

Progetto per il Corso Introduzione alla Programmazione per il Web a.a. 2024-2025

Istruzioni da leggere attentamente

La consegna del progetto deve essere effettuata su Moodle nella sezione Progetto nella cartella creata per l'appello a cui vi iscrivetevi (es. AppelloLuglio).

La consegna consiste di tre file:

1. Un file pdf contenente la relazione. La relazione del progetto deve essere redatta sull'apposito template che potete scaricare da Moodle nella sezione Progetto. La lunghezza della relazione è di massimo 5 pagine tutto incluso
2. Una cartella zippata contenente la web app MatchWeb
3. Una cartella zippata contenente la web app REST PartiteWeb

Ognuno dei 3 file essere nominato come segue: "TeamID_Relazione", "TeamID_MatchWeb", "TeamID_REST", dove ID è l'ID associato al vostro gruppo (es. Team04). L'ID del gruppo è riportato nell'apposito file su Moodle nella sezione Progetto (il file sarà disponibile nei prossimi giorni). Va consegnato sia il codice che la versione deployata di ogni web app.

Attenzione!!! Prima di fare lo zip, eliminate dalla cartella target di ogni app i file .war. e war.original.

Se la taglia finale degli zip supera i 100M. Estrarre le cartelle SNAPSHOT, zipparle separatamente e consegnare un ulteriore file. Consiglio per limitare lo spazio: usare immagini a bassa risoluzione.

Per venire incontro alle esigenze degli studenti, le date delle consegne per gli appelli di Giugno e Luglio sono così fissate:

Giugno *entro le 23:59 del 18 Giugno*

Luglio: *entro le 23:59 del 16 Luglio*

Per tutti gli altri appelli, la consegna è fissata a sette giorni prima della data del quiz.

Attenzione: ad ogni appello, verrà accettata una sola consegna per team. Quindi, in caso di consegne multiple da parte di più studenti appartenenti allo stesso team, verrà valutata l'ultima consegna ricevuta alla scadenza prevista. Pertanto, si suggerisce di delegare un membro del team ad effettuare la consegna.

Una nuova versione del progetto può essere consegnata solo ad una nuova sessione di esame. Durante l'anno accademico, ci sono due sessioni di esame: quella estiva costituita dagli appelli di Giugno, Luglio e Settembre, e quella invernale costituita dagli appelli di Gennaio e Febbraio. In caso di consegna di una nuova versione del progetto, nominare i file consegnati aggiungendo "v2" e consegnare un file supplementare in formato pdf chiamato "Modifiche" riportante la lista delle modifiche apportate al progetto rispetto alla sua versione originale.

Il progetto deve essere sviluppato utilizzando esclusivamente le tecnologie viste durante il corso, ovvero HTML, CSS, Bootstrap, tutta la parte per l'implementazione in Spring del

back-end, JavaScript, AJAX, AJAJ, etc. L'utilizzo di tecnologie diverse porterà ad una consistente penalità. Nota: tecnologie non significa funzioni o simili.

Un accenno particolare alla tecnologia del DB:

- Si usi H2 come DBMS.
- Si consegnino tutti i file riportanti le istruzioni SQL per creare le tabelle usate dalla vostra web app. Tali file devono avere estensione .sql

Il progetto deve usare almeno 3 elementi di Bootstrap scelti a piacere.

Il progetto vale il 50% del voto finale dell'esame e verrà valutato solo se gira. La mancata consegna di alcune parti del progetto non implica automaticamente l'insufficienza.

I criteri di valutazione sono i seguenti:

- 1) Progettazione della web app
- 2) Corretta implementazione delle funzionalità della web app
- 3) Organizzazione e qualità del codice
- 4) Completezza, correttezza e chiarezza della relazione

In caso di progetti identici o molto simili tra due o più team il voto del progetto verrà diviso tra tutti i componenti del progetto. Ricordate che ChatGPT o affini non fanno parte del team ☺
Esempio: per due progetti identici e sei membri totali dei due gruppi, il voto finale sarà: voto/6

Buon lavoro!

Progettare e sviluppare una web app *MatchWeb* che permetta di compilare schedine relative alle partite dei campionati nazionali di massima divisione di 3 sport di squadra. Per incentivare l'uso della web app, i gestori mettono in palio premi per tutti gli utenti registrati. I premi saranno assegnati durante il campionato e consistono in: un biglietto per assistere a una partita della squadra del cuore, una maglia ufficiale di un giocatore della squadra, una serata in pizzeria insieme alla squadra. La web app interagisce con un'altra web, chiamata *PartiteWeb*, che espone dei REST endpoints. Dovrete progettare e sviluppare anche questa seconda web app.

Per garantire semplicità nell'implementazione, tutti gli aspetti temporali relativi al campionato e all'accesso ai risultati sono stati rimossi. Non occorre quindi prendere in considerazione il tempo.

Nota bene: Si richiede l'implementazione del servizio per uno solo dei 3 sport.

La web app *MatchWeb* consiste di una parte pubblica e di una parte privata, quest'ultima accessibile attraverso un'autenticazione con login e password. Tutte le pagine della web app sono disponibili in italiano. Per ciascuna delle pagine il titolo della tab del browser in cui viene visualizzata la pagina web è il nome del servizio che sarà corredato con il logo (da voi prodotto).

La parte pubblica ha le seguenti pagine:

- 1) Una *home page* (chiamata *index*) che deve riportare:

- a. il nome del servizio e il suo logo;
 - b. un testo con un messaggio di benvenuto;
 - c. la lista dei 3 sport per i quali si può giocare la schedina. Per ogni sport si mostri un'immagine sulla quale si può cliccare per accedere a una pagina che riporta una sua breve descrizione e il nome delle squadre partecipanti al campionato;
 - d. i dati relativi all'azienda che gestisce il servizio: indirizzo della sede legale, partita iva, contatti.
- 2) Una pagina di *sign up* che permette ad un nuovo utente di iscriversi al servizio. La pagina contiene una form avente i seguenti campi (tutti obbligatori):
- a. nome;
 - b. cognome;
 - c. data di nascita nel formato GG/MM/AAAA;
 - d. indirizzo email;
 - e. la scelta di uno username;
 - f. la scelta di una password che andrà inserita due volte per conferma. La password deve essere lunga 9 caratteri di cui uno speciale e due cifre, e deve essere la concatenazione della stringa "utente!team_id" dove team_id è il numero del vostro team (e.g. 01, 07, 12, etc);
 - g. un select menù per scegliere lo sport per il quale si vuole accedere al servizio
 - h. un select menù per scegliere la squadra del cuore;
 - i. un pulsante per inviare i dati;
 - j. un pulsante di reset che ripulisce tutti i campi
- 3) Una pagina per avvisare l'utente che la sua registrazione è andata a buon fine
- 4) Una pagina di *login* che contiene una form richiedente username e password
- 5) Una pagina di *logout* che contiene un bottone che permette di fare logout dal servizio
- 6) Una pagina per ognuno dei 3 sport supportati dal servizio. Per ognuna di tali pagine si scriva una breve descrizione dello sport e si riporti la lista delle squadre che partecipano al campionato in ordine alfabetico
- 7) Una pagina con i loghi degli sponsor e una loro breve descrizione
- 8) Una pagina con le recensioni testuali degli utenti del servizio e una votazione numerica tra 1 e 5. La pagina verrà aggiornata con tutti i commenti disponibili in un database embedded (vedi sotto) tramite un servizio interno della web app chiamato *MostraCommenti*.

Ciascuna pagina della parte pubblica deve dare la possibilità all'utente di accedere: alla pagina per iscriversi al servizio, alla pagina per loggarsi ad esso, alla pagina che riporta i loghi degli sponsor e una breve loro descrizione, alla pagina con le recensioni fatte dagli utenti del servizio, alla home page.

In tutte le pagine pubbliche c'è una sezione news gestita tramite AJAX/AJAJ che riporta notizie sul campionato (es. la squadra X ha comprato oggi il giocatore Y). Le news si aggiornano automaticamente ogni 30s.

Il formato di data di nascita (nella pagina di *sign up*) e il formato della password (nella pagina di *sign up* e nella pagina di *login*) dovrà essere validato utilizzando un opportuno script JavaScript prima di inviare

i dati al server quando viene premuto il bottone Submit. Lo script JavaScript verificherà inoltre che l'utente sia maggiorenne per abilitare la registrazione. Nel caso in cui l'utente non selezioni l'unico sport del cui servizio fornirete l'implementazione, lo script JavaScript mostrerà un messaggio che indica che lo sport scelto non è supportato non permettendo anche in questo caso la registrazione. Dopo l'avvenuta verifica dei dati della form di registrazione, viene invocato un service interno alla web app, chiamato *Signup*, che accede ad un database embedded (vedi sotto) che contiene i dati degli utenti registrati e verifica l'esistenza o meno di un utente con lo stesso username. Se esiste un utente con lo stesso username, viene presentata nuovamente la pagina *sign-up* con un messaggio addizionale di errore. Altrimenti, i dati del nuovo utente vengono inseriti nel database e viene visualizzata la conferma della registrazione.

La web app usa una database H2 in modalità embedded. Per la gestione degli utenti, il database conterrà la tabella Users (che contiene, come sapete, ID, username, password, enabled) e la tabella Authorities (che contiene, come sapete, ID, username e il ruolo che può assumere l'utente) gestite automaticamente da Spring Security e un'ulteriore tabella per contenere gli altri dati relativi all'utente. Questa tabella verrà gestita da voi.

I ruoli supportati dalla web app sono: ADMIN, USER, e MODERATOR.

Quando un utente registrato (USER o MODERATOR) o l'amministratore (ADMIN) effettuano l'accesso al servizio, il server verifica la correttezza delle credenziali username e password criptata con uno degli algoritmi resi disponibili da Spring Security. Se le credenziali non sono corrette, gli utenti sono re-inviati alla pagina di *login* e viene mostrato un messaggio di errore del tipo “#team_id: Authentication failed!”. Altrimenti, in caso di credenziali corrette, hanno accesso alle rispettive pagine private.

Per registrare l'amministratore sulla web app: registrate un utente (USER) usando la stringa `ROLE_USER` per specificare il ruolo, e poi modificate manualmente il ruolo in ADMIN agendo sulla tabella Authorities sostituendo la stringa `ROLE_USER` con la stringa `ROLE_ADMIN`.

Le username che userete per i diversi ruoli sono: `user#team_id`, `moderator#team_id`, `admin#team_id`.

Per i ruoli USER e ADMIN la web app implementa una dashboard privata come descritto qui sotto. Per semplicità, i MODERATOR non hanno una dashboard dedicata ma accedono alla dashboard user.

Dashboard User:

La pagina mostra un messaggio di benvenuto per l'utente (e.g. “Benvenuta Giovanna!”) e offre le seguenti funzionalità abilitate premendo i pulsanti seguenti:

Visualizza Profilo che permette di accedere a una pagina con il profilo dell'utente (nome, cognome, data di nascita, mail, username, ruolo, numero di punti ottenuti alla giornata corrente, posizione in classifica utenti, premi vinti, numero di schedine giocate).

Visualizza Calendario Partite che permette l'accesso a una pagina che mostra il calendario delle partite del campionato. Il calendario viene ottenuto interrogando un REST endpoint della web app PartiteWeb (vedere sotto per i dettagli).

Cambio Password che permette di cambiare la password associata all'utente tramite form.

Gioca Schedina che mostra in una pagina una tabella con le partite della giornata corrente ottenute interrogando un REST endpoint della web app PartiteWeb (vedere sotto per i dettagli). La richiesta conterrà come parametro la giornata corrente, La tabella ha una colonna editabile dall'utente che consente all'utente di inserire il suo pronostico: 1 verrà dato se la squadra di casa è data per vincente, 2 verrà dato se la squadra ospite è data per vincente, 0 verrà dato in caso di pareggio. La pagina contiene un pulsante Submit per inviare la schedina. Una volta inviata la schedina, tramite AJAX/AJAJ, mostrate un messaggio di invio avvenuto con successo, aggiungete una colonna supplementare alla tabella della schedina che conterrà i risultati, e mostrate sotto la tabella il punteggio ottenuto dall'utente nella giornata. L'invio della schedina avviene inviando una richiesta a un service interno alla web app, chiamato *CalcolaPunteggio*, che interroga un REST endpoint della web app PartiteWeb per ottenere i risultati delle partite e calcolare il punteggio dell'utente. Anche in questo caso la giornata corrente verrà passata come parametro. Il punteggio è calcolato sommando 1 per ogni pronostico avverato. L'utente può giocare una sola schedina per giornata. Il numero di schedine giocate (cioè il numero di giornate) è salvato in una tabella Giornate del database.

Recensioni che permette di accedere a una pagina che permette all'utente di scrivere un commento sulla sua valutazione della web app e assegnare un voto numerico tra 1 e 5. I commenti e i voti vengono salvati nel database in una tabella Recensione del database.

Dashboard Admin:

La pagina mostra un messaggio di benvenuto per l'utente (e.g. "Benvenuta Giovanna!") e offre le seguenti funzionalità abilitate premendo i pulsanti seguenti:

Visualizza Lista Utenti Iscritti che permette di accedere a una pagina che riporta una tabella con la lista degli utenti ordinata in ordine alfabetico, il ruolo e la loro email.

Visualizza Classifica Utenti che permette di accedere a una pagina che riporta una tabella con la lista degli utenti ordinata per punti ottenuti fino alla giornata corrente.

Assegna Premi che permette, interrogando un opportuno servizio interno alla web app, chiamato *AssegnaPremi*, di assegnare ai primi 3 classificati i 3 premi estratti in modo randomico. I premi saranno inseriti in una tabella Premi del database.

Upgrade che permette di promuovere un utente a moderator.

Attenzione! Una volta che l'utente, il moderatore o l'amministratore sono loggati, bisogna aggiungere il link alla pagina di *logout* e rimuovere i link alle pagine di *sign up* e *login*. Per esempio, si possono predisporre due barre di navigazione diverse che saranno incluse in modo esclusivo a seconda che l'utente, il moderatore o l'amministratore sia loggato o meno.

REST web app PartiteWeb

PartiteWeb è una web app che viene interrogato dal servizio sviluppato e fornisce alcune informazioni relative al campionato: calendario delle giornate del campionato, partite della giornata corrente, risultati delle partite della giornata corrente. PartiteWeb deve generare il calendario delle partite, si assuma un numero pari di squadre partecipanti al campionato. Inoltre deve generare in modo randomico i risultati di tutte le partite di tutte le giornate. Per semplicità l'accesso a PartiteWeb non richiede autenticazione.

Per un paio di punti in più

Per avere 2 punti in più, scegliete **una** tra le seguenti opzioni. Il punto viene assegnato anche se le parti precedenti del progetto non sono state tutte completate.

1. All'atto del Sign-up permettere all'utente, solo se lo desidera, di inviare anche un'immagine (formato png) da inserire nella sua pagina profilo. L'immagine andrà salvata nel database.
2. Usando un'opportuna libreria (ce ne sono diverse! E.g. Rating.js) presentare graficamente le recensioni in modo simile a quello che realizzano i sistemi di e-commerce (stelline o like anziché voti numerici)
3. Usando la libreria la libreria Highcharts per Javascript che trovate a questo indirizzo: <https://www.highcharts.com/> proporre un grafico per visualizzare la classifica degli utenti della funzionalità *Visualizza Classifica Utenti*.

Per capire come utilizzare la libreria potete guardare, ad esempio, la pagina:

<https://www.highcharts.com/docs/getting-started/your-first-chart>

e gli esempi forniti come demo. Non è necessario installare la libreria sulla propria macchina. Fate attenzione alle condizioni di utilizzo che trovate qui:

<https://www.highcharts.com/blog/download/>