



CodeIguanas



HomeSkolar

Création d'une plateforme web permettant la mise en relation de tuteurs bénévoles avec des étudiants en difficulté scolaire.

Cahier des charges

Application HomeSkolar

Spécifications fonctionnelles

La plateforme se présentera comme un site web avec une page d'authentification pour les deux types d'utilisateurs de la plateforme : les élèves et les tuteurs. Le site devra mettre en place 5 fonctions principales :

- **L'authentification**

- ◇ **Créer un compte** : Les utilisateurs doivent créer un compte pour accéder au contenu de la plateforme. Le rôle des tuteurs est attribué par un admin après validation de l'email et du dossier de l'utilisateur. Un tuteur doit être automatiquement attribué à un nouvel étudiant. Un étudiant ne peut avoir qu'un tuteur, mais un tuteur peut avoir plusieurs étudiants.
- ◇ **Login** : Les utilisateurs doivent pouvoir, une fois leur compte validé et leurs rôles attribués, utiliser leurs identifiants pour se connecter.
- ◇ **Modification des données personnelles** : Les utilisateurs doivent avoir la possibilité de modifier leur mot de passe et de mettre à jour leur email ainsi que leurs données personnelles.

- **La communication**

- ◇ **Messagerie** : Les étudiants doivent entrer en contact avec leurs tuteurs une fois par semaine, il faudra donc un système de messagerie en temps réel. Chaque utilisateur pourra envoyer et recevoir des messages.
- ◇ **Notification** : Chaque message non lu doit être notifié.
- ◇ **Pin** : Un message reçu doit pouvoir être épinglé et désépinglé.

- **Le calendrier**

- ◇ **Prise de rdv** : Lorsqu'un échange se termine, ou qu'un nouvel étudiant s'inscrit, le tuteur assigné doit pouvoir ouvrir un calendrier afin de proposer à l'étudiant un rendez-vous la semaine suivante. Le tuteur doit pouvoir modifier le RDV.

- **La liste des tâches**

- ◇ **Ajout et modification de tâches** : Le tuteur doit pouvoir ajouter ou modifier une liste de tâches à accomplir à son étudiant pour la semaine suivante.

- **Les notes**

- ◇ **Ajout et modification de notes** : L'utilisateur doit pouvoir prendre des notes (mémos), les modifier ou les supprimer.

Synthèse de veille technologique

1. Authentification

Technologies :

- Django : Le framework web Django permet une gestion robuste de l'authentification, avec des modules intégrés pour la création de comptes, la gestion des sessions et la protection des données.

Avantages :

- Sécurité intégrée avec la gestion des sessions et la protection CSRF (Cross-Site Request Forgery).
- Flexibilité pour l'attribution des rôles et des permissions.

Sources :

- <https://testdriven.io/blog/django-vs-flask>
- <https://docs.djangoproject.com/en/stable/topics/auth>
- <https://docs.djangoproject.com/en/5.0/ref/csrf>

2. Communication

Technologies :

- Django Channels : Permet d'ajouter des fonctionnalités de communication en temps réel à Django, notamment pour les messageries instantanées.
- WebSockets : Utilisé pour établir une communication bidirectionnelle entre le serveur et le client, essentielle pour les notifications en temps réel et la messagerie instantanée.

Avantages :

- Communication en temps réel fiable et scalable.
- Possibilité d'épingler des messages et de gérer les notifications pour les messages non lus.

Sources :

- <https://channels.readthedocs.io/en/stable/>
- https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

3. Calendrier

Technologies :

- Django-ORM : Permet d'interagir avec votre base de données de manière intuitive et efficace en simplifiant les requêtes.
- JavaScript : Pour une gestion dynamique et réactive des rendez-vous coté client.

Avantages :

- Flexibilité et extensibilité pour ajouter/modifier des rdv.
- Gestion centralisée des événements.
- Affichage dynamique et intégration de snippets simples.

Sources :

- <https://opensource.com/article/17/11/django-orm>
- https://developer.mozilla.org/JavaScript_basics
- <https://www.blogduwebdesign.com/calendriers-html-javascript.html>

4. Listes de tâches et notes

Technologies :

- Django Models : Pour la création de modèles de données robustes permettant de gérer des listes de tâches et des notes/memos.
- JavaScript : Pour une gestion dynamique et réactive des listes de tâches et le plugin open-source Slate permet d'intégrer un éditeur de texte coté front-end.

Avantages :

- Flexibilité et extensibilité pour ajouter/modifier des tâches et des notes.
- Affichage dynamique et responsive comprenant des outils d'édition de texte.

Sources :

- <https://docs.djangoproject.com/models/>
- <https://docs.djangoproject.com/javascript/>
- <https://github.com/slate>

5. Base de Données

Technologies :

- PostgreSQL : Une base de données relationnelle open-source, puissante et fiable, offrant des fonctionnalités avancées telles que les transactions ACID, les jointures complexes, et le support pour JSON.

Avantages :

- Robustesse et scalabilité pour gérer tous les volumes de données.
- Support pour les fonctionnalités avancées de recherche et de filtrage.

Sources :

- <https://www.postgresql.org/about/>

6. Front-End

Technologies :

- CSS : Pour styliser les pages web et assurer une présentation visuelle attrayante et responsive.
- JavaScript : Pour ajouter des interactions dynamiques et rendre l'expérience utilisateur plus fluide. Utilisation de "vanilla" JavaScript pour la simplicité et la légèreté.

Avantages :

- Simplicité et légèreté, facilitant le développement et la maintenance sur des projets de petite échelle.
- Chargement rapide des pages grâce à une utilisation minimale de bibliothèques tierces (ex: bootstrap).
- Facilité de personnalisation et de mise à jour des styles et des interactions.
- Affichage responsive avec la règle CSS @media(max-width: 768px)

Sources :

- <https://www.ironhack.com/differences-html-css-javascript>
- <https://www.easybear.fr/blog/reussir-son-responsive-design>

Liste de sites similaires : www.classdojo.com, www.remind.com, <https://seesaw.com>, <https://openclassrooms.com>, <https://meet.jit.si>, <https://eu.bbcollab.com>, www.zoho.com, <https://whereby.com>

Spécifications techniques

Nous aurions pu envisager un déploiement PHP-Symphony ou NodeJS mais notre équipe a une plus grande expertise de développement en Python. De plus, Python est un excellent choix pour le développement de ce type de projet grâce à sa flexibilité, sa facilité de lecture et sa vaste gamme de frameworks. Ce choix garantit de réduire les coûts de développement tout en offrant à l'association HomeSkolar une plateforme à grande longévité, clés en main à la fois souple et robuste.

- **Back-End**

- ◇ **Django** : Django existe depuis 2005, il bénéficie d'une large documentation, d'une grande communauté et est constamment mis à jour. Ce framework offre une structure plus légère et complète que Flask et possède des fonctionnalités intégrées telles que l'authentification et l'administration qui sont les premiers piliers du projet.
- ◇ **Librairies** : Afin de gérer les interactions avec la base de données de manière efficace et sécurisée nous privilégierons Django-ORM. Cette librairie permettra de centraliser la gestion des événements et de générer les autorisations et les permissions automatiquement. Et nous emploierons Django-Channels pour la mise en place de la messagerie en temps réel.

Le fait que ces modules existent déjà au travers du framework Django assurera à notre équipe de développement un gain de temps conséquent.

- **Front-End**

- ◇ **HTML CSS / Javascript** : Les pages html à afficher côté client ne devront pas nécessairement répondre à un framework css. Le projet nécessitera en revanche de créer des règles de styles css ainsi que du développement javascript qui devront répondre aux standards modernes. De nombreux snippets html/css/javascript existent sur le web, permettant de gagner encore plus de temps. Cela assurera un développement rapide du front-end et l'aboutissement d'un site responsive, rapide et dynamique.

- **BDD**

- ◇ **PostgreSQL** : est parfaitement adapté à l'exploitation des bases de données. Il prend notamment en charge en toute sécurité les opérations complexes, en donnant la priorité à la conformité et à l'extensibilité.

Recommandations

- **Sécurité** : Implémenter des pratiques de sécurité avancées, notamment la protection contre les attaques XSS et CSRF.
- **Scalabilité** : Prévoir des mécanismes de mise en cache et de gestion de la charge pour garantir la performance de l'application.
- **Tests et Validation** : Mettre en place une stratégie de tests exhaustive pour valider la fonctionnalité et la performance de chaque module de l'application.

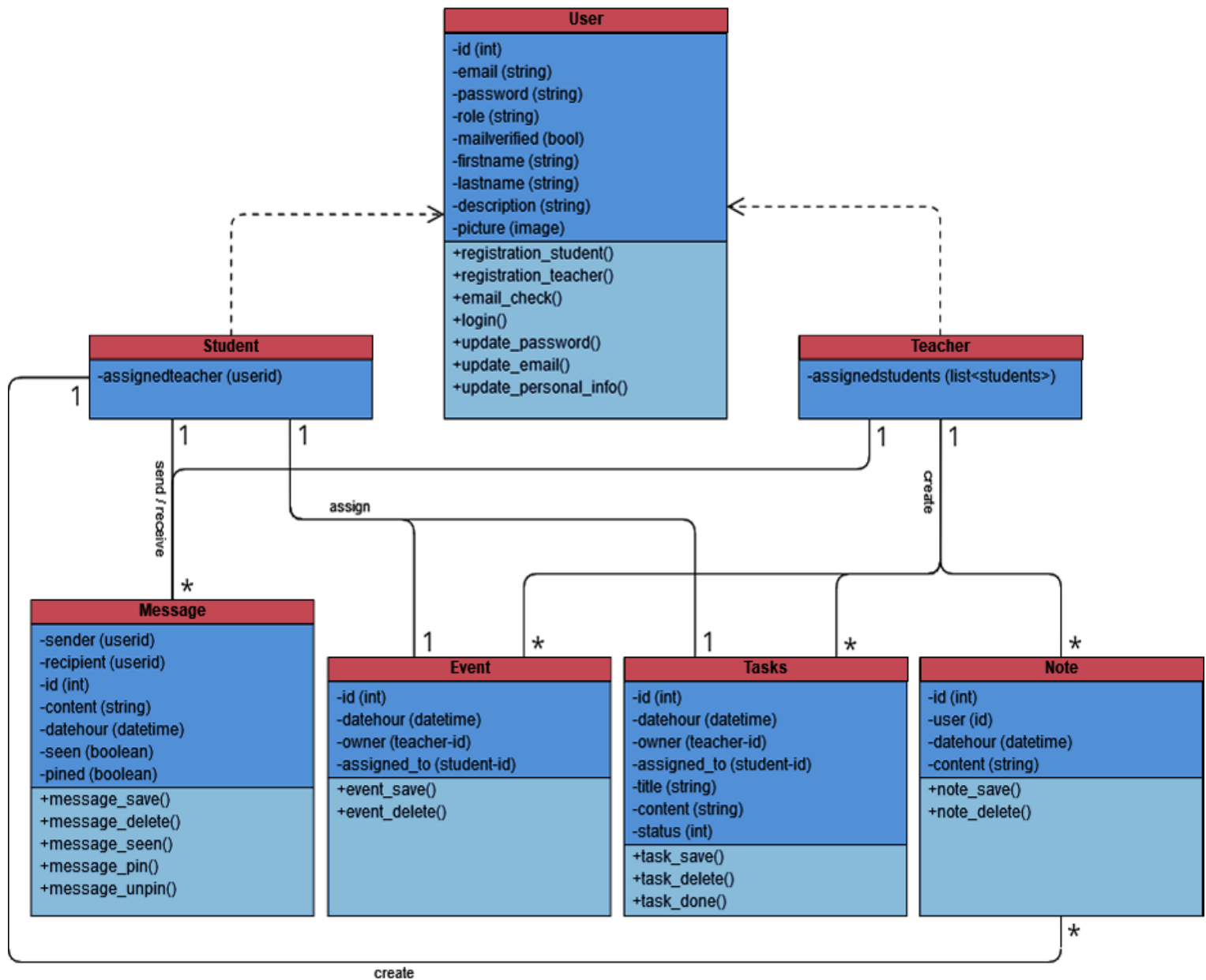
Conclusion

Ce projet repose sur une stack technologique éprouvée et intégrée, combinant la puissance de Django pour le back-end avec la flexibilité et l'interactivité offertes par les technologies front-end JavaScript / CSS. La communication en temps réel, les fonctionnalités de calendrier et de listes de tâches/notes garantissent une expérience utilisateur dynamique et intuitive. L'utilisation de PostgreSQL comme base de données assure une gestion des données robuste et performante. En adoptant ces technologies et pratiques, la plateforme de mise en relation élèves-tuteurs pourra offrir une expérience utilisateur fluide et sécurisée, répondant aux besoins du client et des usagers tout en étant scalable et maintenable à long terme.

Améliorations possibles

- Django-Scheduler pour une gestion de calendrier plus poussée et l'utilisation de Django-Webpush pour les notifications.
- FullCalendar pour afficher et gérer des calendriers interactifs, compatible avec la plupart des frameworks front-end.
- Django-Allauth pour l'authentification et la gestion des utilisateurs, offrant des fonctionnalités telles que la vérification par email, la réinitialisation de mot de passe, et l'intégration des réseaux sociaux.
- OAuth2 pour la gestion des permissions et des rôles, notamment pour permettre à l'admin d'attribuer des rôles spécifiques.
- Fonction listant automatiquement au tuteur ses étudiants sans rendez-vous.
- Créer des canaux de discussion permettant à plusieurs utilisateurs de participer.
- WebRTC pour permettre des entretiens en visioconférences.

Diagramme de classes UML



Piste de réflexion UX

- **L'authentification**

Créer un compte :

Register

First Name*

Last Name*

Email Address*

Phone Number*

Password*

Confirm Password*

Applying as : Student ☐

Register

Login :

Login

Email Address


Password

Forgot password ?

LoginRegister

Modification des données personnelles :

DashboardMessengerNotesPersonal Data



Update Picture

Change Password

FirstName :

Myfirstname

Last Name :

Mylastname

Phone Number :

0102030405

Email :

mymail@address.com

Save

- **La communication**

Messagerie :

Dashboard

Messenger

Notes

Personal Data

Real-time Messenger [Group 1]

Session started 01.01.2025 at 10:31

Penpal's Name - 10:32

Hi, how are you ?

10:33 - You

Hi, I'm fine, how about you ?

Send a message ...

Group members

You

Penpal's Name

Offline User

Quit Chat

Pin / unpin :

Dashboard

Messenger

Notes

Personal Data

Pinned Messages :

Penpal's Name - 11.03.2025, 09:37

The future belongs to those who believe in the beauty of their dreams.

Penpal's Name - 24.01.2025, 10:52

A mind is like a parachute. It doesn't work if it isn't open.

Unpin message

Online Groups :

Online Group 1

Quit

Online Group 2

Join

Real-time Messenger [Group 1]

2 unread messages

Notification :

Dashboard

Messenger

Notes

Personal Data

Tasks :

Project 3 - 2025.05.24

... work in progress

• Correct the technical specifications

• Correct the UML class diagram

• Review technological monitoring

Project 2, Pitch - 2025.05.18

✓ done

• Deliver project files on the platform

• Prepare a PowerPoint presentation

• Practice speaking

Events :

Wednesday 9 november 2025

13:30

• Meeting ~ With : **John Carmack**

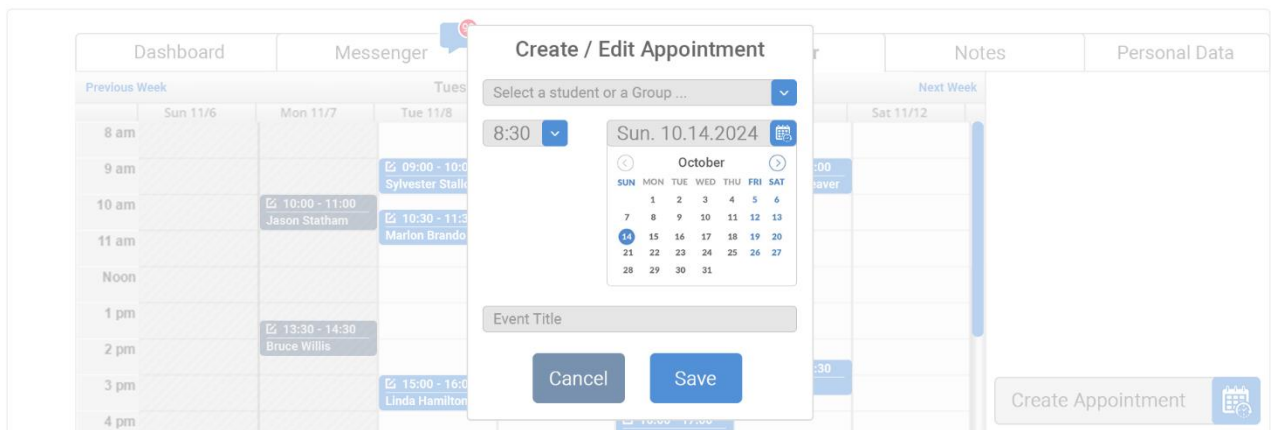
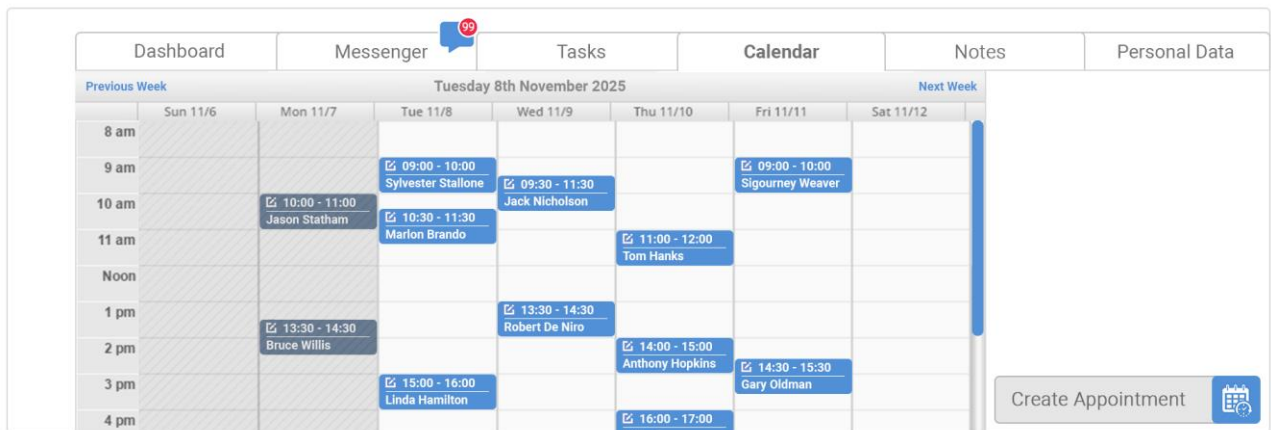
Monday 14 november 2025

10:00

• Group conference ~ Subject : **Motivation**

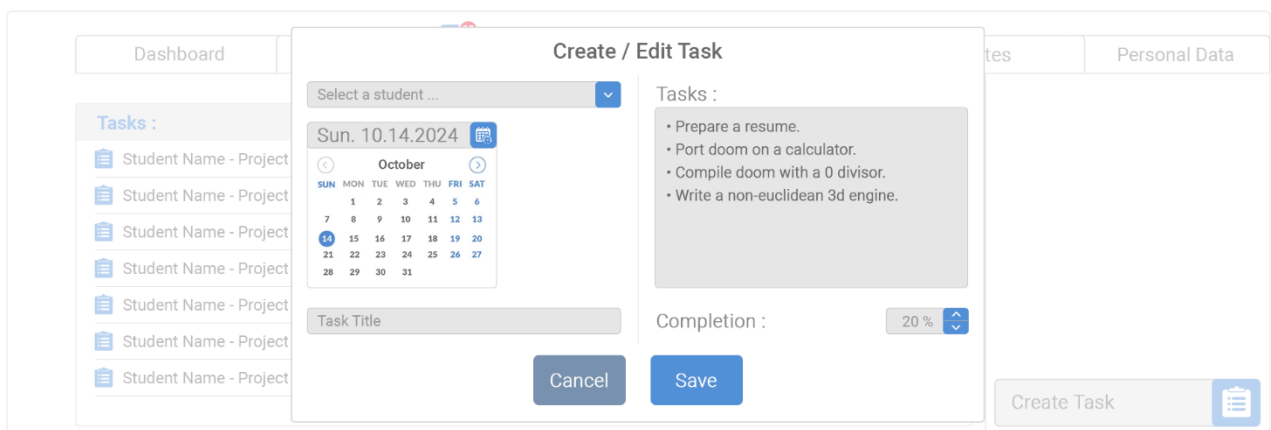
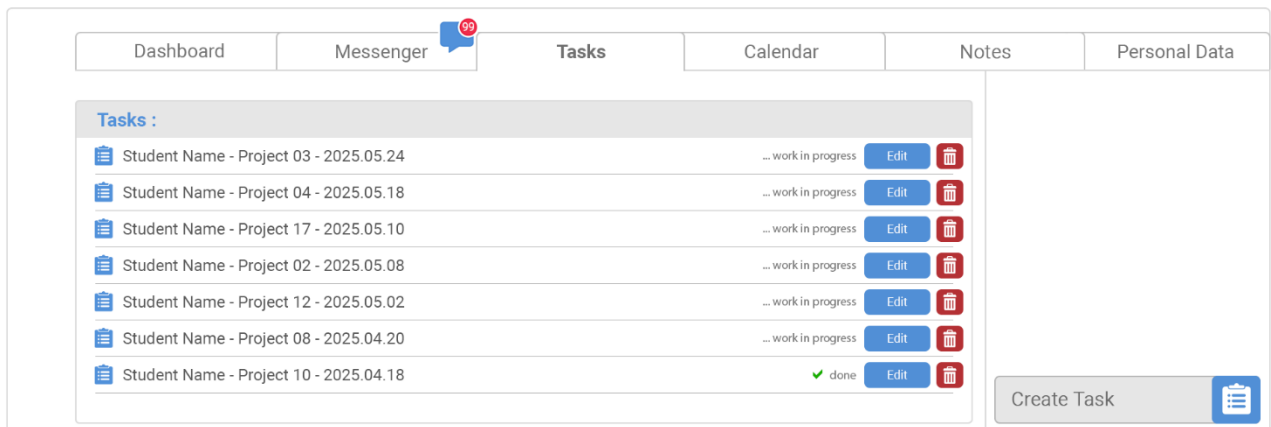
- Le calendrier

Prise de rdv (spécifique tuteur) :



- Liste des tâches

Ajout et modification de tâches (spécifique tuteur) :



- **Les notes**

Ajout et modification de notes :

The interface shows a navigation bar with 'Dashboard', 'Messenger' (with a red notification bubble), 'Notes', and 'Personal Data'. The 'Notes' section is active, displaying a list of 'Saved notes' on the left and a detailed view of 'Note Title 05' on the right. The note content describes the engine's rendering process for walls, mentioning the BSP tree, segs, and a static limit of 256. A 'Save' button is located at the bottom right of the note editor.

- **Proposition de fonctionnalité**

Liste d'étudiants sans rendez-vous (spécifique tuteur) :

Pour faciliter la prise de rdv par le tuteur et identifier plus rapidement les étudiants ayant besoin d'un rendez-vous, nous proposons au client d'intégrer à la plateforme une gestion de rendez-vous automatique et intelligente basée sur le statut de l'étudiant (`student_has_active_meeting = bool`). Nous sommes convaincus que cette fonctionnalité sera un atout en apportant fluidité dans le suivi et confort d'utilisation pour les tuteurs.

The interface shows a navigation bar with 'Dashboard', 'Messenger' (with a red notification bubble), 'Tasks', 'Calendar' (with a red notification bubble), 'Notes', and 'Personal Data'. The 'Calendar' section is active, displaying a weekly calendar view for November 2025. The calendar shows appointments for various students, including Sylvester Stallone, Jack Nicholson, Sigourney Weaver, Jason Statham, Marlon Brando, Tom Hanks, Robert De Niro, Anthony Hopkins, Gary Oldman, and Linda Hamilton. A sidebar on the right lists 'Students waiting for an appointment' with three entries: 'Student Name 1', 'Student Name 2', and 'Student Name 3'.