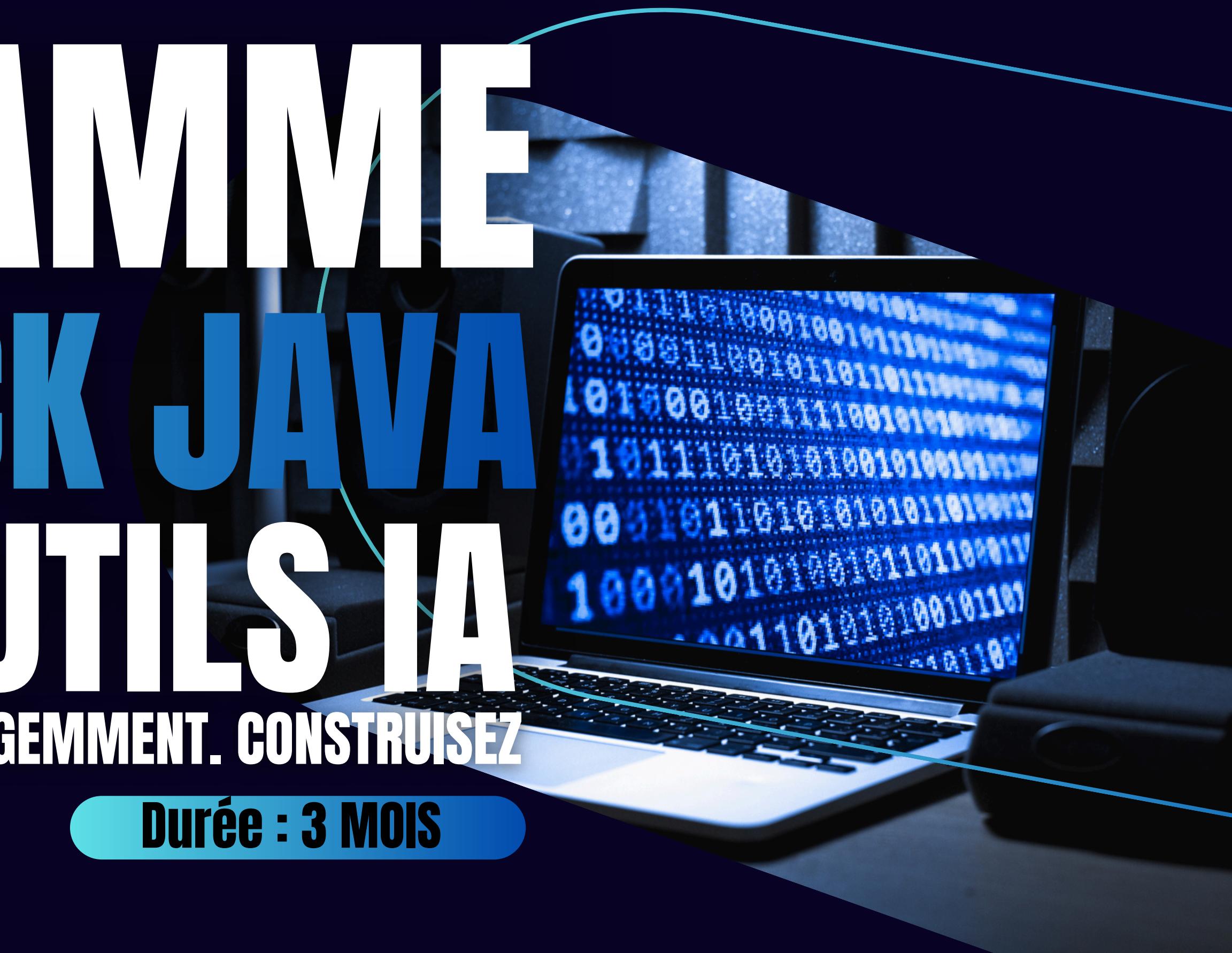
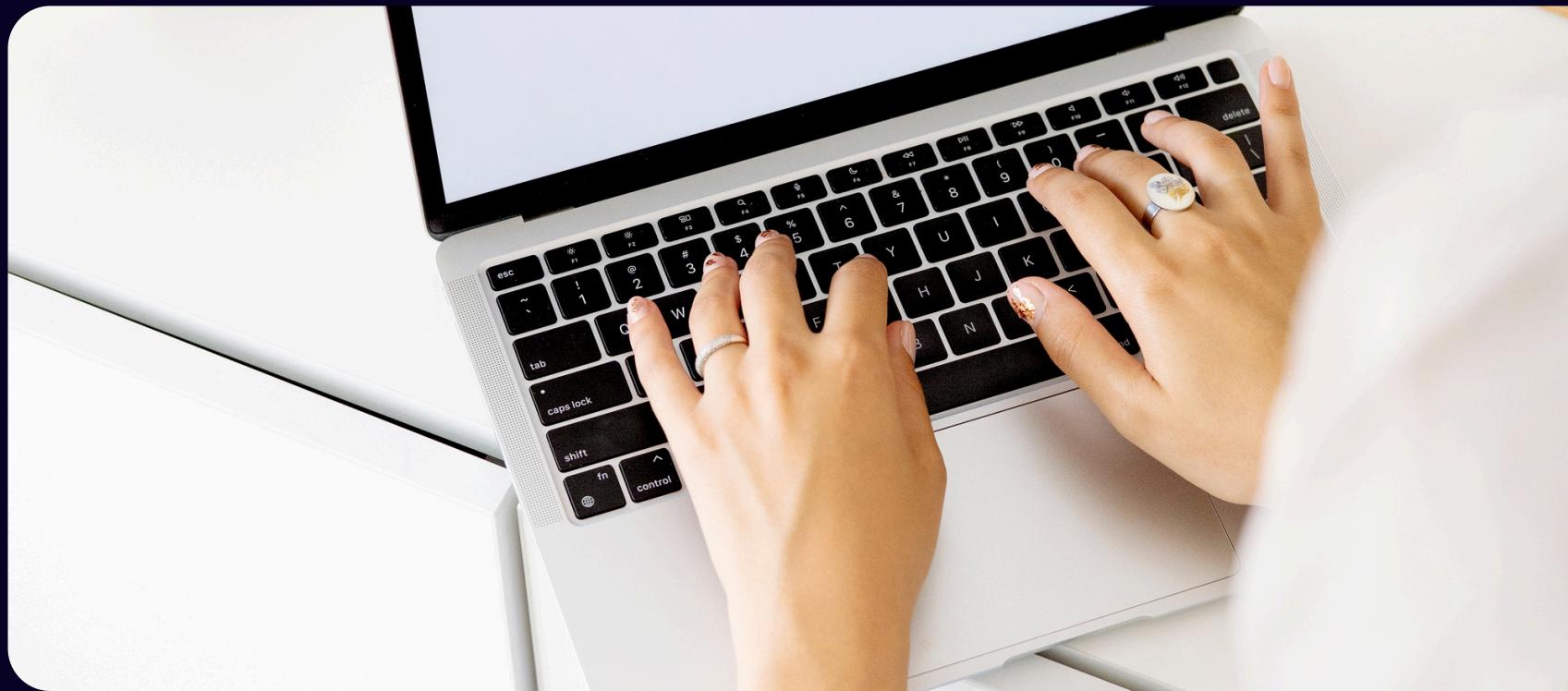


PROGRAMME FULLSTACK JAVA AVEC OUTILS IA

“CODEZ PLUS INTELLIGEMMENT. CONSTRUISEZ
L’AVENIR AVEC L’IA

Durée : 3 MOIS





HTML

- Apprendre les bases Html
- Ecrire HTML sémantique
- Formulaires et Validations
- Accessibilité
- Bases du référencement

CSS

- Apprendre les bases CSS
- Faire les mises en page
- Créer une conception Web réactive

BOOTSTRAP

- Basic Layout
- Component

ANGULAR

- **Introduction Angular**

- Présentation d'Angular
- ECMAScript 6
- Découvrir TypeScript

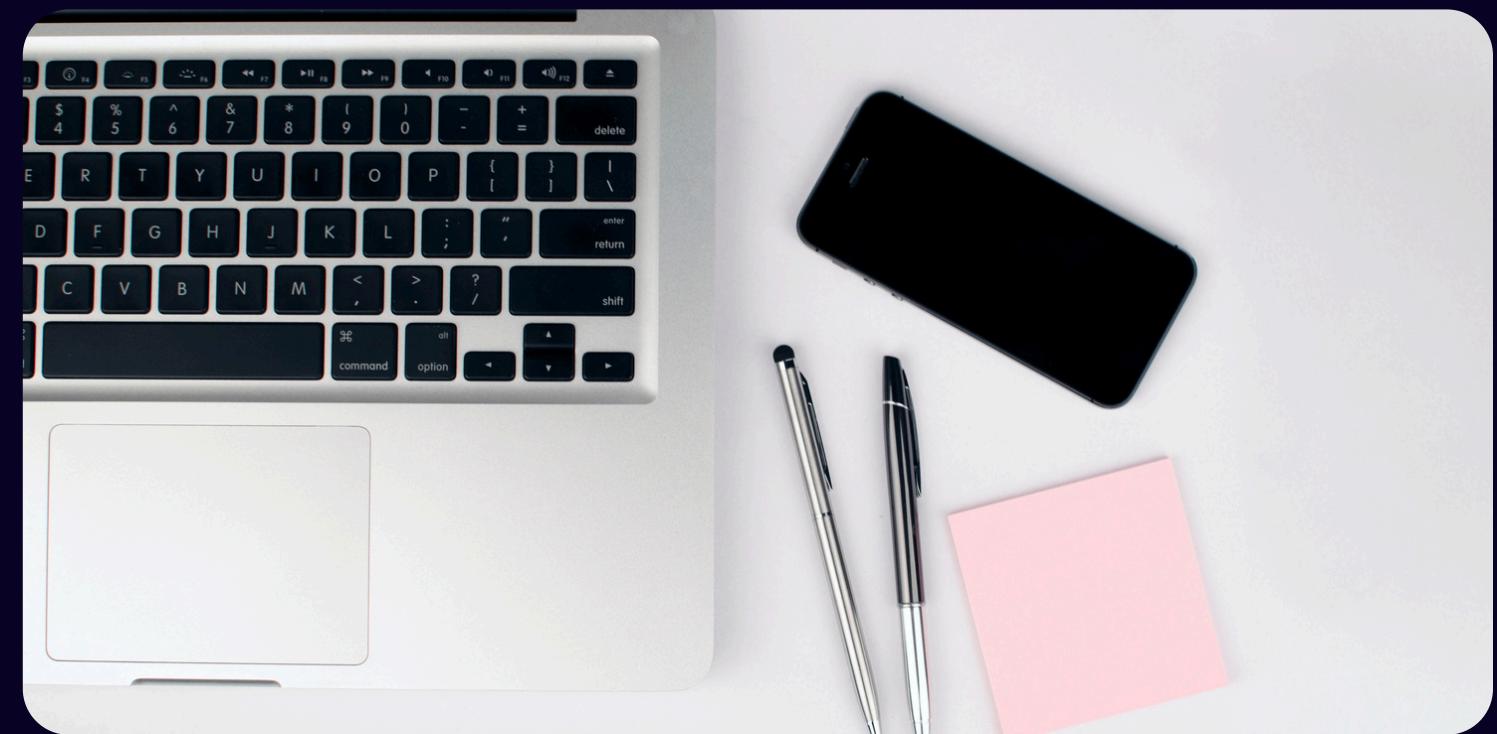
- **Premier pas : Comprendre la structure du framework**

- Concepts fondamentaux et vue d'ensemble de l'architecture composants, services, vues...
- Comment est organisée une application Angular ?
- Organisation du code avec les modules : les conteneurs NgModules et l'encapsulation
- Templates, les nouvelles syntaxes Fonctionnement du databinding pour gérer les données dynamiques
- Rôles des pipes et directives
- Principes de l'injection de dépendances

- **Les Outils d'IA pour Angular (IDE & Assistance)**

- *GitHub Copilot**

- Génère du code TypeScript, HTML, CSS
- Aide à créer composants, services, tests unitaires
- Exemple : auto-complétion intelligente en Angular



- **Composants, templates et liaison de données (databinding)**

- Cycle de vie et structure des Angular components
- Démarrer un projet et créer une hiérarchie de composants
- Les mécanismes d'interaction entre composants

Routing et navigation

- Ajouter des styles aux composants
 - L'interpolation, event binding et property binding
 - Le two-way data binding
 - Création de classes
- Création de composants Angular, ajout de styles et mise en oeuvre des différents types de databinding.

- **Routing et navigation**

- Vue d'ensemble du routage Angular
 - Déclarer et configurer des routes et URLs
 - Paramètres des routes
 - Gérer les redirections
- Définition des routes et navigation dans une application.



• **Formulaires**

- Cycle de vie et structure des Angular components
- Démarrer un projet et créer une hiérarchie de composants
- Les mécanismes d'interaction entre composants

Routing et navigation

- Ajouter des styles aux composants
- L'interpolation, event binding et property binding
- Le two-way data binding
- Création de classes Création de composants Angular, ajout de styles et mise en oeuvre des différents types de databinding.

• **Directives et pipes**

- Directives structurelles (ngFor, ngIf et les ngSwitch)
- Directives d'attribut (ngModel, ngStyle, ngClass)
- Construire ses propres directives
- Utiliser les pipes natifs (DatePipe, UppercasePipe, CurrencyPipe...)
- Créer ses propres pipes Utilisation de directives, manipulation des pipes



- **Services et injection de dépendances**

- Créer un service basique
- Injecter les services fournis par Angular
- Retour sur le système d'injection de dépendances Développement et déclarations de services personnalisés.

- **Les Observables et la bibliothèque RxJS**

- Angular et RxJS (Reactive Extensions for JavaScript)
- Comprendre et utiliser les observables
- Les principaux opérateurs
- Gestion des requêtes HTTP avec le module HttpClient
- Utilisation des observables, interactions avec un serveur

JAVA

- Introduction à Java
- Les bases de la POO en JAVA
- L'encapsulation, Héritage, Polymorphisme, Interface
- Java Collections
- Gestion des exceptions





SPRING BOOT

- **Introduction à la formation Spring Boot**

- Présentation générale de Spring
- Spring boot 3, nouveautés et architecture

- **Spring Boot et configuration**

- Structure projet et options pour la configuration
- Familiarisation avec les fichiers application.properties/.yml
- La prise en main de l'IDE Configurations de Spring, création d'une première application en utilisant Spring Boot.

- **Architecture MVC**

- Pattern MVC
- Spring MVC, architecture et composants utiles
- Architecture MVC
- La notion de Repository
- Les retours sur JPA et Hibernate
- La gestion des transactions : notions et mise en oeuvre
- Accès à la base de données avec un exemple CRUD

- **Architecture REST**

- Comprendre le style d'architecture logicielle REST
- Contrôler les codes de retour HTTP Développement et déclarations de services personnalisés

- **Spring Security**

- Vue d'ensemble, quel problème Spring Security résout-il ?
- Sécuriser des routes et configurer l'authentification
- Mise en oeuvre de la sécurité sur l'application web développée précédemment

SPRING AI (SPRING BOOT STARTER FOR AI)

- Projet officiel de Spring
- Permet de se connecter facilement à des modèles IA (OpenAI, HuggingFace, Ollama, Azure OpenAI, etc.)
- Simplifie l'intégration avec des APIs LLM et embeddings



OPENAI API (CHATGPT, GPT-4, EMBEDDINGS)

- Utilisable dans Spring Boot via RestTemplate / WebClient
- Intégration facilitée par Spring AI (moins de code à écrire)

MYSQL

- Introduction aux Bases de Données (BD)
- Introduction à MySQL
- Modélisation de Données
- Langage SQL (Structured Query Language)
- Administration de MySQL
- Transactions et Concurrency
- Indexation et Optimisation

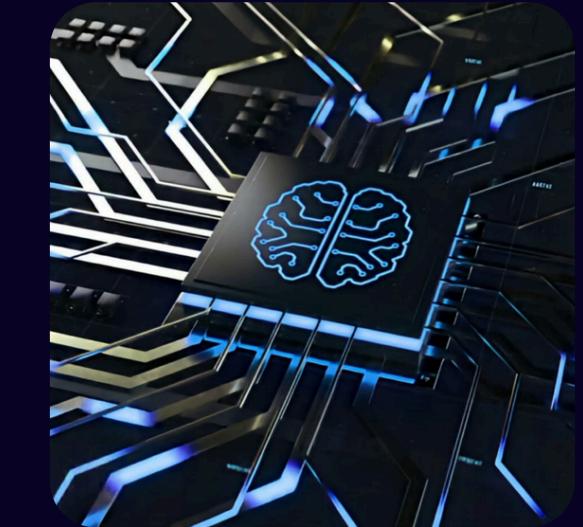
PROJET FINALE : PLATEFORME JOBGATE

- Projet complet pour appliquer toutes les compétences apprises
- Focus sur la création d'une plateforme en utilisant Angular & Spring Boot





POURQUOI NOUS CHOISIR ?



- Technologies utilisées par les leaders du secteur
- Projets 100% concrets directement ajoutables à votre portfolio
- 1 mois de stage garanti pour une expérience professionnelle réelle
- Encadrement premium + accompagnement carrière