WEB ENGINEERING FUNDAMENTALS · MODULE 11

Frontend vs Backend

A complete beginner's guide to how websites re
work — learn the difference between frontend a
backend development, how they communicate,
what technologies each uses, and how they wor
together to build modern web applications.

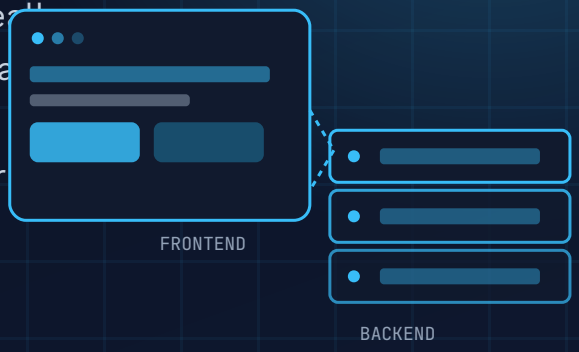


Table of Contents

FRONTEND VS BACKEND — MODULE 11

Introduction

Every website you use is built from two major parts:

Frontend and Backend.

Many beginners learn HTML, CSS, and JavaScript but still struggle to understand how real applications like YouTube, Instagram, or Amazon actually work.

The reason is simple.

Learning programming languages is different from understanding how complete systems work.

Frontend = What Users See

Backend = What Runs Behind It

Frontend is what users see and interact with.

Backend is everything working behind the scenes.

Both are essential.

Without the frontend, users have nothing to interact with.

Without the backend, the website cannot store data, authenticate users, process payments, or perform business logic.

> WHY THIS MATTERS
>
> Understanding this relationship is one of the biggest milestones in becoming an independent developer.

What Is Frontend?

Frontend is the part of a website that runs inside the user's browser.

It is everything users can:

- see
- click
- type into
- scroll
- interact with

Examples include:

- buttons
- navigation bars
- forms
- images
- animations
- dashboards
- menus

The frontend focuses on user experience and presentation.

Main Responsibilities of the Frontend

A frontend application is responsible for:

- displaying information
- collecting user input
- handling interactions
- validating forms
- calling backend APIs
- updating the interface

> THINK OF IT THIS WAY
> Think of the frontend as the "face" of a website.

What Is Backend?

Backend is the server-side part of an application.

Users never see it directly.

Instead, it performs the work required to make the application function.

Examples include:

- user authentication
- database operations
- payment processing
- sending emails
- business rules
- generating reports
- processing uploaded files

> THINK OF IT THIS WAY
> The backend acts as the application's brain.

Main Responsibilities of the Backend

The backend:

- receives requests
- validates data
- communicates with databases
- performs calculations
- applies business logic
- returns responses

Without a backend, most modern applications would not function.

Why Do Websites Need Both?

Imagine an online banking application.

The frontend shows:

- account balance
- transfer forms
- transaction history

The backend:

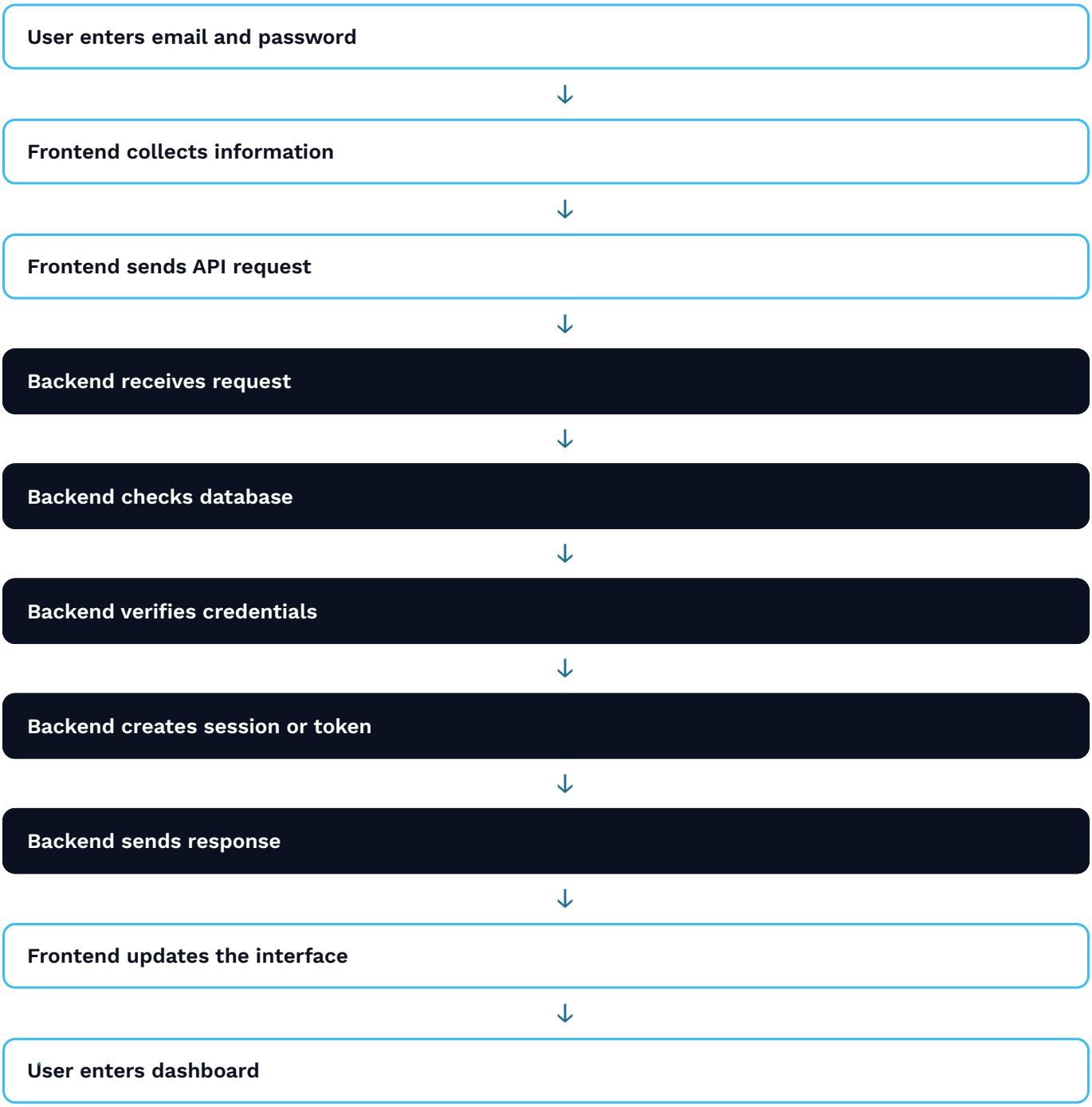
- verifies user identity
- checks available balance
- transfers money
- records transactions
- protects sensitive information

KEY POINT

If either side disappeared, the application would stop working properly.

How Frontend and Backend Work Together

Let's follow a simple login request.



This communication happens thousands of times every day in modern applications.

> LEARN MORE
>
> Read **What Is an API?** to understand how the frontend and backend exchange information.

Frontend Technologies

Frontend developers commonly use:

HTML

Defines the page structure.

CSS

Controls styling and layout.

JAVASCRIPT

Adds interactivity.

POPULAR FRONTEND FRAMEWORKS

- React
- Vue
- Angular
- Svelte

These frameworks simplify building large applications.

Backend Technologies

Backend development involves many technologies.

PROGRAMMING LANGUAGES

- JavaScript (Node.js)
- Python
- Java
- PHP
- C#
- Go
- Ruby

DATABASES

- PostgreSQL
- MySQL
- MongoDB
- SQLite

BACKEND FRAMEWORKS

- Express.js
- NestJS
- Django
- Laravel
- Spring Boot

These tools help developers build scalable server-side applications.

Frontend vs Backend Comparison

Feature	Frontend	Backend
Runs On	Browser	Server
Visible to Users	Yes	No
Handles UI	Yes	No
Stores Data	No	Yes
Uses Databases	No	Yes
Executes Business Logic	Limited	Yes
Authentication	Partial	Primary
Technologies	HTML, CSS, JS	Node.js, Python, Java, etc.

Real World Example

Consider an online food delivery app.

Frontend displays:

- restaurants
- menus
- cart
- checkout page

Backend handles:

- restaurant database
- order processing
- payment verification
- delivery tracking
- notifications

KEY POINT

The customer interacts only with the frontend, while the backend performs the heavy work behind the scenes.

Building Your First Full Stack Flow

Imagine creating a simple contact form.

Step 1 — Frontend displays the form.

↓

Step 2 — User enters information.

↓

Step 3 — Frontend sends the data to the backend.

↓

Step 4 — Backend validates the input.

↓

Step 5 — Backend stores the message in a database.

↓

Step 6 — Backend returns a success response.

↓

Step 7 — Frontend displays: "Your message has been sent successfully."

This small workflow demonstrates how frontend and backend cooperate.

Common Beginner Mistakes

- Thinking HTML alone builds complete websites.
- Believing frontend developers never communicate with servers.
- Assuming backend developers do not need to understand frontend concepts.
- Mixing business logic into the frontend.
- Sending sensitive data directly from the browser without validation.

Practical Action Plan

Choose three websites you use daily.

For each website:

Identify:

- which features belong to the frontend
- which operations happen in the backend

Then explain how information flows between both sides.

> WHY THIS EXERCISE
>
> This exercise strengthens system thinking.

Mini Architecture Challenge

CHALLENGE

Design a simple online library application.

Decide:

- What belongs in the frontend?
- What belongs in the backend?
- What information should be stored in the database?
- Which actions require API communication?

Draw the complete workflow before writing any code.

HAAS.DEV PHILOSOPHY

Exercises like this one build engineering judgment — thinking through a system before touching syntax — instead of memorizing steps.

Key Takeaways

- Frontend runs inside the browser.
- Backend runs on servers.
- Frontend focuses on user interaction.
- Backend manages business logic and data.
- APIs connect frontend and backend.
- Most modern applications require both.

Summary Page

Cheat Sheet

- [x] Frontend = User Interface
- [x] Backend = Server Logic
- [x] Frontend sends requests
- [x] Backend processes requests
- [x] Database stores information
- [x] APIs connect both sides
- [x] Both work together to build complete applications

Frontend vs Backend Decision Framework

Use Frontend For

- Displaying information
- Forms
- Navigation
- Animations
- User interaction

Use Backend For

- Authentication
- Databases
- Payments
- File uploads
- Business logic
- Email sending
- Data validation

Never trust the frontend with sensitive operations.

Related Resources

How Browsers Work

Why read it: Understand where frontend applications execute.

Cookies vs Sessions vs Local Storage

Why read it: Learn how browsers remember users and manage login state.

HTTP vs HTTPS

Why read it: Understand how frontend applications communicate securely with backend servers.

What Is an API?

Why read it: Learn the communication layer between frontend and backend.

What Is a Database?

Why read it: Understand where backend systems permanently store application data.

Recommended Next Learning Path

1. What Is a Website?
2. Website vs Web Application
3. How the Internet Works
4. What Happens When You Type a URL?
5. HTTP vs HTTPS
6. What Is a Domain Name?
7. What Is DNS?
8. What Is Web Hosting?
9. How Browsers Work
10. Cookies vs Sessions vs Local Storage
11. **Frontend vs Backend (Current PDF)**
12. What Is a Full Stack Developer?

HAAS.DEV

Thinking > tutorials. Get every guide in this series at dev-roast-app.vercel.app/resources