



Édition 2025

PRÉSENTATION DU PROJET



NOM DE VOTRE PROJET	SCANIFY
Membre de l'équipe	ANTONINA SAVCHENKO
Membre de l'équipe n°2 (prénom/nom)	ELISA SALIGNON
Niveau d'étude (première ou terminale)	PREMIERE
Établissement scolaire	LE GYMNASSE JEAN STURM
Responsable du dépôt (professeur de NSI)	M. ELOPHE / OLIVIER ELOPHE

1 / PRÉSENTATION GÉNÉRALE

Pouvez-vous présenter en quelques mots votre projet ?

Scanify est un site web qui permet d'analyser les tickets de caisse à l'aide de l'OCR, de ChatGPT et de Python pour mettre en forme les données. Il génère des graphiques récapitulant les dépenses quotidiennes et mensuelles, les types de produits achetés, les magasins visités ainsi que les moyens de paiement utilisés. De plus, l'utilisateur a la possibilité de créer un compte pour sauvegarder ses informations. Le but de Scanify : simplifier la gestion des dépenses au quotidien.

Comment est né ce projet ? Quelle était la problématique de départ ?

Lorsqu'on nous a parlé d'un projet de NSI, nous souhaitions avant tout élaborer un projet pouvant servir aux gens et leur être utile dans leur quotidien. Nous pensons que la programmation est une façon de créer des produits permettant d'améliorer ou de faciliter la vie des individus. De plus, étant conscientes de la part grandissante que l'IA prend dans notre société, notamment avec les récentes avancées dans ce domaine, nous avons envisagé d'intégrer cette dimension dans notre projet. Ce dernier est exclusivement né de notre désir de créer quelque chose d'utile, de tenter une nouvelle expérience, et d'avoir la satisfaction d'aboutir à un résultat concret et fonctionnel, bien que nous eussions des doutes quant à notre capacité à le terminer.

Quels sont les objectifs ? À quels besoins répondez-vous ?

Au quotidien, il peut être difficile de faire le compte de ses dépenses et de bien les gérer. Nous souhaitions donc créer une plateforme pouvant aider les gens à gérer leurs finances de manière simple, à analyser leurs tendances d'achat (que ce soit en termes de magasins, de produits ou de moyens de paiement), et ainsi à gagner du temps chaque jour.

2 / ORGANISATION DU TRAVAIL

Pouvez-vous présenter chaque membre de l'équipe et préciser son rôle dans ce projet ?

Antonina :

"Cela fait quelques années que je développe des petits programmes en python, comme un calendrier de l'avant avec des citations personnalisées pour mes proches. Par conséquent, j'ai décidé de prendre en charge la partie back-end de Scanify. Mon intérêt pour l'intelligence artificielle et ma compréhension du machine learning m'ont naturellement conduit à m'investir dans l'intégration de l'API de ChatGPT. Avec une bonne maîtrise des algorithmes et aimant la résolution de problèmes, je me suis également intéressé à la data science. Cette curiosité m'a amené à me lancer dans la programmation Python, avec un focus sur l'analyse des données et la gestion des bases de données, afin de rendre le traitement des informations plus performant et adapté aux besoins du projet. Même si mon programme n'est pas parfait et peut encore

être optimisé, je suis plutôt contente du résultat, car je ne pensais pas être capable de développer un aussi grand projet.”

Elisa :

“En seconde, nous avions un site à réaliser en classe de SNT. Ce fut la première fois que je découvrais des langages tels que HTML, CSS et JavaScript, et je pris beaucoup de temps et de plaisir à confectionner et embellir mon site, bien qu’il reste assez simple. Au fur et à mesure de ce projet, j’ai décidé de me concentrer sur l’interface du site. Mon objectif était de le rendre à la fois ergonomique, simple et intuitif, pour offrir une expérience utilisateur agréable. Pour optimiser l’aspect visuel et rendre l’interface plus dynamique, j’ai utilisé plusieurs ressources sous licence telles que Chart.js (Copyright (c) 2013-2021 Chart.js Contributors) et Particle.js (Copyright (c) 2017-2021 Vincent Garreau). Bien que le site et son interface puissent encore être améliorées, je suis assez satisfaite du rendu et j’ai hâte de continuer à travailler dessus après le concours. ”

Pourquoi cette organisation du travail et comment avez-vous réparti les tâches ?

Étant donné que nous étions deux et que nous avions chacune des affinités pour différentes parties du projet, nous avons décidé de nous répartir les tâches en fonction de nos connaissances et de nos préférences : l’une prenant en charge le back-end, l’autre le front-end : une organisation ressemblant au monde professionnel, comme l’un de nos professeurs l’a remarqué.

Combien de temps avez-vous passé sur le projet ?

Au début du projet, nous passions une heure, voire deux, par semaine et plusieurs jours pendant les vacances. Cependant, durant les deux derniers mois, notamment avec l’arrivée de la date d’échéance, nous avons dû augmenter la quantité à plusieurs heures par jour, notamment le soir après les cours. Toutefois, nous ne comptons plus les heures à travailler sur le projet et à discuter tard le soir sur les nouvelles fonctionnalités tant nous étions motivées à mener à bien notre projet.

Quels sont les outils et/ou les logiciels utilisés pour la communication et le partage du code ?

Nous avons d’abord utilisé la fonction LiveShare de Visual Studio Code, mais on s’est rapidement rendu compte qu’il n’était possible de modifier que la version partagée du fichier, ce qui nous obligeait à reporter manuellement les changements sur notre version locale. De plus, LiveShare imposait la contrainte d’être en ligne pendant de longues périodes, ce qui s’est révélé difficile à organiser. Après un certain temps, nous avons décidé de passer à GitHub. Au début, la plateforme nous a semblé complexe, mais au fur et à mesure, nous avons appris à l’utiliser, même s’il nous arrive encore de rester bloquées pendant quelques heures sur des problèmes liés au “push” et au “pull” de Github. Elle s’est avérée non seulement pratique pour collaborer, mais nous a également permis de nous initier à un outil très utilisé dans le monde professionnel.

3 / ÉTAPES DU PROJET

Préserver les différentes étapes du projet (de l'idée jusqu'à la finalisation du projet)

→ L'idée est venue quand l'une de nous a constaté que ses parents passaient du temps le weekend à calculer leurs dépenses à partir de leurs tickets de caisse. À partir de là, l'idée a germé : pourquoi ne pas créer un outil pour aider à gérer et suivre les dépenses au quotidien ?

→ On a alors décidé de créer un tableau Canva où on venait rajouter nos idées sur des post-it au fur et à mesure que le projet commençait à se dessiner.

→ Au départ, nous avons envisagé de créer une application avec Kivy, mais notre professeur de NSI nous a parlé des applications web. Nous avons donc décidé de nous orienter vers la bibliothèque Flask de Python, avec laquelle on s'est alors familiarisé.

→ On s'est d'abord concentrées sur l'IA, en utilisant une clé API et en réalisant des programmes de test en envoyant des requêtes avec des données brutes (du texte extrait manuellement d'un ticket de caisse). Nous avons observé comment ChatGPT répondait et, au fur et à mesure, nous avons affiné nos requêtes pour obtenir les réponses dans le format souhaité (par exemple, demander que la réponse de ChatGPT soit sous forme de dictionnaire ou que la date soit au format JJ/MM/AAAA pour tous les tickets de caisse, etc.). Nous avons utilisé la bibliothèque OpenAI pour intégrer des capacités de traitement du langage naturel dans notre programme.

→ Par la suite, nous avons eu l'idée d'intégrer une nouvelle fonctionnalité à notre site : pouvoir créer un compte.

→ On s'est alors informées sur les bases de données et les bibliothèques Python, puis nous avons trouvé la bibliothèque SQLAlchemy qui permet à Flask d'interagir avec des bases de données. S'en est suivie la création d'une première base de données pour les utilisateurs, en faisant appel aux classes en Python (programmation orientée objet).

→ Ensuite, il fallait permettre l'ajout des utilisateurs en fonction du mot de passe et du nom d'utilisateur qu'ils choisissent. Cela nous a amenées à nous intéresser à diverses bibliothèques : Flask-Login pour l'authentification des utilisateurs, Flask-WTF et WTForms pour la gestion des formulaires dans Flask, ainsi que Flask-Bcrypt pour sécuriser les mots de passe. La base de données se présente sous le format suivant : ID / Username / Password.

→ Parallèlement, nous avons également dû nous occuper de la mise en page du site. Tout d'abord, nous nous sommes penchées sur le choix de la palette de couleurs. Il a été assez difficile de trouver un ensemble de couleurs à la fois harmonieux, classique et élégant. Nous avons finalement opté pour des nuances de bleu gris et de blanc, afin d'apporter un aspect sérieux et simple. Ensuite, nous avons exploré différentes combinaisons de polices sur Google Fonts, et avons choisi un contraste entre une police cursive et classique et une police plus moderne. Cela nous a permis de commencer la partie HTML, CSS et JavaScript pour le front-end de notre site.

→ Une fois la première base de données créée, nous étions en mesure d'en créer une deuxième, reliée avec la première par le nom de l'utilisateur. Cette seconde base de données se présente sous le format suivant : ID / User_id / Shop_name / Date / Amount

/ Payment_method / Articles. C'est dans cette base de données que sont stockées les données des tickets de caisse.

→ Nous avons dû porter une attention particulière au format d'enregistrement de chaque donnée, en utilisant JSON pour la manipulation des informations.

→ S'en suit une imposante partie de programmation sur le traitement des données et la préparation des données pour les graphiques. Nous avons alors effectué beaucoup de tests pour obtenir une performance satisfaisante.

→ Nous avons étudié la bibliothèque Matplotlib pour les graphiques, qu'on relierait plus tard avec les données traitées.

→ Une grande partie du travail a également été consacrée à l'OCR, réalisée dans un fichier séparé. Nous avons utilisé la bibliothèque PyTesseract, et défini des énumérations avec enum pour traiter plusieurs langues. Nous avons aussi utilisé la bibliothèque PIL (Pillow) pour le traitement et la manipulation des images, ainsi que la bibliothèque OS pour trouver les images et gérer tout ce qui était relié aux fichiers et répertoires.

→ Nous sommes enfin arrivés à la partie intéressante qu'est la combinaison du programme OCR avec la partie sur le traitement des tickets (bibliothèque os), car précédemment, l'OCR utilisait un texte brut extrait par Google Lens d'une image. La combinaison n'a pas été réussie directement, mais après de nombreux tests et améliorations, nous avons finalement obtenu un résultat fonctionnel !

→ Le back-end du site étant enfin fonctionnel, on a pu se consacrer à la partie design et remplir le contenu du site, optimiser les graphiques et l'interface (on a par exemple créé une zone où l'on pourrait glisser et déposer les tickets de caisse à analyser). Nous avons utilisé le langage Jinja dans les codes HTML pour créer les boucles dans les listes.

→ Quand la fin du projet approchait, nous notions les choses à réaliser dans un fichier texte, et effectuions les dernières améliorations. Comme la célèbre citation l'énonce, les derniers 10 % prennent autant de temps que les autres 90 % car à chaque test, des choses s'ajoutent sur la liste.

4 / FONCTIONNEMENT ET OPÉRATIONNALITÉ

Pouvez-vous présenter l'état d'avancement du projet au moment du dépôt ? (ce qui est terminé, en cours de réalisation, reste à faire)

Au moment du dépôt, le site est opérationnel : on peut créer un compte et déposer un ticket de caisse pour obtenir les graphiques liés à celui-ci à l'aide de l'IA. Même certains bugs liés à la réponse de ChatGPT sont à noter, par exemple, quelquefois quand le nom de l'article n'est pas énoncé clairement sur le ticket, il a du mal à le classer dans la bonne catégorie. Aussi, le montant dépensé par catégories peut parfois être inexact, cependant nous sommes en train de travailler sur ce problème.

De plus, il est important de noter que notre OCR ne fonctionne qu'avec des fichiers JPEG ou JPG pour le moment.

Ce qu'il reste en cours de réalisation est l'amélioration de la partie graphique, avec éventuellement des illustrations ou davantage d'animations à rajouter et une optimisation de la responsivité du design, ainsi qu'une version en anglais du site.

Ce qu'il reste à faire : l'ajout d'une caméra intégrée à notre site pour faciliter le dépôt de tickets de caisse ainsi que la possibilité de supprimer son compte, ainsi que d'autres fonctionnalités pour améliorer l'expérience utilisateur, comme une possibilité de choisir la devise et la langue du ticket afin d'optimiser son analyse.

Par ailleurs, l'analyse des tickets de caisse comportant des réductions ou des articles aux noms particuliers pose actuellement des difficultés. Il est également essentiel que la qualité de la photo soit suffisamment élevée et que le ticket ne soit pas trop long pour garantir une analyse précise. Il nous reste donc à travailler sur ces aspects pour améliorer davantage le service. Nous avons aussi pensé à ajouter une option qui permettrait aussi de rentrer un ticket manuellement.

Quelles approches avez-vous mises en œuvre pour vérifier l'absence de bugs et garantir une facilité d'utilisation de votre projet ?

Nous avons mis en place de nombreux tests, notamment en scannant de vrais tickets de caisse, y compris les nôtres. À chaque test, nous les insérons dans le programme et notons les erreurs rencontrées afin d'améliorer continuellement le programme. Nous avons testé plus d'une centaine de tickets pour garantir la fiabilité et l'efficacité du programme. Par exemple, au bout d'un certain nombre de tests, on a remarqué que nos graphiques n'étaient pas dans l'ordre chronologique, donc nous avons dû y remédier.

Pour la partie interface, nous avons demandé l'avis de nos proches, mais également de nos amis, en leur faisant tester notre application web avec leurs tickets. C'est d'ailleurs une personne qui a testé Scanify qui nous a donné l'idée d'ajouter une option qui permet de voir la provenance de chaque article.

Quelles sont les difficultés rencontrées et les solutions apportées ?

Nous avons rencontré des difficultés liées à la formulation des bons prompts à ChatGPT afin d'obtenir les meilleures réponses possibles. La solution a été de réaliser de nombreux tests et de choisir le prompt le plus optimal. Nous avons également fait face à des problèmes liés à certains tickets de caisse. Pour y remédier, nous avons dû modifier le code afin de l'optimiser en n'excluant pas certains cas particuliers.

Nous avons également rencontré des difficultés liées à GitHub, notamment pour savoir comment gérer une clé API, qui doit rester confidentielle. La solution apportée a été l'utilisation de la bibliothèque Dotenv, qui permet de garder la clé API privée. Nous avons également eu du mal à trouver et utiliser les bibliothèques les plus optimales. Par exemple, pour la génération de graphiques, nous avons d'abord opté pour Matplotlib, mais nous avons appris par la suite que Seaborn serait une solution plus optimale adaptée à nos besoins. Il est donc possible que nous l'utilisions à l'avenir.

À mesure que l'application se développait, il devenait de plus en plus difficile de savoir quels endroits du code étaient modifier, notamment lorsqu'il s'agissait de changer des

noms de variables utilisés à plusieurs reprises. Nous avons donc utilisé le mode recherche dans VS Code ainsi que les commentaires (avec les #..., ou /* suivis de nombreux tirets) pour faciliter la navigation et diviser le code en plusieurs parties.

5 / OUVERTURE

Quelles sont les nouvelles fonctionnalités à moyen terme ? Avez-vous des idées d'amélioration de votre projet ?

Nous avons énormément d'idées de nouvelles fonctionnalités que nous implémenterons après le concours. Nous pensions tout d'abord à traduire notre site dans d'autres langues, notamment en anglais, ainsi qu'à ajouter la possibilité de changer la devise pour les tickets de caisse. Nous travaillerons encore sur l'optimisation de la partie graphique et pensions implémenter de nouvelles fonctionnalités au site, telles que des aides personnalisées à l'économie ou la possibilité d'écrire ses listes de courses.

Pourriez-vous apporter une analyse critique de votre projet ? Si c'était à refaire, que changeriez-vous dans votre organisation, les fonctionnalités du projet et les choix techniques ?

En rétrospective, si notre projet était à refaire, on changerait quelques aspects. Tout d'abord, nous serions plus vigilantes quant à la gestion du temps ainsi que la gestion du projet en général. Ensuite, ayant acquis plus de connaissances en programmation, nous aurions fait d'autres choix concernant certains aspects du programme.

Quelles compétences/appétences/connaissances avez-vous développées grâce à ce concours ?

Grâce à ce concours, nous avons acquis de nombreuses compétences techniques et professionnelles. L'utilisation de bibliothèques telles que Flask, Matplotlib et PyTesseract nous a permis d'approfondir nos connaissances en Python et CSS, en développement web et en analyse de données. Nous avons également pris en main des outils de développement collaboratif largement utilisés dans le milieu professionnel, tels que GitHub. Ce fut l'occasion de nous initier à l'utilisation de branches, à la gestion des conflits, et aux différentes commandes Git.

De plus, cette expérience nous a permis de nous plonger dans un environnement de projet logiciel réel, où nous avons dû résoudre des problèmes techniques et logistiques, ainsi que travailler en équipe, ce qui constitue une expérience précieuse pour notre futur.

En quoi votre projet favorise-t-il l'inclusion ?

Notre projet est facile à prendre en main et intuitif. L'OCR et l'IA simplifient l'analyse des dépenses et Scanify le présente d'une manière simple à comprendre, quel que soit le niveau de familiarité avec les outils numériques. Scanify permet à tous, y compris aux personnes rencontrant des difficultés à gérer leurs finances, d'obtenir un suivi clair de leurs dépenses et de leur mode de consommation. Enfin, notre projet permet de rendre

l'IA accessible à tous et de faire profiter gratuitement au grand public des avancées technologiques.