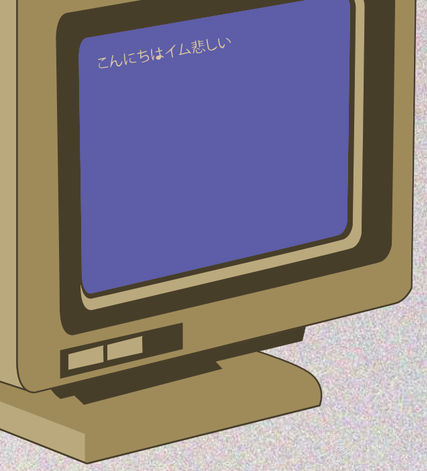
Ing. Olivier Paspuel

INTERACTIVE WEB PAGES

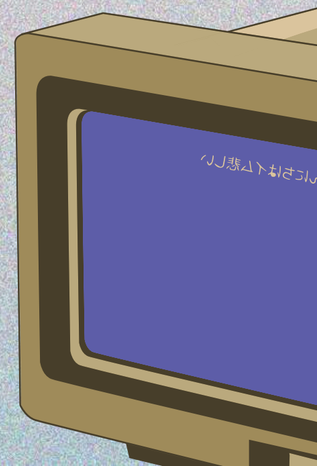
Part 1

PREREQUISITES

- Programming fundamentals
- OOP
- Basic understanding of software development
- Week 0 lecture
- Completion of week 0 psets, knowledge inquiries and classwork



WEB PAGES

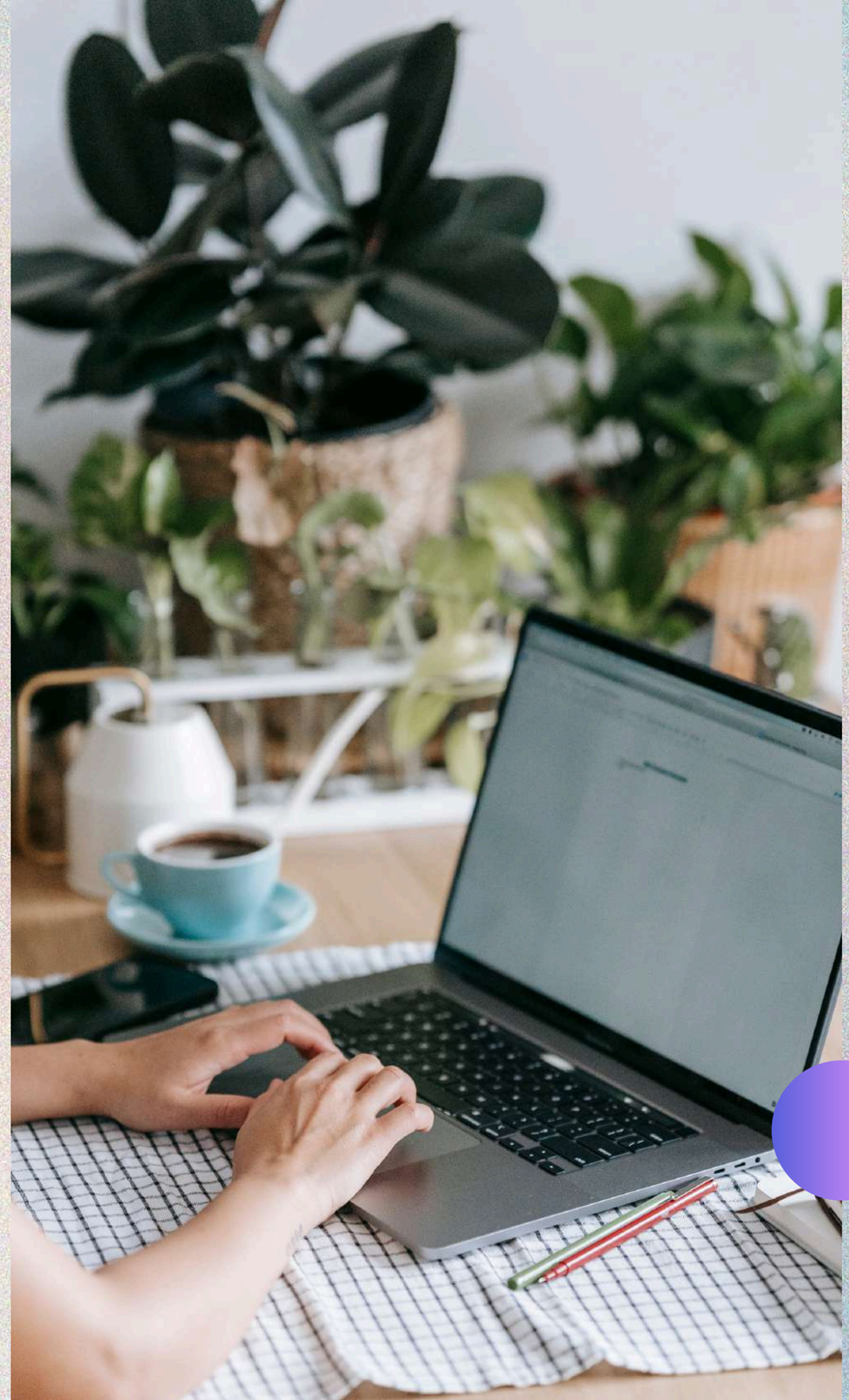


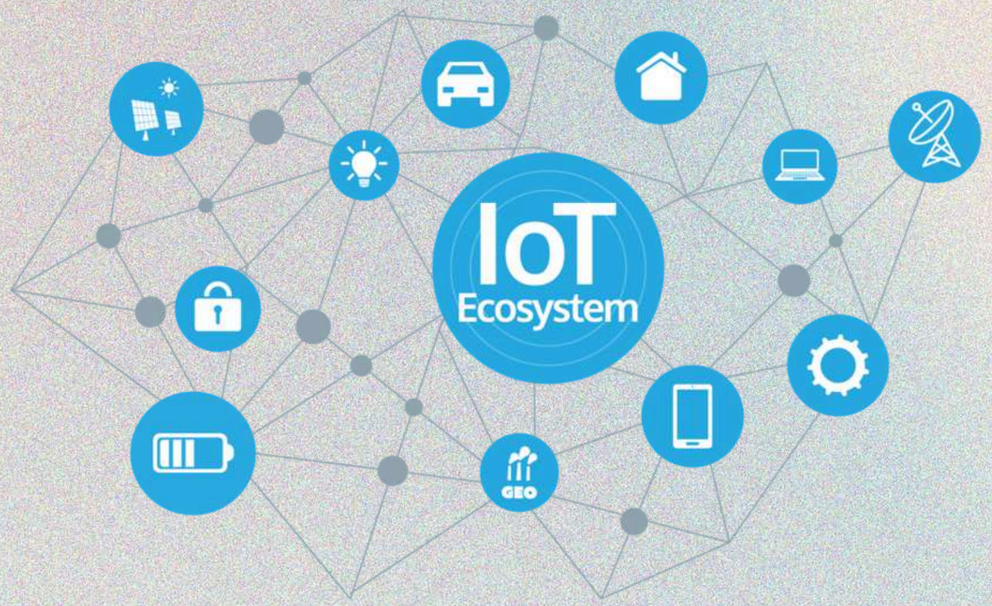
STATIC WEB PAGE

- Content is fixed and remains the same for every visitor.
- Files are usually HTML, CSS, and JavaScript (optional).
- No interaction with databases or external data sources.
- Requires manual updates to change the content.
- Faster load times because content is pre-built.
- Easier and cheaper to host.
- Suitable for small websites or landing pages.

DYNAMIC WEB PAGE

- Content can change based on user interactions or data.
- Often uses server-side technologies (e.g., PHP, Python, Node.js).
- Interacts with databases to fetch and display content.
- Automatically updates content based on data or user input.
- Typically requires more server resources, which can affect load times.
- Can offer personalized experiences for different users.
- Suitable for complex applications like social media sites or e-commerce platforms.



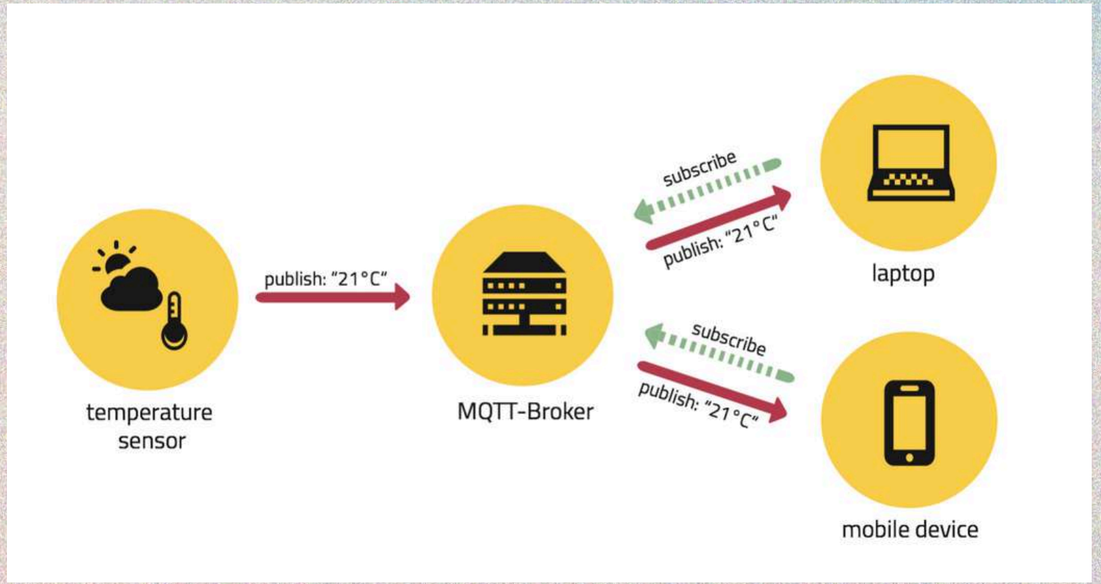


INTERNET OF THINGS (IOT)

- Network of interconnected devices that communicate and exchange data.
- Devices can range from smart appliances, sensors, wearables, and vehicles to industrial machines.
- Relies on embedded systems (hardware + software) to gather and transmit data.
- Often uses wireless communication protocols like Wi-Fi, Bluetooth, Zigbee, or cellular networks.
- Allows for real-time monitoring, control, and automation of devices.
- Data is usually sent to cloud platforms for storage, analysis, and decision-making.
- Widely used in smart homes, healthcare, agriculture, and industrial automation.
- Can improve efficiency, convenience, and safety, but raises concerns around security and privacy.

IOT EXAMPLE

- A good example of the Internet of Things (IoT) is a smart thermostat like Google Nest:
- Smart Thermostat: This device is connected to the internet and allows users to control their home's temperature remotely via a smartphone app or a web interface.
- It uses sensors to detect when people are at home, learning user habits to adjust the temperature automatically.
- It communicates with other smart devices, like lights or smart blinds, to optimize energy consumption.
- The data collected (e.g., temperature preferences, energy usage) is stored in the cloud, allowing users to track energy savings.
- It can even adjust the heating or cooling based on weather forecasts from online services.



JAVASCRIPT

- A high-level, dynamic programming language primarily used for web development.
- Runs in the browser and allows developers to create interactive web pages.
- Supports both client-side (in the browser) and server-side (e.g., with Node.js) scripting.
- Can manipulate the Document Object Model (DOM) to dynamically update HTML and CSS.
- Supports object-oriented, functional, and imperative programming paradigms.
- Commonly used in frameworks and libraries like React, Angular, and Vue.
- Allows asynchronous operations with features like callbacks, promises, and async/await.
- Widely supported by all major web browsers.

JS Frameworks

REACT

- Developed by Facebook.
- Not a framework
- Focuses on building user interfaces with a component-based architecture.
- Uses a virtual DOM for efficient rendering.
- Often used for single-page applications (SPAs)

ANGULAR

- Developed by Google.
- A full-featured framework for building dynamic web applications.
- Uses two-way data binding and dependency injection.
- Ideal for large-scale applications.

VUE.JS

- Lightweight and easy to integrate with existing projects.
- Focuses on the view layer with a reactive data-binding system.
- Good for building user interfaces and single-page applications.

CSS Frameworks

BOOTSTRAP

- One of the most popular CSS frameworks.
- Provides a grid system, ready-made components, and utilities.
- Focuses on responsive design and mobile-first development.

TAILWINDCSS

- Utility-first CSS framework that lets you build custom designs by applying pre-defined classes directly in HTML.
- Highly customizable and avoids predefined components, giving you more control over the design.

MUI

- Implements Google's Material Design guidelines, providing a consistent look and feel.
- Offers a wide range of pre-built components (e.g., buttons, cards, modals) that are customizable and responsive.
- Supports theming and styling to match the design of your application.
- Includes tools for layout, spacing, and typography.
- Provides accessibility features to ensure components are usable for all users.

JS Backend

NODE.JS

- Allows JavaScript to be used for server-side programming.
- Non-blocking, event-driven architecture for scalable network applications.
- Commonly used for building APIs and real-time applications.

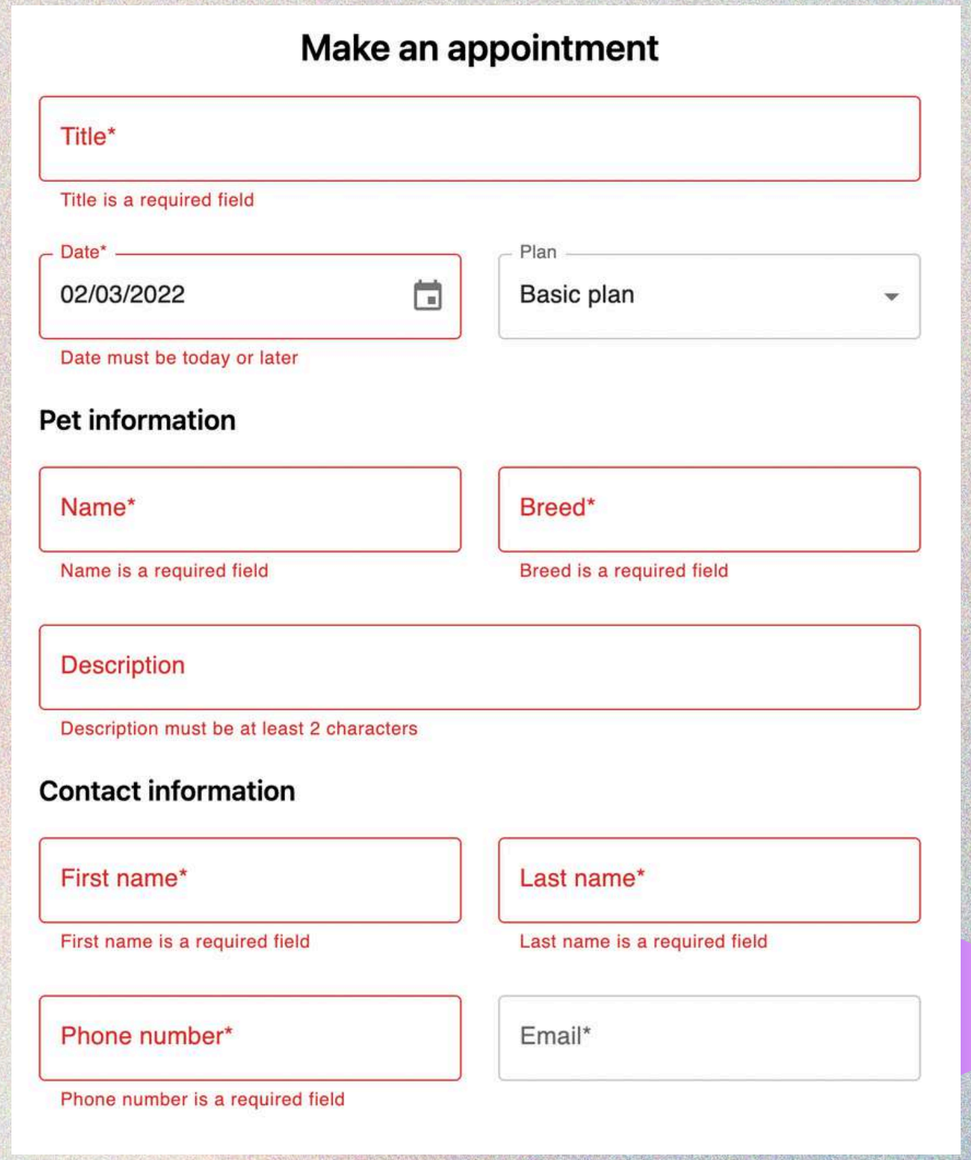
EXPRESS

- A minimal web framework for Node.js.
- Simplifies building web applications and APIs.
- Provides middleware and routing mechanisms.

Application layer

WEB PAGES FORMS

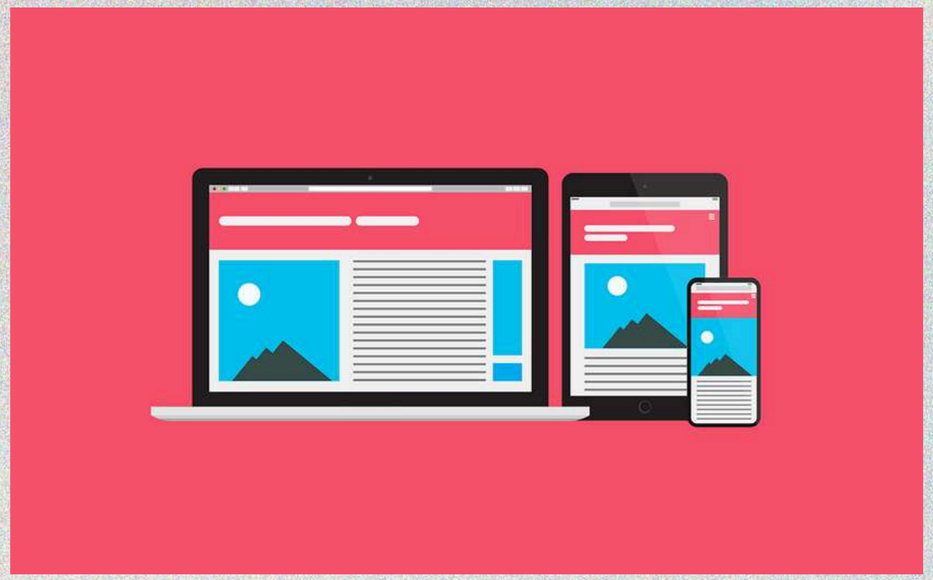
- HTML form elements
- JavaScript interaction
- Form validation
- Dynamic form handling
- Data submission
- Accessibility considerations
- Best practices
- POST, UPDATE



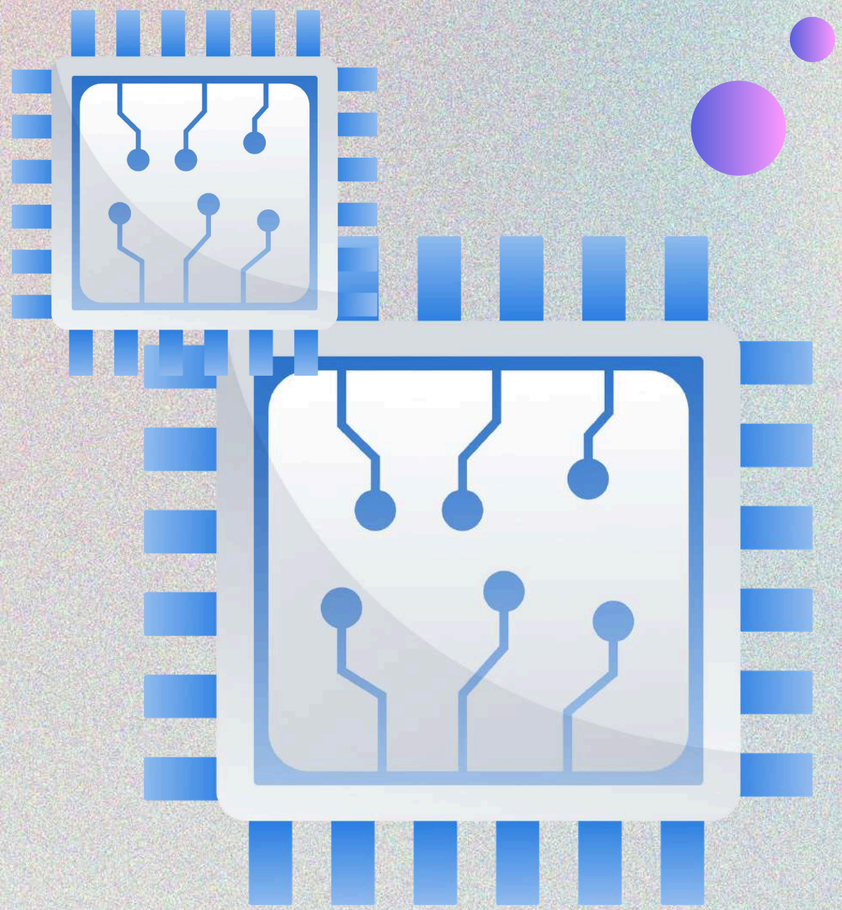
INTERACTIVE WEB PAGES

- EVENT HANDLING
- DOM MANIPULATION
- ATTRIBUTES AND NODES

RESPONSIVE DESIGN



- **MOBILE-FIRST DESIGN**
  The design is created starting from the smallest screen size and enhanced for larger screens, ensuring better performance and usability for mobile users.
- **BREAKPOINTS ‘N’ MEDIA QUERIES**
  Specific screen widths (e.g., 768px for tablets, 1024px for desktops) where the layout changes to provide an optimal viewing experience. CSS rules allow this kind of behavior
- **FLEXIBLE LAYOUTS**
  The website uses a grid system that adjusts depending on the device’s screen size.



CONTENT OPTIMIZATION

- Multimedia optimization
- Mobile-friendly
- Server Side Rendering (SSR)
- Semantic HTML
- Reducing backend use and requests.

THANK YOU!