Parcours: Développeur d'application python

OPENCLASSROOMS

# PROJET 10: Créez une API sécurisée RESTful en utilisant Django REST



Réalisé par : **Sabah EL-AOUNI**

Mentor : **Idriss Ben Geloune**

# PLAN DE LA PRÉSENTATION

- Présentation globale du projet
- Présentation des points de terminaison avec Postman
- Présentation de la documentation de l'API REST.
- Respect des normes de sécurité OWASP
- Conclusion

# Présentation globale du projet 1/3

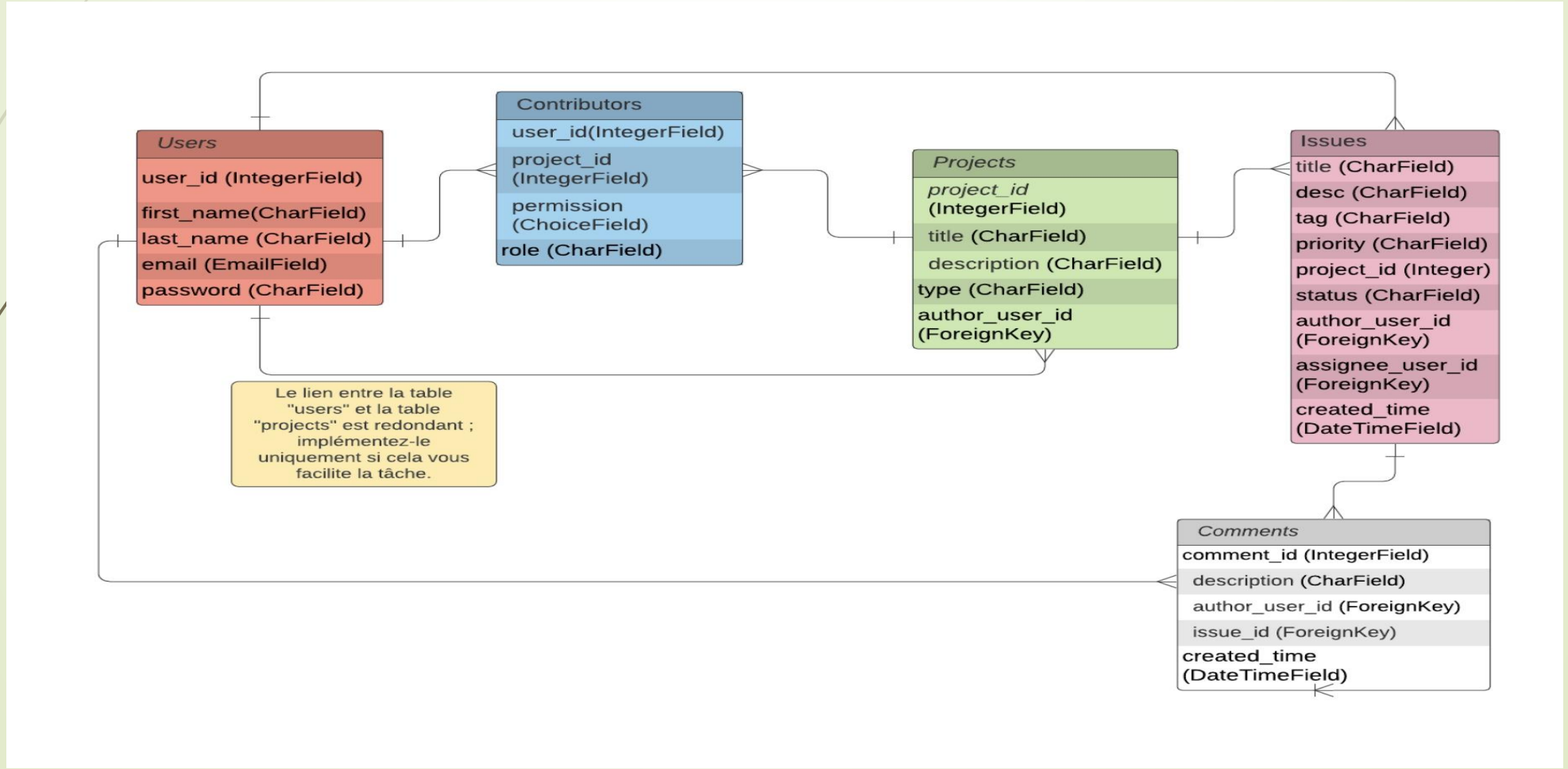
- L'objectif est de développer une API servant des données de gestion et suivi des problèmes liés à des projets
- Il s'agit d'une application de suivi des problèmes qui permet principalement aux utilisateurs de créer divers projets, de leur associer des contributeurs, de créer des problèmes au sein de ces projets et d'attribuer des commentaires à ces problèmes.

**Technologies / outils utilisées:**

- Framework Django REST ;
- SQLite comme base de données de développement locale,
- Postman pour les requêtes
- Pycharm comme IDE

# Présentation globale du projet 2/3

- Le schéma de la base de données:



# 19 requêtes :

| # | Point de terminaison d'API | Méthode HTTP | URI |
|---|---|---|---|
| 1. | Inscription de l'utilisateur | POST | /signup/ |
| 2. | Connexion de l'utilisateur | POST | /login/ |
| 3. | Récupérer la liste de tous les projets (projects) rattachés à l'utilisateur (user) connecté | GET | /projects/ |
| 4. | Créer un projet | POST | /projects/ |
| 5. | Récupérer les détails d'un projet (project) via son id | GET | /projects/{id}/ |
| 6. | Mettre à jour un projet | PUT | /projects/{id}/ |
| 7. | Supprimer un projet et ses problèmes | DELETE | /projects/{id}/ |
| 8. | Ajouter un utilisateur (collaborateur) à un projet | POST | /projects/{id}/users/ |
| 9. | Récupérer la liste de tous les utilisateurs (users) attachés à un projet (project) | GET | /projects/{id}/users/ |
| 10. | Supprimer un utilisateur d'un projet | DELETE | /projects/{id}/users/{id} |
| 11. | Récupérer la liste des problèmes (issues) liés à un projet (project) | GET | /projects/{id}/issues/ |
| 12. | Créer un problème dans un projet | POST | /projects/{id}/issues/ |
| 13. | Mettre à jour un problème dans un projet | PUT | /projects/{id}/issues/{id} |
| 14. | Supprimer un problème d'un projet | DELETE | /projects/{id}/issues/{id} |
| 15. | Créer des commentaires sur un problème | POST | /projects/{id}/issues/{id}/comments/ |
| 16. | Récupérer la liste de tous les commentaires liés à un problème (issue) | GET | /projects/{id}/issues/{id}/comments/ |
| 17. | Modifier un commentaire | PUT | /projects/{id}/issues/{id}/comments/{id} |
| 18. | Supprimer un commentaire | DELETE | /projects/{id}/issues/{id}/comments/{id} |
| 19. | Récupérer un commentaire (comment) via son id | GET | /projects/{id}/issues/{id}/comments/{id} |

# Présentation globale du projet 3/3

Respect des mesures de sécurité **OWASP** par le back-end en appliquant le processus **AAA** *(Authentication, Authorization, Accounting)* :

- **Authentification :** utilisation de JWT (JSON Web Token) pour le back-end d'authentification du framework Django REST. Seuls les utilisateurs authentifiés doivent être en mesure d'accéder à quoi que ce soit dans l'application
- **Autorisation:** décider si *l'utilisateur authentifié* est autorisé à accéder à une ressource. Dans l'application, un utilisateur ne doit pas être autorisé à accéder à un projet pour lequel il n'est pas ajouté en tant que contributeur.
- **Accès : -** Les commentaires doivent être visibles par tous les contributeurs au projet et par le responsable du projet, mais seul leur auteur peut les actualiser ou les supprimer.

- Un problème ne peut être actualisé ou supprimé que par son auteur, mais il doit rester visible par tous les contributeurs au projet.

- Il est interdit à tout utilisateur autre que l'auteur de demander une mise à jour et de supprimer des demandes sur un projet/problème/commentaire.

# Démonstration

- Présentation globale des requêtes sur postman :
- Authentification/ inscription
- GET/PUT/DELETE d'un projet/problème/commentaire
- Documentation de l'API

- Explication du code
- Respect des normes OWASP
- Permissions

# Conclusion

- Nouvelle compétence développée /Approfondies:
  - Django REST
  - Comprendre la logique des requêtes
  - Manipulation de l'outil postman
  - Norme de sécurité OWASP
  - Base de données SQLite (query)