

Documentation du Projet DOODLE

GANAME Abdoulaye Plateforme Éducative Web

1 Présentation du Projet

Résumé

DOODLE est une plateforme web d'apprentissage en ligne conçue pour faciliter la gestion des cours entre professeurs et étudiants. Elle intègre notamment :

- La consultation des cours adaptés au niveau des étudiants,
- Un système de postulation et d'inscription validé par les professeurs,
- Une messagerie interne sécurisée,
- La notation et les commentaires des cours,
- La gestion autonome des cours et des élèves par les enseignants.

2 Fonctionnalités Principales

Gestion des utilisateurs

- **Étudiants** : inscription, consultation, postulation, messagerie, notation.
- **Professeurs** : gestion des cours, validation des postulations, interaction avec étudiants.

Catalogue des cours

- Cours filtrés par niveau d'étude.
- Lien explicite entre un cours et son professeur.

Postulation et inscription

- Postulation par interface dédiée.
- Validation par le professeur.
- Inscription automatique en cas d'acceptation.

Messagerie interne

- Système sécurisé avec suivi des messages lus / non lus.
- Historique accessible dans un espace dédié.

Notation et commentaires

- Notes de 1 à 5.
- Commentaires libres laissés par les étudiants.

3 Structure du Projet

Applications Django

- Etudiants, Professeur, Cours, messagerie, login, Accueil.

Modèles principaux

- **Étudiants, Professeurs** : extensions du modèle User.
- **Cours** : titre, description, niveau, professeur, etc.
- **Postulation** : statut, date, lien étudiant-cours.
- **Classe** : gestion des étudiants validés.
- **Message** : expéditeur, destinataire, contenu, date, état.

4 Installation et Pré-requis

Environnement

- Python 3.12+, Django 5.0+
- Base de données : MySQL ou MariaDB
- Configuration des médias via `settings.py`

Configuration de la base de données MySQL via XAMPP / WAMP

Avant de lancer les migrations Django, il est nécessaire de préparer la base de données MySQL. Vous pouvez utiliser les environnements de développement XAMPP ou WAMP, qui intègrent MySQL.

Étapes à suivre

1. Démarrez le serveur MySQL depuis XAMPP ou WAMP.
2. Accédez à l'interface **phpMyAdmin** via votre navigateur à l'adresse `http://localhost/phpmyadmin`.
3. Créez une nouvelle base de données nommée `doodle_new` en choisissant le jeu de caractères `utf8mb4_general_ci`.
4. Cliquez sur longlet **Importer**.
5. Sélectionnez le fichier `doodle_new.sql` situé dans le dossier `source` du projet.
6. Lancez l'importation.

Une fois la base importée, assurez-vous de bien configurer la connexion MySQL dans le fichier `settings.py` de `GLC_projet`, par exemple :

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'doodle_new',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```

Remarque

Le mot de passe de l'utilisateur **root** est généralement vide dans les configurations par défaut de XAMPP et WAMP. Si vous l'avez modifié, mettez à jour le champ **PASSWORD**.

Installation

```
pip install -r requirements.txt
```

Configurer la base dans **settings.py**, puis :

```
python manage.py migrate
python manage.py createsuperuser #Pour créer un admin
python manage.py runserver
```

5 Utilisation et Parcours Utilisateur

Étudiant

- Inscription et connexion.
- Consultation, postulation, messagerie, notation.

Professeur

- Connexion, création des cours, ajout de ressources.
- Gestion des postulations, classes, messagerie, feedbacks.

6 Architecture Technique

Extraits de Modèles

```
class Postulation(models.Model):
    etudiant = models.ForeignKey(Etudiants, on_delete=models.CASCADE)
    cours = models.ForeignKey(Cours, on_delete=models.CASCADE)
    statut = models.CharField(...)
    date_postulation = models.DateTimeField(auto_now_add=True)
```

Relations

- Un cours est dispensé par un professeur.
- Un étudiant postule à plusieurs cours.
- Une fois acceptée, il est inscrit via le modèle **Classe**.

7 Migrations et Données

```
python manage.py makemigrations
python manage.py migrate
```

8 Personnalisation

- Adapter les niveaux d'étude.
- Modifier les formats de ressources.
- Adapter le design via `/templates` et `/static/css`.

9 Sécurité et Bonnes Pratiques

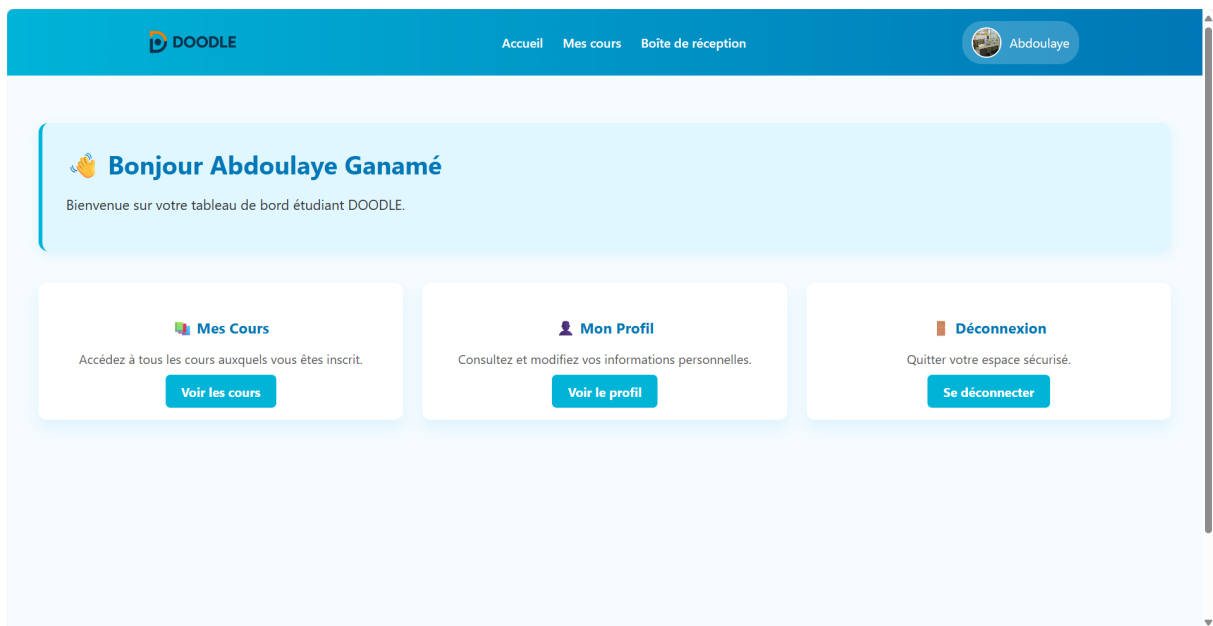
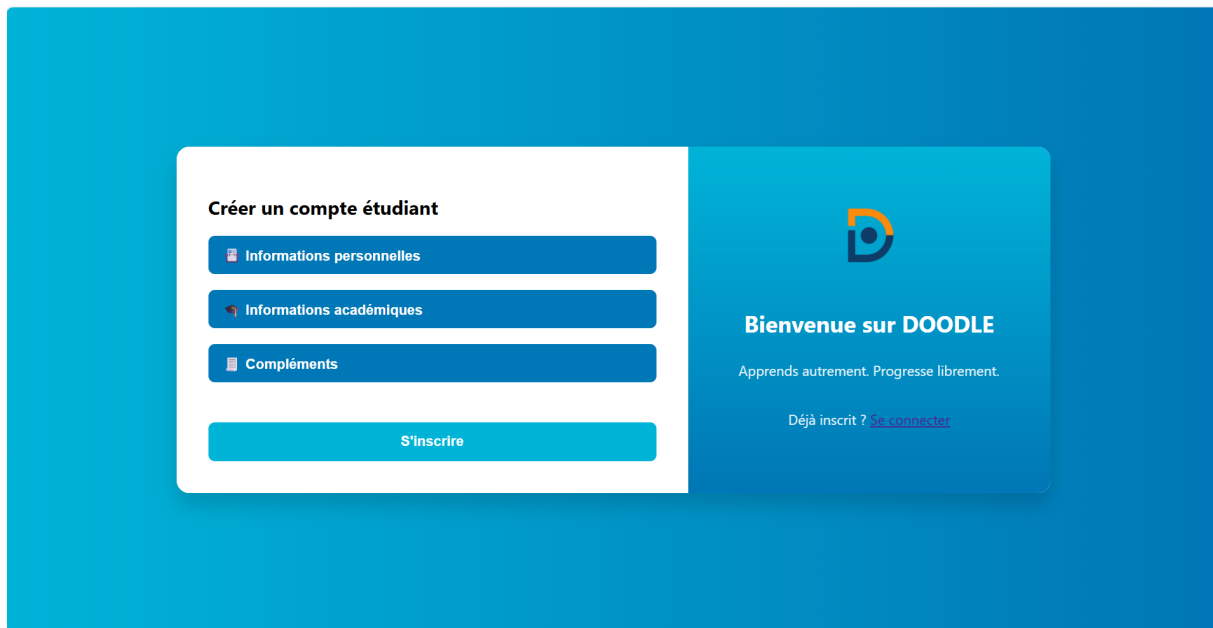
- Hashage des mots de passe par Django.
- Sécurisation des vues via `\login_required`.
- Vérification des droits d'accès.


10 Extensions Possibles

- Intégration de paiement.
- Notifications par mail ou push.
- Messagerie enrichie (groupes, fichiers).
- Statistiques pour les professeurs.


11 Annexes et Visualisations






 DOODLE

Accueil Mes cours Boîte de réception

 Abdoulaye




Traitement des donnees

Ce cours couvre les techniques de traitement et de manipulation des donnees...

Niveau : Licence 1 | 15 heures

Par : **Kouame Dares**

[Contacter le professeur](#)[Postuler](#)




Econometrie des series temporelles

Ce cours traite des techniques econometriques...

Niveau : Licence 1 | 22 heures

Par : **Gbenro Nathanael**

[Contacter le professeur](#)[Postuler](#)




Programmation web

Ce cours introduit les concepts fondamentaux...

Niveau : Licence 1 | 25 heures

Par : **Bobet Goualo Victorien**

[Contacter le professeur](#)[Inscrit](#)[Voir le cours](#)







Data Visualisation

Ce cours enseigne les principes et outils de visualisation...


Niveau : Licence 1 | 20 heures


Par : **Kofi Mensah**

[Contacter le professeur](#)[Postuler](#)




Conversations

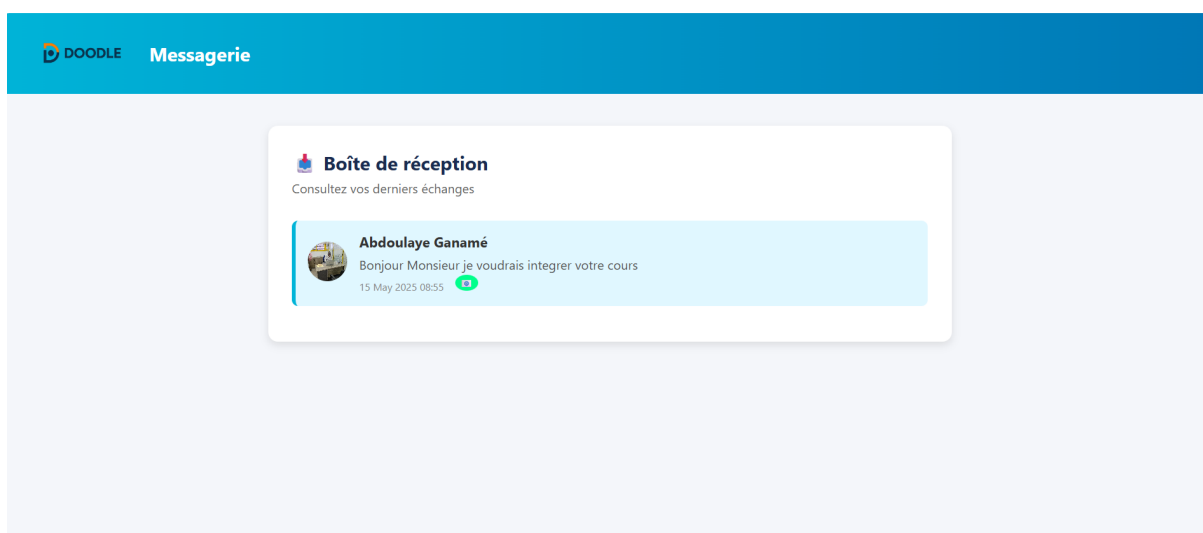
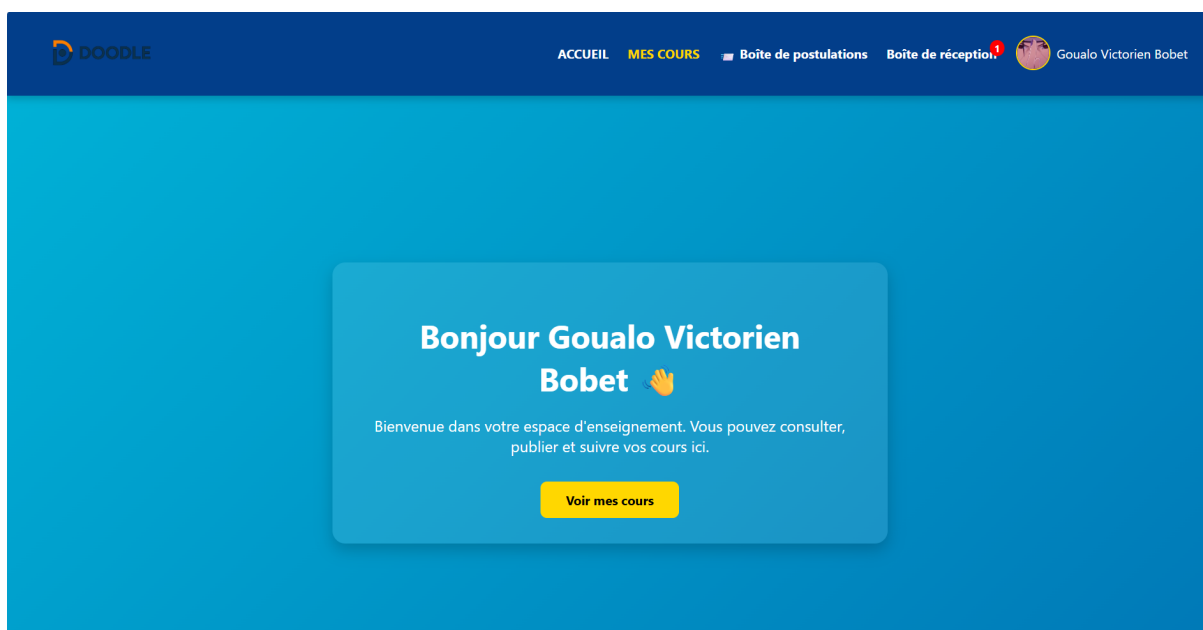
 Goualo Victorien Bobet

 **Goualo Victorien Bobet**

Bonjour Monsieur je voudrais integrer votre cours
15 May 08:55

Écris ton message ici...





Conclusion sur les apports en termes de programmation Web

Ce projet a permis d'explorer et d'appliquer diverses techniques essentielles en programmation web, permettant de concevoir une plateforme robuste, moderne et interactive. L'intégration de Django, une des bibliothèques Python les plus puissantes pour le développement web, a facilité la création de pages dynamiques et interactives, la gestion des utilisateurs et des sessions, ainsi que la mise en place d'un modèle de données relationnel solide.

L'un des principaux apports réside dans la gestion des interactions entre les utilisateurs via des systèmes de messagerie en temps réel, permettant ainsi un environnement collaboratif efficace entre les professeurs et les étudiants. Ce système a mis en évidence l'utilisation des modèles et des vues Django pour manipuler et afficher les messages, ainsi que la gestion des permissions pour garantir la sécurité des échanges.

La gestion des ressources associées aux cours, comme la possibilité d'ajouter des ressources et de notifier les étudiants, a introduit la notion de notifications dynamiques et a renforcé la compréhension de la gestion des formulaires et des requêtes POST. En outre, l'intégration des différents mécanismes de filtrage (comme l'affichage des cours en fonction du niveau de l'étudiant) a montré l'importance de la logique de filtrage au sein de Django pour répondre aux besoins spécifiques des utilisateurs.

Le projet a également permis de mettre en place des interfaces élégantes et intuitives en utilisant du HTML, CSS et du JavaScript, avec un design inspiré de l'univers Doodle, renforçant l'expérience utilisateur (UX). L'intégration de fonctionnalités comme les boutons de postulation, les notifications en temps réel et la gestion des profils ont contribué à créer une plateforme fluide et interactive.

Enfin, ce projet a renforcé les compétences en gestion de base de données, avec une attention particulière portée à la gestion des relations entre différentes entités (utilisateurs, cours, postulations, etc.). Le tout a été structuré pour assurer une navigation intuitive et une logique de traitement des données efficace, ouvrant la voie à une expérience utilisateur sans faille.

En résumé, ce projet a été une opportunité significative d'appliquer des principes de développement web tout en intégrant des fonctionnalités complexes qui répondent à des besoins réels, tout en mettant l'accent sur l'interactivité, la sécurité et la gestion de contenu dynamique.
