

PROGRAMME TEST LOGICIEL

**“Chaque bug détecté vous
rapproche de votre avenir
professionnel”**

Durée : 4 MOIS

À PROPOS DE NOUS

JobGate est la première plateforme d'emploi en Tunisie, leader du recrutement en ligne depuis 2021.

Nous proposons des bootcamps IT intensifs et certifiés à Sousse, basés sur une approche 100 % pratique et orientée marché.

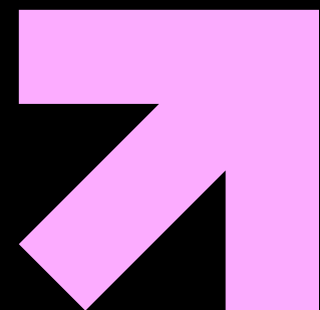
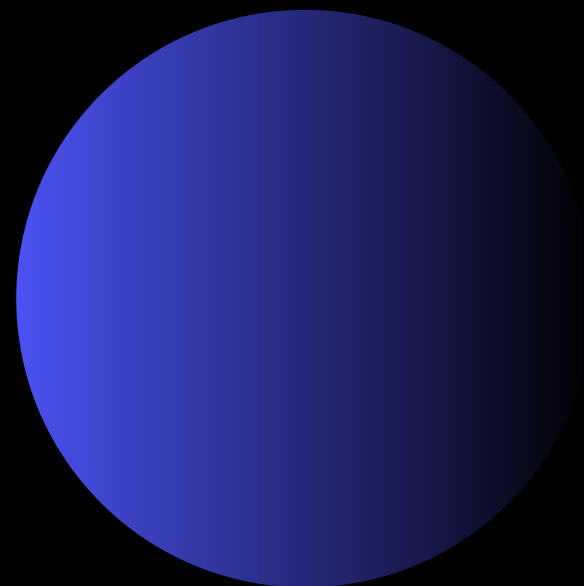
Grâce à un accompagnement personnalisé et à notre réseau d'entreprises partenaires, nous préparons nos candidats à intégrer rapidement le marché du travail.

PUBLIC CIBLE

Tout personne sans pré-requis souhaite reconvertir / changer du domaine

OBJECTIF DU BOOTCAMP

- Acquérir les bases du test logiciel
- Se préparer à la certification ISTQB Foundation Level
- Se préparer à la certification A4Q Foundation Level



- Renforcer les compétences pratiques en test manuel et automatisé
- Développer une culture qualité et esprit critique



MOIS 1 : PRÉPARATION ISTQB

CHAPITRE 1 – PRINCIPES FONDAMENTAUX DU TEST

- Rôles et objectifs du test (pourquoi tester).
- Définitions clés : erreur / défaut (bug) / défaillance, vérification vs validation.
- Les 7 principes du test (p.ex. le test montre la présence de défauts, tests exhaustifs impossibles, début tôt, etc.).
- Processus fondamental du test : planification, conception, exécution, évaluation des critères de sortie, clôture.

CHAPITRE 2 – TESTS TOUT AU LONG DU CYCLE DE VIE LOGICIEL

- Niveaux de test : composant (unité), intégration, système, intégration système, acceptation.
- Modèles de développement et implication du test : V-model, modèle incrémental, Agile/DevOps (tests continus).
- Types de tests : fonctionnels vs non-fonctionnels, tests basés sur la boîte noire/boîte blanche, tests d'acceptation.



CHAPITRE 3 – TEST STATIQUE

- Revue formelle et informelle : inspection, walkthrough, revue technique.
- Processus d'une revue : rôles (auteur, modérateur, relecteur), activités, règles de préparation et de suivi.

CHAPITRE 4 – TECHNIQUES DE CONCEPTION DES TESTS

- Techniques boîte noire : partition d'équivalence, analyse des valeurs limites, tables de décision, transitions d'état, cas d'utilisation.
- Techniques boîte blanche : couverture de déclarations (statement), de décisions/branches — but et limites.
- Techniques basées sur l'expérience : exploratory testing, error guessing, checklist.
- Approche de test basé sur la collaboration

CHAPITRE 5 – GESTION DES TESTS

- Activités de planification et d'estimation des tests (objectifs, périmètre, ressources).
- Stratégies basées sur le risque
- Gestion des anomalies (cycle d'un bug), reporting et métriques (ex. taux d'exécution, taux de passage, densité de défauts).

CHAPITRE 6 – OUTILS DE SUPPORT AU TEST

- Catégories d'outils : gestion des tests, automatisation d'exécution, tests de performance, virtualisation de services, outils de revue/analysis.
- Critères de sélection d'un outil, étapes d'introduction d'un outil, avantages/risques.

MOIS 2 : TEST MANUEL

MODULE 1 - JIRA (GESTION DE PROJETS & SUIVI DES ANOMALIES)

Objectifs :

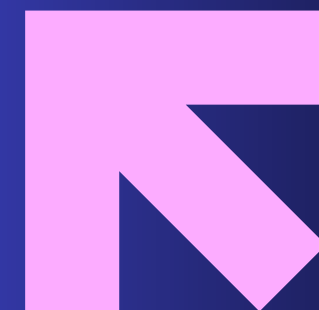
- Comprendre l'utilisation de Jira pour la gestion des projets et des anomalies.
- Gérer backlog, épics, user stories et bugs.

Contenu :

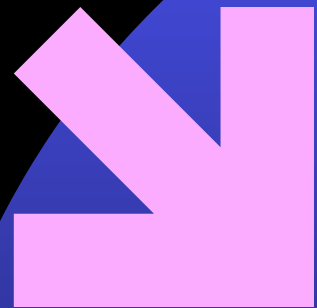
- Présentation de l'interface Jira.
- Création de projets et configuration des workflows.
- Création, suivi et gestion des tickets (stories, tasks, bugs).
- Tableau Scrum.

Activités pratiques :

- Créer un projet Agile.
- Rédiger des user stories et sous-tâches.
- Gérer un bug du signalement à la clôture.



MODULE 2 - SQUASH TM



Objectifs :

- Découvrir Squash TM pour la gestion du référentiel de tests.

Contenu :

- Présentation de l'outil et concepts clés.
- Création de dossiers de test, plans de test et jeux de test.
- Gestion des campagnes d'exécution.
- Suivi de l'avancement via rapports et tableaux de bord.

Activités pratiques :

- Créer un cas de test fonctionnel.
- Exécuter une campagne et générer un rapport.



MODULE 3 - TESTLINK

Objectifs :

- Utiliser TestLink pour planifier et suivre les tests.

Contenu :

- Interface et terminologie (test project, test suite, test case).
- Création de cas de test et planification des exécutions.
- Suivi des résultats et reporting.

Activités pratiques :

- Créer un plan de test pour une application web.
- Associer des cas de test à une exécution.

MODULE 4 - XRAY

Objectifs :

- Intégrer la gestion des tests directement dans Jira via Xray.

Contenu :

- Concepts : test, test set, test execution, test plan.
- Traçabilité exigences :flèche_bidirectionnelle: tests :flèche_bidirectionnelle: anomalies.
- Rapports et dashboards Jira/Xray.

Activités pratiques :

- Créer des cas de test manuels dans Xray.
- Suivre une campagne d'exécution.

MODULE 5 - ZEPHYR

Objectifs :

- Gérer les tests depuis Jira grâce à Zephyr.

Contenu :

- Présentation de Zephyr Essentiel.
- Création de cycles et plans de test.
- Exécution et suivi des résultats.
- Génération de rapports de traçabilité.

Activités pratiques :

- Créer un cycle de test lié à des user stories.
- Exécuter et documenter un test.

MODULE 6 - POSTMAN (TESTS D'API)

Objectifs :

- Automatiser et documenter les tests d'API REST.

Contenu :

- Envoi de requêtes GET/POST/PUT/DELETE.
- Gestion des collections et variables.
- Scripts de tests automatisés en JavaScript.
- Intégration avec CI/CD (Newman).

Activités pratiques :

- Tester une API publique
- Créer une collection avec assertions automatisées.

MODULE 7 - JMETER (TESTS DE PERFORMANCE)

Objectifs :

- Effectuer des tests de charge et de performance.

Contenu :

- Présentation de l'interface JMeter.
- Création de plans de test (test plan, thread group, samplers).
- Tests HTTP, API et bases de données.
- Analyse des résultats (graphes, rapports HTML).

Activités pratiques :

- Lancer un test de charge simple sur une API.
- Interpréter les résultats et identifier les goulots d'étranglement.

MOIS 3 : TEST AUTOMATISÉ

*Soit SELENIUM + IA

1. INTRODUCTION À L'AUTOMATISATION DES TESTS

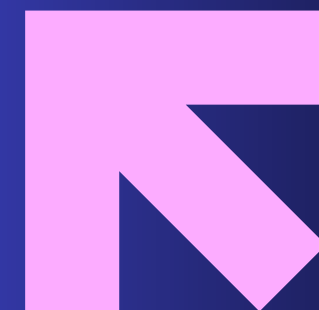
- Comprendre les objectifs de l'automatisation.
- Différence entre tests manuels et tests automatisés.
- Présentation des outils les plus utilisés : Selenium, JUnit/TestNG, Jenkins, GitHub, etc

2. ENVIRONNEMENT DE TRAVAIL

- Installation de Java JDK, IDE (Eclipse/IntelliJ).
- Présentation de Maven et gestion des dépendances

3. INTRODUCTION AUX BASES DE JAVA

- Syntaxe de base : variables, types, conditions, boucles.
- Fonctions, classes, objets et concepts de POO (programmation orientée objet).
- Utilisation des collections (ArrayList, HashMap...).
- Gestion des exceptions et introduction à la structure d'un programme Java.



4. STRUCTURE D'UN PROJET SELENIUM AVEC JAVA

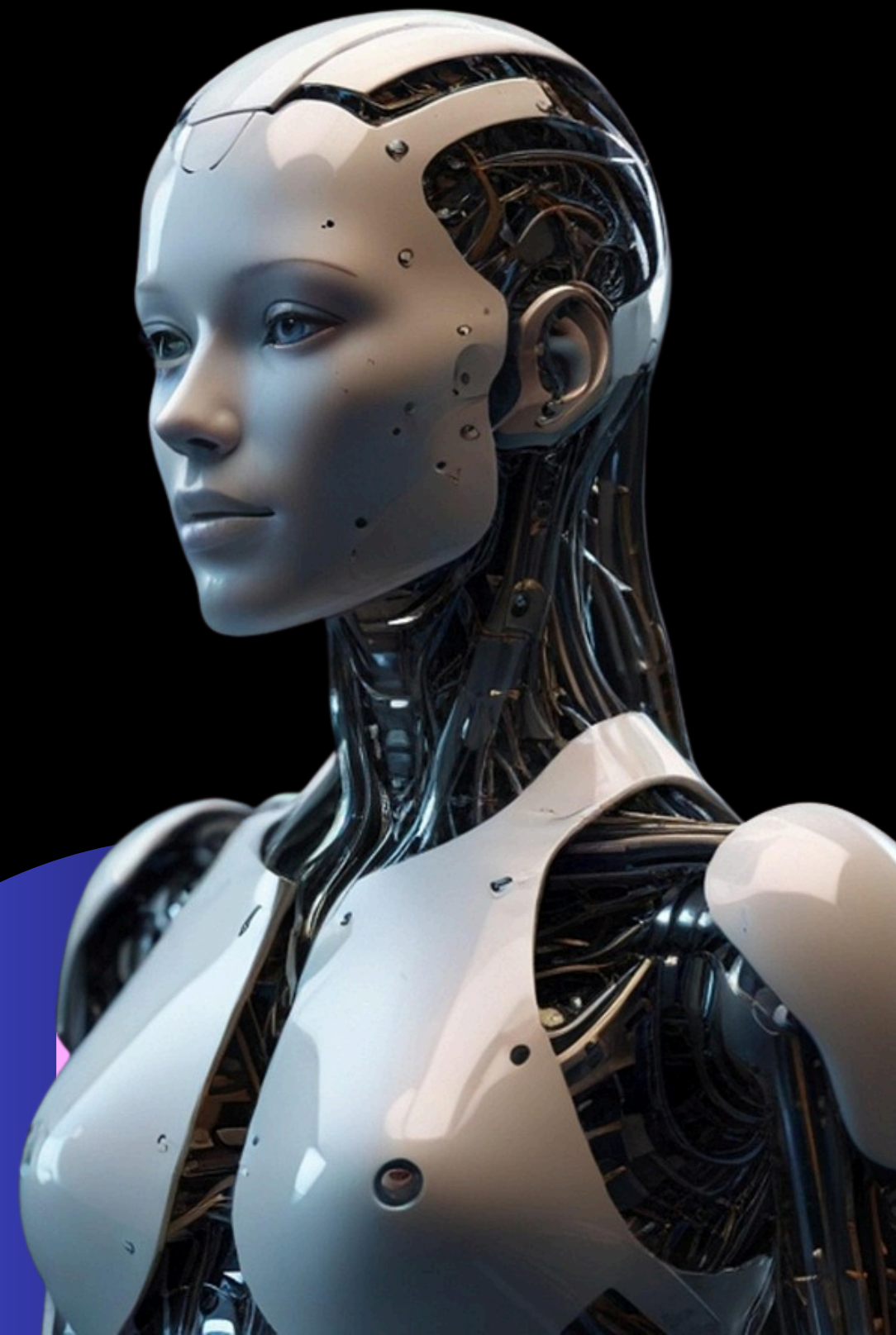
- Téléchargement et configuration de Selenium WebDriver.
- Introduction à GitHub pour la gestion de code source.
- Création d'un projet Maven avec structure adaptée aux tests.
- Organisation du code (src/main/java vs src/test/java).
- Ajout des dépendances : Selenium, WebDriverManager, TestNG/JUnit.
- Premiers tests de lancement de navigateur.

5. TESTS AVEC SELENIUM WEBDRIVER

- Sélection et interaction avec les éléments web (locators : id, name, xpath, cssSelector...).
- Automatisation de scénarios simples : login, navigation, formulaires.
- Synchronisation des tests : waits implicites, explicites, FluentWait.
- Structure de tests avec Page Object Model (POM).
- Utilisation de TestNG ou JUnit pour structurer et exécuter les tests.
- Prise de captures d'écran et logs.
- Génération de rapports de test.

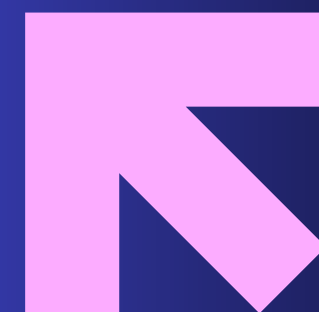
6. CI/CD AVEC GITHUB ET JENKINS

- Introduction à l'intégration continue (CI) et au déploiement continu (CD).
- Présentation de Jenkins : installation, interface, création de jobs.
- Connexion du projet GitHub à Jenkins.
- Configuration d'un pipeline simple pour exécuter les tests Selenium automatiquement.
- Analyse des résultats de build et notification en cas d'échec.



7. AUTOMATISATION INTELLIGENTE AVEC OPENCV ET N8N

- OpenCV:
 - Chargement et comparaison d'images
 - Détection d'objets.
 - Vérification de la présence d'un bouton, d'une icône ou d'un élément graphique.
- n8n:
 - Récupération du code depuis GitHub.
 - Lancement automatique des tests Selenium via une commande.



***Soit APPIUM + IA**

1. Introduction & préparation :

Objectif : Comprendre l'automatisation mobile et préparer l'environnement.

A.Introduction aux tests mobiles et Appium:

- Introduction aux tests mobiles
 - Différence entre tests manuels et automatisés
 - Types de tests (unitaires, fonctionnels, régression, etc.)
- Présentation d'Appium
 - Architecture et fonctionnement
 - Appium vs Selenium
 - Android vs iOS

B.Installation et configuration

- Installation des outils :
 - Java JDK
 - Android Studio + SDK + ADB
 - Node.js + Appium Server
 - Appium Inspector
 - IDE (IntelliJ IDEA)
- Configuration du PATH et vérification avec :
 - java -version
 - adb devices
 - appium-version



2.Java pour l'automatisation :

Objectif : Maîtriser les bases de Java pour écrire des scripts Appium.

A.Java : bases

- Variables et types de données
- Opérateurs et expressions
- Conditions (if, else, switch)
- Boucles (for, while)

B.Java : méthodes et POO

- Méthodes et paramètres
- Classes et objets
- Constructeurs
- Packages et organisation du code

C.Java + Outils de test

- Introduction à Maven et gestion des dépendances
- Introduction à JUnit/TestNG
- Concepts BDD avec Cucumber
- Écrire un premier test unitaire simple



3.Introduction à Appium :

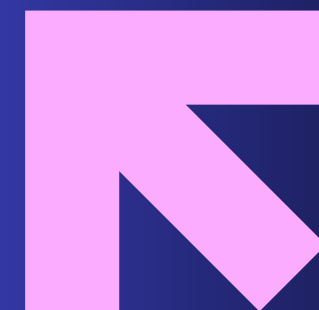
Objectif : Comprendre Appium et écrire un premier script.

A.Découverte d'Appium

- Architecture d'Appium en détail
- Desired Capabilities et configuration
- Drivers (UIAutomator2, Espresso, XCUITest)
- Connexion avec un émulateur ou un appareil réel

B.Premier script Appium :

- Localisation des éléments : ID, XPath, Accessibility ID
- Actions simples : click, sendKeys, getText
- Synchronisation et wait.



4. Appium avancé :

Objectif : Automatiser des scénarios complexes et structurer le code.

A.Actions avancées

- Scroll, Swipe, Drag & Drop
- Gestion des popups et alertes
- Prise de captures d'écran

B.Page Object Model (POM)

- Introduction au POM
- Avantages et bonnes pratiques
- Structurer le code avec POM

C.Framework complet

- Organisation du projet avec Maven
- Gestion des données avec Data Driven Testing
- Génération de rapports (Allure/ExtentReports)

D.Exécution parallèle et optimisation

- Exécution parallèle avec Appium Grid
- Optimisation des temps d'attente (implicit wait, explicit wait)
- Introduction à CI/CD avec Jenkins



5. Utilisation de l'IA dans les test mobiles :

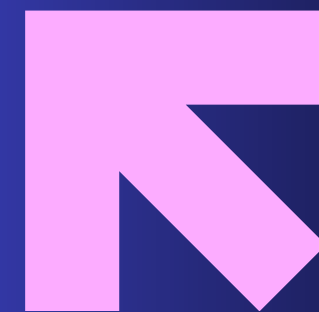
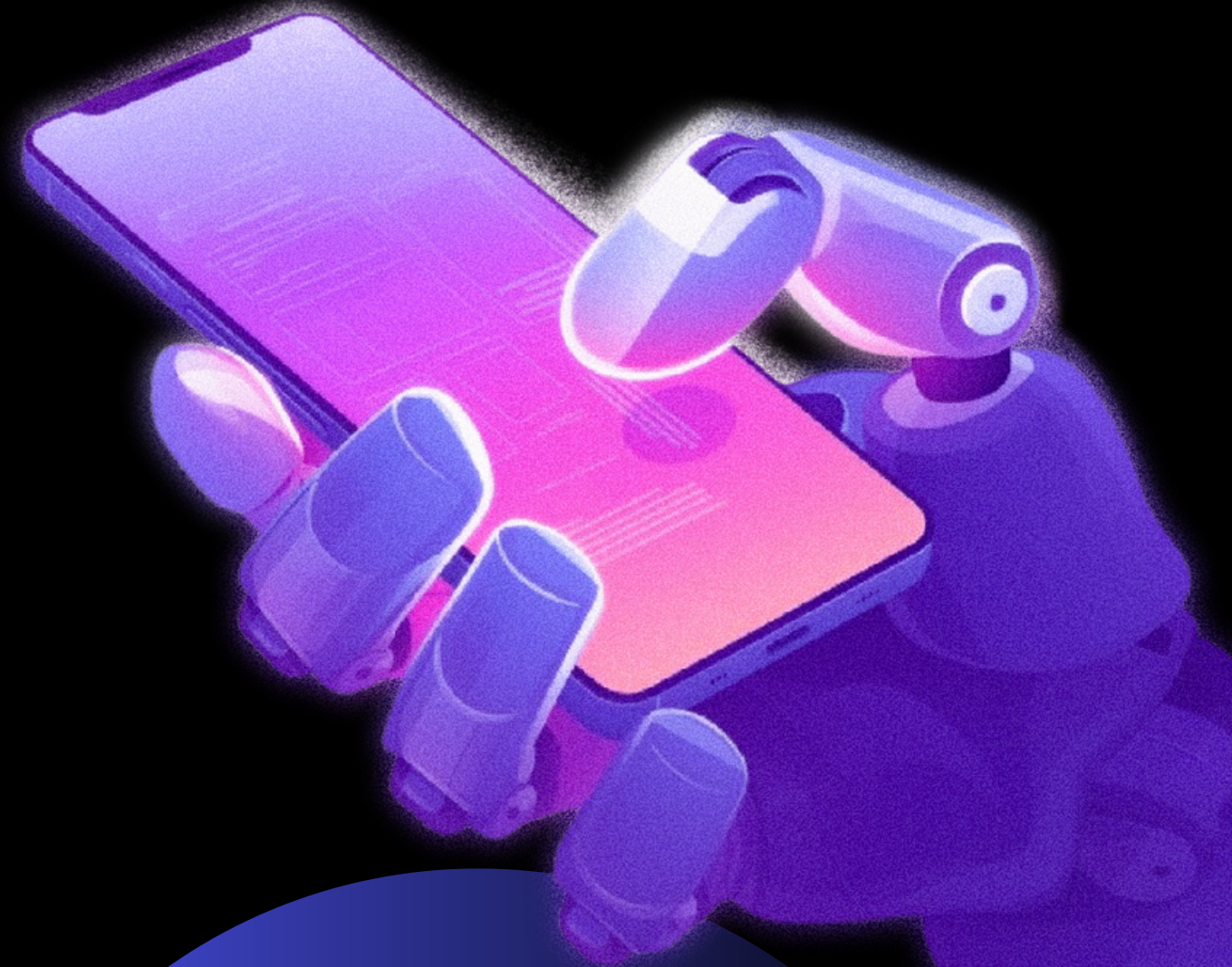
Objectif : Découvrir comment l'IA peut générer et optimiser les tests.

A.Introduction à l'IA dans les tests

- Concepts d'IA appliqués au test logiciel

6. Intégration CI/CD : (2H)

- Connexion avec Jenkins
- Lancer les tests via pipeline
- Générer un rapport automatique



MOIS 4 : PRÉPARATION A4Q

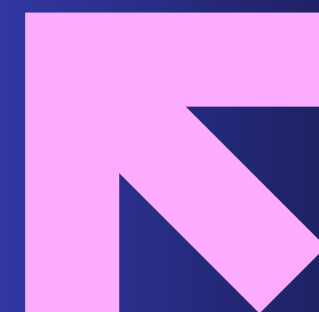
CHAPITRE 1 : FONDAMENTAUX DE L'AUTOMATISATION DES TESTS

- Qu'est-ce que l'automatisation des tests : définitions, objectifs, enjeux
- Différence entre tests manuels et automatisés, avantages et limites.
- Facteurs de succès dans un projet d'automatisation et risques associés.
- Présentation de Selenium (WebDriver, IDE, Grid) et son rôle dans l'architecture de test.

CHAPITRE 2 : TECHNOLOGIES WEB POUR L'AUTOMATISATION

- Notions de base : HTML, XML, DOM.
- Balises HTML, attributs, structure de page.
- Sélecteurs CSS et XPath (syntaxe, axes, fonctions).
- Autres localisateurs (name, linkText, etc.).

A4Q



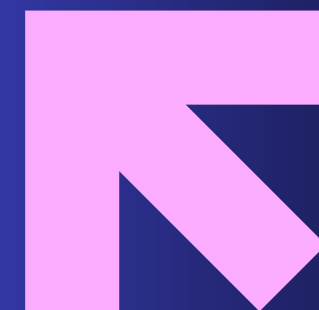
CHAPITRE 3 : UTILISATION DE SELENIUM WEBDRIVER

- Installation et configuration de WebDriver.
- Gestion des navigateurs (Chrome, Firefox, Edge).
- Navigation (URL, back, forward, refresh).
- Interactions utilisateur : clics, formulaires, dropdowns, popups.
- Vérification de l'état des éléments (visibilité, attributs, sélection).
- Captures d'écran et logs.
- Gestion des frames et fenêtres.

CHAPITRE 4 : PRÉPARATION DE SCRIPTS DE TEST MAINTENABLES

- Principes de maintenabilité (lisibilité, modularité, robustesse).
- Mécanismes d'attente (implicites, explicites, fluent waits).
- Modèle Page Object (POM).
- Tests basés sur mots-clés (Keyword-Driven).
- Structuration des suites de tests.
- Facteurs influençant la maintenabilité.

A4Q



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

