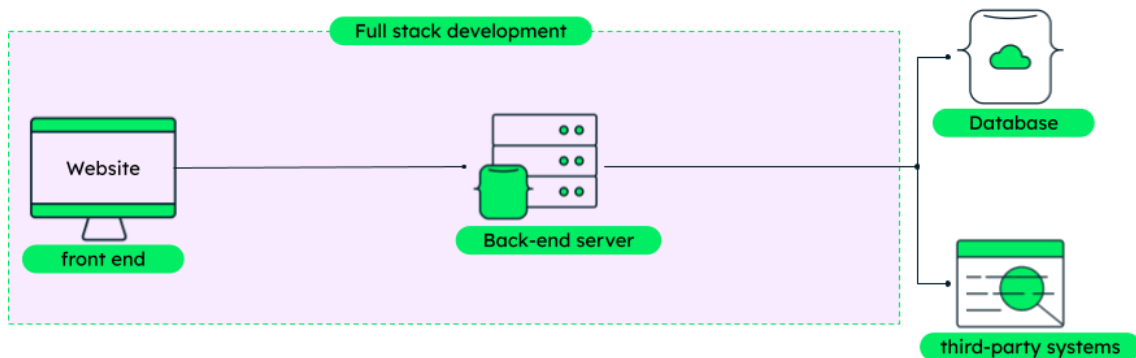


## Unit\_01 - Introduction to Full Stack Development Frameworks

### Full Stack Development

- Full Stack Development refers to the software development process of both front end (client side) and back end (server side) portions of web application.
- “Full-stack development” refers to a collection of abilities and skills that are an indispensable precondition to developing web-based applications and websites on both the front end and back end.
- Full stack development is the process of designing, creating, testing, and deploying a complete web application from start to finish.
- It involves working with various technologies and tools, including front-end web development, back-end web development, and database development. And full stack development is a term used to describe a software engineer or developer who works with both the front and back end of a website or application.
- A full-stack developer is comfortable working with front-end and back-end technologies that power a website or application.
- A full stack developer is one who can single-handedly implement both the front-end and back-end workflows, like placing the order or changing the user profile.



*Fig: Demonstration of full stack development in an end-to-end workflow*

- Full-stack developers are often responsible for the entire web application development process, from start to finish, which means they must have a strong understanding of all the technologies and tools involved in web development. They also need to work effectively with others on a team, as web development is typically a collaborative process.
  - Most full-stack developers have a firm foundation in web development technologies, such as HTML, CSS, and JavaScript. They also have experience with server-side technologies such as PHP, Ruby on Rails, and Node.js. In addition to their technical skills, full-stack developers also deeply understand how the various parts of a website or application work together.
-

## Need of Full Stack Development

In a modern world with a lot of technologies, instead of depending on a number of software professionals separately for front-end and back-end tasks to handle, a need of full stack development is required who can handle all the technologies required for developing a software project. This includes a number of advantages like:

### ➤ Saves Money and Time

- Full-stack development is cost-effective when the same developer or team works on both the front and back end.
- Hiring a full-stack developer lowers the cost of developing an application because the developer is conversant and proficient with both the frontend and the backend technologies. Therefore, businesses need not invest twice as much in designing an application.
- It also facilitates adding fewer individuals to the project team, reducing operational and contractual expenses.
- Full-stack developers have the ability to save business money and resources as they can work on both the front end and the back end of a project on their own. Most full-stack engineers (55.2% back-end and 37.1%) claim to be in the latter category, according to research conducted by Stack Overflow in 2020.

### ➤ Rounded Solution and Adaptable

- However, we also agree that a person with knowledge of all levels can create a comprehensive solution with fewer bugs and greater efficiency.
- In general, frontend and backend development are proficient in their respective technologies. Still, the primary advantage of full-stack development is that the developers are proficient in both frontend and backend technologies and can construct complicated apps efficiently from scratch.
- As well as, the full-stack developer is capable of doing all the duties of a backend developer and a frontend developer alone. This makes troubleshooting the code and testing the product significantly simpler.

### ➤ Unique Codes

- No longer must the team's project manager coordinate with the frontend and backend developers to make the program function.
- Full-stack developers can now develop applications and write flawlessly unique scripts relatively easily.
- This eliminates the requirement to mix and connect code from two distinct application development ends.

➤ **Superior Scalability**

- Full-stack development has the added advantage of enhancing the scalability of an organization's applications.
- Scalability is essential because it enables businesses to manage higher loads or traffic without requiring costly program modifications.
- By increasing the scalability of their apps, businesses may ensure that they can continue to satisfy their customers' needs even as they expand.
- Full-stack development can enhance scalability by providing a robust and versatile architecture.
- With this type of development, adding new features and capabilities to your system as needed is simple.

➤ **Enhanced Speed to Market**

- Another significant advantage of full-stack development is that it expedites product launches for businesses. That said, this particular aspect can prove out be advantageous for enterprises operating in a competitive market, allowing them to launch their items before their competitors.
- Moreover, full-stack development can cut the amount of time spent on development initiatives by enterprises.
- This can free up time and resources for firms to concentrate on other aspects of their operations.

➤ **Enhanced Client Satisfaction**

- Full-stack development can assist firms in developing more user-friendly and customer-centric applications.
- Consequently, organizations that implement full stack development will have a major competitive edge in terms of consumer experience.

➤ **Quick Solutions to Errors**

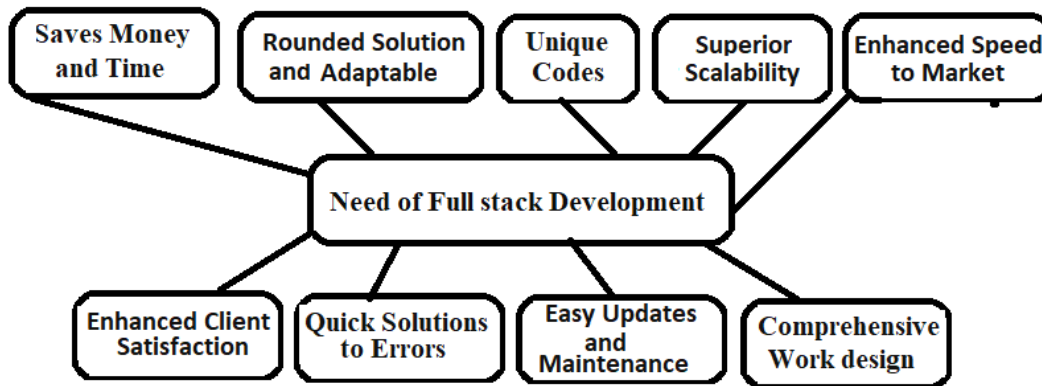
- Occasionally, during the creation of an application, things do not function properly, possibly due to a coding error. Having said that, this process can be handled fast because a full-stack developer is familiar with the arrangement of programs and can easily discover errors.

➤ **Easy Updates and Maintenance**

- Developing apps using full-stack technology enables a developer to issue updates based on customer requirements simply.
- As they are involved in both the frontend and backend of the application, they are familiar with the codes. They can optimize them when the app requires updates and the code demands an additional feature accessible on the web and mobile application.

➤ **Comprehensive Work design**

- While collaborating, members of a team of Full Stack developers can split the design and development task according to the flow's requirements.
- In addition to this, a single Full Stack developer has the ability to switch between tasks in accordance with the requirements of the project at hand.



*Fig: The need of full stack development*

---

## Introduction to Front end Environment Tools

- Front end is the visible part of website or web application which is responsible for user experience. The user directly interacts with the front end portion of the web application or website.
- The front end is a combination of two different elements: the graphic design (the look) and the user interface (the feel). Each of these is created independently, with most of the technical work going into the user interface using web languages like HTML, CSS, and JavaScript.

➤ **Front End Languages**

The front-end portion is built by using some languages which are discussed below.

### 1. HTML

- HTML stands for Hypertext Markup Language. It is used to design the front-end portion of web pages using a markup language. HTML is a combination of Hypertext and Markup language.
- Hyper Text: HyperText simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.
- Markup language: A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

- Web Page: A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. With the help of HTML only, we can create static web pages.

### **Features of HTML**

- It is a very easy and simple language. It can be easily understood and modified.
- It is very easy to make an effective presentation with HTML because it has a lot of formatting tags.
- It is a markup language, so it provides a flexible way to design web pages along with the text.
- It facilitates programmers to add a link on the web pages (by html anchor tag), so it enhances the interest of browsing of the user.
- It is platform-independent because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.
- It facilitates the programmer to add Graphics, Videos, and Sound to the web pages which makes it more attractive and interactive.
- HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.

## **2. CSS**

- CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.
- CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

### **Advantages of CSS**

- CSS saves time – You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- Easy maintenance – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- Global web standards – Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.
- Platform Independence – The Script offer consistent platform independence and can support latest browsers as well.

## **3. 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.

### 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 – It validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- Immediate feedback to the visitors
- Increased interactivity – It is used to create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- Richer interfaces – JavaScript used to include such items as drag-and-drop components and sliders to give a Rich Interface to the website visitors.

## ➤ Front-End Frameworks and Libraries

### 1. AngularJS

- AngularJs is a JavaScript open-source front-end framework that is mainly used to develop single-page web applications (SPAs). It is a continuously growing and expanding framework which provides better ways for developing web applications. It changes the static HTML to dynamic HTML. It is an open-source project which can be free. It extends HTML attributes with Directives, and data is bound with HTML.

### Advantages of AngularJS over other JavaScript frameworks:

- **Dependency Injection:** Dependency Injection specifies a design pattern in which components are given their dependencies instead of hard coding them within the component.
- **Two way data binding:** AngularJS creates a two way data-binding between the select element and the orderProp model. orderProp is then used as the input for the orderBy filter.

- **Testing:** Angular JS is designed in a way that we can test right from the start. So, it is very easy to test any of its components through unit testing and end-to-end testing.
- **Model View Controller:** In Angular JS, it is very easy to develop application in a clean MVC way. You just have to split your application code into MVC components i.e. Model, View and the Controller.

## 2. React.js

- React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based front-end library responsible only for the view layer of the application. It is maintained by Facebook. Moreover, React Js makes Front-end development very easy.

### Advantages of ReactJS:

- **Composable:** We can divide these codes and put them in custom components. Then we can utilize those components and integrate them into one place.
- **Declarative:** In react the DOM is declarative. We can make interactive UIs by changing the state of the component and React takes care of updating the DOM according to it.
- **SEO Friendly:** React affects the SEO by giving you a SPA (Single Page Application) which requires Javascript to show the content on the page which can be rendered and indexed.
- **Community:** React has a huge community because of it's demand each company wants to work with React. Companies like Meta, Netflix, etc built on React.
- **Easy to learn:** HTML-like syntax of JSX make you comfortable with the codes of React, it only requires to need a basic knowledge of HTML, CSS, and JS fundamentals to start working with it.

## 3. Bootstrap

- Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

### Advantages of Bootstrap

- It is Faster and Easier way for Web-Development.
- It creates Platform-independent web-pages.
- It creates Responsive Web-pages.
- It designs responsive web pages for mobile devices too.
- It is a free and open-source framework
- Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera).

## 4. jQuery

- jQuery is an open-source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM),

and JavaScript. Elaborating on the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.

### **Advantages of jQuery**

- Wide range of plug-ins that allows developers to create plug-ins on top of the JavaScript library.
  - Large development community.
  - It is a lot easier to use compared to standard javascript and other javascript libraries.
  - It lets users develop Ajax templates with ease. Ajax enables a sleeker interface where actions can be performed on pages without requiring the entire page to be reloaded.
  - Being Light weight and a powerful chaining capabilities makes it stronger.
- 
- Some other libraries and frameworks are Semantic-UI, Foundation, Materialize, Backbone.js, Ember.js, etc.
- 

## **Introduction to Backend Environment Tools**

The back end refers to the server-side development of web application or website with a primary focus on how the website works. It is responsible for managing the database through queries and APIs by client-side commands. This type of website mainly consists of three parts front end, back end, and database.

➤ **The back-end portion is built by using some languages which are discussed below:**

### **1. PHP**

- PHP is an acronym for "PHP: Hypertext Preprocessor".
- PHP is a server-side scripting language, which is used to manage the dynamic content of the website.
- Since PHP code is executed on the server side, it is called a server-side scripting language.
- PHP is an object-oriented language also it can be embedded into HTML.

### **Why use PHP**

PHP is a server-side scripting language, which is used to design the dynamic web applications with MySQL database.

- It handles dynamic content, database as well as session tracking for the website.
- You can create sessions in PHP.
- It can access cookies variable and also set cookies.
- It helps to encrypt the data and apply validation.
- PHP supports several protocols such as HTTP, POP3, SNMP, LDAP, IMAP, and many more.
- Using PHP language, you can control the user to access some pages of your website.
- As PHP is easy to install and set up, this is the main reason why PHP is the best language to learn.



- PHP can handle the forms, such as - collect the data from users using forms, save it into the database, and return useful information to the user. For example - Registration form.

### Features of PHP

PHP is very popular language because of its simplicity and open source. There are some important features of PHP given below:

- **Performance:** PHP script is executed much faster than those scripts which are written in other languages such as JSP and ASP. PHP uses its own memory, so the server workload and loading time is automatically reduced, which results in faster processing speed and better performance.
- **Open Source:** PHP source code and software are freely available on the web. You can develop all the versions of PHP according to your requirement without paying any cost. All its components are free to download and use.
- **Familiarity with syntax:** PHP has easily understandable syntax. Programmers are comfortable coding with it.
- **Embedded:** PHP code can be easily embedded within HTML tags and script.
- **Platform Independent:** PHP is available for WINDOWS, MAC, LINUX & UNIX operating system. A PHP application developed in one OS can be easily executed in other OS also.
- **Database Support:** PHP supports all the leading databases such as MySQL, SQLite, ODBC, etc.
- **Error Reporting:** PHP has predefined error reporting constants to generate an error notice or warning at runtime. E.g., E\_ERROR, E\_WARNING, E\_STRICT, E\_PARSE.
- **Loosely Typed Language:** PHP allows us to use a variable without declaring its datatype. It will be taken automatically at the time of execution based on the type of data it contains on its value.
- **Web servers Support:** PHP is compatible with almost all local servers used today like Apache, Netscape, Microsoft IIS, etc.
- **Security:** PHP is a secure language to develop the website. It consists of multiple layers of security to prevent threats and malicious attacks.
- **Control:** Different programming languages require long script or code, whereas PHP can do the same work in a few lines of code. It has maximum control over the websites like you can make changes easily whenever you want.

### 2. Java:

- Java is one of the most popular and widely used programming languages and platforms. It is highly scalable.

#### Features of Java

- **Platform Independent:** Compiler converts source code to bytecode and then the JVM executes the bytecode generated by the compiler. This bytecode can run on any platform be it Windows, Linux, or macOS which means if we compile a program on Windows, then we can run it on Linux and vice versa.

- **Object-Oriented Programming Language:** Organizing the program in the terms of a collection of objects is a way of object-oriented programming, each of which represents an instance of the class.
  - **Simple:** Java is one of the simple languages as it does not have complex features like pointers, operator overloading, multiple inheritances, and Explicit memory allocation.
  - **Secure:** Java is best known for its security. With Java, we can develop virus-free systems. Java is secured because:
    - No explicit pointer
    - Java Programs run inside a virtual machine sandbox
    - **Classloader:** Classloader in Java is a part of the Java Runtime Environment (JRE) which is used to load Java classes into the Java Virtual Machine dynamically. It adds security by separating the package for the classes of the local file system from those that are imported from network sources.
    - **Bytecode Verifier:** It checks the code fragments for illegal code that can violate access rights to objects.
    - **Security Manager:** It determines what resources a class can access such as reading and writing to the local disk.
    - Java language provides these securities by default. Some security can also be provided by an application developer explicitly through SSL, JAAS, Cryptography, etc.
  - **Multithreading:** Java supports multithreading. It is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of the CPU.
  - **Portable:** Java is portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.
  - **Dynamic:** Java is a dynamic language. It supports the dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.
    - Java supports dynamic compilation and automatic memory management (garbage collection).
3. **Python:**
- Python is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language.
  - It supports functional and structured programming methods as well as OOP.
  - It can be used as a scripting language or can be compiled to byte-code for building large applications.
  - It provides very high-level dynamic data types and supports dynamic type checking.
  - It supports automatic garbage collection.
  - It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**Advantages of python**

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the users to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – Low-level modules can be added to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

**4. Node.js:**

- Node.js is a very powerful JavaScript-based platform built on Google Chrome's JavaScript V8 Engine.
- It is used to develop I/O intensive web applications like video streaming sites, single-page applications, and other web applications.
- Node.js is open source, completely free, and used by thousands of developers around the world.

**Features of Node.js**

- **Easy Scalability:** Node JS is built upon Chrome V8's engine powered by Google. It allows Node to provide a server-side runtime environment that compiles and executes JavaScript at lightning speeds.
- **Fast Suite:** Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution. It provides microservice architectures in which the applications are breaking down into isolated and independent services.
- **No Buffering:** Node.js applications never buffer any data. These applications simply output the data in chunks.
- **Data Streaming:** NodeJS comes to the rescue since it's good at handling such an I/O process which allows users to transcode media files simultaneously while they are being uploaded. It takes less time compared to other data processing methods for processing data.

➤ **Back-End Frameworks:**

1. **Express :**

- Express is a Nodejs framework used for backend/server-side development.
- Express.js is a small framework that works on top of Node.js web server functionality to simplify its APIs and add helpful new features.
- It makes it easier to organize your application's functionality with middleware and routing. It adds helpful utilities to Node.js HTTP objects and facilitates the rendering of dynamic HTTP objects.
- It is used to build single-page, multi-page, and hybrid web applications. With its help, you can handle multiple different HTTP requests.

2. **Django:**

- Django is a Python web-based framework, following the model-template-views pattern. It is used to build large and complex web applications.
- Django emphasizes reusability of components
- Its features include being fast, secure, and scalable.
- Django follows the MVT design pattern (Model View Template).
  - ✓ Model - The data you want to present, usually data from a database.
  - ✓ View - A request handler that returns the relevant template and content - based on the request from the user.
  - ✓ Template - A text file (like an HTML file) containing the layout of the web page, with logic on how to display the data.

3. **Ruby on Rails:**

- Ruby on Rails or also known as rails is a server-side web application development framework that is written in the Ruby programming language.
- It supports MVC(model-view-controller) architecture that provides a default structure for database, web pages, and web services, it also uses web standards like JSON or XML for transfer data and HTML, CSS, and JavaScript for the user interface.
- The usage of different well-known web development patterns and frameworks, such as:
  - Don't Repeat Yourself (DRY): It is a software development philosophy that seeks to minimize the repeating of data or codes.
  - Convention Over Configuration (CoC): It offers a variety of viewpoints on the best ways to carry out certain tasks in a web-based application.

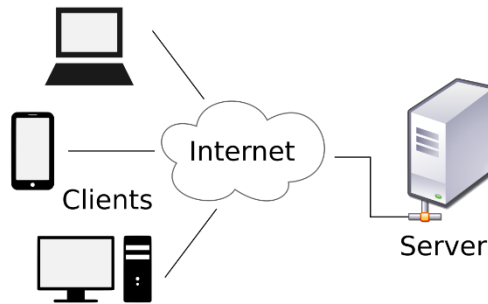
Some more back-end programming/scripting languages are spring, Laravel, C#, Ruby, GO, etc.

---

## **Client/Server architecture**

- A client - server architecture is a model in which computers such as servers provide the network services to the other computers such as clients to perform a user based tasks. This model is known as client-server networking model.

- The application programs using the client-server architecture should follow the given below strategies:



*Fig: Client - Server Model*

- An application program is known as a client program, running on the local machine that requests for a service from an application program known as a server program, running on the remote machine.
- A client program runs only when it requests for a service from the server while the server program runs all time as it does not know when its service is required.
- A server provides a service for many clients not just for a single client. Therefore, we can say that client-server follows the many-to-one relationship. Many clients can use the service of one server.
- Services are required frequently, and many users have a specific client-server application program. For example, the client-server application program allows the user to access the files, send e-mail, and so on. If the services are more customized, then we should have one generic application program that allows the user to access the services available on the remote computer.

#### **Client**

- A client is a program that runs on the local machine requesting service from the server. A client program is a finite program means that the service started by the user and terminates when the service is completed.

#### **Server**

- A server is a program that runs on the remote machine providing services to the clients. When the client requests for a service, then the server opens the door for the incoming requests, but it never initiates the service.
- A server program is an infinite program means that when it starts, it runs infinitely unless the problem arises. The server waits for the incoming requests from the clients. When the request arrives at the server, then it responds to the request.

### ➤ **Client-Server architecture example**

#### **1. Mail servers**

Email servers are used for sending and receiving emails. There are different software that allows email handling.

## 2. File servers

File servers act as a centralized location for files. One of the daily life examples to understand this is the files that we store in Google Docs. The cloud services for Microsoft Office and Google Docs can be accessed from your devices; the files that you save from your computer can be accessed from your phone. So, the centrally stored files can be accessed by multiple users.

## 3. Web servers

Web servers are high-performance computers that host different websites. The server site data is requested by the client through high-speed internet.

### ➤ Components of Client-Server architecture:

Essentially, three components are required to make client-server architecture work. The three components are workstations, servers, and networking devices. Let us, now, discuss them in detail:

- **Workstations**

Workstations are also called client computers. Workstations work as subordinates to servers and send them requests to access shared files and databases. A server requests information from the workstation and performs several functions as a central repository of files, programs, databases, and management policies. Workstations are governed by server-defined policies.

- **Servers**

Servers are defined as fast-processing devices that act as centralized repositories of network files, programs, databases, and policies. Servers have huge storage space and robust memory to deal with multiple requests, approaching simultaneously from various workstations. Servers can perform many roles, such as mail server, database server, file server, and domain controller, in client-server architecture at the same time.

- **Networking devices**

Now that we know about the roles that workstations and servers play, let us learn about what connects them, networking devices. Networking devices are a medium that connects workstations and servers in a client-server architecture. Many networking devices are used to perform various operations across the network. For example, a hub is used for connecting a server to various workstations. Repeaters are used to effectively transfer data between two devices. Bridges are used to isolate network segmentation.

### ➤ How does client-server architecture work?

A client-server architecture has a client node and a server node. While the server functions as the producer, clients are considered consumers. A consumer sends a request

to the server via a network. After processing, the consumer gets the final data per the submitted request.

Here is an explanation of the functioning of the client-server model:

- When a user puts in any file or website's URL, the browser sends this request to the Domain Name System (DNS) server.
- The DNS server looks for the web server's address.
- The DNS server replies with the IP address of the web server.
- The browser sends HTTPS or HTTP requests to the IP address of the web server.
- The server sends the required files for the website.
- The browser provides the files, and you have the website displayed. The rendering is completed with the help of a CSS interpreter, DOM interpreter and JS Engine.

### ➤ Types of client server architecture

There are three types of client server architecture available in database management system. These are listed below.

- (a) Single tier client server computing model
- (b) Two tier client server computing model
- (c) Three tier client server computing model

#### ✓ Single tier client server architecture.

Single tier architecture is the first type of client server computing model. In single tier client server computing model, the client server database system used on a personal computer. In single tier system, the database is centralized, which means the DBMS software and the data in one location and the dumb terminals were used to access the database management system.



#### ✓ Two-tier client server computing model

Two-tier client server architecture is the second type of client server computing model. It uses two computers and one client/server system. At the early stages, client server computing it was called two-tier computing model in which client is considered as data capture and validation tier and server was considered as data storage tier.



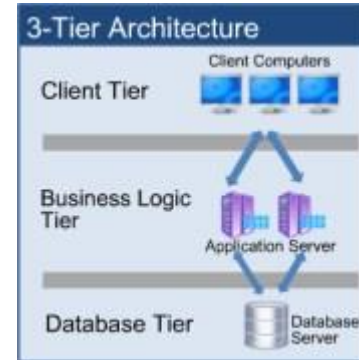
#### Problems in two tier architecture:

In 1990s, when applications were more complex and used with hundreds or thousands of end users, the two tier server model could not meet the need of the enterprise.

- (a) An efficient client requiring considerable resources on client's computer to run effectively. This includes disk space, RAM, CPU.
- (b) Client machines require administration which results overhead at the server end.

✓ **Three tier architecture.**

Three tier architecture is the improvement over two tier architecture. Three tier architecture has three layers. The first layer is the user interface which runs on client system. The second layer is called the application server. It is used for business logic and data processing. The third layer is known as database server. It is a database management system which stores the data as needed by the middle layer.



➤ **Advantages of Client-server architecture:**

- **Centralized:** Centralized back-up is possible in client-server networks, i.e., all the data is stored in a server.
- **Security:** These networks are more secure as all the shared resources are centrally administered.
- **Performance:** The use of the dedicated server increases the speed of sharing resources. This increases the performance of the overall system.
- **Scalability:** We can increase the number of clients and servers separately, i.e., the new element can be added, or we can add a new node in a network at any time.

➤ **Disadvantages of Client-Server architecture:**

- Traffic Congestion is a big problem in Client/Server networks. When a large number of clients send requests to the same server may cause the problem of Traffic congestion.
- It does not have a robustness of a network, i.e., when the server is down, then the client requests cannot be met.
- A client/server network is very decisive. Sometimes, regular computer hardware does not serve a certain number of clients. In such situations, specific hardware is required at the server side to complete the work.
- Sometimes the resources exist in the server but may not exist in the client. For example, if the application is web, then we cannot take the print out directly on printers without taking out the print view window on the web.

---

## Stack Frameworks

A stack framework is a software framework that provides a basic structure for web applications. It typically includes a web server, database, and programming language.



The web server handles requests from clients while the database stores data. The programming language is used to build the web application.

Stack frameworks are popular because they can be used to develop complex applications quickly. They can also be easily extended to add new features. However, stack frameworks can be difficult to learn and use and can be inflexible if the application needs to be customized.

## MEAN Stack

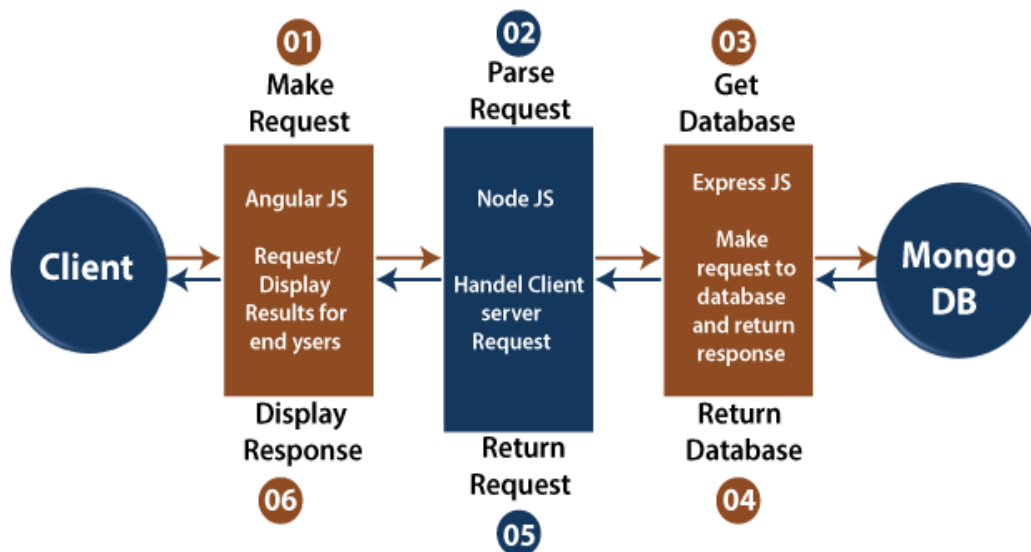
The MEAN stack is a JavaScript-based framework for developing web applications. MEAN is named after MongoDB, Express, Angular, and Node, the four key technologies that make up the layers of the stack.

- MongoDB – document database
- Express(.js) – Node.js web framework
- Angular(.js) – a client-side JavaScript framework
- Node(.js) – the premier JavaScript web server

This stack leads to faster development as well as the deployment of the Web Application. Angular is Frontend Development Framework whereas Node.js, Express, and MongoDB are used for Backend development.

### ➤ Mean Stack Architecture:

Here, each module communicates with the others in order to have a flow of the data from Server/Backend to Client/Frontend.



*Fig: Mean Stack Architecture*

- Initially, the client makes a request which is processed by the AngularJS

- After that, the request moves to NodeJS which will parse the request.
- ExpressJs will make calls to MongoDB to get or set data.
- MongoDB will retrieve the requested data and return that request to the Express JS
- NodeJS will return the request to the client.
- At the client side, AngularJS to display the result fetched from MongoDB.

### ➤ MongoDB

MongoDB is the database used in web development. It is a NoSQL database, and a NoSQL database can be defined as a non-relational and document-oriented database management system. As it is a document-oriented database management system, so it stores the data in the form of documents. The SQL databases use SQL query language to query the database, whereas the MongoDB is a NoSQL database that uses BSON language to query the database. JSON is a text-based format, but it is inefficient in terms of speed and space. In order to make MongoDB efficient in terms of space and speed, BSON was invented. BSON basically stores the JSON format in the binary form that optimizes the space, speed, and complexity.

Mongo DB is an ideal choice for a database system where you need to manage large sized tables with millions of data. Moreover, including a field to Mongo DB is easier as it does not require updating the entire table. With Mongo DB you develop an entire application with just one application, i.e. JavaScript.

### ➤ Express.js

Express is a mature, flexible, lightweight server framework. It is designed for building single, multi-page, and hybrid web applications. This lightweight framework uses the Pug engine to provide support for templates.

### Advantages of Express.js

- It is simple and lightweight software. It is not heavy to get installed in the machine and make the application running.
- It is easy to customize and configure as we can see it provides the flexibility that we require.
- It is a better choice for creating the API as when the application requires various APIs to communicate with different people then the Express.js is a good option

### ➤ Angular.js

- Angular JS is an open-source JavaScript framework. Angular is maintained by Google. The goal of this framework is to introduce MVC (Model View Controller) architecture in the browser-based application that makes the development and testing process easier. The framework helps the user create a smarter web app that supports personalization.
- AngularJS allows us to use HTML as a template language. Therefore, user can extend HTML's syntax to express the components of application.

Advantages of Angular.js

- It is a two-way data binding which means that it keeps the model and view in sync. If any changes are made in the model, then automatically view will also be updated accordingly.
- The Angular.js is designed with testing in mind. The components of angular.js application can be tested with both the testing, such as unit testing and end to end testing.
- With the help of Angular.js, it is easy to develop the application in an MVC architecture.

➤ **Node.js**

- Node.js is an open-source platform and provides a runtime environment for executing the javascript code. It is mainly used for building the back-end application.
- Since there are two types of apps, such as web apps and mobile apps, where web apps run on the browser and mobile apps run on mobile devices. Both web app and mobile app are the client apps to which the user interacts.
- These apps require to communicate with the backend services to store the data, send emails, push notifications.
- Node.js is an ideal platform to build highly scalable, data-intensive, and real-time applications. It can be used for agile development and highly-scalable services. For example, PayPal is a java and spring-based application using Node.js.

Advantages of Node.js

- The node.js applications are faster than the other framework-based applications and require fewer people to build the app.
- It requires fewer lines of code.
- The Node app has a 35% faster response time than the other apps.
- The major advantage of using node.js is that node.js uses javascript. If you are a front-end developer, then you can easily transit from the front-end to the full stack developer.

➤ **Advantages of MeanStack**

- **Simple and fast:** MeanStack is simple to use and fast as it allows the programmers to write the code in a single language for both the server and client-side.
  - **Universal coding is possible in MeanStack:** In MeanStack, the code written in one framework can be easily transferred to another framework.
  - **Highly flexible:** Once the development process of an application is completed, it is easier to test the application on the cloud platform.
  - **Cost-effective:** Since the mean stack uses single language, i.e., javascript so a smaller number of developers required to develop the app using mean stack.
  - **Open source:** All the technologies used in the mean stack are open-source and available for free.
-

## MERN Stack

MERN Stack is a collection of powerful technologies and robust, used to develop scalable master web applications comprising backend, front-end, and database components. It is JavaScript that is used for the faster and easier development of full-stack web applications. MERN Stack is a technology that is a user-friendly full-stack JavaScript framework for building applications and dynamic websites.

MERN Stack consists of **four main components** or can say four main technologies:

### 1. MongoDB

- MongoDB ( Database ), mainly used for preparing document database and is a NoSQL (Non-Structured Query Language ) Database System
- The term 'NoSQL' typically means a non-relational database that does not require a fixed schema or proper relational tables to store the necessary data in it. MongoDB stores the data in a different format other than the relational tables, consisting of rows and columns.
- It implies that MongoDB is not based on the table-like relational database structure. On the other hand, it provides an altogether different mechanism for the retrieval and storage of data.
- The storage format in which the data is stored is known as BSON, which stands for Binary JavaScript Object Notation; its binary structure encodes length and type of information, which allows it to be parsed much more quickly.
- It allows a highly scalable and flexible document structure.
- It is very faster as compared to RDBMS due to its efficient storage and indexing techniques.
- In MongoDB, complex join operations are not available; hence, it cannot support complex transactions.
- MongoDB uses JavaScript for coding as a language which is one of the great advantages.
- It is Schemaless as any data stored which is stored in a separate document.
- It also supports a flexible document model, which is very fast for any developer to create.
- MongoDB is a NoSQL database that scales by adding more and more servers and increases productivity with its flexible document model.

### 2. Express

- **Express**, mainly used for developing Node.js web framework also it is a JavaScript server-side framework that runs within js.
- It provides the developer with a platform to create and maintain robust servers.
- Express is used for building and designing web and mobile applications easily and quickly.
- Express is used to provide server-side logic for mobile and web applications, and as such, it is used all over the place.

- It allows developers to spin up robust APIs (Application Programming Interface) and web servers much easier and simpler.
- Express makes robust web servers easier to organize your application's functionality with routing and middleware.
- It also adds helpful functionalities to Node.js HTTP (HyperText Transfer Protocol) objects.
- It is an important component of the MERN and MEAN Stack and is used to build fast, maintainable, and robust productions web applications.

### 3. React

- **React**, mainly used for developing a client-side JavaScript framework
- Before using react, it has some prerequisites that one should follow, that you must download Node packages in your system with their latest versions. Also, you must have an understanding of HTML, CSS and JavaScript.
- It is used to build user interfaces, especially for a single page web application.
- It is not a JavaScript framework. It is just a JavaScript library developed by Facebook to solve problems we could not solve earlier using other libraries while building web and mobile applications.
- React is also used for making a grip over the view layer for mobile and web applications.
- It allows us to create reusable UI (User Interface) components also it is very fast, simple and scalable.
- It allows developers to create large web applications that can easily change the data of the page even without reloading the page.
- The main objective of reacting is that it only works on user interfaces in the application, whether mobile or web.
- React is also used with a combination of other JavaScript libraries or frameworks.

### 4. Node.js

- **Node.js**, mainly used for developing the premier JavaScript web server
- js is an open-source server environment, and it is a cross-platform runtime environment for executing JavaScript code outside a browser.
- js is not a programming language, and even it is not a framework.
- It is often used for building and developing numerous backend services like net applications, mobile applications.
- It will run on numerous operative systems like Windows, Mac, Linux, Unix, etc.
- It is incredibly simple to urge started with it and may even be used for agile development and prototyping also it provides quick services to the users.

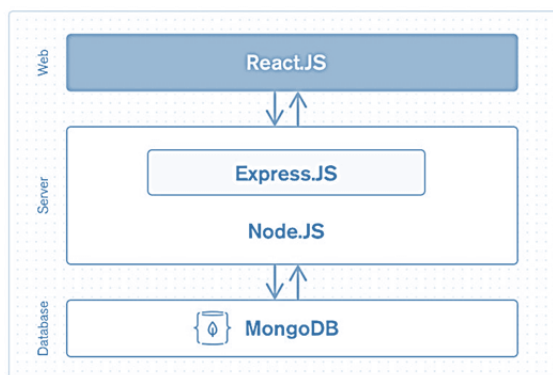
Each of these four technologies plays an important role in providing an end-to-end framework for the developers. Even these four technologies play an important role in the development process of web applications.

➤ **Architectural Structure of MERN Stack:**

MERN has a 3-tier Architecture system mainly consisting of 3 layers -

These layers are as follows:

- Web as front-end tier
- Server as the middle tier
- Database as backend tier



**Fig: MERN Stack Architecture**

**Web or front-end tier** - The top tier of the MERN stack is mainly handled by React.js. It is one of the most prominent used open-source front-end JavaScript libraries used for building Web applications. It is famous for creating dynamic client-side applications. React will help you construct complex interfaces by using single components. It also connects those complex interfaces to data available on the backend server. React is used to create mobile applications (React Native) and web applications. React allows the reusability of code and can easily support it, which has many benefits and is much time saver. It permits users to create large web applications that can easily change the data of the page even without reloading the page.

**Server or middle-tier** - It is just next level from the top layer and is mainly handled by two components of the MERN stack, i.e., Express.js and Node.js. These two's components handle it simultaneously because Express.js maintained the Server-side framework, running inside the Node.js server. Express.js is one of the widely used backend development JavaScript Frameworks. It allows developers to spin up robust APIs (Application Programming Interface) and web servers much easier and simpler. It also adds helpful functionalities to Node.js HTTP (HyperText Transfer Protocol) objects. Whereas on the other hand, Node.js plays a very important role in itself. It is an open-source server environment, and it is a cross-platform runtime environment for executing JavaScript code outside a browser. Node.js continuously uses JavaScript; thus, it's ultimately helpful for a computer user to quickly create any net service or any net or mobile application.

**Database as backend tier** - It is one of the most important levels of the MERN Stack and is mainly handled by MongoDB; the main role of a database is to store all the data related to your application, for example - content, statistics, information, user profiles, comments

and so on. It mainly stores all the data for safety purposes. It maintains a proper record, which usually returns the data to the user whenever required. It mainly stores the data in the database. It generates two or more replica files of the data so that whenever the system fails, it can retrieve the exact information or data that the user wanted earlier. It implies that MongoDB is not based on the table-like relational database structure. On the other hand, it provides an altogether different mechanism for the retrieval and storage of data. Mongo DB is the most popular NoSQL (NoSQL or Non Structured Query Language) database, an open-source document-oriented database. The term 'NoSQL' typically means a non-relational database that does not require a fixed schema or proper relational tables to store the necessary data in it. MongoDB stores the data in a different format other than the relational tables, consisting of rows and columns.

➤ **Advantages of MERN Stack**

- For a smooth development of any web application or mobile app, it supports MVC (Model View Controller) architecture; the main purpose of this architecture is to separate the presentation details with the business logic.
- It covers all the web development stages starting from front-end development to backend development with JavaScript.
- It is an open-source framework mainly used to develop web-based or mobile applications and is supported by the community.
- It is very fast and efficient compared to MEAN Stack and mostly suitable for small applications, whereas MEAN Stack is suitable for developing large applications.

---

## MEVN stack

MEVN stack is one of the most popularly used collections of technologies that enable faster application development. MEVN stands for MongoDB, Express, Vue, and Node, after the four key technologies that make up the stack.

- M – MongoDB: document database
- E – ExpressJS: Node.js web framework
- V – VueJS: A client-side JavaScript framework
- N – NodeJS: Premier JavaScript web server

One of the main reasons the MEVN stack is popular is because the application is built with only one language i.e JavaScript.

It solves the problem of implementing different languages at the frontend and backend.



**Fig: MEVN Stack**

**MongoDB**

- MongoDB is an open-source document NoSQL database. NoSQL database systems offer an alternative to traditional relational databases using SQL (Structured Query Language). The data is stored in tables, rows, and columns in a relational database, with relationships between entities. Whereas, in NoSQL databases such as MongoDB, the data is represented in a JSON-like structure which is stored in documents.
- NoSQL Databases store data in an unstructured format and process non-similar data sets that are associated with each other. Therefore, it has fewer restrictions and more flexibility with the use of dissimilar fields within a collection as compared to relational databases.
- MongoDB Atlas used for a fully-managed cloud database for full-scale applications and mongoose npm library can connect the database with our backend.

**ExpressJS**

- A framework layered on top of NodeJS, used to build the backend of a site using NodeJS functions and structures.
- Express is a Node.js web application framework that provides a range of features for building web applications and APIs. It makes it easy to create and manage routes, handle requests and responses, and more.

**NodeJS**

Node.js is a JavaScript runtime environment that allows developers to create server-side applications using JavaScript. Node.js is used for building the backend of the MEVN stack. It is a highly scalable, non-blocking I/O platform that makes it ideal for real-time applications.

**VueJS**

- VueJS is a highly flexible, lightweight open-source JavaScript Framework used for building single-page web applications, i.e while shifting between different routes, the website won't require a reload.
- VueJS is one of the best frameworks for JavaScript like ReactJS. The VueJS is used to design the user interface layer, it is easy to pick up for any developers. It is compatible with other libraries and extensions as well. In the development field, there may be so many issues that cannot be solved by using a single library, so the VueJS is compatible with other libraries. The VueJS is supported by all popular browsers like Chrome, Firefox, IE, Safari, etc.
- VueJS uses a two-way binding model which makes it very easy to update app data and browser view. It is developer friendly framework, i.e it is easier to learn as it allows both HTML and .jsx syntax and also has detailed documentation to refer to and learn.



➤ **Advantages of MEVN Stack:**

- The biggest advantage of MEVN is that javascript is used on all levels of development from client-side to server-side which simplifies the process and makes development faster and efficient.
  - It is platform-independent.
  - MVC architecture in the back-end organizes the server-side and hence back-end development becomes faster and more efficient.
  - In MEVN stack the Vue.JS is faster and easy to learn.
- 

## **LAMP Stack**

- LAMP is an open-source Web development platform that uses Linux as the operating system, Apache as the Web server, MySQL as the relational database management system and PHP/Perl/Python as the object-oriented scripting language.
- LAMP is also called a LAMP stack because it has four layers in its platform. Stacks on various operating systems may be installed.
- LAMP is an example of a web service stack, named as an acronym. The LAMP components are largely interchangeable and not limited to the original selection. LAMP is suitable for building dynamic web sites and web applications.
- The LAMP model, although traditionally composed of open source free software, has been adapted to another part since its development.
- Developers that use these tools with a Windows operating system instead of Linux are said to be using WAMP, with a Macintosh system MAMP, and with a Solaris system SAMP.
- Linux, Apache, MySQL and PHP, all of them add something unique to the development of high-performance web applications. Originally popularized from the phrase Linux, Apache, MySQL, and PHP, the acronym LAMP now refers to a generic software stack model.

➤ **LAMP Stack Components**

Linux based web servers consist of four software components. These components are arranged in layers supporting one another and make up the software stack. Websites and Web Applications run on top of this underlying stack. The common software components are as follows:

**1. Linux:**

- Linux started in 1991. It sets the foundation for the stack model. All other layers are run on top of this layer.
- It is an open-source and free operating system. It is endured partly because it's flexible, and other operating systems are harder to configure.

**2. Apache:**

- The second layer consists of web server software, typically Apache Web Server. This layer resides on top of the Linux layer.

- Apache HTTP Server is a free web server software package made available under an open-source license. It used to be known as Apache Web Server when it was created in 1995.
- It offers a secure and extendable Web server that's in sync with current HTTP standards. Web servers are responsible for translating from web browsers to their correct website.

### 3. MySQL:

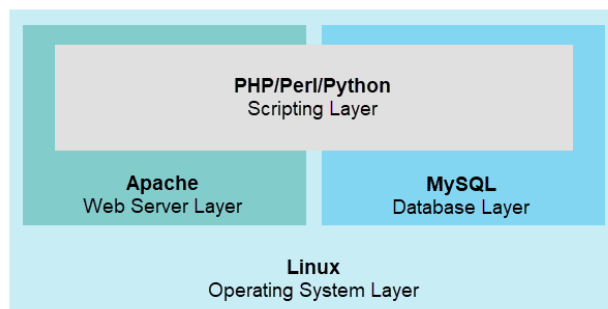
- MySQL is a relational database management system used to store application data. It is an open-source and keeps all the data in a format that can easily be queried with the SQL language.
- SQL works great with well-structured business domains, and a great workhorse that can handle even the most extensive and most complicated websites with ease.
- MySQL stores details that can be queried by scripting to construct a website. MySQL usually sits on top of the Linux layer alongside Apache. In high-end configurations, MySQL can be offloaded to a separate host server.

### 4. PHP:

- The scripting layer consists of PHP and other similar web programming languages.
- The PHP open-source scripting language works with Apache to create dynamic web pages. We cannot use HTML to perform dynamic processes such as pulling data out of a database.
- To provide this type of functionality, we drop PHP code into the parts of a page that you want to be dynamic. Websites and Web Applications run within this layer.
- PHP is designed for efficiency. It makes programming easier and allowing to write new code, hit refresh, and immediately see the resulting changes without the need for compiling.

### ➤ LAMP Architecture

- LAMP has classic layered architecture, with Linux at the lowest level. The next layer is Apache and MySQL, followed by PHP.
- Although PHP is at the top or presentation layer, the PHP component sits inside Apache.



*Fig: LAMP Architecture*

- The LAMP stack order of execution shows how the elements interoperate. The process starts when the Apache webserver receives requests for web pages from a user's browser. If the request is for a PHP file, Apache passes the request to PHP, which loads the file and executes the code contained in the file. PHP also communicates with MySQL to fetch any data referenced in the code.
- PHP then uses the code in the file and the data from the database to create the HTML that browsers require to display web pages. The LAMP stack is efficient at handling not only static web pages but also dynamic pages where the content may change each time it is loaded depending on the date, time, user identity and other factors.
- After running the file code, PHP then passes the resulting data back to the Apache webserver to send to the browser. It can also store this new data in MySQL. And of course, all of these operations are enabled by the Linux operating system running at the base of the stack.

➤ **Features of LAMP**

**1. Efficiency**

LAMP will minimize production time since it is an open-source stack usable for a decade. We are willing to draw on and make things easier for those in the past. Function in an Apache module that can configure the last 20 percent to 80 percent and save tremendous time.

**2. Flexibility**

- Though LAMP uses Linux as an OS, other modules of an alternative OS may be used to satisfy those specifications. There is, for instance, a Microsoft Windows WAMP stack.
- LAMP is open source and non-proprietary so we can avoid lock-in. We have the flexibility to select the right components for specific projects or business requirements.
- LAMP offers flexibility in other ways as well. Apache is modular in design, and we will find there are existing, customizable modules available for many different extensions. These modules range from support for other languages to authentication capabilities.

➤ **Advantages of LAMP**

LAMP has the following advantages, such as:

- The LAMP stack consists of four components, all of which are examples of Free and Open-Source Software (FOSS). As they are free and available for download, it attracts the attention of many users who wish to avoid paying large sums of money when developing their website.
- Because it is FOSS, the source code of the software is shared and available for people to make changes and improvements, enhancing its overall performance.
- The LAMP stack has proven to be a secure and stable platform thanks to its vast community that contributes when any problems arise.

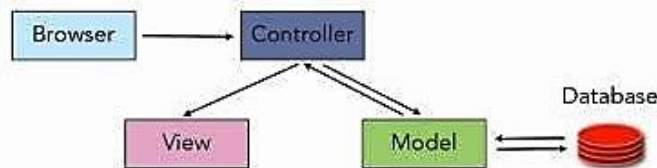
- We can easily customize the stack and interchange the components with other open-source software to suit the needs.
  - LAMP is its secure architecture and well-established encryption practices that have been proven in the enterprise.
- 

## Rails or Ruby on Rails

- Ruby on Rails or also known as rails is a server-side web application development framework that is written in the Ruby programming language.
- It supports MVC(model-view-controller) architecture that provides a default structure for database, web pages, and web services, it also uses web standards like JSON or XML for transfer data and HTML, CSS, and JavaScript for the user interface.
- The usage of different well-known web development patterns and frameworks, such as:
  - Don't Repeat Yourself (DRY): It is a software development philosophy that seeks to minimize the repeating of data or codes.
  - Convention Over Configuration (CoC): It offers a variety of viewpoints on the best ways to carry out certain tasks in a web-based application.

### ➤ MVC architecture

The MVC architecture helps to separate the different components of the application, making it easier to understand, maintain, and improve. For example, if the user need to change the way the data is displayed, they can make the changes in the view without affecting the model or the controller. This makes the development process smoother and more efficient.



*Fig: MVC architecture*

*Component of the MVC architecture are.*

#### **Model:**

The model contains the business logic of the application, and rules to modify the data. In Ruby on Rails, they handle the interaction with related elements in the database. They indicate the data in the database and perform accurate verifications.

#### **View:**

It maintains the view display logic. It displays the data retrieved from the model in a visually appealing way, using HTML, CSS, and JavaScript.

#### **Controller:**

The controller is the glue that connects the model and the view. It retrieves data from the model, processes it, and passes it on to the view to be displayed. It also handles user

interactions, such as submitting forms and communicates with the model to update the data.

### Feature of Ruby on Rails

With both the front and backend, Ruby on Rails is a complete version of creating web applications. Other frameworks are either exclusively for the back end or the front end. Among Ruby on Rails' key attributes are the following:

- **Model-view-controller Architecture:** Ruby on Rails used MVC architecture, and it contains three components, i.e., model, view, and controller. Here, the model is used to maintain the relationship between object and database, the view is templates that are used to build the data users for web applications, and the controller is used to merge model and view together. MVC is generally used for developing user interfaces that divide the data into three interconnected components so that it can separate the internal representation of the information from the way it presents to and get from the user.
- **Active Records:** The active record framework is introduced in Ruby on Rails. It is a powerful library that allows the developer to design the database interactive queries.
- **Built-in Testing:** Ruby on Rails provides its own set of tests that will run on your code. It will save time and effort.
- **Programming Language:** This syntax of Ruby on Rails is simple because the syntax of the Ruby programming language is close to English, so it is always easier to structure your thinking and writing it into code.
- **MetaProgramming:** Ruby on rails uses the metaprogramming technique to write programs.
- **Convention over configuration:** In Ruby on Rails, a programmer can only specify the unconventional aspects of the application.
- **Scaffolding:** Ruby on rails provides a scaffolding feature in which the developer is allowed to define how the application database works. After defining the work of the application database the framework automatically generates the required code according to the given definition. This technique creates interfaces automatically.

### Advantages of Ruby on Rails

- **Tooling:** Rails provides tooling that helps us to deliver more features in less time.
  - **Libraries:** There's a 3rd party module (gem) for just about anything we can think of.
  - **Code Quality:** Ruby code quality significantly higher than PHP or NodeJS equivalents.
  - **Test Automation:** The Ruby community is big into and test automation and testing.
  - **Large Community:** Ruby is large in the community.
  - **Productivity:** Ruby is incredibly fast from another language. Its productivity is high.
-

## Django

- Django is a Python framework that makes it easier to create web sites using Python.
- Django emphasizes reusability of components, also referred to as DRY (Don't Repeat Yourself), and comes with ready-to-use features like login system, database connection and CRUD operations (Create Read Update Delete).
- Django follows the MVT design pattern (Model View Template).
  - Model
  - View
  - Template

### Model

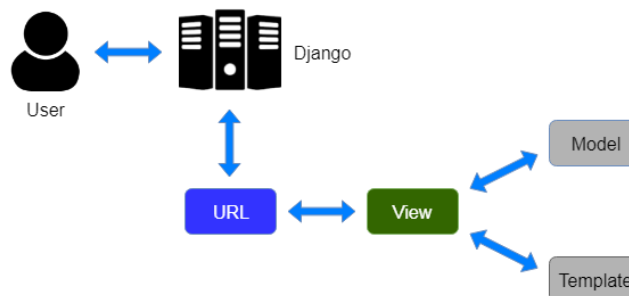
- The model provides data from the database.
- In Django, the data is delivered as an Object Relational Mapping (ORM), which is a technique designed to make it easier to work with databases.
- The most common way to extract data from a database is SQL. One problem with SQL is that you have to have a pretty good understanding of the database structure to be able to work with it.
- Django, with ORM, makes it easier to communicate with the database, without having to write complex SQL statements.

### View

- A request handler that returns the relevant template and content - based on the request from the user.
- A view is a function or method that takes http requests as arguments, imports the relevant model(s), and finds out what data to send to the template, and returns the final result.

### Template

- A text file (like an HTML file) containing the layout of the web page, with logic on how to display the data.
- The Template is a presentation layer which handles User Interface part completely.
- A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.



*Fig: MVT based control flow*

- Here, a user requests for a resource to the Django, Django works as a controller and check to the available resource in URL.

- If URL maps, a view is called that interact with model and template, it renders a template.
- Django responds back to the user and sends a template as a response.

➤ **Features of Django**

- **Rapid Development:** Django was designed with the intention to make a framework which takes less time to build web application. The project implementation phase is a very time taken but Django creates it rapidly.
  - **Secure:** Django takes security seriously and helps developers to avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery etc. Its user authentication system provides a secure way to manage user accounts and passwords.
  - **Scalable:** Django is scalable in nature and has ability to quickly and flexibly switch from small to large scale application project.
  - **Fully loaded:** Django includes various helping task modules and libraries which can be used to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds etc.
  - **Versatile:** Django is versatile in nature which allows it to build applications for different-different domains. Now a days, Companies are using Django to build various types of applications like: content management systems, social networks sites or scientific computing platforms etc.
  - **Open Source:** Django is an open source web application framework. It is publicly available without cost. It can be downloaded with source code from the public repository. Open source reduces the total cost of the application development.
  - **Vast and Supported Community:** Django is a one of the most popular web framework. It has widely supportive community and channels to share and connect.
-