

Tecnologie Web L-A  
A.A. 2009/2010

Simulazione d'esame

Tempo Disponibile:  
2 ore (all'esame saranno 3)

Ai candidati è consentita la consultazione di manuali e materiale didattico e l'accesso alle risorse disponibili all'interno del sito del corso.

Per motivi pratici e tecnologici, si richiede la consegna di TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in genere ai fini della valutazione (es: descrittori, risorse statiche o dinamiche, codice Java, ecc...) e NON dell'intero progetti.

### **Esercizio 1. Programmazione Client-Side**

Si richiede di implementare una pagina Web in tecnologia AJAX per il reperimento dei piani di studio degli studenti di una Università. La pagina è costituita da una form contenente un campo di testo editabile dall'utente.

Il campo di testo deve consentire all'utente di inserire la matricola dello studente.

La form è dotata di un bottone che consente di inviare la request al server. A seguito della pressione del bottone la pagina deve:

- 1) verificare che la matricola dello studente inserita dall'utente sia costituita da soli caratteri numerici
- 2) nel caso in cui la matricola dello studente non sia stata immessa dall'utente in formato numerico si richiede che sia presentato all'utente un pop-up che segnala il problema e impedisce l'invio della richiesta.
- 3) nel caso in cui la matricola sia costituita da soli caratteri numerici si richiede di inserire il contenuto della risposta sotto la form, all'interno di un elemento *div* che riporta nome e cognome dello studente, la sua matricola, e la lista (di elementi non numerati) degli esami presenti nel suo piano di studio.

Per semplicità assumiamo che sia sempre presente sul server uno studente con la matricola specificata e che i dati che restituiti siano in formato XML ed organizzati come segue:

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<studente>
  <identificazione>
    <nome> nome dello studente </nome>
    <cognome> cognome dello studente </cognome>
    <matricola> matricola dello studente </matricola>
  </identificazione>
  <piano>
    <nomeesame> nome esame </nomeesame >
    <nomeesame> nome esame </nomeesame >
    ...
    <nomeesame> nome esame </nomeesame >
  </piano>
</studente>
```

...dove *nome*, *cognome* e *matricola* rappresentano il nome, il cognome e la matricola dello studente, mentre *nomeesame* è la denominazione di un esame nel suo piano di studio.

Le informazioni devono essere richieste al server in modo asincrono tramite una chiamata GET e passando il codice di matricola dello studente.

### **Note:**

- pur senza sviluppare alcun tipo di logica server-side, è possibile (e, anzi, apprezzato!) sperimentare realmente l'interazione con il server indirizzando le richieste AJAX verso l'URL di una risorsa statica realmente presente sul server e costituita da un semplice file compatibile con le specifiche riportate
- in sede di esame, dato il maggior tempo a disposizione, potrebbe essere richiesto di validare i dati inseriti dall'utente mediante espressioni regolari
- in sede di esame, potrebbe essere richiesto di analizzare dati provenienti dal server in formato JSON

## Esercizio 2. Programmazione Server-Side

Si richiede di implementare una sezione dell'applicazione Web per la gestione degli studenti di un Ateneo. In particolare, si richiede che il candidato implementi due distinte pagine JSP che consentano di inserire nuovi studenti che si vogliono immatricolare (*InserisciStudente.jsp*) e di leggere i dati sugli studenti iscritti (*LeggiDatiStudenti.jsp*).

Le JSP hanno accesso ad un database costituito da una sola tabella, costituita dalle seguenti 4 colonne di tipo testo:

<b>Matricola</b>	<b>Nome</b>	<b>Cognome</b>	<b>CodCorsoDL</b>
------------------	-------------	----------------	-------------------

...dove *Matricola* è una stringa che rappresenta la matricola di uno studente, *Nome* e *Cognome* sono stringhe che rappresentano il nome ed il cognome dello studente, infine *CodCorsoDL* è una stringa che rappresenta il codice del corso di laurea cui è iscritto.

La pagina *InserisciStudente.jsp* consente di inserire un nuovo studente nel database; a tal fine:

- 1) richiede il passaggio dei parametri *Matricola*, *Nome*, *Cognome* e *CodCorsoDL* via HTTP POST.
- 2) una volta ricevuti i parametri, verifica che non esista nel database alcuno studente con la matricola specificata.
- 3) in questo caso, inserisce il nuovo studente nel database e restituisce all'utente una pagina di conferma dell'avvenuto inserimento. In caso contrario viene presentata all'utente una pagina in cui si segnala l'errore.

La pagina *LeggiDatiStudente.jsp* consente di visualizzare i dati che caratterizzano uno studente. A questo fine:

- 1) richiede il passaggio di una richiesta HTTP GET che specifica la matricola dello studente desiderato.
- 2) ricevuta la matricola dello studente, controlla che questo sia presente nel database
- 3) in questo caso, presenta all'utente una pagina in cui è inclusa una tabella strutturata come segue:

<b>Cognome</b>	<i>Cognome dello studente</i>
<b>Nome</b>	<i>Nome dello studente</i>
<b>Matricola</b>	<i>Matricola dello studente</i>
<b>Corso di Laurea</b>	<i>Corso di Laurea a cui lo studente è iscritto</i>

(la prima colonna della tabella è costituita da un contenuto statico che deve essere presentato all'utente, mentre la seconda colonna presenta i dati relativi allo studente e reperiti dal database).

4) nel caso in cui non esista alcuno studente registrato nel database con la matricola specificata, viene mostrata all'utente una pagina che segnala l'errore.

**Note:**

- si richiede di implementare l'accesso al database mediante pattern DAO (classi factory, interfacce e DTO), ma di lasciare l'implementazione concreta degli oggetti DAO come ultima cosa.
- sono graditi i commenti nel codice, specie a corredo di funzioni non completate e/o correttamente funzionanti.