

Tecnologie Web T
05 Luglio 2019 – Compito

Tempo a disposizione: 3 ore

La soluzione comprende la **consegna elettronica** dei seguenti file mediante l'apposito applicativo Web **esamix** (<http://esamix.labx>):

Autocomp.zip file zip contenente il sorgente java/class e pagine Web per punto 1
Google.zip file zip contenente pagine Web e file txt per punto 2
Unibo.zip file zip contenente il sorgente java/class e txt per punto 3

Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto

N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), equamente distribuiti sui tre esercizi, ovvero almeno 6 punti sul primo esercizio, 6 punti sul secondo esercizio e 6 punti sul terzo esercizio

Studenti in debito di Tecnologie Web L-A

Viene richiesto lo svolgimento dei soli esercizi 1 (17 punti) e 2 (16 punti). Tempo a disposizione: 2 ore.

I 18 punti necessari per l'ammissione all'orale sono così distribuiti: almeno 10 punti sul primo esercizio e almeno 8 punti sul secondo

ESERCIZIO 1 (11 punti)

Si realizzi un'applicazione Web, principalmente basata su **tecnologie JSP, Java servlet, Javascript, Ajax e JSON** per la realizzazione di una funzionalità di **completamento automatico di parole di ricerca** basata su **popolarità globale e di gruppo**.

L'applicazione deve consentire a utenti autenticati (tramite username e password in pagina login.html; in fase di autenticazione all'utente sarà richiesto anche un *id* intero di gruppo di appartenenza) di cominciare a inserire una parola come parametro di ricerca su un ipotetico search engine Web. Tale parola deve essere automaticamente completata server-side sulla base delle passate parole di ricerca più popolari su tutti gli utenti e sugli utenti dello stesso gruppo del richiedente.

In particolare, il cliente deve preliminarmente controllare che il carattere inserito sia un alfabetico minuscolo; in caso contrario, gli alfabetici devono essere convertiti da maiuscoli a minuscoli, i non alfabetici devono essere rimossi. Gli alfabetici minuscoli validi inseriti devono essere trasferiti verso il server in formato JSON. Il servitore deve determinare l'autocompletamento più appropriato selezionando la parola più popolare secondo un punteggio di popolarità costituito per il 75% dalla popolarità globale della parola e per il 25% dalla popolarità della parola sul gruppo. Anche la parola autocompletata deve essere trasferita in formato JSON.

Infine, deve essere data la possibilità all'amministratore (pagina admin.jsp) di azzerare le statistiche di popolarità di un determinato gruppo o globali; tutte le statistiche si azzerano comunque, automaticamente, dopo 7 giorni di utilizzo continuativo dell'applicazione.

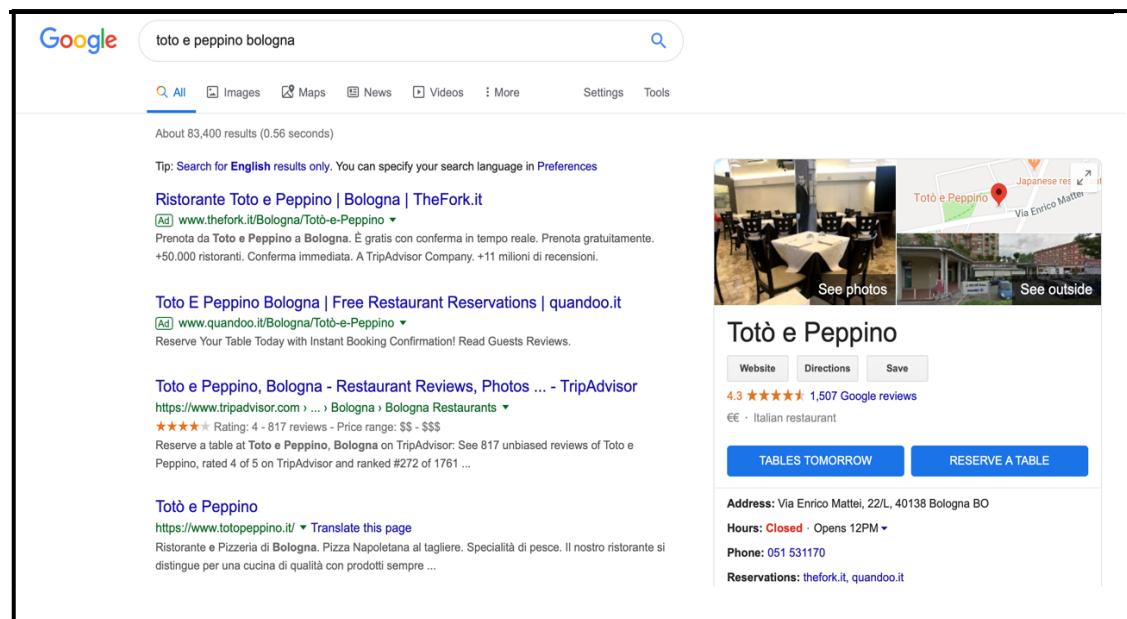
Tecnologie Web T

05 Luglio 2019 – Compito

ESERCIZIO 2 (11 punti)

Si realizzino le **pagine Web dinamiche** (basate su **tecnologia HTML, CSS, e Javascript**) in grado di **riprodurre il contenuto e il layout grafico** dello “snapshot” del sito Web Google di seguito riportato. Nello specifico, la pagina rappresenta il risultato della ricerca Google “*toto e peppino bologna*” eseguita mediante l’apposito campo di ricerca (N.B. il campo di ricerca può ammettere solo stringhe alfabetiche formate da esattamente 4 parole scritte tutte in minuscolo). La pagina presenta poi un **menù orizzontale composto da link** ai servizi Google su dati specifici (i.e., “All”, “Images”, “Maps”, “News”, “Video”, ecc.) di cui “All” rappresenta la scelta corrente. La parte centrale della pagina è dedicata alla visualizzazione del risultato della ricerca. La stessa è divisa verticalmente in due sezioni: quella di sinistra riporta i risultati testuali della ricerca mentre quella di destra mostra un mosaico di tre contenitori visuali nella sula parte superiore, seguito da informazioni e link testuali, tutti relativi al ristorante di interesse “Totò e Peppino”. Nel dettaglio, il primo contenitore del mosaico visuale punta a una galleria di fotografie, il secondo alla mappa con relative indicazioni stradali, mentre il terzo al servizio “street view”. Si modellino come **link** tutti i **bottoni e le parti testuali visualizzate nello snapshot in azzurro/verde** mentre come testo semplice tutto il resto. Lo studente è tenuto a giustificare adeguatamente le scelte relative ai principali costrutti HTML e CSS usati nella soluzione proposta nel file **Google.txt**.

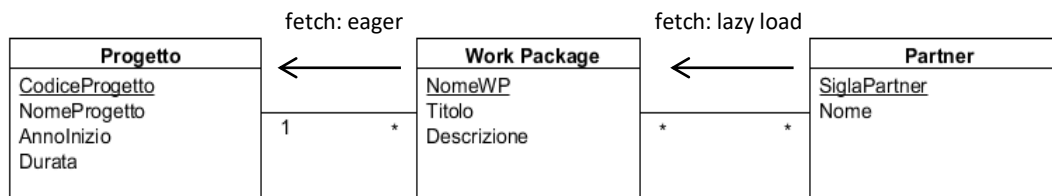
N.B. La soluzione NON deve far uso del costrutto HTML `frame`. Per rappresentare le immagini riportate nello snapshot, si utilizzino figure di esempio a piacere.



Tecnologie Web T
05 Luglio 2019 – Compito

ESERCIZIO 3 (11 punti)

Partendo dalla realtà illustrata nel **diagramma UML** di seguito riportato, si fornisca una soluzione alla gestione della persistenza basata su **Pattern DAO** in grado di “mappare” efficientemente e con uso di ID surrogati il modello di dominio rappresentato dai **JavaBean Partner, Work Package e Progetto** del **diagramma UML** con le corrispondenti **tabelle relazionali** derivate dalla **progettazione logica** del **diagramma** stesso.



Nel dettaglio, dopo aver creato da applicazione Java le tabelle all’interno del proprio **schema** nel database **TW_STUD** di **DB2** (esplicitando tutti i **vincoli** opportuni di **PK** e **FK**), **implementato i JavaBean** e **realizzato le classi** relative al **Pattern DAO** per l’accesso **CRUD** alle tabelle, si richiede la realizzazione dei seguenti metodi:

- **un metodo che restituisca il numero di progetti (e relativi nomi) a cui partecipa l’Università di Bologna** (la cui sigla è “Unibo”);
- **un metodo che restituisca i nomi dei partner che partecipano a progetti triennali.**

Si crei poi un **main di prova** in grado di:

- (i) inserire due o più tuple nelle tabelle di interesse;
- (ii) fare uso corretto dei metodi realizzati ai punti precedenti al fine di produrre la stampa dei risultati sul file **Unibo.txt**;

il tutto, mediante opportuna gestione delle transazioni.

N.B. L’implementazione del **Pattern DAO** deve limitarsi al solo **DBMS DB2**. La soluzione Java **deve sfruttare esplicitamente i mapping 1-N e N-M specificati nell’UML**. Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.