

Titre du TP : Développement d'un site web complet avec Node.js

Objectif : Concevoir et développer un site web dynamique en utilisant Node.js, Express.js et MongoDB.

Durée : Une journée complète (environ 6-8 heures).

Prérequis :

- Connaissance de base en JavaScript (variables, fonctions, objets, etc.).
- Connaissance basique du développement web (HTML, CSS).
- Familiarité avec les bases de données et MongoDB serait un plus.

Plan de la journée :

1. Introduction du projet

- Présentation du projet : créer un site web complet avec un backend Node.js.
- Définition des fonctionnalités attendues pour le site.

2. Configuration de l'environnement

- Installation de Node.js, Express.js et MongoDB.
- Initialisation du projet avec npm et création de la structure de dossiers.

3. Mise en place du serveur

- Création d'un serveur Express.js.
- Configuration du serveur pour gérer les requêtes HTTP.

4. Mise en place du modèle de données

- Définition du schéma de la base de données MongoDB.
- Connexion à la base de données depuis Node.js à l'aide de Mongoose.

5. Création du backend

- Gestion des routes pour les différentes fonctionnalités du site (inscription, connexion, affichage des données, etc.).
- Implémentation des contrôleurs pour gérer les actions associées à chaque route.

6. Création du frontend

- Structuration du frontend en utilisant HTML, CSS et JavaScript.
- Utilisation d'un moteur de templates (par exemple, EJS) pour rendre les pages dynamiques.

7. Gestion des utilisateurs

- Mise en place du système d'inscription et de connexion des utilisateurs.
- Authentification et autorisation des utilisateurs pour accéder à certaines parties du site.

8. Gestion des sessions et des cookies

- Utilisation de sessions et de cookies pour maintenir l'état de l'utilisateur.
- Implémentation de la déconnexion de l'utilisateur.

9. Interaction avec la base de données

- Affichage des données à partir de la base de données sur les pages du site.
- Ajout, modification et suppression de données à partir du site.

10. Intégration d'une API externe

- Intégration d'une API externe pour enrichir les fonctionnalités du site (par exemple, Google Maps pour la localisation).

11. Validation des données

- Validation des données côté serveur pour assurer l'intégrité des informations soumises par les utilisateurs.

12. Gestion des erreurs et des exceptions

- Gestion des erreurs et des exceptions pour améliorer la robustesse de l'application.

13. Déploiement du site

- Options de déploiement pour le site web.
- Déploiement sur un serveur ou une plateforme cloud.

14. Sécurité

- Conseils de sécurité pour les applications Node.js et les sites web.

15. Conclusion

- Récapitulation des concepts clés appris pendant la formation.
- Encouragement à continuer d'apprendre et de développer des projets.