



Build a Fullstack App with
Vanilla JS and Go

Maximiliano Firtman
firt.dev

X x.com/firt

in linkedin.com/in/firtman

github.com/firtman

About me

Maximiliano Firtman



MOBILE+WEB DEVELOPER

HTML since 1996

JavaScript since 1998, +160 web apps

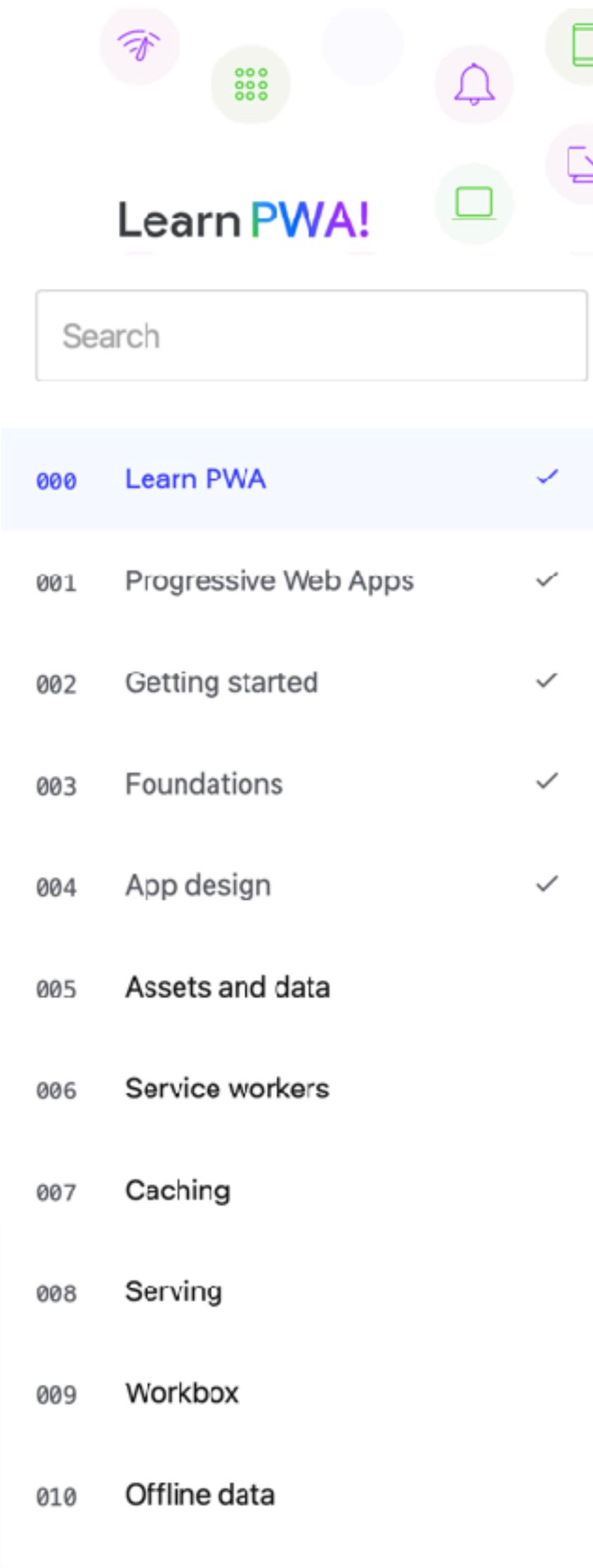
AUTHOR

Authored 14 books, +70 courses



About me

Maximiliano Firtman



The screenshot shows the 'Learn PWA!' course interface. At the top, there are several circular icons with icons inside: a purple Wi-Fi icon, a green grid icon, a white circle icon, a purple bell icon, a green smartphone icon, and a purple download icon. Below these is the title 'Learn PWA!' in a purple and white font. Underneath the title is a search bar with the placeholder 'Search'. A vertical red line on the left marks the start of the sidebar. The sidebar contains a list of 10 modules, each with a blue number and a title, followed by a checked checkbox icon:

- 000 Learn PWA
- 001 Progressive Web Apps
- 002 Getting started
- 003 Foundations
- 004 App design
- 005 Assets and data
- 006 Service workers
- 007 Caching
- 008 Serving
- 009 Workbox
- 010 Offline data

000

Learn PWA

A course that breaks down every aspect of modern progressive web app development.

Welcome to Learn Progressive Web Apps!

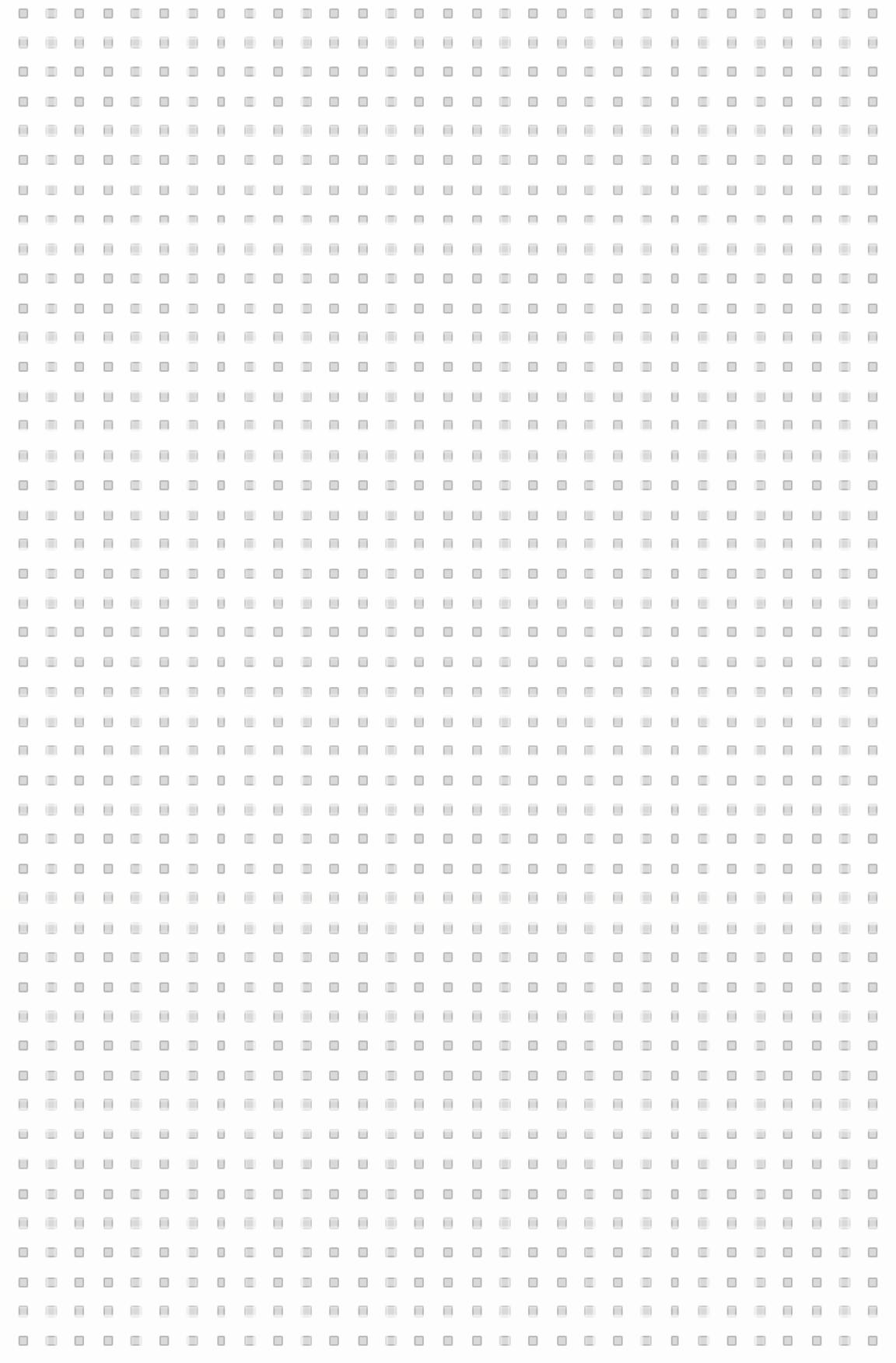
Welcome to Learn Progressive Web Apps!

This course covers the fundamentals of Progressive Web App development in easy-to-understand pieces. Over the following modules, you'll learn what a Progressive Web App is, how to create one or upgrade your existing web content, and how to add all the pieces for an offline, installable app. Use the menu pane to navigate the modules. (The menu is at left on desktop or behind the hamburger menu on mobile.)

You'll learn PWA fundamentals like the Web App Manifest, service workers, how to design with an app in mind, how to use other tools to test and debug your PWA. After these fundamentals, you'll learn about integration with the platform and operating system, how to enhance your PWA's installation and usage experience, and how to offer an offline experience.

About me

Maximiliano Firtman



Frontend **Masters**

Frontend Courses

Mobile App Courses

Backend Courses

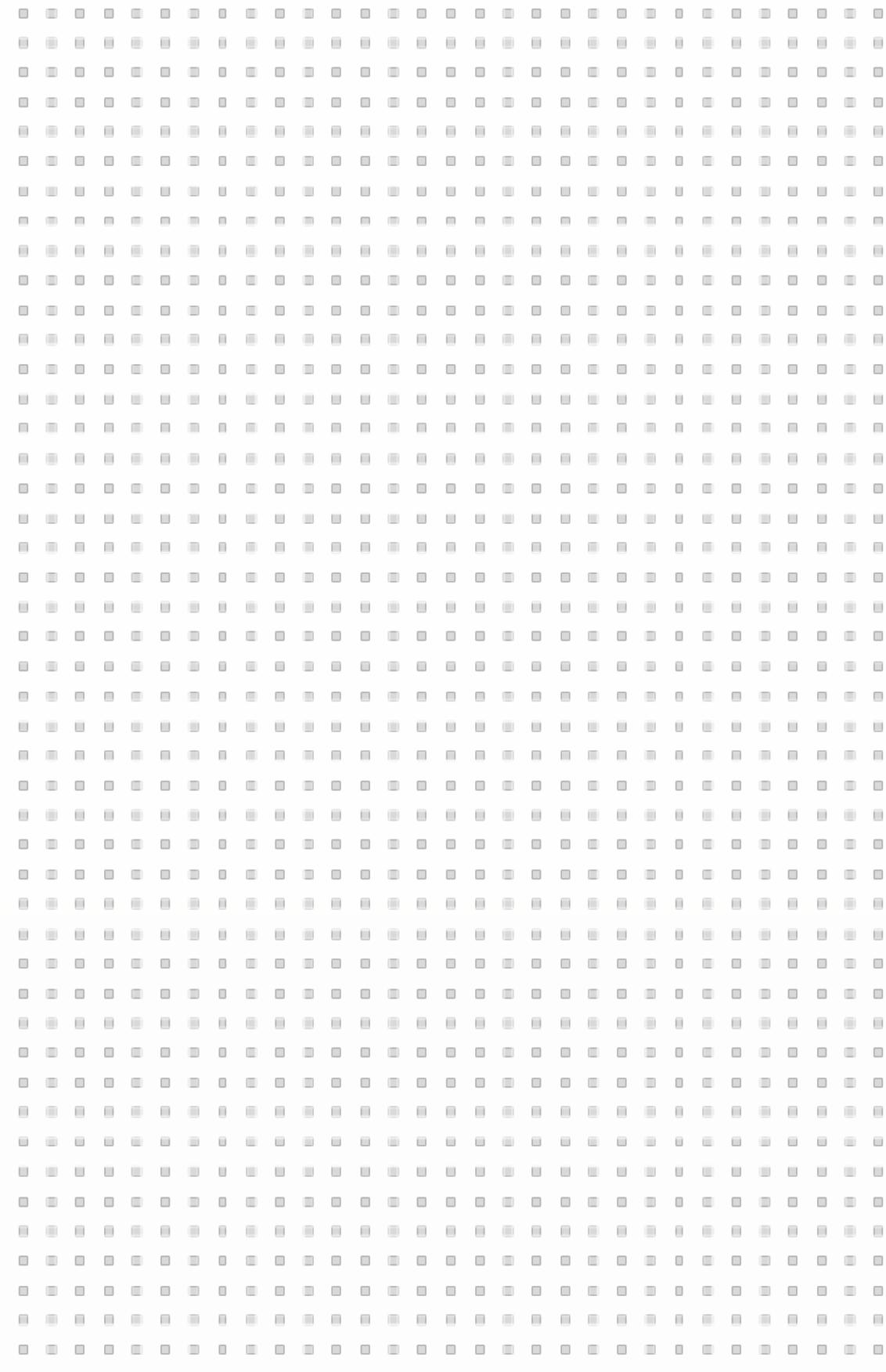
About me

Maximiliano Firtman

Frontend *Masters*



Contents



Introduction to the Project

Setting Up a Backend with Go

Database Schema and REST API

Setting Up the Frontend

Web Components

Dynamic Rendering

Contents

Search, Filter and Sort

Client-side Routing

State Management

Implementing Authentication

Adding Favorites and Watchlist

Pre-requisites

Basic JavaScript, HTML, and CSS experience

Familiarity with JavaScript and DOM APIs

Basic Go backend development

Basic SQL and database concepts

Go 2.24+

Code editor and a browser

Workshop Site

github.com/firtman/go-vanillajs

Questions?



The Project

'future' movies

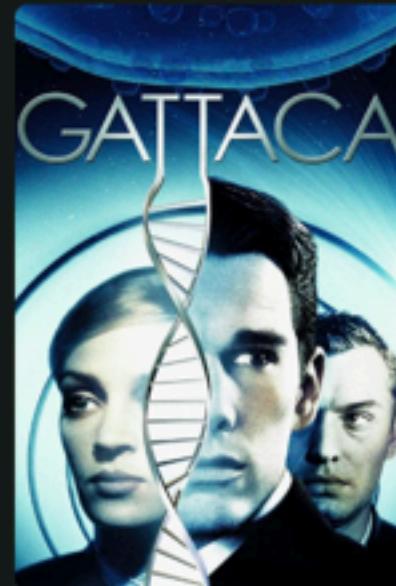
▼

Sort by Score

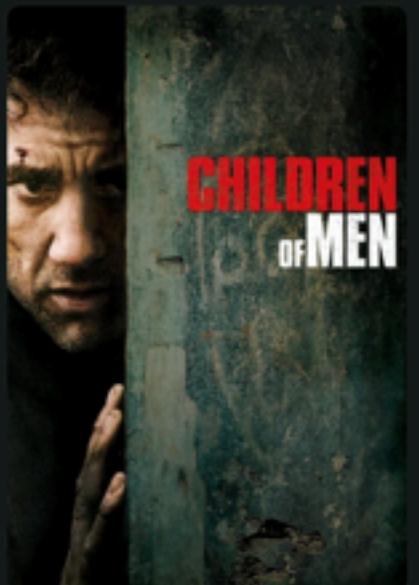
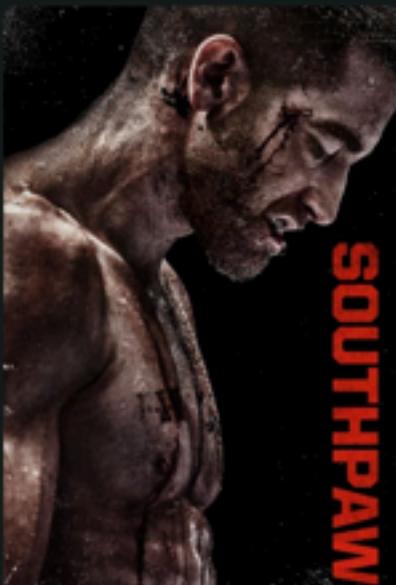
▼

Back to the Future
(1985)

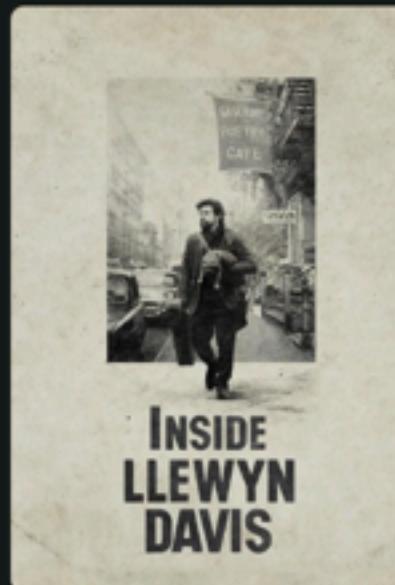
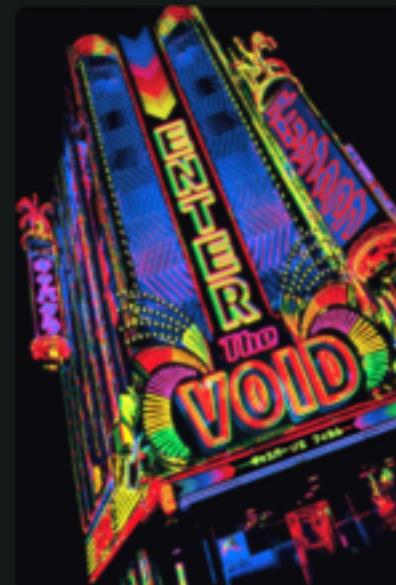
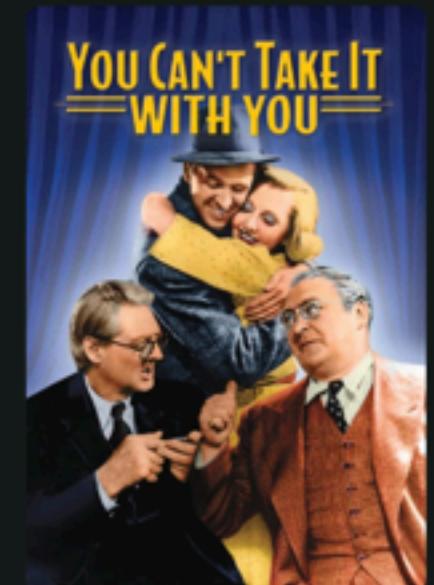
Her (2013)

Terminator 2:
Judgment Day
(1991)Pride & Prejudice
(2005)

Gattaca (1997)

X-Men: Days of
Future Past (2014)Back to the Future
Part II (1989)Children of Men
(2006)The Terminator
(1984)

Southpaw (2015)

Inside Llewyn Davis
(2013)Enter the Void
(2009)The Muppet
Christmas Carol
(1992)You Can't Take It
With You (1938)

Her

A Spike Jonze Love Story



Her - Official Trailer 2 [HD]

Watch later Share

Watch on YouTube

Release Date
2013

Score
7.9 / 10

Original language
en

Add to Favorites

Add to Watchlist

Science Fiction

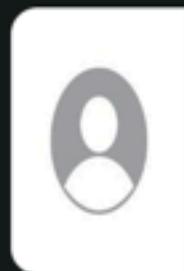
Drama

Romance

In the not so distant future, Theodore, a lonely writer purchases a newly developed operating system designed to meet the user's every needs. To Theodore's surprise, a romantic relationship develops between him and his operating system. This unconventional love story blends science fiction and romance in a sweet tale that explores the nature of love and the ways that technology isolates and connects us all.



Scarlett Johansson



Amy Adams



Bill Hader

Technologies

- **Backend:** Go
- **Database:** Postgres
- **Frontend:** HTML+CSS+JavaScript (Vanilla)
- **Communication:** JSON RESTful APIs

Best Practices

- Single Page Application (SPA)
- Repository
- Clean code
- No frameworks
- External Libraries to the minimum
- Fast to execute and render
- Keeping the code easy to understand
- Use Environmental Variables
- Keep the Data Provider agnostic

Output

- Go backend project
- Go installation project
- Frontend public directory for frontend
- Compiled Go backend
- Basic documentation

Database

- PostgreSQL
- Your own installation (full or with Docker)
- Cloud-based provider
 - Supabase: **supabase.com**
 - Aiven: **aiven.io/postgresql**
 - Neon: **neon.tech**

Data



- The Movie Database (TMDB)
- It's an open dataset
- ~5,000 subset movies with meta data
- We have it already available, we just have to populate our database with it
- Images will come from TMDB online server
- Video trailers will come from YouTube

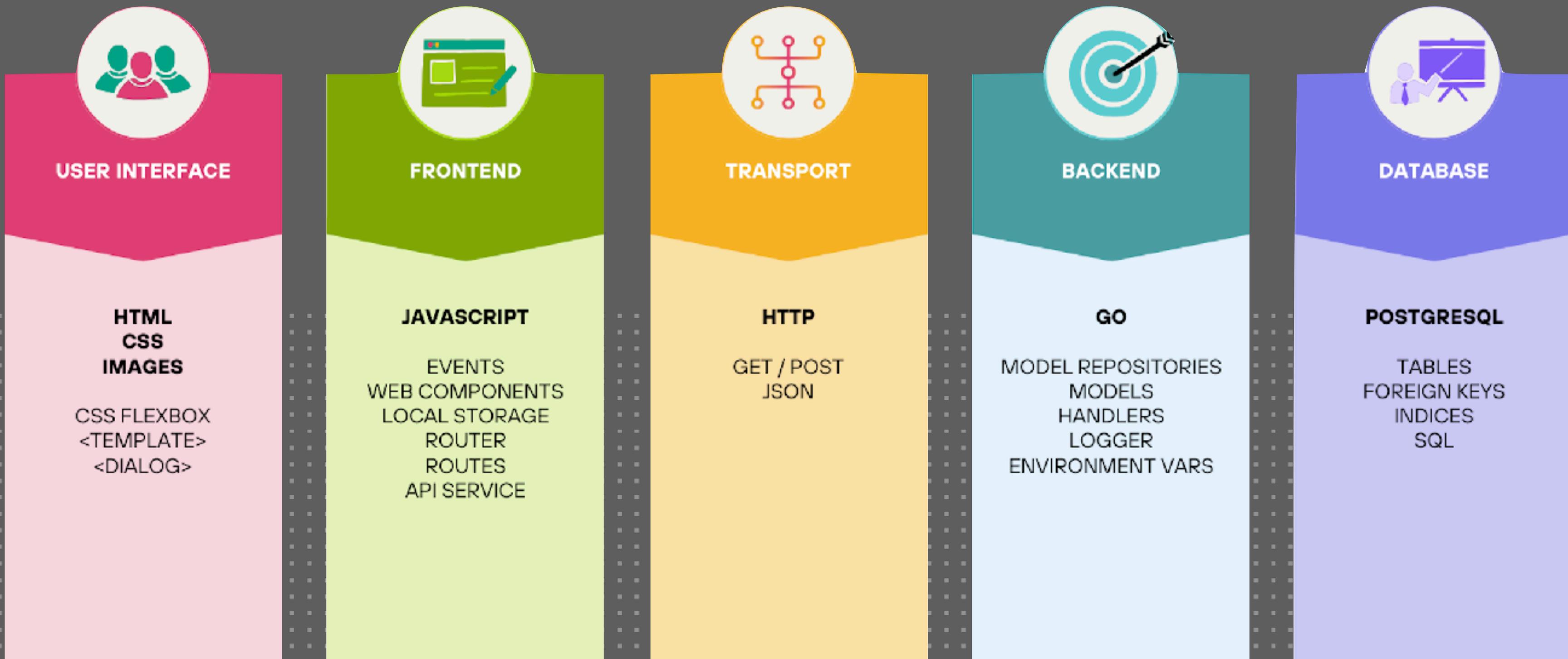
Options available

- See Top and Recent Movies
- Search Movies
- Movie Details
- User Registration
- User Authentication
- Favorites Movies
- Watchlist

Lab time

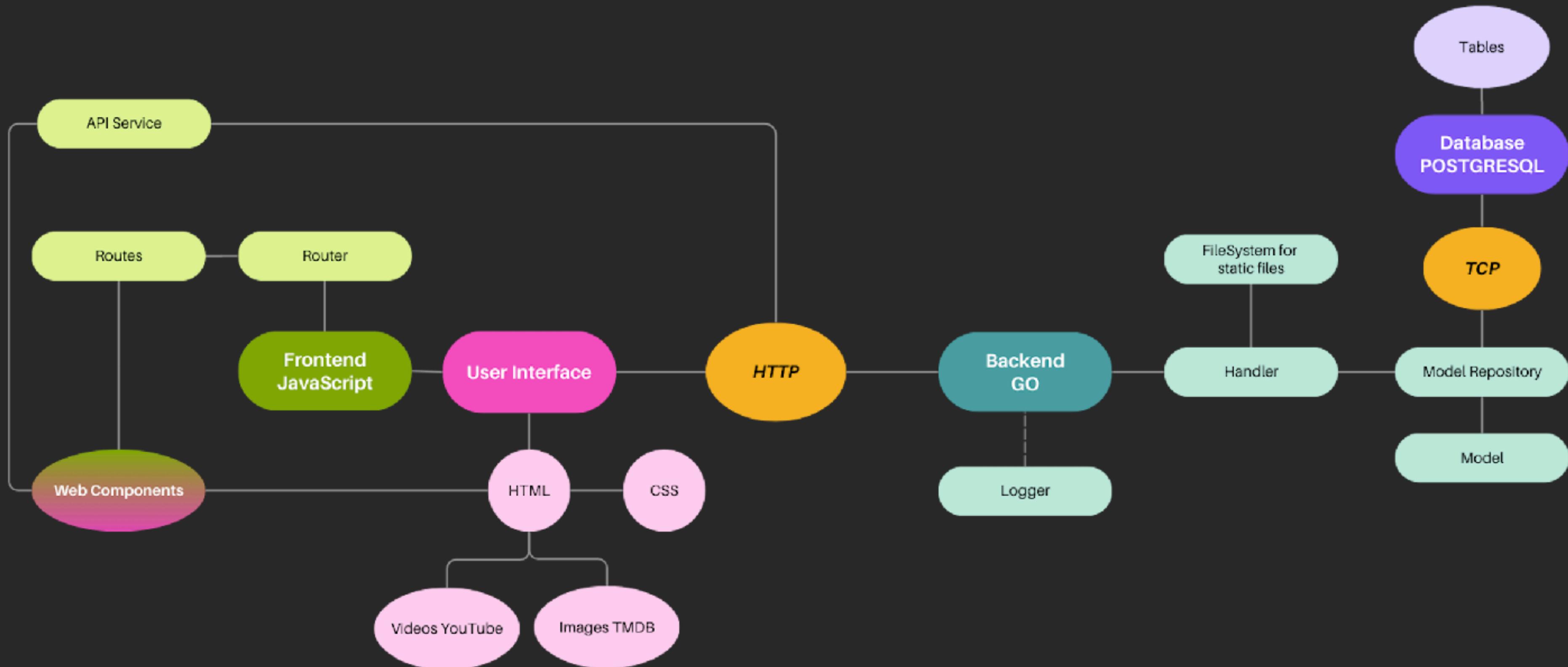
Architecture

Main aspects



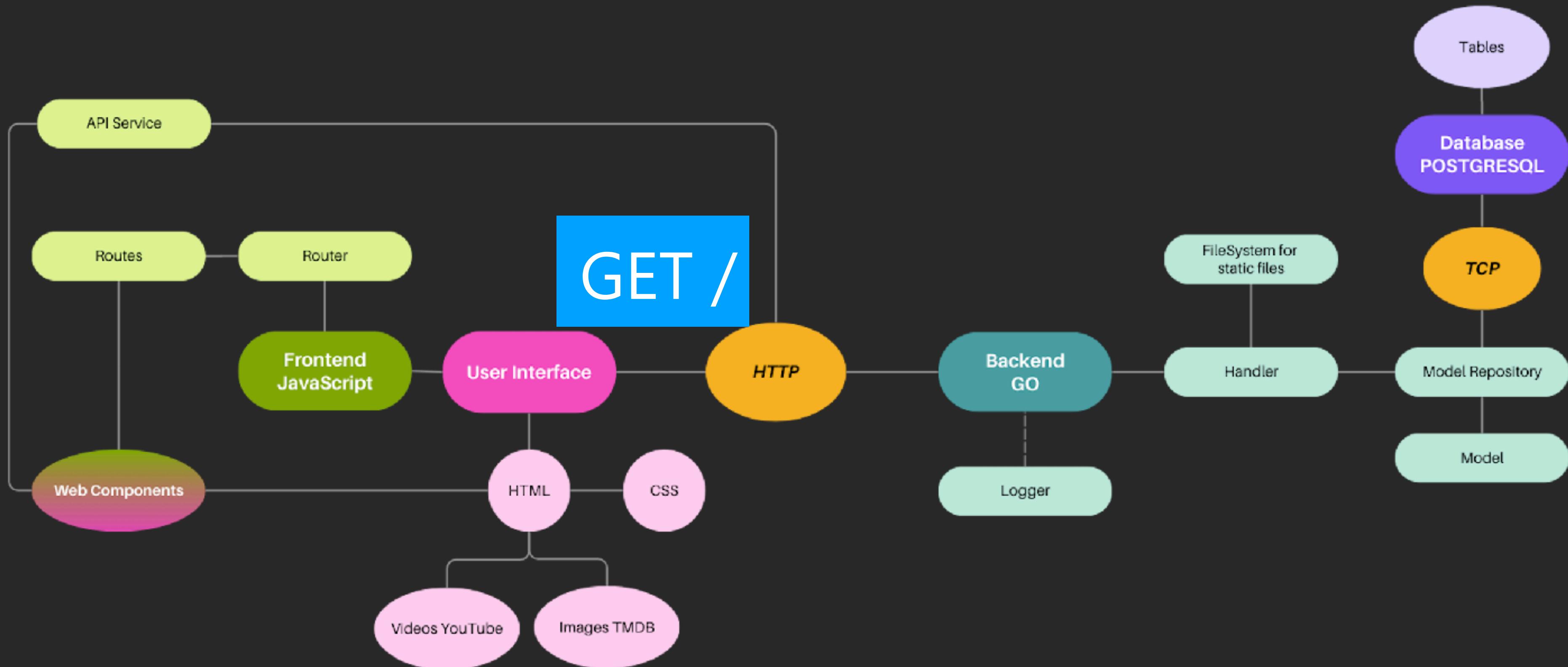
Architecture

Main aspects



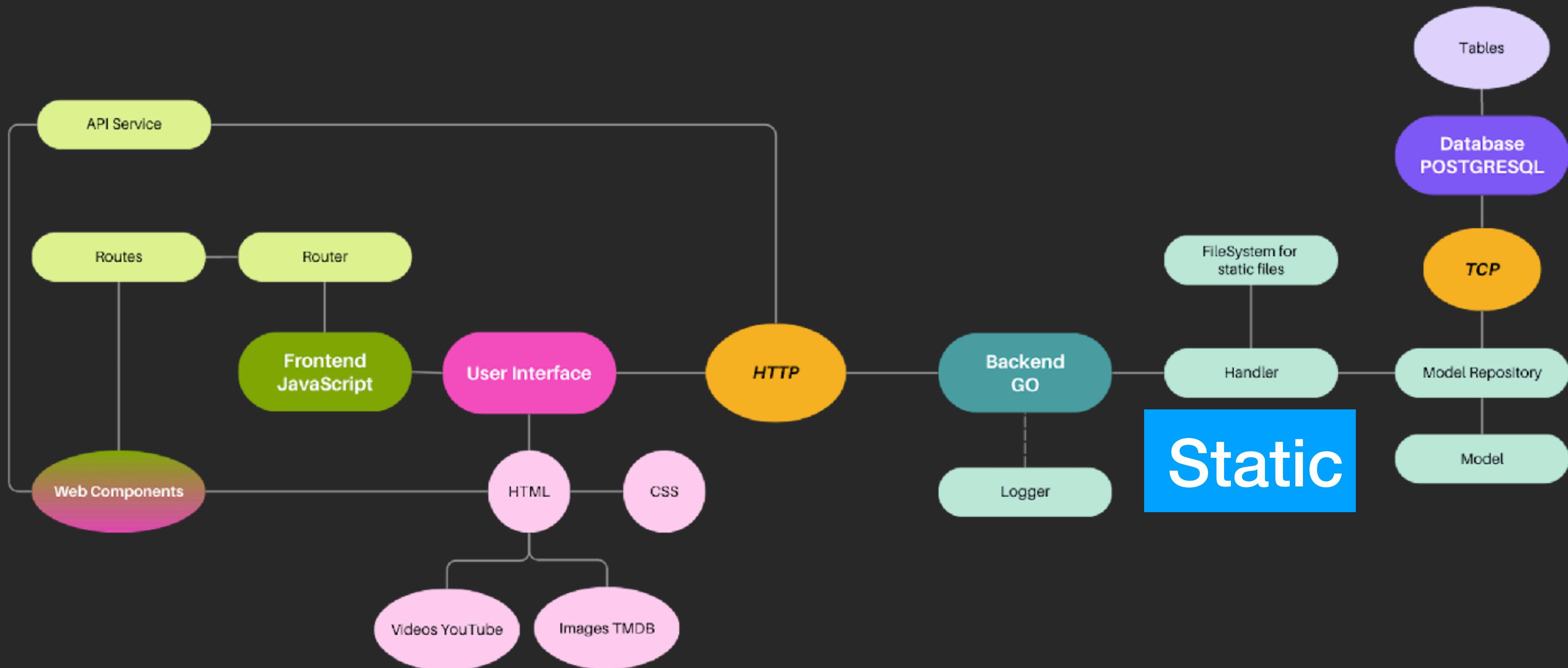
Architecture

Main aspects



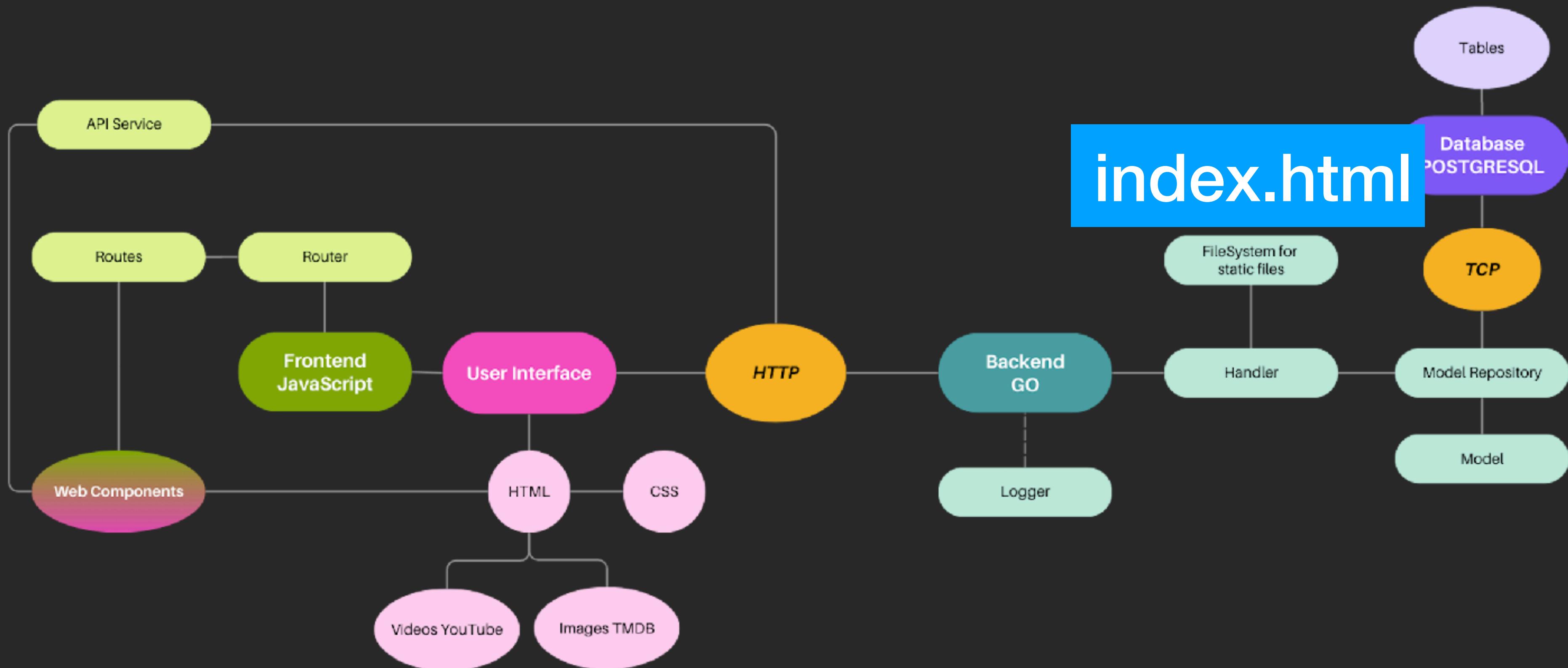
Architecture

Main aspects

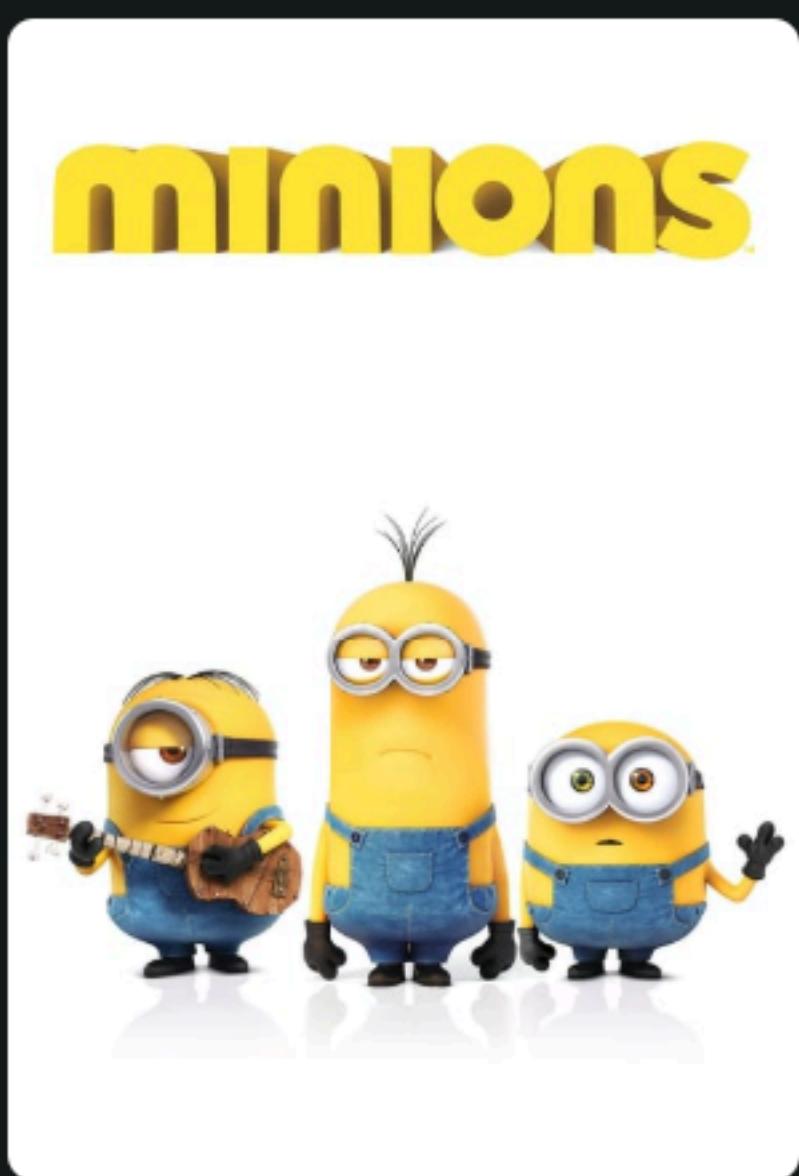


Architecture

Main aspects



This Week's Top 10



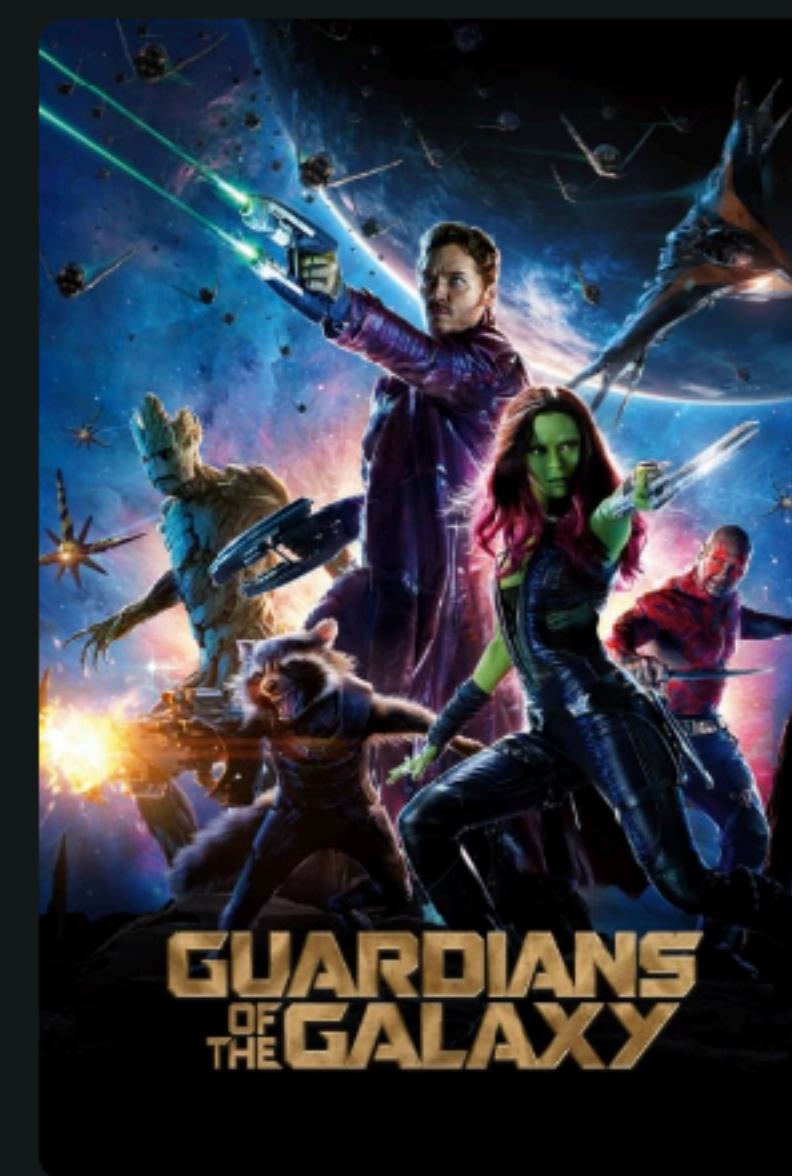
Minions (2015)



Interstellar (2014)



Deadpool (2016)

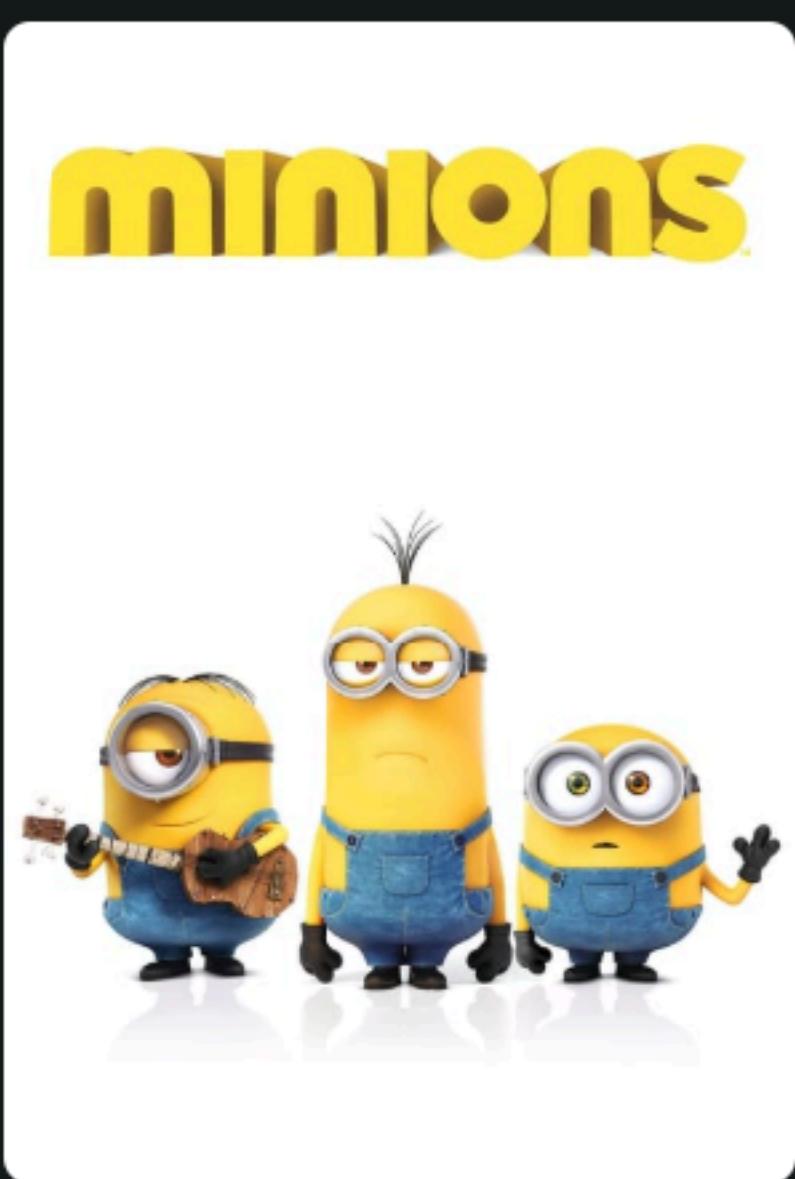


Guardians of the Galaxy (2014)



Mad M

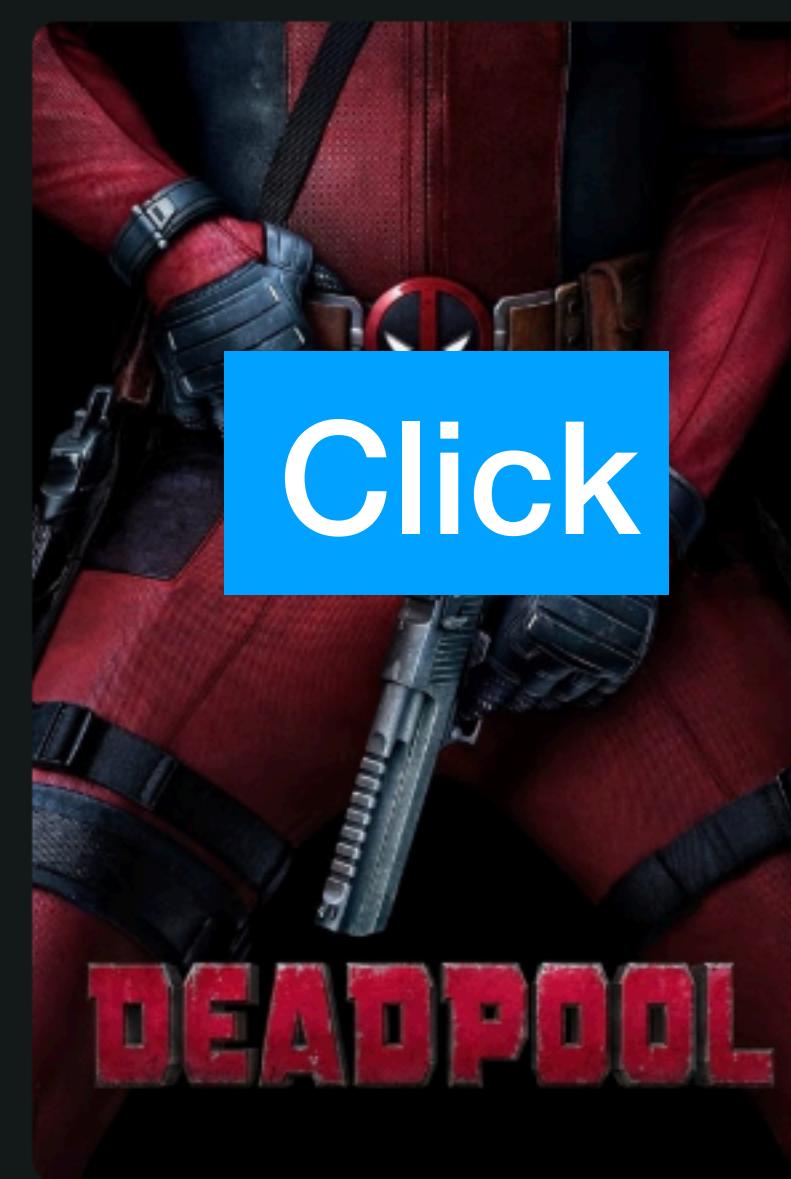
This Week's Top 10



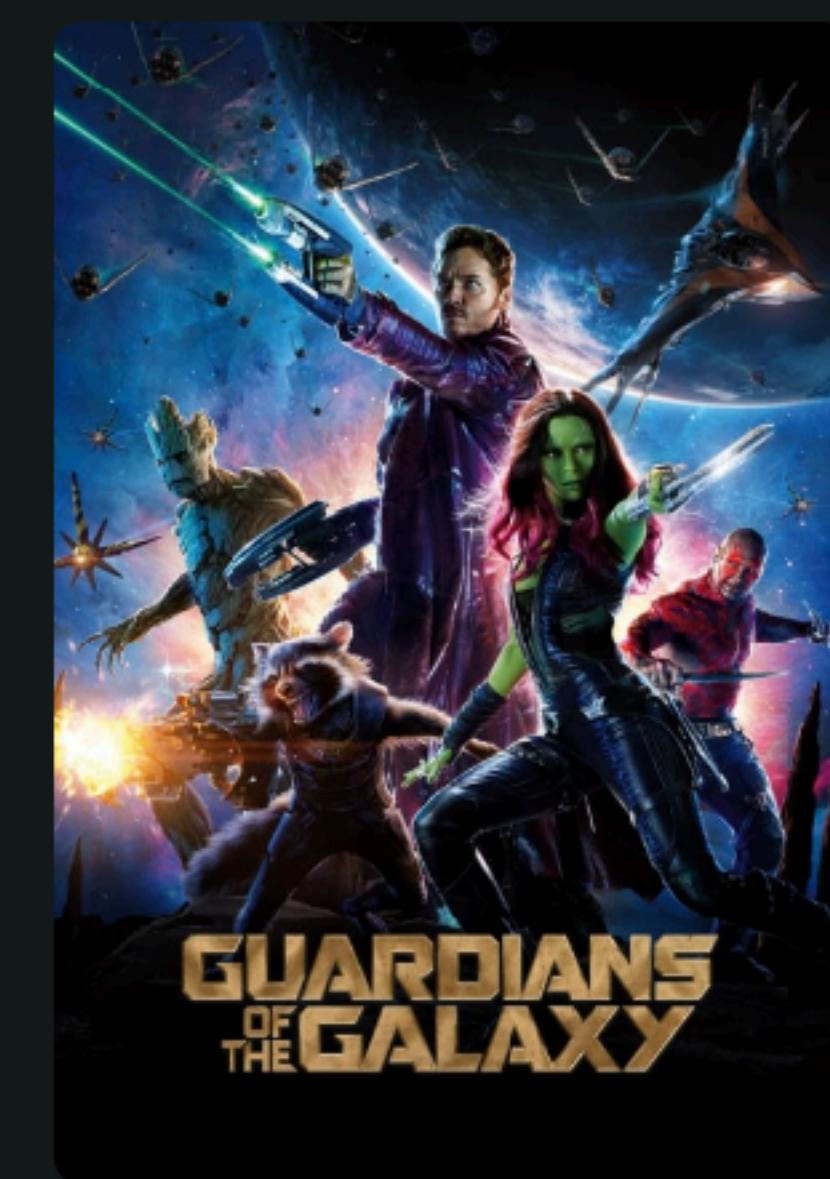
Minions (2015)



Interstellar (2014)



Deadpool (2016)



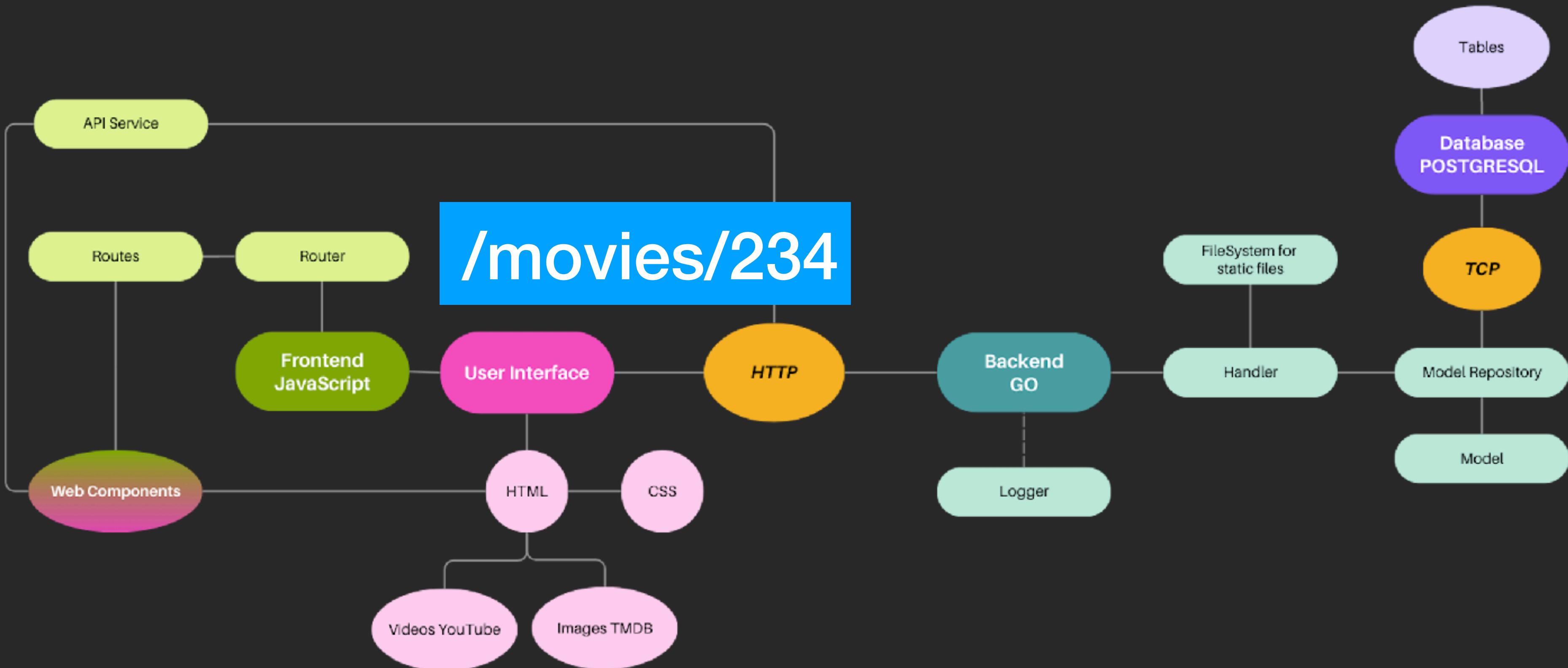
Guardians of the Galaxy (2014)



Mad M

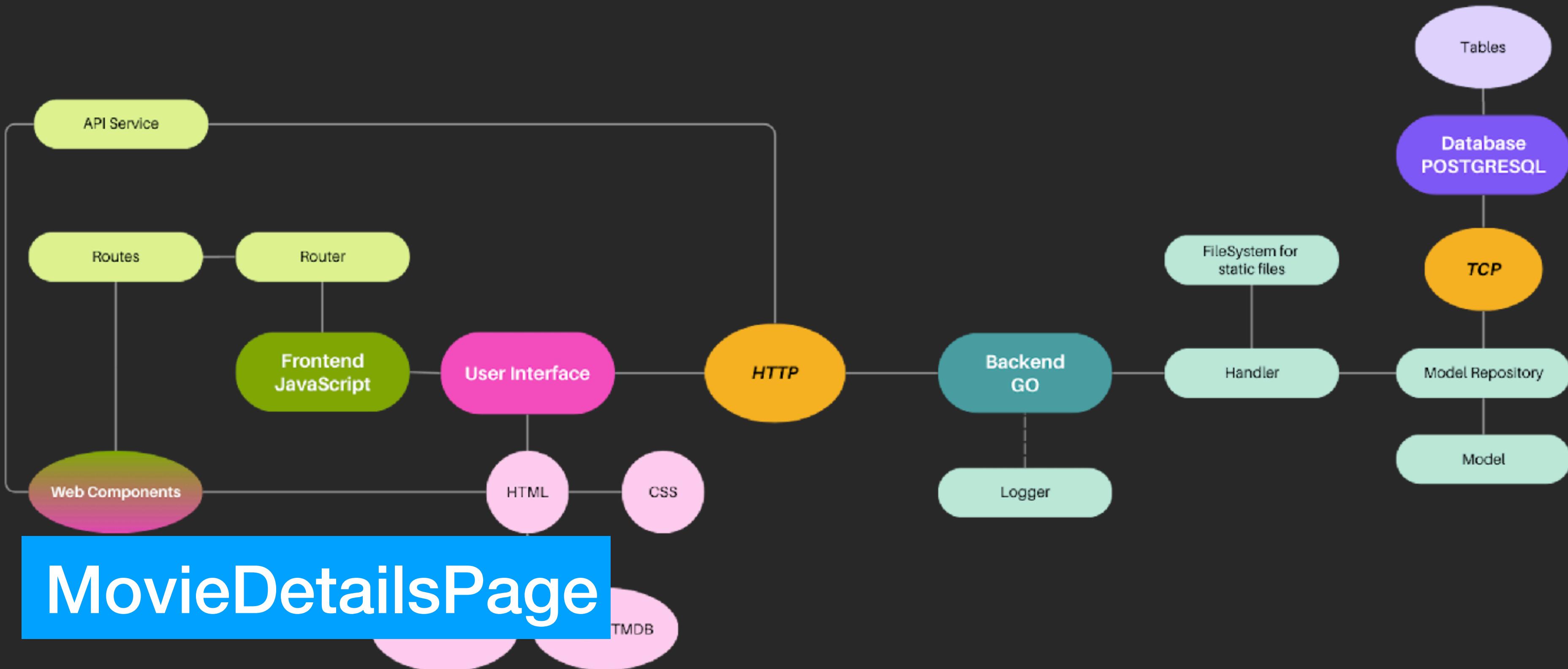
Architecture

Main aspects



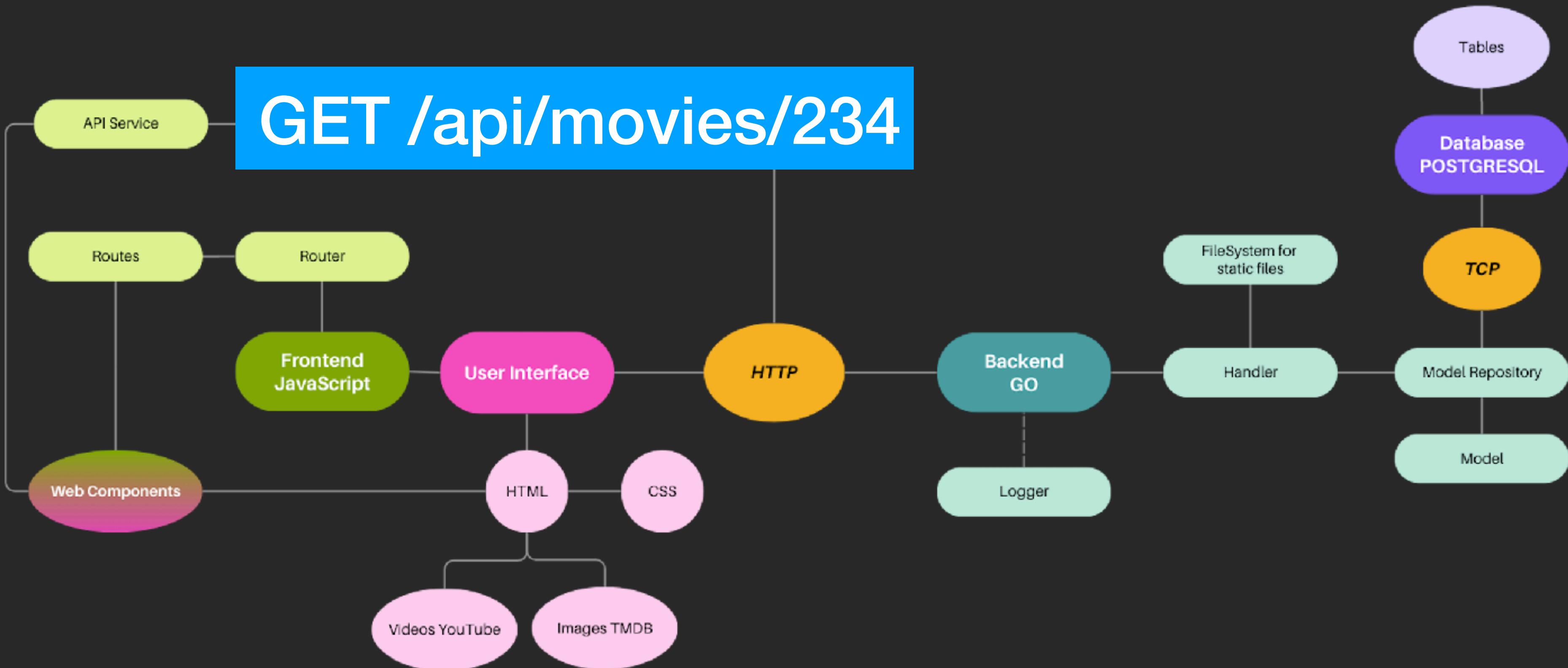
Architecture

Main aspects



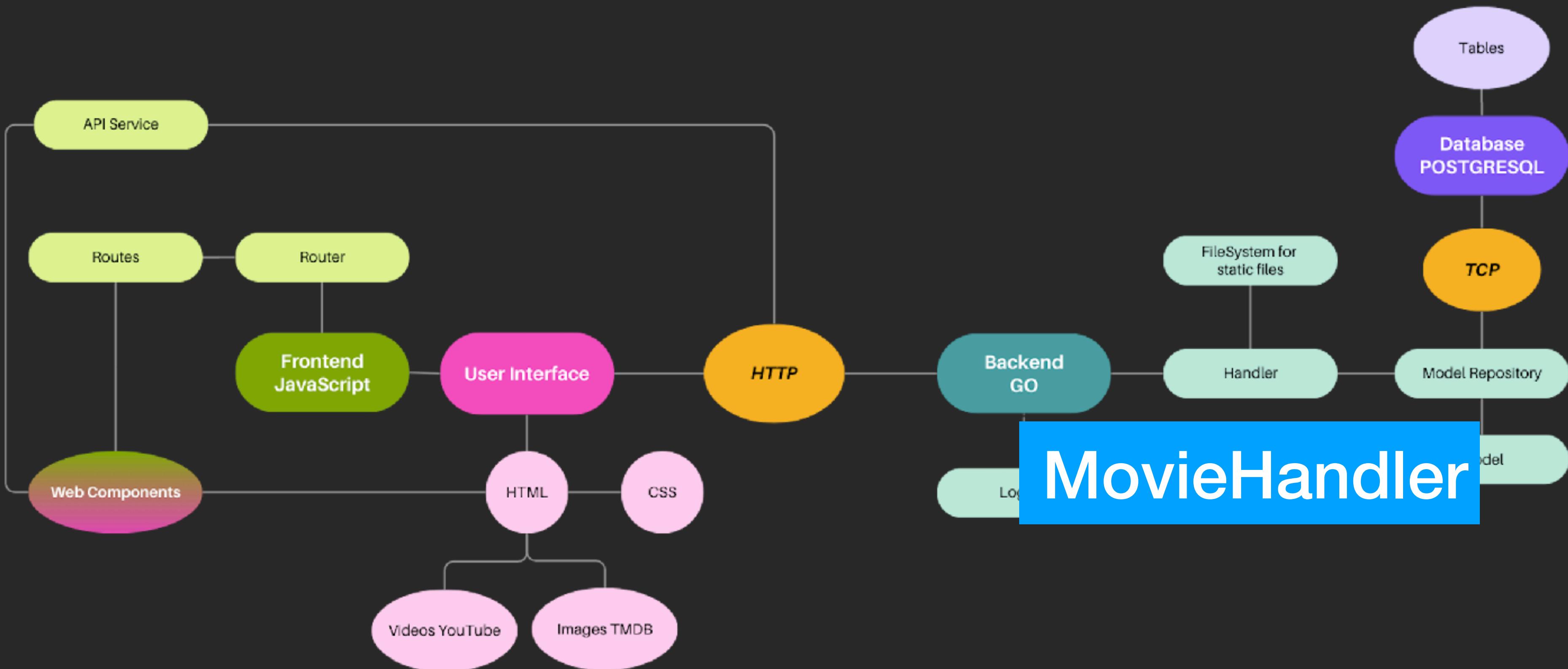
Architecture

Main aspects



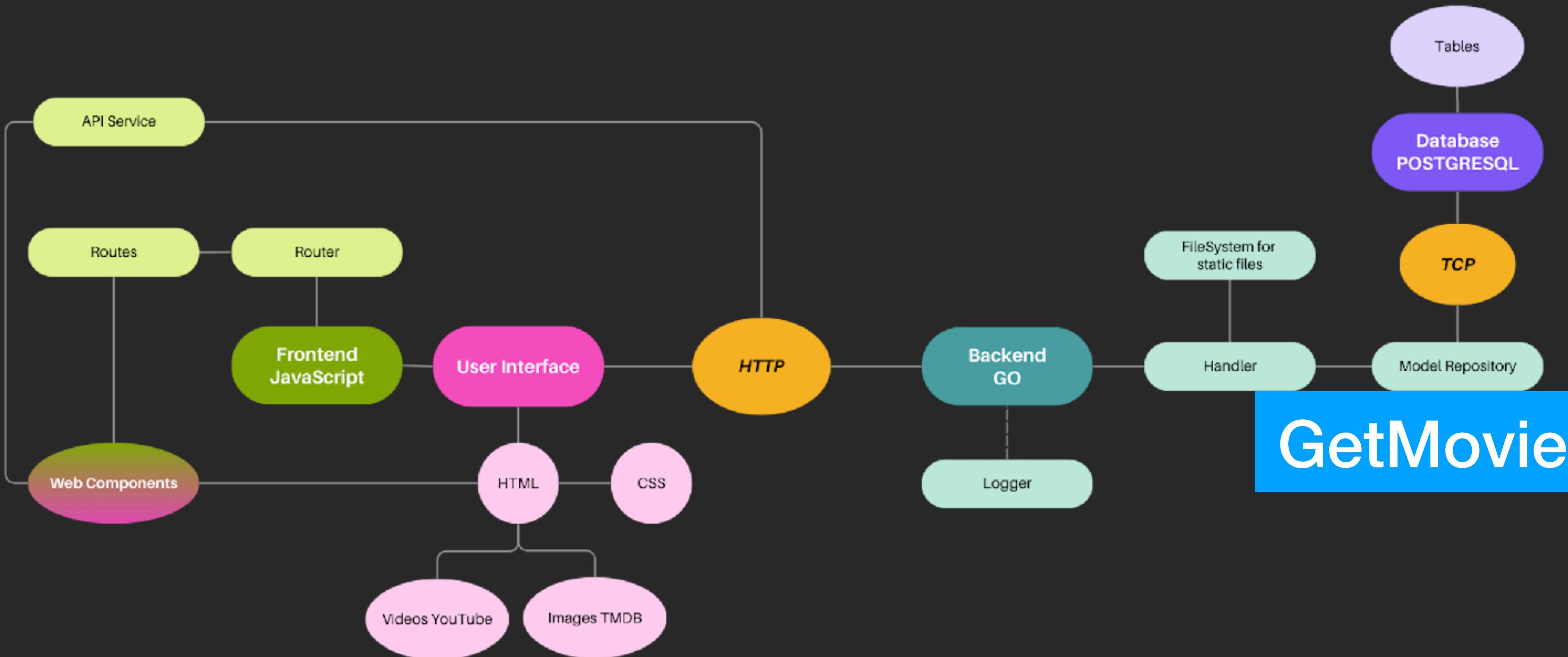
Architecture

Main aspects



Architecture

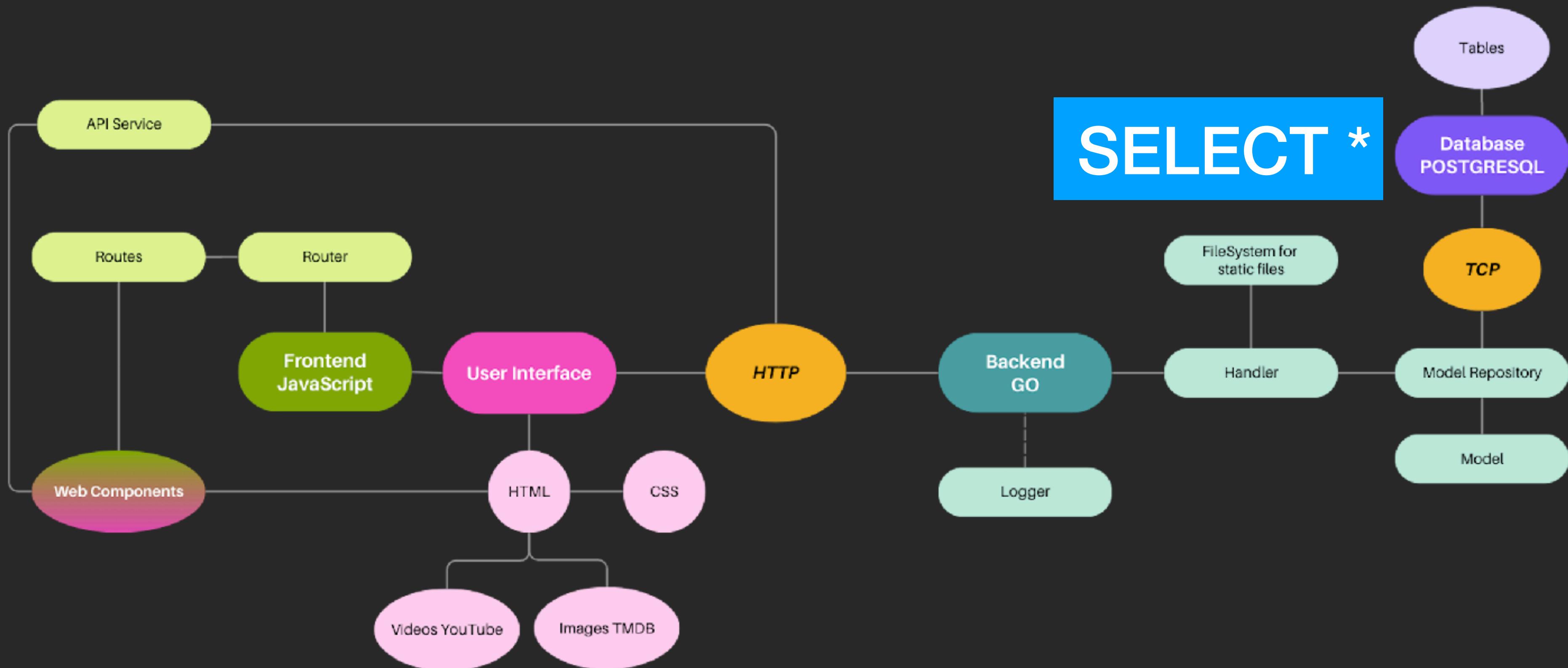
Main aspects



GetMovie

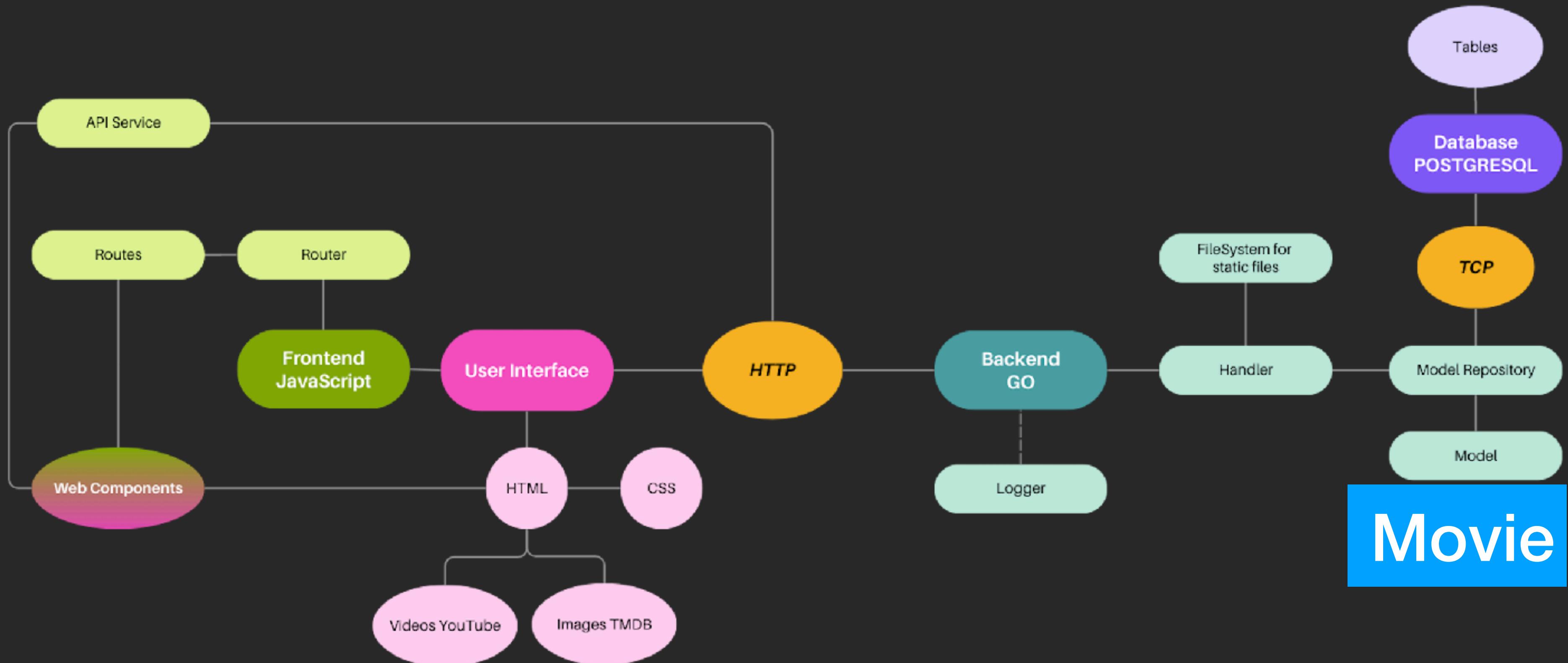
Architecture

Main aspects



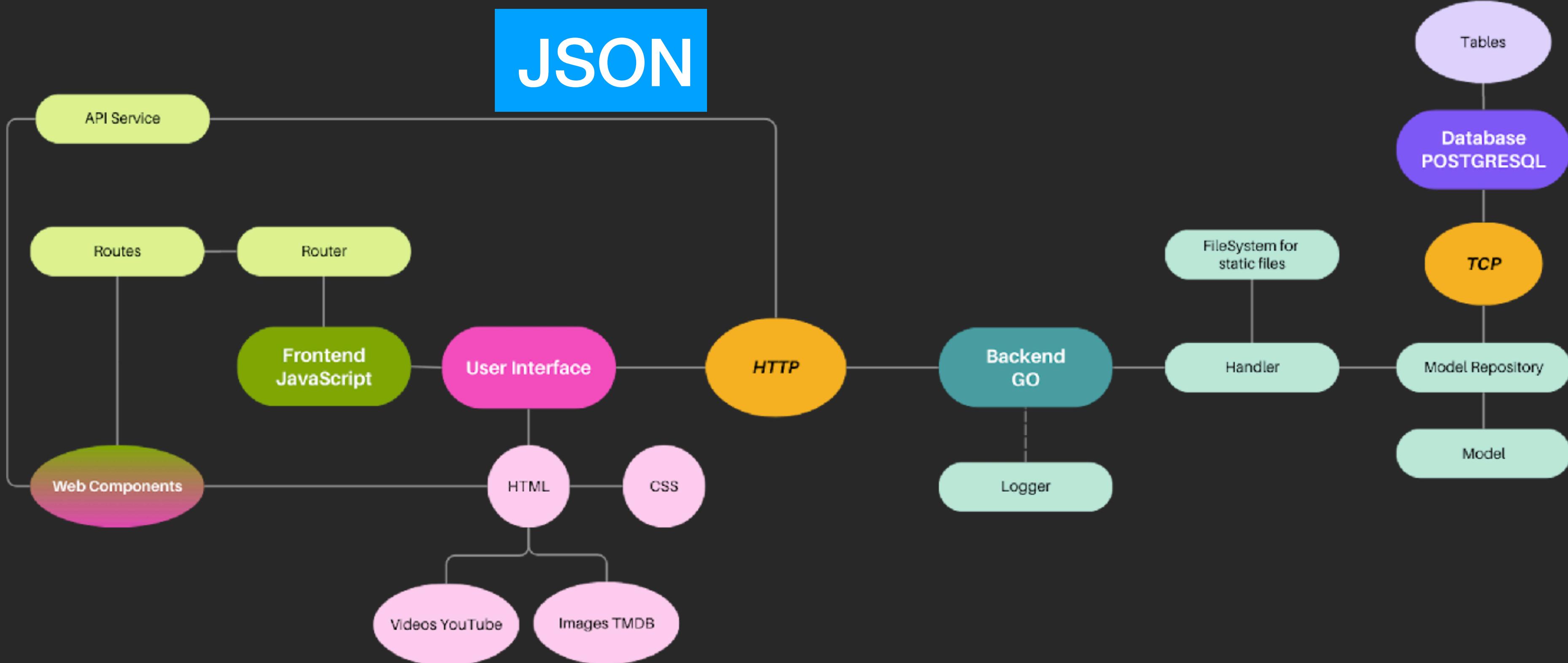
Architecture

Main aspects



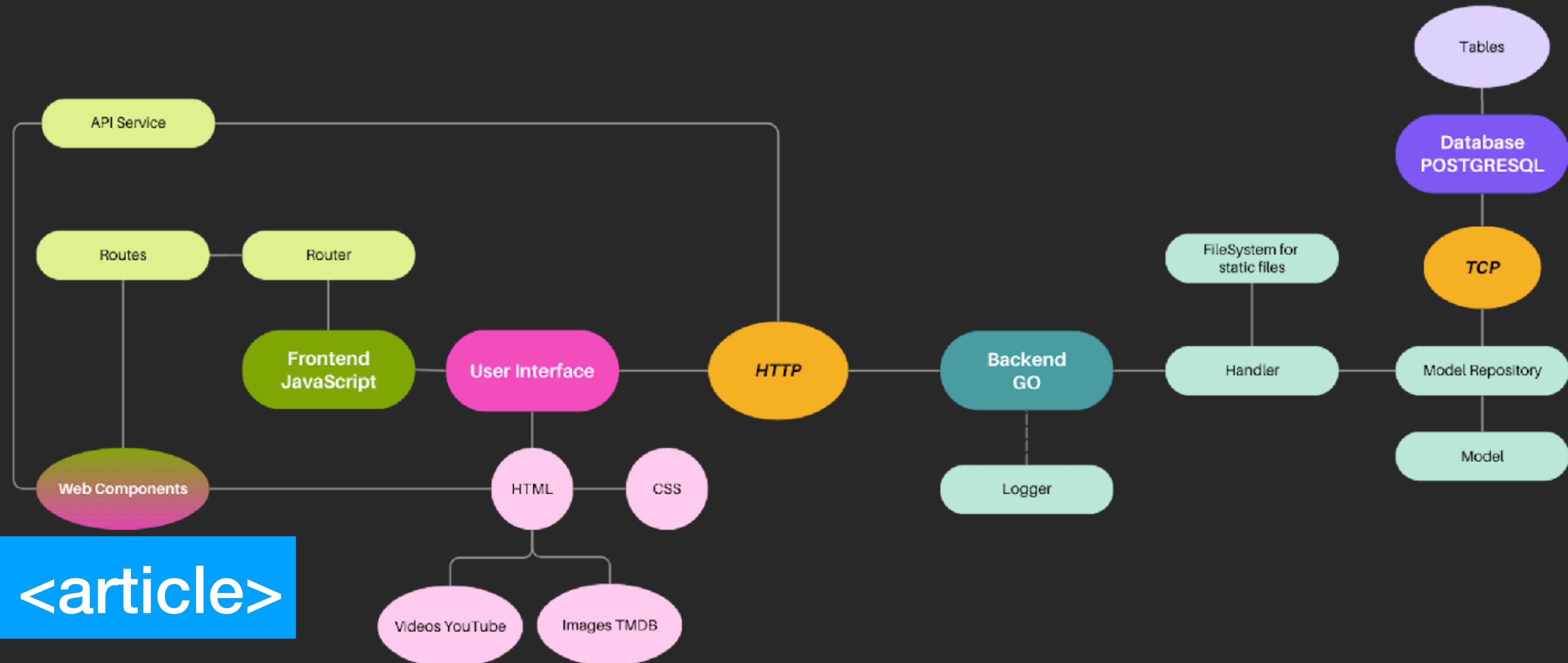
Architecture

Main aspects



Architecture

Main aspects



Deadpool

Witness the beginning of a happy ending

A screenshot from the official trailer for Deadpool 2. It shows the character in his suit, holding a sword and looking determined. The title "DEADPOOL" is at the top, followed by "FEBRUARY 12". A YouTube play button icon is overlaid on the video frame. To the right, there are "Watch later" and "Share" buttons. At the bottom left, it says "Watch on YouTube".

Deadpool | Official Trailer 2 [HD] | 20th Century FOX
FEBRUARY 12

Watch later Share

Watch on YouTube

Release Date
2016

Score
7.4 / 10

Original language
en

Add to Favorites

Add to Watchlist

Action

Comedy

Adventure

Deadpool tells the origin story of former Special Forces operative turned mercenary Wade Wilson, who after being subjected to a rogue experiment that leaves him with accelerated healing powers, adopts the alter ego Deadpool. Armed with his new abilities and a dark, twisted sense of humor, Deadpool hunts down the man who nearly destroyed his life.



Backend

Initial Requirements

- Go 2.20+
- Air github.com/air-verse/air
- Libraries:
 - GoDotEnv github.com/joho/godotenv
 - PG github.com/lib/pq

Lab time

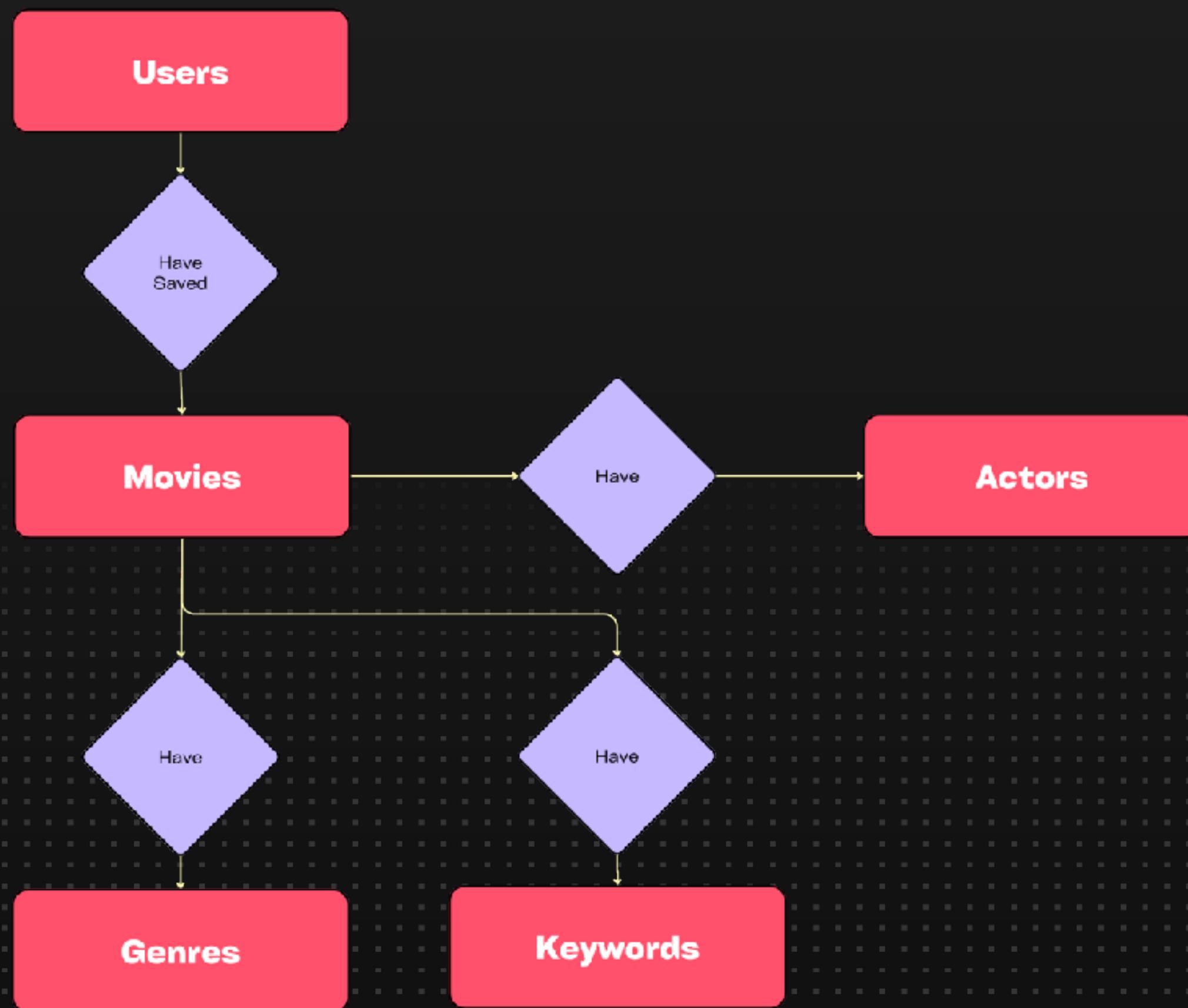


Database Schema

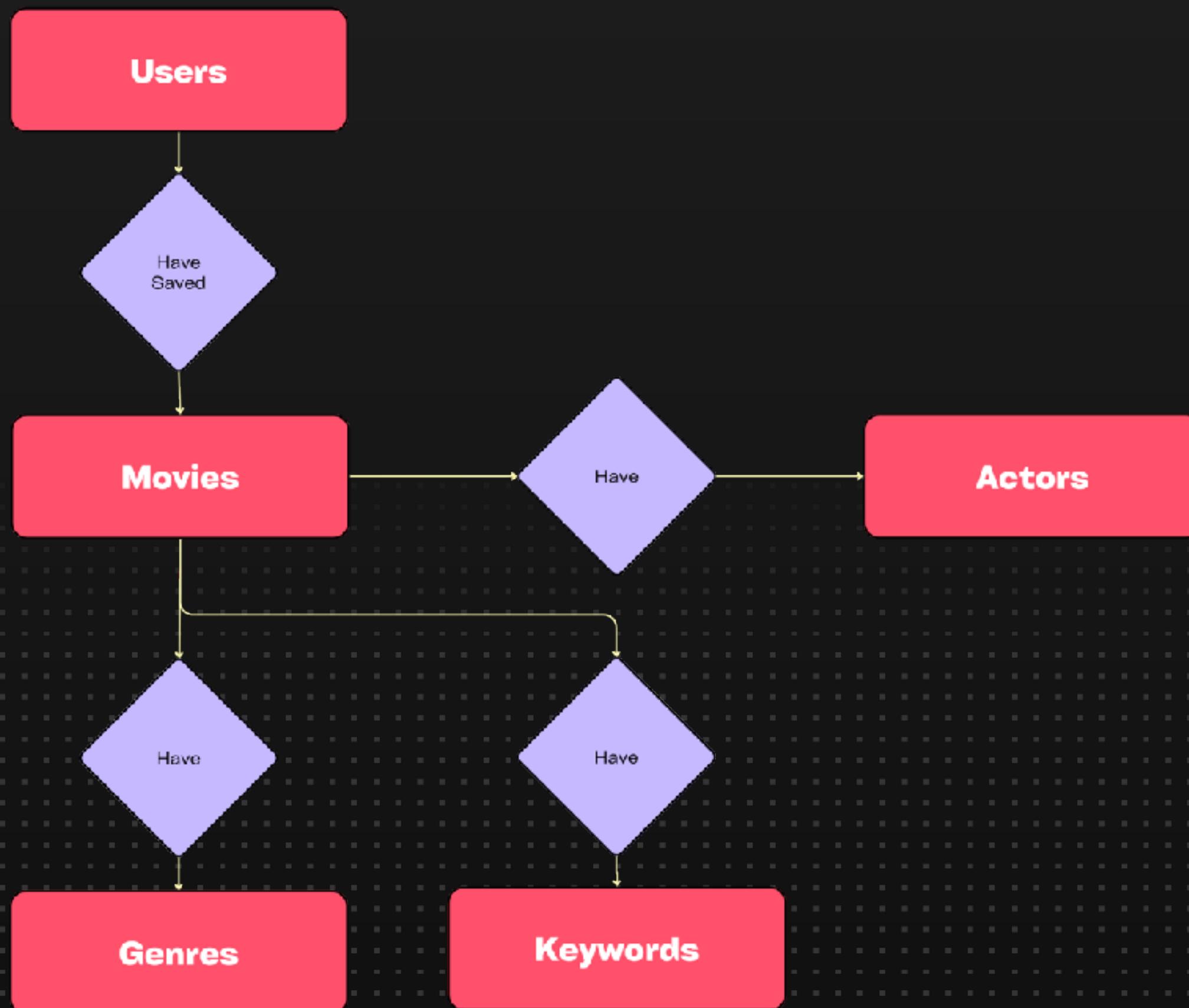
Models

- Movie
- Genre
- Actor
- User

Entity-Relation Diagram



Entity-Relation Diagram



Tables

- users
- movies
- genres
- actors
- user_movies
- movie_cast
- movie_genres
- movie_keywords

Lab time



Frontend

Lab time



Web Components

Definition

Web Component



A modular, reusable building block for web development that encapsulates a set of related functionality and user interface elements.

Definition

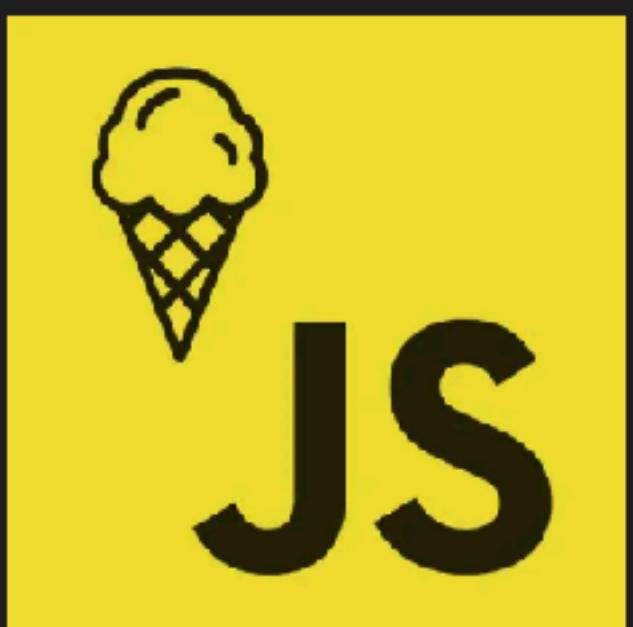
Web Component



In short, your own custom HTML tag element.

Web Components

- Compatible with every browser
- It's actually a set of standards:
 - **Custom Elements**
 - **HTML Templates**
 - **Shadow DOM**
 - **Declarative Shadow DOM**
- It's similar to the idea of components on most of the libraries for JavaScript
- We have freedom of choice on how to define them and use them

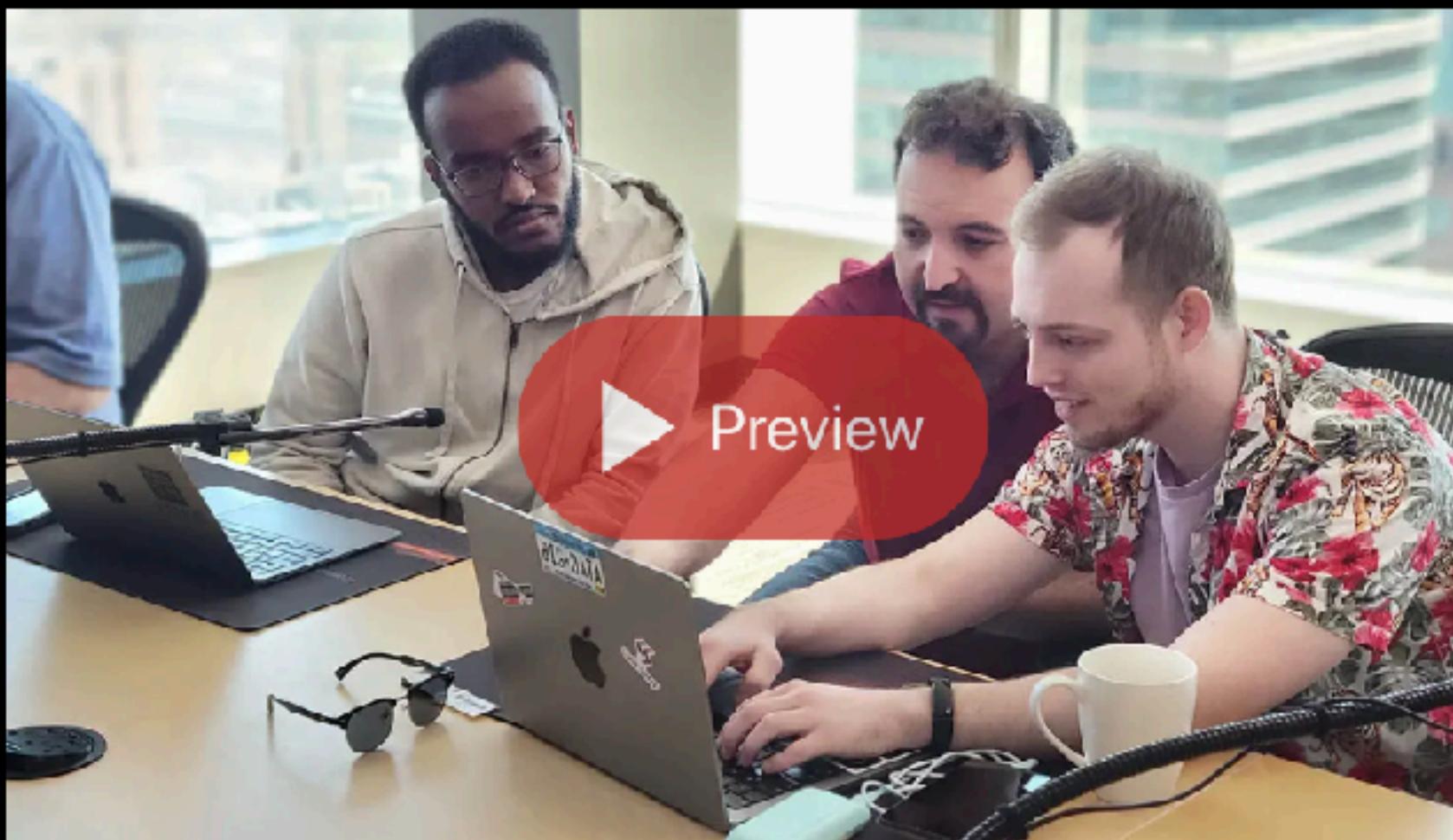


Vanilla JS: You Might Not Need a Framework

**Maximiliano Firtman**

Independent Consultant

5 hours, 52 minutes



Course Description

Gain speed and simplicity in writing web apps by taking a "vanilla" approach! See how you can build rich web apps and websites without depending on libraries or frameworks, using only the core JavaScript language. You'll learn fundamental concepts like the DOM API, finding and modifying elements in the DOM, and event handling. Build a real web app, "Coffee Masters," to demonstrate advanced topics like SPA routing, web components and shadow DOM, and reactive programming with proxies.

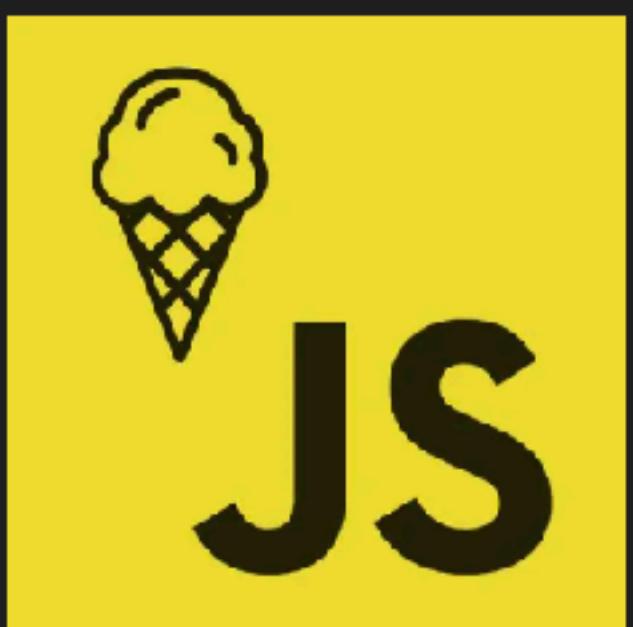
This course and others like it are available as part of our Frontend Masters video subscription.

[Join Now](#)[or Start a Free 7-Day Trial ▶](#)

Lab time



Client-side Routing

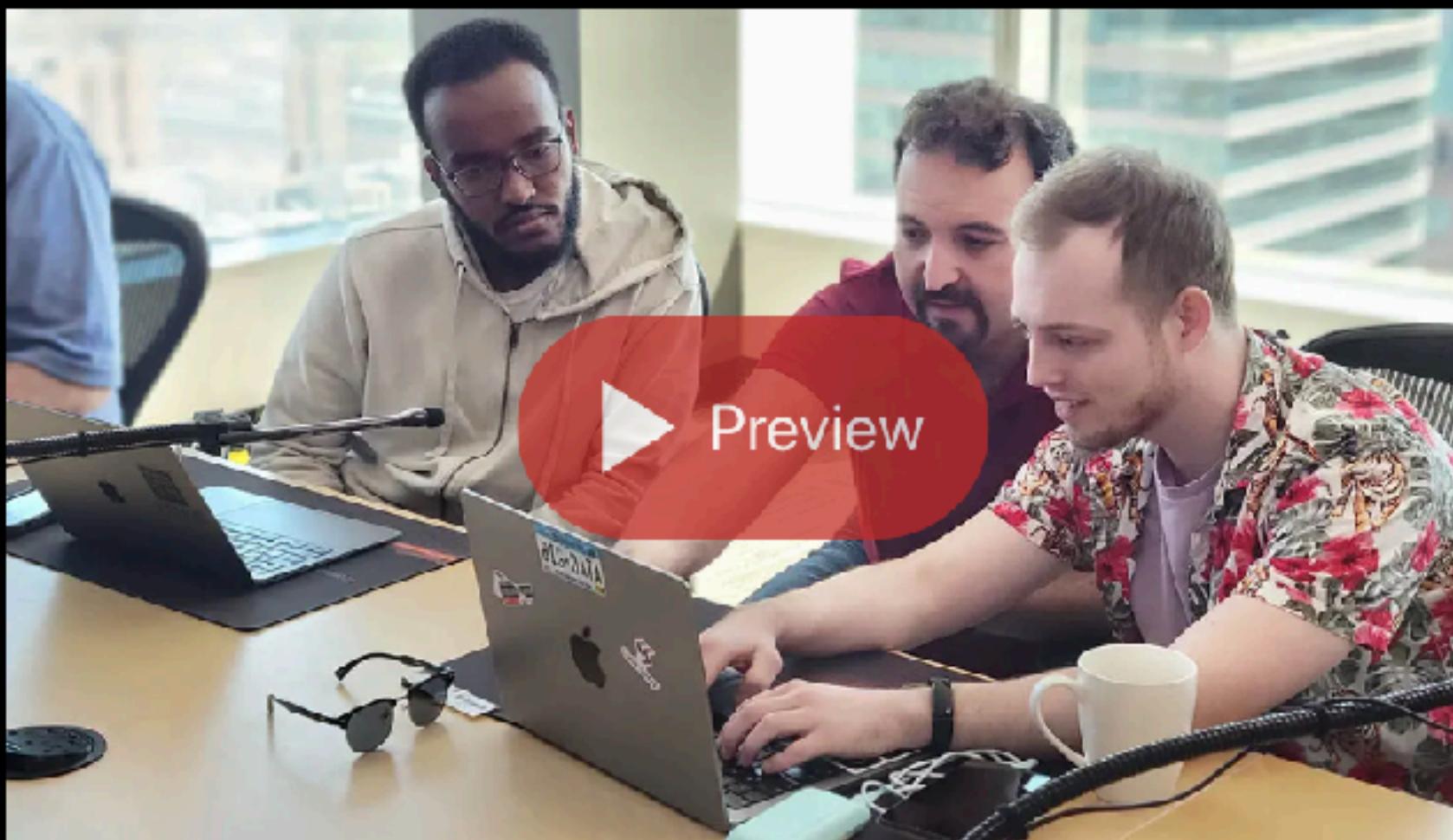


Vanilla JS: You Might Not Need a Framework

**Maximiliano Firtman**

Independent Consultant

5 hours, 52 minutes



Course Description

Gain speed and simplicity in writing web apps by taking a "vanilla" approach! See how you can build rich web apps and websites without depending on libraries or frameworks, using only the core JavaScript language. You'll learn fundamental concepts like the DOM API, finding and modifying elements in the DOM, and event handling. Build a real web app, "Coffee Masters," to demonstrate advanced topics like SPA routing, web components and shadow DOM, and reactive programming with proxies.

This course and others like it are available as part of our Frontend Masters video subscription.

[Join Now](#)[or Start a Free 7-Day Trial ▶](#)

Lab time



Thanks!

Maximiliano Firtman

firt.dev

X x.com/firt

in linkedin.com/in/firtman

github.com/firtman