

## Descriptif de l'entreprise

---

Folhomee est l'application qui accompagne les acheteurs tout au long de leur projet immobilier. Elle agrège les annonces immobilières parmi plus de 800 sites qui correspondent parfaitement à leurs besoins et elle conseille les acquéreurs jusqu'à l'emménagement (emprunts, travaux, architecte, décoration etc.). Folhomee c'est une application iOS & Android.

- <https://apps.apple.com/fr/app/folhomee/id1497318064>
- <https://play.google.com/store/apps/details?id=fr.folhomee.webapp>

Folhomee a aujourd'hui plus de 180,000 utilisateurs actifs dans les grandes métropoles françaises.

## Descriptif de la mission

---

Notre plateforme d'agrégation d'annonces est aujourd'hui la plus performante du marché. Elle permet aux acheteurs de trouver leur bien immobilier en fonction de différents critères (prix, superficie, nombre de pièces, travaux à envisager, ...), sur des centaines de sites immobiliers, leur faisant gagner un temps précieux.

Notre ambition au travers de votre appel à projet serait de trouver la meilleure manière de valoriser toutes les données des annonces immobilières préalablement agrégées et traitées dans notre base de données. Et cela afin d'améliorer l'expérience utilisateur en termes de pertinence des résultats proposés.

## Problématique

Les étudiants devront pour ce projet se concentrer sur chacune des catégories d'informations présentes dans les annonces immobilières. En effet, ces dernières contiennent de nombreuses précisions comme par exemple la localisation, le prix, la surface, les attributs ou encore la description textuelle.

À ce jour, ce sont des outils dédiés au traitement des données qui nous font défaut, sachant que nous agrégeons tous les moins entre 400 et 500 mille annonces.

## Tâches

Les principales tâches seront :

1. Étude préalable à partir de l'entièreté de la base de données des annonces pour expliciter les principales fonctionnalités à mettre en place.
2. Définition d'une architecture applicative pour l'intégration au sein de l'infrastructure de l'entreprise.
3. Développement des fonctionnalités attendues.

## Accompagnement

Lors de la première semaine, une introduction détaillée sera assurée par le CTO (directeur technique) afin d'expliquer la sémantique des données, et d'aider les étudiants à spécifier les principales fonctions à implémenter.

Par la suite, un accompagnement sera mis en place afin d'aider les étudiants à avancer sur le projet (réunions régulières avec les membres de l'équipe technique dédiés à ce projet, accès aux outils de communication de l'entreprise).

## Livrables attendus

Les livrables pourront intégrer les fonctionnalités suivantes :

- Mise en place de modèles d'apprentissages pour déterminer si, en fonction de différents critères dont le prix du marché, le bien en vente est dans la moyenne des prix :
  - Pour un investissement
  - Pour une résidence principale
  - Selon son positionnement géographique
  - Selon ses critères de choix
- Mise en place d'outils de classification d'annonces selon des critères déterminés (prix, type de biens, situation urbanistique) pour l'aide à la décision, en réfléchissant à un scoring d'annonces
- À la fin du processus d'inscription, possibilité d'indiquer combien d'annonces l'utilisateur va recevoir par jour/semaine
  - Donc déduire si la recherche semble trop large, correcte ou trop restreinte et faire des recommandations à l'utilisateur
- Mise en place de modèles d'apprentissage supervisés pour déterminer si une annonce fait partie d'un programme neuf à partir du texte de description
- Mise en place de modèles d'apprentissage supervisés pour déterminer l'état des travaux sur un bien à partir du texte de description
- Mise en place de modèles d'apprentissage supervisés pour l'extraction des données contextuelles des annonces (étage, nombre de pièces, nombre de chambres, ...)

## Technologies utilisées

Le choix des technologies à utiliser pour le projet est laissé libre. Les étudiants devront être à même de s'intégrer avec la stack existante :

- Langages de programmation : NodeJS et Golang
- Broker de Messagerie : RabbitMQ
- Base de données : MongoDB et PostgreSQL