

Enseignant(s)

**BALASOUPRAMANIANE Amrta Devy**

**DAVID Aurélien**

Email(s)

[abalasoupramaniane1@myges.fr](mailto:abalasoupramaniane1@myges.fr)

[contact@spyl.net](mailto:contact@spyl.net)

## Projet libre

### 1 Matières, formations et groupes

Matière liée au projet :

Formations : -

Nombre d'étudiant  
par groupe :

**2 à 3**

Règles de constitution des groupes: **Imposé**

Charge de travail  
estimée par étudiant : **15,00 h**

### 2 Sujet(s) du projet

Type de sujet : **Imposé**

#### Sujet libre avec fonctionnalités attendues

Objectif n°1 : développer une API GraphQL la plus complète possible ! Une complexité minimum est attendue :

- utiliser les différentes parties de GraphQL vues en cours (queries, types, enums, mutations, arguments, identification global, connexion)
- une BDD avec plusieurs entités/reliations avec CRUDs
- gestion de droits d'accès, erreurs

Objectif n°2 : consommer l'API avec un client (simple) sous forme de front-end ou application mobile. Une complexité minimum est attendue :

- plusieurs pages/écrans avec chargement de données
- plusieurs interactions nécessitant des appels à l'API

### 3 Détails du projet

**Objectif du projet (à la fin du projet les étudiants sauront réaliser un...)**

Développer une API GraphQL complète

Titre : Expert en Ingénierie Sciences U

Bloc 5 : Administrer et optimiser les données d'un système d'information

Critères d'évaluation :

\_ Un flux de données en temps réel d'une ou plusieurs bases est maintenu. Les performances de la BDD sont améliorées. Les outils d'audit et de contrôle sont mis en place. L'assistance technique (méthode, produit...) est fournie, suivant le domaine d'intervention, aux équipes de production ou d'études et aux utilisateurs [Audit, contrôle et performances de BDD - Formation]

\_ Le volume de stockage et la gestion des flux est surveillé. L'activité du système d'information est monitorée. Les données sont stockées et gérées. [Monitoring des volumes, stockage et flux de données]

### Descriptif détaillé

Objectif n°1 : développer une API GraphQL la plus complète possible ! Une complexité minimum est attendue :

- utiliser les différentes parties de GraphQL vues en cours (queries, types, enums, mutations, arguments, identification global, connection)

- une BDD avec plusieurs entités/relation avec CRUDs

- gestion de droits d'accès, erreurs

Objectif n°2 : consommer l'API avec un client (simple) sous forme de front-end ou application mobile. Une complexité minimum est attendue :

- plusieurs pages/écrans avec chargement de données

- plusieurs interactions nécessitant des appels à l'API

**Ouvrages de référence (livres, articles, revues, sites web...)**

**Outils informatiques à installer**

**4**

**Livrables et étapes de suivi**

**1**

**Rendu final**

**Rendu final du projet**

**Soutenance**

- Partie en groupe

- Présentation du projet et des fonctionnalités réalisées.

- Présentation de l'API, de son implémentation et de ses problématiques.

- Points bonus pour la qualité du code, les tests fonctionnels et unitaires,

une API bien conçue, UX / design propre, etc...

- Partie en individuelle

- Quelques questions de cours !

**mardi  
05/11/2024  
0h00**

**5**

**Soutenance**

Durée de présentation  
par groupe :

**20 min**

Audience : **A huis clos**

Type de présentation :

**Démonstration**

Précisions :