

FULLSTACK DEVELOPMENT

ELYSIUM ACADEMY SPARK NOTES

VERSION 2.0

01. Overview of Full Stack Development

- **Full Stack Developer** - A programmer skilled in both front-end and back-end development, capable of building a complete web application.
- **Front-End** - The part of a web application that interacts directly with the user.
- **Back-End** - The server-side part of a web application, handling business logic, database interactions, and server configuration.

02. Front-End Development

- **Languages** -
 - HTML - Structures web content.
 - CSS - Styles the web content (layout, colors, fonts).
 - JavaScript - Adds interactivity to web pages.
- **Frameworks/Libraries**
 - React.js - A JavaScript library for building user interfaces.
 - Angular - A TypeScript-based framework by Google for building
- **Web Applications** -
 - Vue.js - A progressive JavaScript framework for building UIs.
 - Bootstrap - A CSS framework for responsive design.
- **Tools** -
 - Webpack - Module bundler for JavaScript applications.
 - Babel - JavaScript compiler that converts ES6+ code to a backwards-compatible version.
 - Sass/Less - CSS preprocessors that add features like variables and nesting to CSS.

03. Back-End Development

- **Languages -**

- o JavaScript (Node.js) - Server-side JavaScript environment.
- o Python (Django/Flask) - High-level programming language; Django is a full-fledged framework, Flask is micro.
- o Ruby (Rails) - A web application framework written in Ruby.
- o React.js - A JavaScript library for building user interfaces.
- o PHP - Server-side scripting language; commonly used with WordPress.
- o Java (Spring) -): A robust framework for building enterprise-level applications.

- **Databases -**

- o SQL Databases -

- MySQL - Popular open-source relational database.
- PostgreSQL - Advanced open-source relational database.
- SQLite - Lightweight, file-based database.

- o NoSQL Databases -

- MongoDB - Document-oriented NoSQL database.
- Cassandra - Wide-column store NoSQL database for large-scale data.
- Redis - In-memory data structure store, used as a database, cache, and message broker.

- **APIs -**

- REST (Representational State Transfer) - Web service architecture style using HTTP requests.
- GraphQL - Query language for APIs that allows clients to request specific data.
- SOAP (Simple Object Access Protocol) - Protocol for exchanging structured information in web services.

- **Server Management -**

- Express.js - Web application framework for Node.js.
- Nginx/Apache - Web servers that serve web content.
- Docker - Platform for developing, shipping, and running applications in containers.
- Kubernetes - Container orchestration platform for managing containerized applications.

04. Full Stack Frameworks

- **MEAN Stack -**

- MongoDB - Database.
- Express.js - Back-end framework.
- Angular - Front-end framework.
- Node.js - Runtime environment.

- **MERN Stack -**

- MongoDB - Database.
- Express.js - Back-end framework.
- React.js - Front-end library.
- Node.js - Runtime environment.

- **LAMP Stack -**

- Linux - Operating system.
- Apache - Web server.
- MySQL - Database.
- PHP/Python/Perl - Scripting language.

- **Django Stack -**

- Django - Back-end framework (Python).
- PostgreSQL - Database.
- React/Vue.js - Front-end library/framework.

05. Version Control

- **GIT** - Database.
- **GitHub/GitLab/Bitbucket** - Platforms for hosting Git repositories and collaborating on code.

06. DevOps and CI/CD

- **DevOps** - Practices that combine software development (Dev) and IT operations (Ops).
- **CI/CD (Continuous Integration/Continuous Deployment)** - Automates the process of software testing, building, and deployment.
 - Jenkins - Automation server for CI/CD.
 - Travis CI/CircleCI - Continuous integration services.
 - Docker - For containerization in CI/CD pipelines.
 - Kubernetes - Manages containerized applications in CI/CD environments.

07. Testing

- **Unit Testing**- Testing individual components of a software.
 - Jest (JavaScript) - Testing framework for JavaScript
 - JUnit (Java) - Testing framework for Java applications.
- **Integration Testing** - Testing the integration of multiple components.
- **End-to-End Testing** - Testing the flow of an application from start to finish.
 - Selenium - Tool for automating web browsers.
 - Cypress - Front-end testing tool.

08. DeploymentCD

- **Hosting Providers -**
 - AWS (Amazon Web Services) - Cloud platform for deploying applications.
 - Heroku - Platform-as-a-service (PaaS) for deploying applications.
 - Netlify - Hosting for static websites and serverless functions.
 - Vercel - Deployment platform for front-end frameworks and static sites.

09. Security

- **OWASP Top 10 -** Standard awareness document for developers and web application security.
- **SSL/TLS -** Protocols for securing communication over a computer network.
- **JWT (JSON Web Tokens) -** Compact, URL-safe means of representing claims to be transferred between two parties.
- **OAuth -** Open standard for access delegation, commonly used for token-based authentication.

10. Tools and IDEs

- **VS Code** - Popular code editor for multiple programming languages.
- **WebStorm** - Integrated development environment for JavaScript.
- **PyCharm** - IDE for Python development.
- **Postman** - Tool for testing APIs.
- **Docker** - Containerization platform.
- **Webpack** - Module bundler for JavaScript applications.

This spark notes provides a quick reference guide for full stack development, covering key concepts, technologies, tools, and best practices needed to build, deploy, and maintain web applications.

*Thank
you*
For Your Learning Today

 [elysiumacademy.org](mailto:info@elysiumacademy.org) |  info@elysiumacademy.org

Scan Here for More
Spark Notes

