

# Progetto Fondamenti dei Sistemi Web:

## Emscripten

### Descrizione

L'argomento che abbiamo deciso di scegliere è **WebAssembly**, tecnologia sempre più utilizzata specialmente per quanto riguarda applicazioni Web che richiedono quantità di calcolo non banali, quali **giochi**, **applicazioni grafiche** e di **elaborazione dati** (machine learning, AI...).

Più specificatamente, abbiamo illustrato **Emscripten**, un toolkit in grado di compilare codice **C/C++** in WebAssembly. In altre parole, fa in modo che questo codice sia eseguibile dal browser.

### Tecnologie Utilizzate

- Vue.js
- Vue Router
- HTML5, CSS
- Bootstrap
- Emscripten
- Axios

## Progettazione

Inizialmente abbiamo pensato a un design stile “vecchia scuola”, ma successivamente, tenendo in considerazione l’accessibilità e ancora più la responsiveness del sito, abbiamo optato per utilizzare **Bootstrap**. Questa libreria ci ha permesso di ottenere un design minimale, semplice (ma moderno) e non invasivo, oltre che adattabile al display con cui viene utilizzato, desktop o mobile che sia. Inoltre, il design del sito è risultato più moderno, e più consono all’argomento da esso trattato.

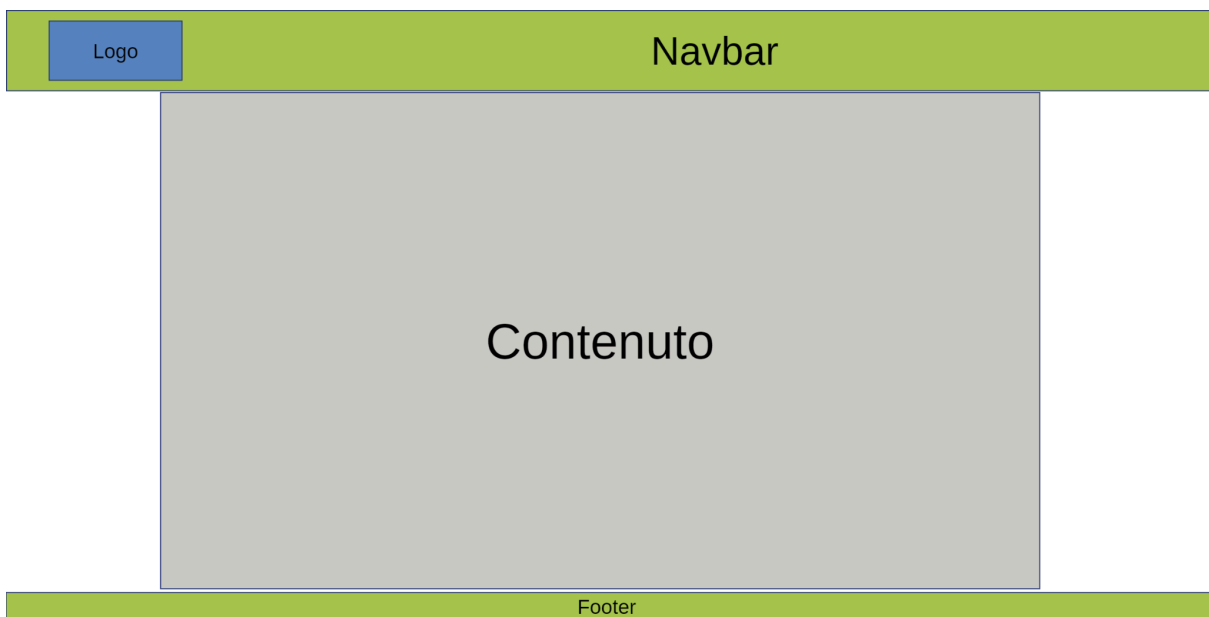
Con Bootstrap abbiamo anche avuto modo di implementare comodamente la parte “front-end” della nostra applicazione WebAssembly, facendola risultare comoda e gradevole alla vista.

Nonostante abbiamo tenuto in considerazione **programmatici grafici/game developers** come i **principali target** del nostro sito (da come è possibile notare dai numerosi esempi/screenshot raffiguranti giochi e motori grafici), abbiamo deciso di implementare come esempio una semplice applicazione testuale, ovvero una calcolatrice. Ciò ci ha permesso di dare più importanza al fattore accessibilità, in quanto utilizzabile da ogni tipologia di utente, e su ogni dispositivo.

# Design & Mockup



Design originale del nostro sito, con CSS custom (e poco responsive)



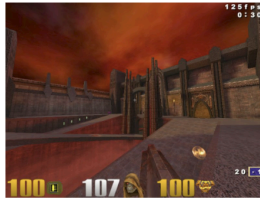
Mockup design complessivo

## What is Emscripten?

Emscripten is a compiler that converts C and C++ code into WebAssembly (WASM) or JavaScript, allowing developers to run native code in web browsers or Node.js environments. Here's a brief history of Emscripten and an overview of what it does.

### Early days (2010-2011)

Emscripten was created by Alon Zakai, a Mozilla engineer, in 2010. At the time, Zakai was working on a project to port the Quake III Arena game engine to run in a web browser. He realized that compiling C code to JavaScript would be a more efficient approach than rewriting the entire engine in JavaScript. Zakai's initial implementation, called "Emscripten," was a simple C-to-JavaScript compiler.



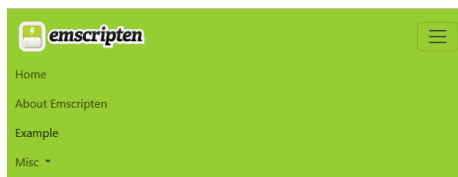
A screenshot of id Software's game, Quake 3 Arena

### Development and growth (2011-2013)

In 2011, Emscripten was open-sourced and gained popularity as a tool for porting native games and applications to the web. The compiler was initially based on the LLVM compiler infrastructure project, which provided a robust framework for building compilers. Emscripten's development accelerated, and it began to support more C and C++ features, including dynamic linking, exceptions, and RTTI (Run-Time Type Information).

### WebAssembly (2015)

Design finale del nostro sito, utilizzante Bootstrap.  
Lo stile risulta meno "pesante" alla vista, ed è responsive



## Calculator

### A basic calculator

Number 1      Number 2

Number 1      Number 2

Add Subtract Multiply Divide

Result

Result

This is a simple calculator which uses WebAssembly to work. Its core is written in C, and after being compiled to WebAssembly using Emscripten SDK it can be interfaced to a web-page (thanks to a JavaScript wrapper, generated by Emscripten during compiling process).

Here is the C source code:

```
#include <emscripten.h>

int EMScripten_KEEPALIVE add(int a, int b) {
    return a + b;
}

int EMScripten_KEEPALIVE subtract(int a, int b) {
    return a - b;
}
```

Navbar dinamica in base al display/dimensione del display, così da offrire una navigazione comoda sia su mobile che su desktop (per esempio nel caso ci siano più finestre aperte)