

UNIVERSITÀ DI PISA
Facoltà di Ingegneria
Master Universitario di I livello in
Tecnologie Internet

Edizione 2010/2011

Elaborato per il corso “Progetto di Siti WEB”

Finalità

L'elaborato proposto consiste nello sviluppo di una applicazione web contestualizzata in un ben preciso ambiente d'utilizzo (quale p.es. aziende o enti pubblici). Il progetto è finalizzato a familiarizzare lo studente con la progettazione e l'implementazione di sistemi per la generazione dinamica di documenti HTML, a fronte richieste che prevedono l'utilizzo di dati recuperabili da database.

Strumenti software da utilizzare

DBMS: *PostgreSQL*; altre opzioni sono possibili previa discussione con il docente
Programmazione server-side: *Java ServerPages* e/o *Java Servlet* (a discrezione dello studente)
Web server: *Jakarta Tomcat*
Tecnologia di accesso a database: *JDBC*
Programmazione client-side: *JavaScript*, *Java Applet* (opzionali)
IDE: *NetBeans*

Traccia

Il progetto deve essere sviluppato in due fasi successive:

- 1) Progetto e implementazione della base di dati (eventuale riuso di un DB già sviluppato)
- 2) Progetto e sviluppo dell'applicazione web al di sopra della base di dati

La prima fase è mirata a organizzare in modo appropriato i dati che saranno necessari per l'applicazione web.

Nella seconda fase occorre progettare un'applicazione web che operi sullo schema della base di dati precedentemente definito (o su un suo sottoinsieme). Si dovranno pertanto identificare:

- 1) le funzionalità che si intendono supportare
- 2) le pagine da implementare a sostegno delle funzionalità prescelte
- 3) i dati coinvolti in ciascuna pagina
- 4) **lo schema di navigazione tra le pagine.**

Lo sviluppo dell'applicazione web potrà essere basato sull'appropriato riuso degli esempi sviluppati durante le esercitazioni.

Requisiti aggiuntivi

Ogni singolo progetto deve essere portato avanti da un gruppo di due/tre persone.

L'applicazione web dovrà prevedere **almeno**:

- uno o più layout di pagina (per tutte le pagine dell'applicazione) strutturati in sezioni, che permettano l'impiego di tecniche di inclusione nella sua composizione dinamica
- almeno una sezione a cui accedere esclusivamente attraverso **autenticazione** (da implementarsi a livello applicazione, eventualmente utilizzando anche *servlet filters*)
- una pagina che presenta il **risultato di una query di selezione** con join su due o più tabelle
- una pagina di **popolamento di record di una o più tabelle**
- una pagina con una porzione dedicata a banner pubblicitari, di cui almeno uno da implementarsi con tecnologia **JavaScript** (oppure: integrazione nell'applicazione di quanto sviluppato per il corso di "Tecnologie Web"); eventualmente, si può scegliere di integrare una Applet
- utilizzo di almeno una servlet
- (opzionale) utilizzo di almeno un JavaBean
- (opzionale) utilizzo del meccanismo di "connection pooling"
- (opzionale) utilizzo di XHTML

Si raccomanda che tutte le pagine dell'applicazione abbiano una barra di navigazione per facilitare gli spostamenti tra le diverse pagine; **la grafica del sito dovrà essere basata su fogli CSS**.

Al momento della consegna, l'applicazione dovrà essere esterna all'IDE utilizzato, dopo opportuno deployment su un web server indipendente da NetBeans.

Documentazione richiesta

Si richiede una adeguata documentazione del lavoro svolto, coerentemente con la traccia sopra delineata.

In particolare si richiede di fornire preventivamente una descrizione **funzionale** dell'applicazione sviluppata, a cui deve poi seguire una **panoramica architettuale** del sistema. E' necessario indicare chiaramente con un qualche formalismo grafico rigoroso (UML o simili) quali sono le principali caratteristiche di navigazione tra le pagine dell'applicazione.

Suggerimenti

Sviluppare il progetto *in maniera incrementale*, implementando in ciascun passo piccolissimi gruppi di pagine correlate.

Utilizzare una convenzione predefinita per i nomi dei parametri HTTP, e attenersi a essa; mantenere una lista dei nomi dei parametri da poter consultare durante lo sviluppo dell'applicazione.

Controllare lettera per lettera i nomi di tabelle e campi all'interno delle stringhe che contengono il testo di query SQL; ricordare che in PostgreSQL gli identificatori che contengono lettere maiuscole devono essere inseriti tra doppi apici.

Usare, per le query parametriche, i "PreparedStatement" per evitare errori sintattici di JDBC.

Prevedere, in fase di sviluppo, la presenza di messaggi a video a scopo di debugging.

Fare attenzione, nel caso di aggiornamento/popoloamento di tabelle coinvolte in vincoli di foreign key, a verificare preventivamente il soddisfacimento di tali condizioni.

Tener presente che gli errori più subdoli segnalati da Tomcat nell'accesso a JSP derivano spesso da problemi nell'indicazione di nomi di sorgenti dati, tabelle e campi, ed eventualmente da scorretta sintassi nelle query JDBC.