

Semaine 1 : Introduction et Bases

1. **Jour 1-2 : Hello World**

- Créez une simple page React affichant "Hello World".
- Créez une API ASP.NET retournant un message de bienvenue.

2. **Jour 3-4 : Formulaire de Contact**

- Créez un formulaire React avec des champs pour le nom, l'email et le message, sans enregistrement.
- Connectez React à l'API ASP.NET pour soumettre et afficher les données du formulaire.

3. **Jour 5-7 : Liste des Utilisateurs**

- Créez une API ASP.NET renvoyant une liste d'utilisateurs.
- Affichez cette liste dans un composant React.
- Implémentez les opérations CRUD pour les utilisateurs.

Semaine 2 : Intégration et Gestion des Données

4. **Jour 1-2 : Validation de Formulaire**

- Ajoutez la validation côté client et côté serveur pour le formulaire d'utilisateur.

5. **Jour 3-4 : Pagination et Recherche**

- Ajoutez la pagination à la liste des utilisateurs.
- Ajoutez une barre de recherche pour filtrer les utilisateurs.

6. **Jour 5-7 : Mise à Jour et Suppression**

- Ajoutez des fonctionnalités pour mettre à jour et supprimer des utilisateurs.

- Implémentez un feedback visuel pour les actions de mise à jour et de suppression.

Semaine 3 : Authentification et Autorisation

7. **Jour 1-3 : Authentification**

- Implémentez un système de connexion avec JWT dans ASP.NET.
- Créez un formulaire de connexion dans React.

8. **Jour 4-5 : Autorisation**

- Protégez certaines routes et fonctionnalités avec des rôles d'utilisateur.
- Utilisez React Router pour gérer les routes protégées.

9. **Jour 6-7 : Profil Utilisateur**

- Ajoutez des profils utilisateurs avec modification des informations personnelles.

Semaine 4 : Téléchargement et Gestion des Fichiers

10. **Jour 1-3 : Téléchargement de Fichiers**

- Implémentez une fonctionnalité de téléchargement de fichiers dans ASP.NET.
- Créez un composant React pour uploader des fichiers.

11. **Jour 4-5 : Affichage des Fichiers**

- Affichez les fichiers téléchargés dans une liste avec des options pour les visualiser ou les supprimer.

12. **Jour 6-7 : Notifications**

- Ajoutez des notifications pour les actions utilisateur (par exemple, enregistrement réussi).

Semaine 5 : Thèmes et Gestion des États

13. **Jour 1-3 : Thèmes**

- Implémentez un changement de thème (clair/sombre) dans l'application React.

14. **Jour 4-5 : Gestion des États Globaux**

- Utilisez Redux ou Context API pour gérer l'état global de l'application.

15. **Jour 6-7 : Test Unitaire**

- Écrivez des tests unitaires pour les composants React et les contrôleurs ASP.NET.

Semaine 6 : Fonctionnalités Avancées

16. **Jour 1-3 : Chat en Temps Réel**

- Créez un chat en temps réel avec SignalR et WebSockets.

17. **Jour 4-5 : Dashboard**

- Créez un tableau de bord avec des graphiques en utilisant Chart.js.

18. **Jour 6-7 : Recherche Avancée**

- Implémentez une recherche avancée avec filtres et tris.

Semaine 7 : Applications Complètes

19. **Jour 1-3 : Gestion des Tâches**

- Créez une application de gestion des tâches avec des statuts (à faire, en cours, terminé).

20. **Jour 4-5 : Blog**

- Implémentez un système de blog avec création, lecture, mise à jour et suppression d'articles.

21. **Jour 6-7 : E-commerce**

- Créez un panier d'achat simple avec ajout et suppression de produits.

Semaine 8 : Internationalisation et Améliorations

22. **Jour 1-3 : Internationalisation**

- Ajoutez la prise en charge de plusieurs langues dans l'application.

23. **Jour 4-5 : Application de Recettes**

- Créez une application où les utilisateurs peuvent soumettre et rechercher des recettes.

24. **Jour 6-7 : Profil Utilisateur Avancé**

- Ajoutez des fonctionnalités avancées au profil utilisateur (avatar, biographie, etc.).

Semaine 9 : Applications Complexes

25. **Jour 1-3 : Application de Notes**

- Créez une application pour prendre des notes avec la possibilité de les organiser en catégories.

26. **Jour 4-5 : Calendrier**

- Implémentez un calendrier avec événements et rappels.

27. **Jour 6-7 : Application de Suivi d'Habitudes**

- Créez une application pour suivre les habitudes quotidiennes.

Semaine 10 : Finalisation et Déploiement

28. **Jour 1-2 : Optimisation et Révision**

- Optimisez le code de vos applications précédentes.
- Revoyez les concepts clés et assurez-vous que tout fonctionne correctement.

29. **Jour 3-4 : Tests et Documentation**

- Écrivez des tests supplémentaires pour garantir la robustesse de vos applications.
- Documentez votre code et les fonctionnalités de chaque application.

30. **Jour 5-7 : Déploiement**

- Déployez vos applications sur une plateforme comme Heroku, Vercel ou Azure.
- Assurez-vous que tout fonctionne en production et faites des ajustements si nécessaire.

En suivant ce programme intensif, vous aurez l'occasion de travailler sur une variété de projets qui vous permettront de développer vos compétences en React.js et ASP.NET de manière progressive et pratique. N'oubliez pas de régulièrement revisiter et renforcer les concepts clés, et de chercher de l'aide sur des forums ou des communautés en ligne si vous rencontrez des difficultés.