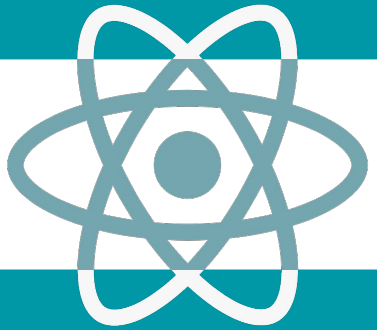
COESIONE ITALIA 21-27 PIEMONTE
Cofinanziato dall'Unione europea
REGIONE PIEMONTE



Programmazione Web

React

Davide Mantovani

synesthesia
the digital experience company

Intervento realizzato da
ITS TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

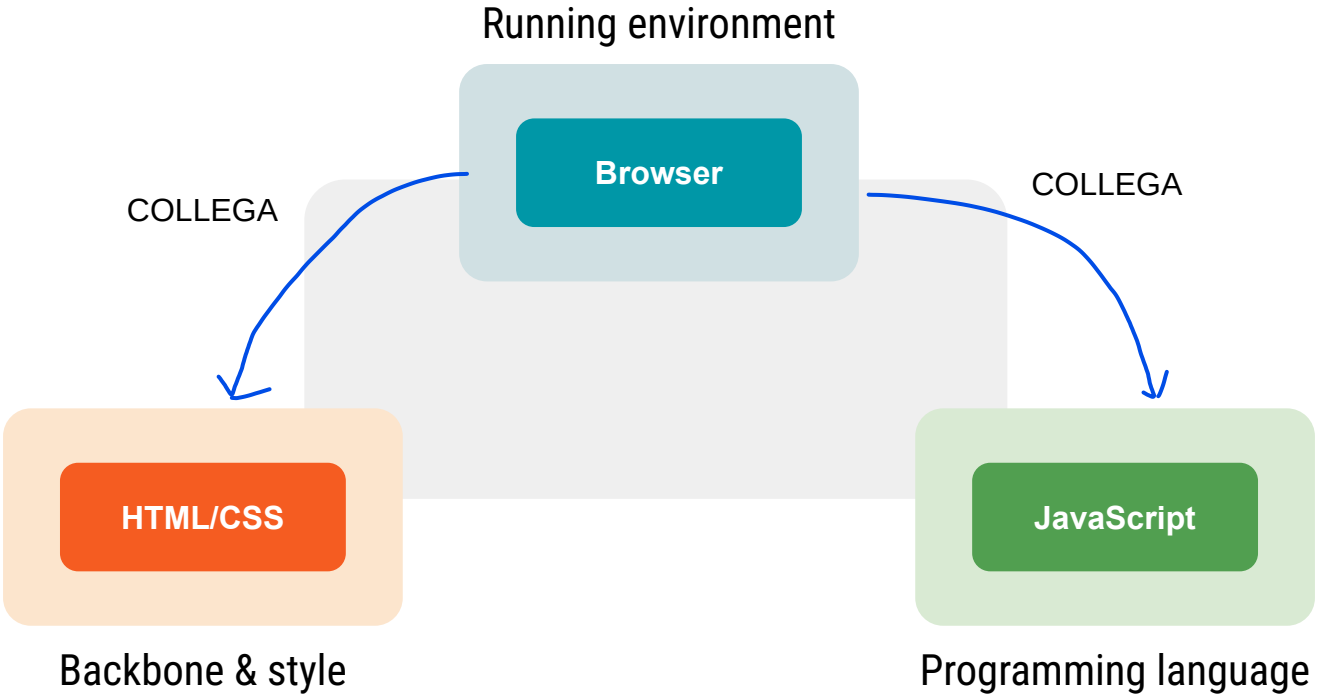
Queste slide introducono i principali "mattoni" del frontend e il ruolo del client: Mattoni del frontend:

- Browser: l'ambiente in cui l'app gira.
- HTML/CSS: definisce la struttura e lo stile dell'interfaccia.
- JavaScript: fornisce la logica e il comportamento dinamico.

Being a Client

Frontend building blocks

Il browser funge da ambiente di esecuzione per le tecnologie frontend, combinando questi elementi per costruire un'interfaccia utente.



Responsibilities

Il client ha compiti cruciali per garantire un'applicazione reattiva e sicura:

Richieste al server:

- The client is responsible for **sending requests to the server** to retrieve information or perform operations.

Gestione dell'interfaccia utente:

- It must **manage the user interface** effectively to ensure a good user experience.

Gestione degli errori e connessioni:

- It is important to **handle errors and connections** clearly and effectively to maintain communication with the server.

Validazione iniziale dei dati:

- It is responsible for an **initial validation** of the user's input.

Vulnerabilities

evidenziamo i RISCHI principali del frontend, illustrando come evitarli:

- **Insecure coding practices in client-side scripts** can lead to security risks such as cross-site scripting (XSS) or injection attacks.*
- **Weaknesses in browser security** can expose users to risks such as unauthorized access to sensitive data or malicious code execution (eg. bugs or local access to browser storage).
- Heavily **dependent on infrastructure security status** (eg. SSL) and **backend security implementations** (eg. OAuth2).

APPUNTI + INFO

Vulnerabilities

- **Insecure coding practices in client-side scripts** can lead to security risks such as cross-site scripting (XSS) or injection attacks.*
- Wea
such
exec
- Heav
SSL)

***Example of Cross-Site Scripting (XSS):**

Suppose a web application allows users to input comments that are displayed on a webpage without proper input validation.

An attacker could craft a malicious script and inject it into the comment field.

When other users view the comments section, the malicious script executes in their browsers, potentially stealing sensitive information or performing unauthorized actions on their behalf.

Vulnerabilities

- **Insecure coding practices in client-side scripts** can lead to security risks such as cross-site scripting (XSS) or injection attacks.*
- **Wea**
  such
  exec
- Heav
  SSL)

***Example of Injection Attack (both FE and BE):**

If a web application fails to sanitize user input properly, an attacker could exploit this vulnerability to inject malicious code into the backend.

For instance, in a login form where user inputs are not properly validated, an attacker could input SQL commands in the username field to manipulate the database backend.

This could lead to data breaches, unauthorized access to sensitive information, or even the deletion of critical data.