

Tecnologie Web T (9 cfu)
Prova d'Esame – 11 Gennaio 2022 – Versione B

Tempo a disposizione: 180 minuti

La soluzione comprende la consegna elettronica dei seguenti file:

B1.zip	file zip contenente il sorgente java/class e pagine Web per punto 1
B2.zip	file zip contenente il sorgente java/class e pagine Web per punto 2
B3.zip	file zip contenente pagine Web e codice React.js 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 casuale server-side di un testo**, basandosi principalmente sulle tecnologie Java Servlet e JSP.

L'applicazione Web deve permettere a utenti non autenticati di inserire un testo di non più di 1000 caratteri, esclusivamente alfabetici. All'inserimento del carattere speciale “£” o all'esaurimento dello spazio a disposizione di 1000 caratteri, **il testo deve essere inviato automaticamente**, senza alcun intervento esplicito dell'utente, al servitore per una elaborazione in pipeline del seguente tipo: una **JSP J1** dovrà scegliere casualmente un carattere alfabetico ed eliminarne tutte le occorrenze dal testo; il risultato dell'elaborazione di J1 dovrà essere inviato a una **servlet S2** che dovrà avere il medesimo comportamento e in più calcolare la lunghezza totale del testo risultante. Il testo modificato più il conteggio dovranno essere restituiti al cliente in formato XML.

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

- La lista delle sessioni che sono state attive negli ultimi 7 giorni;
- Il numero di richieste di elaborazione testo sottomesse in ogni sessione.

Inoltre, deve essere possibile per l'amministratore forzare **la terminazione, il più possibile immediata**, di una qualunque richiesta di elaborazione corrente.

Tecnologie Web T (9 cfu)
Prova d'Esame – 11 Gennaio 2022 – Versione B

ESERCIZIO 2 (11 punti)

Si realizzi una applicazione Web per la **compilazione collaborativa di un cruciverba**; l'applicazione Web deve essere basata principalmente su tecnologie Javascript, AJAX e servlet.

In particolare, l'applicazione Web deve permettere a soli **utenti autenticati** di accedere a un cruciverba costituito da una matrice $10 * 10$ di caratteri. Ogni utente potrà inserire in ogni momento caratteri nella matrice; ogni carattere inserito deve essere necessariamente un **carattere alfabetico maiuscolo**. A ogni digitazione di un carattere, deve essere possibile visualizzare la versione aggiornata del cruciverba a ogni utente collegato al cruciverba stesso, il prima possibile.

Quando una parola è completata (10 caratteri presenti in una qualsiasi riga o colonna), allora un componente server-side deve concorrentemente verificare se la parola inserita è corretta, verificandone la presenza in un dizionario costituito dal file "cruciverba.txt", già presente lato servitore; il file "cruciverba.txt" è formattato con una sola parola per linea.

Infine, si realizzi una **estensione (versione 2)** dell'applicazione descritta sopra che **impedisca di inserire più di 500 caratteri per cruciverba a ogni utente**.

ESERCIZIO 3 (11 punti)

Si realizzi in React un'applicazione Web lato cliente che simuli il gioco "**Campo minato**". L'applicazione dovrà eseguire interamente sul browser senza interagire con alcun server remoto.

L'interfaccia dell'applicazione sarà composta dalle seguenti sezioni:

- **Sezione Configurazione.** In tale sezione è presente un elemento di input per l'inserimento della dimensione della griglia di gioco (d) e un elemento di input per l'inserimento del numero di passi (p , con $p > 3$) da effettuare per ogni partita. Solo dopo avere acquisito questi dati, l'utente potrà iniziare a giocare.
- **Sezione Griglia.** Tale sezione deve contenere una griglia quadrata ($d * d$) di celle inizialmente tutte di colore giallo, ciascuna delle quali può potenzialmente nascondere una mina. Per ogni riga della griglia, vengono posizionate 2 mine su altrettante celle scelte in modo del tutto casuale. L'utente deve percorrere p passi sulla griglia con l'obiettivo di non calpestare alcuna mina. Ogni passo viene simulato da un click sulla cella che l'utente intende calpestare. Se l'utente calpesta una cella senza mina, la cella assume il colore blu e l'utente guadagna 5 punti. Se l'utente calpesta una cella con una mina, la cella assume il colore rosso, la partita termina e all'utente appare una finestra di dialogo con la scritta "Partita terminata: percorso non completato!". Se invece l'utente avrà terminato i p passi senza mai calpestare una mina, gli apparirà una finestra di dialogo con la scritta "Partita terminata: percorso completato!". Dismessa la finestra di dialogo, la griglia di gioco verrà resettata (mantenendo le stesse dimensioni acquisite in fase di configurazione) e l'utente potrà cominciare una nuova partita.
- **Sezione Conteggio.** In questa sezione, occorre visualizzare due contatori che tengono traccia, rispettivamente, del numero di percorsi completati e del numero di percorsi non completati. Occorre inoltre visualizzare il punteggio complessivo accumulato dall'utente in tutte le partite giocate.