// WEB ENGINEERING FUNDAMENTALS — PDF 01

What Is a Website?.

A complete beginner's guide to understanding how websites actually work — before you write a single line of code.

// TABLE OF CONTENTS

Inside this guide

01 — INTRODUCTION

Every day, billions of people use websites without thinking about what's behind them

Search on Google. Watch videos on YouTube. Shop on Amazon. Read an article. Every one of those moments is a website — built from many technologies working together.

Most beginners jump straight into learning HTML, CSS, or JavaScript. They learn how to write code, but never learn what they're actually building. That gap between "knowing syntax" and "understanding the system" is where most confusion starts.

This guide solves that problem — not by teaching code first, but by teaching you how websites work as complete systems. Once this foundation is in place, everything you learn afterward — frameworks, backend logic, deployment — becomes far easier to place in context.

```
// THE MINDSET SHIFT
```

Understand the system before you touch the syntax. A developer who can reason about how a request flows from browser to server and back will out-learn one who only memorizes tags and commands.

02 — WHAT IS A WEBSITE?

A collection of connected pages, stored on a server, delivered over the internet

Each website is identified by a unique domain name — **google.com**, **wikipedia.org**, **amazon.com**. When someone types that address into a browser, the browser requests the website from a server and displays it on screen.

A website is more than text and images. It's a combination of different technologies — frontend, backend, database, server — working together to deliver information and functionality.

// THINK OF IT LIKE A LIBRARY

A simple analogy that makes the system click before the jargon does:

the building
is the server.

the books
are the web pages.

the address
is the domain name.

the librarian
is DNS, finding the right server.

When you request a book, the librarian brings it to you. When you request a page, the server sends it to your browser. Same pattern, different medium.

03 — WHY WEBSITES MATTER

Websites are the foundation of the modern internet

- **Businesses** use websites to sell products.
- **Schools** use websites to deliver education.
- **Hospitals** use websites to manage appointments.
- **Governments** use websites to deliver public services.
- **Individuals** use websites to share knowledge, portfolios, and businesses.

Without websites, most of the online services we rely on every day simply wouldn't exist. For a developer, understanding how they work isn't optional trivia — it's the first real step toward building something useful.

04 — WEBSITE VS WEB APPLICATION

Two words people use interchangeably — and shouldn't

WEBSITE

Delivers information

— Blogs
— Company websites
— News websites
— Documentation sites

WEB APPLICATION

Enables interaction

— Gmail
— Google Docs
— Trello
— Canva

Key difference: a website primarily delivers information; a web application lets users perform actions and interact with data. Most modern products blend both.

05 — THE BUILDING BLOCKS OF A WEBSITE

Four parts, working together

FRONTEND

What users see

— Buttons
— Images
— Menus
— Forms

BACKEND

What's hidden

— Processes requests
— Manages users
— Talks to the database

DATABASE

Stores information

— User accounts
— Blog posts
— Product info
— Orders

SERVER

Where it all lives

— Stores website files
— Responds to requests worldwide

Every website you've ever used is some combination of these four pieces — even the simplest one-page portfolio site.

06 — HOW A WEBSITE WORKS

Simpler than it first appears — six steps, a few milliseconds

1. A user enters a website address into a browser.
2. The browser locates the server where the website is hosted.
3. The browser sends a request to that server.
4. The server processes the request.
5. The server sends the website files back to the browser.
6. The browser renders and displays the website to the user.

// GO DEEPER — 07 · WHAT HAPPENS WHEN YOU VISIT

Read **"How the Internet Works"** next to see how data physically travels across networks before it ever reaches a server.

08-11 — REAL-WORLD EXAMPLE

Walking through an online clothing store

You open the homepage. The browser requests it from the server. The server responds with HTML, CSS, JavaScript, images, and product data — your browser combines all of it into the page you see.

Search for a product, and another request fires: the server searches the database and returns matching results. This request-response cycle repeats every single time you interact with the site — every click, every scroll, every search.

// UNDERNEATH THE SURFACE

Whether a site is static (fixed content, same for every visitor) or dynamic (content generated per request), and whatever technologies power it, this same request → process → respond → render cycle is the constant. Learn this pattern once, and every framework you touch later will make more sense faster.

13 — COMMON BEGINNER MISTAKES

Where most beginners get stuck

- × Thinking HTML alone creates complete websites.
- × Believing websites and web applications are identical.
- × Ignoring how browsers and servers actually communicate.
- × Learning frameworks before understanding web fundamentals.
- × Focusing on code without understanding the overall system.

14 — PRACTICAL ACTION PLAN

Do this before you write a line of code

1. Open five different websites.
2. Decide whether each one is mainly a website or a web application.
3. Identify their visible frontend components.
4. Think about what backend processes might be happening behind the scenes.
5. Draw a simple diagram of how your browser talks to a server.

15 — KEY TAKEAWAYS

What to carry forward

- A website is a collection of connected web pages hosted on a server.
- Browsers request websites from servers using the internet.
- Websites consist of frontend, backend, databases, and servers working together.
- Websites mainly present information; web applications focus on user interaction.
- Understanding how websites work makes learning web development far easier.

// THE TAKEAWAY

You don't need to memorize this list. You need to be able to explain it in your own words, out loud, to someone who's never coded. That's the real test of understanding.

// 16 — SUMMARY PAGE

Cheat sheet

The whole guide, compressed to eight lines.

website
Collection of connected web pages

server
Stores website files

browser
Displays websites

frontend
What users see

backend
Handles business logic

database
Stores data

domain
Human-friendly website address

the cycle
Every website follows request → response

17 — WEBSITE UNDERSTANDING CHECKLIST

Before you move to Part 2

- [ ] I can explain what a website is.
- [ ] I know the difference between a website and a web application.
- [ ] I understand the roles of the frontend, backend, server, and database.
- [ ] I can explain how a browser loads a website.
- [ ] I understand why websites are the foundation of web development.

If you can check all five honestly, you're ready for the next PDF in this series.

// 18 — RELATED RESOURCES

Keep going

why read it

Frontend vs Backend

Understand how user interfaces and servers work together to build complete web experiences.

why read it

How the Internet Works

Learn how requests travel across networks before reaching a website.

why read it

Static vs Dynamic Websites

Understand why some websites never change while others generate content in real time.

// 19 — RECOMMENDED NEXT LEARNING PATH

Where to go from here

1. **What Is a Website? — you are here**
2. Website vs Web Application
3. How the Internet Works
4. What Happens When You Type a URL?

haas.dev

Engineering mindset over syntax memorization. Learn to think like a systems builder, one fundamental at a time.

haas.dev