

Formation Vibe coding : prototyper une application complète avec l'IA

Réf. SF34744

Passez d'une idée à un prototype d'application en ligne en deux jours, en pilotant l'IA par la conversation, sans perdre la main sur la qualité et la sécurité de ce qu'elle produit.

2 j
Durée

1 500 € HT
Inter / participant

0
Sessions à venir

Objectifs pédagogiques

- ✓ Choisir l'outil de vibe coding adapté à un projet (app-builder ou agent de code) et identifier les cas où le développement traditionnel reste préférable
- ✓ Concevoir le cadrage d'un produit (parcours, écrans, rôles) et le formaliser en un contexte réutilisable par l'IA
- ✓ Développer par la conversation un prototype d'application web complet, intégrant base de données, authentification et déploiement
- ✓ Débugger et faire évoluer le code généré par le dialogue, en sécurisant chaque itération avec Git
- ✓ Auditer la qualité et la sécurité d'une application générée avant de la partager

Public visé & prérequis

Public visé Tout public

Prérequis- Savoir lire du code, sans forcément l'écrire couramment - Comprendre l'architecture d'une application web : interface, serveur, base de données, API - Avoir déjà utilisé un éditeur de code ; des notions de Git sont un plus

Programme détaillé

Comprendre le vibe coding et sa place dans le développement

- Vibe coding : décrire une intention en langage naturel et confier l'écriture du code à l'IA, en se concentrant sur le produit
- Du copilote à l'agent : différence entre complétion ligne à ligne, assistant conversationnel et agent qui exécute une tâche de bout en bout
- Ce qu'il faut savoir des modèles de langage : rôle du contexte, hallucinations, réponses non déterministes et leurs conséquences sur le code produit
- La boucle de travail : intention, contexte, génération, vérification, puis itération
- Cas d'usage adaptés : prototype, MVP, outil interne, preuve de concept
- Limites : situations où le développement traditionnel reste préférable, et nouveau rôle du développeur, de l'implémentation à la supervision

Activités pratiques :

- Création d'une première petite application web uniquement par conversation, en moins de 30 minutes
- Comparaison d'une demande brute et d'une demande contextualisée sur la même fonctionnalité

Panorama et choix des outils

- App-builders IA (par exemple Lovable, Bolt, v0) : génération d'une application complète dans le navigateur, stack imposée, mise en ligne intégrée
- Agents de code et IDE assistés par IA (par exemple Claude Code, Cursor, Codex) : travail sur un projet local, contrôle complet du code et de la stack
- Critères de choix : niveau de contrôle sur le code, export et propriété du code, stack imposée ou libre, confidentialité des données
- Coûts et quotas : modèles d'abonnement, consommation lors d'un usage intensif, maîtrise du budget
- Configuration et prise en main : création du projet, connexion au dépôt Git, réglages essentiels
- Combiner les familles : démarrer dans un app-builder, poursuivre dans un agent de code

Activités pratiques :

- Réalisation de la même mini-application avec un app-builder et avec un agent de code, puis comparaison des résultats et du code obtenu
- Grille de choix d'outil appliquée à trois projets types

Cadrer le produit et construire le contexte pour l'IA

- Concevoir avant de prompter : profils utilisateurs, parcours clés, cas limites et rôles d'accès
- Structurer l'interface : liste des écrans, navigation, hiérarchie de l'information, pages de connexion et de profil
- Décrire le « quoi » plutôt que le « comment » : exprimer le besoin, les contraintes métier et le résultat attendu
- Références visuelles et exemples : maquettes, captures et données d'exemple pour guider la génération
- Fichier d'instructions du projet (AGENTS.md, règles, spécification produit) : un contexte persistant que l'outil relit à chaque demande
- Dialogue itératif : une demande à la fois, validation du résultat avant d'enchaîner

Activités pratiques :

- Cadrage du projet fil rouge : parcours utilisateurs, écrans et rôles
- Rédaction du fichier d'instructions du projet et comparaison de générations avec et sans ce contexte
- Construction d'une page d'accueil avec formulaire de contact par itérations successives

Construire le prototype par itérations

- Découper le produit en fonctionnalités : ordre de construction, périmètre de chaque itération
- Générer l'architecture initiale : structure du projet, composants, routage entre les écrans
- Construire écran par écran : listes, formulaires, vues détaillées, états de chargement, d'erreur et vide
- Données fictives d'abord : simuler les données et les rôles pour valider les parcours avant de brancher le back-end
- Tester en continu : vérifier chaque incrément dans le navigateur avant de passer au suivant
- Git comme filet de sécurité : commits fréquents, points de restauration et retour arrière après une génération ratée

Activités pratiques :

- Génération du squelette de l'application fil rouge et des premiers écrans à partir du cadrage
- Ajout progressif des fonctionnalités, avec un commit à chaque étape validée
- Retour arrière volontaire après une génération qui casse une fonctionnalité existante

Déboguer et faire évoluer le code par la conversation

- Décrire une erreur à l'IA : message, comportement attendu et observé, étapes pour la reproduire
- Techniques de débogage par le dialogue : faire formuler des hypothèses, ajouter des traces, isoler le problème
- Demander des explications : faire commenter le code généré pour le comprendre avant de le modifier
- Affiner sans tout régénérer : demandes ciblées sur un fichier ou un comportement précis
- Régressions et effets de bord : les détecter, les signaler à l'IA, revenir à un état stable
- Savoir reprendre la main : quand une correction manuelle est plus rapide qu'une nouvelle demande

Activités pratiques :

- Correction d'une application volontairement buggée uniquement par le dialogue
- Analyse d'une régression introduite par une génération et retour à un état fonctionnel

Données, authentification et mise en ligne

- Plateforme de backend (par exemple Supabase, Firebase) ou base PostgreSQL : modèle de données, tables et relations générées à partir du cadrage
- Remplacer les données fictives : branchement des écrans sur les données réelles et vérification des parcours
- Authentification : inscription, connexion, gestion de session
- Rôles et contrôle d'accès : appliquer les droits définis au cadrage, côté interface et côté données
- Connexion à des API externes : clés d'API, appels côté serveur, gestion des erreurs
- Intégrations possibles selon le projet : paiement en ligne, fonctionnalités temps réel
- Déploiement (par exemple Vercel, Netlify) : mise en ligne, variables d'environnement, déploiements successifs

Activités pratiques :

- Ajout d'une base de données et de l'authentification à l'application fil rouge
- Tests d'accès avec plusieurs profils utilisateurs
- Mise en ligne de l'application et partage de l'URL

Qualité, sécurité et bon usage du vibe coding

- Évaluer le code généré : lisibilité, duplication, structure du projet, dépendances ajoutées
- Refactorisation et optimisation à la demande : améliorer le code sans changer le comportement
- Tests automatisés générés par le dialogue, et documentation du projet
- Failles typiques des applications générées : clés d'API et secrets exposés côté navigateur, règles d'accès à la base mal configurées, dépendances inexistantes ou malveillantes suggérées par l'IA
- Propriété intellectuelle et licences du code généré, confidentialité des données transmises aux outils
- Travailler en équipe : conventions de projet, fichier d'instructions partagé, revue du code généré
- Vibe coding ou développement traditionnel : critères de décision selon la criticité et la durée de vie du projet, et veille sur des outils qui évoluent vite

Activités pratiques :

- Audit de l'application fil rouge à l'aide d'une checklist qualité et sécurité, puis corrections par le dialogue
- Génération d'une suite de tests sur les parcours principaux
- Présentation des prototypes et revue collective des choix et des points de vigilance

Pourquoi choisir Sparks Formation ?

20 ans d'expertise IT, des formateurs experts et des sessions garanties contre l'annulation.

20 ans
d'expertise

1 068+
formations

11 178
apprenants formés/an

+67
Net Promoter Score

Avis des participants

4,9/5 — 39 avis

Maxime V. — session très agréable, dense mais toujours agréable. formateur très à l'écoute, prends le temps de répondre aux questions. on apprend l'utilisation des fonctions mais aussi les processus mis...

Cécile L. — Très bon instructeur qui explique toujours en montrant du code, aucune slide et une présentation vivante ce qui permet de rester intéressé et concentré tout au long !...

Ils nous font confiance

Cette année notre filière Développement Web & Fullstack a accompagné plus de 111 entreprises dont :

ARPEGE APSIDE TOP CEGEDIM CONSERVATOIRE NATIONAL DES ARTS ET METIERS ECONOCOM APPS, CLOUD & DATA ORANGE BUSINESS
ORANGE BUSINESS SERVICES SA SOPRA STERIA ADVENTIEL AGATE AGENCE ALPINE DES TERRITOIRES +101 autres

Financement & certifications

Cette formation est éligible aux financements OPCO et FNE-Formation. Nos équipes vous accompagnent dans le montage de votre dossier.



Réservez votre place dès maintenant

20 ans d'expertise · 1 000+ formations · NPS +67 · réponse sous 24h ouvrées

0 805 950 800

demande@sparks-formation.com