

**Tecnologie Web T (9 cfu)**  
**Prova d'Esame – 14 Aprile 2022 – Appello Straordinario**

**Tempo a disposizione: 180 minuti**

---

La soluzione comprende la consegna elettronica dei seguenti file:

<b>A1.zip</b>	file zip contenente il sorgente java/class e pagine Web per punto 1
<b>A2.zip</b>	file zip contenente il sorgente java/class e pagine Web per punto 2
<b>A3.zip</b>	file zip contenente il sorgente java/class e pagine Web 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), ben distribuiti sui 3 esercizi, ovvero in ciascuno dei tre esercizi si deve raggiungere una valutazione almeno quasi sufficiente.**

---

**ESERCIZIO 1 (11 punti)**

Si realizzi una applicazione Web per l'**elaborazione server-side di un testo**, basandosi principalmente sulle tecnologie Java Servlet e JSP.

L'applicazione Web deve permettere a utenti autenticati di inserire due stringhe: la prima stringa *target* rappresenta la parola da cercare in un testo; la seconda stringa *nomeFile* è il nome di un file già disponibile server-side; si controlli che entrambe le stringhe contengano solo caratteri alfabetici.

Nel momento in cui l'utente avrà inserito il carattere speciale '€' come ultimo carattere (da eliminarsi) della stringa *nomeFile*, i dati di input devono essere inviati automaticamente al servitore, senza alcun intervento esplicito dell'utente, per una elaborazione in pipeline del seguente tipo: una **servlet S1** dovrà aprire il file *nomeFile* e trasformarne ogni carattere alfabetico minuscolo nel corrispondente maiuscolo; il risultato dell'elaborazione di S1 dovrà essere inviato a una **JSP J2**, che dovrà contare il numero di occorrenze di *target* (con tutti i suoi caratteri in maiuscolo) nel testo passato da S1. Il conteggio, insieme al testo ricevuto tutto in maiuscolo, dovrà essere restituito al cliente in formato JSON.

In ogni momento di esecuzione dell'applicazione, previa autenticazione, un amministratore deve avere la possibilità di visualizzare:

- Il numero di richieste totali di servizio sottomesse da utenti non amministratori negli ultimi 60 minuti;
- Il numero di richieste totali sottomesse da parte dell'amministratore dall'inizio della esecuzione della applicazione Web.

**Tecnologie Web T (9 cfu)**  
**Prova d'Esame – 14 Aprile 2022 – Appello Straordinario**

**ESERCIZIO 2 (11 punti)**

Si realizzi una applicazione Web per **determinare se una matrice  $6 * 6$  è simmetrica**, ovvero se ogni suo elemento in posizione  $(i, j)$  ha lo stesso valore dell'elemento in posizione simmetrica  $(j, i)$ . L'applicazione Web deve essere basata principalmente su tecnologie Javascript, AJAX e servlet.

In particolare, l'applicazione Web deve permettere a utenti non autenticati di inserire i 36 elementi interi della matrice considerata, controllando che siano effettivamente numeri interi. Solo dopo avere terminato l'inserimento di tutti gli elementi validi, l'applicazione deve invocare automaticamente l'esecuzione server-side dell'operazione concorrente di verifica della simmetria matriciale: in modo completamente concorrente, due **servlet S1 e S2** dovranno controllare la condizione di simmetria, ciascuna per metà degli elementi della matrice di partenza. Ogni servlet dovrà restituire in modo autonomo un valore "vero"/"falso" a seconda del risultato del suo controllo parziale.

Solo dopo avere ricevuto tutti e due i risultati, il cliente dovrà visualizzare il risultato complessivo, ovvero se gli elementi inseriti per la matrice costituiscono una matrice simmetrica oppure no.

Infine, si realizzi una *estensione* dell'applicazione descritta sopra che permetta a un **amministratore** di **selezionare una qualsiasi richiesta di servizio** in corso **per poterla interrompere**.

---

**ESERCIZIO 3 (11 punti)**

Si realizzi la sola parte server-side di una applicazione Web che realizzi una semplice **chat multi-utente in cui sia impossibile scambiare messaggi proibiti**; l'applicazione deve essere basta sull'utilizzo di tecnologia WebSocket.

In particolare, ogni utente non autenticato deve avere la possibilità di connettersi alla chat e di vedere visualizzati tutti i messaggi scambiati sulla chat a partire da quel momento; inoltre, come usuale, deve essere possibile per l'utente inviare un messaggio a tutti gli utenti correntemente connessi alla chat. Il servitore non distribuirà agli utenti tutti i messaggi proibiti che eventualmente si tenteranno di scrivere sulla chat: ogni messaggio che comincia per "A" oppure "S" costituisce un messaggio proibito da eliminare.

Infine, la chat deve essere automaticamente terminata dopo 30 minuti dal suo inizio delle attività, ovvero dal momento in cui il secondo utente si è connesso alla chat stessa.