Web技術の分類

2024年1月18日　22:51

		技術	学習教材	
フロントエンド	クライアントのブラウザやWebアプリケーションで動作するユーザが直接操作するアプリケーションのクライアントアプリ	View: HTML、CSS、JavaScript フレームワーク: React、Angular、Vue.js Store/State: Flux, MobX, Redux	学習教材	
API・通信	ロントエンドとバックエンド間のデータの受け渡しや通信を行うためのAPIや通信制御	RestAPI・GraphQL・WebSocket・Ajax・Push Notificatigon	https://www.lambdanote.com/products/wbs-ebook https://www.udemy.com/course/bbkuomwx/	
バックエンド	サーバサードの開発。	• **サーバーサイド言語（例: Node.js、Python、Java** • **バックエンドフレームワーク（例: Express.js、Django、Spring）** •	https://www.udemy.com/course/api-django/ https://www.udemy.com/course/graphql-tutorial-with-newsapp-api/	
セキュリティ	• **HTTPS: セキュアな通信を確保するために、SSL/TLSプロトコルを使用したHTTPS接続が必要です。** • **認証と認可: ユーザー認証やアクセス制御の仕組みを構築し、セキュリティを確保します。** 貼り付け元 <https://chat.openai.com/c/1a0871ef-3d01-4674-b94f-12921b08ce05>		https://www.amazon.co.jp/OAuth%E5%BE%B9%E5%BA%95%E5%85%A5%E9%96%80-%E3%82%BB%E3%82%AD%E3%83%A5%E3%82%A2%E3%81%AA%E8%AA%8D%E5%8F%AF%E3%82%B7%E3%82%B9%E3%83%86%E3%83%A0%E3%82%92%E9%81%A9%E7%94%A8%E3%81%99%E3%82%8B%E3%81%9F%E3%82%81%E3%81%AE%E5%8E%9F%E5%89%87%E3%81%A8%E5%AE%9F%E8%B7%B5-Justin-Riche-ebook/dp/B07L5M7DXS	
テスト	• **アプリケーションの品質を担当し、テスト計画やテストケースを作成します。** • 自動テストツールも活用して、**ユーザビリティテストやパフォーマンステストを行う。**	**テスト自動化ツール（例: Selenium、JUnit、Jest）** **パフォーマンステストツール（例: Apache JMeter、LoadRunner）**	TypeScriptではじめるWebアプリケーションテスト入門 | Udemy	
DevOps	・自動化ツールを用いて開発・運用プロセスのインフラストラクチャの構築・運営。			

■Webアプリの種類
ハイブリッドアプリ
https://it-trend.jp/development_tools/article/32-0038

Webアプリケーションを開発するためには、さまざまな要素技術が必要です。以下に、基本的な要素技術をいくつか挙げてみますが、プロジェクトの要件によってはこれに加えて他の技術も必要になるかもしれません。

1. **フロントエンド技術:**
   - **HTML/CSS/JavaScript:** Webページの構造（HTML）、スタイリング（CSS）、動的な挙動（JavaScript）を担当します。
   - **フレームワーク（例: React, Angular, Vue.js）:** コンポーネントベースの開発を容易にし、効率的なUIの構築をサポートします。

2. **バックエンド技術:**
   - **サーバーサイド言語（例: Node.js, Python, Ruby, Java）:** ユーザーのリクエストを受け取り、処理を行うサーバーサイドのコードを書くための言語が必要です。
   - **フレームワーク（例: Express.js, Django, Ruby on Rails, Spring）:** サーバーサイド開発を効率化し、共通のタスクを簡略化します。
   - **データベース（例: MySQL, PostgreSQL, MongoDB）:** アプリケーションのデータを永続化し、管理するためのデータベースが必要です。

3. **API (Application Programming Interface):**
   - **RESTful APIやGraphQL:** フロントエンドとバックエンド間のデータの受け渡しや通信を行うためのAPIを設計・実装します。

4. **セキュリティ:**
   - **HTTPS:** セキュアな通信を確保するために、SSL/TLSプロトコルを使用したHTTPS接続が必要です。
   - **認証と認可:** ユーザー認証やアクセス制御の仕組みを構築し、セキュリティを確保します。

5. **デプロイとホスティング:**
   - **クラウドプラットフォーム（例: AWS, Azure, Google Cloud）:** アプリケーションをデプロイし、運用するためにクラウドサービスを利用します。
   - **コンテナ化技術（例: Docker, Kubernetes）:** アプリケーションとその依存関係をコンテナにまとめ、効率的にデプロイ・スケーリングするために使用します。

6. **開発ツールとバージョン管理:**
   - **IDE（Integrated Development Environment）:** 開発を効率化するための統合開発環境を使用します。
   - **バージョン管理ツール（例: Git）:** チームでの協力やコードの管理を円滑に行うためにバージョン管理が必要です。

これらの技術要素は、Webアプリケーションの開発において一般的に必要なものですが、プロジェクトの性質や要件によっては、これに加えて特定の技術やツールが必要になることがあります。