1. Système de Réservation de Salles de Formation:

1. Contexte et Objectifs

Les entreprises et établissements éducatifs ont besoin d'un système efficace pour gérer les **réservations de salles** en fonction de la disponibilité et des besoins des utilisateurs (formateurs, étudiants, entreprises). Ce projet vise à développer une **plateforme centralisée** permettant aux utilisateurs de **réserver des salles en ligne**, consulter la disponibilité en temps réel et recevoir des confirmations de réservation.

2. Périmètre du Projet

- L'application doit permettre :
  - La gestion des salles (ajout, suppression, modification).
  - La réservation en ligne avec affichage des disponibilités.
  - L'annulation et modification des réservations.
  - L'envoi de notifications aux utilisateurs.

3. Spécifications Fonctionnelles :

3.1 Gestion des Salles

- CRUD (Ajout, Modification, Suppression) des salles.
- Informations requises :
  - Nom de la salle.
  - Capacité d'accueil.
  - Équipements disponibles (vidéoprojecteur, Wi-Fi, tableau blanc, etc.).
  - Horaires d'ouverture et fermeture.
- Affichage d'un **calendrier interactif** des disponibilités.

3.2 Réservation des Salles

- Formulaire de réservation avec :
  - Nom du demandeur.
  - Date et heure de réservation.
  - Motif de la réservation.
- Vérification automatique de la disponibilité.
- Confirmation immédiate ou en attente de validation.
- Envoi d'un **email de confirmation** après réservation.

3.3 Modification et Annulation de Réservation

- Annulation possible jusqu'à **24h avant** la réservation.
- Modification des créneaux si la salle est disponible.

- Historique des réservations pour chaque utilisateur.

3.4 Notifications et Suivi

- Envoi automatique d'un **rappel par email** avant la réservation.
- Notification des administrateurs en cas d'annulation.

4. Gestion des Utilisateurs

- **Utilisateurs (Employés/Étudiants/Formateurs)** : Effectuent des réservations.
- **Administrateurs** : Ajoutent et gèrent les salles, valident les réservations.

5. Parcours Utilisateur

Utilisateur

1. Se connecte à la plateforme.
2. Consulte les disponibilités via un calendrier interactif.
3. Effectue une réservation.
4. Reçoit une confirmation par email.
5. Accède à l'historique et peut modifier ou annuler.

Administrateur

1. Ajoute/modifie des salles et équipements.
2. Gère les réservations en attente de validation.
3. Reçoit les notifications en cas d'annulation.

6. Contraintes Techniques

- **Backend :** Django, Django REST Framework.
- **Frontend :** HTML, CSS, JavaScript (ou Vue.js/React).
- **Base de données :** PostgreSQL, MySQL ou Sqlite
- **Authentification :** Django Auth.
- **Affichage des disponibilités** : Intégration d'un calendrier interactif (FullCalendar.js ou autre).

2. Plateforme de Gestion des Candidatures

1. Contexte et Objectifs

De nombreuses entreprises reçoivent quotidiennement des candidatures via divers canaux (emails, formulaires, LinkedIn). Le suivi manuel est fastidieux et entraîne des pertes d'informations et un manque de visibilité sur le recrutement. L'objectif de cette plateforme est d'**automatiser la gestion des candidatures**, centraliser les informations et permettre aux recruteurs de filtrer, suivre et répondre efficacement aux candidats.

2. Périmètre du Projet

- L'application doit permettre :
    a. La publication d'offres d'emploi par les recruteurs.
    b. La soumission de candidatures avec pièces jointes (CV, lettre de motivation).
    c. Le filtrage des candidatures selon divers critères.
    d. Le suivi des candidatures avec notifications et statuts.

3. Spécifications Fonctionnelles

3.1 Gestion des Offres d'Emploi

- CRUD (Création, Lecture, Mise à jour, Suppression) des offres.
- Champs obligatoires : titre, description, compétences requises, type de contrat, localisation, date de publication.
- Possibilité d'attacher une image et un fichier PDF descriptif.

3.2 Soumission des Candidatures

- Formulaire avec :
    - Nom, prénom, email, téléphone.
    - Téléchargement du CV (PDF).
    - Téléchargement de la lettre de motivation (PDF).
- Envoi d'un email automatique au candidat après soumission.

3.3 Filtrage et Recherche de Candidatures

- Recherche avancée par mots-clés, compétences, niveau d'études, localisation.
- Tri par date de candidature ou score de pertinence.
- Filtrage par statut (en attente, accepté, refusé).

3.4 Suivi des Candidatures

- Modification du statut (en attente, accepté, refusé, entretien planifié).
- Système de messagerie intégrée pour communiquer avec les candidats.
- Génération de rapports sur les candidatures reçues.

4. Gestion des Utilisateurs

- **Candidats** : créer un compte, soumettre des candidatures, suivre l'état de leur demande.
- **Recruteurs** : publier des offres, filtrer les candidatures, contacter les candidats.
- **Administrateur** : gérer les utilisateurs, surveiller l'activité de la plateforme.

5. Parcours Utilisateur

Candidat

1. S'inscrit sur la plateforme.
2. Consulte les offres d'emploi.
3. Soumet une candidature (formulaire + CV/lettre).
4. Reçoit un email de confirmation.
5. Suit le statut de sa candidature.

Recruteur

1. Crée un compte.
2. Publie des offres d'emploi.
3. Consulte et filtre les candidatures reçues.
4. Change le statut et envoie des messages aux candidats.

6. Contraintes Techniques

- **Backend :** Django, Django REST Framework.
- **Frontend :** HTML, CSS, JavaScript (ou Vue.js/React).
- **Base de données :** PostgreSQL, MySQL ou Sqlite.
- **Authentification :** Django Auth (gestion des rôles candidats/recruteurs/admins).
- **Stockage des CVs et lettres de motivation** : Système de fichiers Django

3. Système de Gestion des Tickets de Support

1. Contexte et Objectifs

Les entreprises et services informatiques doivent gérer efficacement les **demandes de support** des utilisateurs. Ce projet vise à développer un **système de gestion des tickets** permettant aux utilisateurs de soumettre des demandes d'assistance et aux agents de support de suivre et résoudre ces tickets.

2. Périmètre du Projet

- L'application doit permettre :
  a. La création et le suivi des tickets par les utilisateurs.
  b. L'assignation des tickets aux agents de support.
  c. La mise à jour des statuts et l'envoi de notifications.
  d. L'historique des tickets et la génération de rapports.

3. Spécifications Fonctionnelles

3.1 Création et Gestion des Tickets

- CRUD des tickets (Création, Lecture, Modification, Suppression).
- Champs obligatoires :
  - Titre du ticket.
  - Description du problème.
  - Catégorie (logiciel, matériel, réseau, autre).
  - Niveau de priorité (faible, moyen, élevé).
  - Pièce jointe (capture d'écran, PDF, max 5 Mo).
- Statut du ticket : **Ouvert, En cours, Résolu, Fermé**.

3.2 Assignation et Gestion des Tickets

- Un ticket peut être **attribué à un agent de support**.
- Possibilité pour un utilisateur de commenter un ticket en cours.
- Suivi des actions effectuées sur le ticket (logs, historique des mises à jour).

3.3 Notifications et Alertes

- Email envoyé à l'utilisateur lors de la création du ticket.
- Notification à l'agent de support lorsqu'un ticket lui est attribué.
- Rappel automatique si un ticket reste sans réponse pendant X jours.

3.4 Génération de Rapports et Statistiques

- Nombre de tickets traités par agent.
- Temps moyen de résolution des tickets.
- Répartition des tickets par catégorie et priorité.

4. Gestion des Utilisateurs

- **Utilisateur standard** : Peut créer et suivre ses tickets.
- **Agent de support** : Peut voir, attribuer et résoudre les tickets.
- **Administrateur** : Peut gérer les agents et surveiller l'activité globale.

5. Parcours Utilisateur

Utilisateur

1. Se connecte et crée un ticket via un formulaire.
2. Reçoit un email de confirmation.
3. Consulte l'état de son ticket et échange avec l'agent.
4. Ferme le ticket une fois la solution trouvée.

Agent de Support

4. Consulte la liste des tickets non attribués.
5. Prend en charge un ticket et le met à jour.
6. Communique avec l'utilisateur pour résoudre le problème.
7. Change le statut du ticket une fois résolu.

Administrateur

1. Gère les agents et surveille l'évolution des tickets.
2. Analyse les statistiques pour améliorer la qualité du support.

6. Contraintes Techniques

- **Backend :** Django, Django REST Framework.
- **Frontend :** HTML, CSS, JavaScript (ou Vue.js/React).

- **Base de données :** PostgreSQL , MySQL ou SQlite
- **Authentification :** Django Auth.
- **Stockage des pièces jointes** : Système de fichiers

4. Plateforme de Gestion des Événements et Inscriptions

1. Contexte et Objectifs

Les entreprises, écoles et associations organisent régulièrement des **événements (conférences, séminaires, workshops, webinaires)** et doivent gérer efficacement les **inscriptions, les participants et la communication** autour de ces événements. Cette plateforme permettra aux organisateurs de créer et gérer des événements et aux utilisateurs de s'inscrire facilement.

2. Périmètre du Projet

- L'application doit permettre :
    a. La gestion des événements (création, modification, suppression).
    b. L'inscription des participants avec envoi de confirmation.
    c. La gestion des listes de participants et des statistiques.
    d. L'envoi de notifications et rappels automatiques.

3. Spécifications Fonctionnelles

3.1 Gestion des Événements

- CRUD des événements (Ajout, Modification, Suppression).
- Champs obligatoires :
    - Nom de l'événement.
    - Description.
    - Date et heure.
    - Lieu (physique ou en ligne).
    - Nombre de places disponibles.
    - Affiche ou bannière (image).
- Fonctionnalité d'événements publics/privés (avec accès restreint).

3.2 Inscription des Participants

- Formulaire d'inscription avec :
    - Nom, prénom, email.
    - Numéro de téléphone.
    - Option d'ajouter des notes ou préférences (ex. régime alimentaire).
- Vérification du nombre de places disponibles avant validation.
- Email de confirmation automatique après inscription.

3.3 Gestion des Participants et Listes

- Liste des participants avec export en Excel/PDF.

- Fonctionnalité de validation manuelle pour certains événements.
- Statut des inscriptions : **Confirmé, En attente, Annulé**.

3.4 Notifications et Rappels

- Envoi automatique d'un **rappel 24h avant** l'événement.
- Notifications en cas d'annulation ou modification de l'événement.

4. Gestion des Utilisateurs

- **Utilisateurs standard** : Peuvent s'inscrire aux événements.
- **Organisateurs** : Peuvent créer et gérer leurs événements.
- **Administrateur** : Gère les événements et surveille l'activité.

5. Parcours Utilisateur

Participant

1. Se connecte et consulte la liste des événements disponibles.
2. Sélectionne un événement et s'inscrit via un formulaire.
3. Reçoit un email de confirmation.
4. Reçoit un rappel automatique avant l'événement.

Organisateur

1. Crée un nouvel événement avec tous les détails.
2. Consulte la liste des inscriptions en temps réel.
3. Modifie ou annule un événement si nécessaire.

Administrateur

1. Supervise les événements et gère les droits des organisateurs.
2. Analyse les inscriptions et génère des statistiques.

6. Contraintes Techniques

- **Backend:** Django, Django REST Framework.
- **Frontend :** HTML, CSS, JavaScript (ou Vue.js/React).
- **Base de données :** PostgreSQL, MySQL ou SQLite.
- **Authentification :** Django Auth avec rôles utilisateur/organisateur/admin.
- **Stockage des images des événements** : Système de fichiers

5. Plateforme de Gestion de Formations en Ligne

1. Contexte et Objectifs

Les établissements d'enseignement et les entreprises offrent des formations en ligne. Cette plateforme permettra aux formateurs de créer et de gérer des cours, aux étudiants de s'inscrire et de suivre les formations à distance. L'objectif est de faciliter la gestion des inscriptions, des contenus pédagogiques et du suivi des progrès des étudiants.

2. Périmètre du Projet

- L'application doit permettre :
    a. La création et gestion des cours (création, modification, suppression).
    b. L'inscription des étudiants et gestion de leurs progrès.
    c. L'envoi de notifications et rappels pour les échéances des devoirs et examens.
    d. La gestion des évaluations et des certifications des étudiants.

3. Spécifications Fonctionnelles

3.1 Gestion des Cours

- CRUD des cours (Ajout, Modification, Suppression).
- Champs obligatoires :
    - Titre du cours.
    - Description du contenu.
    - Durée estimée du cours.
    - Niveau (débutant, intermédiaire, avancé).
    - Liste des modules du cours avec ressources associées (vidéos, PDF, quiz).
    - Date de début et de fin.

3.2 Inscription et Suivi des Étudiants

- Formulaire d'inscription avec :
    - Nom, prénom, email.
    - Choix des cours à suivre.
    - Suivi en temps réel des progrès (vidéos vues, quiz réussis).
    - Option de certification à la fin du cours.
- Envoi de notifications pour les examens et rappels pour les échéances.

3.3 Évaluations et Certifications

- Création de quiz et devoirs à la fin de chaque module.
- Génération de certificats numériques pour les étudiants ayant terminé un cours avec succès.

- Gestion des évaluations avec possibilité de révision.

4. Gestion des Utilisateurs

- **Étudiants :** Peuvent s'inscrire à des cours, suivre les modules, passer des évaluations, et recevoir des certificats.
- **Formateurs :** Peuvent créer, modifier et supprimer des cours, suivre les progrès des étudiants, et évaluer leurs résultats.
- **Administrateurs :** Gèrent l'ensemble des cours et des utilisateurs, génèrent des rapports et des statistiques.

5. Contraintes Techniques

- **Backend :** Django, Django REST Framework.
- **Frontend :** HTML, CSS, JavaScript (React ou Vue.js).
- **Base de données :** PostgreSQL, MySQL ou SQLite .
- **Authentification :** Système d'authentification via email/password ou Django Auth.
- **Stockage des contenus multimédia :** Système de fichiers

6. Plateforme de Gestion de Réservations d'Hôtels

1. Contexte et Objectifs

Les hôtels, complexes touristiques et auberges doivent offrir une gestion fluide de leurs réservations en ligne. Cette plateforme permettra aux utilisateurs de rechercher des hôtels, réserver des chambres et aux administrateurs de gérer les réservations, les disponibilités et les tarifs en temps réel.

2. Périmètre du Projet

- L'application doit permettre :
    a. La gestion des hôtels (création, modification, suppression).
    b. La gestion des réservations et disponibilités des chambres.
    c. La gestion des avis clients et des évaluations des hôtels.
    d. La gestion des promotions et des offres spéciales.

3. Spécifications Fonctionnelles

3.1 Gestion des Hôtels

- CRUD des hôtels (Ajout, Modification, Suppression).
- Champs obligatoires :
    - Nom de l'hôtel.
    - Description.
    - Adresse et coordonnées (email, téléphone).
    - Type de chambres (single, double, suite).
    - Tarifs des chambres par période (saison haute, saison basse).
    - Disponibilité des chambres (dates disponibles et occupées).

3.2 Gestion des Réservations

- Formulaire de réservation avec :
    - Date d'arrivée et de départ.
    - Nombre d'adultes et d'enfants.
    - Type de chambre souhaitée.
    - Informations sur le client (nom, prénom, email, téléphone).
    - Option de paiement en ligne (par carte bancaire ou PayPal).
    - Envoi de confirmation de réservation par email.
- Gestion des annulations et modifications des réservations avec notifications automatiques.

3.3 Gestion des Avis et Évaluations

- Les clients peuvent laisser des avis sur les hôtels après leur séjour.

- Système de notation (1 à 5 étoiles) sur différents critères : propreté, accueil, confort, services.
- Modération des avis par les administrateurs pour éviter les abus.

3.4 Gestion des Promotions et Offres Spéciales

- Création de promotions saisonnières ou événementielles.
- Application automatique des réductions lors de la réservation.
- Notifications aux utilisateurs des nouvelles promotions via email.

4. Gestion des Utilisateurs

- **Clients :** Peuvent rechercher des hôtels, effectuer des réservations, laisser des avis et payer en ligne.
- **Hôteliers :** Peuvent ajouter, modifier et supprimer leurs hôtels, gérer les chambres et les réservations.
- **Administrateurs :** Gèrent l'ensemble des hôtels, réservations, et modèrent les avis des clients.

5. Parcours Utilisateur

- **Client** :
  - Se connecte ou s'inscrit sur la plateforme.
  - Recherche un hôtel en fonction de la destination, des dates, et du nombre de personnes.
  - Réserve une chambre et effectue le paiement en ligne.
  - Reçoit une confirmation de réservation et une facture.
  - Laisse un avis après son séjour.
- **Hôtelier** :
  - Crée un profil pour son hôtel avec toutes les informations nécessaires.
  - Ajoute les chambres disponibles, les tarifs et les périodes de disponibilité.
  - Gère les réservations et les annulations.
  - Consulte les avis des clients.
- **Administrateur** :
  - Gère les utilisateurs (clients et hôteliers).
  - Modère les avis et assure la bonne gestion des promotions et offres spéciales.
  - Génère des rapports sur les réservations et les performances des hôtels.

6. Contraintes Techniques

- **Backend :** Django, Django REST Framework.
- **Frontend :** HTML, CSS, JavaScript (React ou Vue.js).
- **Base de données :** PostgreSQL, MySQL ou SQLite
- **Authentification :** Système d'authentification par email et mot de passe, avec possibilité de gestion de session ou Django Auth.