



TELECOM Nancy

Projet CS54

Drevocracia

VOISIN Victor
CHANEL Valentin
ELAZIZ Elyas
PALLET Thomas

Responsable de module :
FESTOR Olivier



Table des matières

1	Introduction au sujet	5
1.1	Contexte du sujet	5
1.2	Organisation du document	5
1.3	Présentation de Drevocracia	5
2	Etat de l'art	7
2.1	Principes des applications de la Civic Tech	7
2.2	Etude de l'état de l'art des applications de la Civic Tech	7
3	Conception et implémentation : base de données	9
3.1	Conception : analyse des besoins	9
3.2	Implémentation de la base de données	10
4	Conception et implémentation : serveur web	11
4.1	Structure de notre serveur Web	11
4.2	Implémentation	12
5	Conception et implémentation : algorithmes de traitement	15
5.1	Fonctions usuelles	15
5.1.1	Analyse des principales fonctions avec traitement de données	15
5.1.2	Division d'un texte	16
5.1.3	Récupération du fil des questions/solutions/messages	16
5.1.4	Vérification de la validité de l'email	16
5.1.5	Cryptage et decryptage XOR	16
5.2	Backtracking	16
6	Tests et performances	17
6.1	Tests sur la base de données	17
6.2	Tests sur les algorithmes	17
6.3	Performances	18
7	Gestion de Projet	19
7.1	Composition de l'équipe du projet	19
7.2	Outils de gestion de projet	19
7.2.1	Définition des objectifs	19
7.2.2	Prise en compte des risques	19
7.3	Organisation du projet	20
7.4	Comptes-rendus des réunions	21
7.4.1	3 Novembre 2021	21
7.4.2	10 Novembre 2021	22
7.4.3	24 Novembre 2021	22
7.4.4	29 Novembre 2021	25
7.4.5	6 Décembre 2021	25
7.4.6	16 Décembre 2021	26
7.4.7	22 Décembre 2021	27
7.4.8	27 Décembre 2021	28
7.4.9	29 Décembre 2021	29
7.4.10	3 Janvier 2022	30
7.4.11	5 Janvier 2022	32

8	Bilan du projet	33
8.1	Bilan global du projet	33
8.2	Bilan du projet membre par membre	33
8.2.1	VOISIN Victor	33
8.2.2	CHANEL Valentin	34
8.2.3	ELAZIZ Elyas	34
8.2.4	PALLET Thomas	34
8.3	Travail réalisé	34

Chapitre 1

Introduction au sujet

1.1 Contexte du sujet

Ce projet a été réalisé dans le cadre du module de CS54 (Computer Science 54) de la première année du cycle ingénieur sous statut étudiant de TELECOM Nancy.

L’objectif est de concevoir et d’implémenter une application de démocratie participative pour une ville.

Cette application doit être capable de faciliter la démocratie participative locale à l’aide d’un appui sur une base de données, des algorithmes de traitements avancés et une accessibilité via le Web.

Le travail s’est déroulé en plusieurs parties, avec un travail de conception, d’implémentation et de gestion de projet.

1.2 Organisation du document

Dans le chapitre 2, nous présenterons la Civic Tech ainsi que ses objectifs et les applications existantes sous forme d’un état de l’art.

Dans le chapitre 3 à 6, nous présenterons les différents programmes permettant à notre application Web de réaliser certaines opérations.

Dans le chapitre 7, nous présenterons les éléments et les outils de gestion de projet que nous avons mis en place et utilisés.

Dans le chapitre 8, nous réalisons un bilan du projet, d’un point de vue personnel et global.

1.3 Présentation de Drevocracia

Drevocracia est un outil de résolution de problème participatif qui permet aux utilisateurs de construire progressivement une solution à un problème local. Le but est d’éviter le piège de la trop simplification de la participation souvent réduite à voter “oui” ou “non” à des questions parfois beaucoup trop complexes pour être ramenées à des choix binaires.

L’idée est de créer un arbre de décision où chaque citoyen vote à chaque nœud. Le nombre de feuilles disponibles et la hauteur de l’arbre casse la binarité des choix et permet de poser successivement des questions sur différentes modalités du problème.

Chapitre 2

Etat de l'art

2.1 Principes des applications de la Civic Tech

La Civic Tech ou Technologie Civique en français est une technologie qui permet l'engagement, la participation ou qui améliore la relation entre des citoyens et leur gouvernement en améliorant la communication avec ceux-ci et la décision publique. [1]

2.2 Etude de l'état de l'art des applications de la Civic Tech

Nous avons établi différents critères de comparaison entre applications et offre du marché qui sont indiqués dans les colonnes du tableau ci-dessous.
Ce tableau est décomposé selon les différents sites Web sur lesquels nous nous sommes rendus selon l'ordre suivant :

— Maire et Citoyens

— Democraciaos

— Cityzenmap

— Fluicity

Niveaux d’importances : information, incitation, action	Atouts	Limites, faiblesses et risques de l’application
Echanger avec les administrés	Facile d’utilisation	Public limité (jeunes, éduqués, connectés)
Sonder la population	Couvre beaucoup d’options	Fracture culturelle, numérique
Prendre en compte l’opinion majoritaire	Possibilité pour les commerçants locaux de faire leur pub	Option premium
Alerter les habitants de la commune des dangers imminents	Adaptation selon la population de la ville	Coût
Offrir à la population un outil de communication communale à la hauteur de leurs besoins		
Echanger avec les administrés	Beaucoup de widget différents	Public limité (jeunes, éduqués, connectés)
Sonder la population	Partipatory Budget	Fracture culturelle, numérique
Offrir à la population un outil de communication communale à la hauteur de leurs besoins	Public Consultation	Besoin d’une équipe pour gérer et mettre en place les différents procédés
Offre à la population de nombreuses démarches possibles	Crowd Law Making	Coût
	Goal tracking/Voting of Authorities	
Echanger avec les administrés	Une idée peut être exploitée	Public limité (jeunes, éduqués, connectés)
Sonder la population concernée par le projet	Gratuité ; inspiration par d’autres projets dans des villes différentes	Fracture culturelle, numérique
Reconnaissance d’une demande globale par les élus	Outil de communication ludique pour informer les habitants	Peu de projets mis en oeuvre
Offrir à la population un outil de communication communale à la hauteur de leurs besoins	Interface simple pour gérer ou créer des projets sur le territoire et suivre l’activité des citoyens	Manque d’un impact important de la part des projets
	Récolte le soutien et les avis des habitants	Coût
Informers les citoyens	Engager un vrai dialogue avec les citoyens	Public limité (jeunes, éduqués, connectés)
Inciter à une participation plus importante	Gérer la participation citoyenne	Fracture culturelle, numérique
Offre à la population de nombreuses démarches possibles	Décider en fonction des commentaires des citoyens	Besoin d’une équipe pour gérer et mettre en place les différents procédés
Offrir à la population un outil de communication communale à la hauteur de leurs besoins	Etablir un lien de confiance avec les citoyens	Nécessite l’investissement de beaucoup de citoyens pour avoir une légitimité
	Pilotage de la plate-forme citoyenne sans effort	Coût

Chapitre 3

Conception et implémentation : base de données

3.1 Conception : analyse des besoins

* **Définition du système** Pour mettre en place notre arbre de décision, nous avons finalement besoin de 6 tables :

- Une table *solutions* : cette table contient un identifiant unique caractérisant chaque solution, un même nombre entier pour toutes les solutions associées à la même question, un titre et une description de la solution proposée ainsi qu'un nombre de votes permettant de choisir parmi les différentes solutions d'une question.
- Une table *question* : cette table contient un identifiant unique caractérisant chaque question, sa date de création ainsi que le texte relatif à la question et son créateur. Des identifiants entiers pour que les questions puissent se succéder au sein d'une même problématique et les mettre en lien avec la problématique initiale. Un entier *branche_morte* qui caractérise l'état de la branche, 0 pour vivante et 1 pour morte, pour indiquer les branches rendues obsolètes par le backtracking. On note cependant qu'on n'a finalement pas eu le temps d'implémenter un historique pour ces branches, donc la donnée n'est pas utilisée et les branches correspondantes supprimées de la base de données.
- Une table *utilisateurs* : cette table stocke les différentes informations nécessaires lors de l'inscription de chaque utilisateur. Elle contient une adresse email unique caractérisant chaque utilisateur, le nom et le prénom d'un utilisateur ainsi que son mot de passe qui sera inscrit dans la base de données à l'aide d'un cryptage XOR.
- Une table *pb* : cette table contient un identifiant unique caractérisant chaque problématique, la date de création de la problématique, un titre et une description de cette dernière ainsi que l'adresse email de son créateur.
- Une table *msg* : cette table contient un identifiant unique caractérisant chaque message, le contenu du message, l'email de l'utilisateur, l'identifiant de la question sur laquelle le message porte, ainsi que la date d'envoi du message.
- Une table *votes* : cette table stocke des données qui permettent d'enregistrer un vote. Pour effectuer cet enregistrement, la table contient l'email de l'utilisateur, ainsi que les identifiants de la solution et de la question votée.

On intègre des contraintes d'intégrité au système :

Nom de la table	Contraintes
solutions	<i>id</i> comme clé primaire <i>question_id</i> comme clé étrangère de la table question(<i>id</i>)
question	<i>id</i> comme clé primaire <i>auteur_email</i> comme clé étrangère de la table utilisateurs(email) <i>question_parent_id</i> comme clé étrangère de la table question(id) <i>pb_parent_id</i> comme clé étrangère de la table pb(id) <i>branche_morte</i> avec une valeur par défaut de 0
utilisateurs	<i>email</i> comme clé primaire
pb	<i>id</i> comme clé primaire <i>utilisateur_email</i> comme clé étrangère de la table utilisateurs(email)
msg	<i>id</i> comme clé primaire <i>utilisateur_email</i> comme clé étrangère de la table utilisateurs(email) <i>question_id</i> comme clé étrangère de la table question(id)
votes	<i>utilisateur_email</i> comme clé étrangère de la table utilisateurs(email) <i>solution_id</i> comme clé étrangère de la table solutions(id) <i>question_id</i> comme clé étrangère de la table question(id)

La schéma relationnel de la base de données est représenté sur la figure suivante :

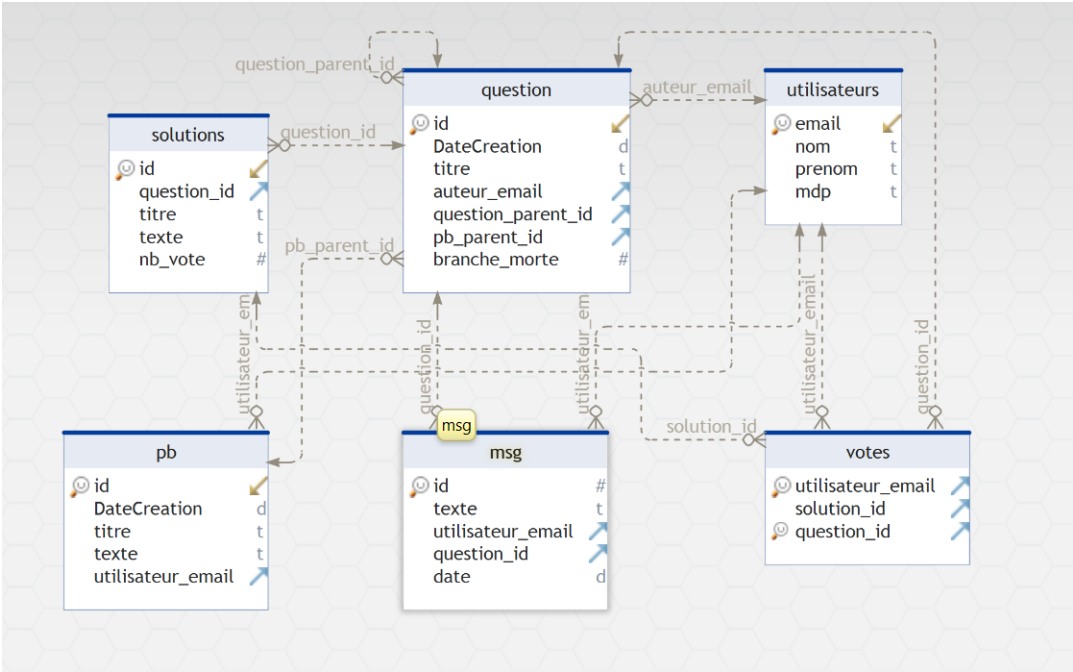


FIGURE 3.1 – Visualisation du schéma relationnel

3.2 Implémentation de la base de données

Pour créer et réinitialiser la base de données, on a établi divers fichiers tels que :

- Un fichier tables.sql contenant le schéma de la base de données, rédigé à la main.
- Divers fichier csv contenant des données d'initialisation que l'on peut insérer dans la base d'une simple commande.
- Un fichier maintenance récapitulant les actions à effectuer pour avoir la base de données opérationnelle et remplie (on aurait pu automatiser l'opération en python mais le besoin ne s'est pas fait sentir.)
- La base de données, database.db que l'on peut facilement explorer grâce au logiciel DB Browser.

Un fichier fonctions pratiques inclut toutes les fonctions interrogeant la base de données.

Chapitre 4

Conception et implémentation : serveur web

4.1 Structure de notre serveur Web

Pour notre application, nous mettons en place plusieurs pages, l'enchaînement entre les pages est décrit sur la figure ci-dessous. Sur toutes les pages, nous mettons en place une barre de navigation permettant d'accéder à tout instant aux pages : Accueil, Login, Problématiques, Aide.

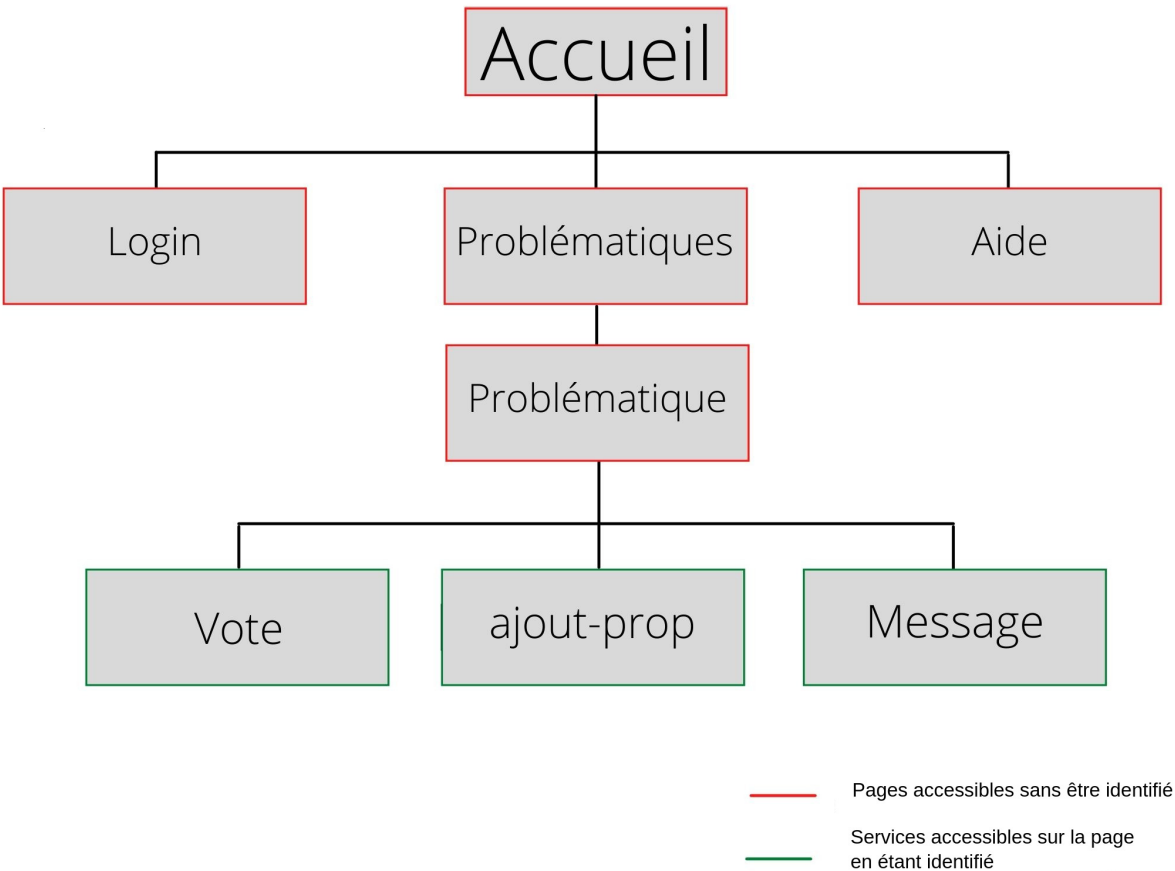


FIGURE 4.1 – Diagramme de navigation

Description des pages :

- La page d'Accueil : Cette page inclut une description de l'application ainsi qu'un bouton permettant de se diriger directement vers la page Problématiques.
- La page Login : Elle permet aux utilisateurs de se login, pour cela, l'authentification requiert un mail et un mot de passe. L'inscription requiert un nom, un prénom, un mail et un mot de passe. Cette authentification repose sur une fonction python *register* appartenant au chapitre suivant.
- La page Logout : Elle permet aux utilisateurs de se logout. Elle requiert de s'être login avant. Elle renvoie directement sur la page Accueil.

- La page d'Aide : Elle permet aux utilisateurs de comprendre via un exemple l'utilisation de chaque fonctionnalité.
- La page Problématiques : Cette page est composée d'une liste de toute les problématiques proposées. Toutes ces problématiques sont des liens permettant d'accéder aux pages Problématique correspondantes.
- La page Problématique : Elle est accessible via la page Problématiques, chaque problématique a sa propre page. Il y a toutes les questions posées et solutions trouvées pour une problématique, les questions ou solutions proposées du moment avec la possibilité d'en ajouter et le choix de vote, ainsi qu'un espace commentaire.
- Les pages Erreur : Elles s'affichent quand il y une erreur.

Croquis des pages :

Les différents croquis des pages sont représentés sur les figures 7.5 et 7.6.

4.2 Implémentation

Pour l'implémentation, l'équipe projet utilise le module Flask permettant de lancer un serveur Web en local. Nous utilisons aussi le langage python, html, CSS et JavaScript pour créer les différentes pages.

Interface

Victor conçoit une barre de navigation qui sera visible sur toutes les pages du sites. En effet, cette dernière a pour objectif de simplifier l'utilisation de utilisateur en proposant des boutons qui redirigent vers les différentes pages lui permettant donc d'accéder aux principaux contenus du site Web. Cette barre de navigation fait l'objet de la figure suivante :



FIGURE 4.2 – Barre de navigation

Implémentation route par route :

- Sur toutes les routes nous utilisons des templates html
- Sur la route Problématiques, nous mettons en place une fonction capable d'afficher la liste des différentes problématiques dans la base de données ainsi qu'un bouton permettant d'en ajouter.

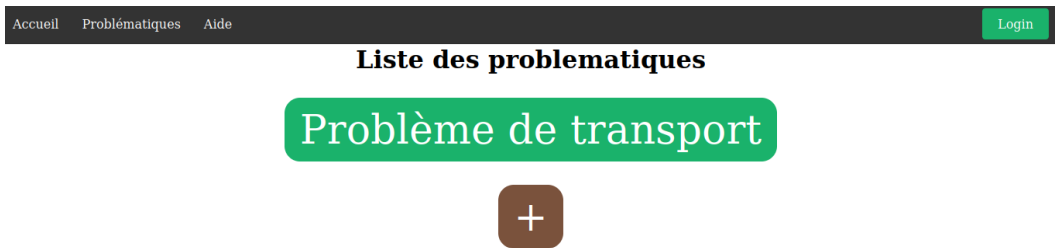


FIGURE 4.3 – Rendu de la page Problématiques

- Sur la route Problématique, comme précisé dans la base de données, chaque problématique est associée à un identifiant unique qui est un nombre entier et qui apparaît dans la route. Sur cette page, on peut voir les différentes questions-solutions associées à cette dernière. Cette structure représente une structure en arbre, qui est en constante évolution et qui peut être croissante comme décroissante. Il est nécessaire pour un utilisateur de se connecter via le bouton Login de la barre de navigation pour pouvoir voter, ajouter une question ou une solution et envoyer des messages. Un vote se fait en cliquant sur le bouton question ou solution désiré et s'affichera pour l'utilisateur en brun (plutôt que jaune). La victoire au vote de chacune des questions et des solutions passées est visible via un pourcentage s'affichant à la droite de celles-ci.

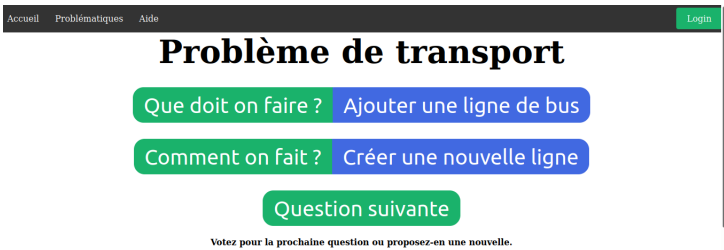


FIGURE 4.4 – Exemple de la problématique "problème de transport"

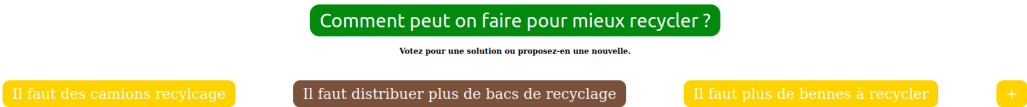


FIGURE 4.5 – Exemple de vote bruns et jaunes



FIGURE 4.6 – Exemple des pourcentages

- La route ajout-prop (où prop = proposition est une question ou une solution) est associé à une problématique et donc elle dépend de cette-dernière route. Elle est sur la page Problématique et permet à tout utilisateur connecté d’ajouter une proposition. Cet ajout requiert un titre et un texte (un descriptif) qui sont demandés.

FIGURE 4.7 – Formulaire d’ajout d’une proposition

- La route Login permet à tout utilisateur de se connecter ou de s’inscrire. Pour cela, il est mit en place un système de login décrit dans la base de données. Ainsi, via un formulaire de connexion, il est possible pour tous les utilisateurs de se connecter.

Si vous avez déjà un compte, connectez vous

Sinon inscrivez vous !

FIGURE 4.8 – Formulaire de connexion

Chapitre 5

Conception et implémentation : algorithmes de traitement

5.1 Fonctions usuelles

Pour la conception de notre application, nous avons besoin de fonctions basiques permettant d'effectuer des requêtes simples. Pour cela, nous décidons donc de créer un fichier *fonctions_pratiques* en python qui contient une trentaine de fonctions, ce qui va nous permettre ensuite de les appeler facilement lors de leur utilisation par le serveur Web.

5.1.1 Analyse des principales fonctions avec traitement de données

Récupération de toutes les solutions/questions

Dans ces fonctions, on récupère grâce à des requêtes SQLite, pour chaque question, la liste des solutions associées à cette question. On souhaite, pour chacune de ces solutions, ajouter le pourcentage de vote qu'elles ont reçu.

Ajout pourcentage de vote

On effectue d'abord un parcours de la liste de choix, nommé *choix*, on ajoute le nombre de votes récoltés par chacun des *c* choix dans un accumulateur nommé *nb total vote*. Ensuite, on effectue un second parcours au cours duquel on modifie chaque *c* choix en le transformant en liste, ajoutant le pourcentage et le retransformant en tuple. Ce changement de type est coûteux. La complexité totale est de l'ordre de $\mathcal{O}(n * m)$ où *n* est la longueur de *choix* et *m* est la moyenne des longueurs des *c* à l'intérieur de *choix*, hors *m* est difficile à calculer car chaque *c* choix contient lui-même plusieurs *string* qui influe sur sa taille.

Création de problèmes/solutions

Pour ces fonctions, on ajoute dans la base de données *database.db* un problème ou une solution via des requêtes SQLite.

On crée un nouvel identifiant pour ce problème ou cette solution et on insère dans la base de données un titre et une description. Seule la création de problème est associée à un utilisateur.

Extension de l'arbre de décision

Il existe différentes fonctions pour augmenter la taille de l'arbre.

Une fonction permet à chaque utilisateur de voter, via un compteur dans la base de donnée. En effet, un utilisateur peut voter au maximum pour une unique solution/question, il faut donc mettre à jour la base de données sur la table utilisateurs et aussi du côté solutions/questions.

Ainsi, grâce à une fonction qui permet de renvoyer la solution qui a reçue le plus de vote, il est possible de connaître la solution choisie pour une problématique.

Alors, une fonction *Etend_Branche* est capable d'augmenter la taille de l'arbre en créant une nouvelle question selon le choix des utilisateurs et aussi propose une solution de retour en arrière appelée "*Backtracking*" à chaque étape.

5.1.2 Division d'un texte

Il est aussi utile d'avoir une fonction capable d'insérer un séparateur à l'intérieur d'une chaîne de caractères. En effet, la fonction *Break_Text* est utilisée pour le formatage du texte dans l'affichage du chat des utilisateurs. Cette fonction est capable d'insérer un "
" à l'intérieur d'une chaîne de caractères selon un espace choisi.

5.1.3 Récupération du fil des questions/solutions/messages

Ici, on part de la première question créée lors de l'initialisation de la problématique. On la reconnaît car elle a pour parent elle-même. On va ensuite chercher récursivement les enfants des questions. La condition d'arrêt est qu'une question n'ait pas d'enfant, cela correspond à la question actuelle et c'est la dernière que l'on doit récupérer.

La complexité est linéaire de l'ordre de la hauteur de l'arbre.

De même, pour récupérer toutes les solutions/messages associés à une question, il suffit d'effectuer une requête SQLite en connaissant l'identifiant de la question.

5.1.4 Vérification de la validité de l'email

On découpe le *string* passé en paramètre pour vérifier qu'il est de la forme (en regex) :
`^[^@]*@[^@]*\.(com|fr|eu|net)$`

5.1.5 Cryptage et decryptage XOR

Comme indiqué dans le fichier, ces fonctions ont été développées lors d'un projet de première année de CPGE par CHANEL Valentin en collaboration avec FUCHS Léon aux lycées Claude Louis Berthollet.

Les complexités sont en temps linéaires par rapport à la taille du *string* passé en entier.

5.2 Backtracking

Revenir en arrière sur une problématique se fait en deux temps : les utilisateurs votent pour initier la phase de backtracking, qui elle permet de voter pour les différents points de retour possibles, ces-derniers étant les solutions votées au fil du problème.

Pour cela, nous avons besoin de deux fonctions pythons : une qui va initialiser le backtracking, et une autre qui va le réaliser. La première insère dans la base de données les solutions avec une description qui indique l'id de la question correspondante, ce qui sera très utile pour la fonction suivante qui saura que ce sont toutes les questions postérieures à celle inscrite qu'il faudra supprimer. Elle modifie ensuite l'id de cette question pour, ce qui la rend en pratique nouvelle car l'id est la clé primaire.

Chapitre 6

Tests et performances

6.1 Tests sur la base de données

Comme on peut le voir dans la figure 3.1, nous avons intégré les dépendances fonctionnelles à la base de données.

6.2 Tests sur les algorithmes

On va utiliser la librairie `pytest` et créer, pour les fonctions pour lesquelles c'est pertinent, des fichiers exécutables de test.

Nous allons suivre la philosophie du Right - BICEP (extrait de cours de CS54)

- > Right : Are the results right ?
- > B : Are all the boundary conditions CORRECT ?
- > I : Can you check inverse relationships ?
- > C : Can you cross-check results using other means ?
- > E : Can you force error conditions to happen ?
- > P : Are performance characteristics within bounds ?

Ajout de pourcentage de vote

On effectue une batterie de test assez extensive, notamment pour vérifier la correcte levé d'exception lorsque les arguments sont incorrectes. Il est difficile de gérer les conditions d'erreurs à cause du type complexe de *solution*.

On utilise les fonctions *sample* et *tuple sample* pour générer aléatoirement des *solutions* et effectué des tests dessus. On arrondit les pourcentages à deux chiffres après la virgule, on va donc tolérer une incertitude l'ordre de 0.01 x longueur de l'argument plus une toute petite marge.

Fonction XOR

Encryptage XOR :

On effectue les tests sur le type mais aussi sur l'appartenance des lettres à l'alphabet (sans accents) ou aux chiffres.

On utilise également une fonction fournie lors de l'examen 4 pour générer des *strings* aléatoires et vérifier avec la fonction décryptage les relations inverses.

Decryptage XOR :

De même, on effectue des tests sur le type, la non-nullité de la clef et des cas de bases. On réutilise la fonction précédente de génération de *strings* pour vérifier la propriété de réversion.

En effectuant des tests, on remarque que certains échouent car en générant un *string* aléatoire en hexadécimal, on n'est pas sûr qu'il ai d'équivalent dans le système utf-8, ce qui provoque des erreurs.

Le fait que le test ai échoué est intéressant mais heureusement vu que nous décryptons uniquement des choses que nous cryptons nous même, il n'y aura pas d'erreur dans le contexte d'utilisation de l'application.

Valide Login

On effectue des tests sur la manière dont cette fonction détecte la construction des emails, selon la fin de ce-dernier ou encore la présence du symbole "@". Il est facile de trouver des cas d'erreurs basiques au vu de la détection mise en place par la fonction.

6.3 Performances

On va tester les performances de la fonction de cryptage XOR grâce aux librairies python datetime,matplotlib et random. On génère des clefs et des textes et on regarde le temps d'exécution de l'algorithme. Les abscisses sont les tailles des string et clef, l'ordonnée est le temps d'exécution en microsecondes.

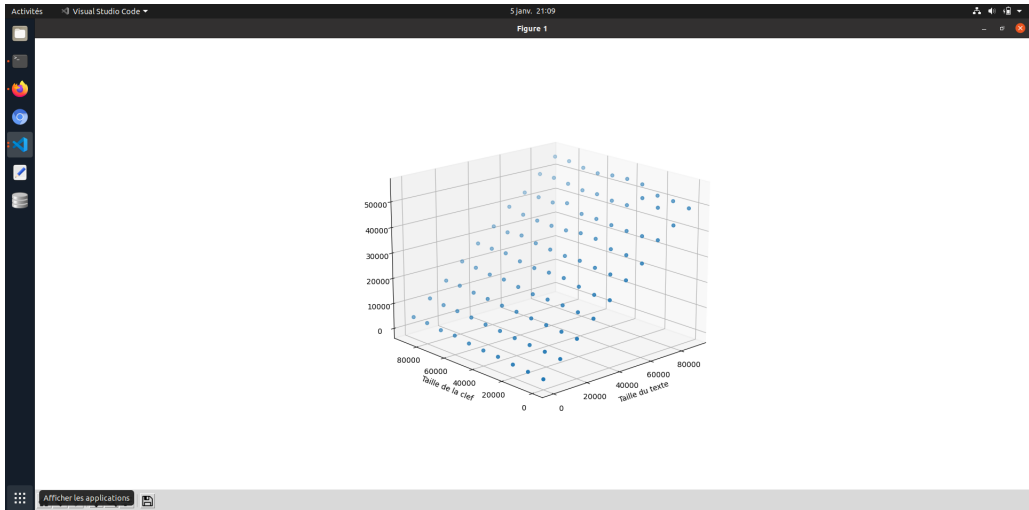


FIGURE 6.1 – Graphique des performances

Chapitre 7

Gestion de Projet

7.1 Composition de l'équipe du projet

L'équipe se compose de quatre étudiants en première année :

- VOISIN Victor
- CHANEL Valentin
- ELAZIZ Elyas
- PALLET Thomas

Valentin est nommé chef de projet, il sera responsable de suivre et de superviser le projet.

L'équipe s'est réunie dans les locaux de TELECOM Nancy pour définir les objectifs, répartir le travail et aussi le réaliser. Lors des vacances scolaires, l'équipe s'est réunie à travers la plateforme Discord.

Les éléments de programmation ont été créés, partagés et utilisés à partir du Git fourni par l'école. De même, pour la rédaction du rapport, l'équipe a utilisé le Leaf fourni par l'école.

7.2 Outils de gestion de projet

7.2.1 Définition des objectifs

On utilise la méthode SMART pour définir nos objectifs :

	Critère	Indicateur
S	Spécifique	L'objectif est clairement défini.
M	Mesurable	L'objectif est mesurable, par des indicateurs chiffrés ou des livrables.
A	Atteignable	L'objectif doit être motivant pour l'équipe 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 défini dans les temps : avec une date de fin.

7.2.2 Prise en compte des risques

Nous avons établi la matrice SWOT suivante :

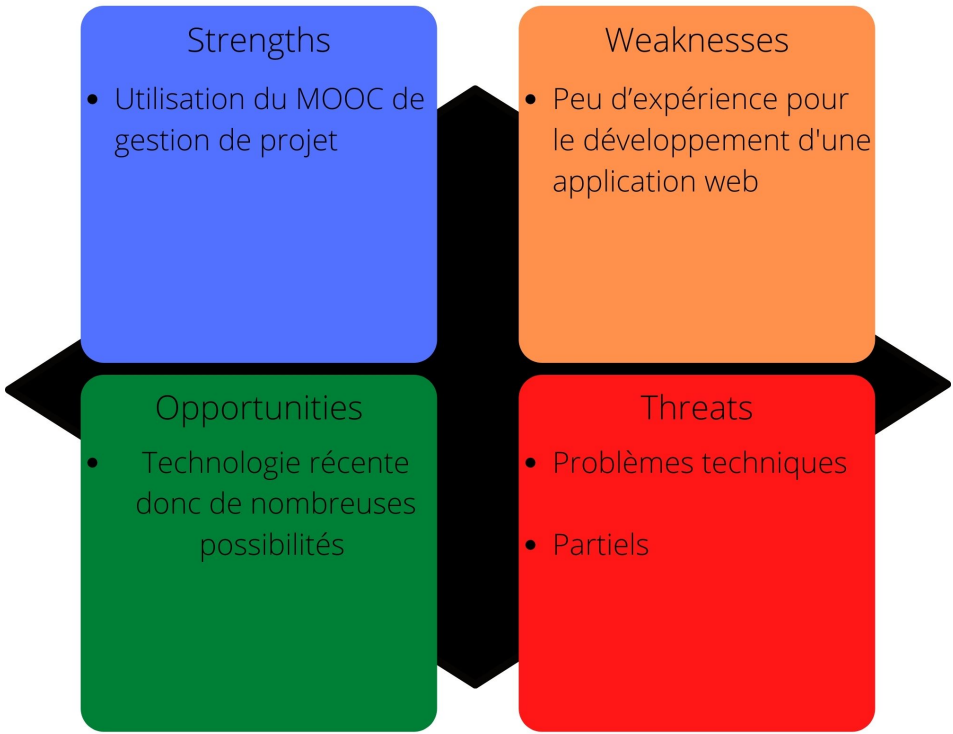


FIGURE 7.1 – Matrice SWOT

7.3 Organisation du projet

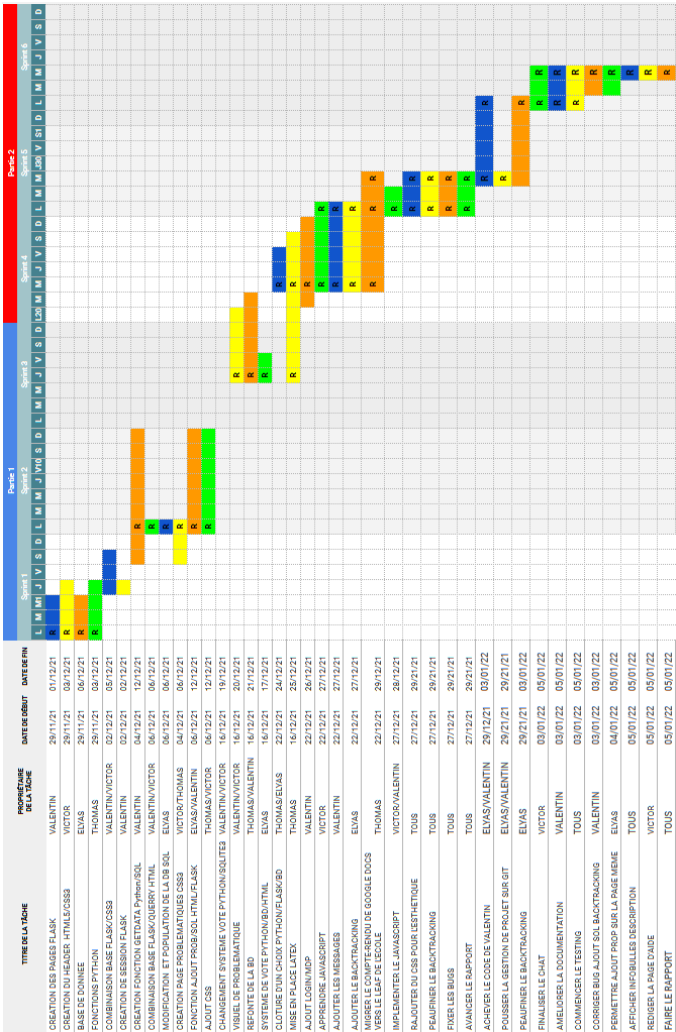


FIGURE 7.2 – Diagramme de Gantt

7.4 Comptes-rendus des réunions

7.4.1 3 Novembre 2021

Motif : Démarrage du projet		Lieu : TELECOM Nancy en salle 1.18
Présent	Absent	Durée : 2h
VOISIN Victor		
CHANEL Valentin		
ELAZIZ Elyas		
PALLET Thomas		

Ordre du jour

- Mise en place de la gestion de projet
- Réflexion sur le sujet

Informations échangées

- Mise en place de la gestion de projet

Planification des réunions

L'équipe souhaite se réunir au minimum une fois par semaine. La communication des réunions se fera à travers la plateforme discord ou via l'application messenger.

Esquisse du diagramme de Gantt

Nous utilisons un modèle de diagramme de Gantt proposé par Google Sheets. Nous adaptons ce dernier en le décomposant en deux cycles, chacun composé de 3 semaines.

Mise en place d'une matrice SWOT

Nous remplissons ensemble notre matrice SWOT.

Objectif mi-projet

L'équipe projet se fixe comme objectif d'obtenir d'ici 3 semaines une application fonctionnelle, puis ensuite de développer l'aspect esthétique.
- Réflexion sur le sujet

On met en place un *brainstorming* sur les possibilités que le projet peut potentiellement nous offrir, en se basant sur l'état de l'art. On souhaite empêcher un problème qui est que la participation des citoyens est souvent réduite à voter "oui" ou "non" à des questions beaucoup trop complexes pour être ramenés à des choix binaires que les utilisateurs pourraient proposer.

On se met d'accord sur l'idée d'un arbre de décision où chaque citoyen peut voter à chaque nœud. Le nombre de feuilles disponibles casse la binarité des choix, de plus l'arbre permet de poser successivement des questions sur différentes modalités du problèmes.

TO-DO LIST

- Établir l'état de l'art

7.4.2 10 Novembre 2021

Motif : Avancement état de l'art		Lieu : TELECOM Nancy devant la machine à café	
Présent	Absent		Durée : 15min
VOISIN Victor			
CHANEL Valentin			
ELAZIZ Elyas			
PALLET Thomas			

Ordre du jour

- * Confrontation des différentes recherches

Informations échangées

- Confrontation des différentes recherches

L'équipe met en commun les recherches de chacun. Nous utilisons un fichier Google Sheet et mettons en place un tableau. Nous souhaitons démarquer notre application par rapport aux applications appartenant à la Civitech déjà disponibles.

TO-DO LIST

- Affiner nos réflexions sur notre projet

7.4.3 24 Novembre 2021

Motif : Rédaction du document de conception		Lieu : TELECOM Nancy en salle 1.18	
Présent	Absent		Durée : 2h30
VOISIN Victor			
CHANEL Valentin			
ELAZIZ Elyas			
PALLET Thomas			

Ordre du jour

1. Suite de la gestion de projet
2. Rédaction du document de conception de l'application

Informations échangées

- Suite de la gestion de projet

SCRUM Master
L'équipe souhaite adopter les méthodes agiles pour la gestion de projet vu avec le MOOC gestion de projet avec la spécialité SCRUM. Nous devons donc élire un SCRUM Master.

Candidat	Valentin Chanel
Nombre de votant	4

Abstention	Blanc	Pour	Contre
0	0	4	0

Compte-rendu des votes

Valentin est désigné comme SCRUM Master.

Matrice SWOT

L'équipe établit ensemble une matrice SWOT.

Etat de l'art

L'équipe valide le tableau final.

- Rédaction du document de conception de l'application

Cahier des charges Esquisse de la base de données Croquis des pages web

L'équipe débat sur la forme de l'application pour finalement aboutir à un arbre de décision. Nous souhaitons construire une application aux nombreuses caractéristiques, ce qui correspondra a notre cahier des charges :

- Répondre à des problèmes complexes d'une population locale avec l'avis des citoyens et d'un système de choix de solutions face à ces problèmes.
- Permet de mettre en avant des exigences ou besoins pour aboutir à des actions concrètes.
- Possibilité d'invalider une proposition finale, sous l'accord de la majorité.
- Possibilité de remonter dans l'arbre de décision pour ne pas recommencer totalement le problème initialement posé.
- Permet à tous les citoyens de proposer leur réponse aux questions et de les soumettre à un vote.
- Possède un système de forum où les citoyens pourront débattre de chaque proposition.

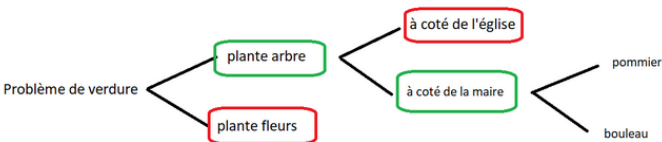


FIGURE 7.3 – Arbre de décision

Esquisse de la base de données

Nous avons établi initialement 5 tables dans la base de données, où les relations sur ces tables sont données sur la figure suivante :

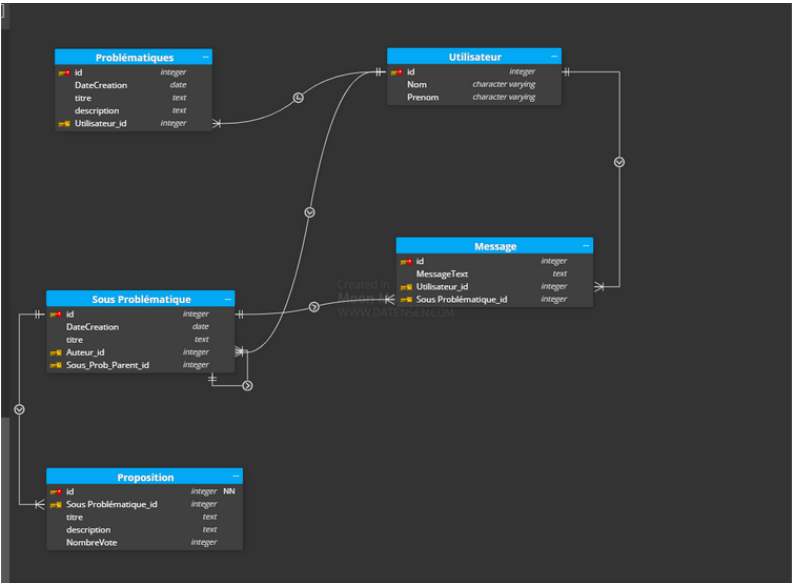


FIGURE 7.4 – Esquisse de la base de données

Croquis des pages webs

Nous avons établi le croquis des différentes pages web de notre application. Nous expliquerons les liens entre ces dernières dans le chapitre suivant.



FIGURE 7.5 – Page d’Accueil et page Problématiques

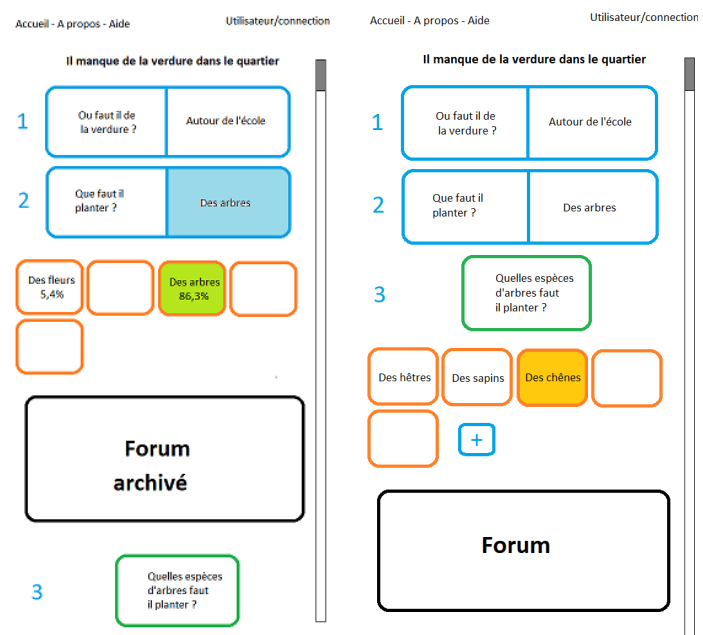


FIGURE 7.6 – Page Problématique

TO-DO LIST

— Préparer l’environnement de travail

7.4.4 29 Novembre 2021

Motif : Avancement état de l'art		Lieu : TELECOM Nancy en salle 1.18	
Présent	Absent		Durée : 1h
VOISIN Victor CHANEL Valentin ELAZIZ Elyas PALLET Thomas			

Ordre du jour

- 1. Préparation de l'implémentation

Informations échangées

- Préparation de l'implémentation
 - Nous utilisons une WBS (Work Breakdown Structure) avec quatre lots de travail :
 - Création de l'arborescence de pages, de la navigation entre elles.
 - Création des pages avec un minimum de graphismes.
 - Création de la base de données avec toutes les contraintes.
 - Création des fonctions qui interagissent avec la base de données en prenant des arguments depuis la page web. (python, Flask)

TO-DO LIST

- Objectifs du premier sprint

7.4.5 6 Décembre 2021

Motif : Fin Semaine 1		Lieu : TELECOM Nancy en salle 1.18	
Présent	Absent		Durée : 2h
VOISIN Victor CHANEL Valentin ELAZIZ Elyas PALLET Thomas			

Ordre du jour

- 1. Compte rendu de la semaine
- 2. Planification de la suite

Informations échangées

- Compte rendu de la semaine
 - Création de la base du projet*
 - L'équipe a pu coder pendant 4h en présentiel, le projet a donc beaucoup avancé. Un système de navigation de page en page est opérationnel sur Flask.
 - On créé rapidement une en-tête CSS qui sera en haut de toutes les pages.
 - On créé des fonctions qui récupèrent des informations SQL.
 - Base de donnée*
 - Pendant que l'on préparait les pages, on se rend compte que le schéma relationnel de la BD pourra poser problèmes lorsque l'on voudra ajouter / récupérer des éléments. On réalise qu'on doit le modifier, on se met d'accord sur les changements.
- Planification de la suite
 - Menace potentielle*

Les partiels représentent une réel menace pour notre organisation. On choisit donc de faire la prochaine réunion en présentiel le 16 avant que chacun rentre chez soi.
 - Objectifs*

On souhaite que pour la prochaine réunion, toutes les fonctionnalités de base soit implémentées, avec un minimum de CSS.

TO-DO LIST

- Modifier le schéma de la base de données
- Implémenter l'ajout de propositions et de problématiques ainsi que le vote
- Faire un minimum de CSS pour ne plus avoir de pages basiques.
- Continuer à développer les fonctions récupérant des données

7.4.6 16 Décembre 2021

Motif : Réunion d'Avancement		Lieu : Telecom Nancy
Présent	Absent	Durée : 1h30
VOISIN Victor		
CHANEL Valentin		
ELAZIZ Elyas		
PALLET Thomas		

Ordre du jour

1. Compte rendu de la semaine
2. Planification de la suite

Informations échangées

- Compte rendu de la semaine
 - Temps de travail*

Les partiels ont été très prenants mais il y a eu du progrès, certains membres ont codé mais n'ont pas encore push ou ont partagé par un autre moyen que Git. On se met d'accord pour éviter cela à l'avenir.

Changement du nom du fichier python

Elyas souligne qu'en renommant le fichier app.py, cela nous éviterait de pointer le fichier python à chaque fois que l'on veut lancer Flask. La modification est acceptée.

Compte rendu

Suite à la formation sur le rapport de projet organisé par une élève de troisième année, on se rend compte qu'il nous manque beaucoup de données, qu'on devra apprendre un nouveau langage, le LATEX, et on prend conscience que le rapport demandera beaucoup de travail.

Objectifs atteints

Tous les objectifs de la to-do list ont été atteint.

- Planification de la suite

Problème clef primaire

On décide modifier la clef primaire d'utilisateur qui devient un email, les précédentes clefs primaires étant difficiles à gérer.

Problème de nommage

On se rend compte que les noms *propositions* et *sous_problématique* sont porteur de confusion, on décide de les changer.

On programme la prochaine réunion pour très bientôt sur discord. On a le temps de travailler donc on décide de raccourcir les cycles.

TO-DO LIST

- Finir d'implémenter les fonctionnalités de base, peaufiner le vote.
- Faire les changements sur la base de données et les requêtes.
- Continuer à ajouter du CSS, notamment sur la page Problématique.
- Appliquer les changements d'informations sur les utilisateurs au système de session.
- Utiliser le système de session pour gérer les cas d'incohérence si l'utilisateur n'est pas connecté.
- Mettre en place l'environnement LaTeX suite au conseil reçu.

7.4.7 22 Décembre 2021

Motif : Réunion d'Avancement		Lieu : Discord
Présent	Absent	Durée : 2h
VOISIN Victor		
CHANEL Valentin		
ELAZIZ Elyas		
PALLET Thomas		

Ordre du jour

1. Compte rendu de la semaine
2. Planification de la suite

Informations échangées

- Compte rendu de la semaine

Inscription / Login

Valentin présente le système d’inscription, de login et soulève la question du stockage des mots de passe. L’équipe se met d’accord pour stocker les mots de passes dans la base de données avec le cryptage XOR qui est familier à Valentin.

Cloture

Elyas propose le système de clôture manuelle d’un vote et nous explique comment il compte l’implémenter.

Changement de la BD

Les changements sur la base de données pour qu’elle soit plus claire on été effectués, il y a eu beaucoup de renommage et donc beaucoup de tests à refaire pour vérifier qu’il n’y avait pas de problème.

Différenciation connecté / non connecté

L’équipe souligne la grosse différenciation du site selon que l’on soit connecté ou non.

SCRUM

On s’est rendu compte des limites de SCRUM qui n’est peut être pas adapté à des petits projets. On a décidé de le continuer tant bien que mal, sans se focaliser dessus.
- Planification de la suite

Après débat, l’équipe est d’accord avec l’idée de Victor d’utiliser JavaScript pour rendre le site plus dynamique.

Au sujet du backtracking, après réflexion sur la méthode d’implémentation, on convient qu’on commencera par se contenter d’un système sans mémoire qui supprime définitivement les branches remontées à l’issu du sprint.

On programme la prochaine réunion pour le 27/12, joyeux Noël à tous !

TO-DO LIST

- Continuer d’améliorer le système de vote
- Ajouter la clôture manuelle
- Apprendre JavaScript
- Ajouter le système de messages
- Ajouter le BackTracking
- Migrer le Compte-Rendu de Google Docs vers le LEAF de l’école
- Optionnel : rédiger la page d’aide

7.4.8 27 Décembre 2021

Motif : Réunion d’Avancement		Lieu : Discord
Présent	Absent	Durée : 1h
VOISIN Victor		
CHANEL Valentin		
ELAZIZ Elyas		
PALLET Thomas		

Ordre du jour

1. Compte rendu de la semaine
2. Planification de la suite

Informations échangées

- Compte rendu de la semaine

Messages

On présente l’outil des messages, on remarque un petit bug qui est corrigé.

Backtracking

Elyas nous présente ses avancements dans le système de backtracking, il y a un léger problème mais le principal est fait.

Ajout de JavaScript

Comme prévu dans le réunion précédente, Victor nous montre les progrès qu’il a fait sur le JavaScript sur sa démo personnelle, l’équipe est satisfaite mais il faut maintenant l’implémenter sur le projet.

Covid et avancement du compte rendu de Projet

Thomas a bien avancé, il a mis en place des templates avec des placeholders, il est facile de rédiger un nouveau compte-rendu de réunion ou autre. Il nous informe néanmoins qu’il a contracté la COVID et que son état de santé risque d’affecter son aptitude à s’investir dans le projet autant qu’il le souhaiterait. L’équipe lui souhaite un bon rétablissement.

Objectifs atteints

Les objectifs principaux de la to-do list ont été atteints.

- Planification de la suite

On définit ce que chacun doit faire, les objectifs sont dans l’ordre l’esthétique, la correction de bug et la prise en charge d’erreur.

On programme la prochaine réunion pour le 29 décembre.

TO-DO LIST

- Implémenter le JavaScript.
- Rajouter du CSS pour l’esthétique.
- Peaufiner le Backtracking.
- Corriger les bugs et gérer les cas d’erreurs.
- Optionnel : rédiger la page d’aide.

7.4.9 29 Décembre 2021

Motif : Réunion d’Avancement		Lieu : Discord
Présent	Absent	Durée : 1h
VOISIN Victor		
CHANEL Valentin		
ELAZIZ Elyas		
PALLET Thomas		

Ordre du jour

1. Recontruction des fonctions interrogeant la base de données

- 2. Visuel du site
- 3. Gestion de projet et mail d'Olivier Festor

Informations échangées

- Compte rendu du début de semaine

Recontruction des fonctions interrogeants la base de données

Lors de sa tâche pour améliorer le backtracking et implémenter le JavaScript Valentin a pris l'initiative de changer les fonctions de base pour qu'elles puissent nous fournir des données plus faciles à exploiter avec le code de Victor. Il nous les explique et prévient qu'elles nécessitent une correction pour être effectives. Elyas propose son aide.

Visuel du site

On estime qu'une fois que le code de Valentin sera achevé, il sera aisé d'obtenir une version convenable du visuel du site avec notre code CSS et JavaScript.

Gestion de projet et mail d'Olivier Festor

On discute de l'email d'avertissement reçu la veille. On constate qu'il y a eu un problème de communication dans l'équipe car bien que la gestion de projet eu été bien faite sur des ressources externes, personne ne la poussait sur le git. On convient qu'on doit en urgence mettre en ordre et pousser les documents.

- Planification de la suite

On fixe l'échéance pour pousser la gestion de projet à cette nuit. Il reste à travailler sur le code pour la fin du sprint samedi.

On programme la prochaine réunion pour le premier janvier.

TO-DO LIST

- Achever le code de Valentin
- Pousser la gestion de projet sur git
- Peaufiner le BackTracking
- Corriger les bugs et gérer les cas d'erreurs
- Optionnel : rédiger la page d'aide

7.4.10 3 Janvier 2022

Motif : Réunion d'Avancement		Lieu : Telecom Nancy
Présent	Absent	Durée : 1h
VOISIN Victor CHANEL Valentin ELAZIZ Elyas PALLET Thomas (en distanciel)		

Ordre du jour

- 1. Branche alternative
- 2. Rédaction du rapport
- 3. Gestion de projet et mail d'Olivier Festor

Informations échangées

- Compte rendu du week-end

Suite reconstruction des fonctions interrogeant la base de données

Elyas a proposé une version alternative des fonctions que Valentin avait poussées la dernière fois, qui revient en arrière sur une partie des fichiers tout en gardant certaines nouvelles fonctions, et offre les variables nécessaires pour implémenter le JavaScript et CSS de Victor. Cela a donc été poussé sur une branche différente. Puis après avoir été relue, ajustée et améliorée par Valentin et le reste de l'équipe, la branche a été rendue branche maître.

Rédaction du rapport

Thomas a avancé la rédaction du rapport, on note notamment le diagramme de navigation qui est très clair.

Gestion de projet et email d'Olivier Festor

On discute du nouveau retour d'Olivier Festor qui est mitigé, qu'on remercie au passage d'avoir pris le temps de revenir sur notre projet. On convient que nos comptes rendus de réunions jusqu'ici ne sont pas satisfaisant en terme de contenu et d'orthographe, et qu'on aurait dû et qu'on devra s'assurer de les rédiger immédiatement après chaque réunion à partir de notes prises durant celles ci. On a décidé de retourner corriger les anciens comptes rendus.

- Planification de la suite

On doit enfin achever d'implémenter le JavaScript et CSS déjà écrit en surmontant les complications Jinja/JavaScript. Il y a également le système de chat dont l'affichage est à retravailler.

Les descriptions de chaque problème, questions et solutions sont bien présentes dans la base de données mais ne sont pas encore visibles pour les utilisateurs. On brainstorme sur une manière ergonomique et esthétique de les afficher, et on choisit les infobulles. Les comptes rendus de réunions doivent être corrigés. On

constate qu'on a beaucoup de fonctions à tester, on pourrait les optimiser et essayer d'en retirer quelques unes au profit de fonctions plus génériques (certaines sont redondantes)

TO-DO LIST

- Implémenter JavaScript CSS
- Finaliser le chat
- Avancer le rapport (chapitres 3, 4 et 5) et faire le bilan personnel
- Corriger les comptes rendus
- Améliorer la documentation
- Commencer le testing
- Rédiger la page d'aide
- Corriger le bug où on peut ajouter une solution lors du backtracking
- Permettre l'ajout d'une proposition sur la page même plutôt que sur une page séparée
- Mettre en place les infobulles pour afficher les descriptions
- Restaurer le travail sur les pourcentages de votes perdu lors du changement de branche

7.4.11 5 Janvier 2022

Motif : Réunion finale		Lieu : Telecom Nancy
Présent	Absent	Durée : 30min
VOISIN Victor		
CHANEL Valentin		
ELAZIZ Elyas		
PALLET Thomas (en distanciel)		

Ordre du jour

1. Vérification de la TODO LIST de la dernière réunion
2. Discussion de ce qui est prioritaire

Informations échangées

- Compte rendu de la semaine
Explication de ce qui a été fait
Chacun explique ce qu’il a fait rapidement, la grande majorité des objectifs ont été atteint.
Objectifs manquées
La documentation et les tests ont été avancés mais pas à un niveau satisfaisant. La page d’aide étant un objectif secondaire de la semaine dernière, elle n’a pas été commencée.
- Planification de la suite
Priorité
L’échéance étant très proche, l’équipe discute des priorités et doit faire des choix
Objectifs principaux
On décide à l’unanimité que la page d’aide sera un objectif prioritaire. L’utilisation de l’application peut porter à confusion : il faut absolument un endroit où son fonctionnement est clairement détaillé.

Le rapport est à finalisé aujourd’hui, en mettant l’accent sur les chapitres 3 et 5.

TO-DO LIST

- Faire la page d’aide
- Avancer les chapitre 3 et 5 du rapport
- Continuer de faire de la documentation / du testing
- Faire des tests de performances de certaines fonctions
- Travailler sur le diagramme de Gantt
- Chaque membres doit rédiger sa part du chapitre 8

Chapitre 8

Bilan du projet

8.1 Bilan global du projet

Travail attendu	Travail réalisé
L'application résultante doit faciliter la démocratie participative locale L'application résultante doit être réalisable dans les délais du projet L'application résultante doit s'appuyer à minima sur : une base de données, des algorithmes de traitement avancés et être accessible via le Web	La condition est remplie mais l'application est vraiment limitée à l'échelle locale. Les délais ont été respectés. Les parties Web et BD sont respectées, la partie algorithme est présente mais un peu faible.

	Points positifs	Points négatifs	Expérience
Gestion de projet	Les comptes rendus n'ont pas été fait le jour même ce qui a causé des problèmes d'organisation.	On a réalisé l'importance de la gestion de projet pour ce qui est de l'organisation.	Accélérer la mise en place de la gestion de projet en équipe.
Écriture du code	Le code fonctionne	On a pas toutes les fonctionnalités souhaitées (sauvegarde des branches mortes)	Renforcer la documentation python régulièrement.

8.2 Bilan du projet membre par membre

8.2.1 VOISIN Victor

Points positifs	J'ai adoré travailler avec les gars,tout le monde s'est bien appliqué et il y avait une super ambiance. J'ai beaucoup appris dans tous les domaines, c'était très enrichissant.
Difficultés rencontrées	Surtout d'organisation, les comptes rendus, le Gantt et les sprints n'étaient pas trop organisés. Petite difficulté pour bien communiquer avec Thomas qui était en distanciel.
Expérience personnelle	C'était plus compliqué pour moi de travailler à distance, sans voir les gars, quelques pertes de motivation mais les réunions en vocal m'ont bien aidé.
Axes d'amélioration	Il faut écrire les comptes rendus et modifier le Gantt le jour même sinon on oublie et c'est l'enfer pour s'y retrouver.

8.2.2 CHANEL Valentin

Points positifs	Le projet était passionnant et c'est très rare que je complète un projet personnel jusqu'à ce stade, c'est très satisfaisant. De plus l'équipe était très bonne.
Difficultés rencontrées	En tant que chef de projet, je n'ai pas assez mis l'accent sur la gestion des comptes rendu et du Gantt. Je me sentais souvent responsable des problèmes que l'on a rencontrés.
Expérience personnelle	J'ai beaucoup appris sur la gestion de groupe mais aussi sur le travail en collaboration, l'organisation. J'ai été obligé de faire confiance aux autre alors que je préfère tout faire moi même d'habitude.
Axes d'amélioration	Il faut que j'améliore ma gestion de groupe et la gestion de projet.

8.2.3 ELAZIZ Elyas

Points positifs	En tant que premier projet c'était une excellente expérience de découvrir le travail d'équipe en informatique, et d'observer la cohésion du groupe
Difficultés rencontrées	La gestion de projet était intimidante donc je priorisais la conception fonctionnelle de l'application. Ce fut une erreur de la négliger, on a perdu du temps à la reprendre plus tard.
Expérience personnelle	J'ai pu devenir bien plus confortable en web, et aussi en git avec les nombreux soucis de conflits, et la gestion des branches.
Axes d'amélioration	Je devrais rédiger la rédaction de gestion de projet, le testing et la documentation au fur et à mesure plutôt qu'à la fin.

8.2.4 PALLET Thomas

Points positifs	Un sujet intéressant et une bonne expérience de travail en équipe ainsi qu'une bonne dynamique.
Difficultés rencontrées	J'ai eu des soucis techniques ainsi que quelques problèmes sur ma machine virtuelle. L'organisation de la gestion de projet était compliquée.
Expérience personnelle	J'ai pu apprendre le langage Latex. J'ai pu progresser en algorithmie.
Axes d'amélioration	Améliorer l'organisation de gestion de projet et régulariser la documentation du rapport.

8.3 Travail réalisé

Étapes	Valentin	Victor	Elyas	Thomas
État de l'art	2h	3h	2h	3h
Gestion de Projet	14h	11h	13h	12h
Conception des algorithmes	44h	41h	46h	36h
Conception du rapport	6h	7h	5h	16h
TOTAL	66h	62h	66h	67h

Bibliographie

[1] L. Segantini, “Qu’est ce qu’une civic tech et que peut-elle apporter ?,” 2020.