

Modern Web Application Architecture

by Devmountain Tutorials



WHAT'S COVERED

This section will explore the structure of web applications by discussing:

1. THREE-TIER ARCHITECTURE
2. FRONT END AND BACK END

1. THREE-TIER ARCHITECTURE

In software development, the term architecture is used as a metaphor that's analogous to the architecture of a building. If architecture is the process and product of planning, designing, and constructing buildings, **software architecture** is the process and product of planning, designing, and constructing a software system. In architecture, there are structures that are so commonly used that they're given names like domes, columns, and arches. There are common patterns and structures in software architecture too; most web applications are structured based on the three-tier architecture.

The three-tier architecture is a **multi-tier architecture** or **multi-layered architecture**. You'll commonly find these patterns in computer systems that involve clients and servers like the Internet. In multitiered systems, code is strategically partitioned into separate tiers so that, whenever developers need to modify an application, they'll only have to work with one tier at a time instead of tearing apart and reworking the entire application. The three tiers of three-tier architecture are:

Presentation tier

- The part of the application that is the user interface. It translates user input into computational tasks and computer-generated results into something visual that users can understand. It's the most visible part of an application, and it's the tier you're looking at right now while you're reading this sentence.

Logic tier

- Like an orchestra conductor, this layer coordinates the application, performs calculations, makes logical decisions and evaluations, and moves data between the presentation tier and the data tier.

Data tier

- This is where information is stored in something like a database or file system.

The descriptions above are all pretty abstract so don't worry if you're not *completely* sure how they work together yet. It's probably more helpful to talk about the specifics of how this architecture is used for web

applications.



TERM TO KNOW

Software architecture

The process and product of planning, designing, and constructing a software system.

Three-tiered architecture (also called Multi-tier and Multi-layer)

In multitiered systems, code is strategically partitioned into separate tiers so that, whenever developers need to modify an application, they'll only have to work with one tier at a time instead of tearing apart and reworking the entire application.

2. FRONT END AND BACK END

In web development, **front end** refers to the highest tier — the presentation tier — and **back end** refers to the lowest tier — the data tier.

For web applications, the front end is the content rendered by the browser. It's the content that users can see and interact with. In other words, it's the web page itself. Colloquially, back end is used to refer to everything else that's not the front end — that is, the logic tier and data tier. Technically, the back end specifically refers to the data tier which is comprised of the database or file system where data is stored and the software that manages and provides access to the data. In web applications, the logic tier is made up of software that can process user input from the front end, retrieve data from the back end, and process that data so it can be displayed by the front end. For the rest of this course, we'll use the colloquial definition of back end, the one that encapsulates both the logic and data tier.

Another way to differentiate the two is by the languages used in each end. When working in the front end, web developers are restricted to using coding languages that web browsers can understand. In most cases, these languages are HTML, CSS, and JavaScript. In the back end, there are no restrictions on programming languages, since back end software runs on computer hardware instead of running through a web browser.



TERM TO KNOW

Front End

The highest tier, the presentation tier, in a web application.

Back End

The lowest tier, the data tier, in a web application.



TERMS TO KNOW

Back End

The lowest tier, the data tier, in a web application.

Front End

The highest tier, the presentation tier, in a web application.

Software architecture

The process and product of planning, designing, and constructing a software system.

Three-tiered architecture (also called Multi-tier and Multi-layer)

In multitiered systems, code is strategically partitioned into separate tiers so that, whenever developers need to modify an application, they'll only have to work with one tier at a time instead of tearing apart and reworking the entire application.