

ENSF 381

Full Stack Web Development

Lecture 18: Front-end Frameworks

Slides: Ahmad Abdellatif, PhD

Instructor: Novarun Deb, PhD

Outline

- Front-end frameworks.
- Benefits of using front-end frameworks.
- Popular front-end frameworks.
- Terminologies.

What are front-end frameworks?

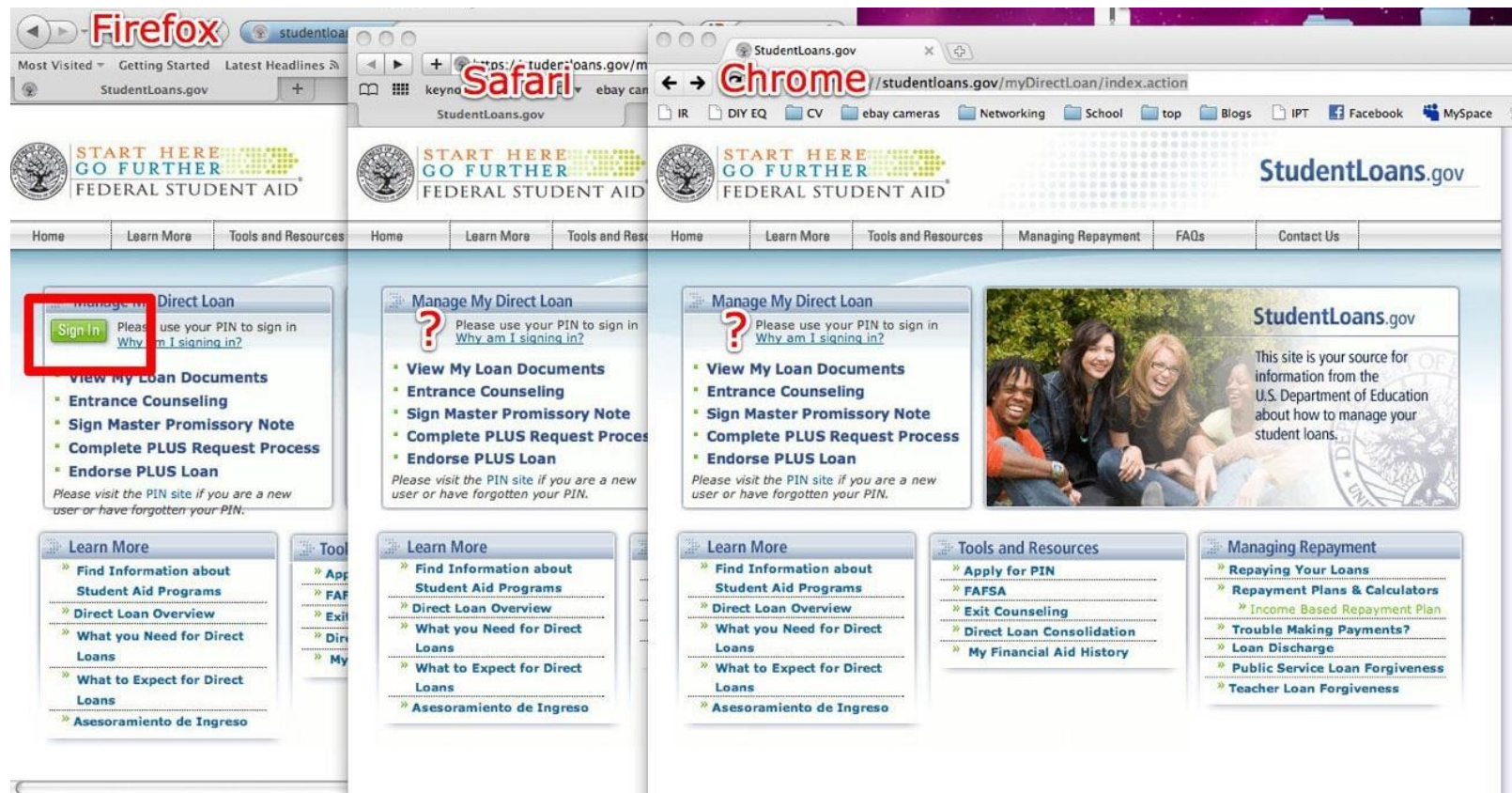
- Front-end frameworks are predefined, standardized code libraries or structures that expedite and streamline the web development process.
- Over the years, frontend frameworks have evolved to meet the demands of modern web applications, incorporating best practices and innovative solutions.

Benefits of using front-end frameworks

- **Code Organization and Structure:** frameworks provide a structured and organized way to develop frontend applications.
- **Reusability and Modularity:** components, a fundamental concept in many frameworks, encourage code reusability.
- **Consistent UI/UX Design:** frameworks often come with design systems and UI components that ensure a consistent look and feel across the application.
- **Improved Performance:** optimizations, such as virtual DOM in React, contribute to improved performance by minimizing unnecessary re-renders.
- **Cross-Browser compatibility:** frameworks handle browser-specific quirks, ensuring a consistent experience across different browsers.

Cross-Browser compatibility

- Challenge of Cross-Browser compatibility:
 - Browsers may render the same code differently.
 - Developers need to test and ensure compatibility with multiple browsers.



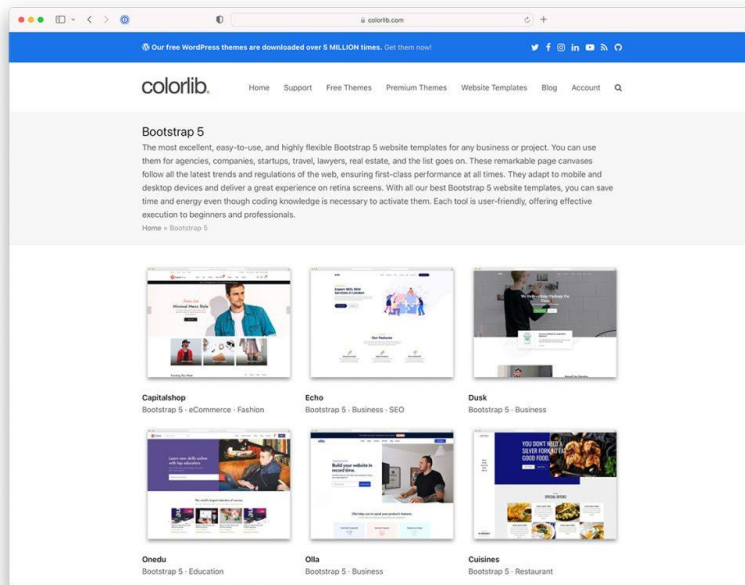
Popular front-end frameworks

- Front-end frameworks contribute to maintaining consistency across various browsers.
- There are many front-end frameworks such as:
 - Bootstrap
 - React
 - Angular
 - Vue.js
 -
 -
 -

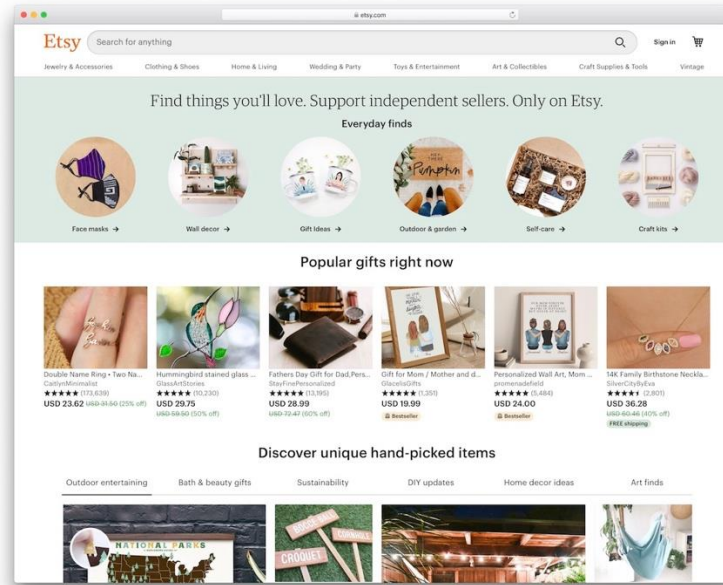
Popular front-end frameworks - Bootstrap

- An open-source front-end framework developed by Twitter.
- Bootstrap offers a rich set of pre-designed components (buttons, navigation bars, forms) that can be easily customized.
- Originally created to streamline internal tools, it became immensely popular for its simplicity and consistency.
- Bootstrap makes it easy to create responsive designs with classes that adjust based on the device's screen size.

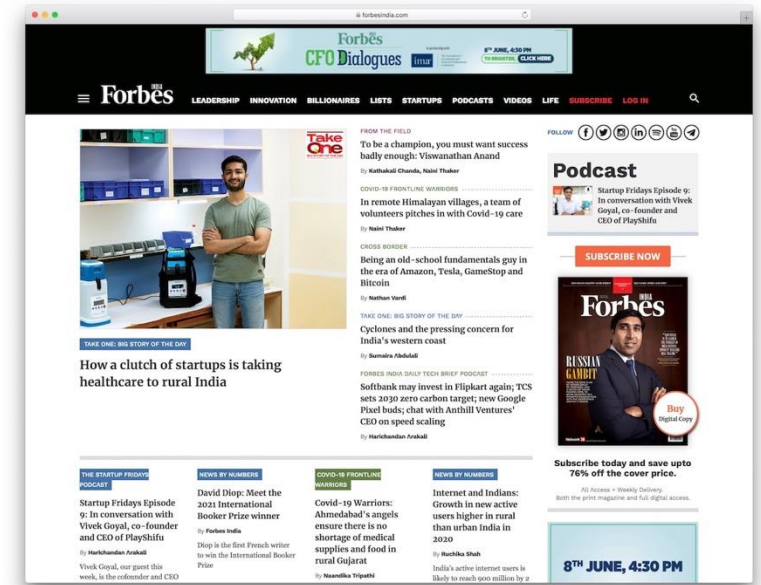
Examples



Colorlib



Etsy



Forbes India

Popular front-end frameworks - React

- A powerful and widely-used JavaScript library for building user interfaces, developed by Facebook.
- Uses a declarative syntax that makes it easy to understand and debug.
- React applications are built using components, encapsulating their own state and rendering logic.
- Focuses on building reusable UI elements.

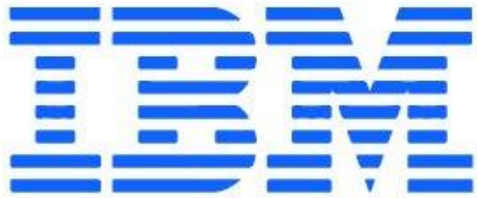
Examples



Popular front-end frameworks - Angular

- A TypeScript-based open-source framework developed by Google.
- A full-featured framework that includes tools for building both small and large-scale applications.
- Angular directives extend HTML with additional attributes and tags such as **ngIf**.
- Like React, Angular follows a component-based architecture.

Examples



**The
Guardian**

Popular front-end frameworks - Vue.js

- A progressive JavaScript framework for building user interfaces.
- Developers can choose to use as much or as little of the framework as they need, and they can integrate it into their projects gradually.
- Known for its simplicity and ease of integration.

Examples

trivago®



Adobe

Nintendo

Use Cases

Bootstrap:

- Rapid prototyping or development.
- Projects where a consistent, recognizable design is acceptable.

React:

- Projects requiring high interactivity and real-time updates.
- Large-scale applications with a focus on component reusability.

Use Cases

Angular:

- Enterprise-level applications with complex requirements.
- Projects where two-way data binding is crucial.

Vue.js:

- Projects of varying scale due to its progressive nature.
- Teams with diverse skill levels, as it is approachable for beginners.

Questions...

- Is it possible to incorporate multiple front-end frameworks simultaneously?

Yes

- What are the potential benefits and challenges associated with adopting such an approach in web development?

- **Benefits:**

Different frameworks can be chosen for their strengths, allowing for a modular approach to development where each part of the application is built with the most suitable tool.

- **Challenges:**

- Complexity: integrating and managing multiple frameworks introduces complexity in terms of development workflows, debugging, and maintenance.
 - Performance Overhead: loading multiple frameworks can result in a performance overhead, impacting the overall speed and efficiency of the website.

Things to consider when selecting the framework

- **Project Requirements:** evaluate the specific needs of your project, considering factors like complexity, scalability, and real-time updates.
- **Performance:** evaluate the performance of each framework, considering factors such as rendering speed, optimization techniques, and the impact on user experience.
- **Flexibility and Customization:** examine the flexibility and customization options offered by each framework.
- **Learning Curve:** assess the learning curve associated with each framework. Consider the expertise of your development team and the time available for learning.
- **Community Support:** examine the size and activity of the framework's community. Active communities often provide valuable resources, support, and updates.

Things to consider when selecting the framework

- **Integration with other Technologies:** consider how well each framework integrates with other technologies, libraries, or systems your project relies on.
- **Scalability:** assess the scalability of each framework to ensure it can accommodate future growth and handle increasing complexities in your application.
- **Documentation:** well-documented frameworks are easier to learn, troubleshoot, and maintain. Clear documentation contributes to a smoother development process.

Terminologies

- **Single Page Application (SPA)**: a web application or website that interacts with the user by dynamically rewriting the current page rather than loading entire new pages.
- **User Experience (UX) Design**: designing the overall experience a user has with a product, including usability and user satisfaction.
- **User Interface (UI) Design**: designing the visual aspects of a website or application, focusing on the presentation and interactivity.

Terminologies

- **Responsive Web Design**: a design approach that ensures a web application's layout adapts to different screen sizes and devices. Critical for providing a consistent user experience across desktops, tablets, and mobile devices.
- **Progressive Web Apps (PWA)**: web applications that provide a native app-like experience with features like offline access, push notifications, and fast loading.
- **Web Accessibility**: ensures that websites and web applications are usable by various types of users.

Questions

