Web Development

* **Frontend development**: A frontend developer is, in large part, responsible for what a user sees. The 3 main tools a frontend developer works with are HTML, CSS, and JavaScript. Their primary focus should be to create a great user experience and ensure that the design and layout of the website or web app are cohesive throughout.
* **Backend development**: A backend developer on the other hand works on the nuts and bolts of what goes on behind the scenes. Backend developers work with a wide range of libraries, APIs, web services, etc. They are responsible for the implementation of database systems, ensuring proper communication between various web services, generating backend functionality, and more.

# Framework

## Frontend Frameworks:

1. React
2. Angular
3. Vuejs
4. jQuery
5. Emberjs
6. Backbonejs
7. Semantic-UI
8. Foundation
9. Svelte

**React** is an open-source framework that stands out because of its virtual Document Object Model (DOM), which offers its exceptional functionality. It is used for building the user interface, especially when you want to develop single-page applications. It is the most robust frontend framework when you want to develop an interactive interface with less time since you can reuse the components.

Pros:

* The reusability of components makes it easy to collaborate and reuse them in other parts of the application.
* Consistent and seamless performance with the use of virtual DOM.
* The best alternative to writing components in React hooks, it allows you to write components without classes and lets you learn React more easily.
* React dev tools are advanced and super useful.

## Backend Framework:

|  |  |  |
| --- | --- | --- |
| **Framework** | **Programming Language** | **Famous Use Cases** |
| Django | Python | Instagram  Pinterest  Coursera  Mozilla |
| Laravel | PHP | Deltanet Travel  Neighborhood Lender  MyRank |
| Ruby on Rails | Ruby | ZendDesk  Shopify  GitHub |
| ExpressJS | NodeJS | MySpace  GeekList  Storify |
| Flask | Python | Red Hat  Rackspace  Reddit |
| Asp .NET | C# | Microsoft  Godaddy  Ancestry |
| Spring Boot | Java | Trivago  Via Varejo  Intuit |
| Koa | NodeJS | – |

**Django** is a high-level Python framework that is built with the idea of "batteries included". Meaning almost everything a developer would want is included out of the box. Therefore there is less a need for third party plugins and everything in Django works together. Django however, is built for larger applications. Therefore, if you're planning on building something small, Django may not be the best option as it can make a small project bloated with unnecessary features.

Features:

* Highly customizable
* No need to reinvent the wheel, encourages rapid development
* Very scalable
* Extensive community and documentation

For smaller projects, it is good to go with Flask.

* Very flexible
* More lightweight than Django, great for smaller projects
* Great documentation
* Offers the ability to built prototypes quickly

## Styling Framework:

1. Bootstrap - The most widely used free and open-source CSS framework

1) Powerful responsive design

2) Built-in libraries of resources

3) Low learning curve

4) Quickly build prototypes

5) Good documentation

2. Foundation - The most advanced responsive front-end framework in the world But it is a way more sophisticated framework.

3. Pure - The lightweight CSS framework designed keeping mobile in mind

4. Semantic UI - A development framework using human-friendly HTML

5. UI kit - A lightweight and modular front-end framework for creating fast and powerful web interfaces. It offers almost all the features of other frameworks.

6. Materialize CSS - A modern responsive front-end framework based on Material Design

And others like - Milligram, Skeleton, Spectre, Base, Mustard UI

# List of Cloud Service Providers

1. Amazon Web Service (AWS)
2. Microsoft Azure
3. Google Cloud Platform
4. IBM Cloud Services
5. VMware
6. Red Hat
7. Salesforce
8. Oracle Cloud
9. SAP

# Selection of Frontend Framework:

* Server-side rendering
* Must be flexible
* Support for mobile development
* Performance and productivity
* Future support for the framework

# Selection of Backend Framework:

* Core Library
* Vulnerability
* Learning Curve
* Documentation
* Hosting Mode

# Considerations When Choosing a Database

* Data Model
* Data Consistency
* Data Security
* Data Protection
* Multi-Access and Integration
* Efficiency
* Usability
* Implementation and Service Costs

# Wireframe Tools

Sketch (macOS) for passing wireframes off to third-party apps

InVision Studio (macOS) for wireframing for multiple screen sizes

Adobe XD (macOS, Windows) for a complete wireframe > prototype tool

Figma (Web, macOS, Windows, Linux) for real-time collaboration

Balsamiq (macOS, Windows, Web) for beginners

Proto.io (Web) for user experience testing

Moqups (Web) for wireframing interaction design flows

Adobe Illustrator (macOS, Windows, iOS) for detailed vector wireframes

Adobe Photoshop (macOS, Windows, iOS) for detailed pixel-based wireframes

Marvel (Web) for pre-designed wireframe templates

Canva (Web) for quick marketing wireframes and visual hand-offs