

Linee guida per la consegna del progetto

Il progetto può essere sviluppato da massimo tre persone. Il progetto obbligatoriamente deve avere un titolo, una categoria (ad esempio Ambiente, Turismo, Cultura, Politica ecc) e una o più sorgenti da cui sono stati prelevati i dati (ad esempio Istat), che verranno specificati in fase di caricamento.

Per poter caricare i progetti sul server centrale occorre distinguerli univocamente, per tale ragione abbiamo scelto di utilizzare il numero di matricola di uno dei componenti del gruppo. Quindi al fine di poter caricare il progetto si devono seguire alcune regole pratiche. Supponiamo che la matricola scelta sia il numero 100000. Avremo i seguenti vincoli da rispettare:

1-Il nome della **cartella** contenente il progetto deve essere **100000**.

2-il nome del **database** contenente tutte le tabelle deve essere **100000**.

3-Il nome del file **dump del database** deve essere **100000.sql**

4-La **home page** del progetto che risiede nella cartella 100000 si chiama **index.html**

5-Tutte le url del utilizzate nel codice a partire dal file 100000/index.html devono essere relative e non assolute. Questo vale sia lato server (php) che lato client (html, javascript). Ad esempio una chiamata Ajax deve essere effettuata nel seguente modo:

```
$.getJSON("api/myapi.php", function(dati){
    // mio codice
});
```

Non si può utilizzare una url assoluta (ad esempio <http://localhost/mioprogetto/api/myapi.php>) per effettuare la chiamata Ajax.

6-E' obbligatorio l'uso della **dbLibrary.php** da inserire nella cartella libraries

7-E' obbligatorio l'uso file **config.php** che va editato inserendo il numeromatricola di cui non vanno modificate il nome delle variabili.

```
<?php
include('../libraries/dbLibrary.php');

$database = '100000'; // inserire il numero matricola
$password = '';
$username = 'root';
$servername= 'localhost';
$db = openDB($database, $password, $username,$servername);
?>
```

Questo file va incluso in tutte le api php realizzate e si deve utilizzare la variabile \$db ogni volta che si chiama una funzione della libreria. All'interno delle API vanno richiamate le funzioni messe a disposizione dalla libreria in particolare select() sql() e closeDB().

8-La struttura del progetto deve essere la seguente che si può scaricare da questo link:
<http://wafi.iit.cnr.it/lab/LPW/download/matricola.zip>:

100000

```
api -> config.php, ... (altre api che interrogano il db)
css (file di stile css)
dump -> 100000.sql
libraries -> dbLibrary.php (altre librerie js)
php (altri programmi php ex un web scraper)
index.html
relazione.pdf
```



api

script php che interrogano il DB



css

fogli di stile



dump

DUMP in formato sql del database comprendente tabelle e dati



images

immagini usate



js

script javascript realizzati



libraries

librerie javascript e php utilizzate



php

ulteriori script php oltre alle api



index.html

pagina iniziale dell'applicazione



relazione.pdf

relazione in formato pdf

Se necessario, si possono aggiungere altre cartelle a piacere.

9-La relazione, che non deve essere superiore alle 6 pagine, deve rispettare la struttura definita dal template, disponibile a questo indirizzo:

<http://wafi.iit.cnr.it/lab/LPW/esame/doc/TemplateRelazione.docx>

Verifica della correttezza del progetto

Una volta caricato il progetto sul server, verrà chiesto di verificare che esso funzioni correttamente.

I passi da seguire per la verifica sono i seguenti:

1. provare ad usare il sito web e vedere se tutto funziona correttamente
2. in caso non funziona qualcosa, aprire la console javascript e verificare eventuali problemi
3. provare ad usare le API implementate direttamente dal lato server. Dal browser, digitare <http://wafi.iit.cnr.it/lab/LPW/progettini/numerodimaticola/api> e provare ad eseguire le singole api implementate
4. verificare la correttezza del codice HTML.

In caso di problemi, provare a risolverli localmente, sulla propria macchina, e annullare la procedura di caricamento del progetto. Una volta corretti i problemi provare ad avviare una nuova procedura di caricamento del progetto.

Problemi comuni

- Il progetto non funziona correttamente in quanto nella creazione degli array si usa la sintassi `$mio_array = []`. Per creare un array utilizzare la sintassi `$mio_array = array()`.
- la piattaforma di caricamento dei progetti è stata testata con Google Chrome. Sono stati segnalati problemi con l'uso di Firefox.
- nei sistemi windows spesso per includere un file si usa la barra `\` per identificare il percorso. I sistemi unix invece utilizzano la barra `/` per l'inclusione dei file. Dato che il nostro server è un sistema unix, ogni volta che si include un file, sostituire la barra `\` con la barra `/`.