Projet CVL

Site pour le **C**onseil de la **V**ie **L**ycéenne

**BTS-SIO** **2 Option SLAM** 2017

SAPHAR-CADET Marie

VUE Pierre

LESCENE Laurent

Table des matières

[I. Présentation du sujet 3](#_Toc503730133)

[1. Contraintes techniques 3](#_Toc503730134)

[2. Fonctionnalités attendues 3](#_Toc503730135)

[a) Organisation des élections : 3](#_Toc503730136)

[b) Le votant : 3](#_Toc503730137)

[c) Interlocuteur : 4](#_Toc503730138)

[3. Fonctionnalités présentes 4](#_Toc503730139)

[a) La connexion : 4](#_Toc503730140)

[b) Pour les élèves : 4](#_Toc503730141)

[c) Pour les techniciens : 4](#_Toc503730142)

[d) Pour les CPE : 4](#_Toc503730143)

[II. Organisation 5](#_Toc503730144)

[1. Répartition des tâches 5](#_Toc503730145)

[2. Méthodes 5](#_Toc503730146)

[III. La base de donnée 6](#_Toc503730147)

[1. Les besoins 6](#_Toc503730148)

[Nous avons eu besoin : 6](#_Toc503730149)

[2. Le modèle 7](#_Toc503730150)

[Le script de la base de données : 7](#_Toc503730151)

[BDD CVL 11](#_Toc503730152)

[IV. La gestion de la connexion 12](#_Toc503730153)

[Menu de connexion : 12](#_Toc503730154)

[Savoir si l’élève a déjà chiffré sont mot de passe : 12](#_Toc503730155)

[Changement de mot de passe : 13](#_Toc503730156)

[Gestion de date de connexion : 13](#_Toc503730157)

[Différencier élève, technicien et admin : 14](#_Toc503730158)

[Déconnecter si élève a déjà voté : 14](#_Toc503730159)

[V. La gestion des administrateurs 15](#_Toc503730160)

[Après connexion du CPE : 16](#_Toc503730161)

[Après connexion du Technicien : 16](#_Toc503730162)

[VI. La gestion des élèves 21](#_Toc503730163)

[VII. Les Difficultés 21](#_Toc503730164)

[Laurent 21](#_Toc503730165)

[Marie 21](#_Toc503730166)

[Pierre 21](#_Toc503730167)

# Présentation du sujet

Chaque année, le Conseil pour la Vie Lycéenne est renouvelé, soit 5 sièges à pourvoir. Le vote se fait au suffrage universel direct. Il se fait au scrutin plurinominal à 1 tour.

Les électeurs ont possibilité de voter pour 5 personnes maximum (choix de 5 personnes). Au-delà, le vote est considéré comme nul. Ils peuvent, en revanche, ne voter que pour 1, 2, 3 ou 4 personnes. Ils peuvent également voter « blanc ».

Il faut pour cela mettre en place une application qui permet à tout lycéen et étudiant de voter électroniquement de n’importe quel poste informatique du lycée. Le vote se ferait à une date et sur un créneau horaire déterminé. De la même manière, ne pourra voter un élève qu’après son identification formelle via les paramètres de log fournis en début d’année par le lycée.

## Contraintes techniques

* La solution s’appuie sur un site Intranet, écrit en PHP. Le vote sera effectué par les électeurs grâce aux navigateurs installés sur les machines de l’établissement.
* Programmation objet.
* On peut envisager une solution DOTNET pour la gestion de l’organisation de ces élections. *Le CPE utilise une solution de ce type. Elle n’est pas à écrire.*
* La base de données sera de type MySQL.
* Les mots de passe stockés dans la base de données seront cryptés.
* L’accès à la base de données sera contrôlé via des privilèges.
* La liste des électeurs peut être obtenue auprès des techniciens par extraction du fichier des élèves et étudiants.
* En aucun cas on ne doit pouvoir connaître qui a voté pour qui.

## Fonctionnalités attendues

### Organisation des élections :

* Insérer les électeurs depuis un fichier.
* Générer ou régénérer les mots de passe. (Les mots de passe seront composés d’au moins 6 caractères (lettres et chiffres)).
* Enregistrer les candidats et les suppléants.
* Imprimer par classe la liste des électeurs. Après découpage, les identifiants de connexion seront remis par le professeur principal à chaque élève.
* Définir les dates et plages horaires d’ouverture du vote.
* Suivre la participation.
* Définir les messages d’invite à l’élection.
* Visualiser les résultats. Ceux-ci ne doivent être accessibles qu’après les élections.
* Sauvegarder les données et restaurer
* Lister, par classe, les électeurs ayant voté.

### Le votant :

* Lorsqu’il se connecte sur un ordinateur, une invite peut lui rappeler les élections. (script VBS)
* Avant de pouvoir voter, il doit impérativement s’identifier.
* Il doit confirmer son vote par la saisie d’un code de validation.
* Il doit être avisé que son vote a été enregistré.
* Si un électeur a déjà voté, sa connexion doit être refusée.
* Les modalités du scrutin doivent impérativement être visibles au moment du vote.

### Interlocuteur :

Mr Pascal Giorgi, professeur, est chargé de l’organisation de ces élections.

## Fonctionnalités présentes

### La connexion :

* Entrer l’identifiant et le mot de passe fourni par un administrateur ou le mot de passe configuré par l’élève.
* Si l’élève se connecte pour la première fois il est rediriger vers un page de changement de mot de passe, le nouveau mot de passe est chiffré en md5 dans la BDD.
* Empêcher une double connexion.
* Expiration de session à 30 min.
* Afficher erreur et changer couleur du formulaire s’il y a une erreur.

### Pour les élèves :

### Pour les techniciens :

* Importer la liste des élèves dans la base de données.
* Exporter les Login des élèves classés par division.
* Exporter les résultats d’un suffrage donné.
* Ajouter une division.
* Supprimer une division.
* Ajouter un élève.

### Pour les CPE :

* Ajouter un suffrage.
* Ajouter un candidat dans un suffrage donné.

# Organisation

## Répartition des tâches

Légende: Marie  Pierre  Laurent

1. Importer le fichier de la liste des élèves
2. Pages d’accueil avec description de l'événement, connexion (vérification session du cpe ou élève) (bouton connexion ou déconnexion suivant).
3. Vérification de la double connexion pour les comptes
4. mot de passe change obligatoirement à la première connexion (crypter le mot de passe)
5. Page 1 et 2 CPE création du suffrage : sujet, date début date fin, pages 2 => inscription des candidats binômes et suppléants et détails de leurs professions.
6. Page 3 confirmation, retour vers l’accueil.
7. Ajout d’une page Technicien pour gérer les exports, ajout, suppression des divisions et ajout possible d’un élève arrivé en cours d’année ?
8. Facultatif : Gérer la panne pendant l’évènement
9. A chaque log pop-up  aujourd’hui se déroule l’élection.
10. Page 1 « voter maintenant », « voter plus tard » => retour vers accueil mais la prochaine fois encore pop-up !!
11. Vue connexion étudiant/CPE/technicien
12. Page 2 Explication des modalités de vote, liste des candidats, puce pour voter blanc, puces devant les candidats (avec leur profession). (choix max 5 personnes, si plus alors le vote est nul (pop-up !).
13. Page 3 confirmation d’enregistrement retour vers l’accueil
14. Affichage sur l’accueil et l’exportation : le taux de participation + autre statistique (par section) et résultat par candidat.

* Pouvoir extraire la liste des candidats élus (voir BDD clé étrangère)
* Générer ou régénérer les mots de passes (6 caractères).
* Pouvoir extraire par classe la liste des élèves avec leur mot de passe.
* SI déjà voté alors sa connexion est refusée !
* Mettre le taux de participation dans la page d’accueil
* L’élève valide son vote par code de validation ! api recaptcha Google + code pour confirmation avec récapitulatif.

## Méthodes

Nous avons utilisé Git hub afin de gérer les différents ajouts que ce soit en salle de CFA ou chez nous. Dans le dossier Git Hub il n’y a qu’une branche « master ».

En classe :

* Un poste possède la base donnée CVL avec des occurrences de test.
* Un poste intègre le site web.
* Un poste a un accès aux sources.

En fin de séance un document sous OneDrive est mis à jour contenant les choses à faire et les choses faites. Cette méthode a été utilisée en attendant l’installation du projet Git Hub.

Les Graphes d’activités Git Hub :

Notre fréquence de commit du début du projet au 21 Décembre :



Pierre

# La base de donnée

## Les besoins

### Nous avons eu besoin :

* Du code INE « ECodeINE » dans la Table « elect » (électeur) afin de compléter les données des élèves (le code INE étant utilisé dans chaque dossier administratif lié à l’élève).
* De la clef étrangère entre la Table « elect » et la Table « candid » afin de pouvoir récupérer les informations des candidats et celles de son binôme pour l’exportation des résultats du suffrage.
* De la Table Admin afin de gérer les connexions des Techniciens et des CPE séparément des élèves. Elle contient : le « ALogin », « APwd » et le « Adroit ». Le « Adroit » permet de différencier la connexion entre un CPE et un Technicien via les deux valeurs possibles : « CPE » et « TECH ».

## Le modèle

### Le script de la base de données :

[CODE]

-- phpMyAdmin SQL Dump

-- version 4.1.4

-- http://www.phpmyadmin.net

--

-- Client : 127.0.0.1

-- Généré le : Jeu 11 Janvier 2018 à 20:14

-- Version du serveur : 5.6.15-log

-- Version de PHP : 5.4.24

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

--

-- Base de données : `cvl`

--

-- --------------------------------------------------------

-- Structure de la table `admin`

--

CREATE TABLE IF NOT EXISTS `admin` (

`AId` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`ALogin` varchar(20) NOT NULL,

`APwd` varchar(50) NOT NULL,

`ADroit` varchar(4) NOT NULL,

PRIMARY KEY (`AId`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1 ;

-- --------------------------------------------------------

-- Structure de la table `candid`

--

CREATE TABLE IF NOT EXISTS `candid` (

`CId` varchar(10) NOT NULL,

`CIdBinome` varchar(10) NOT NULL,

`CNbV` int(5) NOT NULL DEFAULT '0',

`CIdSuffrage` int(10) unsigned NOT NULL,

PRIMARY KEY (`CId`,`CIdBinome`),

KEY `candidatSuffrage\_ibfk\_2` (`CIdSuffrage`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

-- Structure de la table `divis`

--

CREATE TABLE IF NOT EXISTS `divis` (

`DCode` varchar(10) NOT NULL,

PRIMARY KEY (`DCode`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

-- Structure de la table `elect`

--

CREATE TABLE IF NOT EXISTS `elect` (

`EId` varchar(10) NOT NULL,

`ENom` varchar(30) NOT NULL,

`EPrenom` varchar(30) NOT NULL,

`ECodeINE` varchar(11) DEFAULT NULL,

`EVote` datetime DEFAULT NULL,

`EPwd` varchar(50) NOT NULL,

`ELogin` varchar(20) NOT NULL,

`EIdDivis` varchar(10) NOT NULL,

`EDateLogin` datetime DEFAULT NULL,

`EAdresseIP` varchar(20) DEFAULT NULL,

`ELastLogin` datetime DEFAULT NULL,

`ESession` varchar(100) DEFAULT NULL,

`EDateLogout` datetime DEFAULT NULL,

`EModif` int(1) NOT NULL DEFAULT '0',

PRIMARY KEY (`EId`),

KEY `ElectDivision\_ibfk\_1` (`EIdDivis`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

-- Structure de la table `suffrage`

--

CREATE TABLE IF NOT EXISTS `suffrage` (

`SId` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`SChoix` int(2) NOT NULL DEFAULT '5',

`SDateDeb` datetime NOT NULL,

`SDateFin` datetime NOT NULL,

`SDescription` varchar(40) NOT NULL,

`SBlancs` int(4) NOT NULL DEFAULT '0',

`SNuls` int(4) NOT NULL DEFAULT '0',

PRIMARY KEY (`SId`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1 ;

--

-- Contraintes pour les tables exportées

--

-- Contraintes pour la table `candid`

--

ALTER TABLE `candid`

ADD CONSTRAINT `candidatSuffrage\_ibfk\_2` FOREIGN KEY (`CIdSuffrage`) REFERENCES `suffrage` (`SId`),

ADD CONSTRAINT `fk\_client\_numero` FOREIGN KEY (`CId`) REFERENCES `elect` (`EId`);

--

-- Contraintes pour la table `elect`

--

ALTER TABLE `elect`

ADD CONSTRAINT `ElectDivision\_ibfk\_1` FOREIGN KEY (`EIdDivis`) REFERENCES `divis` (`DCode`);

--Les inserts pour jeux de données--

INSERT INTO admin

SET AId=1, ALogin='VBEY56', APwd='58HTGR', ADroit='TECH';

INSERT INTO admin

SET AId=2, ALogin='OLMDQ4', APwd='AKEC8V', ADroit='CPE';

INSERT INTO divis

SET DCode=`1EEC`;

INSERT INTO divis

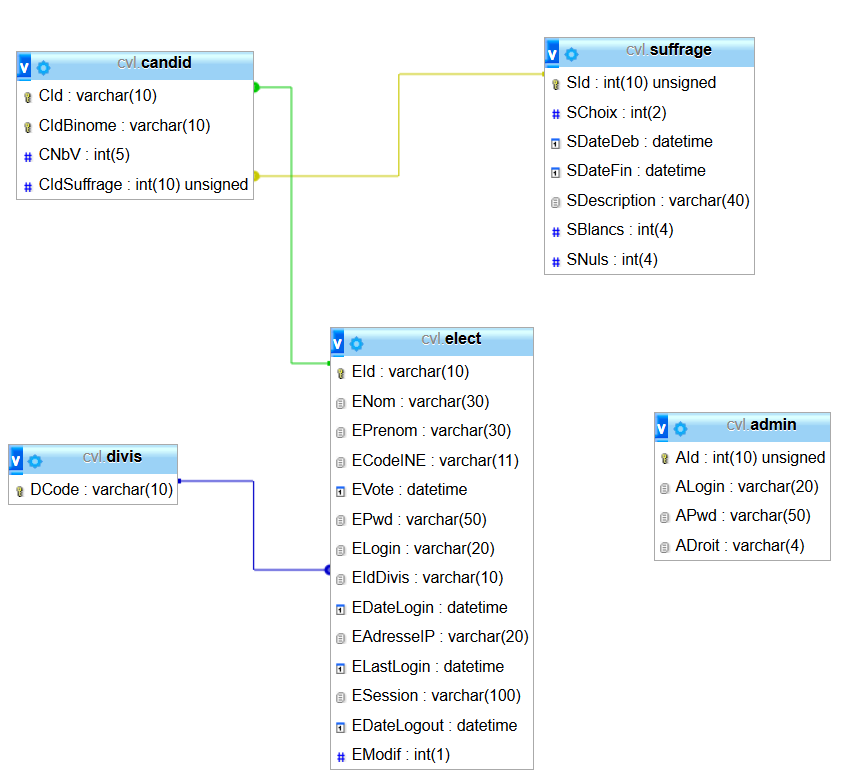
SET DCode=`PTA`;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/; [/CODE]

### BDD CVL

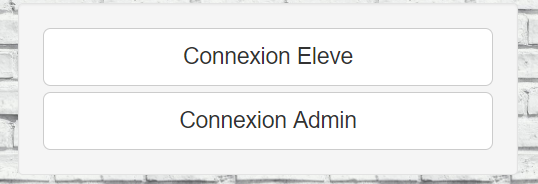


# La gestion de la connexion

## Menu de connexion :

Affiche :

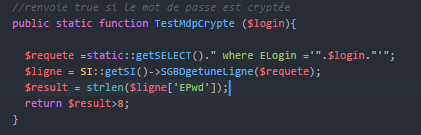
C:\Users\llesc\Desktop\Capture.PNG



Puis :

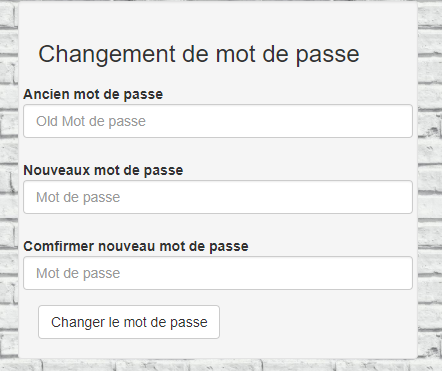
## Savoir si l’élève a déjà chiffré sont mot de passe :



Vérifier la longueur du champ pour savoir s’il est chiffré.

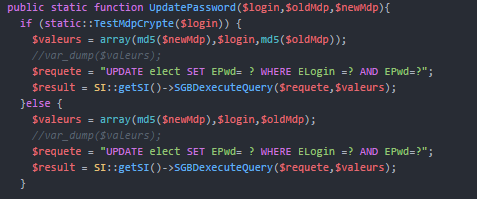
## Changement de mot de passe :

Formulaire :



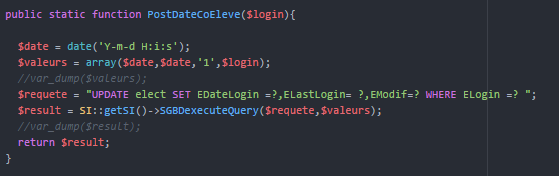
Code :

Vérifier si l’ancien mot de passe est chiffré et faire une update SQL.

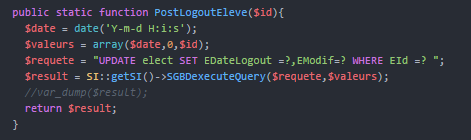


## Gestion de date de connexion :

Envoyer la date de connexion et dire que le compte est en cours d’utilisation avec le champ Emodif.



Envoyer la date de déconnexion et dire que le compte n’est plus en cours d’utilisation avec le champ Emodif.



## Différencier élève, technicien et admin :

Afficher la page des élèves si une session d’élève existe.

Afficher la page du technicien si le droit de l’admin est « TECH », s’il est « CPE » affiche la page de CPE.



## Déconnecter si élève a déjà voté :

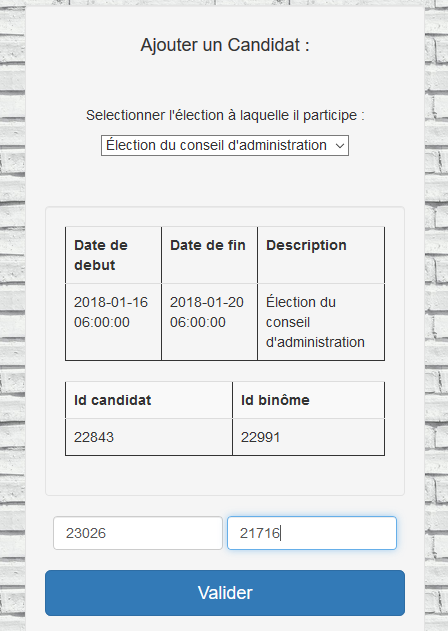
Déconnecter, détruit variable de session et poster la date de déconnection si l’élève a déjà voté ou si le bouton déconnexion a été activé.



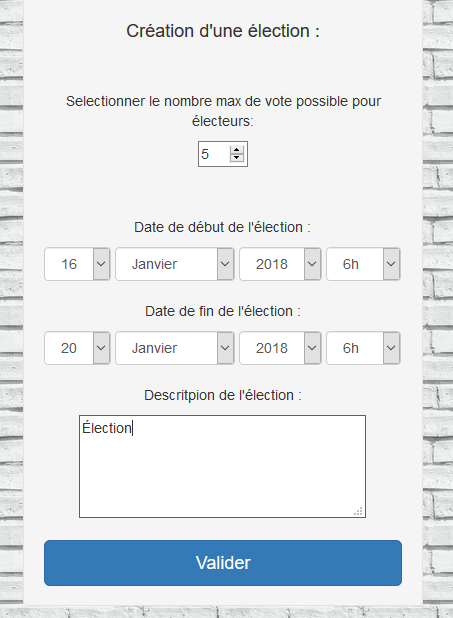
# La gestion des administrateurs

Les administrateurs ont deux droits possibles : « CPE » ou « TECH »

## Après connexion du CPE :

-Le CPE à la possibilité d’ajouter un candidat :

Il peut l’ajouter après sélection d’une élection. Il a le visuel des candidats/binôme déjà présents.



-Le CPE peut créer une nouvelle élection:

Le CPE peut modifier le nombre maximum de candidats sélectionnable par l’électeur au moment du vote. Si l’électeur dépasse le nombre c’est un vote nul.

L’élection doit commencer après la date du jour afin que les candidats y soient enregistrés avant le début des votes.

Une élection doit obligatoirement avoir une description, une date de début inférieur à la date de fin.

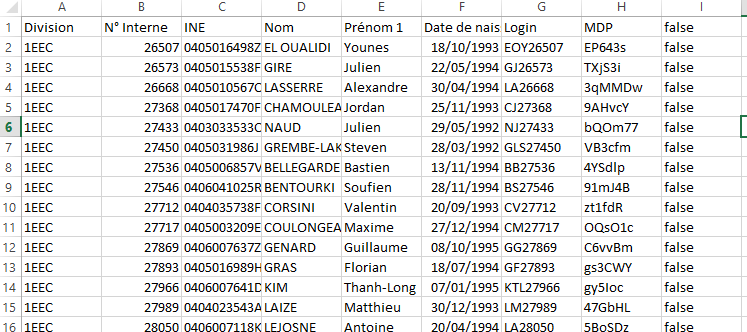
## Après connexion du Technicien :

Première partie de la page technicien :

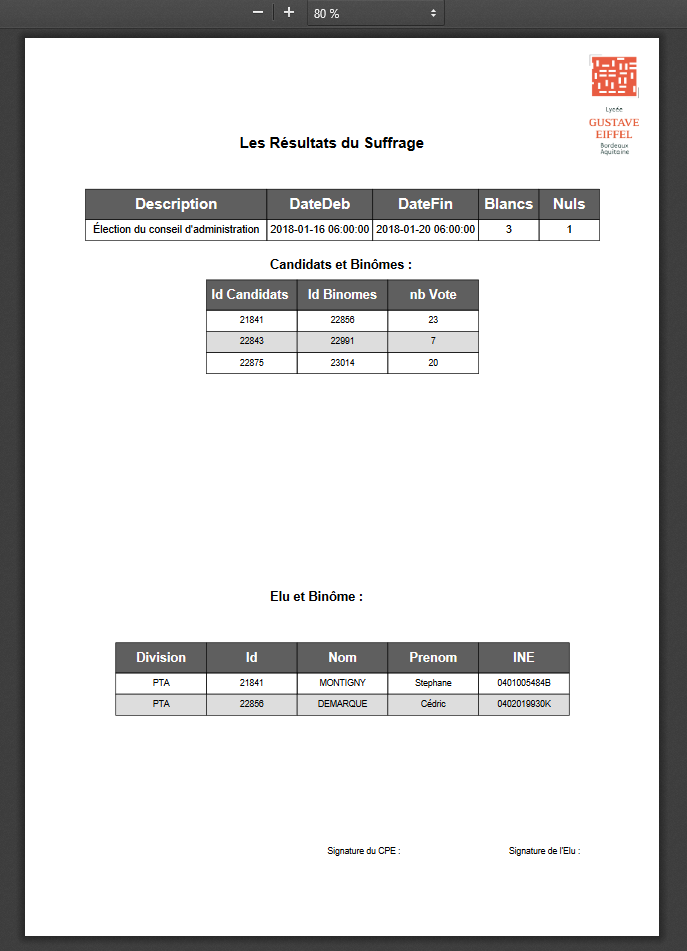


-Le Technicien peut importer la liste des élèves :

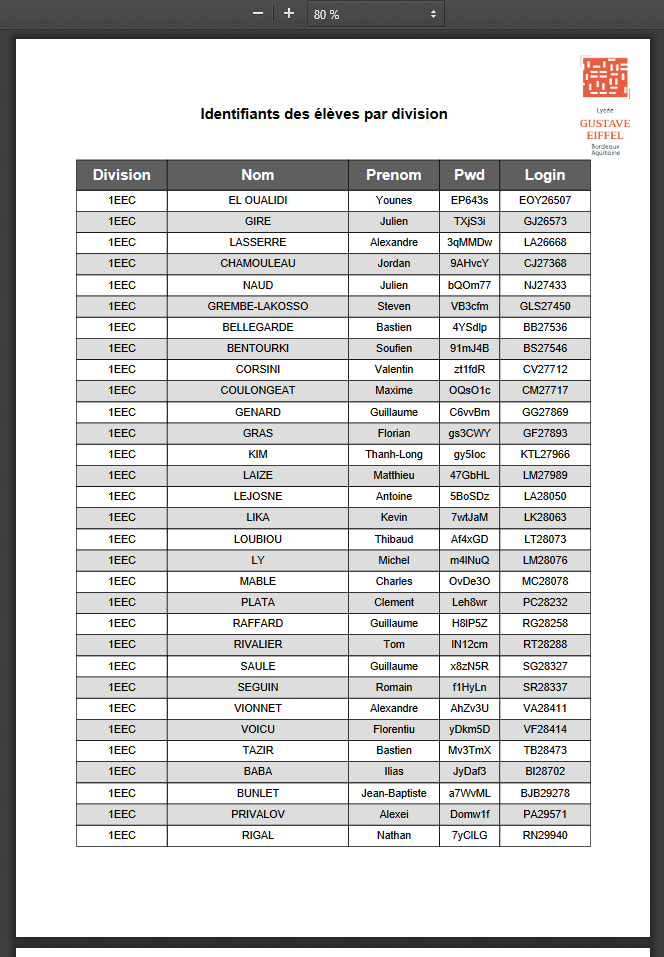
En cliquant sur « Parcourir » il navigue dans ses dossiers afin de sélectionner le fichier .CSV qui import les coordonnées de chaque élève ainsi que les logins et mots de passes.

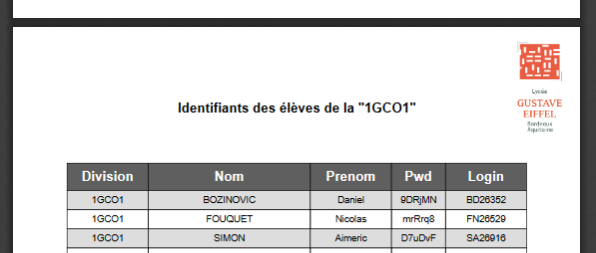
Le fichier doit comporter une dernière colonne ayant pour valeur false :

-Il peut exporter les résultats d’un suffrage :

Le Technicien sélectionne le suffrage et clique sur « Exporter », ainsi sur la nouvelle page il peut télécharger ou imprimer le fichier pdf.

-Ou/et Exporter aussi les Logins et mot de passe des élèves par classe sur un fichier pdf.





Deuxième partie de la page technicien :

-Le Technicien peut Ajouter une division :

Il a le visuel sur les divisions existantes, il peut donc ajouter une division dans le champ et cliquer sur « Valider ».

-Il peut supprimer une division :

Il sélectionne la division et clique sur « supprimer ».

-Si un élève arrive en cour d’année le technicien peut ajouter un élève :

Il saisit les informations nécessaires : id, nom, prénom… et sélectionne la division du nouvel élève puis clique sur « Valider »

# La gestion des élèves

# Les Difficultés

## Laurent

* Trouver un retour à la ligne à la fin du champ mot de passe crée lors de l’importation des élèves.

## Marie

* Difficultés dans l’ensemble sur la programmation objet (l’appel des fonctions, l’utilisation des méthodes du modèles : élément, pluriel, SI. La Navigabilité…).
* Comment récupérer les informations (nom, prénom) du candidat et son binôme.
* Les vérifications/gestions d’erreurs lors de la saisie dans un formulaire.
* Gérer le rafraichissement de la page formulaire (ne pas enregistrer en doublon).
* Lors de l’import du fichier :
  + Problème de réseau au CFA qui affiche une erreur à cause du temps de traitement trop long alors que toutes les données sont enregistrées dans la base.
  + Problème d’indentation des valeurs de la dernière colonne dans la base de données (solution la dernière colonne a pour valeur « false »).

## Pierre