FRENCHTECH FACTORY

Construire une application grâce au Vibe Coding

🎓 Certification au RS de France Compétences
💻 E-learning et classe virtuelle
🕓 25 heures



Cette formation prépare à l'examen final de la **certification « Construire une application grâce au Vibe Coding »** enregistrée à France Compétences sous le numéro **RS7554**. L'examen final, à distance, est obligatoire. Le certificateur est IQ Education (BRAINEE). À l'issue de la formation, le stagiaire s'engage à passer l'examen préparant à la certification.

Cette formation professionnelle de par son caractère certifiant peut être prise en charge par le CPF. Toute action ne rentrant pas dans le cadre de l'article L6313-1 du code du travail ne rentre pas dans ce dispositif de prise en charge.

1. Objectifs pédagogiques

- Comprendre les bases du développement d'un produit Tech et comprendre le changement de paradigme à l'ère de l'IA.
- Concevoir un User Flow et identifier la core feature de son produit
- Apprendre à manipuler l'IA générative et les agents IA pour développer
- Comprendre les grands concepts des agents de code : prompting, itérations, rollback, deploy
- Apprendre à construire les fonctionnalités et intégrer des APIs / MCP (Model Context Protocol)
- Apprendre à lancer une version avancée et fonctionnelle de son produit

2. Public cible et prérequis

Public cible :

- Professionnels non techniques intervenant dans l'innovation numérique, product managers, designers, responsables innovation, entrepreneurs, consultants, marketeurs.

Prérequis :

- Aucun diplôme ou certification exigés.
- Maîtriser les usages numériques de base
- Disposer d'un accès à un outil d'IA générative (exemple : ChatGPT, Claude ou équivalent)
- Disposer d'un accès à un outil de développement assisté par IA (exemple : Claude Code, Codex ou équivalent)
- Être en mesure de travailler en autonomie sur un projet de conception d'application

3. Intérêt de la formation

Dans les contextes d'innovation, de produit, de marketing et d'entrepreneuriat, transformer rapidement une idée en application fonctionnelle devient une nécessité. Cette formation permet aux participants de concevoir, prototyper et déployer eux-mêmes une application grâce à l'IA générative, sans dépendre de développeurs. Ils apprennent à cadrer un produit, à piloter les outils de génération de code, à connecter des services externes, à mettre en ligne une application sécurisée et à adopter un usage responsable de l'IA.

4. Durée

25 heures : 18 heures d'e-learning, puis une journée de 7 heures en classe virtuelle synchrone à la fin du parcours.

5. Modalités de formation

La formation se déroule entièrement à distance. Les 18 heures d'e-learning sont suivies en autonomie sur une plateforme dédiée, à son rythme. La journée synchrone de 7 heures est animée en classe virtuelle via une plateforme de visioconférence. Les supports de cours et les exercices pratiques sont accessibles à tout moment.

Fil rouge : un projet guidé commun, la même application pour tous les apprenants, construite pas à pas dans les vidéos. Il est présenté comme un projet d'entraînement, distinct de l'épreuve de certification, qui porte sur un sujet imposé par le certificateur.

Outils utilisés : un agent de code principal, Claude Code ou Codex, utilisé dans sa version web puis en local ; Git et GitHub ; React/TypeScript et Supabase pour l'interface, les données et l'authentification ; Vercel pour la mise en ligne.

6. Compétences visées

À l'issue de cette formation, les participants seront capables de :

- Cadrer un produit numérique en traduisant une idée en parcours utilisateur et fonctionnalités structurées afin de produire des instructions exploitables par une intelligence artificielle générative à l'aide de techniques d'ingénierie des instructions (prompt engineering)
- Générer et itérer un prototype fonctionnel par cycles successifs d'interaction avec l'intelligence artificielle générative afin d'obtenir un produit stable et cohérent en appliquant des correctifs, retours en arrière (rollbacks) et ajustements guidés
- Configurer un environnement de développement local à partir d'un projet généré via une interface web d'intelligence artificielle générative afin d'assurer la continuité et la reprise du code en synchronisant les modifications via un système de gestion de versions et en structurant le projet selon les standards de développement
- Interfacer le produit avec des services externes (API, authentification, bases de données, MCP…) afin d'assurer son fonctionnement complet en connectant les briques techniques nécessaires
- Déployer le produit sur un environnement de production afin de le rendre accessible aux utilisateurs en respectant les principes fondamentaux de sécurité et en documentant la démarche technique
- Maintenir une veille sur l'évolution des outils d'intelligence artificielle générative appliqués au développement d'applications (Vibe Coding) afin d'adapter sa démarche aux évolutions techniques et réglementaires en évaluant les outils émergents, les bonnes pratiques et les usages responsables

7. Méthodes et moyens pédagogiques

- **E-learning :** vidéos, apports théoriques et démonstrations d'outils, suivis en autonomie.
- **Projet guidé :** construction pas à pas d'une application d'entraînement commune, du cadrage jusqu'à la mise en ligne.
- **Exercices pratiques :** un livrable à produire à chaque module e-learning (dossier de cadrage, prototype, dépôt GitHub, application en ligne).
- **Classe virtuelle :** une journée synchrone pour reprendre les compétences abordées, avec deux ateliers de présentation et de correction des projets.
- **Évaluation des acquis :** un quiz à l'issue de chaque module, un QCM blanc, une mise en situation chronométrée et une auto-évaluation avec la grille officielle.
- **Évaluation sommative :** évaluation par le certificateur RS7554 (QCM et mise en situation professionnelle).

8. Politique handicap, modalités et délais d'accès

Politique handicap :

- En cas de handicap, contactez notre référent handicap : contact@french-techfactory.com

Modalités d'inscription :

- Inscription sur entretien, analyse de votre projet et test de positionnement.
- Les candidatures sont ouvertes tout au long de l'année.

Délai d'accès :

- Les accès à la formation sont envoyés après expiration de votre délai de rétractation légal.

9. Tarif

Le prix de cette formation est de 1 650€ HT pour 25 heures de formation. Ce tarif inclut l'accès à la plateforme e-learning, des supports de formation, une journée en classe virtuelle avec des ateliers pratiques et des évaluations pour garantir la bonne assimilation des enseignements.
**TVA non applicable de l'article 261.4.4 CGI.*

10. Contenu de la formation

Le parcours comprend 10 chapitres d'e-learning (18 h), puis une journée de classe virtuelle (7 h) en fin de parcours.

Partie 1 : E-learning (18 h)

Chapitre 1 : Panorama des outils et prise en main d'un agent de code (1 h)

- **1.1 Découvrir le projet et les objectifs :** observer l'application cible, repérer les six compétences et les livrables de certification
- **1.2 Panorama des outils du vibe coding :** plateformes web de génération (Lovable, Bolt.new, Replit…), agents de code (Claude Code, Codex, Cursor…), assistants conversationnels, services de données et d'authentification (Supabase, Firebase…), hébergeurs (Vercel, Netlify, Render…), gestion de versions (GitHub). Rôle de chaque famille, forces, limites et coûts
- **1.3 Nos choix pour le parcours et pourquoi :** agent principal utilisé en version web puis en local, GitHub, React/TypeScript, Supabase, Vercel ; ce que ces choix permettent, leurs limites et ce qui changerait avec d'autres outils
- **1.4 Modèle, assistant et agent :** comprendre la boucle de l'agent : lire le contexte, proposer, agir, vérifier

- **1.5 Contexte, limites et permissions :** hallucinations, limites de contexte, consommation et périmètre autorisé ; identifier les informations à ne pas transmettre
- **1.6 Première modification guidée :** demander une petite évolution dans un environnement d'essai, comparer résultat attendu et résultat observé
- **Production :** première modification vérifiée, journal de bord ouvert et fiche « panorama des outils », première version du tableau comparatif
- **Évaluation :** quiz court et correction commentée ; vérification du livrable selon les critères annoncés.

Chapitre 2 : Cadrer son application (2 h)

- **2.1 Problème, objectifs et utilisateurs :** décrire le besoin à partir d'un brief, rédiger les objectifs produit, les profils utilisateurs et le scénario d'usage principal
- **2.2 Priorités et contraintes :** classer les besoins fonctionnels par priorité (indispensable, souhaitable, reporté) et les justifier par des scénarios d'usage ; expliciter les contraintes
- **2.3 Parcours utilisateur :** dessiner le parcours par un schéma ou un diagramme, associer une fonctionnalité à chaque étape et vérifier qu'aucune étape ne crée de rupture fonctionnelle
- **2.4 Périmètre et critères d'acceptation :** définir une première version réalisable et des résultats observables
- **Production :** dossier de cadrage v1 : objectifs, profils, priorités, scénarios, contraintes et schéma du parcours
- **Évaluation :** quiz court et correction commentée ; vérification du livrable selon les critères annoncés.

Chapitre 3 : Préparer la stack et configurer l'agent (1 h 30)

- **3.1 Architecture et choix de stack :** distinguer interface, serveur, données, authentification et services externes ; formaliser la stack, l'architecture cible et les contraintes techniques
- **3.2 Comptes, dépôt et outils locaux :** préparer les comptes, créer le dépôt GitHub du projet, installer Git, l'éditeur et Node.js
- **3.3 Instructions persistantes :** rédiger les règles de génération et les commandes de contrôle dans AGENTS.md ou CLAUDE.md, adaptées à l'agent choisi (structure, conventions, vérifications)
- **3.4 Extensions utiles :** activer la configuration guidée FTF, n'ajouter que les skills et plugins utiles, examiner leur provenance et leurs permissions
- **Production :** cadre technique, dépôt GitHub créé, fichier d'instructions adapté et environnement préparé
- **Évaluation :** quiz court et correction commentée ; vérification du livrable selon les critères annoncés.

Chapitre 4 : Générer le premier prototype sur le web et le reprendre en local (2 h)

- **4.1 Définir l'objectif :** choisir la fonctionnalité centrale, son périmètre et les conditions de réussite
- **4.2 Rédiger les instructions initiales :** rôle, objectif, contexte, contraintes techniques et de style, critères de validation ; faire proposer un plan, le relire et corriger les écarts au cadrage
- **4.3 Générer la première version via l'interface web de l'agent :** connecter l'interface web (Claude Code sur le web ou Codex web) au dépôt GitHub, générer le prototype, puis valider la pull request
- **4.4 Cloner le projet et l'exécuter en local :** cloner le dépôt depuis GitHub, installer les dépendances, configurer et lancer le projet ; comparer avec la version web
- **4.5 Premier cycle de synchronisation :** faire une modification en local avec l'agent, puis commit, push et pull ; consigner la continuité web → GitHub → local
- **Production :** objectif, prompt initial, plan examiné, prototype, preuve de clonage et premier commit depuis le poste local
- **Évaluation :** quiz court et correction commentée ; vérification du livrable selon les critères annoncés.

Chapitre 5 : Itérer, tester et corriger (2 h)

- **5.1 Découper une évolution :** transformer une fonctionnalité en tâches courtes avec critères d'acceptation, sans modifier inutilement le reste
- **5.2 Fournir le bon contexte :** fichiers, captures, messages d'erreur et comportement attendu ; adapter le prompt, ajouter du contexte ou décomposer la tâche
- **5.3 Résoudre une erreur complexe :** reproduire, diagnostiquer, corriger et expliquer une erreur ou une hallucination ; renseigner la fiche erreur
- **5.4 Vérifier et versionner :** contrôler les régressions ; commit, push, pull et retour arrière (rollback) ; composants réutilisables et conventions de nommage
- **Production :** évolution fonctionnelle, historique de versions et fiche « erreur complexe »
- **Évaluation :** quiz court et correction commentée ; vérification du livrable selon les critères annoncés.

Chapitre 6 : Ajouter les données, les comptes et appliquer le RGPD (2 h)

- **6.1 Modèle de données et persistance :** passer des données fictives à un schéma simple et persistant, en ne collectant que les données nécessaires
- **6.2 Comptes et routes privées :** mettre en place inscription, connexion et déconnexion ; tester l'accès aux fonctions privées avant et après authentification

- **6.3 Droits d'accès et RLS :** définir la propriété des données, appliquer les politiques RLS et documenter les droits de lecture et de modification
- **6.4 Vérification avec deux comptes :** vérifier que A ne lit ni ne modifie les données de B, au-delà de la seule interface
- **6.5 Appliquer le RGPD dans l'application :** politique de confidentialité et information des utilisateurs, collecte limitée, suppression du compte et des données
- **Production :** authentification, persistance, politiques RLS, preuves des tests d'accès et mesures RGPD en place
- **Évaluation :** quiz court et correction commentée ; vérification du livrable selon les critères annoncés.

Chapitre 7 : Connecter une API externe (1 h 30)

- **7.1 Choisir un service utile :** comprendre requête, réponse et dépendance externe ; identifier une API pertinente pour le parcours
- **7.2 Réaliser l'intégration :** faire développer l'appel, vérifier le traitement des réponses sur un scénario nominal
- **7.3 Protéger les secrets :** distinguer données publiques et secrets, garder les clés côté serveur, vérifier le dépôt et les échanges du navigateur
- **7.4 Gérer et documenter les erreurs :** tester réponse vide, indisponibilité et limites d'usage ; lister les services et le rôle de chacun dans le dossier de cadrage
- **Production :** API opérationnelle, résultats des tests et architecture des services actualisée
- **Évaluation :** quiz court et correction commentée ; vérification du livrable selon les critères annoncés.

Chapitre 8 : Approfondir les skills, MCP et sous-agents (1 h)

- **8.1 Choisir le bon mécanisme :** distinguer règles persistantes, skill, plugin, MCP et sous-agent ; justifier le mécanisme retenu pour une tâche
- **8.2 Adapter une skill de recette :** transformer une procédure récurrente en instructions réutilisables et vérifier son application
- **8.3 Connecter un MCP pertinent :** examiner capacité, provenance, données accessibles et permissions ; en expliquer l'apport
- **8.4 Déléguer une revue ciblée :** définir objectif, périmètre et résultat attendu du sous-agent ; retenir ou rejeter ses conclusions, puis vérifier les corrections
- **Production :** procédure adaptée, choix d'extension argumenté et rapport de revue avec décisions
- **Évaluation :** quiz court et correction commentée ; vérification du livrable selon les critères annoncés.

Chapitre 9 : Sécuriser, déployer et transmettre (2 h)

- **9.1 Recette fonctionnelle et sécurité :** vérifier le parcours et l'absence d'erreur bloquante ; contrôler les cinq indicateurs de sécurité (HTTPS, secrets, routes privées, données propres à chaque utilisateur, RLS)
- **9.2 Déploiement en production :** identifier la plateforme, les variables d'environnement et les étapes ; rédiger une procédure reproductible
- **9.3 Vérifier la version publique :** rejouer depuis l'URL de production, sur ordinateur et sur mobile, les fonctionnalités clés, l'authentification, l'API et les contrôles d'accès
- **9.4 Transmettre le projet :** compléter le README (installation, configuration, utilisation), l'arborescence (code, documentation, configuration), les fichiers de configuration et les dépendances ; vérifier la reprise en clonant le dépôt sur un poste neuf
- **Production :** URL publique, recette remplie, procédure de déploiement et dépôt GitHub documenté
- **Évaluation :** quiz court et correction commentée ; vérification du livrable selon les critères annoncés.

Chapitre 10 : Justifier sa démarche et préparer la certification (3 h)

- **10.1 Comparer les outils et organiser sa veille :** finaliser le tableau comparatif amorcé au chapitre 1 (fonctionnalités, coûts, performances, limites dont contexte et dépréciation) ; construire une routine de veille
- **10.2 Données, éthique et accessibilité :** documenter les mesures RGPD, la confidentialité, le chiffrement, les droits, l'usage des données par l'agent, la transparence, les biais, l'accessibilité et l'impact environnemental
- **10.3.1 Assembler et s'auto-évaluer :** préparer les livrables (dossier, application, dépôt, vidéo) et l'attestation sur l'honneur d'authenticité, vérifier la grille officielle
- **10.3.2 Vidéo** : démonstration, choix techniques et récit de l'erreur complexe
- **10.4 QCM blanc :** 50 questions, dans les conditions de l'examen
- **10.5 Mise en situation chronométrée** : sur un sujet type inconnu, réaliser en 1 h une version réduite de l'épreuve (cadrage express, prompt initial, première version, push sur GitHub et mise en ligne), puis comparer le résultat aux critères de la grille
- **Production :** dossier final, tableau comparatif, routine de veille, vidéo d'entraînement, auto-évaluation, résultat du QCM blanc et de la mise en situation chronométrée
- **Évaluation :** quiz court et correction commentée ; vérification du livrable selon les critères annoncés.

Partie 2 : Journée synchrone en classe virtuelle (7 h)

Session 1 : Du besoin au prototype (3 h 30)

- Cadrer son produit : objectifs et profils utilisateurs, priorités justifiées par des scénarios, contraintes, parcours utilisateur, cadre technique et fichiers d'instructions de l'agent, prompt initial avec critères de validation
- Générer et itérer : objectif, plan, cycles d'itération, stratégies de débogage, hallucinations, retour arrière, stabilité de l'application
- Environnement local : reprendre en local le projet généré sur le web via GitHub, synchroniser ses modifications, et organiser le code pour qu'un tiers puisse le reprendre (arborescence, README, composants, conventions de nommage)
- **Atelier 1 :** chaque apprenant partage son écran et présente son dossier de cadrage, son fichier d'instructions, sa fiche « erreur complexe » et son dépôt GitHub (historique et README) ; correction par le formateur avec la grille officielle

Session 2 : Du prototype à la production (3 h 30)

- Connecter des services externes : données, authentification, API, MCP, secrets gardés côté serveur, rôle de chaque service
- Sécuriser et déployer : routes privées, RLS et test à deux comptes, RGPD appliqué dans l'application, HTTPS, déploiement et vérification de la version publique
- Veille et usage responsable : comparaison des outils, limites, AI Act, éthique (biais, impact environnemental, accessibilité)
- **Atelier 2 :** chaque apprenant présente son application en ligne (démonstration, API et authentification, sécurité, mesures RGPD, choix des outils et section éthique) ; correction par le formateur avec la grille officielle
- Présentation du déroulé de l'examen (convocation, QCM, épreuve projet de 5 h dans le mois qui suit, rattrapage)

11. Modalités d'évaluation des acquis et validation des résultats

- **Amont** : test de positionnement et vérification des prérequis
- **Évaluation formative** : quiz et corrections commentées à chaque chapitre, livrables, journal de bord, revues en classe virtuelle, recette, auto-évaluation avec la grille officielle, QCM blanc avec durée limitée, mise en situation chronométrée et plan de remédiation
- **Satisfaction** : À chaud, puis à froid 1 mois après la formation
- **Évaluation sommative** : Évaluation par le certificateur RS7554, décrite ci-dessous
- **Validation** : Remise d'une attestation individuelle

12. Modalités d'évaluation finale (évaluation sommative)

L'évaluation est organisée par le certificateur, séparément des évaluations pédagogiques de French Tech Factory. Les apprenants sont convoqués à la fin de la formation. Les épreuves se déroulent à distance, sous surveillance (webcam et partage d'écran).

Épreuve 1 : QCM de connaissances

- 50 questions couvrant l'IA générative appliquée au développement, le prompt engineering, la sécurité, le déploiement, les outils émergents et les bonnes pratiques

Épreuve 2 : Mise en situation professionnelle

- Le candidat conçoit, développe et déploie une application web fonctionnelle répondant à un besoin utilisateur identifié, sur l'un des deux sujets proposés par le certificateur, avec l'outil d'IA de son choix
- L'application intègre au minimum une connexion API externe et une fonctionnalité d'authentification ou de persistance des données
- Durée : réalisation en 5 h, en une session surveillée à distance (webcam et partage d'écran), au plus tard dans le mois qui suit la réussite au QCM
- Livrables attendus : dossier de cadrage produit, application web déployée et accessible en ligne, code source versionné sur GitHub, vidéo de présentation et attestation sur l'honneur d'authenticité du travail

Conditions d'obtention : la certification est délivrée si les deux épreuves sont réussies.

13. Moyens matériels

Plateforme numérique dédiée :

- L'e-learning est accessible sur une plateforme en ligne. La journée en classe virtuelle se déroule sur une plateforme de visioconférence, avec des outils interactifs (partage d'écran, chat) permettant de favoriser l'échange et l'interaction.

Supports pédagogiques :

- Les vidéos, supports, modèles (cadrage, journal de bord, recette) et exercices sont accessibles à chaque participant pour lui permettre de revenir sur les contenus abordés au cours de la formation.

14. Modalités d'assistance technique

Afin de garantir une expérience d’apprentissage optimale, un support technique est disponible pour résoudre toute difficulté rencontrée lors de la connexion à la formation, de l’utilisation de la plateforme e-learning ou de la plateforme de vidéoconférence. En cas de problème de connexion ou de difficulté technique, le formateur pourra être contacté par mail, SMS ou WhatsApp. Cette assistance permettra de résoudre rapidement toute difficulté liée à l’accès aux sessions ou à l’utilisation des outils en ligne, assurant ainsi la continuité de la formation.

15. Modalités d'assistance pédagogique

Un accompagnement pédagogique est également mis en place pour répondre aux questions relatives au contenu de la formation. En cas de besoin d’éclaircissements ou de précisions sur les concepts abordés, le formateur sera disponible par mail, SMS ou WhatsApp. Ce dispositif vise à garantir un suivi personnalisé et à s’assurer que les enseignements sont bien compris et intégrés dans le cadre professionnel des apprenants.