**Plan pour la Section "Événements et Annonces"**

**I. Analyse et Conception**

1. **Définition des Besoins**
   * **Identification des utilisateurs cibles** : Qui utilisera cette fonctionnalité ? Animateurs de la radio, auditeurs, administrateurs, etc.
   * **Types d'événements** : Quels types d'événements seront annoncés ? Concerts, interviews, émissions spéciales, etc.
   * **Fonctionnalités nécessaires** :
     + Création et gestion des événements.
     + Affichage des événements sur le site.
     + Notifications et rappels pour les utilisateurs.
     + Intégration avec des API de calendrier (comme Eventbrite).
2. **Conception des Maquettes et Wireframes**
   * **Wireframes** : Dessiner les écrans principaux (page d'accueil, page des événements, page de détails de l'événement).
   * **Maquettes** : Utiliser un outil comme Figma ou Sketch pour créer des maquettes détaillées.

**II. Développement Frontend**

1. **Structure HTML**
   * Création des pages principales :
     + Page d'accueil avec un aperçu des événements.
     + Page de liste des événements.
     + Page de détail pour chaque événement.
2. **Stylisation CSS**
   * Utilisation de CSS3, SASS/SCSS, Bootstrap pour créer un design attrayant et réactif.
   * Mise en place des styles globaux et spécifiques à chaque page.
3. **Composants React.js**
   * Création de composants réutilisables :
     + Composant pour l'affichage de la liste des événements.
     + Composant pour l'affichage détaillé d'un événement.
     + Composant pour les formulaires de création/édition d'événements.

**III. Développement Backend**

1. **API REST avec Node.js et Express.js**
   * **Architecture de l'API** :
     + Endpoint pour créer un événement : POST /events
     + Endpoint pour obtenir tous les événements : GET /events
     + Endpoint pour obtenir un événement spécifique : GET /events/:id
     + Endpoint pour mettre à jour un événement : PUT /events/:id
     + Endpoint pour supprimer un événement : DELETE /events/:id
2. **Base de Données**
   * Utilisation de MongoDB ou MySQL pour stocker les informations sur les événements.
   * Modèle de données pour les événements :

{

"id": "unique identifier",

"title": "Event Title",

"description": "Detailed description of the event",

"date": "2024-05-20T14:00:00Z",

"location": "Event Location",

"createdBy": "User ID"

}

**IV. Intégration des API Externes**

1. **API de Calendrier d'Événements (Eventbrite)**
   * Intégration pour récupérer et afficher des événements externes.
   * Utilisation de l'API Eventbrite pour la création et la gestion des événements.

**V. Tests et Déploiement**

1. **Tests Unitaires et d'Intégration**
   * Tests pour chaque composant frontend avec Jest et React Testing Library.
   * Tests pour les endpoints de l'API avec Mocha et Chai.
2. **Déploiement**
   * Déploiement de l'application frontend sur un service comme Netlify ou Vercel.
   * Déploiement de l'API backend sur un service comme Heroku ou AWS.

**VI. Documentation et Formation**

1. **Documentation**
   * Documenter chaque endpoint de l'API.
   * Documenter les composants React et leur utilisation.
2. **Formation**
   * Préparer des tutoriels ou guides pour les utilisateurs finaux sur l'utilisation de la section événements.