Two-Tier Architecture

Introduction

Two-tier architecture was developed in the 1980s from the file server software architecture design. This two-tier architecture is intended to improve usability by supporting a forms-based, user-friendly interface. The two-tier architecture improves scalability by accommodating up to 100 users. This architecture is mainly used in non-complex, non-time critical information processing systems.

Now we will discuss this architecture in detail.

What is Two-Tier Architecture?

A two-tier system consists of a server and a client. In a two-tier system, the database is stored on the server, and the interface installed on the client is used to access the database.

The user system interface is generally located in the user's desktop and the database management services are generally in a server that is a more powerful machine that services many clients. Processing management is split between the user system interface environment and the database management server environment. The database management server provides stored procedures and triggers.

Example of Two-Tier DBMS Architecture

Client Layer

- **User Interface (UI)**: This layer includes applications that users interact with directly. For example, a web application used by employees to manage inventory in a retail store.

- **Database Client**: It communicates with the database server to send queries and receive results. In our example, this could be SQL queries generated by the UI to fetch inventory details or update stock levels.

Server Layer

- **Database Server**: This layer manages and stores the actual data. It responds to queries from the client, processes transactions, and maintains data integrity.

- **DBMS (Database Management System)**: The software responsible for managing the database on the server. Examples include MySQL, PostgreSQL, or Oracle Database.

Check this out, Data Warehouse Architecture

Here's your problem of the day

Test your skills and sharpen your expertise with today's problem

Skill covered: **Programming**

Which data type is used to store whole numbers?

intfloatbooleandouble

Submit**Choose another skill to practice**

Characteristics of 2 Tier Architecture

- Two-tier architectures consist of three components distributed in two layers - client and server. These components are
    - 1) User system interface
    - 2) Processing management
    - 3)Database management.

- Two-tier architecture allocates the user system interface exclusively to the client.

- This architecture places database management on the server and splits the processing management between client and server, creating two layers.

Advantages of 2 Tier Architecture

- As processing is shared between the client and server, many users can interact with a two-tier architecture system.

- The two-tier architecture improves scalability.

- It also improves flexibility by allowing data to be shared within a homogenous environment.

- The two-tier architecture requires minimal operator intervention.

Disadvantages of 2 Tier Architecture

- The performance of this architecture degrades when the number of users increases.
- In this case, it is difficult to implement reliable security as users need to have login information for every database server.

Frequently Asked Questions

What are the use cases of two-tier architecture?

Two-tier architectures are often used when the user requirements are moderately heavy, and the system is expected to have minimum interactions and maintenance.

What are some limitations of a two-tier architecture?

The major limitations of using two-tier architecture patterns for designing systems are:

1. Since there is a separation between the server and the client in this type of architecture, the overhead and the cost of the maintenance of the system increase.
2. As the number of users keep increasing in the system, the overall fault tolerance and reliability of the system reduces.

What is a 2 tier architecture?

A 2-tier architecture refers to a client-server architecture where the client interacts directly with the server. It consists of two main layers: the client or front-end and the server or back-end, typically used in simple applications where scalability and flexibility are less critical.

What is 2 and 3 tier architecture?

A 2-tier architecture involves a client directly communicating with a server, suitable for simpler applications. In contrast, a 3-tier architecture adds an intermediate application server layer between the client and server, enhancing scalability, separation of concerns, and facilitating distributed computing.

What is the 2 tier architecture in JDBC?

In JDBC (Java Database Connectivity), the 2-tier architecture involves direct communication between the Java application (client) and the database server (server). JDBC allows Java programs to interact with databases using SQL queries, managing connections, sending queries, and processing results directly through JDBC drivers installed on the client side.

What is a web app?

A web application also known as a web app can be accessed through a web browser. Web applications are usually stored on remote servers, and users can access them through web browsers. When accessed via a mobile device, a web app looks and behaves like a mobile app but the two aren’t the same. If you're looking to build your enterprise web application, understanding the distinction between web and mobile apps is crucial.

Using browsers allows web applications to be compatible with most operating systems and standard computers. Additionally, the apps don’t take up memory on a computer’s hard drive and can be accessed from almost any computer or device. Many people can use the same application simultaneously and collaborate.

Some examples of web applications include shopping carts, online forms, spreadsheets, word processors, file conversion programs, video and photo editing software, file scanning tools, and email programs such as Gmail and Yahoo.

User experience (UX) usually dictates web application design. With traditional web design, server-side programmers decide what might enhance usability. But web applications have an application program interface (API) that pulls data from the user side and funnels it into automation.

Is creating a web app easy?

Developing custom web applications is much easier now than a decade ago. Application development software helps skilled employees to create web apps without coding. Many new app development technologies, from online forms to word processors, are web apps. They help companies to be more productive. Research shows that web forms are the most popular way to capture leads. Every company looking to grow must start building web apps.

To build a web application

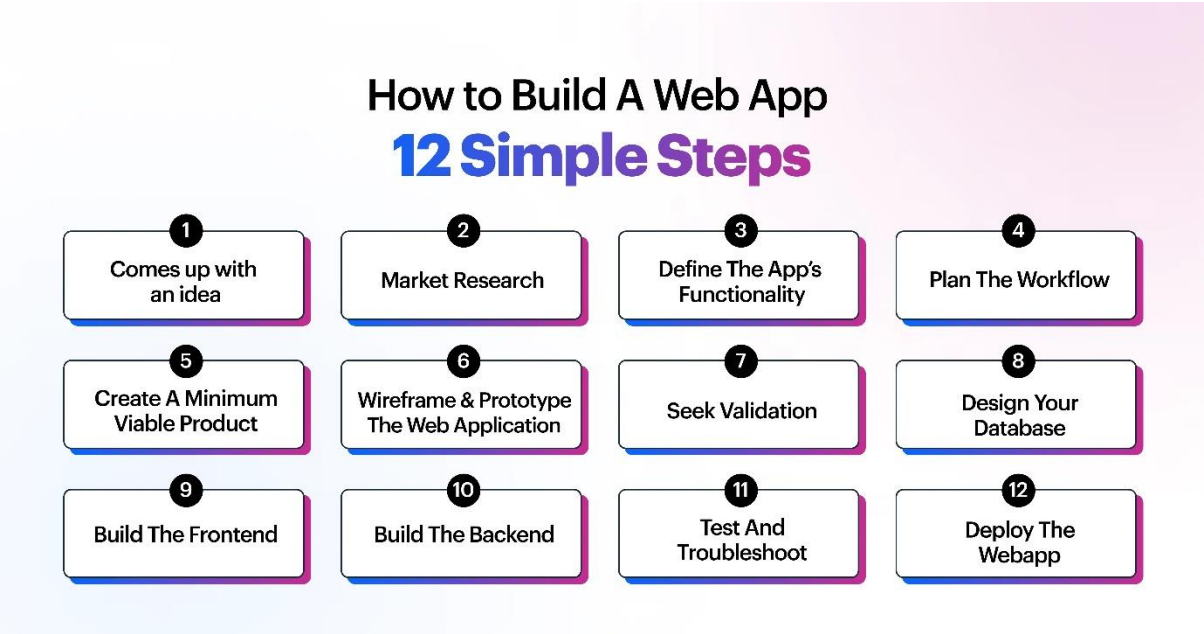
- A developer first tries to find a solution to a specific problem.
- Next, they create the web app by choosing the appropriate development framework. During the development stage, the developer engages with the end users to ensure they build the right solution.
- Lastly, they test the solution and then deploy the web app.

There are two major parts of web-based application development:

1. The client side.
2. The server side.

The client-side helps users see data from the server, while the server side helps developers store and retrieve data from the database. Enterprise web application development is easier and more affordable than app development for mobile devices.

How to create a web app - 12 simple steps



Following are the 12 steps to create and build a web application for your business in 2024.

1. Come up with an idea
2. Market research
3. Define the app's functionality
4. Plan the workflow
5. Create a minimum viable product (MVP)
6. Wireframe and prototype the web application
7. Seek validation
8. Design your database

1. Come up with an idea

You don’t need an app for every business idea. Do your research first, identify the problem you want to solve and try to find solutions. Interview people who experience the problem and ask if they have possible solutions. If you feel a web application will streamline work, proceed to the next stage.

2. Market research

You must have an accurate understanding of the user, their problem, and the scope of the problem. Research if there are other ways to solve the problem out there. Getting insight into the user and the competition helps inform the technical direction of the web app.

3. Define the app’s functionality

Come up with a list of features that will address the problem. But keep your distance and add only a few functionalities. Apps with more features take longer to build. Only focus on the features that solve your target market's problems.

4. Plan the workflow

If your app has competitors, sign up for their free trials and use the solutions. Pay attention to the workflow and note what’s good and bad. Sit down and sketch different workflows for your app that is better than the competition.

5. Create a minimum viable product (MVP)

First, sketch out the structure of your web app’s user interface—it doesn’t have to be exact. Outline how your app should work, and don’t overcomplicate the design. Next, create a basic but complete version of the app that showcases its main features.

6. Wireframe and prototype the web application

Wireframing means designing your web application’s blueprint, while prototyping means adding an interactive display. You can wireframe and prototype your app using a no-code/low-code platform. Prototyping makes it easier to explain your web app when seeking validation.

7. Seek validation

Join relevant forums and social media groups and present your solution. Find out what you need to change and what to keep. The information you get will help direct the development of your product. If you get great feedback and product validation, start building your web app.

8. Design your database

A database allows you to store data securely to access it later. The data you store, and user functions will determine the type of database required to run your web app.

9. Build the frontend

The frontend is what users see and interact with. It's the visual element of the application and is developed with coding language. The framework is set up on a no-code/low-code platform, and almost no coding is needed to build a frontend. Not all web applications need frontend frameworks.

10. Build the backend

The backend manages your data, servers, databases, and everything users can't see in a web application. Building a backend is a job for skilled developers, but a no-code/low-code tool can take away many of the complexities.

11. Test and troubleshoot

Test the app before you publish it. Use it in run mode and see if it functions as expected. If something is wrong, diagnose the problem using the no-code/low-code platform. Some platforms have a debug mode and issue checker to make your job easy.

12. Deploy the web app

First, choose a host for your web app to make it available on the cloud. Users will be able to access it from anywhere in the world. Next, deploy the web app. This step involves getting your web application from your computer to your cloud hosting.

Feeling overwhelmed by all the buzz around low-code?

This guide simplifies everything and helps you choose the right path for building web apps quickly. Claim your free copy today!

What are the different stages in web app

Information gathering

Businesses mostly build web applications to solve problems. They come up with ideas for apps and outline the applications' goals, features, budgets, visions, and future plans. Developers go through the outline to understand the app's objectives, goals, focus industry, target audience, and other critical elements.

Discussions and questionnaires help them to get more clarity on project goals. The developers then prepare a proposal to document the deliverables.

Planning the web application development process

With the insights gathered in the first stage, the developer creates a blueprint that determines the overall structure of the web application. The blueprint includes flowcharts and sketches. Developers keep clients in the loop during this stage to ensure that the application is perfect.

The time spent in this stage depends on the complexity of the web application. A developer creating a Minimum Viable Product might spend around two weeks on this.

Web application design

This stage is all about perfecting the interactive elements of the web application. The developer works with graphics, templates, colour schemes, style guides, and much more to complete the design of the web application. The final mockup is sent to the client for review and feedback. Mockup changes and design iterations go on until the client agrees to everything.

Web application programming

At this stage, developers create the envisioned features. They develop frameworks, deploy APIs, build app features, add security layers, and many other capabilities. For web development companies, Developing complex web applications is more time-consuming, depending on the technologies used. Some technology stacks have capabilities that can be easily tweaked and integrated.

Testing and launch

Testing is the most important part of web app development. A million things can go wrong even after the application has been developed correctly. Testing ensures that the web app works as intended and meets organization, industry, and global standards.

Even if everything has been double-tested, it is a good idea to initially launch the web application in the beta version. If resources are few and stakes are high, the application can be released in phases to different audience groups.

Application maintenance

Every digital product needs routine checkups and enhancements, whether it's a single-page application or a complex web application. As time goes by, you may need to undertake product pivots, integrate new features, and launch the next version.

How web apps can help enterprise businesses

With enterprise low-code application platforms, organizations can build cost-effective and accurate apps and minimize the wastage of resources. Low-code development efforts maximize a company's available resources.

When a company spends less on professional development, it becomes more innovative and productive. Statista reports that many customer-focused apps built with low-code increase business revenue. Low-code tools allow developers and business users to make their digital ecosystems more robust.

Leverage Kissflow's app development platform to build powerful enterprise applications faster. It has an intuitive user interface with extensive cross-platform integration to streamline processes, increase productivity and incorporate your regular development workflows. You'll heighten security with role-based access control and use a highly-visual workflow design to empower citizen developers.