**Rapport de Tests CEM**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sample** | **Section** | **Frequency** | **Detector** | **Polarization** | **Antenna** | **Margin** | **Overtaking** | **Conformity** | **Limit** |
| CHAPITRE I. CONCEPTSTHEORIQUES |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.Technologies de développement web |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les technologies web désignent l'ensemble des outils et des langages utilisés pour créer des sites web et des applications web. Ces technologies permettent de développer | de gérer et de publier du contenu sur Internet. Elles comprennent des langages de programmation | NaN | des librairies | des CMS (Content Management Systems) et des bases de données. Les technologies web évoluent rapidement | et il est essentiel pour les développeurs web de rester à jour avec les dernières avancées. | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.1. Langages de programmation Web |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.1.1. Définition |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Un langage de programmation web est un langage utilisé pour créer des applications et des sites web interactifs. Il permet aux développeurs de définir et de contrôler le comportement des éléments sur une page web | d'interagir avec les utilisateurs | NaN | telles que la logique côté serveur | la manipulation des bases de données | l'affichage des pages web et la gestion des interactions utilisateur. Ils permettent de construire des fonctionnalités interactives | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| de données et de communiquer avec d'autres services web. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.1.2. Langages de programmation couramment utilisés |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| HTML (HyperText Markup Language): HTML est le langage de balisage de base utilisé pour créer la structure et le contenu des pages web. Il utilise des balises pour définir différents éléments tels que les titres | les paragraphes | NaN | les liens | etc. HTML est une norme du World Wide Web Consortium (W3C) et est pris en charge par tous les navigateurs web. |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages de HTML : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Facile à apprendre et à utiliser. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Prise en charge par tous les navigateurs web. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Structure claire et hiérarchique du contenu. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients de HTML : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Limité en termes de fonctionnalités interactives. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Nécessite l'utilisation d'autres langages comme CSS et JavaScript pour une mise en page et une interactivité avancée. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| CSS (Cascading Style Sheets): CSS est un langage de feuille de style utilisé pour décrire la présentation et l'apparence des pages web écrites en HTML. Il permet de contrôler les couleurs | les polices | NaN | les bordures | les mises en page et d'autres aspects visuels des éléments HTML. CSS sépare la présentation du contenu | ce qui facilite la maintenance et la mise à jour des sites web. | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages de CSS : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Permet une personnalisation complète de l'apparence des pages web. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Favorise la séparation des préoccupations (separation of concerns) en séparant la présentation du contenu. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Permet la création de mises en page réactives et adaptables. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients de CSS : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Difficulté à gérer la compatibilité entre les navigateurs. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Peut nécessiter une connaissance approfondie pour résoudre des problèmes de mise en page complexes. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| JavaScript : JavaScript est un langage de programmation de script côté client qui permet d'ajouter de l'interactivité et de la dynamique aux pages web. Il peut être utilisé pour manipuler le contenu HTML | effectuer des calculs | NaN | communiquer avec les serveurs | etc. JavaScript est supporté par tous les navigateurs modernes et est essentiel pour le développement web interactif. |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages de JavaScript : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Permet l'interaction avec l'utilisateur et la manipulation du contenu web en temps réel. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Facilite la création d'applications web réactives et interactives. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Large écosystème de bibliothèques et de Frameworks disponibles. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients de JavaScript : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Différences de compatibilité entre les navigateurs. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Risques de sécurité potentiels si mal utilisé. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| NB : Actuellement il est possible d’exécuter et d’utiliser JavaScript du côté serveur grâce à NodeJs |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| PHP (Hypertext Preprocessor): PHP est un langage de script côté serveur largement utilisé pour le développement web. Il est souvent utilisé pour créer des sites web dynamiques et interagir avec les bases de données. PHP est intégré dans le code HTML et peut être utilisé en conjonction avec des bases de données pour créer des applications web puissantes. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages de PHP : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Facile à apprendre et à intégrer dans le code HTML. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Grande communauté de développeurs et de ressources en ligne disponibles. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Bonne prise en charge des bases de données. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients de PHP : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Pas d’inconvénients majeurs à souligner |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Python: Python est un langage de programmation polyvalent qui peut être utilisé pour le développement web | entre autres. Il est apprécié pour sa syntaxe claire et lisible | NaN | tels que Django et Flask. [4] |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages de Python : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Syntaxe claire et expressive | facile à lire et à comprendre. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Large écosystème de bibliothèques et de frameworks pour le développement web. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Grande communauté de développeurs et de ressources en ligne. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients de Python : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Pas d’inconvénients majeurs à souligner |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.2. Les Frameworks et les bibliothèques Web |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.2.1. Définitions |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Un Framework est un ensemble de bibliothèques | d'outils et de composants préétablis qui fournissent une structure et des fonctionnalités communes pour faciliter le développement d'applications. Il fournit un cadre de travail pour organiser et structurer le code | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Une bibliothèque web est un ensemble de fonctions | de routines et demcomposants pré écrits qui fournissent des fonctionnalités spécifiques pour le développement d'applications web. Les bibliothèques sont généralement utilisées pour résoudre des tâches particulières | NaN | les requêtes AJAX | la gestion des événements ou le traitement des données. Elles sont conçues pour être réutilisées et intégrées dans le code d'une application pour fournir des fonctionnalités spécifiques. |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.2.2. Différences entre Framework et Bibliothèque[6] |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Contrôle de flux : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Framework : Dans un framework | le contrôle de flux est généralement inversé | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Bibliothèque : Dans une bibliothèque | le développeur a le contrôle total du flux de l'application. Il utilise la bibliothèque pour des tâches spécifiques | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Niveau d'abstraction : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Framework : Un framework fournit une structure et une architecture globale pour le développement d'applications. Il abstrait souvent les détails complexes et fournit des modèles de conception et des conventions prédéfinis. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Bibliothèque : Une bibliothèque se concentre sur des fonctionnalités spécifiques et ournit des méthodes et des outils pour effectuer des opérations particulières. Elle n'impose pas de structure globale à l'application. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Flexibilité : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Framework : Les frameworks sont souvent moins flexibles car ils suivent des conventions et des modèles prédéfinis. Cela peut limiter la liberté de choix et depersonnalisation dans certaines parties de l'application. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Bibliothèque : Les bibliothèques offrent généralement plus de flexibilité car elles sont utilisées pour des tâches spécifiques et peuvent être intégrées de manière sélective dans une application. Il convient de noter que la distinction entre un framework et une bibliothèque peut parfois être floue | car certains outils peuvent avoir des caractéristiques des deux. La terminologie peut également varier selon les sources et les langages de programmation. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.2.3. Frameworks et bibliothèques Web couramment utilisés |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| React (Bibliothèque) : React est une bibliothèque JavaScript développée par Facebook. Elle permet de construire des interfaces utilisateur interactives et réactives en utilisant des composants réutilisables. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Réutilisation des composants | rendu côté serveur possible avec Next.js | NaN | performances élevées grâce à la virtual DOM. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Courbe d'apprentissage initiale | nécessite l'utilisation d'autres outils et bibliothèques pour construire une application complète. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Angular (Framework) : Angular est un framework JavaScript développé par Google. Il permet de construire des applications web complètes en utilisant le paradigme du développement dirigé par les composants. [8] |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Structure solide et cohérente | prise en charge complète de la création d'applications | NaN | tests unitaires intégrés. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Courbe d'apprentissage abrupte | complexité accrue pour les petites applications | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Vue.js (Bibliothèque) :Vue.js est une bibliothèque JavaScript progressive et facile à apprendre. Elle permet de construire des interfaces utilisateur interactives en se concentrant sur la vue des données. [9] |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Courbe d'apprentissage douce | flexibilité pour intégrer progressivement dans les projets existants | NaN | taille réduite. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Écosystème moins mature que React ou Angular | moins adapté aux applications très complexes. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Django (Framework) : Django est un framework web Python qui suit le principe du DRY (Don't Repeat Yourself). Il fournit une structure solide pour développer des applications web complexes. [10] |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Développement rapide grâce à la génération automatique du code | sécurité ntégrée | NaN | administration administrative prête à l'emploi. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Courbe d'apprentissage initiale | moins adapté aux applications légères ou simples. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Flask (Framework) : Flask est un framework web minimaliste écrit en Python. Il met l'accent sur la simplicité et la facilité d'utilisation | tout en offrant les fonctionnalités nécessaires pour développer des applications web légères. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Facilité d'apprentissage et de démarrage | flexibilité pour créer des applications simples ou complexes | NaN | documentation complète. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Moins d'opinions et de conventions prédéfinies | nécessite d'assembler différents modules pour obtenir une application complète. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Express.js (Framework) : Express.js est un framework web minimaliste et flexible pour Node.js. Il facilite la création d'applications web et d'API en utilisant JavaScript. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Simple et facile à apprendre | grande flexibilité | NaN | performances élevées. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Moins d'opinions et de structure prédéfinies | nécessite d'assembler différents modules pour obtenir une application complète. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Laravel (Framework) : Laravel est un framework web écrit en PHP. Il est conçu pour faciliter le développement d'applications web robustes et élégantes en suivant une syntaxe expressive et en offrant de nombreuses fonctionnalités prêtes à l'emploi. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Syntaxe claire et expressive | grande productivité grâce aux fonctionnalités intégrées | NaN | système de routage simple | large communauté de développeurs. |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Courbe d'apprentissage initiale | certaines fonctionnalités avancées peuvent être complexes à maîtriser. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Symfony (Framework) : Symfony est un framework web puissant écrit en PHP. Il suit le principe du développement basé sur des composants et fournit une architecture solide pour la création d'applications web évolutives et performantes. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Architecture modulaire basée sur des composants | flexibilité pour personnaliser et étendre les fonctionnalités | NaN | documentation complète. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Courbe d'apprentissage initiale | complexité accrue pour les petites applications | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| AdonisJs (Framework) :AdonisJs est un framework web pour Node.js qui facilite le développement d'applications web. Il se concentre sur la simplicité et la productivité | offrant une structure solide et des fonctionnalités intégrées comme la gestion des routes | NaN | l'authentification et bien plus encore. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| AdonisJs offre une architecture MVC (Modèle-Vue-Contrôleur) bien définie | ce qui facilite l'organisation du code. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avec Lucid | l'ORM d'AdonisJs | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| La courbe d'apprentissage est relativement douce | surtout si tu as déjà des connaissances en JavaScript. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| AdonisJs inclut de nombreuses fonctionnalités prêtes à l'emploi | comme la gestion des sessions | NaN | et les validations. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| La documentation est bien structurée et facile à comprendre | ce qui est un atout pour les développeurs. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Comparé à d'autres frameworks comme Express ou Laravel | la communauté est moins vaste | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Bien qu'il soit performant pour de nombreuses applications | il peut être un peu plus lent que des frameworks plus minimalistes comme Express pour certaines tâches. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.3. Les CMS |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.3.1. Définition |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Un CMS (Système de gestion de contenu) est une plateforme logicielle utilisée pour créer | gérer et organiser le contenu d'un site web de manière intuitive. Il s'agit d'un outil puissant et polyvalent qui permet aux utilisateurs de créer et de gérer des sites web sans avoir à maîtriser des langages de programmation complexes. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Le principal objectif d'un CMS est de simplifier le processus de création et de gestion de contenu en offrant une interface conviviale et des fonctionnalités spécifiques. Grâce à son interface intuitive | un CMS permet aux utilisateurs de créer et de structurer facilement leur contenu | NaN | des pages | des images | des vidéos ou d'autres types de médias. Il offre souvent un éditeur de texte WYSIWYG (What You See Is What You Get) qui permet de formater le contenu de manière visuelle | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.3.2. Avantages des CMS : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Facilité d'utilisation : Les CMS sont conçus pour simplifier la création et la gestion de sites web | même pour les utilisateurs non techniques. Leur interface conviviale permet d'effectuer des tâches telles que l'ajout de contenu | NaN | sans nécessiter de compétences en programmation. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Gestion du contenu simplifiée : Les CMS offrent des fonctionnalités de gestion du contenu qui permettent aux utilisateurs de créer | modifier et organiser facilement le contenu de leur site web. Les fonctionnalités telles que l'éditeur de texte WYSIWYG (What You See Is What You Get) simplifient la mise en forme du contenu sans avoir à écrire de code HTML. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Personnalisation et flexibilité : Les CMS proposent souvent une vaste gamme de thèmes | de modèles et de plugins/extensions pour personnaliser l'apparence et les fonctionnalités du site web. Cela permet de créer un site web unique et adapté aux besoins spécifiques de l'utilisateur | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Évolutivité : Les CMS sont conçus pour gérer des sites web de toutes tailles | du | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| simple blog personnel aux sites web d'entreprise complexes. Ils offrent des |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| fonctionnalités évolutives qui permettent d'ajouter de nouvelles pages | de gérer des | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| utilisateurs et de développer le site web au fur et à mesure de sa croissance. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Mises à jour et sécurité : Les CMS sont régulièrement mis à jour pour améliorer les fonctionnalités | corriger les bugs et résoudre les problèmes de sécurité. Les utilisateurs peuvent facilement installer les mises à jour pour bénéficier des 13 dernières améliorations et des correctifs de sécurité | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.3.3. Inconvénients des CMS : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Courbe d'apprentissage initiale : Certains CMS peuvent présenter une courbe |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| d'apprentissage initiale | en particulier pour les utilisateurs moins familiers avec les | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| concepts et les fonctionnalités spécifiques au CMS. Il peut être nécessaire de |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| consacrer du temps à l'apprentissage du système et à la compréhension de ses |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| fonctionnalités pour utiliser le CMS de manière optimale. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Limitations de personnalisation : Bien que les CMS offrent des options de |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| personnalisation | ils peuvent avoir des limitations par rapport à la création d'un site | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| web entièrement personnalisé à partir de zéro. Certains CMS peuvent limiter les |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| options de conception ou avoir des contraintes sur la mise en page | en raison des | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| modèles et des thèmes prédéfinis disponibles. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Dépendance vis-à-vis du CMS : L'utilisation d'un CMS peut rendre les utilisateurs dépendants de la plateforme. Cela signifie qu'ils doivent respecter les directives et les fonctionnalités fournies par le CMS | et peuvent être limités dans leurs choix et leurs possibilités de personnalisation. Changer de CMS peut également être un processus complexe et nécessiter la migration du contenu existant. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Performance du site web : Certains CMS peuvent être moins optimisés en termes de performance par rapport à un site web entièrement développé sur mesure. Des fonctionnalités supplémentaires | des plugins ou des thèmes lourds peuvent affecter la vitesse de chargement du site web | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les CMS offrent une solution pratique pour créer | gérer et organiser le contenu d'un site web avec facilité. Ils offrent des avantages tels que la facilité d'utilisation | NaN | la personnalisation | l'évolutivité et les mises à jour de sécurité. Cependant | il est important de prendre en compte les inconvénients potentiels tels que la courbe d'apprentissage initiale | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.3.4. Les CMS couramment utilisés dans le Web |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| WordPress : WordPress est le CMS le plus populaire | utilisé par des millions de sites web dans le monde. Il offre une grande flexibilité et une vaste gamme de thèmes et de plugins pour personnaliser et étendre les fonctionnalités. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : Facilité d'utilisation | grande communauté de développeurs | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| bibliothèque de thèmes et de plugins | convient à différents types de sites web. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : Vulnérabilité aux attaques de sécurité en raison de sa popularité |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| nécessite des mises à jour régulières | performances moins optimisées pour les sites web très complexes. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Joomla : Joomla est un CMS populaire et polyvalent. Il est apprécié pour sa flexibilité et ses capacités de gestion des utilisateurs | ce qui en fait un choix courant pour les sites web communautaires | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : Interface d'administration conviviale | gestion avancée des utilisateurs | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| nombreuses extensions disponibles | adapté aux sites web communautaires. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : Courbe d'apprentissage initiale | moins de thèmes et de plugins | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| disponibles par rapport à WordPress | potentiellement moins adapté pour les sites web de grande envergure. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Drupal : Drupal est un CMS open source qui met l'accent sur la sécurité et la flexibilité. Il est capable de gérer des sites web complexes et hautement personnalisés | et est souvent utilisé pour les sites web institutionnels | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : Haut niveau de personnalisation et de flexibilité | sécurité avancée | NaN | évolutivité pour les grands sites web. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : Courbe d'apprentissage abrupte | moins convivial pour les utilisateurs non techniques | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Magento : Magento est un CMS conçu spécifiquement pour le commerce électronique. Il offre des fonctionnalités avancées de vente en ligne | de gestion des produits et des commandes | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : Puissantes fonctionnalités de commerce électronique | gestion avancée des produits et des commandes | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : Courbe d'apprentissage initiale | nécessite une expertise technique pour l'installation et la personnalisation | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Shopify : Shopify est une plateforme CMS dédiée au commerce électronique en mode SaaS (Software as a Service). Il permet de créer rapidement et facilement des boutiques en ligne avec des fonctionnalités de vente | de gestion des commandes et de paiement intégrées. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Avantages : Facilité d'utilisation | conception conviviale pour les débutants | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| hébergement et sécurité gérés | nombreuses intégrations de paiement disponibles. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Inconvénients : Personnalisation limitée par rapport à d'autres CMS | frais mensuels récurrents | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Il est important de noter que la sélection d'un CMS dépend des besoins spécifiques du projet | de la complexité du site web | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.4. Les logiciels et programmes utilisés dans le developpement Web |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Le développement Web est un processus complexe qui implique l'utilisation de divers logiciels et programmes pour concevoir | créer et maintenir des sites Web. Ces outils jouent un rôle essentiel dans la productivité et l'efficacité des développeurs Web | NaN | de collaborer | de tester et de déployer leurs projets. Dans cette section | nous explorerons les différentes catégories de logiciels utilisés dans le développement Web | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.4.1. Éditeurs de texte et IDE |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les éditeurs de texte sont des logiciels de base utilisés pour écrire et modifier du code source. Ils offrent des fonctionnalités telles que la coloration syntaxique | l'indentation automatique et la numérotation des lignes pour faciliter la lecture et la compréhension du code. Certains éditeurs populaires pour le développement Web sont Sublime Text | NaN | Atom et Notepad++. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les environnements de développement intégrés (IDE) sont des logiciels plus avancés qui offrent des fonctionnalités supplémentaires spécifiquement conçues pour le développement logiciel. Ils incluent souvent un éditeur de texte | un débogueur | NaN | Microsoft Visual Studio | Eclipse et NetBeans. |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.4.2. Systèmes de contrôle de version |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les systèmes de contrôle de version sont des outils qui permettent de gérer les |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| modifications apportées au code source d'un projet au fil du temps. Ils enregistrent les différentes versions du code | facilitent la collaboration entre les membres d'une équipe et permettent de revenir à des versions antérieures si nécessaire. Git est le système de contrôle de version le plus utilisé | NaN | GitLab et Bitbucket. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.4.3. Serveurs Web |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les serveurs Web sont des logiciels qui permettent de servir des pages Web aux utilisateurs finaux. Ils interprètent les requêtes HTTP et renvoient les fichiers appropriés pour afficher les sites Web. Apache HTTP Server et Nginx sont deux exemples populaires de serveurs Web largement utilisés. Ils offrent des fonctionnalités avancées de configuration | de gestion de la sécurité et de performances pour répondre aux besoins des sites Web à fort trafic. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.4.4. Outils de test et de débogage |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les outils de test et de débogage sont essentiels pour garantir la qualité et la stabilité des sites Web. Ils permettent de détecter et de résoudre les erreurs de code | les problèmes de compatibilité entre les navigateurs et les performances médiocres. Des outils tels que Chrome DevTools | NaN | la modification en direct et le débogage du code JavaScript et des interfaces utilisateur. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.1.4.5. Outils de gestion de projets et de collaboration: |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Pour faciliter la gestion des projets Web | des outils de gestion de projets et de collaboration sont utilisés. Ils permettent de suivre les tâches | NaN | de gérer les problèmes et de faciliter la communication entre les membres de l'équipe. Des exemples d'outils populaires sont Jira | Trello | Asana et Basecamp. | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.Technologies de développement mobile |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Dans notre société de plus en plus connectée | les applications mobiles jouent un rôle essentiel dans notre quotidien. Le développement d'applications mobiles nécessite l'utilisation de technologies spécifiques pour créer des expériences utilisateur optimisées sur les appareils mobiles tels que les smartphones et les tablettes. Dans cette section | NaN | en mettant l'accent sur les langages de programmation | les Frameworks et bibliothèques | les plateformes de développement et les outils de test. | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.1. Langages de programmation populaires pour le développement des applications mobiles |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les langages de programmation sont la base du développement mobile. Différents langages sont utilisés pour développer des applications mobiles pour différentes plateformes. Voici une analyse des langages couramment utilisés dans le développement mobile : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.1.1. Java pour Android |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Java est le langage de programmation principal pour le développement d'applications Android. Il offre une grande flexibilité | une vaste communauté de développeurs et un écosystème solide de bibliothèques et de frameworks. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.1.2. Kotlin pour Android |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Kotlin est un langage de programmation moderne développé par JetBrains | qui est devenu le langage préféré pour le développement d'applications Android. Kotlin est entièrement compatible avec Java | NaN | ce qui permet de réduire la quantité de code nécessaire tout en améliorant la lisibilité. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.1.3. Swift et Objective-C pour iOS : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Swift est le langage de programmation moderne utilisé pour le développement |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| d'applications iOS. Il est rapide | sécurisé et offre une syntaxe concise. Objective-C | NaN | est encore utilisé dans certains projets existants. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.1.4. JavaScript pour le développement d'applications hybrides : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les frameworks basés sur JavaScript tels que React Native | Ionic et PhoneGap permettent de développer des applications hybrides qui fonctionnent à la fois sur Android et iOS. Ces frameworks utilisent des technologies Web (HTML | NaN | JavaScript) pour créer des interfaces utilisateur natives. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.2. Frameworks et bibliothèques pour le développement mobile |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les frameworks et bibliothèques offrent des outils et des fonctionnalités supplémentaires pour faciliter le développement d'applications mobiles. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Voici une exploration de certains d'entre eux : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.2.1. Android Studio |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Il s'agit de l'IDE (Integrated Development Environment) principal pour le développement d'applications Android. Il comprend un ensemble complet d'outils | de bibliothèques et de frameworks | NaN | Retrofit pour les appels réseau | Dagger pour l'injection de dépendances | et bien d'autres. | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.2.2. Xcode |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| L'IDE d'Apple pour le développement iOS. Il offre des fonctionnalités avancées telles que l'éditeur de code en plein écran | l'interface de conception d'interface graphique | NaN | CoreData pour la gestion des données | et bien d'autres. |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.2.3. React Native |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Un framework basé sur JavaScript développé par Facebook | permettant de développer des applications mobiles multiplateformes en utilisant des composants réutilisables. Il offre une approche basée sur le code partagé | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.2.4. Flutter |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Un framework développé par Google | utilisant le langage Dart | NaN | avec un vaste ensemble de widgets prêts à l'emploi. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.3. Plateforme de développement Mobile |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les plateformes de développement mobile fournissent des outils | des API et des services pour faciliter le développement | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.3.1. Android |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Plateforme de développement mobile de Google | utilisée pour créer des applications pour les appareils fonctionnant sous le système d'exploitation Android. Elle offre une grande flexibilité | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.3.2. iOS : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Plateforme de développement mobile d'Apple | utilisée pour créer des applications pour les appareils iPhone | NaN | une intégration étroite avec l'écosystème Apple et une |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| sécurité renforcée. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.4. Outils de test et de débogage pour le développement mobile |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les outils de test et de débogage sont essentiels pour garantir la qualité et la stabilité des applications mobiles. Voici une exploration plus détaillée de certains outils couramment utilisés dans le développement mobile : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.4.1. Android Studio : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| L'IDE d'Android Studio propose des fonctionnalités complètes pour le développement | le débogage et les tests d'applications Android. Il permet de créer des tests unitaires | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.4.2. Xcode : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| L'IDE Xcode d'Apple offre des outils de test et de débogage avancés pour le développement d'applications iOS. Il comprend des outils tels que l'Assistant de test | les profilers d'instruments et l'interface de débogage pour identifier et résoudre les problèmes de performance et de bugs. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.4.3. Appium : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Une plateforme open-source pour l'automatisation des tests d'applications mobiles. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Appium prend en charge les tests multiplateformes et permet d'écrire des scripts de test en utilisant des langages tels que Java | JavaScript | NaN | etc. Il est compatible avec Android et iOS. |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.2.4.4. Firebase Test Lab |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Une plateforme de test dans le cloud proposée par Google. Firebase Test Lab permet de tester automatiquement les applications Android et iOS sur une large gamme de périphériques réels | en fournissant des rapports détaillés sur les performances | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.Les API REST |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.1. Vue globale |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les API REST (Representational State Transfer) sont un ensemble de principes |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| architecturaux qui permettent la conception de services Web évolutifs | interopérables et basés sur le protocole HTTP. Elles ont été présentées pour la première fois par Roy Fielding | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| dans sa thèse de doctorat en 2000. Les API REST sont largement utilisées dans le |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| développement d'applications modernes | notamment les applications Web et mobiles. Les API REST sont basées sur le concept de ressources | NaN | des documents ou toute autre entité identifiable. Chaque ressource est généralement identifiée par une URI (Uniform Resource Identifier). Les opérations sur les ressources sont effectuées en utilisant les verbes HTTP tels que GET | POST | PUT et DELETE. Les API REST utilisent également des représentations de ressources | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Interopérabilité : Les API REST sont basées sur des standards ouverts tels que |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| HTTP | JSON et XML | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| plates-formes. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Scalabilité : Les API REST sont conçues pour être stateless | ce qui permet une | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| meilleure gestion de la scalabilité horizontale. Les serveurs peuvent être répliqués |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| et les requêtes peuvent être distribuées sans dépendre d'un état côté serveur. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Réutilisabilité : Les API REST permettent de concevoir des services Web |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| modulaires et réutilisables. Les ressources peuvent être manipulées |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| indépendamment les unes des autres | ce qui favorise la réutilisation du code et la | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| modularité de l'architecture. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Gestion de la complexité : Les principes RESTful permettent de gérer la |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| complexité des applications en séparant clairement les préoccupations (separation |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| of concerns). Cela facilite la maintenance | la compréhension et l'évolutivité du code. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.2. Principes fondamentaux des API REST |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.2.1. Ressources et identifiants des ressources |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les API REST sont basées sur le concept de ressources | qui représentent des entités identifiables | NaN | des documents ou des services. Chaque ressource est généralement identifiée par une URI (Uniform Resource Identifier) | qui est une chaîne de caractères utilisée pour localiser et accéder à la ressource. Les URIs sont souvent structurées de manière hiérarchique avec des segments qui représentent les différentes parties de l'URI. |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Exemple :URI d'une ressource utilisateur : /utilisateurs/123 |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Dans cet exemple | "/utilisateurs" est le segment de niveau supérieur qui représente la collection de tous les utilisateurs | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.2.2. Verbes http et opérations sur les ressources |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les API REST utilisent les verbes HTTP pour spécifier les opérations à effectuer sur les ressources. Les verbes HTTP les plus couramment utilisés dans les API REST sont : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • GET : utilisé pour récupérer une représentation d'une ressource. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Par exemple | une requête GET sur "/utilisateurs/123" récupérerait les détails de | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| l'utilisateur avec l'identifiant 123. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • POST : utilisé pour créer une nouvelle ressource. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Par exemple | une requête POST sur "/utilisateurs" créerait un nouvel utilisateur. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • PUT : utilisé pour mettre à jour une ressource existante ou pour créer une ressource |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| si elle n'existe pas déjà. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Par exemple | une requête PUT sur "/utilisateurs/123" mettrait à jour les détails de | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| l'utilisateur avec l'identifiant 123. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • DELETE : utilisé pour supprimer une ressource. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Par exemple | une requête DELETE sur "/utilisateurs/123" supprimerait l'utilisateur | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| avec l'identifiant 123. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.2.3. Représentation des ressources |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| La représentation des ressources constitue un aspect fondamental des API REST. Elle se réfère à la manière dont les données sont structurées et formatées dans les réponses de l'API. Les formats couramment utilisés pour représenter les ressources sont JSON (JavaScript Object Notation) et XML (eXtensible Markup Language). |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Parmi ces formats | JSON est devenu le plus populaire en raison de sa simplicité et de sa lisibilité. Lors de la conception d'une API REST | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Par exemple | pour une ressource "utilisateur" | NaN | "email" et "rôle". Il est également possible d'établir des relations entre les ressources en utilisant des identifiants ou des liens. Prenons l'exemple d'une API REST qui gère des utilisateurs. La représentation JSON d'un utilisateur pourrait ressembler à ceci : |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| } |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.2.4. Composants clés du corps d'une requête et d'une réponse dans une API REST |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 1. Requête vers l’API : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Lors de l'envoi d'une requête vers une API REST | le corps de la requête contient | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| les données que vous souhaitez envoyer au serveur. Voici les éléments essentiels qui doivent être présents dans une requête vers une API : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Méthode HTTP : L'une des méthodes HTTP standard doit être spécifiée | telles que GET | NaN | PUT | PATCH ou DELETE. Chaque méthode a une signification |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| spécifique et indique l'opération à effectuer sur la ressource. Cfr le point I.3.2.2 |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • En-têtes (Headers) : Les en-têtes HTTP fournissent des informations |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| supplémentaires sur la requête. Ils peuvent inclure des informations telles que |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| l'authentification | le type de contenu accepté | NaN | etc.23 |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Paramètres de requête (Query Parameters) : Les paramètres de requête sont |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| utilisés pour spécifier des valeurs supplémentaires dans l'URL de la requête. Ils sont |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| généralement utilisés pour filtrer | trier ou paginer les résultats. Les paramètres de | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| requête sont ajoutés à l'URL après un point d'interrogation (?) | et ils sont séparés | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| par des ampersands (&). |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Corps de la requête (Request Body) : Le corps de la requête est utilisé pour |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| envoyer des données supplémentaires avec la requête. Il est principalement utilisé |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| avec les méthodes POST | PUT ou PATCH pour créer ou mettre à jour une ressource. Le format du corps de la requête dépend du type de contenu (Content-Type) spécifié dans les en-têtes. Les formats couramment utilisés sont JSON | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 2. Réponse d'une API : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Lorsqu'une API REST renvoie une réponse | celle-ci est composée de plusieurs | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| éléments importants : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Code de statut (Status Code) : Les codes de statut HTTP sont des valeurs |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| numériques renvoyées par le serveur pour indiquer le résultat d'une requête. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Voici quelques codes de statuts couramment utilisés et leurs importances : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| o 200 OK : La requête a été réussie. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| o 201 Created : La ressource a été créée avec succès. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| o 204 No Content : La requête a été réussie | mais il n'y a pas de contenu à | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| renvoyer. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| o 400 Bad Request : La requête est mal formée ou contient des paramètres |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| invalides. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| o 401 Unauthorized : L'accès à la ressource est refusé en raison d'une absence |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| ou d'une invalidité d'authentification. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| o 403 Forbidden : L'accès à la ressource est refusé même avec une |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| authentification valide. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| o 404 Not Found : La ressource demandée n'a pas été trouvée. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| o 500 Internal Server Error : Une erreur interne s'est produite du côté du |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| serveur. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • En-têtes de réponse (Response Headers) : Les en-têtes de réponse fournissent des |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| informations supplémentaires sur la réponse renvoyée par le serveur. Ils peuvent |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| inclure des informations telles que le type de contenu de la réponse | la date et l'heure | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| de la réponse | les cookies | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Corps de la réponse (Response Body) : Le corps de la réponse contient les données |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| renvoyées par le serveur dans le format spécifié dans les en-têtes de réponse |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| (Content-Type). Le format couramment utilisé pour le corps de la réponse est JSON |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| mais il peut également être XML | HTML | NaN | etc. Le corps de la réponse 24 |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| peut contenir des informations sur la ressource demandée | des messages d'erreur | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| des métadonnées | etc. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.3. Architecture et Conception des API REST |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.3.1. Modélisation des ressources et des relations |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| La modélisation des ressources et des relations est un aspect fondamental de la conception des API REST (Representational State Transfer). Elle consiste à définir les ressources nécessaires pour représenter les données manipulées par l'API et à établir les relations entre ces ressources. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Définition des ressources : La première étape de la modélisation consiste à identifier les entités principales que l'API va gérer. Une ressource représente une entité du système | telle qu'un utilisateur | NaN | une commande | etc. Chaque ressource doit être identifiée par un URI (Uniform Resource Identifier) unique | qui permettra de l'adresser de manière univoque. | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Structure des données : Une fois les ressources identifiées | il est nécessaire de | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| déterminer la structure des données associées à chaque ressource. Cela peut être |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| réalisé à l'aide de modèles de données tels que les schémas JSON ou les schémas |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| XML. Les schémas permettent de définir les attributs et les types de données de |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| chaque ressource | ainsi que les éventuelles contraintes ou relations avec d'autres | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| ressources. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Relations entre les ressources : Les ressources peuvent être liées entre elles par des relations. Par exemple | une commande peut être liée à un utilisateur | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.3.2. Gestion des erreurs et des codes de statut HTTP |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| La gestion des erreurs et des codes de statut HTTP est un aspect crucial pour assurer la fiabilité et la robustesse des API REST. Les codes de statut HTTP fournissent des informations sur l'état de la requête et de la réponse | tandis que la gestion des erreurs permet de signaler les problèmes survenus lors du traitement des requêtes. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.3.3. Sécurité et authentification dans les API REST |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| La sécurité et l'authentification sont des aspects essentiels lors de la conception d'API REST | afin de protéger les ressources et de garantir l'accès uniquement aux utilisateurs autorisés. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Voici quelques concepts importants à considérer : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 1. Authentification : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| L'authentification permet de vérifier l'identité d'un utilisateur ou d'une application qui accède à l'API. Il existe différentes méthodes d'authentification | telles que : | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Basic Authentication : L'utilisateur envoie ses identifiants (nom d'utilisateur et mot de passe) dans l'en-tête de la requête. Cependant | cette méthode n'est pas sécurisée car les identifiants sont envoyés en clair et peuvent être interceptés. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Token-based Authentication : L'utilisateur obtient un jeton d'authentification |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| lorsqu'il se connecte avec succès. Ce jeton est ensuite inclus dans les en-têtes de |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| toutes les requêtes ultérieures pour prouver son identité. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • OAuth : Il s'agit d'un protocole d'autorisation permettant aux utilisateurs de partager des informations restreintes avec des applications tierces sans divulguer leurs identifiants de connexion. OAuth utilise des jetons d'accès pour authentifier les demandes. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 2. Autorisation : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Une fois qu'un utilisateur est authentifié | il est important de définir des mécanismes d'autorisation pour contrôler les actions qu'il peut effectuer sur les ressources. Cela peut être réalisé en définissant des rôles et des permissions pour chaque utilisateur ou groupe d'utilisateurs. Par exemple | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 3. Sécurité des données : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Outre l'authentification et l'autorisation | il est également essentiel de protéger les | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| données transitant entre le client et le serveur. Cela peut être réalisé en utilisant des protocoles de communication sécurisés tels que HTTPS (HTTP sécurisé) qui chiffrent les données échangées. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.3.4. Bonnes pratiques de conception des API REST |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Pour concevoir des API REST efficaces et faciles à utiliser | il est recommandé de suivre certaines bonnes pratiques. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Voici quelques principes clés à considérer : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 1. Utilisation des verbes HTTP : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les API REST utilisent les verbes HTTP (GET | POST | NaN | DELETE | etc.) pour |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| indiquer l'action à effectuer sur une ressource. Il est important de respecter les |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| conventions d'utilisation des verbes HTTP pour chaque type d'opération. Par exemple | GET est utilisé pour récupérer une ressource | NaN | PUT pour la mettre à jour | et DELETE pour la supprimer. |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 2. Utilisation des URI significatifs : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les URI (Uniform Resource Identifier) doivent être significatifs et intuitifs. Il est |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| recommandé d'utiliser des noms de ressources au singulier plutôt qu'au pluriel | et | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| d'éviter les verbes dans les URI. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Par exemple | "/users" est préférable à "/getUsers". | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les URI doivent également être cohérents et prévisibles pour faciliter la navigation dans l'API. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 3. Versioning de l'API : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Lorsque des changements majeurs sont apportés à une API | il est important de prévoir des mécanismes de versioning pour maintenir la compatibilité avec les clients existants. Le versioning peut être réalisé en incluant la version dans l'URI de l'API ou en utilisant des en-têtes personnalisés. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 4. Pagination des résultats : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Lorsqu'une liste de ressources est renvoyée en réponse à une requête | il est | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| recommandé de mettre en place un mécanisme de pagination pour limiter la quantité de données renvoyées à la fois. Cela permet d'améliorer les performances de l'API et de faciliter la manipulation des résultats par les clients. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 5. Documentation complète : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Une documentation complète et précise est essentielle pour permettre aux |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| développeurs d'utiliser correctement l'API. Elle doit fournir des informations sur les ressources disponibles | les opérations supportées | NaN | les codes de statut renvoyés | etc. La documentation peut être réalisée sous la forme de fichiers JSON | YAML ou HTML | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.3.3.5. Technologies courantes de développement des API REST |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 1. Langages de programmation : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • JavaScript : Langage de programmation largement utilisé pour le développement web | JavaScript est souvent utilisé pour créer des API REST grâce à des frameworks tels que Express.js | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Python : Python est un langage polyvalent qui offre plusieurs frameworks |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| populaires pour le développement d'API REST | notamment Django REST | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Framework | Flask et FastAPI. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Ruby : Ruby est le langage de programmation utilisé dans le framework Ruby on Rails | qui facilite la création d'API RESTful. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Java : Java est un langage populaire pour le développement d'API REST | et des | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| frameworks tels que Spring Boot et JAX-RS sont couramment utilisés. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • C# : C# est utilisé avec le framework ASP.NET pour développer des API REST | et ASP.NET Web API est l'un des frameworks couramment utilisés dans cet | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| écosystème. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 2. Outils de documentation et de test : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Swagger (OpenAPI) : Un outil populaire pour la documentation des API REST. Il permet de décrire les ressources | les opérations disponibles | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| réponses attendues. Il facilite également la génération automatique de la |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| documentation interactive et fournit des fonctionnalités pour tester les API. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Postman : Une application qui facilite le test et le débogage des API REST. Il |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| permet d'envoyer des requêtes HTTP personnalisées | de visualiser les réponses | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| gérer les environnements de test et d'automatiser les tests API. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.4.Les Systèmes de paiement en ligne |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.4.1. Définition |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les systèmes de paiement en ligne sont des solutions électroniques sécurisées qui |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| permettent aux utilisateurs d'effectuer des transactions monétaires via Internet. Ces |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| systèmes offrent aux consommateurs et aux commerçants des moyens pratiques | rapides et sécurisés pour effectuer des achats et des paiements en ligne. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.4.2. Moyens de paiement en ligne |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.4.2.1. Les cartes |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les cartes de crédit et de débit sont des moyens de paiement en ligne largement utilisés. Elles offrent aux utilisateurs une méthode sécurisée et pratique pour effectuer des paiements en ligne. Les cartes de crédit sont émises par des banques et des institutions financières. Elles permettent aux utilisateurs de dépenser jusqu'à une limite de crédit prédéfinie. Les transactions effectuées avec une carte de crédit sont facturées à l'utilisateur et doivent être remboursées selon les termes et conditions établis. Les cartes de débit sont liées à un compte bancaire et permettent aux utilisateurs de dépenser les fonds disponibles dans ce compte. Les paiements effectués avec une carte de débit sont déduits directement du solde du compte. Voici quelques exemples de cartes de crédit et débit les plus utilisées : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Visa : Visa est l'une des principales marques de cartes de paiement au niveau |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| mondial. Elle propose des cartes de crédit et de débit qui sont acceptées dans de |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| nombreux pays et sur de nombreux sites de commerce électronique. Les cartes Visa offrent des fonctionnalités de sécurité avancées | telles que la vérification 3D Secure | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Figure I-1: Illustration carte visa |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Mastercard : Mastercard est une autre marque de carte de paiement largement |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| reconnue. Elle propose des cartes de crédit et de débit qui sont acceptées à l'échelle |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| mondiale. Les cartes Mastercard offrent également des fonctionnalités de sécurité |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| avancées pour protéger les utilisateurs lors des transactions en ligne.[B1] |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Figure I-2: Illustration carte Mastercard |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • American Express (Amex) : American Express est une société de services |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| financiers qui émet des cartes de crédit et de débit | principalement utilisées par une | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| clientèle aisée. Les cartes American Express offrent des avantages exclusifs tels que |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| des programmes de récompenses et des services de conciergerie. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Figure I-3: Illustration carte Amex |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Discover : Discover est une société de services financiers basée aux États-Unis. Elle propose des cartes de crédit et de débit qui sont largement acceptées aux États-Unis. Les cartes Discover offrent des programmes de remise en argent et d'autres avantages pour les titulaires de carte. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Figure I-4: Illustration carte Discover |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • JCB : JCB (Japan Credit Bureau) est une société de services financiers japonaise qui émet des cartes de crédit et de débit. Les cartes JCB sont principalement utilisées au Japon | mais elles sont également acceptées dans certains pays étrangers | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Figure I-5: Illustration JCB carte |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • UnionPay : UnionPay est un réseau de paiement chinois qui émet des cartes de |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| crédit et de débit. Il est largement utilisé en Chine et est également accepté dans de |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| nombreux pays à travers le monde. UnionPay est souvent utilisé par les touristes |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| chinois pour effectuer des paiements lors de leurs voyages à l'étranger. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Figure I-6: Illustration Upay Carte |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.4.2.2. Portefeuilles électroniques : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les portefeuilles électroniques sont des systèmes de paiement en ligne qui permettent aux utilisateurs de stocker et de gérer leurs fonds électroniquement. Ils offrent une alternative pratique aux paiements par carte et sont largement utilisés dans les transactions en ligne. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • PayPal est l'un des portefeuilles électroniques les plus largement utilisés dans le |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| monde. Il permet aux utilisateurs de lier leurs comptes bancaires ou leurs cartes de |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| crédit/débit à leur compte PayPal. Les utilisateurs peuvent effectuer des paiements |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| en ligne en utilisant leur solde PayPal ou en sélectionnant leur source de |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| financement préférée. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Google Pay est un portefeuille électronique développé par Google. Il permet aux utilisateurs de stocker leurs cartes de crédit/débit dans l'application et de les utiliser pour effectuer des paiements en ligne et en magasin. Google Pay offre également des fonctionnalités de paiement peer-to-peer. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Apple Pay est un portefeuille électronique exclusif aux appareils Apple tels que |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| l'iPhone et l'Apple Watch. Il utilise la technologie NFC (Near Field Communication) pour permettre aux utilisateurs de payer en magasin et en ligne en utilisant leurs appareils Apple. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.4.2.3. Paiements mobiles en RDC : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| En République démocratique du Congo (RDC) | les paiements mobiles ont connu une adoption croissante | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • M-Pesa : M-Pesa est un service de paiement mobile populaire en RDC proposé par Vodacom. Les utilisateurs peuvent effectuer des paiements | des transferts d'argent et d'autres transactions financières via leur téléphone mobile. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Airtel Money : Airtel Money est une plateforme de paiement mobile proposée par Airtel en RDC. Les utilisateurs peuvent créer un compte Airtel Money lié à leur numéro de téléphone mobile et effectuer diverses transactions financières. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Orange Money : Orange Money est proposé par Orange en RDC et permet aux |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| utilisateurs de créer un compte lié à leur numéro de téléphone mobile pour effectuer des transactions financières telles que les paiements | les transferts d'argent et les achats en ligne. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Africell Money : Africell Money est une solution de paiement mobile proposée par Africell en RDC. Elle permet aux utilisateurs de réaliser des transactions financières telles que les paiements | les transferts d'argent et les achats en ligne. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| I.4.3. Plateformes d’intégration des systèmes de paiement en ligne au sein des applications |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les plateformes d'intégration des systèmes de paiement en ligne sont des services qui permettent aux développeurs d'ajouter des fonctionnalités de paiement à leurs applications ou sites web. Ces plateformes fournissent des API (interfaces de programmation |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| d'applications) et des outils pour faciliter l'intégration des systèmes de paiement | gérer les transactions financières et assurer la sécurité des paiements en ligne. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Voici quelques-unes des principales plateformes d'intégration de systèmes de paiement en ligne : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Stripe : Stripe est une plateforme de paiement populaire qui permet d'accepter les paiements par carte de crédit et de débit dans plus de 135 devises. Elle offre une API puissante et facile à utiliser | ainsi que des fonctionnalités avancées telles que la gestion des abonnements | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • PayPal : PayPal est une plateforme de paiement en ligne bien connue qui prend en charge les paiements par carte de crédit | de débit | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| portefeuilles électroniques. Elle offre également une API pour faciliter l'intégration des paiements dans les applications. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Freshpay : Freshpay est une plateforme d’intégration des paiements mobiles au sein des applications | la plateforme evolue en RDC et prend en charge tous les systèmes de paiement mobile local ( M-pesa | NaN | Orange money et Afrimoney). |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • FlexPay : FlexPay est une plateforme d'intégration des paiements mobiles au sein des applications | qui évolue en RDC et prend en charge divers systèmes de paiement mobile locaux tels que M-pesa | NaN | Orange money et Afrimoney. Avec FlexPay | les développeurs peuvent facilement intégrer ces options de paiement dans leurs applications grâce à une API conviviale et intuitive. |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| • Zoya Pay : Zoya Pay est une plateforme d'intégration des paiements mobiles conçue pour faciliter les transactions financières au sein des applications. Disponible en RDC | Zoya Pay prend en charge une variété de systèmes de paiement mobiles locaux tels que M-pesa | NaN | Orange money et Afrimoney. Grâce à son API conviviale | Zoya Pay offre une solution simple et pratique pour les développeurs qui souhaitent intégrer des fonctionnalités de paiement mobile dans leurs applications. |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| ChatGPT a dit : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| ChatGPT |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les Progressive Web Apps (PWA) sont construites en utilisant des technologies web standards qui permettent de combiner les avantages des applications web et des applications mobiles natives. Voici les technologies principales utilisées pour créer des PWA : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 1. HTML | CSS | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les technologies de base pour toute application web | y compris les PWA | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| HTML (HyperText Markup Language) : pour structurer le contenu de la page. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| CSS (Cascading Style Sheets) : pour la mise en page et le design de l’application. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| JavaScript : pour rendre l’application interactive et gérer les comportements dynamiques de la page. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 2. Service Workers |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les Service Workers sont des scripts JavaScript qui s'exécutent en arrière-plan pour permettre des fonctionnalités comme : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Le fonctionnement hors ligne : en interceptant les requêtes réseau et en fournissant des réponses depuis le cache (Cache API). |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les notifications push : pour envoyer des messages aux utilisateurs même lorsque l'application n'est pas ouverte. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| La gestion du cache : pour optimiser les performances en stockant des ressources (images | fichiers | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 3. App Shell Model |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Le App Shell Model est un design pattern utilisé pour PWA qui permet de charger rapidement une version de base de l'application (l'interface utilisateur) en priorité | puis de charger dynamiquement le contenu supplémentaire. Cela aide à donner une expérience plus fluide | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 4. Web App Manifest |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Le manifest.json est un fichier JSON qui décrit l’application | avec des informations comme : | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Le nom de l’application. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les couleurs de l'interface. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les icônes pour l’écran d’accueil. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| L'orientation de l'application (portrait ou paysage). |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les écrans de démarrage. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Cela permet aux PWA de s'installer sur l'écran d'accueil de l'utilisateur | ressemblant à une application native. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 5. IndexedDB et Cache API |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| IndexedDB est une base de données côté client permettant de stocker des données structurées de manière persistante dans le navigateur | ce qui est utile pour les PWA qui doivent fonctionner hors ligne ou stocker des données locales. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Cache API : utilisée par les Service Workers pour stocker et récupérer des ressources web | permettant à la PWA de fonctionner hors ligne ou de se charger plus rapidement lors de la prochaine utilisation. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 6. Frameworks et Bibliothèques JavaScript |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| De nombreux frameworks et bibliothèques JavaScript peuvent être utilisés pour faciliter le développement des PWA | notamment : | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| React : un framework JavaScript populaire pour construire des interfaces utilisateur réactives. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Vue.js : un autre framework JavaScript léger et facile à utiliser pour construire des PWA. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Angular : un framework plus complet | souvent utilisé pour des applications web plus complexes | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Svelte : un framework moderne qui compile les composants en JavaScript optimisé pour de meilleures performances. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 7. API Web |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Les PWA utilisent plusieurs API web pour améliorer l'expérience utilisateur et ajouter des fonctionnalités avancées : |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Geolocation API : pour intégrer des cartes ou la localisation géographique dans l’application. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Push Notifications API : pour envoyer des notifications push aux utilisateurs. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Background Sync API : pour permettre la synchronisation des données lorsque l'utilisateur est en ligne. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Web Payments API : pour intégrer des paiements en ligne directement depuis la PWA. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 8. Outils de développement et de déploiement |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Workbox : une bibliothèque JavaScript qui facilite l’utilisation des Service Workers et la gestion du cache. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Lighthouse : un outil de Google pour auditer les PWA et vérifier leur performance | leur accessibilité et leur conformité aux meilleures pratiques. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| PWA Builder : un outil en ligne pour générer un manifeste et une structure de PWA de base. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| 9. Backend et Serveur |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Bien que la PWA soit une application front-end | elle peut se connecter à des API backend via des requêtes HTTP pour récupérer des données en ligne. Les technologies backend couramment utilisées pour alimenter une PWA incluent : | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Node.js avec des frameworks comme Express pour créer des API REST. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| GraphQL pour une gestion plus flexible des données. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Firebase : pour des solutions en temps réel | un stockage facile et des notifications push. | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| Django ou Flask (Python) pour des applications backend. |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
| En résumé | une PWA repose principalement sur des technologies web standards (HTML | NaN | JavaScript) et utilise des outils comme les Service Workers et les Web App Manifests pour offrir une expérience proche de celle des applications natives. Le choix des frameworks et outils dépend de la complexité du projet et des préférences de développement | mais ces technologies sont essentielles pour développer une PWA performante et accessible. |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |
|  |  | NaN |  |  |  | NaN | NaN |  |  |