THE NEW COLLEGE (AUTONOMOUS), CHENNAI-14.

DEPARTMENT OF COMPUTER APPLICATIONS – SHIFT - II

STUDY MATERIAL

SUBJECT: OPEN SOURCE TECHNOLOGY SUB CODE: 20BHM615

CLASS: III BCA STAFF: Dr. K. SANKAR

UNIT – I

1. INTRODUCTION OF JAVASCRIPT

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript,** but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform

CLIENT-SIDE JAVASCRIPT

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

ADVANTAGES OF JAVASCRIPT

- **Less server interaction** − you can validate user input before sending the page off to the server. This saves server traffic, which means fewer loads on your server.
- **Immediate feedback to the visitors** − they don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity** − you can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** − you can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

LIMITATIONS OF JAVASCRIPT

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features −

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multi-threading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

2. JAVASCRIPT FRAMEWORKS

2.1 What is a JavaScript Framework?

Frameworks provide developers with the basic foundation necessary for building JavaScript applications. This saves developers the effort of starting from scratch by utilizing a functional base to get things rolling. In JavaScript's case, this base includes a collection of code libraries. The libraries compile code that elicits specific functionality for the specific type of app you may be working on. In essence, the framework will define the structure of the entire application.

2.2 What Does a JavaScript Framework Do?

Each JavaScript framework serves a different purpose. JavaScript is a steadfast choice for web development and many of its frameworks revolve around this undertaking. Building web apps and websites from start to finish can take quite a bit of work. Web frameworks – or really, JavaScript frameworks – take advantage of the fact that every website and web app has common features.

In this way, you can think of the term framework quite literally. Houses also rely on frameworks as a skeleton for the overall support and shape of an outer framing. Software frameworks do the same.

JavaScript frameworks render pre-written JavaScript code that produces routine programming features, ultimately easing development.

3. NEEDS OF JAVASCRIPT FRAMEWORKS

JavaScript frameworks are an essential part of modern front-end web development, **providing developers with tried and tested tools for building scalable, interactive web applications**.

1. ANGULAR

- **Progressive Web Apps:** Has modern web platform capabilities to deliver app-like experiences that are high performance, offline, and zero-step installation.
- **Desktop:** Desktop-installed apps across Mac, Windows, and Linux can be created using the same Angular methods like the web plus the ability to access native OS APIs.
- **Code Generation:** Angular turns templates into code that's highly optimized for JavaScript virtual machines, giving hand-written code benefits.
- **Code Splitting:** With the new Component Router, angular apps load quickly, delivering automatic code-splitting.
- **Templates:** Create UI views with simple and powerful template syntax.
- **Angular CLI:** Command line tools allow you to start building fast, add components and tests, and then instantly deploy.
- **Animation:** Create high-performance, complex choreographies, and animation timelines with very little code through Angular's intuitive API.
- **Accessibility:** Create accessible applications with ARIA-enabled components, developer guides, and built-in test infrastructure.

2. REACT

- **Declarative:** Creates interactive and dynamic UI for websites and mobile applications. Declarative views make the code readable and easy to debug.
- **Virtual DOM:** For every DOM object, there is a corresponding "virtual DOM object." It creates a virtual copy of the original DOM and is a representation of a DOM object.
- **Event handling:** React has its fully compatible W3C object model event system created. It also provides a cross-browser interface to a native event.
- **JSX:** JSX is a markup syntax that closely resembles HTML. JSX makes writing React components easier by making the syntax almost identical to the HTML injected into the web page.
- **Performance:** React uses one-way data binding with an application architecture called Flux controls. ReactJS helps update the View for the user with Flux controlling the application workflow.

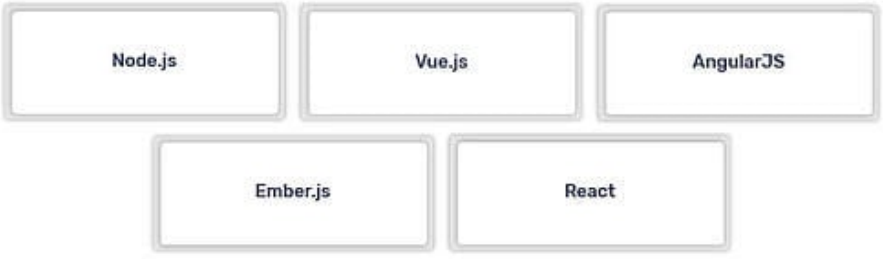
- **React Native:** React Native is a custom renderer for React; it uses native components instead of web components like React as building blocks.
- **Component-Based:** Everything is a component of the web page, divided into small components to create a view(or UIs). Components in ReactJS are used to define the visuals and interactions in applications.

3. NODE.JS

- **Fast:** The library of Node.js is fast when it comes to code execution, as it is built on the V8 JavaScript engine of Google Chrome.
- **Asynchronous and Event-Driven I/O:** All the APIs are asynchronous, which means that its server does not wait for the API to come back with data.
- **Single-threaded:** Node.js, along with event looping, follows a single-threaded model.
- **Highly scalable:** Node.js follows an event mechanism that makes it possible for the server to respond in a non-blocking manner, which makes it scalable.
- **No buffering:** When it comes to uploading audio and video files, Node.js cuts down processing time significantly. It does not buffer any data, and the application gets the data out in chunks.
- **Open source:** Being open-source, Node.js community has come up with several amazing models that can be used to add more capabilities to the Node.js applications.

4. TYPES OF JAVASCRIPT FRAMEWORKS

The 5 most popular JavaScript frameworks are **Node.js, Vue.js, Angular.JS, Ember.js and React**.



1. Node.js

Node.js is not exactly a JavaScript framework; it's a runtime environment. While JavaScript can be written directly into the web browser, this is not always desirable. This is why Node.js lends the capacity for command-line tools and server-side scripting.

Though JavaScript usually operates on the client-side or the front-end, server-side scripting begets faster load times as browser technology is not needed. This can decrease user frustration and increase SEO.

2. Vue.js

Vue.js calls itself the "progressive" JavaScript framework. The name stems from its philosophy of incremental adoption. In Vue.js, the core library is focused on the view layer only so any additional functionality must be adopted in increments.

The framework uses a model-view-view model (MVVM) architectural pattern. This pattern separates the graphical user interface (UI) – or the view – from the business logic of the application – or the model. The view model layer is a converter medium that synchronizes data.

3. AngularJS

AngularJS is maintained by Google and addresses common complications in building single-page applications (SPAs). This framework works by leveraging HTML vocabulary on dynamic web pages. In the past, HTML could only be used for static content.

SPAs work by dynamically loading content from the web server rather than the web browser. As a result, SPAs function in a similar fashion to mobile applications and do not need to be reloaded.

4. Ember.js

Ember.js is a JavaScript framework employing a component-service pattern. Opposed to the traditional model-view-controller (MVC) architecture, components in Ember.js are central to the framework.

Almost everything in Ember.js can be categorized as a service or component.

Components are transient and manipulate the markup text and styles of an application UI. Services are objects that live for the duration of an application. They can be made available for different parts of your applications and are best used for persistent states.

5. React

React is a JavaScript framework developed by Facebook that simplifies the process of building interactive UIs. It is the base of React Native, an adjacent framework for building mobile applications.

Both frameworks have a one-way data flow, which is considered more intuitive than bi-directional data binding. Hot reload is another popular feature of the React frameworks allowing developers to immediately see changes as they are applied.

5. COMPARISON OF JAVASCRIPT FRAMEWORKS

This is a comparison of web frameworks for front-end web development that are heavily reliant on JavaScript code for their behavior.

Framework	Version compared	Size	License	Source language
Angular	14.0.4 29 June 2022	563 kB (minified & compressed)	MIT	TypeScript
AngularJS	1.5.0 5 Feb 2016	144 kB (minified & compressed)	MIT	JavaScript
Backbone.js	1.2.1 June 2015	7.3 kB (Packed and gzipped)	MIT	JavaScript
Dojo	1.10.4 18 Jan 2015	Variable. Base size: 41 kB (minified & gzipped), 155 kB (minified), 598 kB (uncompressed)[1]	BSD & AFL	JavaScript + HTML
jQuery (library)	3.6.0 3 Mar 2021	70.7KB (slim, minified), 87.4KB (minified),[4] 282 KiB (uncompressed)[4]	MIT	JavaScript

Parameter	React	VueJs	Angular	NodeJs	EmberJs
Developed by	Facebook and Community	Evan You	Google	OpenJS Foundation	Ember Core Team
Type	JavaScript library	JavaScript framework	Web framework	Runtime environment	JavaScript library
Initial Release	May 29, 2013	February 2014	October 20, 2010	27 May 2009	8 December 2011
Programming language	JavaScript	JavaScript	Typescript	JavaScript	JavaScript
DOM	Virtual DOM	Virtual DOM	Real DOM	VM Module	Virtual DOM

--- End of Unit-I ---