

# Projet de conception de serveur web

## Introduction

Ce projet a pour but de vous permettre de mettre en pratique les cours de programmation de serveur web. Le projet comprend deux parties :

- 1- La première partie, qui fait l'objet de ce document, exclusivement sur la conception de l'API qui permet de donner les informations relatives à toutes les routes. Cette partie porte sur le backend et doit utiliser obligatoirement GitHub, Node/Express, sequelize (MySQL) et Postman.
- 2- La deuxième partie portera sur l'interface graphique réalisée avec ejs ou vuejs et fera l'objet d'un autre document dans quelques semaines.

## Constitution des équipes

Chaque équipe doit avoir **obligatoirement** entre **trois** et **quatre** membres et vous êtes libres de vous mettre avec les personnes de votre choix.

## Sujet du projet

Chaque équipe est libre de choisir son sujet et je recommande de choisir des projets qui peuvent être montrés plus tard à des entreprises car à la fin du semestre vous aurez une application complète du backend au frontend.

## Exigences

- 1- Chaque équipe doit obligatoirement avoir des tables permettant l'authentification et l'autorisation (utilisateurs et rôles par exemple)
- 2- Chaque membre de l'équipe doit avoir au moins **deux tables** dont il doit développer les API avec un nombre minimal de six tables par projet. Donc si un groupe est seulement composé de deux membres chaque membre devra avoir au moins **trois tables**.

## Echéances du projet

Pour les différentes étapes du projet, suivre le calendrier ci-dessous et les résultats de chaque semaine doivent être montrés au professeur en classe.

### Semaine du 27 octobre

- Etablir le modèle entité-associations de votre base de données (vous pouvez utiliser [lucidchart](#) pour le diagramme). Voir cette [vidéo](#).

- En déduire le modèle physique de votre base de données (vous pouvez utiliser lucidchart pour ce diagramme).
- Créer un repo sur [GitHub](#) qui servira de base pour votre code
- Donner accès au professeur à votre GitHub
- Soumettre dans le pigeonnier vos résultats ou les montrer au professeur en classe.

### Semaine du 3 novembre

- Créer les modèles de toutes vos tables sans oublier les relations entre elles dans express
- Créer les controllers pour toutes actions qui seront utilisées dans votre application
- Créer toutes les routes requises pour vos controllers
- Ajouter la pagination et des variables de query dans les routes de type GET qui le nécessitent
- Utiliser les routes dans votre fichier racine (index.js ou server.js)
- Tester les routes dans Postman
- Soumettre dans le pigeonnier vos résultats ou les montrer au professeur en classe.

### Semaine du 10 novembre

- Ajouter les validations aux controllers en utilisant le module express-validator
- Ajouter l'authentification à l'API en utilisant le module JWT en utilisant les champs email et mot de passe de la table users
- Tester de nouveau les routes avec Postman
- Soumettre dans le pigeonnier vos résultats ou les montrer au professeur en classe.

### Semaine du 17 novembre

- Dernières modifications et dernier test avec Postman
- Mise jour final du repo GitHub
- Soumission du projet complet (**aucun retard ne sera accepté**)

## Critères d'évaluation

Le projet sera évalué suivant les critères suivants

Fonction	Pourcentage
Modèle entité-associations et model physique avec lucidchart	10%
Les modèles avec les relations (tables dans express)	10%
Les controllers (express) et validations	15%
Les routes (express)	5%
Authentification (express)	5%
Github	10%
Fichier Postman	5%