

Relazione progetto programmazione per il web 2021/2022

D'Ales Alessio
1903817

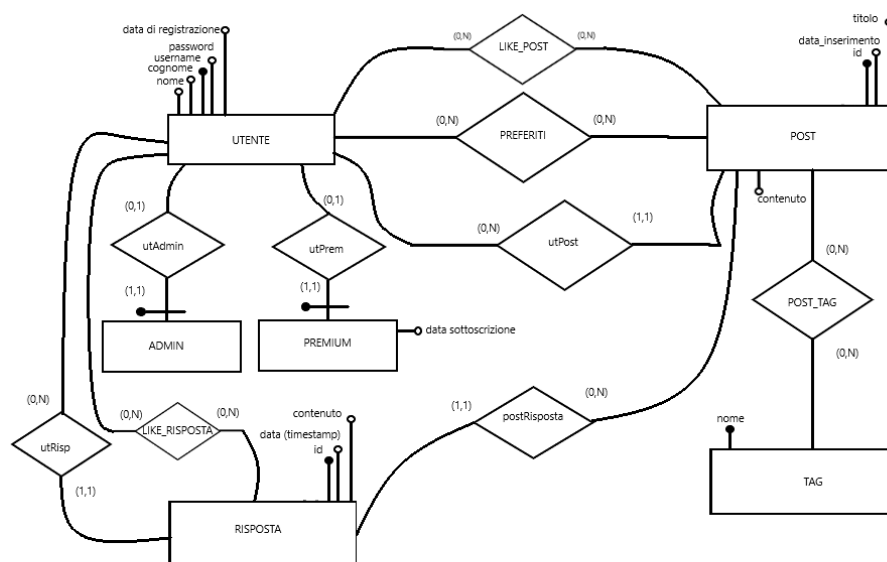
Albertini Tommaso
1893643

Iltchev Federico
1842707

1 Introduzione

OutOfMemory è un sito web in cui si possono fare domande riguardo a vasti argomenti di programmazione. Il sito fornisce la possibilità agli utenti di chiedere o rispondere a domande e di lasciare un like ai post oppure alle risposte più utili. Gli utenti hanno un profilo in cui possono visualizzare i like ricevuti e i post pubblicati.

2 Il diagramma ER



Il database si compone delle seguenti tabelle:

1. **UTENTE**

- 1.1. nome: mantenimento di informazioni base dell'utente (facoltativo)
- 1.2. cognome: mantenimento di informazioni base dell'utente (facoltativo)
- 1.3. username: chiave primaria , usato come riconoscimento utente univoco
- 1.4. data di registrazione
- 1.5. password: viene salvato l'hash della password ottenuto con BCrypt

2. **ADMIN**

- 2.1. username: FK per inserire un utente nella categoria "admin"

3. **PREMIUM**

- 3.1. data di sottoscrizione: viene salvata la data in cui l'utente diventa premium, per un possibile scenario in cui si può pagare per diventarlo
- 3.2. username: FK per inserire un utente nella categoria "premium"

4. **POST**

- 4.1. id: (serial) auto incrementale come chiave primaria per riconoscimento post univoco
- 4.2. premium: (booleano) se un post è rivolto ad utenti premium
- 4.3. creatore: FK per inserire l'username del creatore del post
- 4.4. data: (timestamp) che verrà usato per ordinare i post in base alla data
- 4.5. titolo
- 4.6. contenuto

5. **PREFERITI**

- 5.1. utente: FK per indicare un utente esistente
- 5.2. id post: FK per indicare un post esistente

6. **POSTTAG**

- 6.1. tag: FK per indicare un tag esistente
- 6.2. id post: FK per indicare un post esistente

7. **RISPOSTA**

- 7.1. id: (serial) auto incrementale come chiave primaria per riconoscimento risposta univoca

- 7.2. creatore: FK per inserire l'username del creatore della risposta
- 7.3. data: (timestamp) che verrà usato per ordinare le risposte in base alla data
- 7.4. contenuto

8. TAG

- 8.1. nome: per l'inserimento successivo di tag personalizzati

9. LIKEPOST

- 9.1. username
- 9.2. id post

10. LIKERISPOSTA

- 10.1. username
- 10.2. id risposta

3 Funzionalità

3.1 Il database

Il database si occupa di salvare tutte le informazioni che necessitano persistenza, come i dati degli utenti, dei post e delle risposte.

Tecnologia utilizzata: MYSQL

Risponde al requisito: Database degli utenti con diversi diritti di accesso, mantenimento di informazioni relative a un utente, Database di risorse caricabili dall'admin e accessibili agli user, eventualmente con diversi vincoli

3.2 Navigation bar e footer

Navigation bar e footer sono due pagine jsp incluse e persistenti in ogni pagina tramite azione standard include. Permettono la navigazione in tutto il sito e funzioni come la ricerca, il passaggio alla lingua inglese e il login.

Tecnologia utilizzata: JSP, azioni standard

Risponde al requisito: Presenza di pagine informative raggiungibili da ogni altra pagina

3.3 Login

La pagina di login permette l'inserimento di username e password con successivo riconoscimento dell'utente in caso di password corretta. Il meccanismo prevede il recupero dell'hash salvato nel database associato al nome utente che viene

confrontato con la password appena inserita tradotta in hash dall'algoritmo di hashing. Questa risorsa viene bloccata da un filtro nel caso in cui l'utente sia già entrato con le sue credenziali.

Un utente può essere riconosciuto come utente premium, come utente admin o come utente base. Un utente base visualizza solo i post base cioè inseriti da utenti base. Un utente premium può inserire post premium e può visualizzare tutti i post. Un utente admin può visualizzare tutto, inserire post premium e cancellare qualsiasi cosa (post e risposte)

Tecnologia utilizzata: Servlet, Filtri, JDBC, JSP, sessioni

Risponde al requisito: Database degli utenti con diversi diritti di accesso, Mantenimento di informazioni relative a un utente

3.4 Registrazione

La registrazione dell'utente prevede l'inserimento di due campi facoltativi (nome e cognome) e tre campi obbligatori, quali username e la coppia di password che deve combaciare. Se l'username scelto non esiste già, allora inserisce nel database tutti i dati immessi nel form; altrimenti restituisce una pagina di errore.

Tecnologia utilizzata: Servlet, Filtri, JDBC, sessioni

Risponde al requisito: Database degli utenti con diversi diritti di accesso, Mantenimento di informazioni relative a un utente

3.5 Traduzione

E' possibile visualizzare i testi statici in lingua italiana e inglese; la lingua corrente è memorizzata in un attributo di sessione. Cliccando sulla bandiera corrispondente nel footer viene impostato l'attributo di sessione e tutte le pagine useranno i testi tradotti in inglese.

Tecnologia utilizzata: JSP, sessioni, JSP taglib fmt e c, file properties (contiene i testi nelle varie lingue)

Risponde al requisito: Gestione della localizzazione

3.6 Bacheca post

La bacheca post visualizza tutti i post inseriti dagli utenti, divisi in tre categorie:
post risposti : hanno ricevuto almeno una risposta
post non risposti : non hanno ricevuto nessuna risposta
post premium : bloccati dall'accesso e dalla visualizzazione agli utenti base; questi post protetti possono servire in uno scenario tipo in cui un utente definito "esperto" vuole che a sua volta rispondano solo utenti "esperti"
Vengono visualizzati i tag associati al post e l'utente che lo ha creato

Ogni pagina genera al massimo 5 post, è possibile lo scorrimento delle pagine. Ogni post è un hyperlink al dettaglio post contenente la domanda e le relative risposte.

E' divisa in tre tab:

Tutti : visualizza tutti i post

Preferiti : visualizza solo i post contrassegnati come preferiti, uno scenario tipico è l'utente che vuole salvare uno specifico post per una successiva visualizzazione

Tag : Visualizza il risultato di una ricerca per titolo o per tag

Tecnologia utilizzata: JSP (visualizzazione dati), JDBC (recupero dati post), URL rewriting

Risponde al requisito: Database di risorse caricabili dall'admin e accessibili agli user, eventualmente con diversi vincoli, Categorizzazione delle risorse, user-generated content, con limitazioni su chi può inserire contenuti, Mantenimento di carrello e preferenze persistente

3.7 Inserimento post

Gli utenti possono scrivere post attraverso la compilazione di un form.

Il form da compilare è composto da 4 campi:

Titolo: il titolo del post

Contenuto: la domanda scritta dall'utente

Tags: il linguaggio utilizzato per il codice

Premium: visualizzato solo da utenti premium, permette di scegliere la visibilità della domanda

Tecnologia utilizzata: Servlet (inserimento nel DB), JDBC

Risponde al requisito: user-generated content, con limitazioni su chi può inserire contenuti

3.8 Rispondere ai post

Gli utenti che hanno effettuato il login possono rispondere a un post, tramite metodo post. Gli utenti premium sono gli unici che possono rispondere a tutti.

Tecnologia utilizzata: Servlet (inserimento risposta nel DB), JDBC

Risponde al requisito: user-generated content, con limitazioni su chi può inserire contenuti

3.9 Inserire un like ad un post o una risposta

Ogni post o risposta può ricevere un like dagli utenti, che si andranno a sommare al conto del creatore delle stesse. Questa funzionalità può servire in uno scenario tipo per far diventare un utente premium. Ad esempio raggiunti i 100 like, l'utente diventa "esperto" e ha una reputazione positiva.

Tecnologia utilizzata: Servlet, JDBC , URL rewriting
Risponde al requisito: Database degli utenti con diversi diritti di accesso

3.10 Aggiungere un post ai preferiti

Sui post vi è un campo "Aggiungi ai preferiti" in cui viene inserita una riga nella tabella preferiti. Viene anche controllato che non esista già.

Tecnologia utilizzata: Servlet, JSP, JDBC
Risponde al requisito: Mantenimento di preferenze persistenti

3.11 Rimuovere un post o una risposta

Una Servlet una volta premuto il tasto "Cancella Risposta" permette di rimuovere dal forum il post o la risposta ad un post.

Tecnologia utilizzata: Servlet, JDBC
Risponde al requisito: user-generated content, con limitazioni su chi può inserire contenuti

3.12 Scorrimento pagine nella bacheca

Gli utenti tramite il bottone, nell'URL viene salvato un attributo pagina in cui viene visualizzata solo quella pagina, e tramite URL rewriting, si viene reindirizzati alla pagine richiesta.

Tecnologia utilizzata: JSP, URL Rewriting, JDBC
Risponde al requisito: Presenza di pagine informative raggiungibili da ogni altra pagina

3.13 Ricerca e Tag

La barra di ricerca è implementata nella navbar e permette la ricerca per titolo e per tag I tag sono associati ad un post e inseriti durante la pubblicazione dello stesso. E' possibile filtrare per al più un tag e quindi visualizzare tutti i post associati Ogni qual volta viene cercato qualcosa, viene salvato un cookie, per mostrare le ricerche recenti.

Tecnologia utilizzata: JSP, JDBC, URL rewriting, cookies
Risponde al requisito: Possibilità di navigazione tramite riferimenti fra le risorse, Presenza di pagine informative raggiungibili da ogni altra pagina, Possibilità di raccomandazioni su criteri diretti (tag), Possibilità di navigazione tramite riferimenti fra le risorse

3.14 Profilo

La pagina del profilo permette di visualizzare nome: il nome e cognome facoltativi da inserire

username: nome utente con cui si viene identificati nel forum

post: numero post pubblicati

likes: numero like ricevuti

e da quanto fai parte della community

Tecnologia utilizzata: JSP, JDBC

Risponde al requisito: /

3.15 CSS

Per mantenere l'identità visiva del sito.

Tecnologia utilizzata: CSS, bootstrap, fontawesome icons

Risponde al requisito: Mantenimento di identità visiva dell'applicazione sulle varie pagine.