

22/11/2024. Prova in itinere di WSOS

Hai visto una super offerta on line ma puoi fare subito l'acquisto e temi di dimenticarlo ?
Non tenere! Abbiamo l'app che fa per te: WSOS!
Con WSOS puoi creare una nuova offerta da aggiungere alla tua wishlist, consultare l'elenco delle offerte visualizzando le offerte scadute/in scadenza, eliminare/aggiornare le offerte. Basta un semplice browser e la voglia di fare shopping!

WSOS utilizza una tabella "offers" avente i seguenti campi:

- Id: identificativo unico dell'offerta;
- Descrizione offerta;
- Prezzo: Valore decimale in Euro;
- Validità offerta: giorni di validità offerta (per evitare di usare date, assumiamo sia un campo calcolato);
- Acquistato: Indica se offerta è stata acquistata, default a 0 (false);

L'applicazione deve implementare le seguenti operazioni:

- a) Lista offerte: pagina web contenente la lista delle offerte con tutti i campi disponibili;
- b) Aggiungi offerta: pagina web con un form HTML per la creazione dell'offerta;

Inoltre l'applicazione deve dare la possibilità di:

- c) Eliminare offerta: pulsante/link nella lista per permettere di cancellare offerta;
- d) Aggiornare offerta: pulsante/link nella lista che permette di editare il campo prezzo, giorni validità, acquistato;

Bonus

Nel punto a)

- Evidenziare in rosso le offerte scadute (validità < 0), in giallo quelle in scadenza (validità tra 0 e 1 giorno)
- Offerta *all you can buy*: pulsante magico che permette di acquistare tutti i prodotti con lo sconto del 50%
- Visualizzare la somma del totale speso (prodotti acquistati)

Operazioni necessarie all'avvio:

- 1) *setup* del DB, da terminale nella home directory dell'utente:
`mysql -uuser -ppassword exam < wsos.sql`
- 2) Boilerplates: da terminale nella home directory dell'utente
`tar xzf wsos-boilerplates.tgz`
- 3) Nella directory del boilerplate scelto eseguire il codice per scaricare le dipendenze
`mvn jetty:run`

Tabella offers: a questo punto, grazie al *setup* appena eseguito, sarà presente, nel database **exam**, la tabella **offers** già popolata, avente schema SQL e dati di prova:

```
create table exam.offers (
  id          int auto_increment primary key,
  description  varchar(32) not null,
  price       decimal not null,
  validity    int ,
  purchased   smallint default 0
);

-- Insert example rows into the `exam.offers` table
INSERT INTO exam.offers (description, price, validity, purchased)
VALUES
('Smartwatch', 199.99, -5, 0),
('Noise Cancelling Headphones', 299.99, 365, 0),
('Gaming Console', 499.99, 1, 1),
('Personalized Necklace', 59.99, 0, 0),
('Electric Scooter', 349.99, 10, 1);
```

Info utili:

- Il driver mysql è `com.mysql.cj.jdbc.Driver`
- Il boilerplate contiene un `web.xml` da riempire con servlet name, class e mapping con url pattern, se si usano annotazioni e' preferibile cancellare il file
- Il file di persistence definisce
 - `jakarta.persistence.jdbc.url`
 - `jakarta.persistence.jdbc.user`
 - `jakarta.persistence.jdbc.password`