

Projet L1 - CMI Informatique
Année universitaire 2023-2024
Encadrant : Baert Anne-Elisabeth



Rapport de Projet -Budizz-

Ambre Akato - Manal Ettahiri - Eléa Weber

05 mai 2024

Contents

1	Introduction	3
1.1	Choix du projet	3
1.2	Choix du nom	3
2	Cahier des charges	4
2.1	Préambule	4
2.2	Description générale du projet	4
2.3	Fonctionnalités principales attendues	4
2.4	Exigences fonctionnelles	4
2.5	Utilisateurs/Système	5
2.6	Plateformes/logiciels utilisés	5
3	Planification et organisation du projet	6
4	Zoom sur le projet	7
4.1	Architecture principale de l'outil	7
4.2	Récupération des données	8
4.3	Traitement des données récupérées	8
4.4	Difficultés rencontrées	8
5	Manuel d'utilisation	9
6	Conclusion	10
7	Bibliographie - Sitographie	11

1 Introduction

1.1 Choix du projet

Au terme de notre première année de CMI Informatique, nous devions réfléchir à un projet informatique à mettre en œuvre.

Désireuses de faire évoluer la cohésion de groupe dans notre promo, nous avons donc eu l'idée d'une plateforme-quiz qui permettrait à une classe, un groupe ou des collaborateurs de sociabiliser plus facilement selon les points communs ou les aspirations de chacun.

Cela permettrait de consolider un début d'esprit d'équipe.

Le but est de répondre à un problème que l'on peut rencontrer au quotidien. Le nôtre tournait essentiellement autour du fait de passer des journées entières avec les mêmes personnes et ce pendant toute une année scolaire, sans jamais oser se parler ou apprendre à se connaître.



Figure 1: Logo Budizz

1.2 Choix du nom

A l'issue de nos différentes réflexions autour du concept de notre application, nous avons eu l'idée d'associer les mots "bud", "buddy" (ami/collègue en anglais) et "quizz" (jeu par questions et réponses), représentant tous deux l'esprit de notre application pour former le mot "BUDIZZ".

2 Cahier des charges

2.1 Préambule

Il est important de mentionner le fait que ce cahier des charges correspond à ce que l'on attendait et ce que l'on espérait faire pour notre projet mais ne correspond pas totalement au résultat final.

Des modifications ont été apportées tout au long du processus de développement de l'application pour correspondre à nos compétences techniques et à nos nouvelles idées.

2.2 Description générale du projet

Notre objectif est de réaliser une application Web permettant aux différents membres d'un groupe de renseigner leurs passions et centres d'intérêts de manière individuelle pour que par la suite les différentes données des utilisateurs soient regroupées pour faire émerger des centres d'intérêts communs.

Le but étant, à la fin, de parvenir à un classement global des personnes avec lesquelles l'on présente le plus de points communs.

Cette application web s'adresse aux lycéens, étudiants, collègues, collaborateurs dans le but de casser la glace lors d'une rentrée des classes, lors d'un séminaire ou pendant un premier jour à l'université.

2.3 Fonctionnalités principales attendues

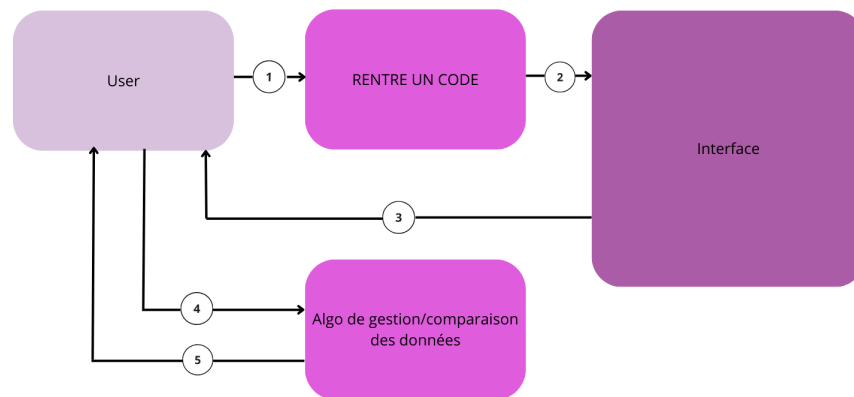
- Accès à la plateforme par toute personne ayant un smartphone à l'aide d'un code
- Accès au renseignement de données
- Algorithmes de gestion/comparaison des données
- Affichage d'un résultat cohérent et global

2.4 Exigences fonctionnelles

- Accès à la plateforme par toute personne ayant un smartphone à l'aide d'un code. Ce dernier redirige directement vers une page web donnant accès aux champs de renseignement des différentes données des utilisateurs.

- Accès au renseignement de données : il est ensuite nécessaire de fournir aux utilisateurs une interface leur permettant de renseigner leurs informations personnelles. Pour ce faire, nous avons développé différentes page web en HTML et CSS comportant les champs nécessaires à remplir par les utilisateurs.

2.5 Utilisateurs/Système



- 1 l'utilisateur rentre le code
- 2 la saisie du code renvoie à une interface utilisateur
- 3 l'interface demande à l'utilisateur de renseigner des données
- 4 l'utilisateur renseigne son prénom, sa ville et ses hobbies
- 5 ces données sont collectées et comparées avec celle des autres utilisateurs pour renvoyer les résultats à l'utilisateur

Figure 2: Schéma illustrant les interactions utilisateur/interface

2.6 Plateformes/logiciels utilisés

Nous avons tout d'abord commencé par créer un répertoire Github pour pouvoir partager nos fichiers. Nous partageons également un Google Drive et un dossier Latex pour ce qui est de la rédaction du rapport. Au niveau du code, nous utilisons VS Codium comme environnement de programmation.

3 Planification et organisation du projet

Diagramme de Gantt

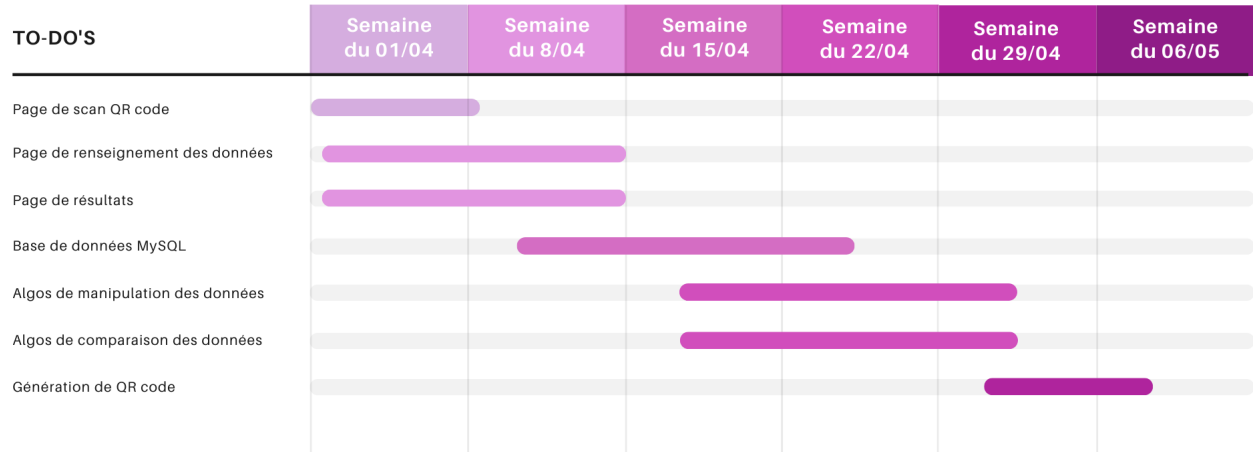


Figure 3: Prévisions initiales des tâches

Initialement, nous avons prévu un planning précis des tâches à accomplir. Mais finalement, certaines tâches se sont avérées plus rapides à faire que prévu, puis d'autres se sont ajoutées et d'autres encore sont devenues inutiles. Notre planning de prévisions est donc devenu arriéré et nous avons décidé de commencer par la partie interface utilisateur.

Cependant, au fur et à mesure de l'avancée du projet, nous avons procédé à des modifications de fonctionnalité. Les ajouts et suppressions seront détaillées dans la partie "Difficultés rencontrées".

Voici ci-après le nouveau planning prévisionnel que nous avons finalement suivi.

Diagramme de Gantt

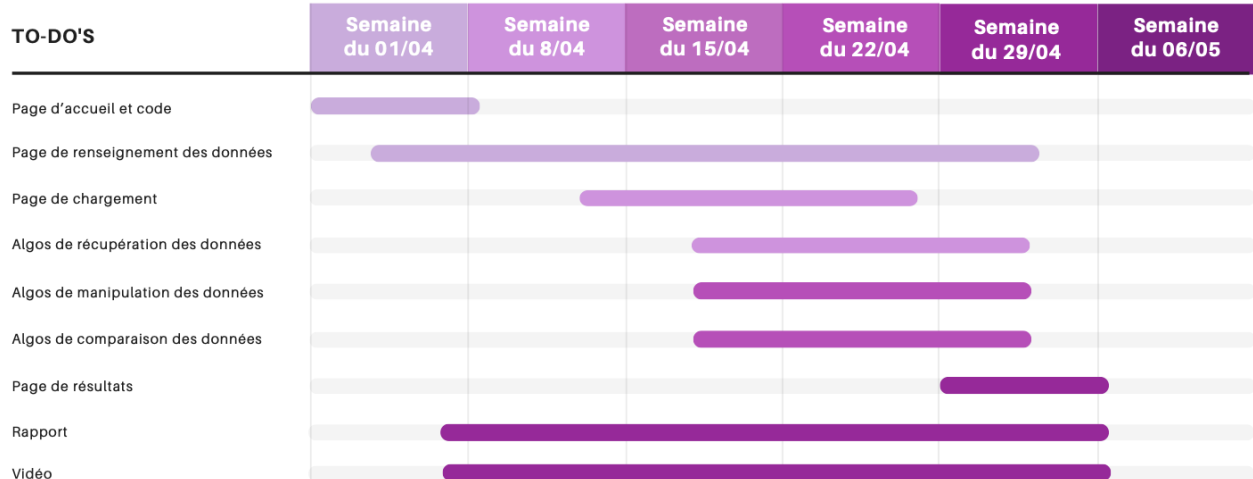


Figure 4: Nouveau planning prévisionnel suivi

4 Zoom sur le projet

4.1 Architecture principale de l'outil

L'application BUDIZZ se découpe en 5 pages web, toutes écrites à l'aide des langages de programmation HTML et Javascript et stylisées en CSS. Elles constituent l'architecture principale de l'application et contribuent en grande partie à l'expérience utilisateur. C'est donc pour cela que nous avons décidé d'y consacrer autant de temps (cf Figure 4).

De plus, dans le but de correspondre à une identité graphique, toutes les pages sont équipées d'un fond de même couleur (en l'occurrence, rose). Certains éléments principaux sont eux aussi roses et d'autres jaunes pour rappeler la couleur du logo. Ce dernier est aussi toujours présent dans l'en-tête des différentes pages pour apporter de la cohérence entre celles-ci.

De manière plus précise, la première page comporte un bouton redirigeant vers la deuxième invitant les utilisateurs à entrer un code dans le but d'obtenir l'accès au formulaire de renseignement des données. Le but à l'origine était d'avoir plusieurs groupes de personnes réalisant l'expérience BUDIZZ en même temps : différents codes auraient permis d'avoir plusieurs sessions ouvertes en même temps. Cependant, par faute de moyens liés à l'hébergement de l'application, nous avons décidé de simplement la faire tourner en localhost sur une seule et même machine.

Ensuite, après avoir entré le code correspondant (dans ce cas-ci, le code est 'BUDIZZ'), les utilisateurs ont accès à la page leur permettant de renseigner leurs informations. Cette dernière est composée d'un formulaire lui-même découpé en 3 parties distinctes. La première partie permet aux utilisateurs de remplir

leur nom ainsi que leur ville à l'aide de champs de texte. La deuxième leur permet, à l'aide de cases à cocher, de renseigner leurs différentes passions, divisées en catégories et en sous-catégories. La dernière partie du formulaire se compose de questions de mise en situation et permettent de faire ressortir la personnalité de l'utilisateur.

4.2 Récupération des données

Une fois le formulaire soumis, les données sont récupérées à l'aide du fichier serveur *node js* "budizz.js". Contrairement à ce que nous avions prévu dans le cahier des charges, nous avons opté pour utiliser à la place du langage PHP le langage JavaScript pour mettre en place le serveur de notre application. *Node.js* est une plate-forme de développement serveur en JavaScript et permet de créer des applications temps-réel, tout en mutualisant le code JS entre client et serveur.

Ainsi, le fichier "budizz.js" ouvre dans un premier temps un serveur *node js* en localhost sur le port 8888 de la machine et inclut le module *Express*, un module npm qui apporte les fonctionnalités fondamentales d'un serveur web. Ensuite, dans le fichier "budizz.js", on retrouve la fonction *saveUserData* qui permet d'écrire les informations soumises par l'utilisateur dans le fichier "users.json" en le parcourant. La fonction *push* complète ensuite les champs name, city et interests.

4.3 Traitement des données récupérées

Lorsque le fichier users.json est chargé, les données de celui-ci sont récupérées et traitées en plusieurs étapes. Tout d'abord, une fonction nommée *loadUserData* est définie. Cette fonction est responsable de charger les données JSON à partir du fichier "users.json". Elle utilise l'objet *XMLHttpRequest* pour effectuer une requête asynchrone vers le fichier JSON. Une fois les données chargées avec succès, elles sont transformées en un objet JavaScript "usersData". Cet objet est ensuite utilisé comme argument pour la fonction de rappel appelée *generateInterestsHTML* qui parcourt l'objet pour générer du contenu HTML en fonction des catégories d'intérêts de chaque utilisateur.

Ensuite, une autre méthode de récupération de données est utilisée via 'fetch'. Cette méthode est utilisée après le chargement initial pour obtenir à nouveau les données JSON à partir du même fichier 'users.json'. Cette fois-ci, les données sont récupérées de manière asynchrone via une promesse. Une fois les données obtenues, elles sont analysées et traitées pour regrouper les utilisateurs en fonction de leurs intérêts dans des catégories prédéfinies; "Geek", "Fêtard", etc. Ces données sont utilisées pour créer un diagramme en camembert interactif à l'aide de la bibliothèque Chart.js.

4.4 Difficultés rencontrées

Tout d'abord, lors de la conception de cette application web, des problèmes au niveau des sessions, à l'origine prévues dans le cahier des charges, se sont rapidement fait connaître. Il a donc été nécessaire pour nous de faire le choix de se restreindre à un serveur local sur une machine unique et ainsi limiter la fonctionnalité principale de l'application : l'utilisation en groupe. Il nous a ainsi fallu remplir manuellement des utilisateurs pour effectuer des tests. Ensuite, nous avons trouvé une alternative à l'utilisation de bases de données MySQL (trop complexes à manipuler à notre niveau). Nous les avons alors substituées par l'utilisation d'un fichier au format *JSON*.

Dans un dernier temps, sans session, il s'est avéré compliqué de présenter des interfaces personnalisées pour chaque utilisateur. Nous avons donc opté pour l'affichage d'un résultat global sous forme de camembert et de profils de personnalité.

5 Manuel d'utilisation

Il suffit de télécharger le fichier zip que nous avons mis à disposition avec le rapport. Sur votre terminal, une fois vous être mis.e dans le répertoire "BUDIZZ", tapez "node budizz.js". Vous aurez un message comme quoi le serveur a été démarré sur le port 8888. Ouvrez votre navigateur web et tapez "localhost:8888/" dans la barre de recherche. Vous accédez alors à la toute première page web de notre outil.

Maintenant, suivez les indications ! N'oubliez pas que le code pour parvenir au quizz est BUDIZZ en majuscules. Il ne vous reste plus qu'à répondre aux questions !

Nota bene : si vous voulez réinitialiser le quizz pour recommencer sans les informations des autres utilisateurs, tapez "reset.js" sur votre terminal.

6 Conclusion

Réaliser ce projet a été une expérience extrêmement enrichissante, nous avons pu dans un premier temps apprendre à maîtriser de nouveaux aspects de la programmation web tout en remaniant des notions vues en cours, notamment l'utilisation des fichiers JSON. Nous avons aussi eu l'occasion de faire des recherches dans le but de mettre en place un serveur afin de récupérer des données, une idée qui dépassait le contenu de nos cours.

Par la suite, nous avons la volonté de continuer à développer ce projet pour généraliser et maximiser son utilisation et ainsi dans un futur plus ou moins proche le rendre accessible au plus grand nombre ! Cette évolution passera par son hébergement et l'utilisation de bases de données.

Il est assez important pour nous de mentionner le fait que nous ressentons une grande fierté face au travail accompli. Premièrement de par le fait que nous avons réussi à respecter les deadlines et à anticiper les différentes tâches à accomplir, mais aussi les différents aléas : face aux embûches rencontrées, nous nous sommes beaucoup documentées pour trouver des solutions ou des alternatives.

Nous avons aussi pu reconnaître nos difficultés et/ou manque de compétences techniques pour changer nos objectifs et lignes de conduite, afin de parvenir à proposer un résultat à la hauteur de nos attentes et qui in fine nous plaît encore plus que prévu.

7 Bibliographie - Sitographie

Eléments Visuels :

canva.com

<https://www.remove.bg/fr>

<https://fonts.google.com/>

Ressources pour le code :

<https://www.pierre-giraud.com/php-mysql-apprendre-coder-cours/introduction-manipulation-fichier/>

<https://web.iamvdo.me/js/nodejs/>

<https://www.youtube.com/watch?v=MEN-SAsajSg>

<https://www.youtube.com/watch?v=eqGdudURBCI>