choix de la date de la prochaine réunion

Discussion de l'avancement de chacun

Tout le monde :
Progression sur le remplissage de la base données pour Hamza et de l'affichage des données sur le site pour Lucas

- Questions à poser restantes :
- à quoi sert la colonne voe_ran dans le fichier de la liste des vœux exactement ?
 - les notes max_physique, max_anglais, bonification des résultats représentent quoi ?
 - le fichier listeEtatsReponsesAppels se relie avec quoi ?
 - que représente les fichiers admis_spe par rapport aux fichiers admis ?

Retour sur les questions
Toujours rien

Date de la prochaine réunion :
Samedi 22/05/2021 à 16h

A.7 Compte-Rendu du 22/05/2021

Durée : 23min

- Ordre du jour :**
- Discussion de l'avancement de chacun
 - Répartition des tâches suivantes
 - Choix de la date de la prochaine réunion

Discussion de l'avancement de chacun

Lucas : Avancée sur l'onglet candidats

Hamza : Avancée sur le remplissage de la base de données toujours

Benjamin : Avancée sur le rapport

Répartition des tâches suivantes

Lucas :
Continuer l'onglet candidats

Hamza :
Finir le remplissage de la base de données

Benjamin :
Faire l'onglet statistiques du site web

Date de la prochaine réunion :
Samedi 29/05/2021 à 16h

A.8 Compte-Rendu du 30/05/2021

Durée : 25min

Ordre du jour :

- Discussion de l'avancement de chacun
- Répartition des tâches suivantes
- Choix de la date de la prochaine réunion

Discussion de l'avancement de chacun

Lucas :

- Affichages des vœux et notes de chaque candidat qui en disposent.
- Ajout d'une fonctionnalité qui permet de générer un rapport en html pour le résultat d'une recherche de candidat sur le site.

Hamza :

- Comparaisons des données présentées dans les fichiers xlsx et csv notamment sur les différents rangs présent.

Benjamin :

- Début de l'affichage des statistiques sur les résultats des concours

Répartition des tâches suivantes

Lucas :

- Continuer l'affichage des informations sur les candidats (notes à l'oral, admission)

Hamza :

- Finir d'insérer les dernières éléments dans la base de données (centre, jury, les différents rangs)

Benjamin :

- Finir d'afficher les dernières statistiques

Tout le monde :

- Avancer sur le rapport en rédigeant les parties du projet que l'on a faites

Date de la prochaine réunion :

Séance de travail en commun

Jeudi 03/06/2021 à 14h

A.9 Compte-Rendu du 04/06/2021

Durée : 25min

Ordre du jour :

- Discussion de l'avancement de chacun
- Répartition des tâches suivantes

- Choix de la date de la prochaine réunion

Discussion de l'avancement de chacun

Lucas :

- Fin de l'affichage des voeux et notes de chaque candidats qui en disposent.

Hamza :

- Fin du travail sur la gestion de la base de données

Benjamin :

- Fin de l'affichage des statistiques

Répartition des tâches suivantes

Tout le monde :

- Avancer sur le rapport en rédigeant les parties du projet que l'on a faites
- Corriger des détails/erreurs restants
- Réflexion sur la 2ème partie pour la répartition du travail

Date de la prochaine réunion :

Mercredi 09/06/2021 à 19h

Bibliographie

- [1] Business Casual. Business Casual - Free Template Bootstrap 5. <https://startbootstrap.com/theme/business-casual>. 2021.
- [2] Concours Mines-Télécom. A propos du concours Mines-Télécom. <https://www.concours-mines-telecom.fr/a-propos/>. 09.06.2021.
- [3] designmodo. Bootstrap 5 What's New About It and Release Date. <https://designmodo.com/bootstrap-5/>. 2021.
- [4] Institut Mines-Télécom - Wikipédia. Institut Mines-Télécom - Wikipédia. https://fr.wikipedia.org/wiki/Institut_Mines-T%C3%A9l%C3%A9com. 18.12.2020.
- [5] MDN Web Docs. CSS : Feuilles de style en cascade. <https://developer.mozilla.org/fr/docs/Web/CSS>. 2021.
- [6] MDN Web Docs. HTML (HyperText Markup Language). <https://developer.mozilla.org/fr/docs/Web/HTML>. 2021.
- [7] MDN Web Docs. JavaScript. <https://developer.mozilla.org/fr/docs/Web/JavaScript>. 2021.
- [8] MyScenari. Deuxième forme normale. <https://stph.scenari-community.org/bdd/0/co/norUC024.html>. 2021.
- [9] MyScenari. Première forme normale. <https://stph.scenari-community.org/bdd/0/co/norUC023.html>. 2021.
- [10] MyScenari. Troisième forme normale. <https://stph.scenari-community.org/bdd/0/co/norUC025.html>. 2021.
- [11] Médiamétrie. 1 internaute sur 3 se connecte à internet exclusivement sur smartphone. <https://www.mediametrie.fr/fr/1-internaute-sur-3-se-connecte-internet-exclusivement-sur-smartphone>. 16.12.2020.