4. Introduction to React

4.1 Static vs Dynamic Content

- Static Content → Images, Text, Videos: Displayed as it is.
- Dynamic Content → Displays content based on HTTP Request
  - Web Server: Communicates with Application Server (Back-End): To Generate dynamic content
  - The App Server performs several tasks such as
    - Executing application logic
    - Communicating with Database
    - Checking permissions
- To process dynamic content → Application Server tend to have limited capability
  - A limited number of requests per second can be dealt with
- To resolve such issues: Web Browsers use the concept of caching
  - Caching: a simple copy of the dynamic content which is available on request
  - To minimize the load on App Server → The web server sends a cached copy of the content

---

4.2 Single-Page Application (SPA)

- Traditionally: Applications are Multi-Page
  - It becomes resource intensive for Web Servers to process the application
  - Multi-Page applications consume high bandwidth and CPU to generate content
  - Complex websites will load slower on a poor internet connection
- Single Page Application
  - Comprises of Single HTML Page
  - Two approaches to serve code and resources
    - Bunding
      - Browser requests the application and the web server returns all potential files (HTML, CSS, and JavaScript) necessary for execution
    - Lazy Loading (Code Splitting)
      - Minimum resources are communicated
      - More resources could be provided as requested
  - In a traditional website - a new resource is created every time a request is made
  - In SPA - JSON Object is sent for every POST request
  - SPA uses templates (views) : basically HTML code which changes to represent the overall view

---

4.3 What is React?

- Library, not a Framework, not a MVC framework
  - Less code for more development
  - Maintains code in a better format - simplify testing
  - Component-based Structure
    - Small pieces of user-interface
  - Isolated development and testing
  - Enhances re-usability of component

- React doesn’t use templates
  - Traditionally templates can dictate what can be done and what not
  - However, React provides ability to build components as per the requirements
  - Uses a real and fully-featured language: JavaScript
  - Flexible-Powerful language → Facilitates in building abstractions
- Easier to extend and maintain view logic
  - Unifies markup and view logic
- No manual string concatenation
  - Less XSS (Cross Site Scripting) Vulnerability

4.4 The Virtual DOM

- Updating DOM is resource- and time-intensive
- For every change → DOM is recomputed
- React creates a Virtual DOM rather than updating the real Browser DOM
  - React updates the Virtual DOM by comparing the previous version
  - V-DOM - Allows to update certain elements - rather than re-create the entire page
- React Fiber Architecture (RFA)
  - Many updates performed at once can still cause performance issues
  - using RFA
    - React can spread the changes over-time → Changes occur in incremental model
    - Think of RFA as a priority system

4.5 Alternative Libraries

- Lodash
- Luxon
  - To work with dates and times

    - can allow to display dates in local user format
- Redux
    - State management in React application
    - Comprises of advanced features
- AXIOS
    - Simplifies the process of HTTP requests and response processing
    - Advanced features of cancelling requests
- Jest
    - Helps in writing automates tests for code