IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 7, July 2022

DOI: 10.17148/IJARCCE.2022.11780

Analysis of different Web Development Frameworks

Chinmay Naik

Student, Dept of Information Science and Engineering, R.V. College of Engineering, Bengaluru, India

Abstract: In recent years, there has been a tremendous growth in online applications, especially web applications. From things like net banking or online banking, healthcare or even social media platforms with billions of people worldwide are now using web applications and mobile applications. These applications not only ease our lives but also offer business values. It also offers seamless user experience with its creative user interfaces. All our day-to-day activities become dependent on these applications. And with the development of so many technologies, it is very important to choose the best possible technology or framework. This includes choosing the front-end framework, database and back-end framework. The most trending front-end frameworks in the market are React, Angular, Vue and OJET. While in back-end frameworks Node.js is very popular which uses JavaScript and Django is also gaining popularity and its written in python. Databases can be chosen based on requirements of the project and the scope of the project and can be broadly chosen between either SQL or NoSQL. To develop the API, one can use the REST API principles or can choose GraphQL based on the complexity of the project. These complex web applications can be single page applications (SPA's) or multi page applications (MPA's). When we include all these technologies to build the application, it is called a full stack application. There are a lot of combinations of web stacks like MEAN, MERN etc. This paper discusses the popular web frameworks in the front-end and back-end and database environments.

Keywords: Front-end frameworks, React, Angular, OJET, Back-end frameworks, Node.js, Django, Databases, MySQL, MongoDB.

I. INTRODUCTION

In this fast-moving world of technology, where web applications have become essential parts of our lives, choosing the perfect architecture for your service is quite a tedious task. In the microservice architecture-based approach, a collection of services is developed independently which are easy to develop, easy to maintain and deploy. From the web-based application perspective, we have to choose the front-end framework to design the user interface, database to store the relevant data needed for the website or the data entered by the user and a back-end framework. To communicate between the front-end and back-end we use middleware which consists of an API. The API architecture can be based on the project requirements.
The rest of the paper discusses various front-end and back-end technologies and its use cases with section II detailing about the former while section III the latter. Section IV compares traditional relational databases with the NoSQL database.

II. FRONT-END FRAMEWORKS

React: React is basically a JavaScript library for building dynamic user interfaces. React makes building user interfaces incredibly easy and really intuitive especially when building complex interfaces. It also develops a mindset for the developers to shift from a top to bottom based workflow to developing independent components and state-based approaches. The components are the most significant aspect of React. React helps in creating creative interactive UI's and it is declarative in nature which allows React to efficiently update and render the right UI components when the data is changed. Another important advantage of using react is that it can also render on the server using Node and can be used to create mobile applications using react native.
Complex UI's can be broken down into small, reusable, independent code called "components". React uses JSX or JavaScript XML where essentially one is writing the code in HTML while JSX acts as a precursor which allows us to write HTML in react.

Angular: Angular is developed and maintained by developers at google. Angular is a typescript and JavaScript based web framework which can be used to develop dynamic single page web applications (SPA's). Angular, much similar to React, is a component-based framework. Additional characteristics are introduced to HTML by Angular. Angular creates a proper structure for your application which helps in visualizing and creating your web application much faster and cleaner. Since angular uses typescript instead of JavaScript, it helps developers in detecting bugs during compile time thus reducing overall development time. Angular uses a MVC architecture which stands for Model-View-Controller where Data is managed by the Model and View manages the data display. While the controller serves as a link between the view and model layers. Large apps can be created with Angular in a manageable way.

OJET: Oracle JavaScript Extension Toolkit is a front-end JavaScript framework which can be used to build reusable web components, user interfaces and websites. OJET applications can be developed in virtually any environment that supports JavaScript (or typescript), HTML and CSS. JET applications can be developed using an IDE or any text editor. OJET also provides command-line tooling. After creating a project, the jet application gives you a defined structure of the project so that you can organize your code. Ojet, similar to react, supports hybrid development i.e., one can create web based and mobile applications using OJET. The Oracle JET framework uses the model-view-ViewModel (MVVM) architectural design pattern. The Model in MVVM stands for the data of the app, while the View is how the data is displayed. The ViewModel keeps track of the app's state and provides data from the Model to the view.

TABLE I COMPARISON TABLE FOR FRONT-END FRAMEWORKS

Parameter	Angular	React	OJET
Overview	Full-fledged JS framework	JS library for UI development	JS framework to build engaging UIs for web and mobile and hybrid
Maintained by	Google	Meta	Oracle
Language	Typescript	JavaScript, JSX	JavaScript, Typescript
Data binding	two-way	one-way	two-way
DOM	HTML DOM	Virtual DOM	Virtual DOM
App Size	Small	Relatively small	Relatively small
Project complexity suited for	Very complex	Complex	Relatively complex
Ideal for	Highly active and interactive web applications	Large web applications with frequently variable data	Ideal framework for building hybrid cross-platform mobile applications.

III. BACK-END FRAMEWORKS

Node.js: Node.js is an event-driven JavaScript runtime environment which is used to build scalable network applications. Node.js contains a V8 JavaScript engine, the predefined set of libraries and packages and a few binary files. In reality, it is more complex than that. These packages and libraries present default in Node.js are managed by the node package manager, or in short, npm. These libraries can be used to create low level APIs for example the “fs” module of the Node.js. Node.js uses threads for its execution. So, the advantage is that one doesn’t have to wait for a complex time-consuming process as our JavaScript code is not blocked. Since these operations are running in the background, there needs to be a confirmation mechanism to know when these processes get executed which is called a “callback”. The callback is a JavaScript function, which gets executed once the operation is finished. Owing to its single-threaded nature, Node.js is typically used for non-blocking, event-driven servers. Although it was created with real-time, push-based architectures in mind, it is also incorporated for conventional web pages and back-end API services.

Django: Django is a python based high speed MVC framework for building web applications. The model contains all the business logic. It's going to communicate with the database, represent your objects, and parse all the data. The view contains templates that describe how the UI should look like. The controller is going to give you an interface to define all your routes that your application has to do or even dynamic URLs. Advantages of this framework are - it helps in reducing the development time and encourages rapid development. Django also provides better security compared to Node.js, it provides service of sessions, authentication etc. Django also has a huge set of extensive libraries that can be used as a plugin which will reduce developers' effort.

TABLE II COMPARISON TABLE FOR BACK-END FRAMEWORKS

Parameter	Node.js	Django
Programming Language	JavaScript	Python
Type	Run time environment	Web framework
Architecture	Event-driven model	Model-Template-View model
Flexibility	More flexible	Less flexible
Security	Less secure	More secure
Scalability	Scalable	More scalable
Complexity	Less Complex	Relatively more complex

IV. DATABASES

MySQL: MySQL database is a relational database that is quite efficient and popular for deploying cloud applications. It's a free to use open-source database which stores structured data in the form of tables containing rows and columns. It executes queries to get data from the database and leverages RDBMS to enforce referential integrity between table rows. Thus, SQL databases promise features like accessibility and consistency in terms of storage and query handling. MySQL requires the use of schema to be defined. MySQL supports the use of foreign key, so performing join operations is quite easy and convenient. With all its advantages, a SQL database is prone to a lot of attacks commonly called SQL injection attack.

MongoDB: MongoDB is classified as a NoSQL database paradigm where MongoDB uses JSON like objects with optional schemas. Traditional and relational data that architectures' "rows" and "columns" are replaced with "documents" in MongoDB. This gives developers the ability to cope with evolving data models. MongoDB doesn't require any schema as the data is stored in the form of key value pairs. MongoDB doesn't use the concept of foreign keys and uses aggregation methods instead of traditional join operations as in the SQL database. Since MongoDB doesn't follow any kind of schema and the data is stored as documents, it is much less prone to security attacks.

TABLE III COMPARISON TABLE FOR DATABASE SYSTEMS

Parameter	MySQL	MongoDB
Schema	Requires definite schema	Doesn't require a definite schema
Language	Supports Structured Query Language (SQL)	Supports JSON/BSON Query Language to work with data
Data	Data stored as tables	Data stored as collections

DOI: 10.17148/IJARCCE.2022.11780

Scale	Not so easy to scale (Scales vertically)	Easy to scale (Scales both horizontally and vertically)
Foreign Key	the use of foreign keys is supported	usage of foreign keys is not supported
Replication	supports both master-master and master-slave replication	Supports sharding and replication
Join Operation	Supports Join operation	Doesn’t support Join operation
Performance	Computation is faster	Computation optimized for write performance
Risks	Prone to SQL injection attacks	There is no schema, hence there are fewer vulnerabilities.

V. CONCLUSION

Each database, back-end framework, and front-end framework has its own set of benefits and drawbacks. Therefore, when selecting the framework for the project's execution, various factors such as scalability, accessibility, maintenance, time before the product is released to the market, developers' understanding of that technology, documentation available and community support must all be taken into account. Databases can be selected based on the types of data that will be used in a certain application, such as structured, unstructured, or semi-structured data. Additionally, important performance issues like consistency, availability, and partition tolerance need to be considered carefully.
Prior to developing any substantial websites, it is crucial to take into account all of these variables.

REFERENCES

[1]. Curie Dasari & Jaison, Joyce & Yadav, Jyoti & Fiona, J. (2019). Analysis on Web Frameworks. Journal of Physics: Conference Series. 1362. 012114. 10.1088/1742-6596/1362/1/012114.
[2]. P. Di Francesco, "Architecting Microservices," 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, 2017, pp. 224-229
[3]. Kaluža, Marin & Vukelic, Bernard. (2018). “Comparison of front-end frameworks for web applications development” Zbornik Veleučilišta u Rijeci. 6. 261-282. 10.31784/zvr.6.1.19.
[4]. Gackenheimer C., 2015. “Introducing Flux: An Application Architecture for React. In: Introduction to React”. Apress, Berkeley, CA.
[5]. Teixeira, P., 2012. “Professional Node. js: Building JavaScript based scalable software”. John Wiley & Sons.
[6]. Romanyuk, O. (2020, March 19). Angular vs React: Which One to Choose for Your App. freeCodeCamp.Org. Accessed on 03 July 2022 [online] Available at: https://www.freecodecamp.org/news/angular-vs-react-what-to-choose-for-your-app-2/
[7]. OJET cookbook. (2014). Oracle JET Cookbook. Accessed on 03 July 2022 [online] Available at: https://www.oracle.com/webfolder/technetwork/jet/jetCookbook.html