Introduction to Web Applications

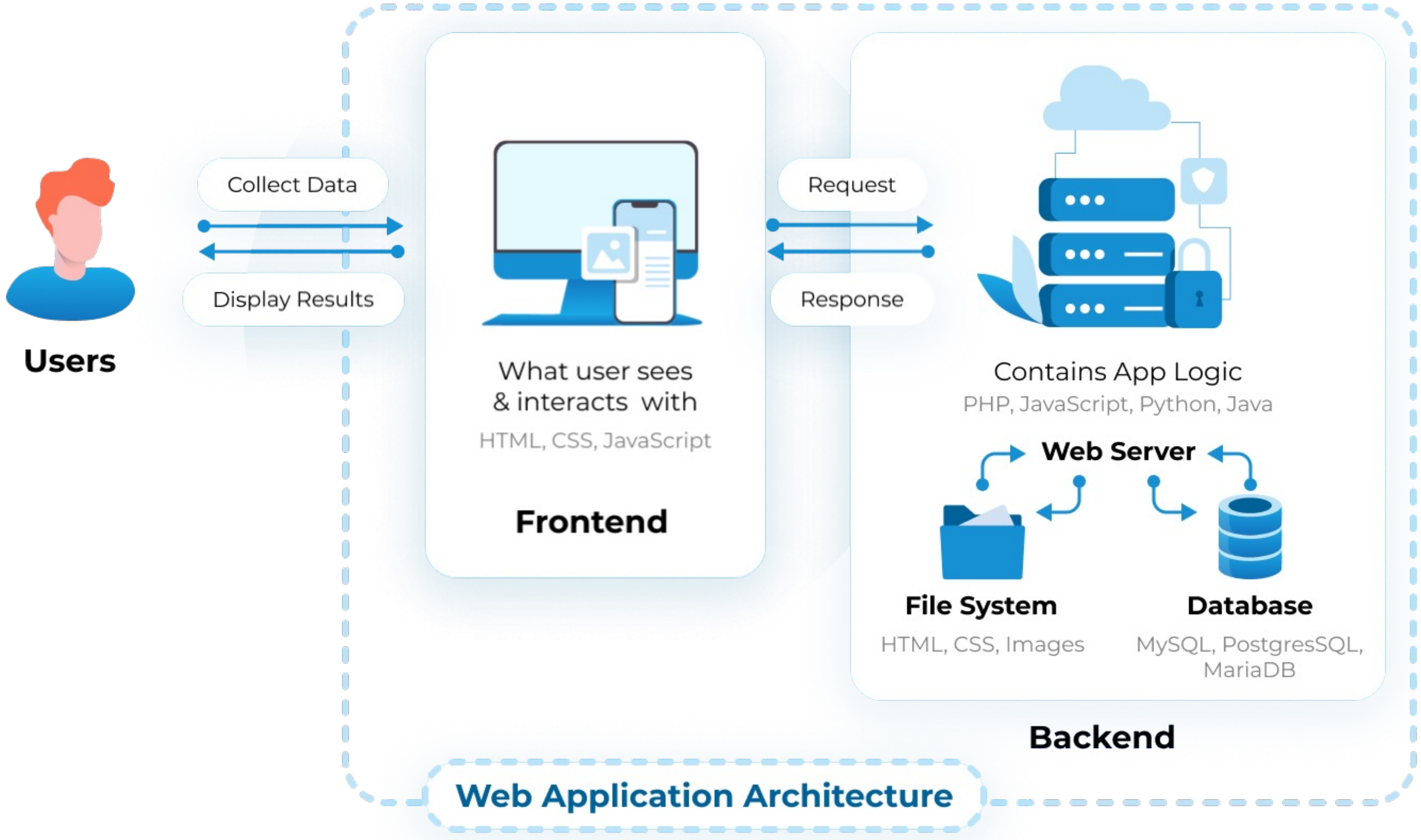
Web architectures

Juan Pablo Sáenz

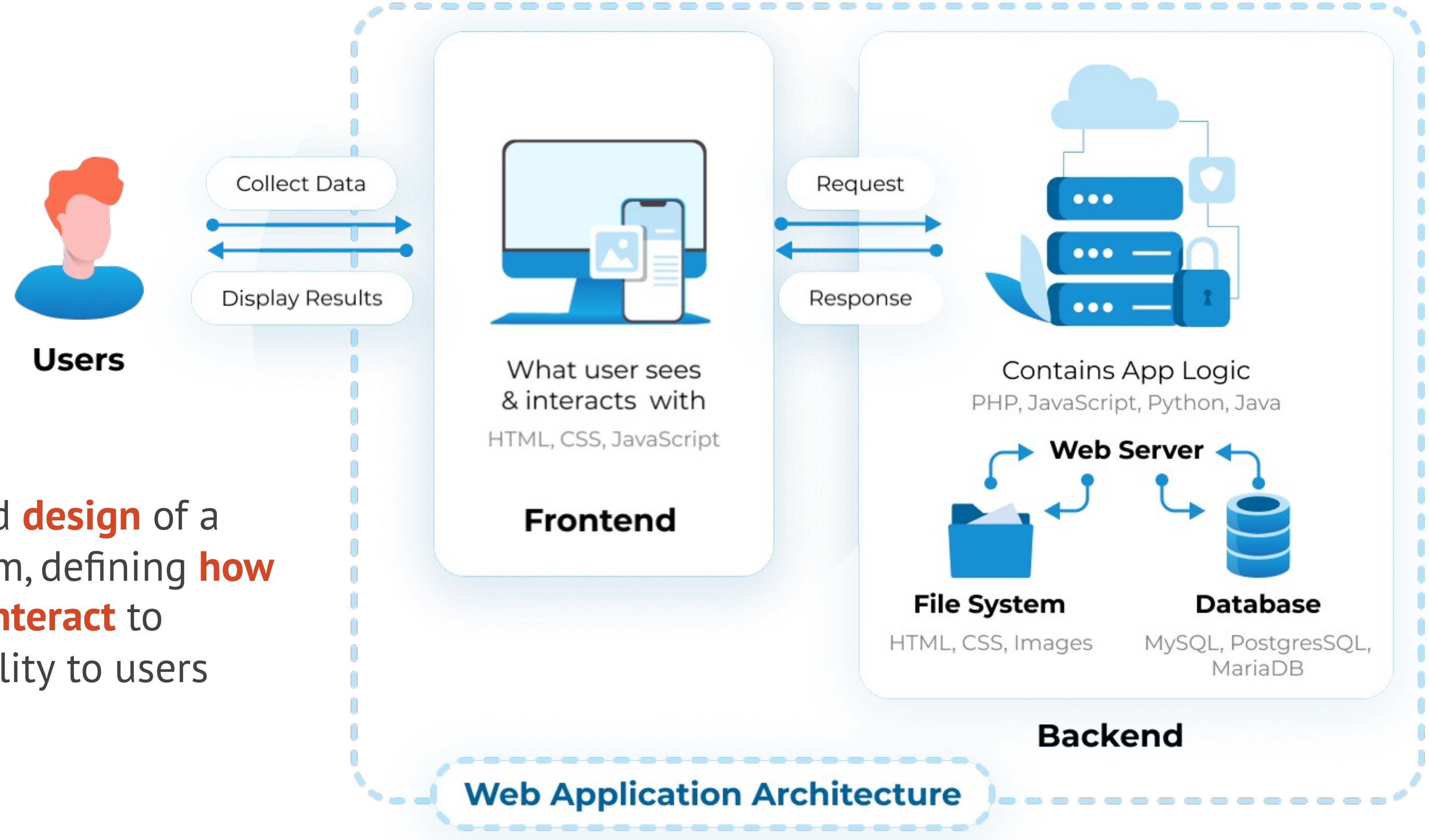
Goals

- Understanding **web architecture** and its importance.
- Identify key components and network **protocols**.
- Analyze **interaction** and **communication** between components.
- Get an overview of the **course topics** 🗺️ and how they fit together in the perspective of building a **complete web application**.

🗺️ The course's most important slide



🗺️ The course's most important slide



The **structure** and **design** of a web-based system, defining **how its components interact** to deliver functionality to users

Web architecture

Once the architecture is established developers can determine the appropriate

- **technologies** (e.g., **databases**, caching systems),
- **frameworks** (**Flask**, React, Django, Spring Boot),
- **languages** (JavaScript, **Python**, Java),
- **protocols** (**HTTP/HTTPS**, WebSockets, GraphQL),
- and **infrastructure** (**cloud platforms**, containerization, server hosting)

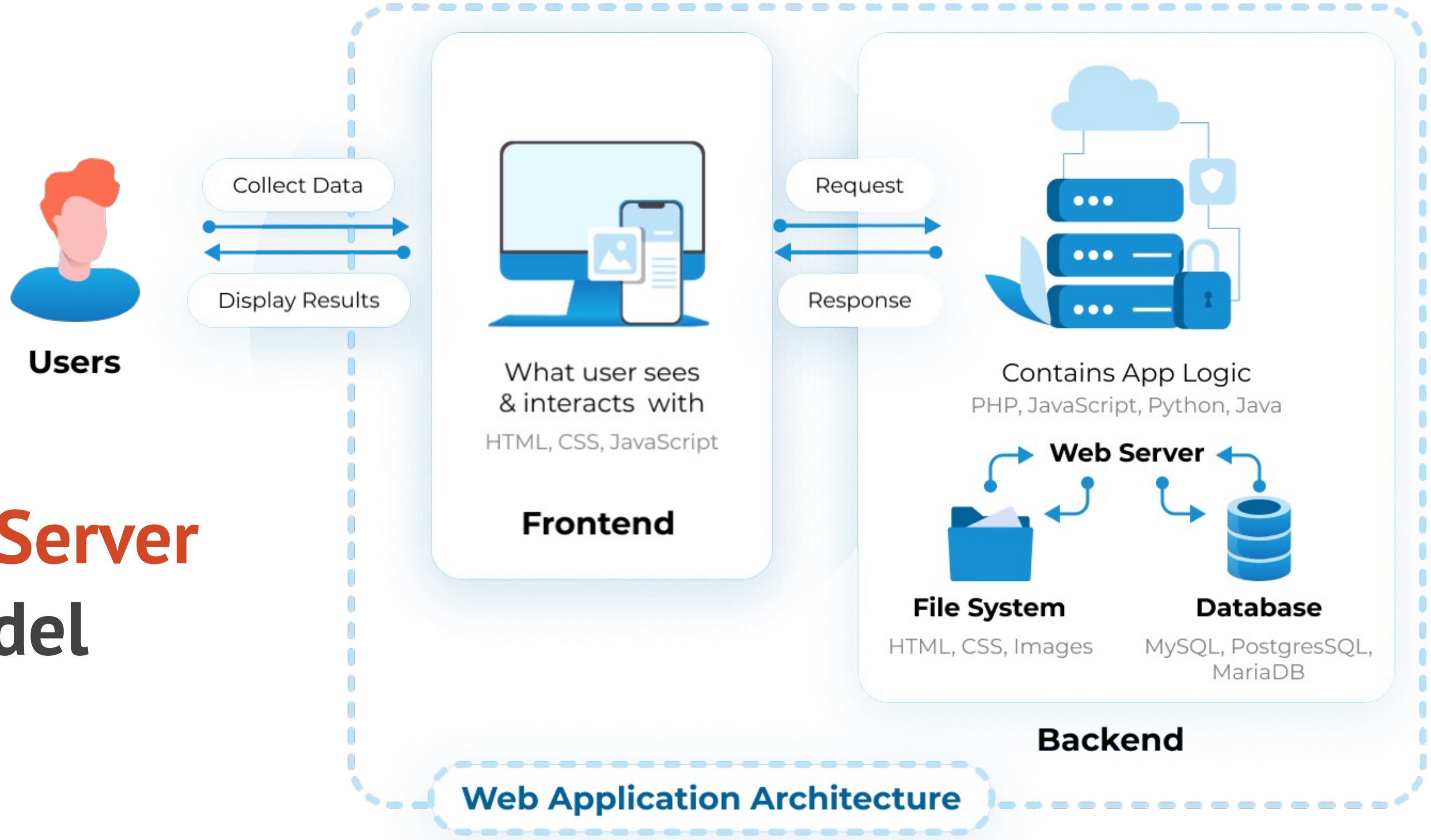
Web architecture

⚠️ Errors in **architecture design** are significantly **more costly** and **difficult** to fix than errors in implementation

- redesigning infrastructure,
- rewriting large portions of the codebase,
- or even migrating to a different technology stack.

Therefore, **careful architectural planning is essential** before moving to implementation!

🗺️ The course's most important slide



Users

Collect Data

Display Results

What user sees
& interacts with

HTML, CSS, JavaScript

Frontend

Request

Response

Contains App Logic

PHP, JavaScript, Python, Java

Web Server

File System

HTML, CSS, Images

Database

MySQL, PostgresSQL, MariaDB

Backend

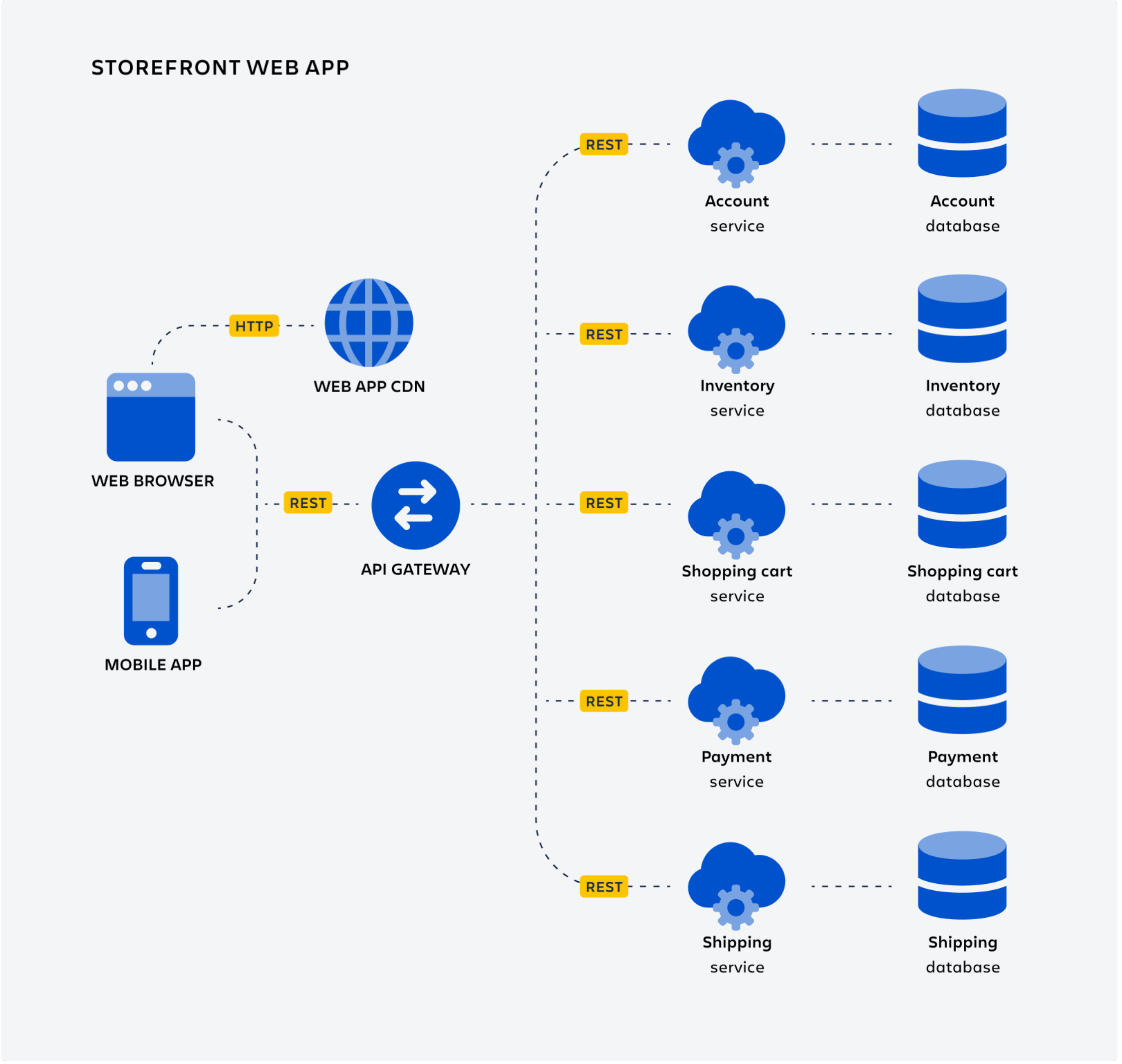
Web Application Architecture

Client-Server Model

A random architecture example

Microservices Architecture:

What can we infer from it?

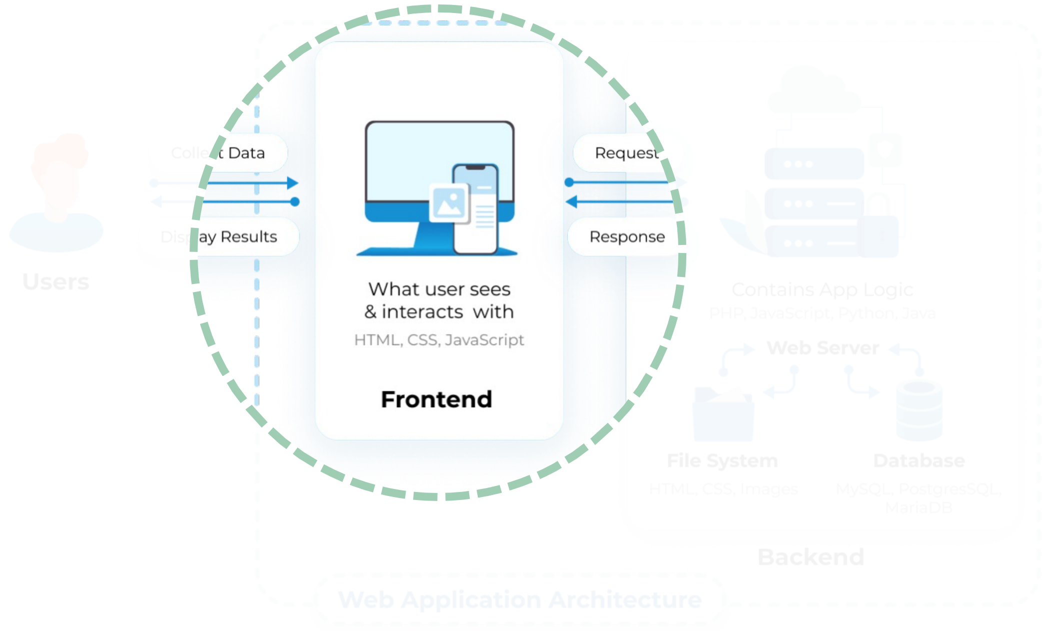


🗺️ Web architecture components: Frontend

Frontend: what the user sees and interacts with

Languages:

- **HTML:** a markup language used to **structure content on the web**. It defines elements like headings, paragraphs, images, and links.
- **CSS:** a style sheet language used to **control the presentation of HTML elements**, such as colors, fonts, and layouts.
- **JavaScript:** a programming language used to **add interactivity and dynamic behavior** to web pages.

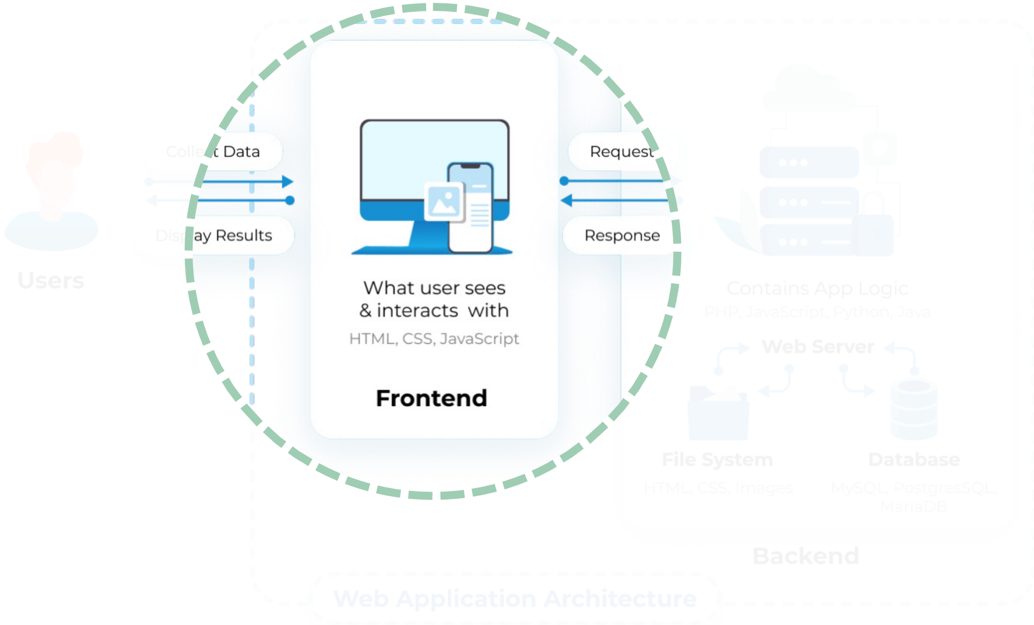


🗺️ Web architecture components: Frontend

Frontend: what the user sees and interacts with

Applications:

- **Browser:** an application that **retrieves, interprets, and displays web content**, including HTML, CSS, and JavaScript.



🗺️ Web architecture components: Backend

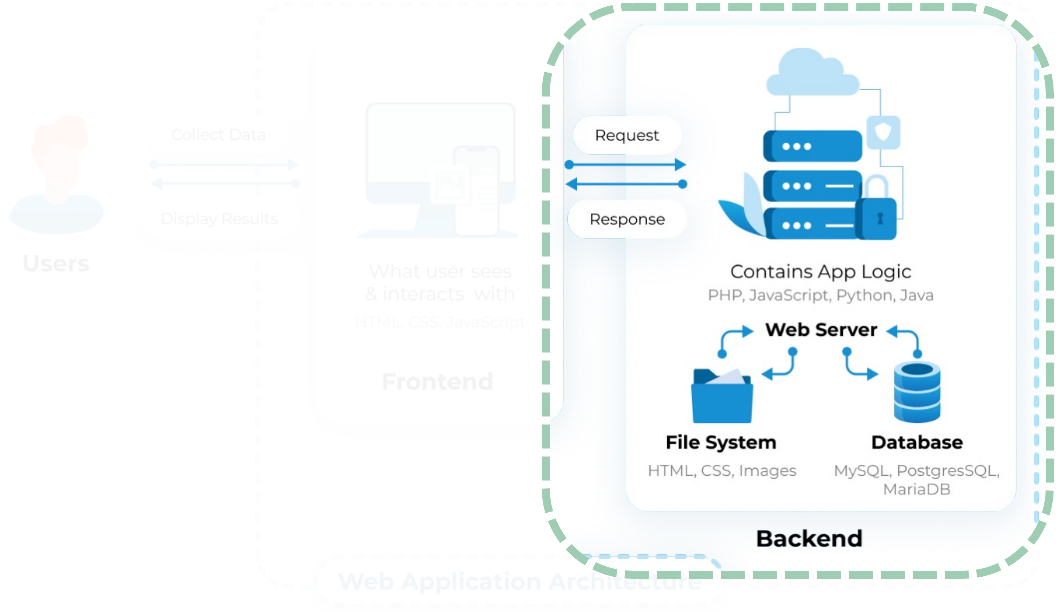
Backend: The part of a web application that works behind the scenes, handling **data**, **logic**, and server **communication**.

🧩 **Components**:

Server: a computer or system that provides **resources**, **data**, **services**, or **programs** to other computers, known as clients, over a network.

Web server: **hosting websites**, **handling HTTP requests** from clients (like browsers), and **delivering web pages**, images, and other content to the client's device.

Intermediary between the client and the **database**

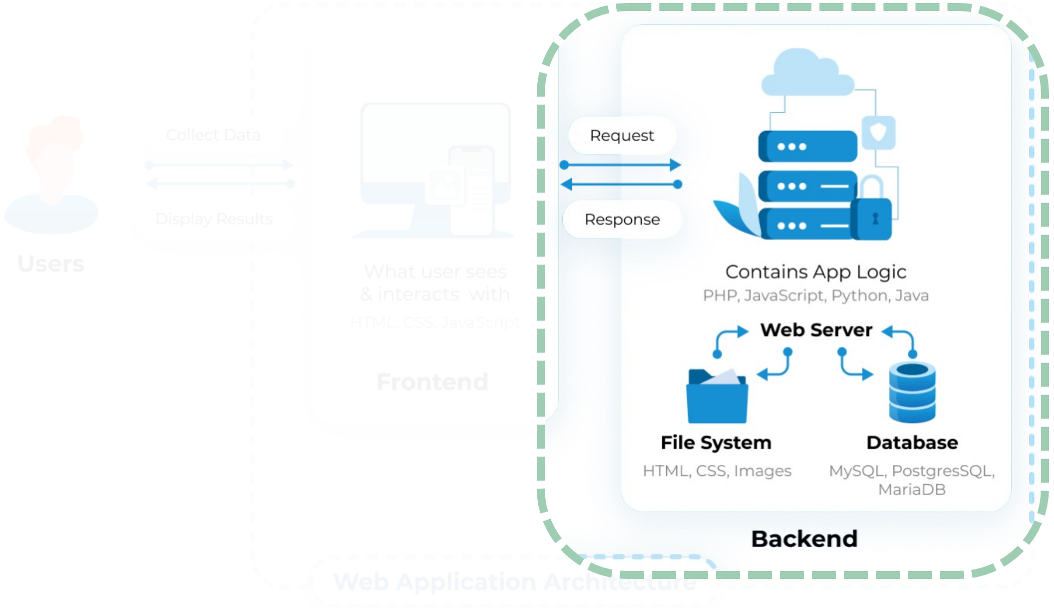


🗺️ Web architecture components: Backend

Database: a system that **stores and organizes data**, making it easy to retrieve, manage, and update.

- It ensures data integrity, security, and performance, often structured in **tables**, **rows**, and **columns**.

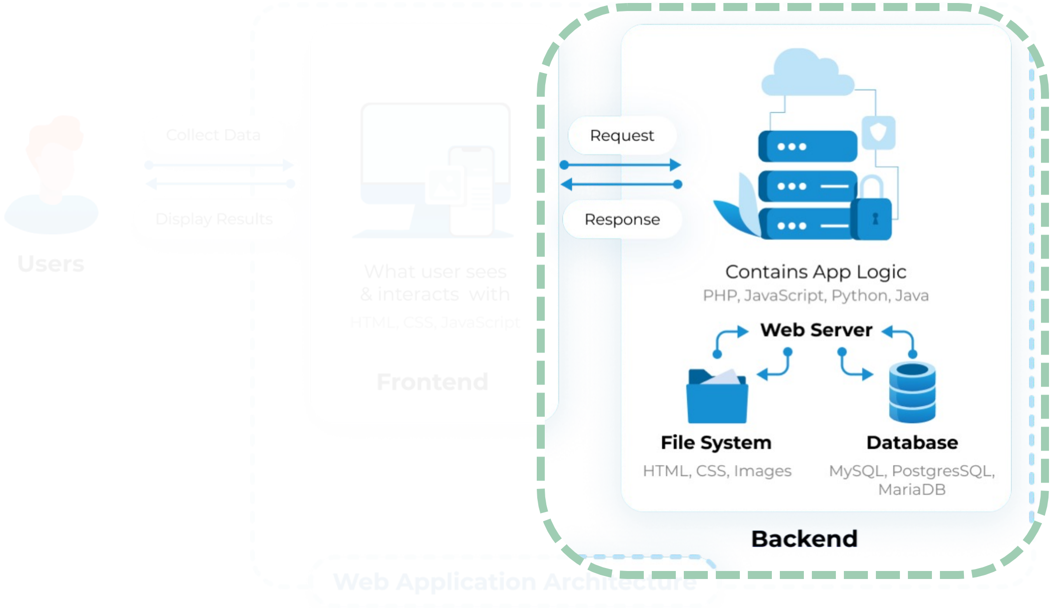
- **SQL (Structured Query Language)** is the language used to **interact with databases**, allowing users to search, insert, update, and delete data.



Web architecture components: Backend

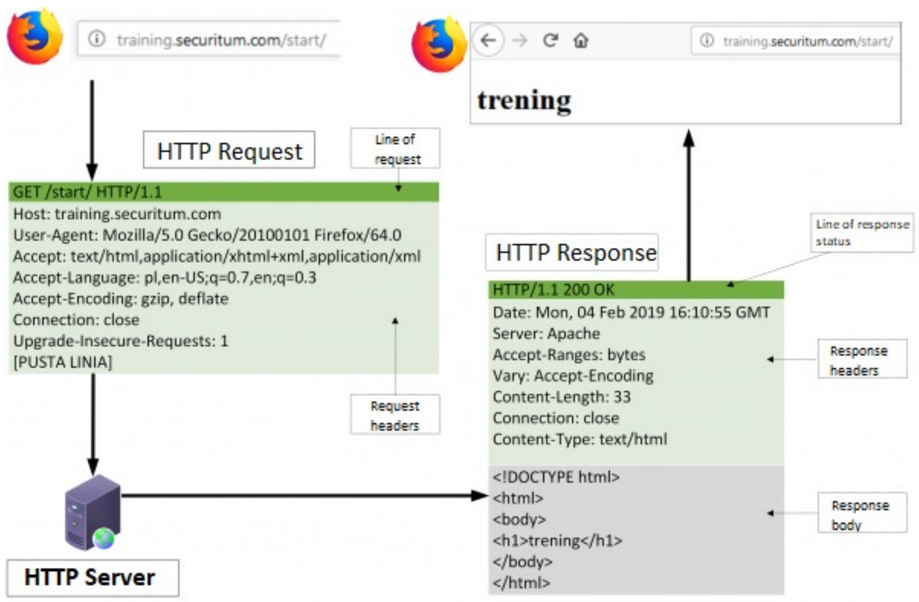
File system: a system that organizes and stores **files** on a storage device.

- Manages how **data is saved**, **accessed**, and **retrieved** in the backend of an application.



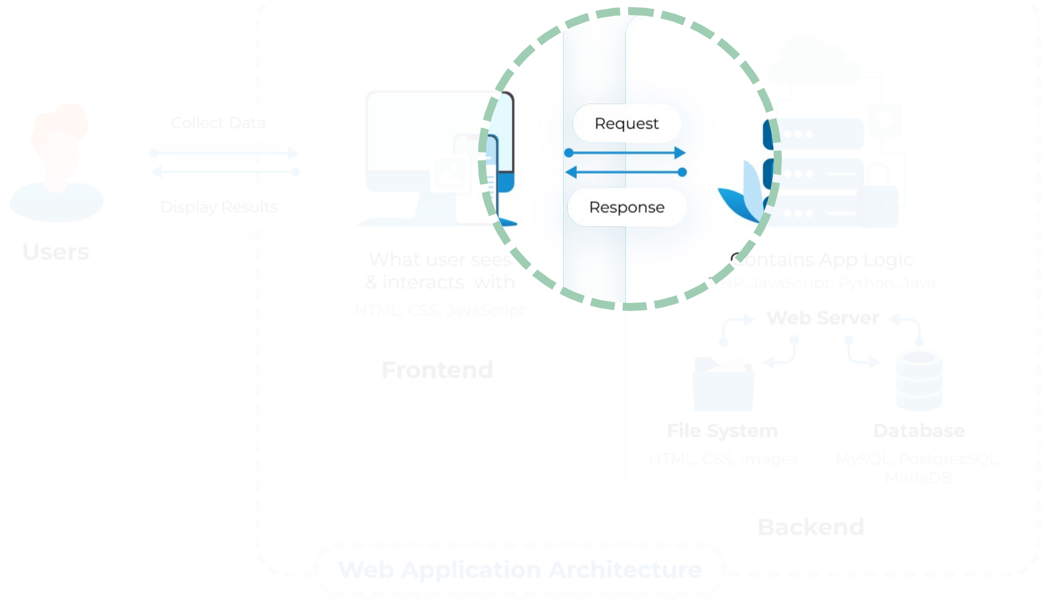
🗺️ Web architecture components: HTTP Protocol

HTTP Protocol: the protocol used for **transferring data over the web**, allowing communication between **clients** (like browsers) and **servers** by sending requests and receiving responses

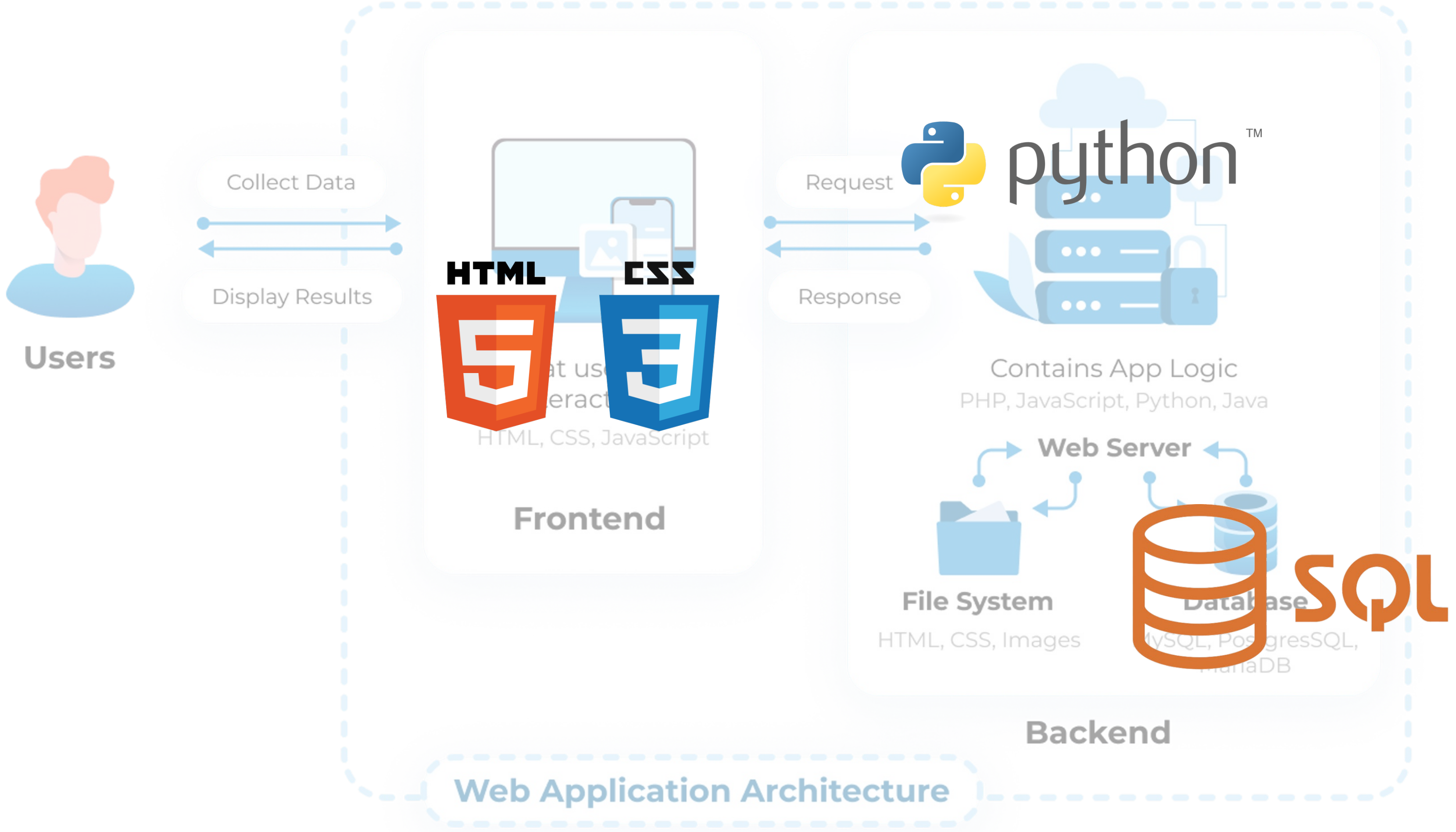


```
GET /start/ HTTP/1.1
Host: training.securitum.com
User-Agent: Mozilla/5.0 Gecko/20100101 Firefox/64.0
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
[PUSTA LINIA]
```

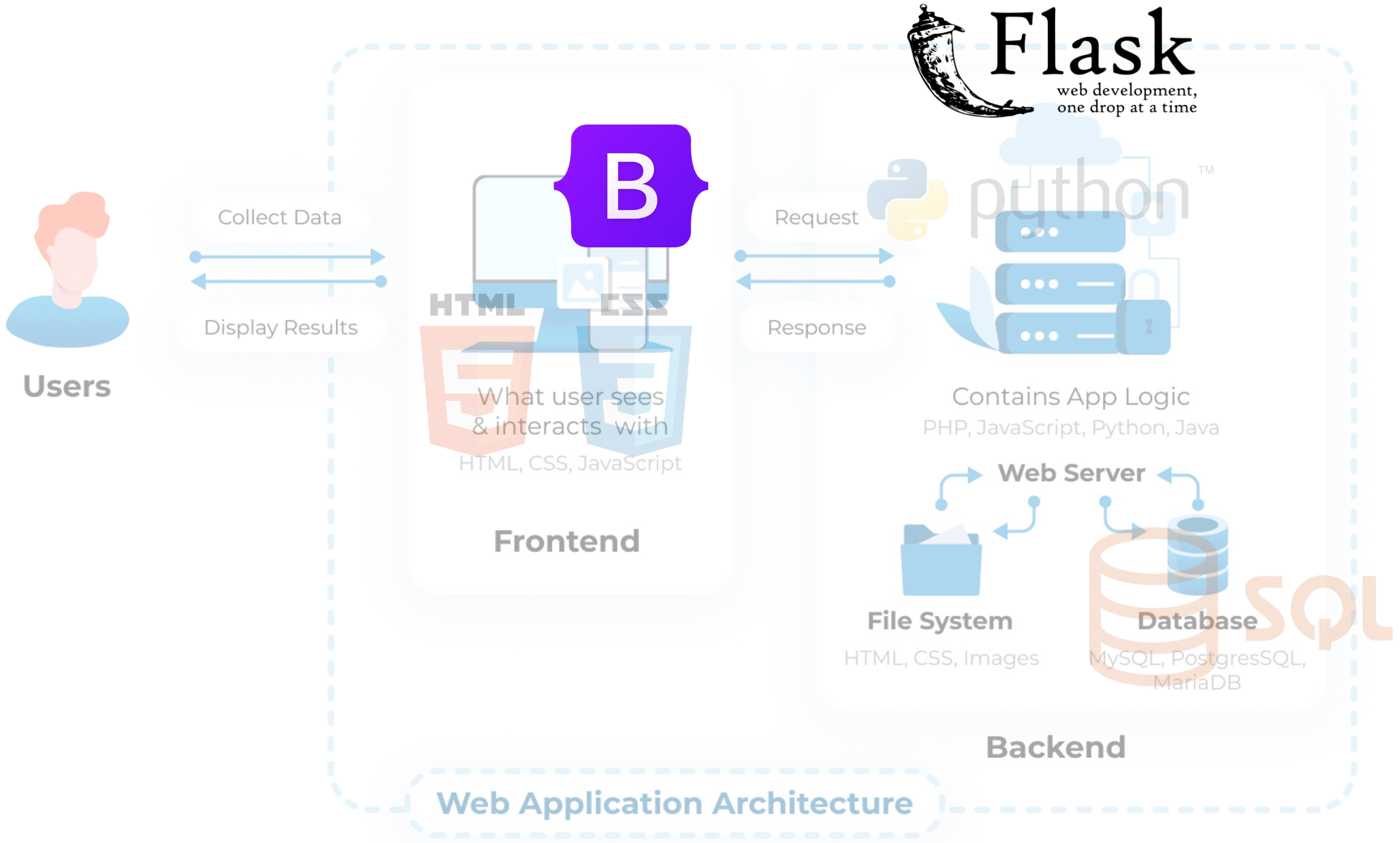
```
HTTP/1.1 200 OK
Date: Mon, 04 Feb 2019 16:10:55 GMT
Server: Apache
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Length: 33
Connection: close
Content-Type: text/html
<!DOCTYPE html>
<html>
<body>
<h1>trening</h1>
</body>
</html>
```



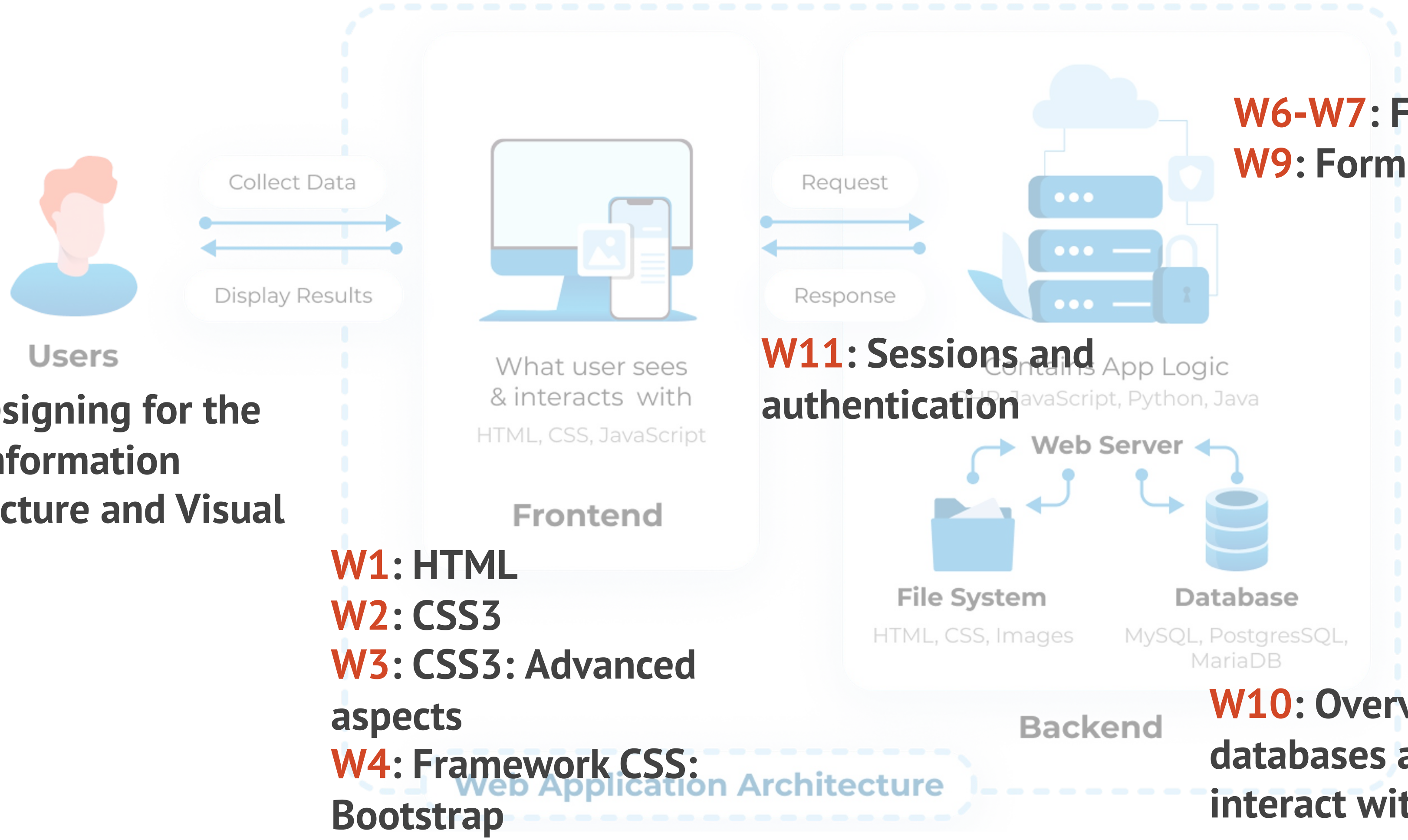
The 📝 languages we will use in the course



The 🛠 frameworks we will use in the course



The 📚 topics we will address throughout the course



W5: Designing for the Web: Information Architecture and Visual Design

W1: HTML
W2: CSS3
W3: CSS3: Advanced aspects
W4: Framework CSS: Bootstrap
W8: Form in HTML

W11: Sessions and authentication

W6-W7: Flask
W9: Form in Flask

W10: Overview of databases and how to interact with them

Licenza

- These slides are distributed under a Creative Commons license "**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**"
- **You are free to:**
  - **Share** — copy and redistribute the material in any medium or format
  - **Adapt** — remix, transform, and build upon the material
  - The licensor cannot revoke these freedoms as long as you follow the license terms.
- **Under the following terms:**
  - **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
  - **NonCommercial** — You may not use the material for commercial purposes.
  - **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  - **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.
- https://creativecommons.org/licenses/by-nc-sa/4.0/