

Web Application to Search and Rank Online Learning Resources from Various Sites

Keshav J¹, Venkat Kumar B M¹, Ananth N¹

¹Student, Department of Computer Engineering, Velammal Engineering College, Anna University, Chennai, Tamil Nadu, India.

Corresponding Author: ananthkalki@gmail.com

Abstract: - E-Learning is becoming one of the most reliable and fast methods for learning in the present days and this trend will only continue to grow in the coming years. However, the process of finding the correct course in the desired topic is becoming harder and harder day by day due to the sheer number of resource material being uploaded to the internet. Currently, whenever there's a need to acquire a new skill or learn something new, people go on different platforms to search for resources to learn them - YouTube for videos, Google for blogs, Coursera/Udemy/Edx for courses. This leads to taking too much time in finding the right resource to start learning. They have to try out multiple resources before finding the right one. We plan to eliminate this with our website, where a single search query will fetch the top videos, blogs, and courses for that query. Users will be able to rate the results, and the upcoming search results will be ordered primarily based on the user ratings. This will allow the most useful blogs, courses, and videos to rank up. Additionally, the resources will have comments and tags, so that the user can read a review before opening a tutorial. The services such as the APIs and the technique of Web Scraping are used to achieve this goal.

Key Words: — *Web Development, API, Web Scraping, E-Learning.*

I. INTRODUCTION

Web development is the work involved in developing a Web site for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing a simple single static page of plain text to complex web applications, electronic businesses, and social network services. A more comprehensive list of tasks to which Web development commonly refers may include Web engineering, Web design, Web content development, client liaison, client-side /server-side scripting, Web server and network security configuration, and e-commerce development.

The first major challenge is that retrieving the data from the various websites and other resources such as blogs. Various sites may contain data in different structures and to retrieve them various techniques must be used. The next major

challenge is to rank each resource based on its ratings. The next challenge is figuring out how the data are going to be stored in the database and what database should be used for optimal performance.

The Final and important challenge to be tackled is how to rate the resources. But except for the courses from e-learning sites all other resources are not rated. So, this poses the challenge of rating the resource which is not rated already.

Web development is the building and maintenance of websites; it's the work that happens behind the scenes to make a website look great, work fast and perform well with a seamless user experience.

Modern web development involves using JavaScript at a level of ambition for which it was not originally intended, but for which it is very rapidly adapting itself. The use of JavaScript "frameworks" such as React and Angular has become standard because these provide so much functionality and context that most developers would be incapable of creating for themselves. Thus, for very many people, "modern web development" means writing web applications within the APIs of JavaScript frameworks, and perhaps using other advanced JavaScript tools

Manuscript revised July 31, 2021; accepted August 02, 2021.
Date of publication August 03, 2021.
This paper available online at www.ijprse.com
ISSN (Online): 2582-7898

(like TypeScript) that also sit “on top of” what is often called “vanilla JavaScript.”

This is only the “front end” side of the equation. The application is written to be downloaded into the user's browser and executed there. But the server-side of the equation (“back end”) is also rapidly evolving. It used to be that the browser side app did little more than present an interface, and the server did everything including composing the contents of the page being viewed in every respect. With the rise of what is called the “Single Page Application” (SPA), the server-side simply provides data as requested by the client-side “front-end” application, which takes care of all aspects of the display. The SPA is driven by the need for the mobile user to make as minimal and as efficient use of data transactions with the server as possible, and let the “front end” application in the mobile device handle as much of the work as possible. This arose naturally in native mobile apps that don't run in the browser and so the present effort is to duplicate that model as closely as possible with “web apps” running in a mobile browser. The move from native apps (downloaded from a “store” and installed locally) to web apps that require no such installation and can be easily linked to (and thus discovered in search engines) is fundamental, irreversible, and certain. It's just a question of web app performance catching up entirely with native app performance.

II. WEB DEVELOPMENT

There are 2 major parts to Web Development. They are 1.Front-End 2.Back-End

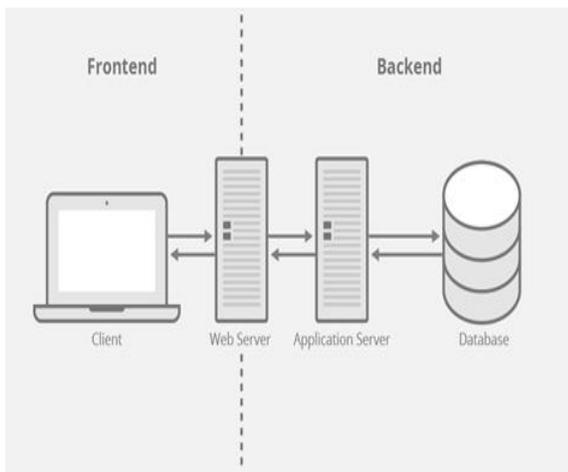


Fig.1. Major parts of web development

A. Front-End

Front-end web development, also known as client-side development is the practice of producing HTML, CSS, and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front-end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

The objective of designing a site is to ensure that when the users open up the site, they see the information in a format that is easy to read and relevant. This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions thus forcing the designer to take into consideration these aspects when designing the site. They need to ensure that their site comes up correctly in different browsers (cross-browser), different operating systems (cross-platform), and different devices (cross-device), which requires careful planning on the side of the developer.

A front-end framework is a scaffold for building your front end. It usually includes some way to structure your files (for example, via components or a CSS preprocessor), make AJAX requests, style your components, and associate data with DOM elements.

Different frameworks address some, but usually not all, of these questions. Some, like Bootstrap and SemanticUI, focus on creating readable, maintainable HTML and CSS, emphasizing consistent visual design. Others, like Vue, React, and Angular, triumph at structuring data flow throughout your application, allowing you to focus on manipulating the data rather than the DOM.

B. Back-End

The *backend* (or “server-side”) is the portion of the website you don't see. It's responsible for storing and organizing data and ensuring everything on the client-side works. The backend communicates with the front-end, sending and receiving information to be displayed as a web page.

Whenever you fill out a contact form, type in a web address, or make a purchase (any user interaction on the client-side), your browser sends a request to the server-side, which returns information in the form of frontend code that the browser can interpret and display. A new site will need to have additional

backend components to make it a *dynamic web application* — a website whose content can change based on what is in its *database*, and that can be modified by user input. This is distinct from a static website, which doesn't require a database because its content generally stays the same.

Backend frameworks are libraries of server-side programming languages that help build the backend structure of a website. Backend frameworks provide ready-made components for developing a dynamic web application. Using frameworks gives a head-start to the developers by eliminating the need to build and configure everything from the ground up.

Back-end development includes the server implementation and it more on the logical interaction of data, how it is stored, and transmitted. Back-end developers typically work with programming languages such as PHP, Python, Go, and new Javascript libraries like Node.

III. FRAMEWORKS

A. Angular – Front- End Framework

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your apps.

The architecture of an Angular application relies on certain fundamental concepts. The basic building blocks of the Angular framework are Angular components that are organized into *NgModules*. *NgModules* collect related code into functional sets; an Angular app is defined by a set of *NgModules*. An app always has at least a *root module* that enables bootstrapping and typically has many more *feature modules*.

Components define *views*, which are sets of screen elements that Angular can choose among and modify according to your program logic and data.

Components use *services*, which provide specific functionality not directly related to views. Service providers can be *injected* into components as *dependencies*, making your code modular, reusable, and efficient.

Modules, components, and services are classes that use *decorators*. These decorators mark their type and provide metadata that tells Angular how to use them.

The metadata for a component class associates it with a *template* that defines a view. A template combines ordinary HTML with Angular *directives* and *binding markup* that allow Angular to modify the HTML before rendering it for display.

The metadata for a service class provides the information Angular needs to make it available to components through *dependency injection (DI)*.

An app's components typically define many views, arranged hierarchically. Angular provides the Router service to help you define navigation paths among views. The router provides sophisticated in-browser navigational capabilities.

B. Node.js – Back-End Framework

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. *Node.js* lets developers use JavaScript to write command-line tools and for server-side scripting running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, *Node.js* represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server-side and client-side scripts.

Though `.js` is the standard filename extension for JavaScript code, the name "Node.js" doesn't refer to a particular file in this context and is merely the name of the product. *Node.js* has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

The *Node.js* distributed development project was previously governed by the *Node.js* Foundation and has now merged with the JS Foundation to form the OpenJS Foundation, which is facilitated by the Linux Foundation's Collaborative Projects program.

Node.js brings event-driven programming to web servers, enabling the development of fast web servers in JavaScript.

Developers can create scalable servers without using threading, by using a simplified model of event-driven programming that uses callbacks to signal the completion of a task. Node.js connects the ease of a scripting language (JavaScript) with the power of Unix network programming.

Node.js was built on top of Google's V8 JavaScript engine since it was open-sourced under the BSD license. It is proficient with internet fundamentals such as HTTP, DNS, and TCP. JavaScript was also a well-known language, making Node.js accessible to the web development community.

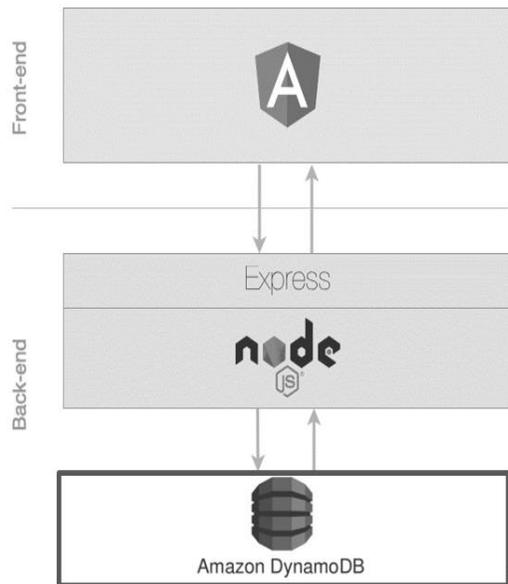


Fig.2. Node.js Framework

IV. RESULTS AND DISCUSSION

Our proposed system works as follows; Users have to Sign Up or Sign-in to the site to use the Course Finder application. After signing in, the user is presented with a search bar where they can enter the resource they want to search. The input can be a specific topic they want to learn or a domain they want to explore.

The resources they want can be in the form of blog websites, videos or courses on e-learning websites. When the user searches a resource, the results will be displayed in an ordered list based on their rank, and the results will be categorized based on the type of the resource, namely - blogs, courses, videos. Then the user can visit the desired resource from either of the three categories. After consuming the resource, the user can rate

and review it. These ratings will be taken into account when ranking resources.

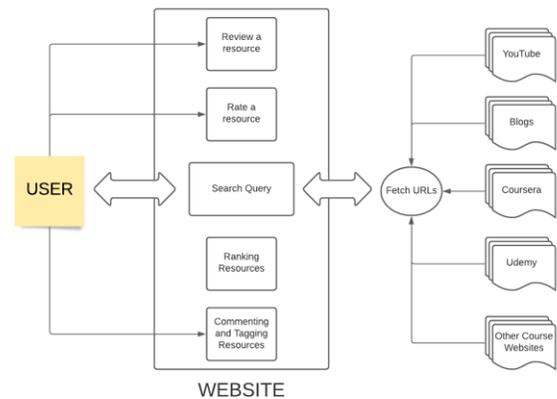


Fig.3. Design flowchart

V. CONCLUSION

The Web application to search and rank online resources from various sites explained in this report uses the fast and reliable method that can replace the conventional and time-consuming process. This requires no additional hardware or software to function. Any computer or smartphone is enough to reach the website.

REFERENCES

- [1]. S. R. Thakkar and H. D. Joshi, "E-Learning Systems: A Review," 2015 IEEE Seventh International Conference on Technology for Education (T4E), Warangal, India, 2015, pp. 37-40.
- [2]. S. K. Gudla, J. Bose and K. R. Sane, "Enhanced Service Recommender and Ranking System Using Browsing Patterns of Users," 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2019, pp. 1-2.
- [3]. Adomavicius, G. and Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on knowledge and data engineering, 2005.
- [4]. Sun J, Ma J, Liu X, Liu Z, Wang G, Jiang H, Silva T. A novel approach for personalized article recommendation in online scientific communities. Proc 46th IEEE HICSS 2013.
- [5]. Lago Vilarino, A.B.; Garcia, I.P., "An E-Learning Platform for Integrated Management of Documents Based on Automatic Digitization," IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, vol.8, no.2, pp.48-55, May 2013.
- [6]. Jiahui Liu, Peter Dolan, and Elin Ronby Pedersen. Personalized news recommendations based on click behavior.

15th International Conference on Intelligent User Interfaces (IUI). 2010. ACM.

- [7]. Daniela Godoy, Analia Amandi. Learning Browsing Patterns for Context-Aware Recommendation. In: Bramer M. (eds) Artificial Intelligence in Theory and Practice. International Federation for Information Processing (IFIP), Vol 217. 2006 Springer.
- [8]. Kenneth Fee, "Delivering E-learning - A complete strategy for design, application and assessment", Kogan Page, 2009.
- [9]. Hamada, M., "An Integrated Virtual Environment for Active and Collaborative e-Learning in Theory of Computation", IEEE Transactions on Learning Technologies, vol.1, no.2, pp.117-130, April-June 2008.
- [10]. Christos Troussas; Maria Virvou; Efthimios Alepis, "Comulang: towards a collaborative e-learning system that supports student group modeling", SpringerPlus, Vol. 2, 2013.