RELAZIONE

* **Introduzione descrittiva**

Il progetto oggetto di questa relazione è stato sviluppato su commissione della **San Martino S.p.A.,**  in occasione del concorso a premi denominato *“San Martino Friendship Day”*, regolamentato dall’art. 11 del D.P.R. 26 ottobre 2001, n. 430.

L'obiettivo del concorso è quello di coinvolgere il pubblico tramite una registrazione online, assegnando un premio ai primi 499 utenti validamente registrati (*meccanica Rush&Win*) e prevedendo anche un’estrazione di consolazione tra i successivi. Il progetto consiste quindi nella realizzazione di un sistema informatico sicuro, scalabile e conforme alle normative vigenti, in grado di gestire in modo automatizzato ed efficace tutte le fasi della partecipazione.

La struttura del progetto si articola in tre componenti principali:

* **Frontend**: un'interfaccia web responsive per l’accesso e la registrazione degli utenti, comprensiva di un controllo preventivo dell’età.
* **Backend**: un'applicazione server-side in Java per la gestione delle registrazioni, l'invio di email di conferma e la verifica della validità delle iscrizioni.
* **Database**: un sistema relazionale MariaDB per il salvataggio sicuro e conforme dei dati personali degli utenti partecipanti.

Tra i requisiti fondamentali richiesti al software vi sono:

* Il salvataggio completo e sicuro dei dati personali dei partecipanti;
* L’assegnazione automatica e affidabile dei premi ai primi 499 utenti validamente registrati e confermati;
* La gestione di token univoci per confermare l’iscrizione via email;
* Il rispetto della normativa GDPR per la protezione dei dati personali;
* La registrazione dei tempi di partecipazione per garantire la correttezza nella logica “first come, first served”;
* L’implementazione di logiche difensive per impedire registrazioni multiple e partecipazioni fraudolente.

Il progetto è stato sviluppato in team e ha richiesto un approccio integrato e multidisciplinare, che ha coinvolto competenze di sviluppo software, sicurezza informatica, progettazione di basi di dati e gestione dei requisiti legali.

* **Sviluppo**

Il progetto è stato suddiviso in più componenti fondamentali, ciascuna sviluppata seguendo le best practice apprese durante il corso. Le parti centrali su cui si è concentrato lo sviluppo sono:

**1. Database e gestione dati**

La progettazione del database ha richiesto un’attenta analisi dei requisiti legali e funzionali del concorso. È stata realizzata una tabella principale (customers) per memorizzare le registrazioni degli utenti, con campi che includono dati anagrafici, residenza, spedizione, accettazione delle policy, e un sistema di token per la conferma via email. Sono stati creati indici per ottimizzare le query di ricerca e ordinamento, in particolare sul campo confirmedDate, necessario per determinare l’ordine di arrivo delle registrazioni.

**2. Backend Java**

Il backend è stato sviluppato in Java, utilizzando il microframework **Spark Java** per l’esposizione delle API REST. Sono stati implementati due controller principali:

* SubmissionsController per la gestione della registrazione e dell’invio delle email di conferma;
* ConfirmationController per gestire la conferma del link ricevuto via email.

L’invio delle email è stato affidato a un ExecutorService con thread pool in modo da non bloccare la logica principale dell'applicazione e garantire una buona esperienza utente.

La connessione al database è gestita tramite una connection pool in modo sia da garantire buone performance sia prevenire bug di rush condition.

È stato integrato un sistema di shutdown hook per il rilascio delle risorse (connessioni al database e thread).

**3. Frontend JavaScript**

Il frontend, sviluppato in HTML, CSS e JavaScript, include:

* Un modulo di controllo dell’età (ageCheck) che verifica localmente se l’utente ha compiuto 18 anni prima di consentire l’accesso al sito (in linea con la normativa sul consumo di alcol);
* Un’interfaccia utente per la registrazione con feedback interattivo;
* L’uso della sessionStorage per memorizzare temporaneamente la data di nascita e migliorare l’esperienza utente.

**Tecnologie utilizzate**

* **Java 17** per lo sviluppo dell’applicazione server;
* **Spark Java** per la gestione delle rotte HTTP;
* **MariaDB** come sistema di gestione del database relazionale;
* **Gson** per la serializzazione/deserializzazione JSON;
* **JavaScript** per la logica client-side;
* **HTML5 e CSS3** per la presentazione;

**Conoscenze applicate**

Nel corso del progetto abbiamo applicato diverse conoscenze acquisite durante il corso, tra cui:

* La progettazione di database relazionali normalizzati;
* La gestione di API RESTful;
* L’uso di token univoci per autenticazione temporanea;
* Il multithreading e l’uso delle pool di thread;
* La sicurezza nella gestione delle informazioni personali (inclusi consensi privacy e antispam);
* La separazione delle responsabilità tramite architettura a livelli (controller, service, repository).
* **Problemi riscontrati e soluzioni adottate**

Durante lo sviluppo del progetto abbiamo affrontato un paio di problematiche di natura tecnica:

**Invio delle email e necessità di multithreading**

Un'altra criticità si è presentata nella gestione dell’invio delle email di conferma. Inizialmente questa operazione veniva eseguita direttamente all'interno della logica di registrazione, causando ritardi nella risposta al frontend. Abbiamo quindi deciso di implementare un sistema di multithreading mediante un ExecutorService, in modo da eseguire l’invio in parallelo e restituire subito una risposta all’utente. Questa scelta ha migliorato significativamente la reattività del sistema.

* **Possibili implementazioni future**

**Una possibile evoluzione del progetto potrebbe essere l’introduzione della funzionalità di modifica** dei dati da parte dell’utente dopo la registrazione, entro un limite temporale predefinito.

Attualmente, per garantire la validità delle iscrizioni e semplificare il controllo dei dati, non è previsto che l’utente possa modificare le informazioni fornite dopo la conferma. Tuttavia, in ottica di miglioramento dell’esperienza utente e per gestire eventuali errori di compilazione (ad esempio un refuso nell’indirizzo di spedizione), si potrebbe sviluppare un sistema che consenta l’accesso a un’area personale tramite un link sicuro ricevuto via email.

L’introduzione di questa feature comporterebbe nuove sfide, sia in termini di sicurezza che di progettazione del database, ma rappresenterebbe un valore aggiunto in termini di usabilità e affidabilità del sistema.