

Cours d'entreprise numérique

“Gestion des inscriptions dans une école”

Travail de groupe – Examen janvier 2024

Professeurs : Mr. Huygens & Mr. Devigne
Emilien Leroux & Charlotte LUCAS
1B IR-TI, groupe G



Sommaire

- I. Introduction & besoin client
- II. Solution apportée
- III. Utilisation et implémentation de
PowerAutomate
- IV. Dépassement : SharePoint
- V. Travail d'équipe : organisation, outils
utilisés, méthodologie & difficultés
rencontrées
- VI. Portefolio Git : réseaux, systèmes
d'exploitation, MySQL et programmation
- VII. Conclusion

Introduction

Dans le cadre du cours d'entreprise numérique, le projet "Gestion des inscriptions dans une école" nous a été confié. Ce projet nécessitait d'allier des outils d'automatisation, gestion de mailing et de collaboration en ligne. Notre travail s'est principalement fait à distance, avec quelques réunions informelles à l'école avant le congé d'hiver. Ce rapport met en lumière la solution apportée au client, les outils utilisés et les possibles complications rencontrées vis-à-vis de ces derniers. Enfin, vous trouverez notre portefeuille Git en fin de document, celui-ci reprenant un condensé des commandes vues pendant certains de nos cours, notamment ceux de réseaux et de système d'exploitation.

Consignes reçues

Description : « Une école vous demande de gérer les inscriptions pour la rentrée 2024, avec des sections telles que l'informatique, latin-grec, mathématiques fortes et sciences fortes. Chaque secrétaire doit pouvoir soumettre les inscriptions via un formulaire. Chaque demande doit être validée manuellement par le responsable des inscriptions. Une fois validée, un courrier électronique de confirmation est envoyé au secrétaire et au futur élève, et les informations sont enregistrées dans un fichier Excel. En cas de refus, un courrier électronique avec la justification est envoyé. »

Dépassement : « Pour aller plus loin, envisagez d'enregistrer ces informations dans une liste SharePoint. De plus, le 1er septembre, vous pourriez automatiser l'envoi d'un courrier électronique par section aux professeurs, contenant les noms des élèves sous forme de liste à puces dans le corps de l'e-mail. »

Solution apportée

L'outil principal utilisé pour mener à bien ce projet a été **PowerAutomate**, et **SharePoint** pour la partie "dépassement". Nous avons créé un flux que l'on va élaborer dans les pages suivantes, et ce dernier est lié à une liste SharePoint.

Utilisation et implémentation de PowerAutomate

Décomposition en étapes

- Un potentiel nouvel étudiant remplit un formulaire,
- le formulaire rempli est envoyé au responsable des inscriptions,
- le responsable choisi de valider ou pas la demande d'inscription,
- en fonction de l'état de validation de la demande, un mail est envoyé au potentiel nouvel étudiant,
- si la demande a été validée, un tableau Excel répertorie ce nouvel étudiant.

Pratique Power Automate

Tout d'abord, il faut créer le formulaire avec **Microsoft Forms**. Ce formulaire reprend des informations basiques, celles-ci pouvant être approfondies si nécessaire (exemples : obtenir une adresse postale, date de naissance...). On demande également de choisir la section désirée.

Lorsqu'un utilisateur remplit le formulaire, il y a obligatoirement une adresse mail enregistrée, le formulaire n'étant pas en anonyme. Demander une adresse permet aussi à l'utilisateur de rentrer une adresse mail spéciale pour l'école ou le travail au cas où il souhaite ne recevoir les mails concernant son inscription que sur cette adresse mail précisément. On va ensuite créer le flux **PowerAutomate**.

The image shows a Microsoft Forms interface with four questions:

1. Nom * (Text input field with placeholder "Enter your answer")
2. Prénom * (Text input field with placeholder "Enter your answer")
3. Adresse mail * (Text input field with placeholder "Enter your answer")
4. Section choisie * (Radio button selection with four options: Informatique, Latin-Grec, Mathématiques, Sciences fortes)

A green "Submit" button is located at the bottom of the form.

L'élément déclencheur du programme sera l'envoi d'une réponse à un formulaire. À chaque fois que quelqu'un répondra au formulaire, le programme se mettra en marche.

On rajoute l'étape *envoyer un mail avec option* qui sera destiné au responsable des inscriptions. Dans ce mail on y met le prénom et nom de l'élève ainsi que la section souhaitée.

Ce mail comportera deux options que devra choisir le responsable : **valider** ou **refuser** l'inscription.

Si validation il y a, une étape est ajoutée pour enregistrer les données afin de les stocker dans un fichier Excel et les réutiliser plus tard dans le programme. La création de ce tableur est expliquée plus bas.

Nom	Prénom	Adresse mail	Section choisie
Leroux	Emilien	etu55886@henallux.be	Informatique
Lucas	Charlotte	etu53705@henallux.be	Latin-Grec
Lethan	Vadim	etu53705@henallux.be	Informatique
Andersen	Michael	etu53705@henallux.be	Latin-Grec
McGuy	Maëlle	etu53705@henallux.be	Sciences fortes
Leroy	Gale	etu53705@henallux.be	Mathématiques
Courucelle	Esteban	etu53705@henallux.be	Latin-Grec
Sacerse	Anthinéa	etu53705@henallux.be	Informatique
Laudelet	Esther	etu53705@henallux.be	Latin-Grec
Ansiaux	Kenneth	etu53705@henallux.be	Informatique

En cas de refus :

Afin de décider de l'action en fonction du choix du responsable, il faut ajouter une étape de condition.

La condition repose sur l'option choisie par le responsable. Si elle est égale à valide, c'est-à-dire si le responsable clique sur valider, il va exécuter une action qu'on verra juste après.

The image shows two parts of a software interface. The top part is a dialog box titled 'Envoyer un e-mail avec des options'. It has fields for 'À' (recipient: etu55886@student.henallux.be), 'Objet' (subject: Nouvelle demande d'inscription), and 'Options utilisateur' (options: Valider, Refuser). There is a link 'Afficher les options avancées'. The bottom part is a 'Condition' configuration panel. It shows a condition: 'Selected...' (from a dropdown) 'est égal à' (is equal to) 'Valider'. There is an 'Ajouter' (Add) button below.

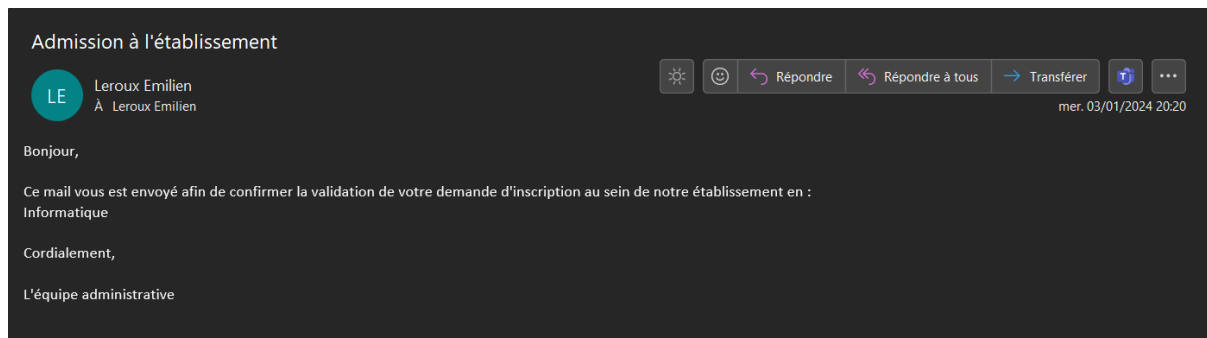
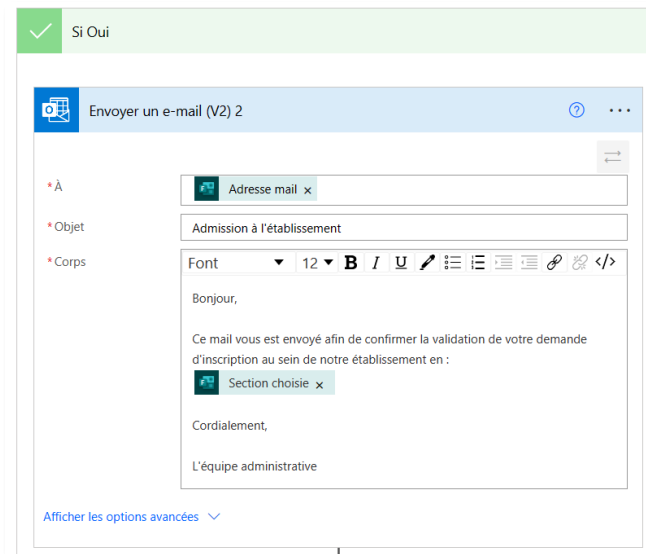
The image shows a 'Si Non' (If No) branch of a workflow. It displays a preview of an email titled 'Envoyer un e-mail (V2) 3'. The email content is as follows:
* À: Adresse mail
* Objet: Refus d'admission
* Corps: Bonjour,
Ce mail vous est envoyé afin de vous prévenir du refus d'admission de votre demande d'inscription en raison d'un manque de place dans la section que vous avez choisie.
N'hésitez pas à revenir vers nous pour plus de renseignements.
Cordialement,
L'équipe administrative

Dans le cas d'un refus d'inscription, ce mail s'enverra et la personne ayant rempli le formulaire recevra un mail de refus avec justification, actuellement limitée au nombre de places.

The image shows an email interface with a dark theme. The email is titled 'Refus d'admission' and is from 'Leroux Emilien' to 'Leroux Emilien'. The email content is the same as the one in the previous screenshot. The interface includes a header with a profile picture, name, and email address, and a toolbar with buttons for 'Répondre', 'Répondre à tous', and 'Transférer'. The date and time 'mer. 03/01/2024 20:13' are shown in the bottom right corner.

En cas de validation :

Si le responsable clique sur valider, un mail s'enverra à l'étudiant ainsi qu'à la secrétaire. Ensuite, l'inscription sera enregistrée dans le fichier Excel susmentionné.



Une autre étape est rajoutée afin de prévenir la secrétaire de la validation du responsable.

Envoyer un e-mail (V2)

* À: etu55886@henallux.be

* Objet: Admission de [Nom] à l'établissement

* Corps:

Font: 12, Bold, Italic, Underline, Link, Unlink, Code

Bonjour,

Ce mail vous est envoyé afin de confirmer la validation de la demande de [Nom] en [Section choisie].


Cordialement,

L'équipe administrative

[Ajouter du contenu dynamique](#)

[Afficher les options avancées](#)

Admission de Leroux Emilien à l'établissement



 Leroux Emilien
À Leroux Emilien

Bonjour,

Ce mail vous est envoyé afin de confirmer la validation de la demande de Leroux Emilien en Informatique.

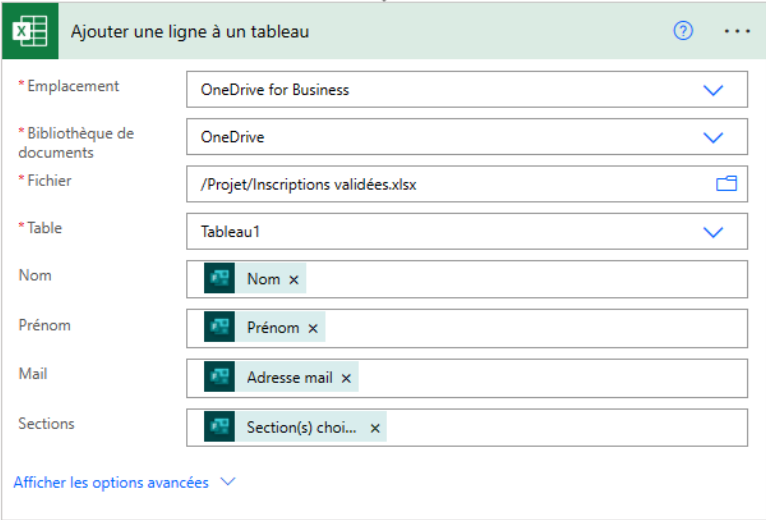
Cordialement,

L'équipe administrative

  [Répondre](#) [Répondre en masse](#)

Ensuite, pour enregistrer les informations dans le tableau Excel, il faut tout d'abord le créer (nommé ici « Inscriptions validées »).

On y insère un tableau vide comprenant 4 colonnes : nom, prénom, adresse mail et section choisie. On pourrait l'améliorer dans le futur avec une colonne ID qui permettra d'identifier chaque étudiant de manière unique. Dans le flux, on ajoute une étape basée sur l'outil Excel. Celui-ci nous permet d'ajouter une ligne à un tableau. Il suffit juste de le sélectionner et de remplir les colonnes avec les enregistrements dynamiques des informations données par le nouvel étudiant.



Ajouter une ligne à un tableau

* Emplacement: OneDrive for Business

* Bibliothèque de documents: OneDrive

* Fichier: /Projet/Inscriptions validées.xlsx

* Table: Tableau1

Nom: Nom x

Prénom: Prénom x

Mail: Adresse mail x

Sections: Section(s) choi... x

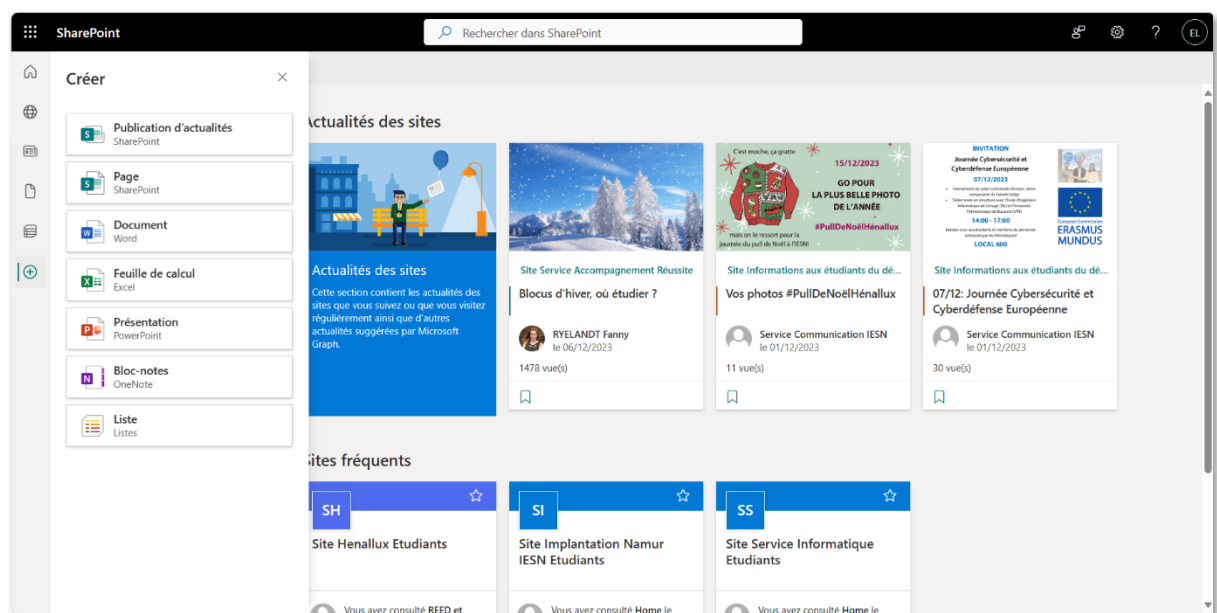
[Afficher les options avancées](#)

Dépassement : SharePoint

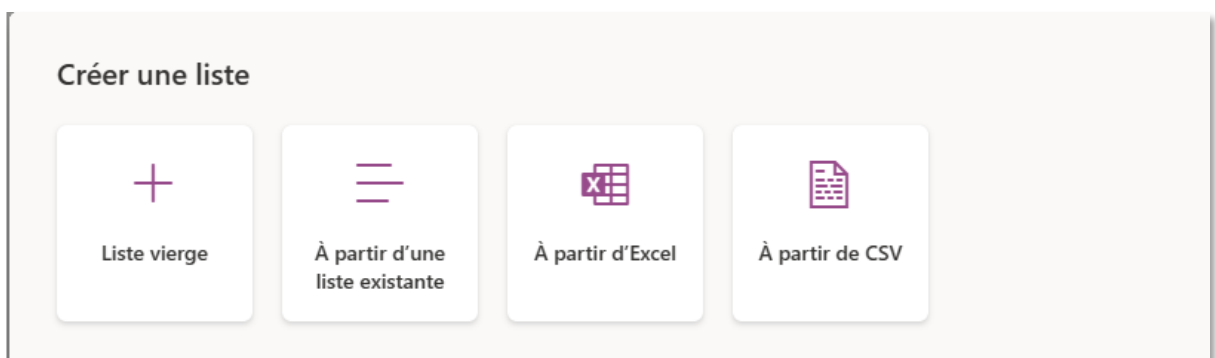
Le dépassement propose de mettre les informations demandées dans une liste **SharePoint** qui permettra ensuite d'envoyer une liste des inscrits par section aux professeurs attirés, et ce de manière automatique au 1^{er} septembre. Ces listes sont sous format puces dans le corps même de l'email : cela implique donc de pouvoir retirer les informations de la liste SharePoint et les placer d'une manière spécifique dans le mail.

Première partie : La liste SharePoint

En premier, on crée la liste SharePoint. Dans le cadre de notre compte étudiant Henallux, nous ne pouvons pas créer de site, mais des listes. Henallux propose déjà trois sites mais nous ne pouvons créer aucune liste sur ces sites. Nous devons donc créer une liste qui ne sera sur aucun site, ceci aura un impact que l'on verra plus tard.



Il est possible de créer une liste en fonction d'un tableau Excel, mais nous allons en créer une simple liste vierge que nous allons automatiser dans le flux.



On va ensuite créer 4 colonnes dans la liste, chacune correspondant aux réponses entrées par le nouvel inscrit.

Rechercher

Automatiser Intégrer

+ Ajouter une colonne

Créer une colonne

[En savoir plus sur la création de colonnes.](#)

Nom *

Prénom

Description

Type

Une ligne de texte

Valeur par défaut

Entrez une valeur par défaut

☐ Utiliser la valeur calculée ⓘ

Plus d'options

Enregistrer Annuler

Voici la liste créée avec les colonnes. Vu que le nom a été entré en premier, il est nommé comme colonne titre.

Mes listes

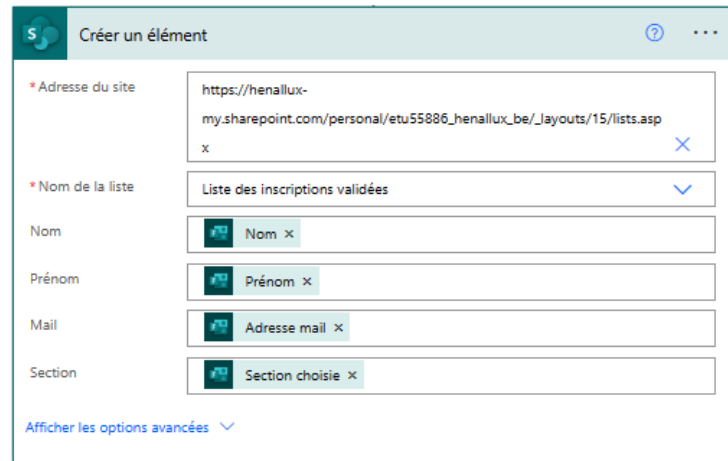
Liste des inscriptions validées

+ Ajouter une colonne

Titre Prénom Mail Section

Deuxième partie : ajouter automatiquement les nouveaux inscrits à la liste via le flux

Nous allons ajouter l'étape « Ajouter un élément » de l'outil SharePoint en dessous de l'étape permettant d'ajouter une ligne dans le tableau Excel. Comme mentionné précédemment, si on ne crée pas la liste sur un des sites proposés à la création mais juste dans « mes listes », il faudra entrer manuellement le site où se trouve cette liste dans votre SharePoint, pour remplir les colonnes par les informations collectées par le formulaire.

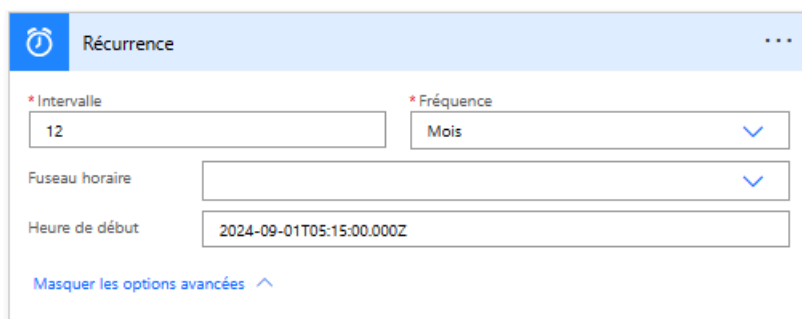


Et voilà, vous avez une liste qui s'actualise au fur et à mesure que vous validez l'inscription des gens qui souhaitent s'inscrire à l'établissement.

Troisième partie : l'envoi de la liste des inscrits le 1er septembre

Le but de cette liste SharePoint étant d'automatiser l'envoi des listes d'élèves aux professeurs en date du 1^{er} septembre, on peut désormais ajouter une **réurrence** dans le PowerAutomate. Pour cela, on fait un deuxième flux. Cela peut se faire dans le premier flux créé, mais par soucis de lisibilité et aisance de correction de bug, nous préférons les diviser.

1. Créer un nouveau flux qui a pour élément déclencheur la date voulue.



2. On va initialiser une variable de chaîne par section.

The screenshot shows a vertical sequence of four 'Initialiser la variable' (Initialize variable) steps, connected by downward arrows. Each step is a card with a purple header and a white body. The steps are:

- Initialiser la variable "Liste des informatique"**: Nom: Informatique, Type: Chaîne, Valeur: Entrer la valeur initiale.
- Initialiser la variable "Liste des Latin-Grec"**: Nom: Liste des Latin-Grec, Type: Chaîne, Valeur: Entrer la valeur initiale.
- Initialiser la variable "Sciences fortes"**: Nom: Sciences fortes, Type: Chaîne, Valeur: Entrer la valeur initiale.
- Initialiser la variable "Mathématiques"**: Nom: Mathématiques, Type: Chaîne, Valeur: Entrer la valeur initiale.

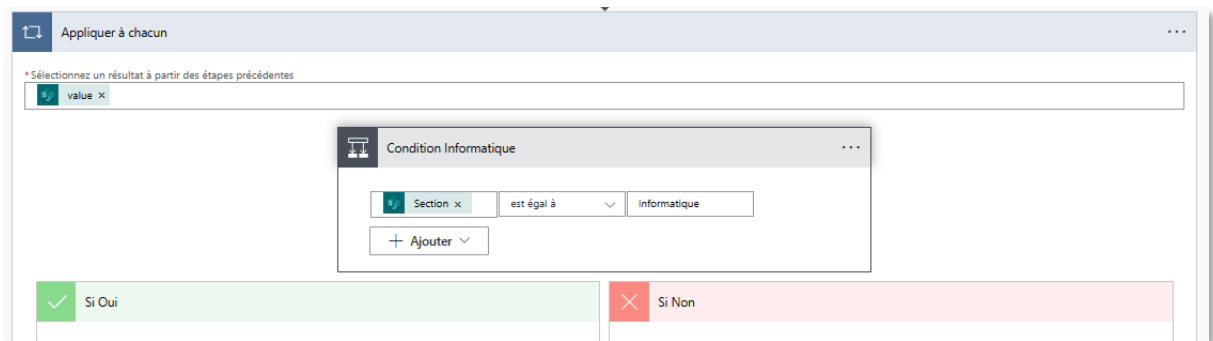
3. On peut renseigner les informations de la liste SharePoint au flux pour pouvoir les insérer dans les variables au moment voulu.

The screenshot shows the 'Obtenir les éléments' (Get items) action card. It has a light blue header with the 'S' icon and the title. The fields are:

- * Adresse du site**: https://henallux-my.sharepoint.com/personal/etu55886_henallux_be/_layouts/15/lists.aspx
- * Nom de la liste**: Liste des inscriptions validées
- Limiter les entrées au dossier**: Sélectionner un dossier ou laisser vide pour l'ensemble de la liste
- Inclure les éléments imbriqués**: Retourner les entrées contenues dans les sous-dossiers (valeur par défaut)

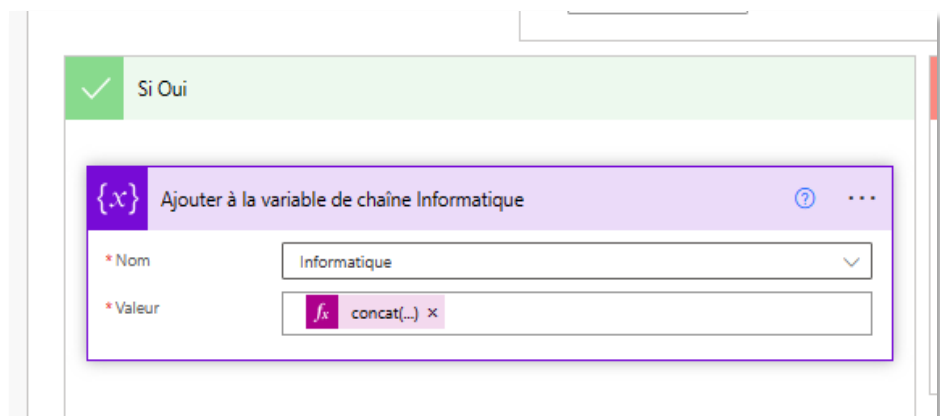
At the bottom, there is a link: [Afficher les options avancées](#)

4. Créer une boucle permettant de faire plusieurs actions pour chaque élément de la liste et rajouter une autre étape de contrôle de condition, cette fois-ci pour sortir de la boucle si la section correspondant à l'élément actuel est égale à « informatique »



5. Si la condition est **vérifiée** et **vraie**, la variable consacrée à la liste de noms des inscrits en informatique sera mise à jour. Dans l'étape « *Ajouter à la variable de chaîne* », il est nécessaire de modifier le code de l'expression et de ne pas uniquement utiliser du contenu dynamique.

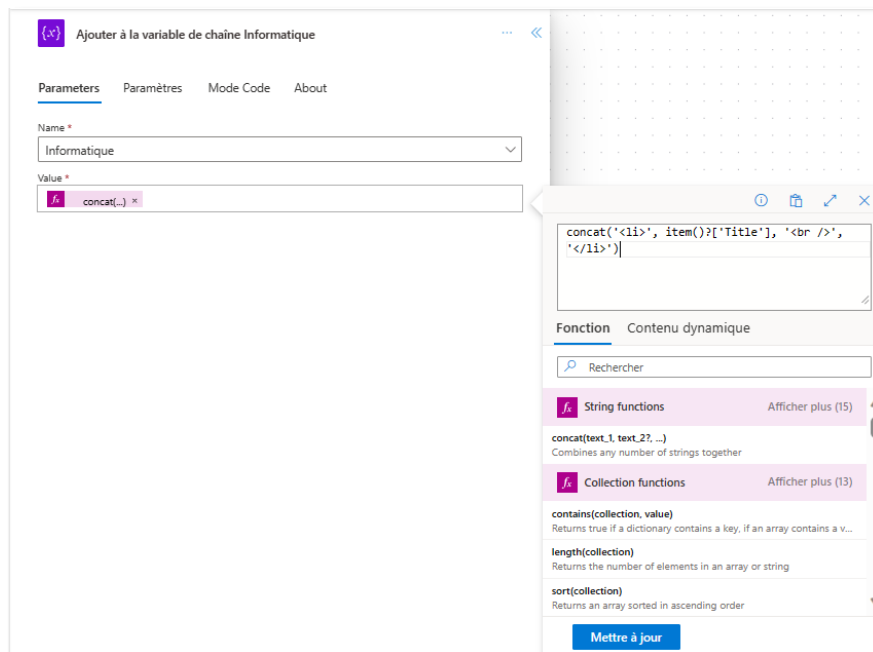
Nous allons commencer par créer la fonction servant aux strings « concat » que nous allons remplir d'une part du contenu



dynamique lié à la liste SharePoint « titre », c'est-à-dire le nom de la personne inscrite en informatique dans le cas présent puisque cela se passe dans la boucle qui vérifie si la section de l'inscrit est informatique. Lorsque nous choisissons le contenu dynamique titre, le code se verra rempli de « **item() ?['Title']** ».

Ensuite, afin de pouvoir avoir dans le corps de l'email une liste à puces, il est nécessaire d'entrer du code html dans la fonction « concat » : il faut d'abord placer un code permettant de retourner à la ligne après le nom. Ceci est défini par « **
** ».

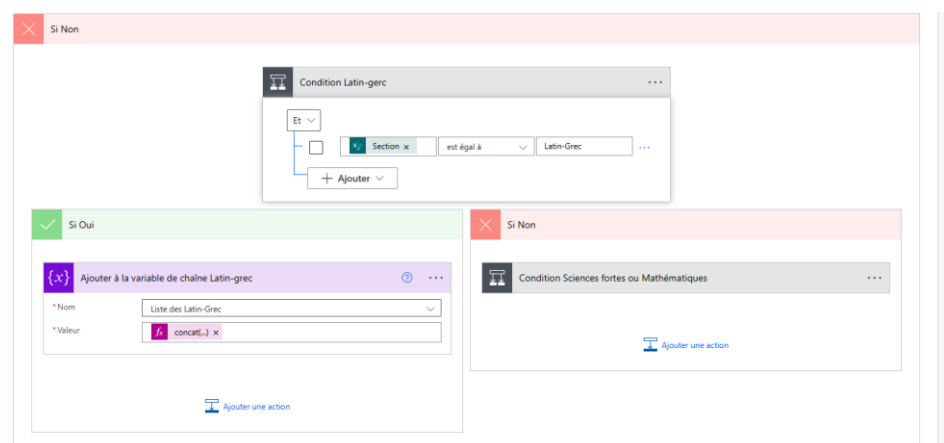
Enfin, pour que tout ceci soit sous une forme de liste à puces, il faut entourer les commandes précédentes par les commandes « **** » et « **** ». Il ne faut pas oublier de séparer les commandes dans le code avec des virgules et voila le code à recopier à chaque fois qu'une liste d'inscrit sera à mettre à jour.



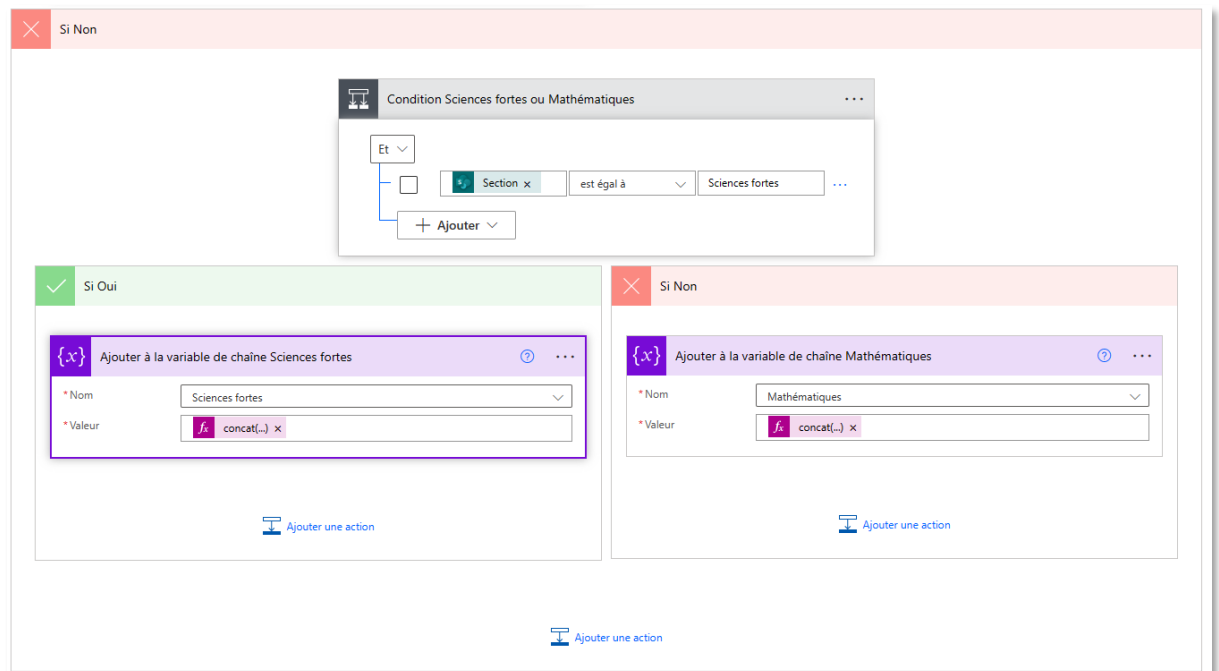
6. Il ne reste plus qu'à manier la condition de section pour faire varier les choix de cette manière : si la section vue précédemment n'est pas égale à « informatique », une autre

condition se basant toujours sur la section mais en se référant à un autre choix possible, ici « Latin-Grec », amènera à

varier la variable destinée à répertorier de la même façon que précédemment et ce si la condition est vérifiée avec exactement le même code que précédemment.



7. Ensuite, il ne reste plus que deux choix possibles donc une seule condition suffira. Celle-ci se basera toujours sur la section et se référera cette fois à l'option « Sciences fortes ».
- Si la condition est vérifiée, la variable concernant les noms des inscrits en science fortes sera mise à jour, sinon, c'est la variable concernant les inscrits en mathématique qui sera mise à jour.



8. Pour ce qui est des mails, il suffit de rajouter 4 étapes en parallèle ou à la suite, ça n'a pas beaucoup d'importance.



Ces 4 étapes seront un envoi de mail au professeur souhaité (ici c'est l'email de test mais on peut le remplacer par n'importe lequel, ou en ajouter un deuxième en les séparant de « ; » comme dans l'autre flux). Chaque mail sera personnalisé de manière à avoir un objet plus adéquat en fonction de la liste envoyée (si c'est la liste des inscrits en informatique, l'objet sera « liste des inscrits en informatique ») et le corps de chaque mail contiendra la variable utile au professeur. Il n'y a pas besoin de rajouter l'option pour faire une liste à puces dans le corps de l'email car il est déjà intégré au code HTML, cela créerait juste deux puces au premier nom.

Envoyer un e-mail au professeur d'informatique

* À: etu55886@student.henallux.be

* Objet: Liste des inscrits en informatique

* Corps: Font 12 B I U {x} Informatique x

[Afficher les options avancées](#)

Liste des inscrits en informatique

Leroux Emilien
À Leroux Emilien

Ansiaux
Colyn
Austen

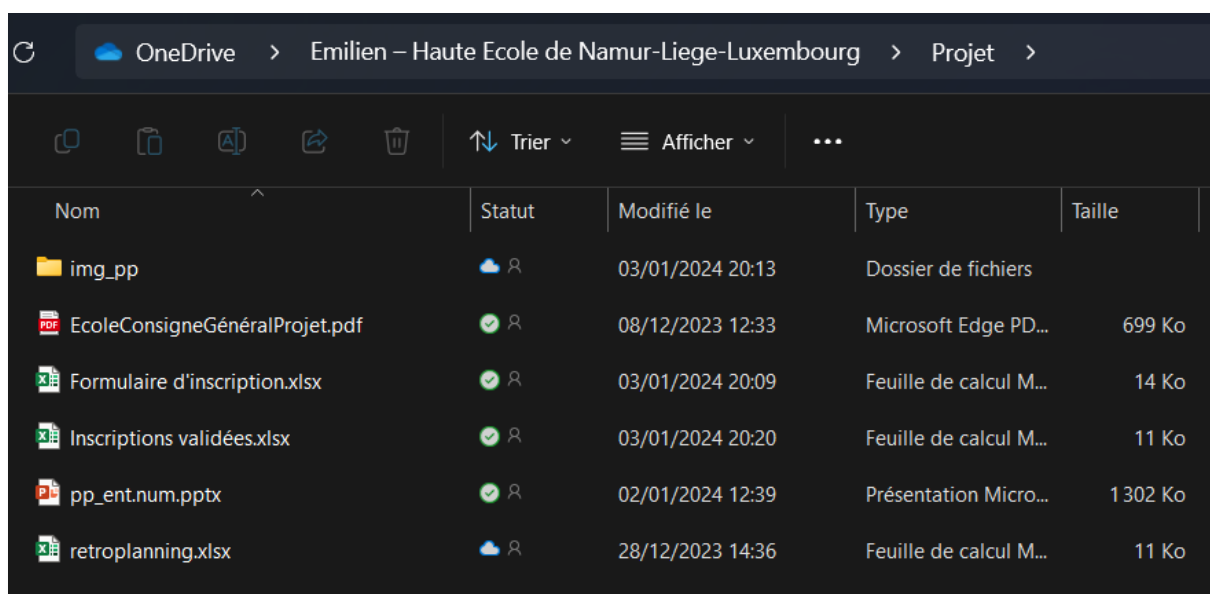
Répondre Répondre à tous Transférer

sam. 30/12/2023 18:03

Travail d'équipe : organisation, outils utilisés & méthodologie

La majeure partie des outils utilisés n'étaient pas inconnus de notre équipe. Ils ont soit été utilisés dans un cadre professionnel ou d'éducation : il n'aura donc pas fallu longtemps pour choisir ce que nous utiliserions et implémenter le tout dans notre collaboration.

Pour **la création et gestion des fichiers** (en dehors du portfolio et du rapport qui sont tous deux sur le GitHub), nous avons utilisé la suite Office avec nos comptes Henallux. Notre PowerPoint, les fichiers Word et Excel étaient donc sur un OneDrive partagé en privé, avec une structure simple. On y retrouve aussi les tableurs liés au PowerAutomate.



The screenshot shows a OneDrive interface with the following structure:

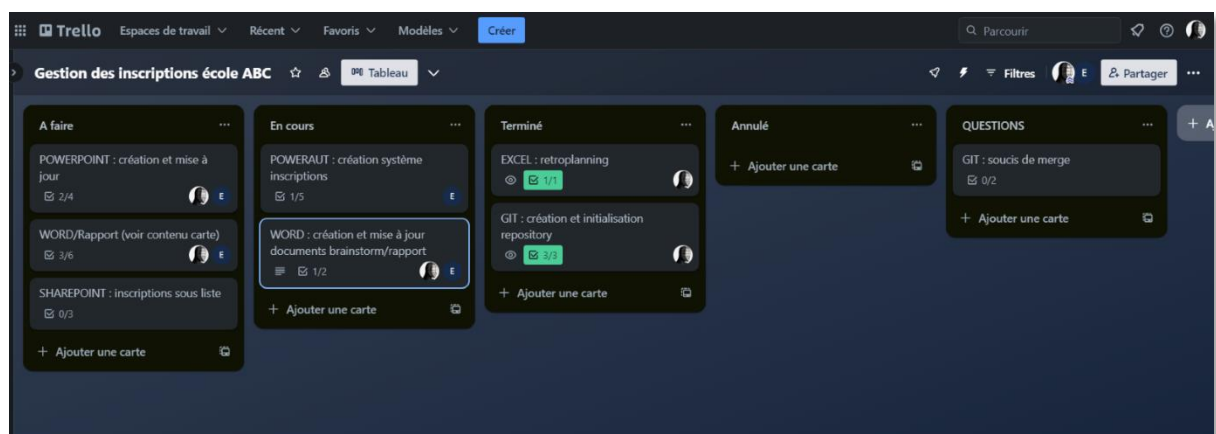
- OneDrive > Emilien – Haute Ecole de Namur-Liege-Luxembourg > Projet

Nom	Statut	Modifié le	Type	Taille
img_pp	Cloud icon	03/01/2024 20:13	Dossier de fichiers	
EcoleConsigneGénéralProjet.pdf	Checkmark icon	08/12/2023 12:33	Microsoft Edge PD...	699 Ko
Formulaire d'inscription.xlsx	Checkmark icon	03/01/2024 20:09	Feuille de calcul M...	14 Ko
Inscriptions validées.xlsx	Checkmark icon	03/01/2024 20:20	Feuille de calcul M...	11 Ko
pp_ent.num.pptx	Checkmark icon	02/01/2024 12:39	Présentation Micro...	1 302 Ko
retroplanning.xlsx	Cloud icon	28/12/2023 14:36	Feuille de calcul M...	11 Ko

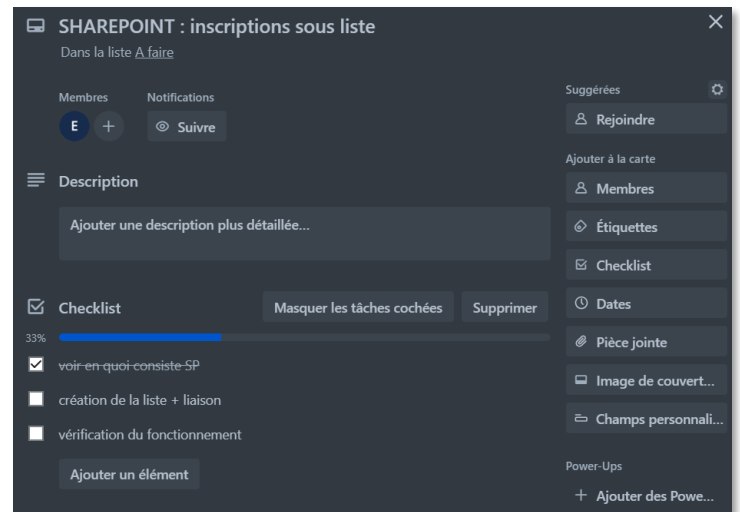
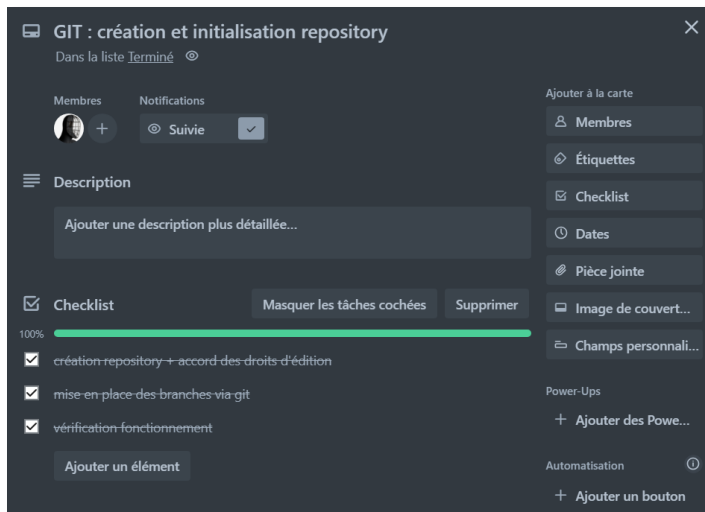
Pour la répartition des tâches nous avons d'abord dégrossi le projet en plus petites tâches qui ont ensuite été placés dans un simple retroplanning Excel, avec des deadlines. C'est aussi à ce moment-là que nous avons défini "*qui ferait quoi*", même s'il était évident que chacun devrait travailler sur tout afin de comprendre comment fonctionnait la solution apportée. Nous avons cependant défini les tâches selon les facilités, expériences et préférences de chaque.

TACHES	11 - 17 décembre	18 - 24 décembre	25 - 31 décembre	1er - 7 janvier			Dates importantes	Légende
Création forms & liaison PA							5 janvier : finalisation tout	PowerAutomate x Forms
Création et initialisation GIT/GitHub							6 janvier : relecture, correction bugs	GIT/GitHub/Portfolio
Mise à jour portfolio via Git							7 janvier : date max remise dossier	Rapport
Vérification contenu Git; finalisation								Autres
Choix méthodologie et implémentation des outils								
Choix template PP + sommaire								
Brouillon PP								
Mise au propre PP + relecture								
Brainstorm contenu rapport								
Premier jet rapport								
Deuxième jet rapport + relecture								
Finalisation rapport + relecture finale				LIMITE : 5 janvier				
Vérification contenu complet + correction bug				LIMITE : 6 janvier				

Pour cela nous avons utilisé **Trello** sur lequel une board dédiée au projet a été créé. Afin d'éviter une surcharge de travail et de micro-management, nous avons créé des catégories basiques mais évidentes : **À faire, En cours, Terminé, Annulé et Questions**.



Les tâches du retroplanning ont ensuite été replacées dans le Trello, avec des “*checklist*” là où c’était nécessaire. Les membres travaillant sur telle ou telle tâche se les sont vus attribués afin d’avoir une vue globale de qui fait quoi. Trello étant un freemium (*outil gratuit avec des avantages déblocables en payant*), nous n’avons pas eu accès à tous les outils proposés, ainsi que les add-ons. Ce ne fut pas un souci dans le cadre de ce projet mais pour une plus grosse équipe et avec beaucoup plus de tâches, la plateforme serait plus intéressante en premium, voir un autre outil du genre (Asana, Monday dot com...).

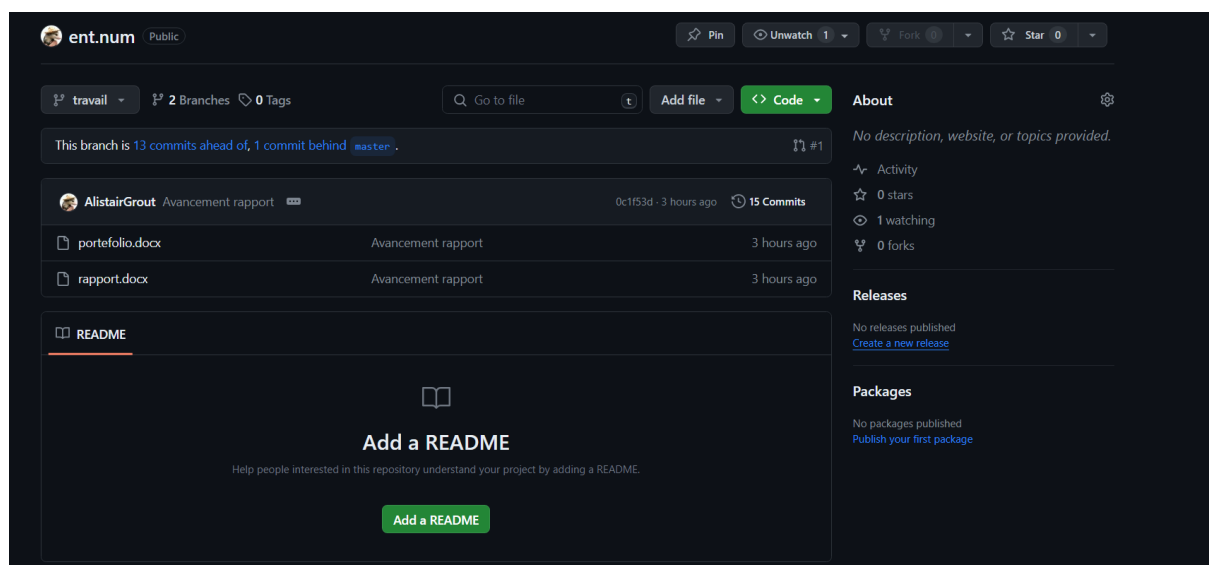


Nos communications se sont principalement faites sur **Discord**, en messages privés. Par facilité, nous avons “pin” à la conversation les liens les plus utiles ainsi que les commandes Git étant nécessaires tout au long du projet. Cette conversation a principalement servi à faire des retours, discuter des problèmes et bugs ainsi qu’envisager des améliorations. Les commentaires Git permettaient aussi de voir ce qui étaient fait, changé ou à faire. Notre organisation a été suffisamment bien pensée pour la taille de notre équipe et le projet que pour ne pas avoir à faire des longues réunions en vocal ou à l’école. Dans un contexte de projet et équipe de plus grande envergure, ce serait à revoir.

Portefolio Git

Pour le portefeuille Git, nous avons décidé de partir sur un document des commandes les plus utilisées, en notre sens, dans différents cours mais aussi de manière globale. Nous avons donc pris les cours de réseaux, systèmes d'exploitation, programmation et base de données (MySQL).

Chacun a d'abord rempli une page « brouillon » où il/elle pouvait déposer les commandes qui lui venait à l'esprit, pour que le tout soit ensuite regroupé en tableau quelques jours avant la fin du projet. La mise en forme tableau permettait une vision plus claire et propre, en notre sens. Vous trouverez les pages de commandes choisies en suite.



Portefolio : Réseaux

Note : beaucoup des commandes mentionnés nécessitent d'ajouter des informations en plus. Par exemple, « taskkill » a besoin du nom ou ID du process pour savoir lequel fermé. En général « ? » permet d'obtenir des informations supplémentaires sur une commande.

Commande	Résultat
ping	Permet de confirmer l'accessibilité de deux machines
ipconfig, ipconfig /all, ipconfig /?	Permet d'obtenir la configuration ip actuelle, avec + ou – de détails
ipconfig /release ipconfig /renew	Dans une configuration dynamique, permet de libérer ou renouveler l'ip
nslookup	Trouver l'adresse ip du serveur dns d'un site en se basant sur son url. Ou inversement
netsh	Permet la configuration du réseau, changement IP, ainsi que lire les informations actuelles de l'interface
netsh interface ip set address "Ethernet" static <ipv4_address> <netmask>	Ethernet : nom carte réseau Static : adresse de type static (DHCP est l'autre) <ipv4_address> : adresse IP en utilisation <netmask> : masque
netsh interface set interface "Ethernet" enable/disable	Active ou désactive une carte réseau
cat /etc/resolv.conf	Affiche le contenu du fichier resolv.conf, ce dernier permettant de voir quels serveurs DNS on utilise et le search domain

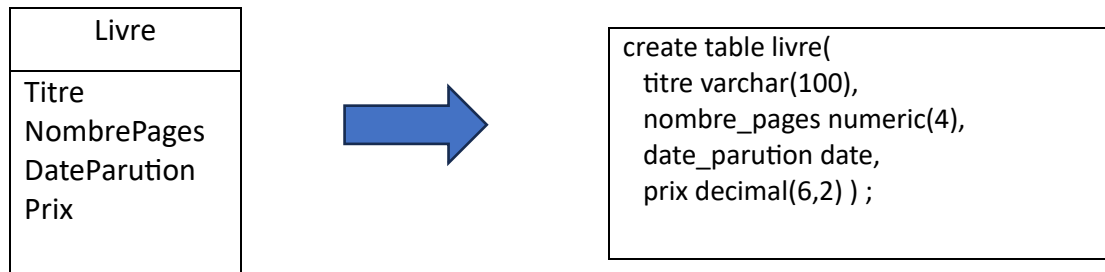
Portefolio : Systèmes d'exploitation

Note : beaucoup des cmds mentionnées nécessitent d'ajouter des informations en plus. Par exemple, taskkill a besoin du nom ou ID du process pour savoir lequel fermé. En général ? permet d'obtenir des informations supplémentaires sur une cmd.

Commandes Linux	Résultat
cmd clear	<ul style="list-style-type: none"> - Ouvre l'invite de commande - Nettoie l'invite de commande
cp mv rm	<ul style="list-style-type: none"> - Copie un fichier/rep - Déplace un fichier/rep - Efface un fichier/rep
nano nomfichier	<ul style="list-style-type: none"> - Ouvre/créer un fichier au nom donné dans le rep
cd, cd /, cd..	<ul style="list-style-type: none"> - Change vers un rep spécifique, remonte à la racine, remonte d'un niveau dans le rep
Ls, ls -r, ls -a	<ul style="list-style-type: none"> - Liste les fichiers et rep dans le rep actuel - Liste les sous-rep - Liste les fichiers cachés
cat nomfichier	<ul style="list-style-type: none"> - Montre le contenu du fichier mentionné
su -root	<ul style="list-style-type: none"> - Permet d'avoir les privilèges admin
mkdir rmdir	<ul style="list-style-type: none"> - Crée un nouveau rep - Supprime un rep
setxkbmap be	<ul style="list-style-type: none"> - Permet de changer la disposition clavier
apt-get install/update	<ul style="list-style-type: none"> - Installer/mettre à jour des paquets

Commande Windows	Résultat
cmd cls help	<ul style="list-style-type: none"> - Ouvre l'invite de commande - Nettoie l'invite de commande - Affiche l'aide de commande
shutdown	<ul style="list-style-type: none"> - Éteint la machine
copy, move dir del/erase cd	<ul style="list-style-type: none"> - Copie/déplace un fichier - Affiche les fichiers et rep dans le rep actuel - Supprime un fichier - Change de répertoire
Systeminfo	<ul style="list-style-type: none"> - Donne les informations du système
tasklist taskkill	<ul style="list-style-type: none"> - Affiche la liste des process en cours sur l'ordinateur - Permet de fermer un process en cours

Portefolio : base de données MySQL



Principes importants : les contraintes d'intégrité

- NOT NULL : rend la colonne obligatoire
- PRIMARY KEY : identifiant principal (clé primaire)
- UNIQUE : Valeur unique (appelé aussi clé secondaire). Peut être facultative
- FOREIGN KEY : définit la colonne comme clé étrangère
- CHECK : contraintes additionnelles

Commande	Résultat	Exemples
CREATE DATABASE/TABLE/USER DROP TRUNCATE ALTER DATABASE	<ul style="list-style-type: none"> - Crée le contenu spécifié - Fait tomber la table (ATTENTION) - Supp les lignes d'une table mais garde sa structure - Modifie la DB 	
UPDATE where	<ul style="list-style-type: none"> - Met à jour toutes les lignes - Avec where, définit la ligne à changer 	update personne set email = test@test.com where matricule = '1234 ';
INSERT INTO	<ul style="list-style-type: none"> - Permet d'ajouter du contenu dans une table déjà existante 	
SELECT * FROM where	<ul style="list-style-type: none"> - Permet de sélectionner d'après une information spécifiée - Idem, avec une condition 	
DELETE FROM where	<ul style="list-style-type: none"> - Supprime tout contenu d'après la colonne choisie - Supprime une ligne spécifique 	delete from book where matricule = '1235' ;
CONCAT LEN ROUND	<ul style="list-style-type: none"> - Concat deux mots en un - Donne la longueur d'un mot - Arrondi un chiffre 	'Hello', 'World' devient HelloWorld LEN('Bonjour') => 6 ROUND(123.456,2)=> 123.46

Portefolio : programmation

Pour cette page, nous sommes partis sur des notions qui nous semblent importantes pour la base de la programmation en C. Comme pour les tableaux de commandes ci-dessus, il s'agit d'une liste non-exhaustive.

Démarrage console

- cl nomfichier.c dans le dossier om se trouve le fichier
- créer un .obj et .exe
- exécuter le .exe pour avoir le code crée.

Priorités opérateurs

-(unaire) ++ -- (type)	←
* / %	→
+ - (binaire)	→
< <= > >=	→
== !=	→
?:	←
= += -= *= /= %=	←



Affectation et opérations

== signifie côté gauche est égal à côté droit

!= différent de

+= supérieur ou égal

-= inférieur ou égal

Clean Code

- Indentation – Longueur des lignes (79 cara max) – utilisation des lignes vide (découpage en bloc) – placement des accolades (choisir un style et s'y tenir)
- Mieux découper son code le rend plus lisible et facile à déboguer.

Variables

- Se tenir à une langue (pas de français)
- Choisir une convention et s'y tenir. Suggestions : constantes/macros en MAJ, fonctions/variables initial en MIN
- Noms composés : choisir une convention. PascalCase, camelCase, caractère_souligné
- Donner des noms logiques à ses variables, encore plus si sa portée est significative. Une autre personne reprenant le code doit pouvoir comprendre.

Conclusion

Ce projet fut un défi intéressant, notamment pour la question du travail d'équipe à distance (principalement accompli pendant les congés d'hiver), mais aussi l'apprentissage plus ou moins approfondi d'outils sur le tas. Si PowerAutomate avait déjà été vu en laboratoire, ce n'était pas pour autant évident pour l'équipe entière, et SharePoint était relativement nouveau. Il en va de même pour Git/GitHub, qui ne sont pas des outils inconnus de notre binôme, mais qui ne sont pas maîtrisés pour autant.

Ça nous a donc demandé une certaine **autonomie** pour assurer que chaque partie pouvait accomplir le projet dans les temps, tout en communiquant régulièrement. Finalement, les seuls soucis vraiment rencontrés venaient de Git (mise en route du repository, branching) et du SharePoint.

Si une troisième personne avait rejoint l'équipe, il aurait probablement fallu repenser l'organisation et les communications, par exemple en implémentant des réunions à distance hebdomadaires.

Des **améliorations** ont été pensées pour le projet même : un suivi par mail automatique après acceptation pour demander au parent/gardien de compléter plus d'informations (comme le fait d'une certaine manière la plateforme d'inscription en ligne de l'Henallux) nécessaires. Mais aussi un choix de justification de refus (manque de document, élève renvoyé de l'école par le passé...), une création d'ID par étudiant comme identifiant unique dans l'Excel/base de données...

La structure de notre projet conviendrait pour une petite école, disons de village, mais pour une structure telle que l'Henallux, un autre schéma devrait être envisager pour assurer un suivi de chaque mail reçu (un parent qui demande pourquoi son enfant a été refusé par exemple) mais aussi une cohérence entre les données et éviter des doublons.

Si tout aurait pu être fait dans le même flux PowerAutomate vis-à-vis du dépassement SharePoint, comme susmentionné dans les pages à ce sujet, il est plus propre et facile à manier de les diviser. Cela assure aussi que pour une tierce personne qui reprendrait le projet, le flux ne soit pas un labyrinthe illisible.