

# Projet Web/Bases de données

HOME

Infotech

PROJETS

CONNEXION



## Le hackerspace de Polytech Montpellier

Infotech a été créé par des étudiants, pour des étudiants dans le but de partager sur divers sujets informatiques. Le principe du hackerspace est que tout le monde peut venir, quelque soit son niveau, et peut :

travailler seul sur un projet personnel  
se faire aider sur un projet personnel  
proposer son aide pour travailler sur d'autres projets

### Experts

Vous connaissez 10 langages de programmation sur le bout des doigts, le 42 est votre nombre fétiche et l'été vous bronzes grâce à la lumière de votre(vos) écran(s) ? Venez partager votre expérience avec des personnes plus novices, ou même obtenir de l'aide de quelqu'un de plus chevronné (eh oui ça existe !) !

### Novices

Votre expérience de l'informatique se résume à surfer sur le web, mais vous voulez plus que ça ? Vous débutez et bataillez avec vos premiers programmes ? Pas de panique ! Infotech est là pour vous ! Des personnes plus expérimentées pourront partager leur expérience avec vous et vous permettre d'avancer.

### Short talks

Au début de chaque session Infotech un short talk (~15 minutes) sera organisé pour présenter un sujet se rapportant à l'informatique. Plutôt axés sur les nouvelles technologies, les short talks permettront de présenter ces nouvelles technologies et de donner envie de les approfondir.

### Projets

L'entraide sera un point central d'Infotech. C'est pourquoi ce site propose un système de partage de projets, pour permettre à chacun de pouvoir :

partager un projet pour lequel il voudrait de l'aide  
proposer son aide sur un projet qui l'intéresse

## 1. Analyse des besoins

### 1.1. Origine du projet

Le projet est né avec la création du club d'informatique de Polytech Montpellier (Infotech).

En effet, le projet d'Infotech est de pouvoir réunir des étudiants de Polytech autour de l'informatique et de projets qu'ils pourront partager.

Ainsi, il était nécessaire de créer une vitrine par laquelle les étudiants pourraient découvrir ce qu'est Infotech. En outre, Infotech avait besoin d'un outil permettant à ses adhérents de partager les projets qu'ils ont envie de réaliser. C'est pourquoi mon projet web a consisté à la réalisation d'un tel outil, sur lequel les utilisateurs pourront partager leurs projets.

### 1.2. Analyse de l'existant

Lors du démarrage du projet, une analyse de l'existant a été réalisée. Il en est ressorti que le projet devait démarrer de zéro car aucune application n'était existante au sein de Polytech, et aucune autre application offrant le même service n'a été trouvée.

L'application web d'Infotech a donc été réalisée « from scratch », sans application de référence pour s'orienter dans les choix techniques ainsi que les axes de développement.

## 2. Architecture logicielle

### 2.1. Choix techniques

#### 2.1.1. Base de données

Infotech n'ayant pas de financement, le choix de la base de données s'est dès le début du projet orienté vers des solutions libres.

C'est alors une base de données PostgreSQL qui a été retenue pour ses performances, sa grande stabilité ainsi que sa capacité à gérer les contraintes d'intégrité afin de garder à tout instant de la vie de la base de données sa cohérence.

La stabilité des bases de données PostgreSQL a donc également été un facteur déterminant dans le choix final car elle permet d'offrir une plus grande robustesse à l'application, qui ne peut fonctionner sans le support de sa base de données associée.

En revanche, un des inconvénients des bases de données PostgreSQL est leur faible rendement, par rapport à des bases de données MySQL par exemple, lors de traitements de faibles quantités de données, en effet la vérification de l'intégrité des données est très chronophage et son utilité plutôt pour de faibles volumes.

### **Modèle relationnel :**

RANK ( id, name, description, tokens\_required);

SECTION ( id, acronym, name);

ACCOUNT ( id, first\_name, last\_name, mail, pseudo, password, admin, tokens, #rank, # section);

PROJECT ( id, name, description, max\_helpers, start\_date, end\_date, achievement, #rank\_required, #owner);

PARTICIPATE ( #id\_project, #id\_helper );

Les informations liées aux utilisateurs sont stockées dans la table ACCOUNT, dont la section à laquelle appartient l'étudiant. Tous les mots de passe sont stockés cryptés et n'apparaissent à aucun moment en clair.

Les projets sont stockés dans la table PROJECT et contiennent des informations telles que : un nombre maximal de participants, des dates de début et fin, et le rang minimal requis pour pouvoir y participer.

Ce système de rang est mis en place grâce à la table RANK, mais n'est pas implémenté au-delà des rangs requis pour les projets dans l'application.

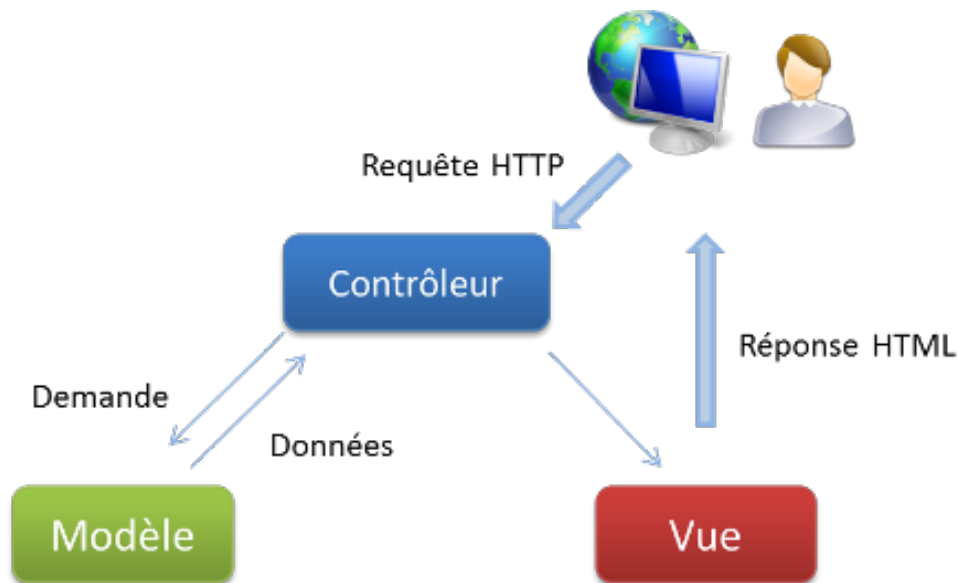
Les fichiers sources de la base de données sont disponibles à l'adresse suivante : <https://github.com/JohanBrunet/Infotech/tree/master/sql>

### **2.1.2. Pattern MVC**

Pour le développement de ce projet, c'est le pattern (patron de conception) MVC (Model-View-Controller) qui a été retenu. De ce fait, la partie métier (interactions avec la base de données) est indépendante de la manière de représenter les données côté client. Cette partie est laissée au contrôleur et à la vue, qui interagissent. L'architecture selon le pattern MVC peut donc se représenter de la manière suivante :

- Un événement survient sur la vue (clic sur un bouton), la vue est alors notifiée de l'action
- La vue transmet l'action au contrôleur qui le prend en charge
- Le contrôleur transmet la requête de la vue au modèle afin de récupérer les informations en base de données
- Lorsque le modèle a fini sa tâche, il renvoie les données au contrôleur
- Ce dernier est alors chargé de mettre en forme la vue avec les informations récupérées

Ci-dessous, un schéma présentant visuellement le pattern MVC :



Le pattern MVC permet d'avoir une gestion transparente pour l'utilisateur des interactions avec la base de données. En effet, il ne sait pas comment ses requêtes sont traitées et une modification du modèle n'entraînera pas de changement de comportement côté utilisateur.

### 2.1.3. Langage de programmation

N'ayant pas de contraintes sur le langage de programmation, il a été fait le choix de Node.JS, un langage performant, réactif et offrant d'importantes possibilités de maintenance et améliorations de l'application.

Ainsi ce langage est associé à un serveur qui permet d'exécuter du code JavaScript côté serveur. Cette possibilité apporte une grande capacité de réactivité de l'application, notamment grâce au système de callback.

Le système de callback permet d'exécuter du code de manière asynchrone et donc non bloquante. En effet, lorsqu'une fonction est appelée, on peut lui passer en paramètre une autre fonction, dite de callback, et qui sera exécutée en premier. Ce n'est que lorsque cette fonction de callback aura terminé sa tâche que la première fonction sera exécutée. Ce mécanisme est particulièrement intéressant pour les interactions avec la base de données car il est possible d'attendre la réponse de la base de données avant d'effectuer d'autres actions. Les fonctions sont donc stockées dans une pile d'appel et le serveur va les exécuter en les défilant une à une jusqu'à la dernière. Cette gestion des fonctions permet d'avoir une pile d'événements (ici fonction) traitée par un unique thread (processus), ce qui donne son côté réactif à Node.JS.

D'autre part, Node.JS est très performant dans la gestion des entrées /sorties (IO) car il les gère en mode non bloquant. C'est-à-dire que l'on peut télécharger deux fichiers simultanément, alors que dans d'autres cas il serait nécessaire d'attendre la fin du téléchargement du premier fichier avant de pouvoir commencer à télécharger le second. Cette possibilité est intéressante car, à terme, l'application Infotech est susceptible d'offrir un service de téléchargement de fichiers.

Enfin, Node.JS est très efficace dans la gestion des dépendances grâce à son Node Packet Manager (npm) qui permet de décrire un packet dans le fichier *package.json* (<https://github.com/JohanBrunet/Infotech/blob/master/package.json>) et d'en obtenir toutes les dépendances sans avoir à les appeler une par une.

Cela facilite la réutilisation de code et même son partage.

### 3. Architecture de déploiement

Infotech ne possédant pas l'infrastructure nécessaire pour héberger une application web, il a donc été nécessaire de trouver un hébergement distant pour l'application. De même que pour la base de données, Infotech a choisi de s'orienter vers une solution gratuite.

C'est l'hébergeur Heroku qui a été retenu par Infotech pour la mise en ligne de son application, car il propose une solution d'hébergement gratuit, un serveur prenant en charge Node.JS et une base de données PostgreSQL (qui sont les deux composantes majeures de l'application) et surtout un système de « webhook » (lien qui permet d'effectuer une action en réponse à un événement) permettant de déployer facilement et rapidement l'application.

En effet, le serveur Heroku sur lequel est hébergé l'application d'Infotech est lié au dépôt GitHub (système de gestion de versions) et le webhook permet de déployer automatiquement l'application sur le serveur lorsqu'une nouvelle version du code source est mise en ligne (push) sur le dépôt GitHub. Ce système permet de gagner du temps de développement car il n'y a pas à gérer le déploiement à la main.

Pour résumer : l'application est réalisée en local sur la machine du développeur, celui-ci « push » ses sources sur GitHub qui déclenche alors le webhook. Celui envoie les sources au serveur Heroku qui réalise un « build » (construction). Lors de cette phase de build, le serveur vérifie que les dépendances ont été importées et qu'elles sont cohérentes. Si le build est validé, l'application est déployée sur le serveur et est accessible par les utilisateurs.

## 4. Déroulement du projet

### 4.1. Difficultés rencontrées

La principale difficulté rencontrée au cours de ce projet a été l'apprentissage de Node .JS, qui m'était jusqu'alors inconnu. Il a ainsi fallu lire beaucoup de documentation afin de pouvoir appliquer au mieux les préceptes de ce langage.

De plus, le peu de temps accordé pour le développement de ce projet a constitué une difficulté majeure, notamment du fait du temps requis par l'apprentissage de nouvelles technologies. De plus, les choix techniques devaient être fait rapidement afin de pouvoir commencer le développement dans les plus brefs délais.

En outre, certains choix concernant l'application ont dû être faits, notamment d'abandonner certaines parties du site pour le moment.

### 4.2. Réussites

La principale réussite de ce projet a été l'apprentissage d'une nouvelle technologie, à savoir Node.JS et le développement de toute une application grâce à celle-ci. D'autre part, l'application des concepts vus en cours a été d'une grande aide tout au long de ce projet afin de pouvoir satisfaire au mieux les exigences du Web.

La création d'une application RESTful basée sur un pattern MVC est pour moi la plus grande réussite de ce projet.

### 4.3. Enseignements

Grâce à ce projet j'ai pu développer une application web bien que cela était une de mes bêtes noires. Ainsi ce projet m'a appris à respecter le Web (REST) et à me forcer à faire des choix rapidement au vu du temps imparti.

Enfin, ce projet m'a permis de me rendre encore plus compte de la valeur du travail d'équipe, c'est pourquoi je remercie tout particulièrement Théo Gauhous, Quentin Bouygues, Tom Somerville Roberts, Mehdi Fakihani et Charlène Pizzoni pour leur soutien et leur aide tout au long de ce projet.

Je remercie enfin M.Casteltort pour les cours dispensés ainsi que pour ses conseils d'expert qui m'ont été d'une aide précieuse.

## 5. Conclusion

Le projet final est quasi fonctionnel, mais certaines fonctionnalités restent à implémenter.

Lien GitHub : <https://github.com/JohanBrunet/Infotech>