



Esempi di quesiti di prove d'esame

Su Virtuale:

Versione elettronica 1 pagina per foglio = EsempiQuesitiEsame.pdf

Testo tipologia esercizio n.1

Gestione della coda in un ufficio comunale,

basata su tecnologie Javascript, Java Servlet e JSP

L'applicazione Web deve permettere a un utente **autenticato** di aggiungersi alla coda di cittadini in attesa e di specificare se è interessato all'ufficio concessioni edilizie oppure a quello di iscrizioni scolastiche; in risposta deve ricevere una pagina Web con l'indicazione del **presunto tempo di attesa**. Il tempo di attesa dipenderà dal numero di cittadini già in coda per quello specifico ufficio, con x minuti per cittadino nel caso di concessioni edilizie, y nell'altro caso. **Siano x e y parametri di inizializzazione configurabili a livello di intera applicazione Web**

Se un cittadino in coda vuole un aggiornamento sul tempo in attesa, deve esplicitamente richiederlo; NON è consentito l'utilizzo del meccanismo delle Websocket

Nel caso in cui arrivi un **cliente VIP** (*username=sindaco, password=vip*), la sua pratica deve **essere evasa immediatamente per prima**, con eventuale modifica dei tempi di attesa per gli altri cittadini in coda

Inoltre, visto che l'ufficio comunale chiude alle 12:30, si faccia in modo che qualora la somma fra ora attuale e massimo tempo di attesa superi le 12:45, l'applicazione Web non accetti eventuali nuovi cittadini, comunicando loro la necessità di ritornare il giorno successivo

Testo tipologia esercizio n.2

Calcolo concorrente della somma di due matrici di numeri interi

i cui elementi sono contenuti all'interno di un file JSON di grande dimension

L'applicazione deve essere basata esclusivamente sulle tecnologie Javascript, Ajax, Java Servlet e JSP (no Websocket)

L'applicazione deve permettere a un utente **non autenticato** di inserire le dimensioni delle matrici considerate ($a*b$) e di fornire in input un file **JSON** contenente i ($2*a*b$) elementi interi delle due matrici, secondo il formato e l'ordine che si preferiscono

Sempre a propria preferenza, **per il file JSON si supporti o l'inserimento manuale in input o il caricamento da un file già salvato presso il cliente**

La somma deve essere eseguita **server-side e concorrentemente**: il cliente deve dividere il file JSON **in tre parti**, circa della stessa dimensione e inviarle a una servlet per il calcolo concorrente server-side da parte di **tre thread distinti**. **Solo dopo** che tutti i risultati della matrice somma saranno ricevuti lato cliente, l'applicazione dovrà visualizzare tutta insieme l'intera matrice somma

Inoltre, in ogni momento, tramite **una servlet di amministrazione**, previa autenticazione, l'amministratore di sistema deve poter visualizzare **quante richieste** di calcolo matriciale sono state già completate **nel giorno corrente e la durata temporale** di ciascuna di esse

Testo tipologia esercizio n.3 (React)

Ricerca della sequenza ordinata di numeri interi.

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

Sezione Gioco. La sezione conterrà una griglia 3x3 di caselle di testo. La griglia è inizializzata riempiendo a caso le 9 caselle di testo con numeri da 1 a 9. Il giocatore inizialmente non vedrà alcun numero contenuto nelle caselle, le quali assumeranno quindi il ***colore grigio (stato di casella coperta)***. Obiettivo del gioco è quello di scoprire, ad una ad una, tutte le caselle secondo la sequenza crescente dei numeri da 1 a 9. La scopertura di una casella avviene cliccando su di essa. Quando l'utente clicca su una casella coperta contenente un numero che rispetta la sequenza, ***la casella si colorerà di verde*** e visualizzerà il numero in essa contenuto. Se invece l'utente clicca su una casella coperta contenente un numero non in sequenza, la casella non cambierà di stato ed apparirà un ***messaggio di alert*** che rivela numero contenuto della casella e invita l'utente a riprovare. Nulla accade se l'utente clicca su una casella scoperta. Infine, la sezione contiene un bottone per resettare il gioco nonché le informazioni contenute nella sezione Punteggio

Sezione Punteggio. La sezione dovrà riportare i seguenti dati aggiornati in tempo reale: ***numero complessivo di tentativi di scopertura di caselle falliti; la sequenza di fallimenti consecutivi più lunga***

Testo tipologia esercizio n.3 (Websocket)

Disseminazione probabilistica di informazioni in un gruppo, basata su tecnologia Websocket

L'applicazione Web deve consentire a un utente ***non autenticato*** di visualizzare ***tutti i messaggi ricevuti dall'inizio della sua sessione*** di interazione e di inviare un messaggio agli altri utenti correntemente connessi. Nel caso di invio, il messaggio inviato deve essere propagato ***solo al 10% degli utenti connessi***, scelti in modo casuale. I messaggi scambiati siano relativi a risultati di partite di Champions League e rappresentati in formato **JSON**

Inoltre, in ogni momento, tramite una **JSP di amministrazione**, previa autenticazione, si consenta all'amministratore di poter effettuare il ***push di tutti i messaggi*** gestiti dall'applicazione a ***tutti gli utenti correntemente connessi***, per riallineare le informazioni già probabilisticamente disseminate in precedenza

URL accessibili per la consultazione della documentazione

<https://lia.disi.unibo.it>

<https://nodejs.org>

www.w3schools.com

<https://cdnjs.cloudflare.com>

<https://virtuale.unibo.it>