

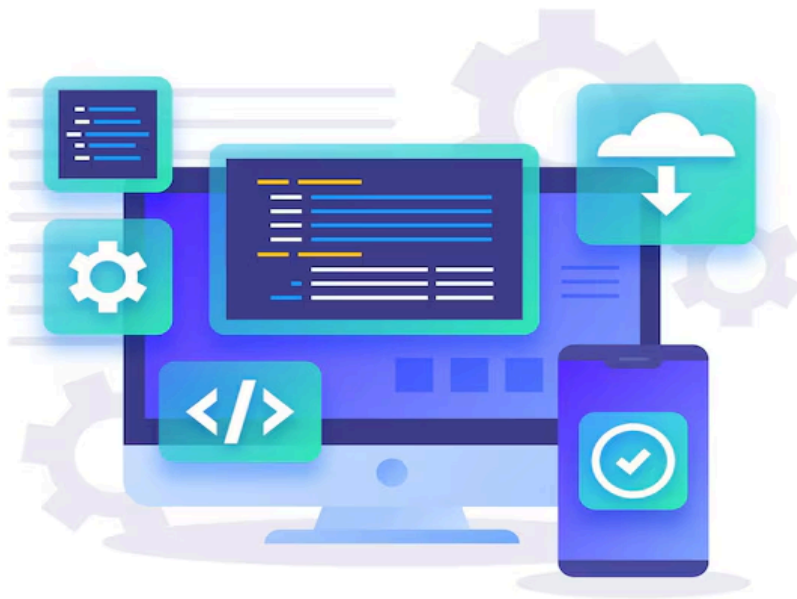
**Nom et prénom :** Rukundo Ronaldo

**Intitulé de la formation :** Développeur Intégrateur Web

**Nom du centre de formation :** Développeur Intégrateur Web

**Nom de la société :** Neatemys

**Nom du responsable de formation :** Raquel Azinheira



# Sommaire

## 1. Introduction

- Présentation générale
- Objectifs du rapport

## 2. Présentation de l'entreprise

- 2.1. Historique et mission de Neatemys
- 2.2. Objectifs stratégiques
- 2.3. Activités principales
  - 3.1. Gestion des conflits par la médiation
  - 3.2. Coaching en bien-être psychologique

## 3. Aperçu technique : Architecture de la plateforme

- 3.1. Technologies utilisées
- 3.2. Architecture MVC de Django
- 3.3. Migration de Bootstrap vers Tailwind CSS
- 3.4. Gestion de version avec Git et Bitbucket
- 3.5. Environnement de développement Ubuntu

## 4. Processus de développement et gestion des tâches

- 4.1. Méthodologie de développement et gestion des tâches
- 4.2. Développement et tests
- 4.3. Pull request et revue de code
- 4.4. Tests en pré-production et déploiement
- 4.5. Gestion des bugs et maintenance
- 4.6. Documentation continue

## 5. Mon équipe

- 5.1. Composition de l'équipe
- 5.2. Collaboration au quotidien

## 6. Conclusion

- Synthèse de l'expérience
- Compétences acquises
- Perspectives futures

# Neatemys

## Introduction

Ce rapport d'alternance présente mon expérience en tant que développeur web au sein de **Neatemys**, une entreprise innovante spécialisée dans la création de solutions numériques pour la gestion des conflits en entreprise et le coaching en bien-être psychologique.

Durant une période de 12 mois, j'ai eu l'opportunité d'intégrer une équipe technique dynamique et de contribuer au développement d'une plateforme web centrée sur la médiation professionnelle et le soutien psychologique des employés. Mon rôle principal consistait à travailler sur l'amélioration des fonctionnalités de la plateforme, notamment pour la planification des sessions de médiation, la personnalisation de l'expérience utilisateur et la migration technologique de certains outils.

Dans un contexte où les conflits internes et les problèmes psychosociaux peuvent significativement impacter la productivité des entreprises, Neatemys se positionne comme un acteur essentiel, alliant innovation numérique et approche humaine. Ce rapport explore les différentes missions et projets auxquels j'ai participé, les technologies employées ainsi que les compétences professionnelles que j'ai pu développer.

Le document est structuré en plusieurs parties : une présentation de l'entreprise, une description des technologies utilisées et des processus de développement, suivie d'une analyse des contributions personnelles et d'un bilan de cette expérience.

---

## Présentation de l'entreprise

### 1. Historique et mission de Neatemys

Fondée dans le but de répondre à une problématique croissante au sein des entreprises – la gestion des conflits internes et la promotion du bien-être psychologique – **Neatemys** s'est imposée comme un acteur incontournable dans ce secteur. L'entreprise propose des solutions numériques innovantes permettant aux entreprises de créer un environnement de travail plus harmonieux et respectueux.

Depuis sa création, Neatemys a su conjuguer éthique, confidentialité et efficacité pour développer une plateforme combinant médiation professionnelle et coaching psychologique. En plaçant l'humain au centre de ses démarches, l'entreprise ambitionne de prévenir les conflits avant qu'ils ne deviennent de véritables obstacles à la performance et à la cohésion des équipes.

## **2. Objectifs stratégiques**

Le modèle économique de Neatemys repose sur la conviction que la prévention et la gestion des conflits, alliées à un accompagnement en bien-être, sont des piliers fondamentaux pour les entreprises modernes. Voici ses principaux objectifs :

- **Prévenir les conflits internes** : Créer un cadre propice au dialogue constructif dès l'apparition des premières tensions.
- **Améliorer le bien-être au travail** : Offrir des services de coaching psychologique personnalisé pour réduire le stress et promouvoir la santé mentale des employés.
- **Réduire les coûts** : Remplacer les longues procédures contentieuses par des méthodes plus rapides, confidentielles et économiques.
- **Renforcer les relations professionnelles** : Préserver et réparer les relations internes grâce à des solutions durables et équitables.

## **3. Activités principales**

Neatemys se distingue par deux axes majeurs de son activité :

### **3.1. Gestion des conflits par la médiation**

La médiation est au cœur de l'offre de Neatemys. Elle repose sur un processus structuré qui permet de régler les différends entre collaborateurs tout en préservant la confidentialité et l'impartialité.

Les étapes clés incluent :

- **Organisation des sessions** : Réunir les parties dans un cadre propice au dialogue.
- **Identification des besoins** : S'assurer que les attentes de chaque partie soient bien comprises.
- **Recherche de solutions** : Proposer des accords durables et acceptables pour toutes les parties.

### 3.2. Coaching en bien-être psychologique

En complément de la médiation, Neatemys propose des services de coaching visant à accompagner les employés dans la gestion de leur stress et l'amélioration de leur bien-être mental. Les outils incluent :

- **Sessions de coaching personnalisées** : Individuelles ou collectives, elles permettent d'aborder des problématiques liées au stress, au burnout, ou au développement personnel.
- **Ressources numériques** : Guides pratiques, exercices de relaxation et outils en ligne sont accessibles sur la plateforme.

Avec ces deux axes d'intervention, Neatemys contribue à réduire les risques psychosociaux tout en favorisant un climat de travail serein et productif.

---

### Aperçu technique : Architecture de la plateforme

La plateforme développée chez Neatemys repose sur une architecture technique robuste et flexible, combinant plusieurs technologies front-end et back-end pour offrir une solution efficace et intuitive aux utilisateurs.

#### 1. Technologies utilisées

La conception de la plateforme intègre les technologies suivantes :

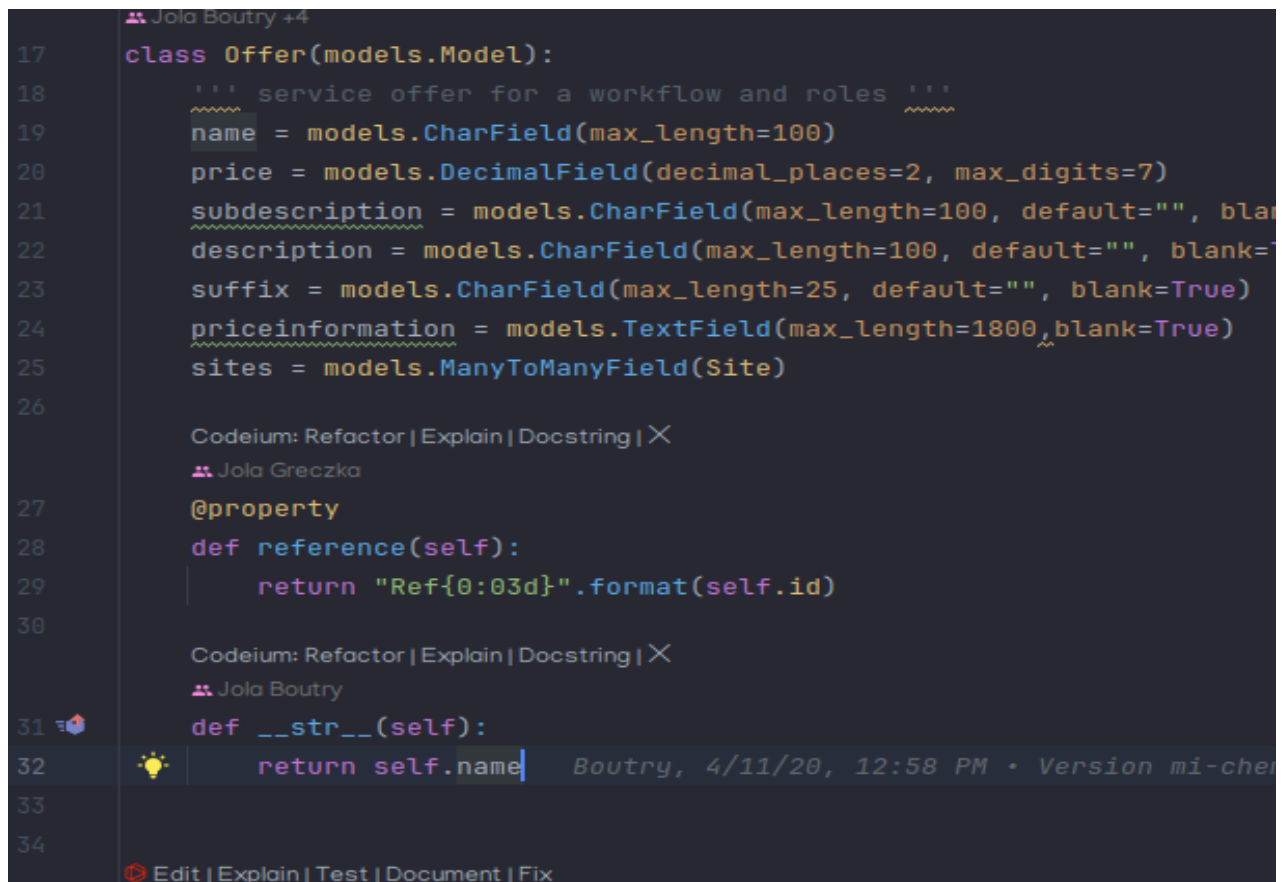
- **Django** : Framework Python basé sur l'architecture MVC (Modèle-Vue-Contrôleur), responsable de la gestion des données, des utilisateurs, et des interactions serveur.

- **HTML/CSS** : Utilisés pour structurer et styliser les interfaces utilisateur, avec une transition progressive de **Bootstrap** à **Tailwind CSS** pour un design moderne et adaptatif.
- **jQuery** : Bibliothèque JavaScript permettant de simplifier les interactions dynamiques sur l'interface, telles que les formulaires interactifs et les animations.
- **Git et Bitbucket** : Outils essentiels pour le contrôle de version et la gestion collaborative du code. Git permet de suivre l'historique des modifications, tandis que Bitbucket offre une interface pour la gestion des branches, des pull requests et des revues de code, facilitant ainsi la collaboration au sein de l'équipe.

## 2. Architecture MVC de Django

Django suit une architecture MVC qui garantit une séparation claire entre la logique métier, l'interface utilisateur et la gestion des données. Voici ses composants principaux :

- **Modèle (Model)** :



```

17 class Offer(models.Model):
18     ''' service offer for a workflow and roles '''
19     name = models.CharField(max_length=100)
20     price = models.DecimalField(decimal_places=2, max_digits=7)
21     subdescription = models.CharField(max_length=100, default="", blank=True)
22     description = models.CharField(max_length=100, default="", blank=True)
23     suffix = models.CharField(max_length=25, default="", blank=True)
24     priceinformation = models.TextField(max_length=1800, blank=True)
25     sites = models.ManyToManyField(Site)
26
27     @property
28     def reference(self):
29         return "Ref{0:03d}".format(self.id)
30
31     def __str__(self):
32         return self.name
33
34

```

Codeium: Refactor | Explain | Docstring | X

Jola Boutry

Codeium: Refactor | Explain | Docstring | X

Jola Boutry

Boutry, 4/11/20, 12:58 PM • Version mi-cher

Edit | Explain | Test | Document | Fix

Gère les structures de données et leur interaction avec la base de données. Cela comprend la création, la mise à jour et la récupération des informations stockées. Les modèles sont configurés pour refléter les entités clés de l'application, comme les utilisateurs, les sessions de médiation ou les résultats de coaching. Ils assurent également la validation des données avant leur enregistrement dans la base de données. Django intègre un puissant **ORM (Object-Relational Mapping)**, qui permet de manipuler les données directement en Python sans avoir besoin d'écrire de requêtes SQL. L'ORM offre des fonctionnalités avancées, telles que les relations entre modèles (OneToOne, ManyToMany), les migrations automatiques pour la base de données et les filtres pour interroger facilement les données.

- **Vue (View) :**

```
@login_required
def show_offers(http_request, procedure_type):
    if http_request.method == "POST" and "offer_id" in http_request.POST:
        offer = OffersHelper.get_offer_by_id(http_request.POST["offer_id"])
        details = OffersHelper.get_details(offer)
        if hasattr(details, "procedure"):
            request = RequestHelper.get_recurrent_request(details.procedure.name, http_request.user)
            # disable recurrent procedure clone except for assistance juridique
            is_active = bool(request and RequestHelper.get_current_state(request).name == "Active")
            if not request or not is_active:
                request = RequestHelper.create_new_request(
                    details.procedure.name, http_request.user,
                    offer, get_current_site(http_request).id)
            return redirect(to="request", request_id=request.identifier)
        else:
            if procedure_type == "litigation":
                request = RequestHelper.create_new_request(
                    details.workflow.name, http_request.user,
                    offer, get_current_site(http_request).id)
                VisioHelper.initialize_availabilities(request)
                return redirect(to="request", request_id=request.identifier)
            return redirect(
                to="pre_creation",
                workflow_name=details.workflow.name,
                offer_id=offer.id)

    return render(http_request, template_name="offers/choice.html", context={
        "procedure": procedure_type,
        "offers": OffersHelper.get_offers_for_procedure(procedure_type, get_current_site(http_request).id)})
```

Responsable de l'affichage des données et des interfaces utilisateur. Elle traduit les données issues des modèles en contenu compréhensible pour l'utilisateur final. Dans le contexte de la plateforme, les vues incluent des fonctionnalités comme la liste des sessions de médiation disponibles, les formulaires d'inscription ou encore les pages de suivi du bien-être des

employés. Les vues jouent un rôle clé dans l'optimisation de l'expérience utilisateur.

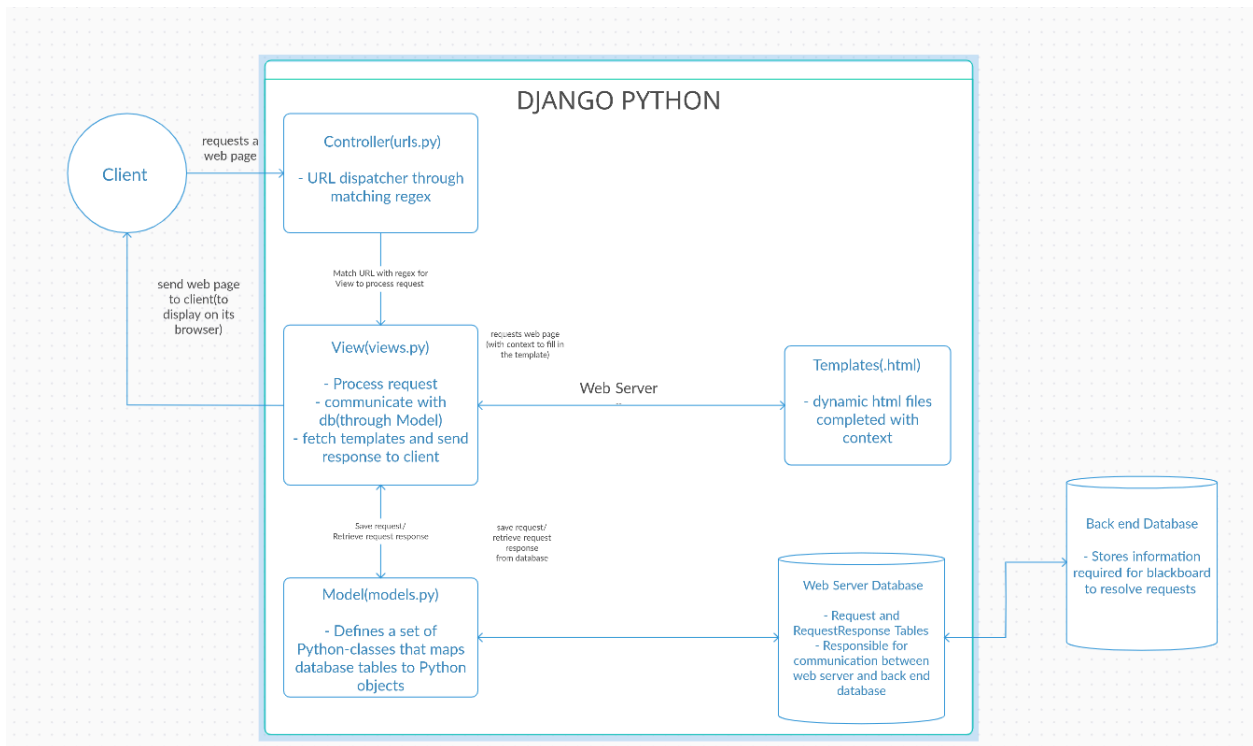
- **Contrôleur (Controller) :**

```
4
5 {% block javascripts %}
6 <script>
7   Codelum Refactor | Explain | Docstring | X
8   function checkDisableSubmit () {
9     $('input[name="choice"]').prop('disabled', !$('#input[name="choice_id"]:checked').length)
10  }
11
12  $(document).ready(function () {
13    $('#input[name="choice_id"]:checked').parent(".pack-choice-card").addClass("pack-choice-card-active")
14
15    $('.pack-choice-card').on('click', function() {
16      $('#input[name="choice_id"]:checked').prop("checked", true);
17      $('#input[name="choice_id"]').parent(".pack-choice-card").addClass("pack-choice-card-active");
18      $('#input[name="choice_id"]').parent(".pack-choice-card").removeClass("pack-choice-card-active");
19    })
20    checkDisableSubmit();
21  })
22 </script>
23 {% endblock %}
24
25 {% block content %}
26 <div class="w-full flex justify-center mb-20" style="background-color: #f0f0f0; padding: 10px; border-radius: 10px;">
27   <form method="post" enctype="multipart/form-data" class="w-full xl:w-3/4 px-4">
28     {% csrf_token %}
29     <div class="flex flex-wrap justify-center mb-10">
30       {% for pack in packs %}
31         <div class="pack-choice-card px-6 m-6 cursor-pointer">
32           <input type="radio" name="choice_id" value="{{pack.offer.id}}" class="hidden"/>
33           <div class="text-md font-bold md:text-xl 2xl:text-3xl py-6 text-center">
34             {{pack.offer.name}}
35           </div>
36           <ul class="list-disc pl-8">
37             {% for info in pack.price_informations %}
38               <li>{{info}}</li>
39             {% endfor %}
39           </ul>
40         </div>
41       {% endfor %}
42     </div>
43   </form>
44 </div>
```

Relie les modèles aux vues en appliquant la logique métier. Il coordonne les requêtes des utilisateurs, gère les interactions avec les modèles et détermine les vues à afficher. Par exemple, lors de la planification d'une session de médiation, le contrôleur reçoit les données utilisateur, vérifie leur validité via le modèle, puis met à jour les informations nécessaires tout en générant une réponse adaptée dans la vue.



## Fonctionnement MVC Django



## 3. Migration de Bootstrap vers Tailwind CSS

L'équipe technique a initié une transition de Bootstrap à Tailwind CSS pour bénéficier d'une approche plus modulaire et adaptée à leurs besoins spécifiques. Cette migration permet une personnalisation accrue du design tout en améliorant la performance globale de l'interface.

## 4. Gestion de version avec Git et Bitbucket

```
Date: Sun Jan 12 11:51:05 2025 +0100

[DOC] update

commit ab94822ca1db6b43f136dd1034cd3dc5ead85b97
Author: Jolanta Greczka <jolanta.greczka@dreaminzzz.com>
Date: Sun Jan 12 11:45:04 2025 +0100

[EVOL] test fix

commit 57f53e377306fb367fca4833b71b9ef301828ea0
Author: Jolanta Greczka <jolanta.greczka@dreaminzzz.com>
Date: Sun Jan 12 11:15:59 2025 +0100

[EVOL] test fix

commit 779e69e8768606cf29fe25decc58ea6c27d06ed2
Author: Jolanta Greczka <jolanta.greczka@dreaminzzz.com>
Date: Sun Jan 12 11:15:48 2025 +0100

[EVOL] deleting unused

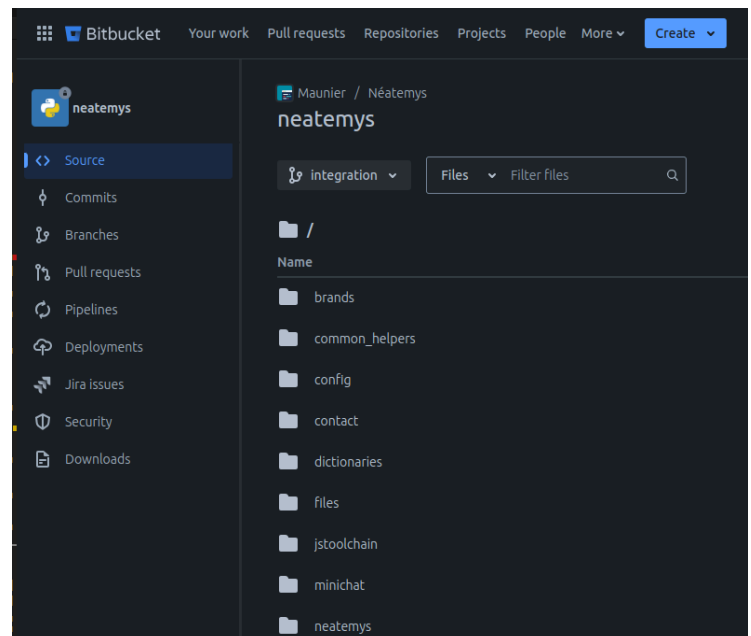
commit c27c8522a26be03ffe7d2f45f54effa6a68ae95d
Author: ronaldo <ronaldo@neatemys.fr>
Date: Fri Jan 10 17:24:31 2025 +0100

fix: remove test file

commit 88894aa6620f4ddad4ca50b60504bacfb8fcb633
Author: ronaldo <ronaldo@neatemys.fr>
Date: Fri Jan 10 17:17:15 2025 +0100

fix stats

commit d8e01371316f1b575c951eccc8086c0d4c3103
```



L'utilisation de **Git** et **Bitbucket** joue un rôle central dans le processus de développement collaboratif chez Neatemys. Voici comment ces outils sont intégrés :

- **Git :**
  - Permet de suivre les modifications du code source à travers un système de versionnement distribué.
  - Offre la possibilité de créer des branches pour chaque fonctionnalité ou correction, facilitant ainsi le travail parallèle des développeurs.
  - Assure un historique complet des changements, permettant de revenir à des versions antérieures si nécessaire.
- **Bitbucket :**
  - Fournit une interface pour gérer les dépôts Git de manière centralisée.
  - Simplifie la collaboration via les **pull requests**, qui permettent aux développeurs de soumettre leurs modifications pour revue avant leur intégration dans la branche principale.
  - Intègre des outils de gestion de projet, comme les pipelines CI/CD (Intégration Continue et Déploiement Continu), qui automatisent les tests et les déploiements.
  - Propose un espace centralisé pour les commentaires sur le code, favorisant l'amélioration continue à travers des retours constructifs.

## 5. Environnement de développement Ubuntu

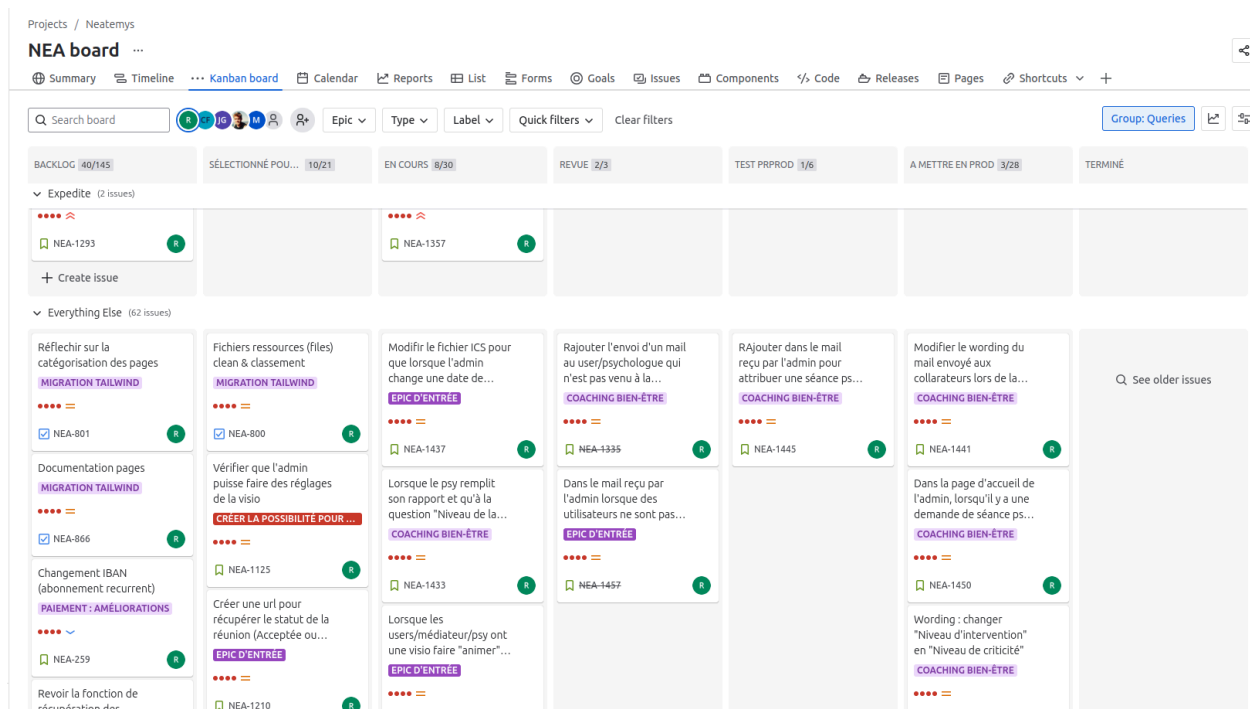
Tous les développements sont réalisés sous **Ubuntu**, une distribution Linux open source reconnue pour sa fiabilité, sa sécurité, et sa compatibilité avec les outils de développement modernes. Ubuntu offre plusieurs avantages pour les développeurs :

- **Gestion des dépendances :** Grâce à son gestionnaire de paquets **apt**, Ubuntu facilite l'installation, la mise à jour et la gestion des bibliothèques nécessaires au projet.
- **Stabilité et performance :** En tant que système d'exploitation léger et stable, Ubuntu garantit une exécution fluide des environnements de développement.

- **Environnement de ligne de commande puissant** : Avec des outils comme **bash**, **cron** et **shell scripting**, les développeurs peuvent automatiser de nombreuses tâches répétitives.
- **Compatibilité avec les frameworks et technologies** : Ubuntu supporte nativement des outils comme Python, PostgreSQL, Git, et les serveurs web comme Nginx ou Apache.
- **Personnalisation** : Les développeurs peuvent configurer leur environnement selon leurs besoins spécifiques, par exemple en utilisant des éditeurs comme **VS Code** ou **Vim**, et en installant des conteneurs via **Docker**.

## 6. Gestion des tâches avec Jira et réunions d'équipe

La gestion des tâches et des projets chez Neatemys repose sur **Jira**, un outil de gestion de projet agile. Cet outil permet de suivre les tâches assignées à chaque membre de l'équipe et de maintenir une vision claire de l'avancement des projets.



- **Utilisation des tickets Jira** :

- Chaque tâche ou fonctionnalité est représentée par un ticket Jira, avec des détails tels que la description, les priorités, les échéances et le statut.
- Les tickets sont organisés en sprints hebdomadaires ou bihebdomadaires, suivant la méthodologie agile.
- Les développeurs mettent à jour les tickets pour indiquer leur progression (en cours, en revue, terminé), assurant ainsi une transparence totale au sein de l'équipe.
- **Réunions d'équipe quotidiennes :**
  - **Réunion matinale** : Chaque matin, l'équipe se réunit pour un stand-up rapide afin de discuter des tâches en cours, des blocages éventuels et des priorités de la journée.
  - **Réunion de fin de journée** : Une réunion de synthèse est organisée pour faire le point sur les avancées réalisées, identifier les tâches reportées et planifier les prochaines étapes.
- **Avantages de cette organisation :**
  - Amélioration de la collaboration et de la communication au sein de l'équipe.
  - Réduction des délais grâce à un suivi rigoureux des tâches et des priorités.
  - Identification rapide des blocages, permettant une résolution plus efficace.

## 7. Processus de développement et outils utilisés

Le processus de développement chez Neatemys suit une approche méthodique et bien structurée pour garantir la qualité et l'efficacité des livrables.

- **Environnement de développement :**
  - Les développeurs utilisent **PyCharm**, un IDE puissant dédié à Python, qui offre des fonctionnalités comme l'auto-complétion, le débogage intégré et la gestion des environnements virtuels.
- **Étapes clés du processus de développement :**
  - **Création d'une branche Git dédiée :**
    - Chaque tâche ou fonctionnalité commence par la création d'une branche Git spécifique à partir de la branche principale (master ou develop). Cela permet d'isoler le développement et de minimiser les conflits de code.

- **Découpage de la tâche :**
  - La tâche est décomposée en sous-problèmes ou étapes plus petites, chacun étant plus facile à résoudre et à tester.
- **Développement :**
  - Le code est écrit et testé localement dans l'environnement PyCharm. Les tests incluent des vérifications de logique métier, des interactions avec la base de données et des réponses API.
- **Tests locaux et pré-production :**
  - Une fois le développement terminé, des tests unitaires et fonctionnels sont exécutés pour garantir que le code répond aux exigences.
  - L'application est ensuite déployée dans un environnement de pré-production pour effectuer des tests en conditions réelles avant le déploiement en production.
- **Intégration et revue :**
  - Une pull request est créée sur Bitbucket pour intégrer la branche. Les membres de l'équipe effectuent une revue de code approfondie avant l'approbation.
- **Déploiement sur branche Git dédiée :**
  - Le code est déployé dans une branche dédiée pour des tests approfondis sur l'environnement de pré-production. Cela permet de s'assurer que toutes les modifications fonctionnent comme prévu sans impacter la branche principale ou l'environnement de production.
- **Documentation continue :**
  - Une attention particulière est portée à la mise à jour de la documentation. Chaque changement de code, ajout de fonctionnalité ou suppression est documenté pour assurer une transparence totale. La documentation inclut :
    - Des notes sur les nouvelles fonctionnalités ajoutées.
    - Des instructions sur les changements apportés aux processus.
    - La suppression ou la dépréciation de fonctionnalités obsolètes.
    - L'impact des modifications sur les utilisateurs finaux ou sur d'autres composants du système.

- **Tests unitaires et fonctionnels :**
  - Des tests unitaires automatisés sont écrits pour chaque fonctionnalité, garantissant que chaque composant individuel fonctionne comme prévu.
  - Les tests fonctionnels vérifient que l'application dans son ensemble répond aux attentes des utilisateurs.

## 8. Mon équipe

Durant mon alternance chez Neatemys, j'ai eu l'opportunité de travailler au sein d'une petite équipe technique composée de profils divers, chacun apportant des compétences complémentaires au développement de la plateforme.

### 8.1. Composition de l'équipe

- **Développeur principal (Lead Developer) :** Responsable de la supervision technique du projet, il veillait à ce que les meilleures pratiques de développement soient respectées, supervisait l'architecture globale de la plateforme, et encadrait les membres de l'équipe, dont moi en tant qu'alternant.
- **Alternant (moi-même) :** En tant qu'alternant développeur web, j'ai contribué activement à la réalisation des différentes fonctionnalités de la plateforme, en front-end et back-end, sous la supervision du développeur principal.
- **Deux développeurs freelance :** Sollicités régulièrement pour apporter un soutien technique sur des tâches spécifiques, notamment sur des fonctionnalités précises et l'optimisation des performances de la plateforme.

### 8.2. Collaboration au quotidien

Malgré la distance avec les deux développeurs freelance, la collaboration au sein de l'équipe était fluide. Nous utilisions des outils de gestion de projet tels que **Jira** pour organiser et assigner les tâches, et **Slack** pour assurer une communication en temps réel. Les réunions quotidiennes permettaient de synchroniser les efforts, de partager les progrès et de résoudre rapidement les éventuels obstacles.

## Conclusion

Mon expérience en tant que développeur web chez **Neatemys** a été particulièrement enrichissante et formatrice. Elle m'a permis de contribuer de manière concrète à un projet innovant et de comprendre les enjeux liés à la gestion des conflits en entreprise et à la promotion du bien-être psychologique.

Au fil des 12 mois, j'ai eu l'opportunité de développer mes compétences techniques sur des technologies modernes telles que Django, Tailwind CSS, PostgreSQL et Git, tout en évoluant dans un environnement collaboratif basé sur des outils comme Bitbucket. Cette expérience m'a également sensibilisé aux bonnes pratiques de développement, notamment la gestion de version, l'intégration continue et la coordination au sein d'une équipe technique dynamique.

Travailler sous un environnement Ubuntu a renforcé mes capacités à configurer et optimiser des environnements de développement, tout en m'initiant à l'automatisation des tâches et à l'utilisation efficace des ressources système. Ce cadre m'a offert une grande autonomie tout en me permettant de m'appuyer sur le savoir-faire de mes collègues.

En conclusion, cette alternance a non seulement confirmé mon intérêt pour le développement web et les solutions numériques innovantes, mais elle a également consolidé ma vision d'un travail alliant technicité et impact humain. Je suis désormais mieux préparé à relever les défis futurs dans ce domaine passionnant, avec une expertise renforcée et une meilleure compréhension des attentes du monde professionnel.