

Tecnologie Web: approcci avanzati

Luca Anselma luca.anselma@unito.it



UNIVERSITÀ
DEGLI STUDI
DI TORINO

Regole d'esame e requisiti del progetto

Esame

Due «momenti»:

1) Consegna del progetto di laboratorio

- entro la scadenza delle iscrizioni all'appello su MyUnito

2) Discussione del progetto di laboratorio

- **Attenzione!** I membri di un gruppo devono sostenere la discussione insieme
- **Attenzione!** Prenotatevi sia su MyUnito che su Moodle

Discussione

- **Discussione orale del progetto di laboratorio**
- Gli studenti e le studentesse devono avere il progetto funzionante sul proprio PC + il codice sorgente visualizzabile
- La discussione verte **sia sul progetto di laboratorio che sugli argomenti del corso**
- La valutazione terrà conto:
 - delle scelte implementative
 - della padronanza del progetto
 - della capacità di spiegarle correttamente
 - della padronanza degli argomenti del corso
 - della conoscenza dei prerequisiti

Esame

L'obiettivo è

- **capire** gli argomenti di cui si parla nel corso
- **saperli utilizzare e comunicare**, NON imparare a memoria le slide (o gli esempi)

Nella discussione occorre dimostrare **padronanza** del progetto e delle tecniche

Un progetto, pur fatto bene di cui non si sa spiegare nel dettaglio il funzionamento sarà **insufficiente**

Il progetto

- Il progetto consiste nella progettazione e nello sviluppo di un'**applicazione web**
- L'**argomento** è a vostra scelta
- Il progetto deve rispettare i requisiti del progetto (v. dopo)
- Il lavoro può essere svolto anche in **gruppo** (composto da 1, 2 o 3 persone)
- La composizione del gruppo deve essere dichiarata alla consegna del progetto (**scriverlo sia sul sito che nel nome del .zip consegnato**)

Requisiti progetto

Requisiti tecnici del progetto

- **HTML e CSS:** L'applicazione **deve** fare uso di HTML e di CSS
- **PHP/JavaScript:** L'applicazione **deve** fare uso di PHP o di JavaScript come linguaggio di programmazione web server-side (è possibile concordare l'utilizzo di linguaggi server-side diversi)
- **PHP Object-Oriented:** L'utilizzo di PHP Object-Oriented (e quindi la definizione e l'uso di classi) è *opzionale*, ma almeno *un esempio di definizione di una classe* è **caldamente consigliato** [N.B. l'argomento fa comunque parte del programma del corso e può essere argomento di **interrogazione**]
- **Form:** L'applicazione **deve** contenere almeno un esempio significativo di interazione server-side con un modulo (form)
- **Database:** L'applicazione **deve** includere un database e mostrare almeno un esempio di *interrogazione*, uno di *inserimento* e uno di *cancellazione* di dati
Si assume l'uso di un DBMS *relazionale* (è possibile concordare l'utilizzo di un DB di tipo diverso). N.B. i parametri di connessione devono stare in un *file separato*.

Requisiti progetto

- **Sessioni utente:** L'applicazione **deve** includere il controllo della validità della sessione utente
- **AJAX, JavaScript:** L'applicazione **deve** contenere almeno un esempio di interazione client-server basata su AJAX
- Nel frontend JavaScript deve interagire con il DOM per la gestione degli eventi e dell'interfaccia utente
- **Framework PHP, Bootstrap:** L'utilizzo di framework PHP è **opzionale** e la scelta del framework è **libera**. L'utilizzo di framework o librerie client-side (es. Bootstrap) è **opzionale** [N.B. l'argomento fa comunque parte del programma del corso e può essere argomento di **interrogazione**]

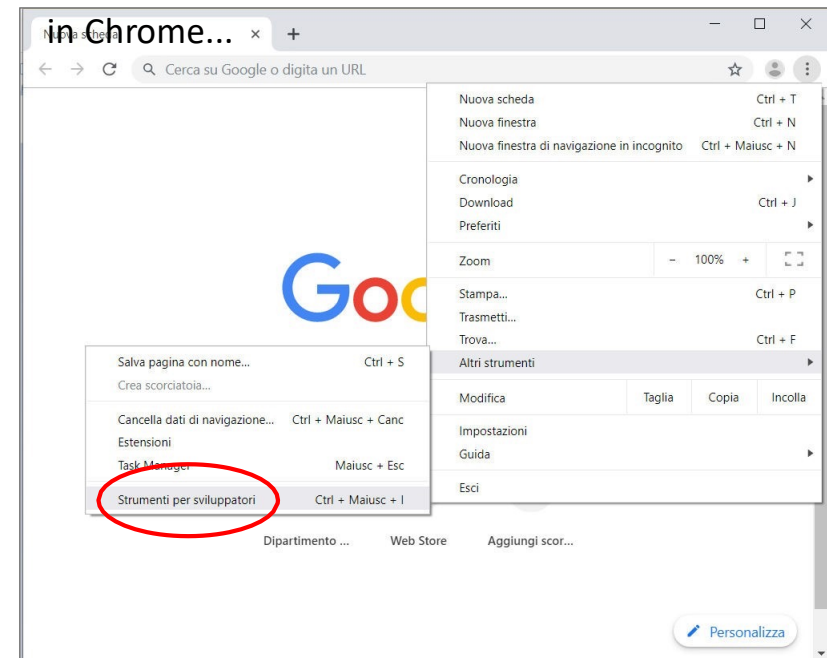
Requisiti progetto

- **REST:** L'applicazione
 - **deve** esporre almeno un servizio secondo il paradigma REST per ognuna delle seguenti funzionalità: ***create, read, update, delete*** (BACKEND)
 - **deve** dimostrarne l'utilizzo da parte di un **client** (FRONTEND)
- **JavaScript e DOM:** Il **client** (frontend) **deve** essere realizzato con JavaScript (è possibile usare jQuery o concordare l'utilizzo di soluzioni client-side diverse) e **deve** mostrare:
 - la **gestione degli eventi**
 - l'invio delle richieste **AJAX** per l'invocazione dei servizi
 - la gestione dei **risultati** dei servizi (cioè dei dati inviati dal server)

N.B. Il client (frontend) deve essere *plausibile dal punto di vista dell'interazione con l'utente* (es.: un frontend analogo al Case Study – edicola fumetti – versione 1 NON è adeguato!)

Requisiti progetto

- Il codice **deve** essere adeguatamente commentato
- Un paio di consigli:
 1. ricordatevi che in **JavaScript** la funzione `console.log(<string>)` scrive sulla console: usatela per il **debugging**!
 2. potete, ovviamente, usare i **case study** e gli esempi visti nelle lezioni, ma cercate di **progettare con la vostra testa** (non limitatevi a realizzare una “copia” dei case study...)



Requisiti progetto

- L'uso di **LLM** (es. **ChatGPT**) **non è vietato** ma fatene buon uso:
 - **NO** farsi fare l'intero progetto da un'intelligenza artificiale (o da un amico informatico naturale)
 - **Sì** al "pair programming":
 - chiedo consiglio e mi faccio aiutare a revisionare il codice e a risolvere i problemi
- Ricordate che alla **discussione** dovrete sapere spiegare in **estremo dettaglio** il funzionamento del codice!

