Une image contenant texte, Police, Graphique, graphisme

Le contenu généré par l’IA peut être incorrect.

**Projet Python**

*3éme Année en Ingénierie Informatique et Réseaux*

**Conception et développement d’une app web pour la Gestion des annonces :**

**Echo-Anc**

***Réalisé par :***

*Fathi Salma*

*Saghir Hiba*

*Rahel Othman*

***Encadrer par:***

*Encadrant Professionnel :*

*Prof ABOUABID Hamza*

***Année universitaire : 2024/2025***

**Dédicaces**

Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce rapport.

À nos parents, pour leur soutien inconditionnel et leurs encouragements tout au long de notre parcours académique. Vous avez toujours cru en nous et nous avez appris l'importance de la persévérance.

À nos amis, pour leurs précieux conseils et leur présence réconfortante durant les moments de doute. Votre amitié a été une source de motivation.

Enfin, à nos enseignants, pour leur expertise et leur dévouement. Grâce à vous, nous avons pu acquérir les connaissances nécessaires pour mener à bien ce projet.

**Remerciements**

Nous tenons à remercier avant tout notre ALLAH, le très miséricordieux, qui grâce à lui. Nous souhaitons exprimer notre profonde gratitude à notre encadrant,**Prof Aboua**. Son soutien constant et ses conseils avisés ont été essentiels tout au long de notre projet.

Madame Ouatiq a su nous guider avec patience, en nous offrant des orientations précieuses et en nous encourageant à développer notre réflexion critique. Sa disponibilité et son écoute ont créé un environnement propice à l'apprentissage et à l'épanouissement de nos idées.

Nous lui sommes particulièrement reconnaissants pour avoir partagé son expertise, ce qui nous a permis d’aborder les défis avec confiance. Grâce à ses retours constructifs, nous avons pu enrichir notre travail et atteindre nos objectifs.

Enfin, sa passion pour le sujet et son engagement envers nos réussites ont été une source d'inspiration pour nous.

**Table des matières**

[Introduction Générale 1](#_Toc199159920)

[Chapitre 1 :](#_Toc199159921) [Présentation du cadre de projet 2](#_Toc199159922)

[Introduction 3](#_Toc199159923)

[Contexte académique 3](#_Toc199159924)

[Étude de l'existant 3](#_Toc199159925)

[Méthodologie de développement 3](#_Toc199159926)

[Planning de réalisation 4](#_Toc199159927)

[Conclusion 4](#_Toc199159929)

[Chapitre 2 :](#_Toc199159930) [Spécification des besoins 5](#_Toc199159931)

[Introduction 6](#_Toc199159932)

[Besoins fonctionnels 6](#_Toc199159933)

[2.1 Interface Utilisateur 6](#_Toc199159934)

[2.2 Interface Administrateur 7](#_Toc199159935)

[Besoins non fonctionnels 7](#_Toc199159936)

[Présentation des cas d'utilisation 8](#_Toc199159937)

[4.1. Acteurs principaux 8](#_Toc199159938)

[4.2. Description des cas d'utilisation 9](#_Toc199159939)

[4.3. Diagramme des cas d'utilisation 10](#_Toc199159944)

[Chapitre 3 :](#_Toc199159946) [Conception du système 12](#_Toc199159947)

[Introduction 13](#_Toc199159948)

[Modélisation Statique 13](#_Toc199159949)

[2.1. Diagramme de Classes 13](#_Toc199159950)

[Modèle relationnel 14](#_Toc199159951)

[Dictionnaire de Données 16](#_Toc199159953)

[Conclusion 18](#_Toc199159955)

[Chapitre 4 :](#_Toc199159956) [Réalisation du système 19](#_Toc199159957)

[Introduction 20](#_Toc199159958)

[Environnement de développement 20](#_Toc199159959)

[2.1. Environnement matériel 20](#_Toc199159960)

[2.2. Environnement logiciel 21](#_Toc199159963)

[Principales interfaces graphiques 23](#_Toc199159964)

[3.1. Interface Utilisateur 23](#_Toc199159965)

[3.2. Interface Administrateur 26](#_Toc199159966)

[Conclusion Générale 27](#_Toc199159967)

**Liste des Figures**

[Figure 1 : Diagramme des cas d’utilisation](#_Figure_1_:) ……………………………………………11

[Figure 2 : Diagramme des Classes](#_Figure_1_:) ………………………………………………..13

[Figure3 : Authentification utilisateur](#_Figure3_:_Authentification) ………………………………………………….23

[Figure4 : Home](#_Figure4_:_Home) ………………………………………………………………………..23

[Figure5 : Ajouter Annonce](#_Figure5_:_Ajouter) …………………………………………………………….24

[Figure6 : Mes Annonce](#_Figure6_:_Mes) ………………………………………………………………..24

[Figure7 : Messages](#_Figure7_:_Messages) …………………………………………………………………….25

[Figure8 : Connexion Admin](#_Figure8_:_Connexion) …………………………………………………………...26

[Figure9 : Tableau de bord](#_Figure9_:_Tableau) ……………………………………………………………...27

[Figure10 :Profil Admin](#_Figure10_:Profil_Admin) ………………………………………………………………...28

**Liste des Tableaux**

[Tableau 1 : Planning prévisionnel](#_Tableau_1_:)…………………………………………………..4

[Tableau 2 : Publication d'une annonce](#_Tableau_2_:) ……………………………………………..9

[Tableau 3 : Signalement d'une annonce](#_Tableau_3_:)…………………………………………….9

[Tableau 4 : Validation d'annonce (Admin)](#_Tableau_4_:)…………………………………………10

[Tableau 5 : Gestion des favoris](#_Tableau_5_:)…………………………………………………….10

[Tableau 6 : Modèle Relationnel Structuré](#_Tableau_6_:)………………………………………… 14

[Tableau 7 : Dictionnaire de données](#_Tableau_7_:)……………………………………………….16

# Introduction Générale

**Contexte et problématique**

Dans l'ère numérique actuelle, les plateformes d'annonces en ligne sont devenues incontournables pour les échanges entre particuliers et professionnels. Cependant, ces systèmes présentent souvent des failles importantes en matière de modération et de sécurité. Les utilisateurs sont fréquemment confrontés à des annonces frauduleuses, des contenus inappropriés et des interactions non sécurisées. Ces problèmes compromettent sérieusement la qualité des échanges et la fiabilité globale des plateformes existantes, créant un besoin urgent pour des solutions plus robustes et mieux contrôlées.

**Solution proposée : Echo-Anc**

Notre projet Echo-Anc a été conçu spécifiquement pour répondre à ces défis. Cette plateforme innovante intègre plusieurs fonctionnalités clés qui la distinguent des solutions existantes. Elle propose notamment un système de validation à double niveau combinant une analyse automatique préliminaire avec une modération manuelle par des administrateurs. La plateforme inclut également des outils de communication sécurisés, comme un chat intégré avec cryptage des messages, ainsi qu'un mécanisme intelligent de signalement des contenus problématiques. Pour les administrateurs, Echo-Anc offre un tableau de bord complet permettant une gestion centralisée des annonces et des utilisateurs.

**Approche méthodologique et innovations**

Le développement s'est structuré autour des forces du framework Django, exploitant pleinement son architecture MVT (Modèle-Vue-Template) et son système d'administration natif. Les innovations techniques majeures incluent l'intégration de Redis pour la gestion du cache, l'utilisation des capacités full-text de PostgreSQL, et l'implémentation de notifications temps réel via Django Channels. Cette approche a permis de concilier performance, sécurité et maintenabilité du code, tout en offrant une base solide pour les évolutions futures.

**Structure du rapport**

Ce rapport technique est organisé en quatre parties principales. La première section présente le cadre général du projet, incluant ses objectifs et sa planification. La deuxième partie détaille les spécifications techniques et l'analyse des besoins fonctionnels. Le troisième chapitre expose la conception détaillée du système à travers divers diagrammes UML. Enfin, la quatrième partie aborde les aspects concrets de la réalisation, présentant les interfaces développées, les défis techniques rencontrés et les résultats des tests de validation. Cette structure progressive permet de suivre l'évolution logique du projet depuis sa conception jusqu'à sa réalisation finale.

# Chapitre 1 :

# Présentation du cadre de projet

# Introduction

Ce chapitre a pour objectif de définir le cadre général de notre projet académique Echo-Annc. Il présentera le contexte de développement, l'analyse des solutions existantes, et la méthodologie adoptée par notre équipe de trois étudiants. Ce travail s'inscrit dans le cadre de notre formation en ingénierie informatique et vise à répondre à des problématiques concrètes des plateformes d'annonces en ligne.

# Contexte académique

Notre projet a été développé dans le cadre du programme de quatrième année d'ingénierie informatique, sans affiliation à une entreprise particulière. Ce contexte académique nous a offert une liberté dans le choix des technologies tout en maintenant une approche professionnelle dans la gestion du projet. L'équipe, composée de trois membres, a travaillé en étroite collaboration selon une répartition claire des tâches.

# Étude de l'existant

**1. Description des solutions actuelles**  
Les plateformes d'annonces existantes présentent souvent des architectures monolithiques peu flexibles. Leur gestion des annonces repose généralement sur des modérations a posteriori, ce qui permet la persistance temporaire de contenus inappropriés.

**2. Limites identifiées**  
Nos recherches ont révélé plusieurs lacunes :

* Absence de validation préalable des annonces
* Systèmes de signalement peu efficaces
* Interfaces administrateur limitées
* Manque d'outils statistiques pour les gestionnaires

**3. Solution proposée**  
Echo-Annc apporte des améliorations significatives :

* Workflow de validation à deux niveaux (automatique et manuelle)
* Système de signalement avec seuil d'action configurable
* Tableau de bord administrateur complet
* Statistiques détaillées sur l'utilisation

# Méthodologie de développement

Nous avons opté pour une approche Agile adaptée à notre contexte :

* Cycles de développement courts (2 semaines)
* Réunions d'équipe bihebdomadaires
* Utilisation de Git pour le versionnement collaboratif
* Revue de code mutuelle entre membres de l'équipe

Le choix de Django comme framework principal s'est imposé pour :

* Sa robustesse et sécurité intégrée
* Son architecture MVT claire
* Son système d'administration natif
* Sa compatibilité avec PostgreSQL

# Planning de réalisation

Le projet s'est déroulé sur 16 semaines selon le calendrier suivant :

# *Tableau 1 : Planning prévisionnel*

| **Phase** | **Semaines** | **Détails** |
| --- | --- | --- |
| Étude préalable | 1-4 | Analyse des besoins, benchmark |
| Conception | 5-8 | UML, modélisation base de données |
| Réalisation | 9-13 | Développement des fonctionnalités |
| Tests | 14-16 | Validation, corrections |

# Conclusion

Ce chapitre a posé les bases de notre projet académique en présentant le contexte de développement, les limites des solutions existantes et notre approche méthodologique. Les choix techniques et organisationnels décrits ici ont guidé l'ensemble de notre travail, comme le détailleront les chapitres

# Chapitre 2 :

# Spécification des besoins

Introduction  
Ce chapitre présente l'analyse détaillée des besoins fonctionnels et non fonctionnels de la plateforme Echo-Annc, basée sur le diagramme de cas d'utilisation fourni. Notre approche s'appuie sur une modélisation UML rigoureuse pour garantir une couverture complète des exigences.

# Besoins fonctionnels

# 2.1 Interface Utilisateur

**2.1.1 Gestion du compte**

* Inscription avec validation par email
* Connexion/déconnexion sécurisée
* Réinitialisation du mot de passe (envoi de code par email)
* Modification du profil (photo, informations personnelles)

**2.1.2 Gestion des annonces**

* Consultation des annonces :
  + Recherche par titre/nom du publieur
  + Filtrage par catégories/sous-catégories
  + Affichage des détails complets
* Publication d'annonces :
  + Soumission avec statut "en attente"
  + Notification email de validation/rejet
  + Gestion des annonces (modification/suppression)

**2.1.3 Interactions**

* Système de favoris (ajout/consultation)
* Messagerie privée entre utilisateurs
* Signalement d'annonces (avec seuil à 2 signalements)
* Commentaires publics sur les annonces

# 2.2 Interface Administrateur

**2.2.1 Modération**

* Validation/rejet des annonces :
  + Notification email automatisée
  + Justification des rejets
* Gestion des signalements :
  + Suppression après 2 signalements
  + Notification au créateur

**2.2.2 Statistiques**

* Tableaux de bord :
  + Distribution par catégorie/prix/localisation
  + Taux de validation/rejet
* Export des données

# Besoins non fonctionnels

**3.1 Sécurité**

* Chiffrement des mots de passe (PBKDF2)
* Protection CSRF/XSS
* Validation stricte des entrées
* Gestion des permissions (modèles Django)

**3.2 Performance**

* Temps de réponse < 1.5s
* Pagination des résultats
* Cache des requêtes fréquentes

**3.3 Expérience utilisateur**

* Interface responsive
* Upload multiple de médias
* Notifications en temps réel

# Présentation des cas d'utilisation

# 4.1. Acteurs principaux

**Utilisateur**

L'utilisateur est l'acteur principal de la plateforme Echo-Annc. Il peut naviguer sur le site pour consulter les annonces publiées, effectuer des recherches par titre ou nom du publieur, et filtrer les résultats par catégories. Pour accéder aux fonctionnalités complètes (publier une annonce, envoyer des messages, etc.), il doit créer un compte ou se connecter. Le système offre une procédure de récupération de mot de passe par email. Une fois authentifié, l'utilisateur peut gérer ses annonces (publication, modification, suppression), interagir avec d'autres membres via un système de messagerie, ajouter des annonces à ses favoris, signaler du contenu inapproprié, et consulter/modifier son profil personnel. Ses annonces publiées sont soumises à validation par l'administrateur avant d'être visibles publiquement.

**Administrateur**

L'administrateur est le modérateur de la plateforme Echo-Annc. Après authentification via une interface dédiée, il a accès à un tableau de bord complet lui permettant de gérer l'ensemble du contenu. Ses principales responsabilités incluent la validation ou le rejet des annonces soumises par les utilisateurs (avec envoi d'une notification email en cas de rejet), la gestion des signalements (suppression automatique après 2 signalements avec notification au créateur), et la surveillance des comptes utilisateurs. L'administrateur dispose également d'outils d'analyse lui permettant de consulter des statistiques détaillées sur les annonces (par catégorie, prix, localisation) et l'activité globale de la plateforme. Il peut modifier ses propres informations de profil et se déconnecter du système.

**Système**

Le système représente l'infrastructure technique automatisée de Echo-Annc. Il orchestre l'ensemble des processus et interactions entre les différents acteurs. Ses fonctions principales incluent la gestion des workflows (validation des annonces, comptage des signalements), l'envoi automatique de notifications (emails de confirmation, codes de réinitialisation, alertes de modération), le stockage et la sécurisation des données, ainsi que la génération de statistiques en temps réel. Le système applique également les règles métier (comme le masquage automatique des annonces ayant reçu 2 signalements) et garantit la protection contre les menaces de sécurité (attaques CSRF, XSS). Il sert d'intermédiaire transparent entre les utilisateurs et les administrateurs, assurant le bon fonctionnement global de la plateforme.

# 4.2. Description des cas d'utilisation

# *Tableau 2 : Publication d'une annonce*

|  |  |
| --- | --- |
| **Élément** | **Description** |
| **Acteur(s)** | Utilisateur, Système |
| **Objectif** | Publier une nouvelle annonce sur la plateforme |
| **Préconditions** | - Utilisateur authentifié - Compte actif |
| **Scénario nominal** | 1. Accéder au formulaire de publication 2. Remplir les champs obligatoires 3. Uploader des photos 4. Soumettre l'annonce 5. Notification de soumission réussie |
| **Scénarios alternatifs** | - Annonce incomplète : message d'erreur - Photos invalides : suggestion de formats acceptés |
| **Post-conditions** | Annonce en statut "En attente" |

# *Tableau 3 : Signalement d'une annonce*

|  |  |
| --- | --- |
| **Élément** | **Description** |
| **Acteur(s)** | Utilisateur, Système, Admin |
| **Objectif** | Signaler un contenu inapproprié |
| **Préconditions** | - Utilisateur authentifié - Annonce visible |
| **Scénario nominal** | 1. Sélectionner "Signaler" 2. Choisir un motif 3. Confirmer le signalement 4. Notification de prise en compte |
| **Règles métier** | - Seuil de 2 signalements pour masquage auto - Notification immédiate à l'admin |
| **Post-conditions** | Signalement enregistré dans l'historique |

# *Tableau 4 : Validation d'annonce (Admin)*

|  |  |
| --- | --- |
| **Élément** | **Description** |
| **Acteur(s)** | Administrateur, Système |
| **Objectif** | Modérer le contenu publié |
| **Préconditions** | - Admin authentifié - Annonces en attente |
| **Scénario nominal** | 1. Consulter la file d'attente 2. Examiner les détails 3. Approuver/Rejeter 4. Saisir motif si rejet |
| **Actions système** | - Email automatique en cas de rejet - Mise à jour immédiate du statut |
| **Post-conditions** | Base de données mise à jour |

# *Tableau 5 : Gestion des favoris*

|  |  |
| --- | --- |
| **Élément** | **Description** |
| **Acteur(s)** | Utilisateur, Système |
| **Objectif** | Sauvegarder des annonces intéressantes |
| **Préconditions** | Utilisateur connecté |
| **Scénario nominal** | 1. Cliquer sur "♡" 2. Vérification en "Mes Favoris" 3. Option de suppression |
| **Actions système** | Pas de limite de favoris |
| **Post-conditions** | Liste personnelle mise à jour |

# 4.3. Diagramme des cas d'utilisation

Ce diagramme décrit les interactions des utilisateurs (publication, recherche, messagerie) et administrateurs (modération, gestion) avec la plateforme d'annonces, incluant les relations entre fonctionnalités et règles métier comme le seuil de 2 signalements pour suppression.



# ***Figure 1 : Diagramme des*** ***cas*** ***d’utilisation***

# Chapitre 3 :

# Conception du système

# Introduction

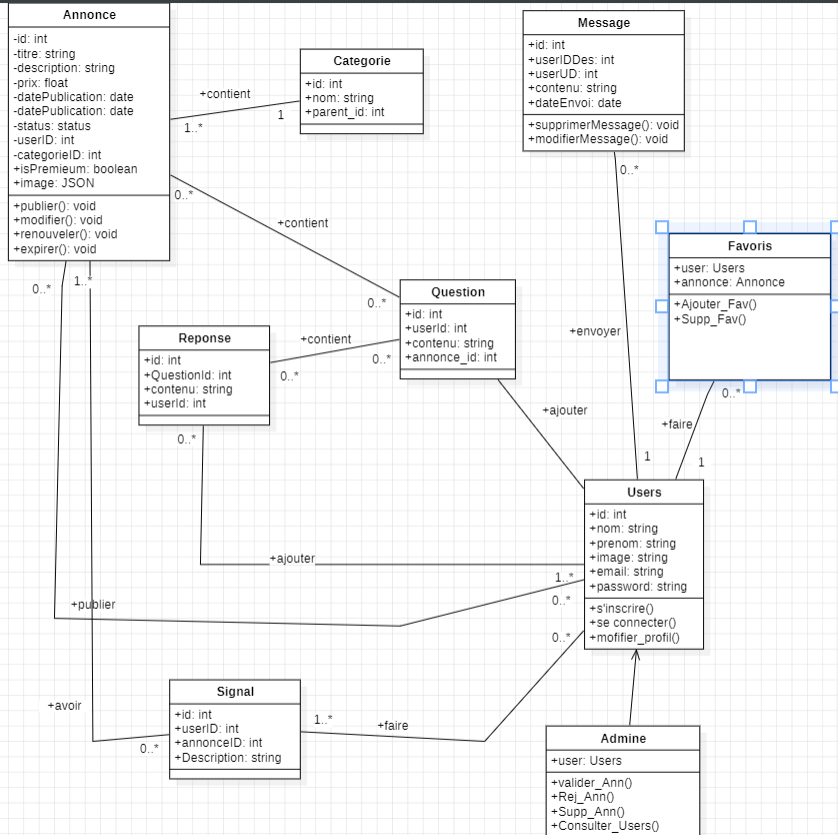
Ce chapitre présente la modélisation complète du système **Echo-Annc** à travers une approche

* **Statique** : Structure des données (diagramme de classes)

# Modélisation Statique

# 2.1. Diagramme de Classes

Le diagramme de classes présente les entités principales, leurs attributs, leurs méthodes, et leurs relations.

*Figure 2 : Diagramme des Classes*

# Modèle relationnel

Ce tableau présente le modèle relationnel d'une base de données pour un système de gestion d’annonces. Il résume les tables principales, leurs attributs, leurs relations et leurs clés (primaires et étrangères).

# *Tableau 6 : Modèle Relationnel Structuré*

|  |  |  |  |
| --- | --- | --- | --- |
| Table | Colonnes | Type&Contraintes | Clés |
| User | id, nom, prenom, email,  password, image, is\_active, date\_joined,  last\_login | SERIAL, VARCHAR(30), VARCHAR(255) UNIQUE, BOOLEAN DEFAULT TRUE, TIMESTAMP | PRIMARY KEY (id) |
| Admin | id, user\_id, role | SERIAL, INT REFERENCES users(id), VARCHAR(20) DEFAULT 'moderator | PRIMARY KEY (id) |
| Categories | id, nom, parent\_id | SERIAL, VARCHAR(30),  INT REFERENCES categories(id) | PRIMARY KEY (id) |
| Annonces | id, titre, description, prix, date\_publication, statut, localisation, is\_premium, images, user\_id,  categorie\_id | SERIAL, TEXT,  JSONB,  INT,  DECIMAL(10,2) CHECK (prix >= 0), | PRIMARY KEY (id), FOREIGN KEY (user\_id) REFERENCES users(id), FOREIGN KEY (categorie\_id) REFERENCES categories(id) |
| Signals | id, description, date\_signalement, user\_id,  annonce\_id | SERIAL, TEXT, INT  TIMESTAMP DEFAULT NOW(), | PRIMARY KEY (id), FOREIGN KEY (user\_id) REFERENCES users(id), FOREIGN KEY (annonce\_id) REFERENCES annonces(id) |
| Messages | id, contenu, date\_envoi, user\_source\_id,  user\_dest\_id | SERIAL, TEXT, INT TIMESTAMP, | PRIMARY KEY (id), FOREIGN KEY (user\_source\_id) REFERENCES users(id), FOREIGN KEY (user\_dest\_id) REFERENCES users(id) |
| Question | id, contenu, date\_publication, user\_id, annonce\_id | SERIAL, TEXT,  INT  TIMESTAMP, | PRIMARY KEY (id), FOREIGN KEY (user\_id) REFERENCES users(id), FOREIGN KEY (annonce\_id) REFERENCES annonces(id) |
| Réponses | id, contenu, date\_publication, user\_id, question\_id | SERIAL, TEXT,  INT  TIMESTAMP, | PRIMARY KEY (id), FOREIGN KEY (user\_id) REFERENCES users(id), FOREIGN KEY (question\_id) REFERENCES questions(id) |
| Favoris | id, user\_id, annonce\_id | SERIAL, INT, INT | PRIMARY KEY (id), FOREIGN KEY (user\_id) REFERENCES users(id), FOREIGN KEY (annonce\_id) REFERENCES annonces(id) |

# Dictionnaire de Données

Chaque colonne issue du modèle relationnel est décrite dans un tableau comme suit :

# *Tableau 7 : Dictionnaire de données*

| **Nom de la colonne** | **Type de données** | **Obligatoire (Oui/Non)** | **Valeur par défaut** | **Clé Primaire** | **Clé Etrangère** | **Nom de la Table** |
| --- | --- | --- | --- | --- | --- | --- |
| **id** | SERIAL | Oui | Auto-incrément | ✔ | - | users |
| **nom** | VARCHAR | Oui | - | - | - | users |
| **prenom** | VARCHAR | Oui | - | - | - | users |
| **email** | VARCHAR | Oui | - | - | - | users |
| **password** | VARCHAR | Oui | - | - | - | users |
| **image** | VARCHAR | Non | NULL | - | - | users |
| **is\_active** | BOOLEAN | Oui | TRUE | - | - | users |
| **date\_joined** | TIMESTAMP | Oui | CURRENT\_TIMESTAMP | - | - | users |
| **last\_login** | TIMESTAMP | Non | NULL | - | - | users |
| **id** | SERIAL | Oui | Auto-incrément | ✔ | - | annonces |
| **titre** | VARCHAR | Oui | - | - | - | annonces |
| **description** | TEXT | Oui | - | - | - | annonces |
| **prix** | DECIMAL | Oui | - | - | - | annonces |
| **date\_publication** | TIMESTAMP | Oui | CURRENT\_TIMESTAMP | - | - | annonces |
| **statut** | VARCHAR | Oui | 'attente' | - | - | annonces |
| **user\_id** | INTEGER | Oui | - | - | users | annonces |
| **categorie\_id** | INTEGER | Non | NULL | - | categories | annonces |
| **id** | SERIAL | Oui | Auto-incrément | ✔ | - | categories |
| **nom** | VARCHAR | Oui | - | - | - | categories |
| **parent\_id** | INTEGER | Non | NULL | - | categories | categories |
| **id** | SERIAL | Oui | Auto-incrément | ✔ | - | signals |
| **description** | TEXT | Oui | - | - | - | signals |
| **user\_id** | INTEGER | Oui | - | - | users | signals |
| **annonce\_id** | INTEGER | Oui | - | - | annonces | signals |

# Conclusion

Ce chapitre a présenté la solution conceptuelle du système en combinant modélisations dynamique et statique. La modélisation dynamique a permis d’illustrer les interactions et les flux de contrôle. La modélisation statique a montré la structure des données et l’architecture de l’application, assurant ainsi une conception robuste et évolutive.

# Chapitre 4 :

# Réalisation du système

# Introduction

Ce chapitre présente l'implémentation concrète de notre application **Echo-Anc** . Nous détaillerons l'environnement technique utilisé pour le développement ainsi que les principales interfaces utilisateur et administrateur. Cette réalisation s'appuie sur les spécifications et conceptions présentées dans les chapitres précédents.

# Environnement de développement

# 2.1. Environnement matériel

# *Tableau 7 : PC1*

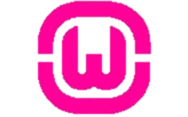
| **Composant** | **Spécifications** |
| --- | --- |
| Poste de développement | Dell Latitude 3410 |
| RAM | 8192 MB |
| Stockage | SSD 512 Go |
| Réseau | Connexion fibre 100 Mbps |
| Système d'exploitation | Windows 11 Professionnel 64 bits |

# *Tableau 8 : PC2/PC3*

| **Composant** | **Spécifications** |
| --- | --- |
| Poste de développement | HP EliteBook 820 G3 |
| RAM | 8192 MB |
| Stockage | SSD 512 Go |
| Réseau | Connexion fibre 100 Mbps |
| Système d'exploitation | Windows 1 Professionnel 64 bits |

# 2.2. Environnement logiciel

**1. WampServer 3.2.6**

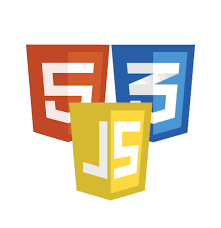
WampServer est un environnement de développement web sous Windows qui permet de créer des applications web dynamiques. Il contient Apache (serveur web), MySQL (système de gestion de base de données) et PHP (langage de programmation). La version 3.2.6 inclut des mises à jour et des améliorations de sécurité.

Une image contenant clipart, symbole, conception

Le contenu généré par l’IA peut être incorrect.**2. Backend : Python + Django**

* **Python** : Langage de programmation polyvalent, connu pour sa lisibilité et sa simplicité. Il est largement utilisé dans le développement web, l'analyse de données et l’intelligence artificielle.
* **Django** : Framework web open-source pour Python, qui facilite le développement rapide d'applications web en fournissant des outils et des fonctionnalités prêtes à l'emploi, comme l'authentification, la gestion des bases de données et le routage des URL.

**3. Frontend : JavaScript, HTML, CSS, Bootstrap 5**

* **JavaScript** : Langage de programmation utilisé principalement pour créer des pages web interactives et dynamiques. Il s'exécute côté client dans le navigateur.
* **HTML (HyperText Markup Language)** : Langage de balisage utilisé pour structurer le contenu des pages web.
* **CSS (Cascading Style Sheets)** : Langage de style utilisé pour décrire la présentation et le design des documents HTML.
* **Bootstrap 5** : Framework CSS populaire qui facilite la création de sites web réactifs et attrayants. Il offre des composants et des outils pour le design.

**4. Base de données : MySQL via WampServer**

MySQL est un système de gestion de base de données relationnelle open-source. Il permet de stocker, gérer et récupérer des données. Dans ce contexte, il est utilisé via WampServer pour gérer les données de l'application.

**5. IDE : Visual Studio Code avec extensions Python/Django**

Visual Studio Code (VS Code) est un éditeur de code source léger et puissant, qui supporte de nombreux langages de programmation. Les extensions Python et Django ajoutent des fonctionnalités spécifiques pour faciliter le développement d'applications en Python et Django, comme l'autocomplétion, le débogage, et la navigation dans le code.

**6. Configuration WampServer**

* **Virtual Host configuré pour le projet** : Permet de créer plusieurs environnements de développement locaux sur une seule instance de WampServer, en attribuant des noms de domaine personnalisés.
* **Accès phpMyAdmin** : Interface web permettant de gérer facilement les bases de données MySQL, de créer des tables, d'exécuter des requêtes SQL et d'effectuer des sauvegardes.
* **Modules Apache activés** :
  + **Rewrite** : Permet de réécrire les URL pour les rendre plus conviviales.
  + **Headers** : Gère les en-têtes HTTP pour le contrôle de la mise en cache et la sécurité.
  + **ssl** : Permet de sécuriser les connexions via HTTPS.

Ces composants forment un environnement de développement complet pour créer des applications web modernes et robustes.

# Principales interfaces graphiques

# 3.1. Interface Utilisateur

Une Interface de connexion simple et épurée pour le service "Echo-Anc", avec des champs pour l'email et le mot de passe, une option "Se souvenir de moi", et un lien pour récupérer un mot de passe oublié.

Une image contenant texte, capture d’écran, Police, conception

Le contenu généré par l’IA peut être incorrect.

# *Figure3 : Authentification utilisateur*

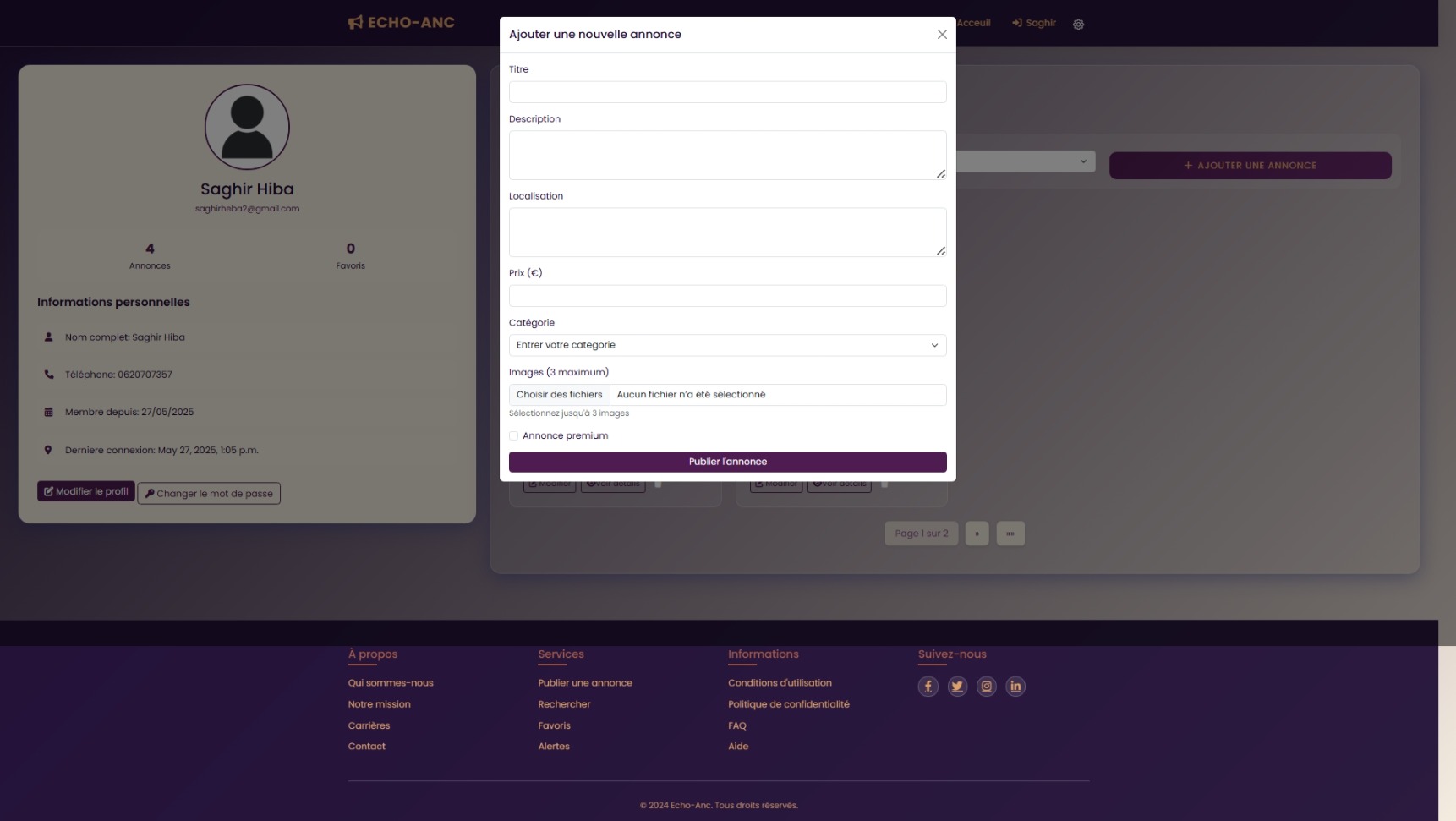
Une plateforme de petites annonces, avec des catégories (Immobilier, Mode, etc.), un tableau de bord utilisateur, des annonces récentes et un pied de page avec des liens pratiques.

Une image contenant texte, capture d’écran, logiciel, Site web

Le contenu généré par l’IA peut être incorrect.

# *Figure4 : Home*

L’image montre une interface d’ajout d’annonce sur la plateforme **Echo-ANC**.



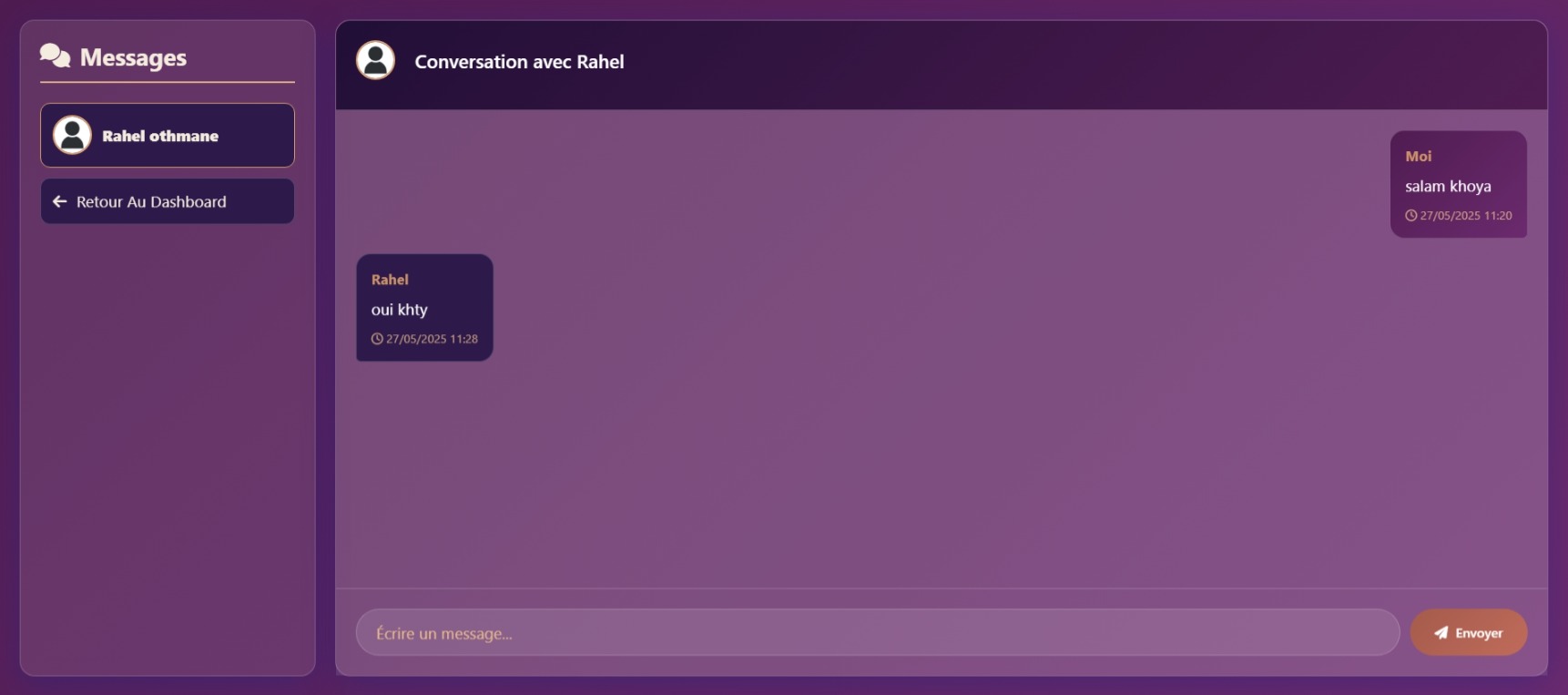
# *Figure5 : Ajouter Annonce*

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.La capture d'écran montre l'interface "Mes Annonces" d'Echo-Annc, avec des options de gestion (modifier/supprimer), des infos légales et des services comme la publication d'annonces et les favoris.

# *Figure****6****:* ***Mes Annonce***

L'interface affiche une conversation entre l'utilisateur et "Rahel", avec des messages. Un champ "Ecrire un message..." permet de répondre, avec un lien de retour au tableau de bord.



# *Figure7 : Messages*

# 

# 3.2. Interface Administrateur

La capture montre l'interface de connexion admin d'Echo-Anc avec champs pour email/mot de passe et options de connexion via réseaux sociaux

Une image contenant texte, capture d’écran, Police, conception

Le contenu généré par l’IA peut être incorrect.

# *Figure8 : Connexion Admin*

La capture montre le Tableau de bord de l’interface Admin qui contient le nombre des annonces en attentes/validée et Utilisateur, ainsi que l’affichage des demandes en attentes et une partie de contrôle Administrateur

Une image contenant texte, capture d’écran, Site web, logiciel

Le contenu généré par l’IA peut être incorrect.

# *Figure9 : Tableau de bord*

Cette interface représente les information de l’admin

Une image contenant texte, capture d’écran, logiciel, multimédia

Le contenu généré par l’IA peut être incorrect.

# *Figure10 :Profil Admin*

# Conclusion Générale

Ce projet académique de conception et développement de la plateforme Echo-Annc a permis de mettre en œuvre une solution complète de gestion d'annonces en ligne, répondant aux objectifs pédagogiques de notre formation en ingénierie informatique. À travers une démarche rigoureuse, nous avons su transformer les exigences fonctionnelles initiales en un système opérationnel comprenant une interface utilisateur complète (publication modérée, interactions sociales, gestion de profil) et un back-office administrateur performant (modération, statistiques). Les résultats techniques obtenus, notamment la modélisation précise des données via UML, l'architecture MVC bien structurée et les mécanismes de sécurité implémentés, démontrent l'acquisition des compétences clés en génie logiciel.

Le travail réalisé a pleinement répondu aux problématiques identifiées en proposant des solutions concrètes : système de validation à double niveau pour les annonces, gestion des signalements avec seuil automatique, et interactions utilisateurs sécurisées. Cependant, une analyse critique révèle que certaines limitations pourraient être surmontées dans une version future, comme l'optimisation des performances pour un grand volume de données ou l'enrichissement des fonctionnalités sociales. Ce projet a non seulement validé notre maîtrise des concepts théoriques, mais a également renforcé nos capacités à travailler en équipe sur un développement complexe. Les compétences acquises en analyse, conception et développement full-stack constituent un socle solide pour notre future carrière d'ingénieur en informatique, tout en démontrant notre capacité à mener à bien un projet ambitieux depuis sa conception jusqu'à sa réalisation concrète.

# Bibliographie

1. W3Schools (https://www.w3schools.com/)

W3Schools est une plateforme d'apprentissage en ligne très populaire pour les développeurs web. Elle propose des tutoriels, des exemples et des exercices interactifs sur des technologies comme HTML, CSS, JavaScript, Python, SQL et bien d'autres. Son approche simple et pratique en fait une excellente ressource pour les débutants comme pour les développeurs expérimentés.

2. Documentation de Django (https://docs.djangoproject.com/en/5.2/)

La documentation officielle de Django est une ressource essentielle pour tout développeur travaillant avec ce framework Python. Elle couvre en détail toutes les fonctionnalités, des bases aux concepts avancés, avec des guides, des références d'API et des exemples de code. La version 5.2 inclut les dernières améliorations et bonnes pratiques pour créer des applications web robustes.

3. Bootstrap (https://getbootstrap.com/)

Bootstrap est l'un des frameworks CSS les plus populaires pour concevoir des sites web responsives et modernes. Il offre des composants pré-stylisés, des grilles flexibles et des utilitaires qui simplifient le développement front-end. Bootstrap est largement utilisé pour créer des interfaces utilisateur attrayantes et compatibles avec tous les appareils.

4. Deuxième documentation Django PDF(version5.1.x)

https://django.readthedocs.io/\_/downloads/en/5.1.x/pdf/)

Cette version PDF de la documentation Django (5.1.x) permet aux développeurs de consulter le contenu hors ligne ou de l'imprimer pour une référence facile. Bien que légèrement plus ancienne que la version 5.2, elle reste utile pour comprendre les concepts de base et les fonctionnalités de Django

5. Site officiel de Python (https://www.python.org/)

Python.org est le site officiel du langage de programmation Python. Il propose des téléchargements pour les différentes versions de Python, une documentation complète, des tutoriels, des actualités sur les mises à jour, ainsi qu'une communauté active. Que vous soyez débutant ou développeur expérimenté, ce site est une ressource incontournable pour apprendre, coder et rester informé sur l'écosystème Python.

6. Notre Code Source dans GitHub ([salma518/projet-django-annonce](https://github.com/salma518/projet-django-annonce))