



Rapport de Projet

3ème année

Ingénierie Informatique et Réseau

Application web de gestion de votes

Réalisé par :

Sara Chetouane

Hafsa Bouzidi

Encadré par :

khalid nafil

REMERCIEMENTS

Nous souhaitons exprimer, à travers ces lignes, notre sincère gratitude à l'ensemble du corps professoral de l'École Marocaine des Sciences de l'Ingénieur (E.M.S.I) pour les enseignements de qualité et les solides bases techniques qu'il nous a transmis tout au long de notre formation. Ces acquis ont grandement enrichi notre capacité d'analyse et nous ont permis d'aborder notre projet avec rigueur et méthodologie.

Nos remerciements les plus chaleureux vont particulièrement à **Monsieur khalid nafil**, notre encadrant, pour sa disponibilité, ses conseils avisés et son accompagnement constant tout au long de ce travail. Son engagement et sa bienveillance ont été des piliers essentiels dans la réalisation de ce projet.

Enfin, nous adressons nos remerciements à toutes les personnes, de près ou de loin, ayant contribué à la concrétisation de ce projet. Votre soutien et vos encouragements ont été précieux, et nous vous en sommes profondément reconnaissants.

Résumé

Dans le cadre de notre formation en ingénierie informatique et réseaux, nous réalisons un projet portant sur la conception et le développement d'une application web de gestion de votes. Ce projet a pour objectif de mettre en pratique nos compétences en développement web avec le Framework Django, ainsi qu'en bases de données et en sécurité des applications.

L'application permettra aux utilisateurs de participer à des votes en ligne de manière simple et sécurisée, aux administrateurs de créer et gérer les scrutins, et aux responsables d'accéder à des statistiques détaillées via des tableaux de bord dynamiques. Elle vise à garantir la transparence des résultats, la traçabilité des votes et une expérience utilisateur fluide et intuitive.

Ce projet représente une mise en œuvre concrète des connaissances acquises au cours de notre formation, tout en répondant à un besoin croissant d'outils numériques fiables pour la prise de décision collective dans les entreprises, les institutions et les communautés.

Introduction Générale

Dans le cadre de notre formation en ingénierie informatique et réseaux, nous sommes amenés à concevoir et réaliser un projet permettant de mettre en pratique les compétences acquises tout au long de notre parcours. Ce projet constitue une opportunité concrète de mobiliser nos connaissances en développement web, en gestion de bases de données, en sécurité informatique et en conception d'interfaces utilisateur.

Le projet que nous avons choisi porte sur la conception et le développement d'une application web de gestion de votes, s'appuyant sur le Framework Django. À l'ère de la transformation numérique, la mise en place de solutions dématérialisées pour la prise de décision collective est devenue essentielle. Les systèmes de vote en ligne permettent de faciliter la participation, de renforcer la transparence des résultats, et de simplifier la gestion des scrutins au sein des organisations.

L'objectif principal de cette application est de permettre aux administrateurs de créer et gérer différents types de votes, aux utilisateurs d'y participer de manière sécurisée, et aux responsables d'analyser les résultats à travers des tableaux de bord clairs et interactifs. Ce projet répond ainsi à un besoin réel tout en nous offrant une expérience professionnelle enrichissante, alliant conception technique, réflexion ergonomique et respect des bonnes pratiques du développement logiciel.

Table da matière :

REMERCIEMENTS	2
Résumé.....	3
Introduction Générale	4
Chapitre 1.....	7
Cadre du Projet.....	7
1. Introduction.....	8
2. Présentation projet	8
3. Conclusion.....	8
Chapitre 2.....	9
Analyse des besoins.....	9
1. Objectifs du projet.....	10
2. Types d'utilisateurs.....	10
2. Définitions des données	11
3. Les besoins fonctionnels.....	12
Chapitre 3.....	14
2. Diagramme de cas d'utilisation :.....	18
3. Diagramme de séquence:	20
Chapitre 4.....	21
1. Backend : Django (Python)	22
2. Frontend : HTML, CSS et JavaScript.....	22
3. Moteur de templates : Django Templates.....	23
4. Base de données : SQLite (ou autre via ORM Django)	23
5. Sécurité.....	23
1.Interface d'administration.....	25
2.Page d'inscription.....	25
3.Page de connexion	26
4.Interface utilisateur après connexion.....	27
5.Message de vote déjà effectué	28
6.Résultats d'un sondage	28
7.Gestion des options dans l'admin Django	29
8.Ajout manuel d'un vote (Admin Django)	29
9.Ajout d'un utilisateur (Admin Django)	30

10.Création d'un sondage (Admin Django)	30
11.Interface d'administration Django - Gestion des sondages	31
Conclusion.....	32

Chapitre 1

Cadre du Projet

1. Introduction

Dans un contexte où la prise de décision collective devient de plus en plus fréquente dans les organisations, la mise en place de systèmes numériques fiables et sécurisés est essentielle. Ce projet de fin d'année a pour objectif de concevoir et développer une application web de gestion de votes, permettant d'organiser des scrutins en ligne, d'en garantir la transparence, et de centraliser les résultats de manière structurée. Cette initiative s'inscrit dans une volonté de moderniser les processus décisionnels tout en offrant une expérience utilisateur simple, sécurisée et accessible.

2. Présentation projet

Le projet consiste à développer une plateforme web de gestion de votes en ligne, destinée à faciliter la création, la gestion et la participation à des scrutins électroniques. Grâce à un espace administrateur, il sera possible de configurer différents types de votes, de gérer les utilisateurs et de suivre les résultats en temps réel via des tableaux de bord. L'application vise à offrir une solution ergonomique, sécurisée et adaptée aux besoins d'organisations souhaitant numériser leurs processus de vote, tout en assurant la traçabilité et la confidentialité des données.

3. Conclusion

Ce projet constitue une opportunité concrète de mobiliser nos compétences en développement web, en sécurité informatique et en gestion de base de données. Il répond à une problématique réelle rencontrée dans le monde professionnel, en proposant une solution moderne, évolutive et sécurisée. En plus d'être un exercice technique complet, ce projet s'inscrit dans une dynamique de professionnalisation et de réponse aux besoins actuels des organisations en matière de gouvernance numérique.

Chapitre 2

Analyse des besoins

1. Objectifs du projet

L'objectif principal de ce projet est de développer une plateforme web permettant d'organiser des votes en ligne de manière simple, sécurisée et transparente. Cette application doit répondre aux attentes des utilisateurs en termes d'ergonomie, de rapidité, de fiabilité et de confidentialité des données..

2. Types d'utilisateurs

L'application distinguera trois types principaux d'utilisateurs, chacun ayant des droits spécifiques :

Administrateur :

- Créer, modifier ou supprimer des scrutins
- Gérer les listes de votants
- Consulter les résultats

Votant (utilisateur final) :

- Accéder aux votes disponibles
- Voter de manière sécurisée (authentification requise)
- Consulter les résultats si autorisé

2. Définitions des données

L'application de gestion de votes repose sur plusieurs entités principales. Chacune joue un rôle clé dans le fonctionnement du système. Voici une description détaillée des données manipulées.

1. Utilisateur

L'entité *Utilisateur* représente toute personne accédant au système. Deux types d'utilisateurs sont définis : les administrateurs, qui ont des droits de gestion, et les votants, qui participent aux scrutins.

Chaque utilisateur possède un identifiant unique, un nom, une adresse email, et un mot de passe. Le mot de passe est stocké de manière sécurisée à l'aide d'un algorithme de hachage. Le champ *rôle* permet de distinguer un administrateur d'un votant.

4. Vote

L'entité *Vote* enregistre l'action d'un votant qui sélectionne un candidat dans un scrutin donné. Un utilisateur ne peut voter qu'une seule fois par scrutin, afin de garantir l'unicité du vote.

Le vote est associé à l'utilisateur, au scrutin et au candidat choisis. Il est également horodaté pour assurer la traçabilité de l'action.

5. Résultat

Le *Résultat* représente le calcul du nombre de votes reçus par chaque candidat dans un scrutin. Cette entité permet de produire les statistiques ou le classement des choix en fonction des votes enregistrés.

Elle peut être automatiquement mise à jour dès qu'un nouveau vote est ajouté, ou recalculée à la fin du scrutin par le système.

6. Authentification

Même si elle ne constitue pas une entité persistante, l'*Authentification* joue un rôle fondamental dans la sécurité du système. Elle intervient chaque fois qu'un

utilisateur tente de se connecter, pour vérifier ses identifiants et déterminer son rôle. Ce processus conditionne l'accès aux fonctionnalités selon le profil de l'utilisateur.

3. Les besoins fonctionnels

Les besoins fonctionnels définissent l'ensemble des fonctionnalités que le système doit offrir aux différents types d'utilisateurs. L'application de gestion de votes distingue principalement deux types d'utilisateurs : **l'administrateur** et **le votant**. Chacun dispose d'un ensemble spécifique de fonctionnalités.

1. Authentification

- Le système doit permettre aux utilisateurs de se connecter à l'aide de leurs identifiants (email et mot de passe).
- Une authentification sécurisée est exigée pour accéder aux fonctionnalités personnalisées.
- Les rôles des utilisateurs doivent être identifiés après connexion (administrateur ou votant).

2. Gestion des votants (administrateur)

- L'administrateur doit pouvoir **ajouter** une liste de votants autorisés à participer à un scrutin.
- Il doit aussi pouvoir **mettre à jour ou supprimer** des votants si nécessaire.
- Il est possible d'importer la liste des votants à partir d'un fichier (CSV ou Excel).

3. Vote (votants)

- Le votant peut **sélectionner un scrutin actif**, consulter les candidats, puis effectuer son vote.
- Le système doit s'assurer que chaque utilisateur ne vote **qu'une seule fois par scrutin**.

- Une fois le vote soumis, il est enregistré de manière sécurisée et ne peut plus être modifié.

4. Affichage des résultats

- L'administrateur peut consulter **les résultats détaillés** de chaque scrutin (nombre de votes par candidat, taux de participation, etc.).
- Si le scrutin est public ou que l'option est activée, les résultats peuvent également être **consultés par les votants**.

Chapitre 3

Analyse et Conception

1. Introduction

Après avoir défini de manière détaillée les besoins fonctionnels et non fonctionnels de l'application de gestion de votes, cette phase de conception a pour objectif de **traduire ces exigences en une architecture logicielle claire, cohérente et bien structurée.**

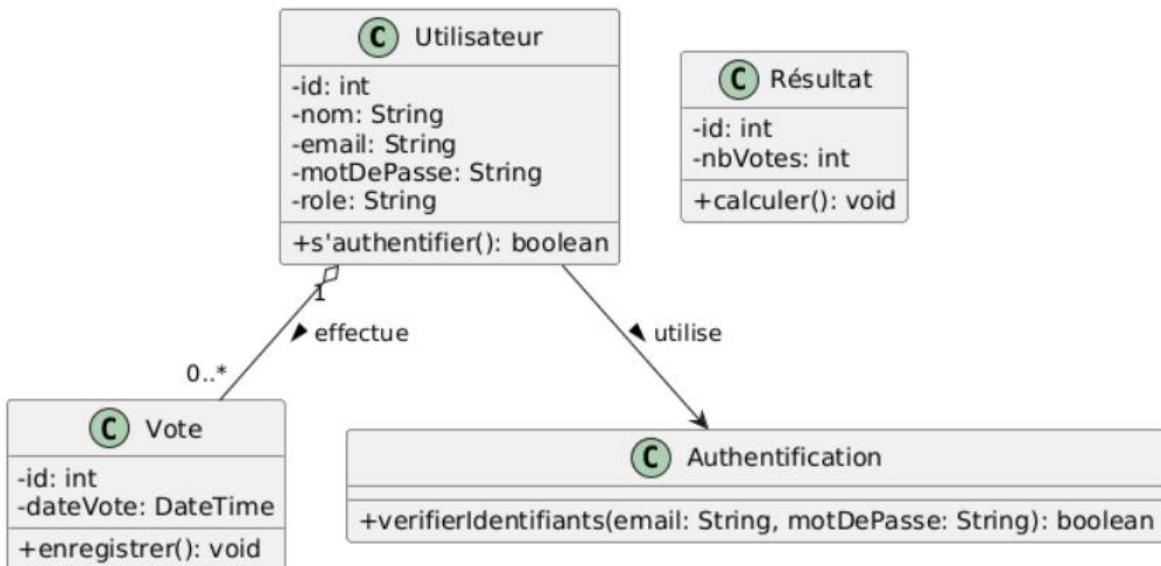
La conception s'appuie sur les **principes de l'analyse orientée objet**, qui permettent de modéliser le système à travers des entités, leurs relations et les interactions entre les différents composants. L'utilisation des **diagrammes UML (Unified Modeling Language)** facilite la compréhension globale du fonctionnement du système, tout en fournissant une base solide pour les développeurs lors de la phase d'implémentation.

Plus spécifiquement, cette étape permet de :

- Identifier les **classes principales** du système et leurs responsabilités,
- Définir les **interactions dynamiques** entre les objets (via des diagrammes de séquence),
- Visualiser les **cas d'utilisation** pour chaque type d'utilisateur,
- Garantir la **cohérence et l'évolutivité** de l'architecture logicielle.

Grâce à cette modélisation, l'équipe de développement peut disposer d'une vision unifiée du système, ce qui **facilite la maintenance, le test**, ainsi que **l'ajout de nouvelles fonctionnalités** dans le futur.

3. Diagramme de classe :



Ce diagramme représente un modèle de classes simplifié qui décrit trois entités principales : **Utilisateur**, **Vote**, et **Authentification**, ainsi que leurs relations.

1. Classe Utilisateur

- **Attributs :**
 - id (int) : Identifiant unique de l'utilisateur.
 - nom (String) : Nom de l'utilisateur.
 - email (String) : Adresse email de l'utilisateur.
 - motDePasse (String) : Mot de passe de l'utilisateur.
 - role (String) : Rôle de l'utilisateur (par exemple, admin, utilisateur standard, etc.).
- **Méthode :**
 - s'authentifier(): boolean : Vérifie si l'utilisateur peut s'authentifier (retourne true ou false).

2. Classe Vote

- **Attributs :**
 - id (int) : Identifiant unique du vote.
 - dateVote (DateTime) : Date et heure du vote.
- **Méthode :**
 - enregistrer(): void : Permet d'enregistrer le vote.

3. Classe Authentification

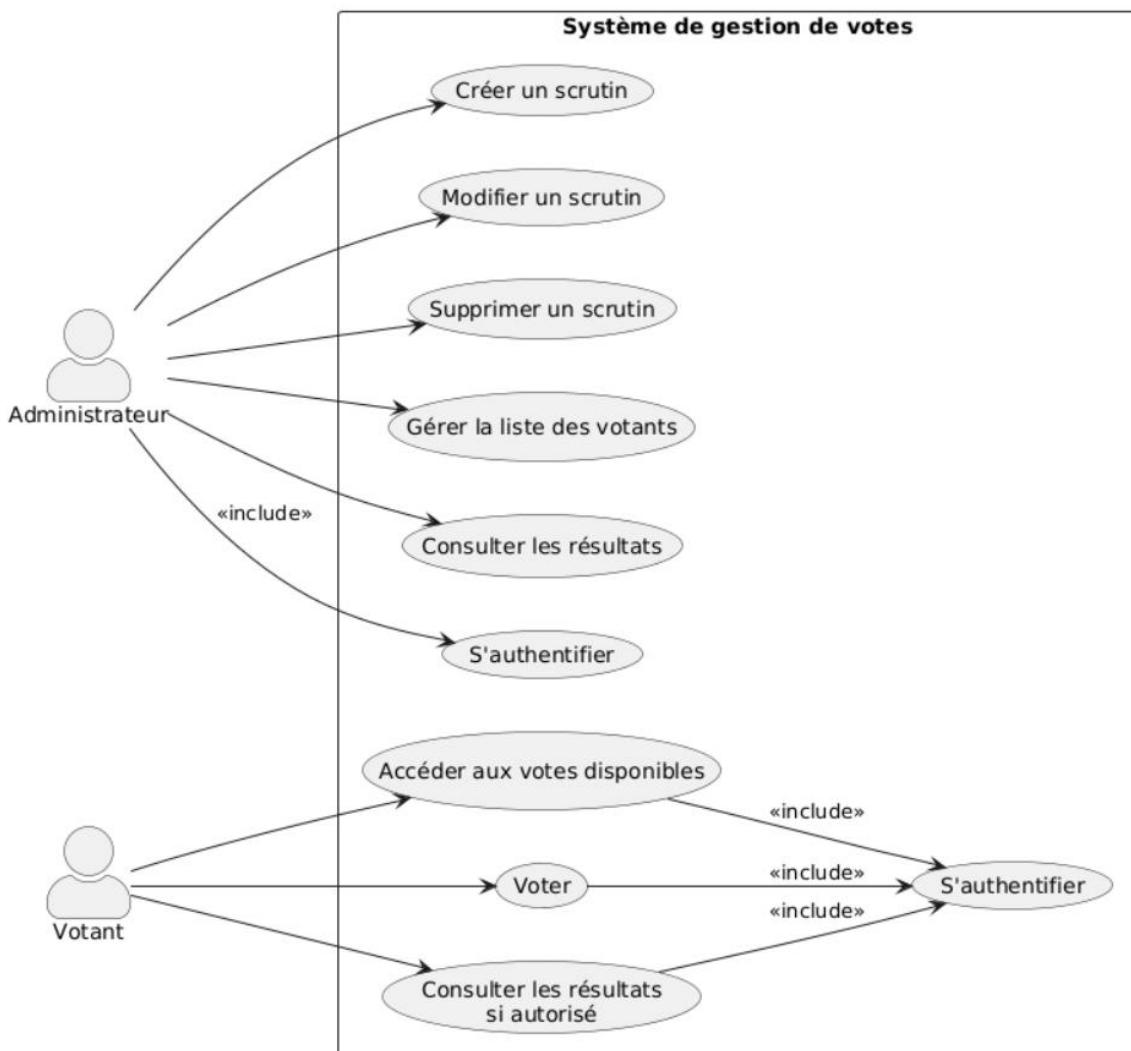
- **Méthode :**

- `verifierIdentifiants(email: String, motDePasse: String): boolean` : Vérifie si les identifiants (email et mot de passe) sont valides (retourne true ou false).

Relations entre les classes

- **Utilisateur → Vote** : La flèche indique qu'un utilisateur peut effectuer un vote. La notation 0.* suggère qu'un utilisateur peut effectuer zéro ou plusieurs votes.
- **Utilisateur → Authentification** : La flèche indique que l'utilisateur utilise le système d'authentification pour vérifier ses identifiants.

2. Diagramme de cas d'utilisation :



Ce diagramme décrit les fonctionnalités d'un **Système de gestion de votes**, en mettant l'accent sur les actions possibles pour deux acteurs principaux : l'**Administrateur** et le **Votant**.

Fonctionnalités générales du système

Le système permet de :

- **Créer un scrutin**
- **Modifier un scrutin**
- **Supprimer un scrutin**

Rôles et actions associées

1. Administrateur

L'administrateur dispose des fonctionnalités suivantes :

- **Gérer la liste des votants** (ajout, suppression, modification).
- **Consulter les résultats** des votes.
- **S'authentifier** pour accéder au système.
- **Accéder aux votes disponibles** (pour vérification ou gestion).
- **Inclure** (relation «include») les actions de base comme la création/modification/suppression des scrutins.

2. Votant

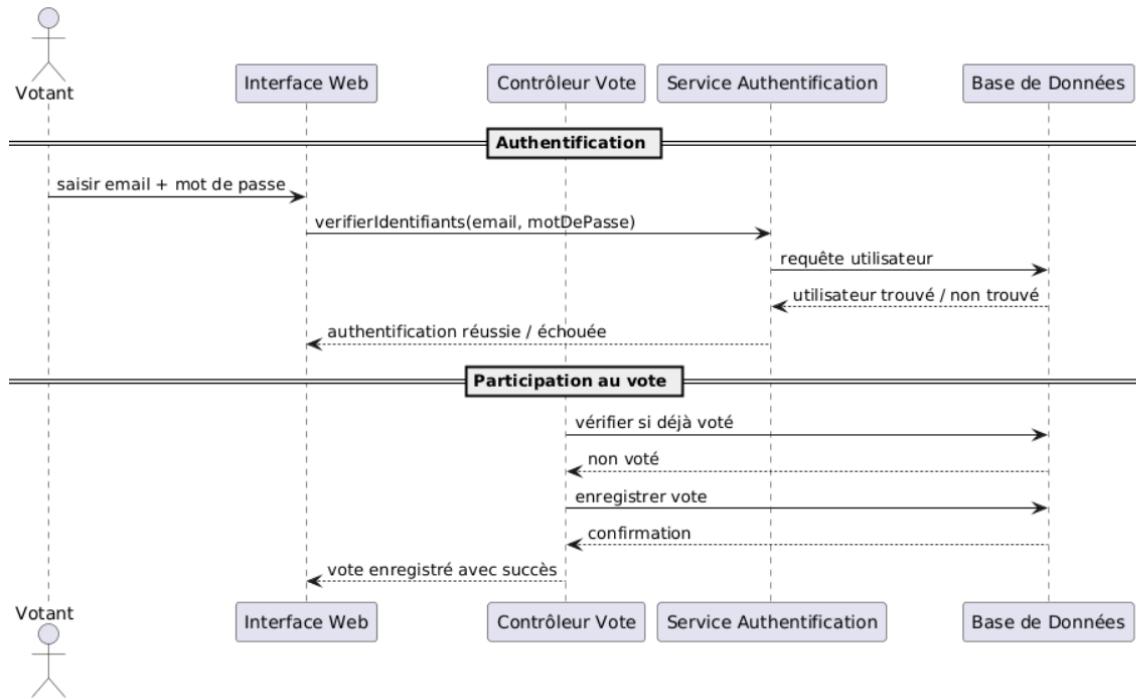
Le votant peut effectuer les actions suivantes :

- **Voter** dans un scrutin disponible.
- **Consulter les résultats** (si autorisé par l'administrateur).
- **S'authentifier** pour accéder aux fonctionnalités.
- **Inclure** (relation «include») l'authentification comme prérequis pour voter.

Relations entre les fonctionnalités

- L'authentification (**S'authentifier**) est un prérequis commun aux deux rôles (indiqué par «include»).
- L'administrateur a des privilèges étendus (gestion des scrutins et des votants), tandis que le votant a un accès limité (voter et consulter les résultats sous conditions).

3. Diagramme de séquence:



Ce diagramme décrit l'architecture et le flux d'une **Interface Web** pour un système de gestion de votes, en se concentrant sur les interactions entre les composants principaux : **Controller Vote**, **Services (Authentification)**, et la **Base de Données**.

Composants principaux

1. Controller Vote

- Orchestre les interactions entre l'interface utilisateur, les services et la base de données.

2. Services

- Authentification** : Gère la vérification des identifiants.

3. Base de Données

- Stocke les informations des utilisateurs, et votes.

Chapitre 4

Technologies

utilisées

Introduction

La mise en œuvre de l'application web de gestion de votes repose sur un ensemble de technologies modernes et éprouvées, assurant à la fois la fiabilité, la maintenabilité et la sécurité du système. Le choix de ces outils s'est basé sur leur adéquation avec les objectifs fonctionnels du projet, leur simplicité de déploiement, ainsi que leur compatibilité avec une architecture MVC (Modèle-Vue-Contrôleur).

1. Backend : Django (Python)

Le cœur de l'application est développé avec Django, un framework web open-source basé sur le langage Python. Django permet une structuration claire du code selon le modèle MVC, facilite la gestion des routes, des formulaires, de l'authentification, et propose un ORM intégré pour la gestion de la base de données.

Ses avantages incluent :

- Un système d'administration généré automatiquement,
- Une gestion sécurisée des utilisateurs et des permissions,
- Une architecture modulaire, extensible et robuste.

2. Frontend : HTML, CSS et JavaScript

L'interface utilisateur est conçue en HTML5 et CSS3, garantissant une compatibilité avec tous les navigateurs modernes. L'ajout de JavaScript permet d'améliorer l'interactivité, notamment lors de la validation des formulaires côté client ou pour des mises à jour dynamiques de l'interface (ex : affichage des résultats de vote en temps réel).

Des bibliothèques JavaScript comme Bootstrap ou jQuery peuvent être utilisées pour accélérer la mise en page responsive et enrichir l'expérience utilisateur.

3. Moteur de templates : Django Templates

Le framework Django intègre son propre moteur de templates, permettant de générer dynamiquement les pages HTML en liant directement les données issues du backend. Cela favorise une séparation claire entre la logique métier et l'affichage.

Les balises de template sécurisent automatiquement les sorties HTML pour éviter les injections XSS.

4. Base de données : SQLite (ou autre via ORM Django)

Par défaut, le projet utilise SQLite, une base de données légère, intégrée et suffisante pour les phases de développement et de test. Grâce à l'ORM de Django, le système peut facilement être migré vers une base plus robuste comme PostgreSQL ou MySQL en production, sans modification majeure du code.

L'ORM permet de manipuler les entités (utilisateurs, votes, scrutins...) sous forme d'objets Python, tout en assurant une abstraction des requêtes SQL.

5. Sécurité

L'application intègre plusieurs mécanismes de sécurité fournis par Django :

- Protection CSRF (Cross Site Request Forgery) activée sur tous les formulaires pour éviter les attaques par falsification de requêtes.
- Validation et nettoyage des formulaires pour empêcher les injections malveillantes.
- Gestion des sessions utilisateur sécurisée, avec expiration automatique et hachage des mots de passe via les algorithmes robustes (PBKDF2 par défaut).

Chapitre 5

Les interfaces

1. Interface d'administration

The screenshot shows the Django Admin interface for the 'Utilisateurs' (Users) model. The top navigation bar includes links for 'BONJOUR, HAFSA-SARA / VOIR LE SITE / MODIFIER LE MOT DE PASSE / DÉCONNEXION'. The left sidebar has a 'Filtrage' section and a 'VOTE_APP' section containing 'Options', 'Polls', 'Utilisateurs', and 'Votes' with 'Ajouter' buttons. The main content area displays a success message: 'L'objet utilisateur « anass-chamar » a été supprimé avec succès.' Below this, a list of users is shown with checkboxes: 'UTILISATEUR', 'SARA123@', 'HAFSA-SARA', and 'hafsaHAFSA@123'. A summary at the bottom indicates '3 utilisateurs'.

- Affiche la gestion des utilisateurs dans une application de vote.
- Permet de filtrer, ajouter ou supprimer des utilisateurs (ex: « anass-charnar » supprimé).
- Liste des utilisateurs disponibles pour modification (ex: SARAT23@, HAFSA-SARA).

2. Page d'inscription

The registration page features a dark header with the text 'Bienvenue sur notre application de vote'. The main form is titled 'Inscription' and contains four input fields: 'Nom d'utilisateur', 'Email', 'Mot de passe', and 'Confirmation du mot de passe'. Below the 'Mot de passe' field is a note: 'Saisissez le même mot de passe que précédemment, pour vérification.' A large 'S'inscrire' button is at the bottom. At the very bottom of the page, a small copyright notice reads '© VraiVote 2025'.

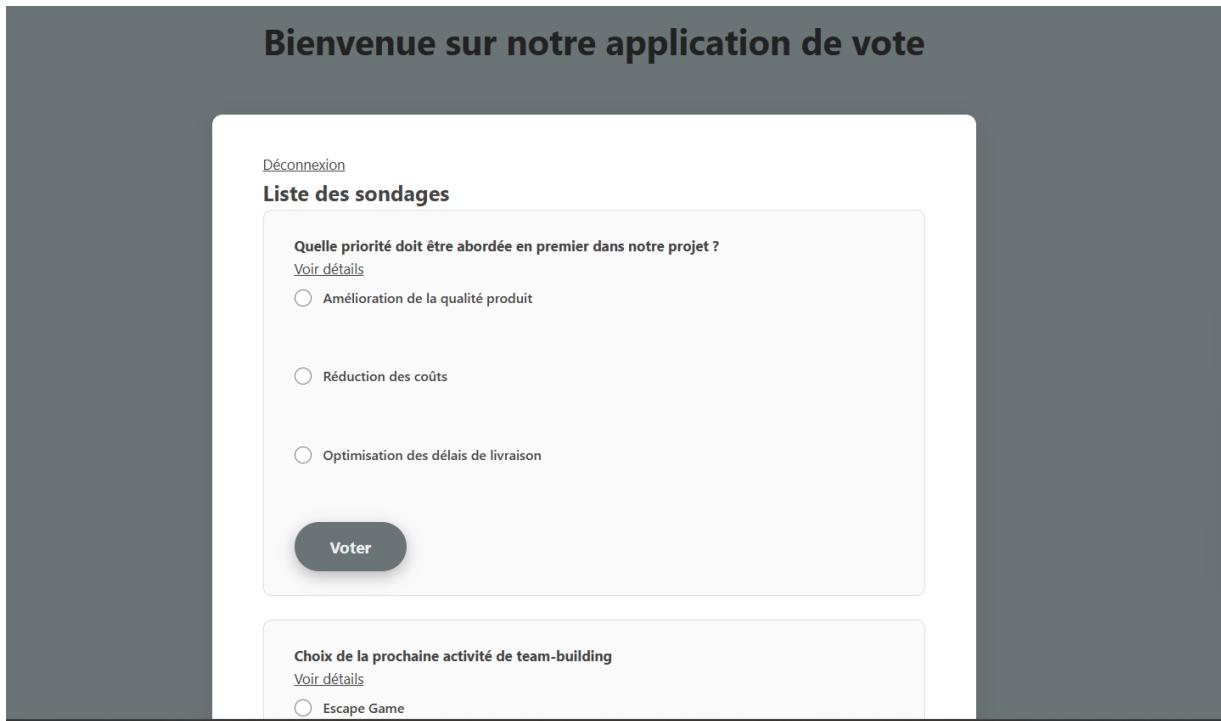
- Formulaire pour créer un compte avec :
 - Nom d'utilisateur (contraintes de caractères).
 - Email et mot de passe (avec confirmation).
- Bouton **S'inscrire**.

3. Page de connexion

The screenshot shows a dark-themed login page. At the top, the text "Bienvenue sur notre application de vote" is displayed. Below it is a white rectangular form with rounded corners. The form has a title "Connexion" at the top left. It contains two input fields: one for "Nom d'utilisateur" (username) and one for "Mot de passe" (password). Below these fields is a blue button labeled "Se connecter". At the bottom of the form, there is a link "Pas encore de compte ? [Inscrivez-vous ici](#)". At the very bottom center of the page, there is a small copyright notice: "© VraiVote 2025".

- Formulaire de connexion avec :
 - Champs pour le nom d'utilisateur et mot de passe.
 - Case à cocher possible (ex: [x]).
 - Lien vers l'inscription si pas de compte.

4. Interface utilisateur après connexion



- Liste des sondages actifs (ex: "Quelle priorité doit être abordée en premier ?").
- Options de vote avec bouton **[Voter]**.
- Lien **Voir détails** pour chaque sondage.

5.Message de vote déjà effectué

The screenshot shows a mobile application interface for a voting platform. At the top, it says "Bienvenue sur notre application de vote". Below that, there's a "Déconnexion" link. The main content area is titled "Quelle priorité doit être abordée en premier dans notre projet ?". It lists three options with descriptions:

- Amélioration de la qualité produit
Cette priorité vise à renforcer la fiabilité, la performance, et la satisfaction client liée au produit ou service final. Cela peut inclure des tests plus rigoureux, une meilleure conception, ou une attention accrue aux retours utilisateurs.
- Réduction des coûts
Optimiser les dépenses liées au projet, que ce soit en ressources humaines, matières premières ou logistique, afin d'améliorer la rentabilité et la compétitivité.
- Optimisation des délais de livraison
Accélérer la mise sur le marché en améliorant la gestion des délais, des processus internes, ou en utilisant des méthodes agiles.

A "Voter" button is at the bottom left, and a "Voir les résultats" link is at the bottom right.

- Notification indiquant que l'utilisateur a déjà voté pour un sondage.
- Bouton **Retour à la liste des sondages**.

6.Résultats d'un sondage

The screenshot shows a mobile application interface for a voting platform. At the top, it says "Bienvenue sur notre application de vote". Below that, a message box contains the text "Vous avez déjà voté pour ce sondage." and a "Retour à la liste des sondages" link. At the bottom of the screen, there's a small copyright notice: "© VraiVote 2025".

- Affiche les résultats statistiques (ex: 50% pour "Amélioration de la qualité produit").
- Total des votes et répartition en pourcentages.

7.Gestion des options dans l'admin Django

Déconnexion

Résultats - Quelle priorité doit être abordée en premier dans notre projet ?
Total des votes : 2

Pourcentage des votes par option :

- Amélioration de la qualité produit : 50,0%
- Réduction des coûts : 0,0%
- Optimisation des délais de livraison : 50,0%

- Interface pour modifier les options de vote (ex: "Optimisation des délais de livraison").
- Permet de sélectionner/désélectionner des options.

8.Ajout manuel d'un vote (Admin Django)

Administration de Django

Bienvenue, HAFSA-SARA. VOIR LE SITE / MODIFIER LE MOT DE PASSE

Accueil > Vote_App > Votes > Ajouter vote

Écrivez ici pour filtrer...

AUTHENTIFICATION ET AUTORISATION

Groupes + Ajouter

VOTE_APP

Options + Ajouter

Polls + Ajouter

Utilisateurs + Ajouter

Votes + Ajouter

Ajout de vote

User : ----- + ☰ + ☰

Poll : ----- + ☰ + ☰

Options : +

Flexibilité des horaires de travail
Télétravail et travail hybride
Amélioration des espaces de travail
Programmes de bien-être au travail
Escape Game
Atelier cuisine
Journée de sport collectif
Amélioration de la qualité produit

Maintenez appuyé « Ctrl », ou « Commande (touche pomme) » sur un Mac, pour en sélectionner plusieurs.

ENREGISTRER Enregistrer et ajouter un nouveau Enregistrer et continuer les modifications

- Formulaire pour enregistrer un vote :
 - Sélection d'un utilisateur, d'un sondage et d'options (multi-sélection possible).
- Boutons ENREGISTRER ou Enregistrer et continuer.

9. Ajout d'un utilisateur (Admin Django)

The screenshot shows the Django Admin interface for adding a user. The left sidebar lists 'Utilisateurs' under the 'VOTE_APP' section. The main form has fields for 'Mot de passe', 'Dernière connexion' (with date and time pickers), and a checkbox for 'Statut super-utilisateur'. Below these are sections for 'Groupes' (with a '+' button to add) and notes about super-user status and group inheritance.

- Formulaire de création d'utilisateur avec :
 - Mot de passe, statut super-utilisateur, groupes.
 - Notes sur les permissions et l'heure du serveur.

10. Crédit d'un sondage (Admin Django)

The screenshot shows the Django Admin interface for adding a poll. The left sidebar lists 'Polls' under the 'VOTE_APP' section. The main form includes fields for 'Title', 'Description', and two sets of date/time inputs for 'Start date' and 'End date'. Notes about server time and hour-ahead offset are present.

- Formulaire pour ajouter un nouveau sondage :
 - Titre, description, dates de début/fin.
 - Réglage de l'heure (avec décalage serveur indiqué).

11. Interface d'administration Django - Gestion des sondages

The screenshot shows the Django admin interface for the 'Votes' application. The left sidebar has a 'VOTE_APP' section with 'Polls' selected. The main area displays a list of three polls: 'Quelle priorité doit être abordée en premier dans notre projet ?', 'Choix de la prochaine activité de team-building', and 'Amélioration des conditions de travail'. Each poll has a checkbox next to it. At the top right, there's a button labeled 'AJOUTER POLL +'. The top navigation bar includes links for 'Accueil', 'Vote_App', and 'Polls', along with user information and a 'DÉCONNEXION' link.

- **Section principale :** Liste des sondages ("Polls") disponibles pour modification :
 - *Exemples :*
 - "Quelle priorité doit être abordée en premier dans notre projet ?"
 - "Choix de la prochaine activité de team-building"
 - "Amélioration des conditions de travail"
 - Permet de sélectionner un ou plusieurs sondages via des cases à cocher.
- **Actions disponibles :**
 - Boutons **Envoyer** (pour appliquer des actions groupées) et indication **0 sur 3 sélectionné**.
- **Menu latéral :**
 - Accès rapide aux modules : **Groupes, Options, Polls, Utilisateurs, Votes** (avec icônes "Ajouter").
- **En-tête :**
 - Message de bienvenue personnalisé : "**BENVENUE: HAFAA-SARA**" avec liens pour :
 - Voir le profil / Modifier le mot de passe / Se déconnecter

Conclusion

Ce projet de développement d'une application web de gestion de votes s'inscrit dans une démarche pédagogique et professionnelle visant à concevoir une solution numérique moderne, sécurisée et adaptée aux besoins d'une organisation souhaitant numériser ses processus de vote.

L'objectif principal était de créer un système simple d'utilisation, mais suffisamment robuste pour permettre aux administrateurs de créer, gérer et superviser des scrutins, tout en offrant aux votants une interface sécurisée et intuitive pour participer aux élections ou sondages. Pour cela, nous avons adopté une architecture basée sur le framework **Django (Python)**, combinant un backend puissant, une interface web claire (HTML/CSS/JS) et une base de données intégrée via l'ORM Django.

Durant ce projet, plusieurs aspects ont été abordés :

- **L'analyse des besoins** fonctionnels et non fonctionnels, permettant de cerner les attentes des utilisateurs finaux et les contraintes techniques.
- **La modélisation UML** du système (cas d'utilisation, diagrammes de classes et de séquence), afin de structurer la solution selon les principes de l'analyse orientée objet.
- **La mise en œuvre technique**, avec des fonctionnalités clés telles que l'authentification sécurisée, la gestion des scrutins et des candidats, le vote unique par utilisateur, et la consultation des résultats.
- **L'intégration de mesures de sécurité**, incluant la protection CSRF, le hachage des mots de passe, et le contrôle d'accès basé sur les rôles des utilisateurs.

Ce travail a permis de développer une solution fonctionnelle, évolutive et facilement déployable dans différents contextes (scolaire, associatif, professionnel). Il a également été une opportunité précieuse pour approfondir les compétences en développement web, gestion de projet, modélisation logicielle, et bonnes pratiques de sécurité.

Perspectives d'amélioration

Bien que l'application réponde aux objectifs de base, plusieurs améliorations peuvent être envisagées pour enrichir le système :

- Intégration de **statistiques en temps réel** sur la participation ou les résultats,
- Ajout de **notifications par email** pour informer les votants du lancement d'un scrutin,
- Passage à une **authentification multifacteur** pour renforcer la sécurité,

- Déploiement dans un environnement cloud sécurisé, avec une base de données relationnelle plus robuste (ex : PostgreSQL),
- Interface utilisateur dynamique grâce à un framework frontend moderne (ex : React.js ou Vue.js).

En conclusion, ce projet a permis de démontrer la faisabilité d'un système de vote numérique fiable, en mettant en pratique les concepts appris tout au long de la formation. Il constitue une base solide pouvant évoluer vers une solution complète à usage réel.