Progetto "DailyTicket" per Tecnologie Web (6 cfu)

Università degli Studi di Torino, A.A. 2020-2021

Lorenzo Botto - 882837

Relazione

Tema del sito e sezioni principali

DailyTicket è un sito di biglietteria per eventi e manifestazioni.
L'obiettivo del sito è di fornire all'utente, anche non registrato, una piattaforma dove è possibile consultare un catalogo contenente vari eventi, con la possibilità di acquistare dei biglietti, visualizzare un proprio carrello e controllare tutti gli ordini effettuati.
Il sito è suddiviso in queste sezioni principali:

- **Home**, contenente lo scopo del sito e una rapida descrizione dell'azienda.
- **Eventi**, in cui è presente tutto il catalogo degli eventi e delle manifestazioni dove l'utente può cercare ciò che desidera, attraverso una barra di ricerca. L'utente può decidere la quantità dei biglietti da acquistare, fino ad un massimo di 5, e controllare le informazioni riguardanti gli eventi.
- **Contatti**, contenente le informazioni per rivolgersi al servizio clienti e per instaurare partnership o rapporti di lavoro.
- **Carrello**, dove l'utente può visionare tutti i biglietti che ha aggiunto al carrello e che desidera acquistare.
- **Login o Registrazione/Sezione privata**, dove l'utente, se ha effettuato il login, avrà a disposizione una sezione privata in cui può vedere tutti gli ordini oppure effettuare il logout, altrimenti potrà registrarsi o fare il login.

Funzionalità

Login/Logout

La possibilità di effettuare login/logout è sempre disponibile nel menu di navigazione situato nell'intestazione (oppure a lato, se la finestra è di dimensioni minori).
Nel caso in cui la finestra sia grande si può effettuare il login nella stessa pagina dove l'utente è situato, attraverso degli input ed un pulsante. Altrimenti se la schermata è di dimensioni minori, l'utente, verrà reindirizzato in una pagina apposita per il login.
Anche il logout è sempre presente nel menù di navigazione situato nell'intestazione, cliccando sulla propria sezione privata. Altrimenti nel menù a lato se la finestra è ridimensionata.
Entrambe le funzioni sono disponibili nel piè di pagina.

Registrazione

La possibilità di effettuare la registrazione è sempre disponibile nel menu di navigazione situato nell'intestazione (oppure a lato, se la finestra è di dimensioni minori).
Nel caso in cui la finestra sia grande si può effettuare la registrazione nella stessa pagina dove l'utente è situato, attraverso degli input ed un pulsante. Altrimenti se la schermata è di dimensioni minori, l'utente, verrà reindirizzato in una pagina apposita per la registrazione. La funzione è anche disponibile nel piè di pagina.

Gestione del contenuto generato dall'utente

L'utente ha a disposizione nel menù di navigazione una barra di ricerca per cercare gli eventi per nome. Quando preme sull'icona di ricerca, viene indirizzato alla pagina degli eventi e può visualizzare, sotto la barra di ricerca per categorie e data, tutti gli eventi che iniziano con il nome che ha digitato. L'utente ha anche a disposizione un piè di pagina, dove ci sono quattro link alle quattro categorie degli eventi, che se cliccati, reindirizzano alla pagina degli eventi, con l'elenco a discesa della categoria già selezionato e tutti i risultati degli eventi per quella categoria. L'utente, nella pagina degli eventi, ha a disposizione una barra di ricerca per categoria, sottocategoria e data, dove può ricercare gli eventi che verranno visualizzati con le loro informazioni. Questi eventi generati dall'utente, potranno essere aggiunti al carrello, modificandone anche la quantità (fino ad un massimo di 5). Così l'utente può generare un proprio carrello con tutti i biglietti che vuole acquistare, che può visualizzare nella pagina dedicata al carrello.
Nel carrello, si potranno rivedere i biglietti che sono stati aggiunti, con la loro relativa quantità (modificabile) e con la possibilità di rimuovere tutti i biglietti di un evento dal carrello.

Si può anche visualizzare il numero totale dei biglietti ed il prezzo totale, accompagnati da due pulsanti per svuotare completamente il carrello e completare l'acquisto (solo per l'utente che ha effettuato il login). L'utente connesso ha anche a disposizione una pagina con tutti gli ordini che ha effettuato, con il prezzo totale e la data dell'ordine.

Caratteristiche

Usabilità

Nella pagina degli eventi, quando l'utente seleziona una categoria, anche l'elenco a discesa della sottocategoria e i selettori delle date appaiono, così l'utente può decidere se cercare solo con la categoria o anche con gli altri filtri. Verranno visualizzati tutti gli eventi con i filtri selezionati dall'utente.
Quando l'utente preme sul pulsante "Aggiungi al carrello", viene rilasciato un riscontro sotto al pulsante se l'operazione è andata a buon fine oppure no. Nella pagina del carrello, quando l'utente preme sul pulsante "Rimuovi dal carrello", il biglietto viene eliminato dalla lista dei biglietti presenti nel carrello.
Quando l'utente preme sul pulsante "Svuota carrello", non viene visualizzata più la lista dei biglietti ed il totale del prezzo, ma viene segnalato all'utente che il carrello è vuoto, con un pulsante per essere reindirizzati alla pagina degli eventi. Quando l'utente preme sul pulsante "Acquista", se l'operazione va a buon fine, non viene visualizzata più la lista dei biglietti ed il totale del prezzo, ma viene segnalato all'utente che l'acquisto è stato completato, altrimenti viene visualizzato l'errore sotto il pulsante. Quando l'utente effettua il login o la registrazione vengono controllati gli input e viene rilasciato un riscontro in caso di errore, sotto al medesimo campo.

Interazione/Animazione

Quando l'utente visualizza la lista degli eventi, può effettuare un drag and drop all'interno del carrello.
L'utente può tenere premuto sulla sezione che delimita il biglietto e trascinarlo sull'icona del carrello, che, in questo modo, verrà aggiunto al carrello in automatico. I feedback sono gli stessi di quando si preme sul pulsante, ma se il biglietto che è stato trascinato non si trova più nella schermata (perché era più in basso), viene fatto uno scroll automatico fino all'altezza del biglietto, in modo da visualizzare il feedback.

Sessioni

La sessione viene iniziata per la prima volta quando si accede alla pagina di Home. La sessione è aperta ogni volta che si carica "header.php", il menù di navigazione in alto, il quale dovrà fare uso della variabile di sessione dell'utente. Così la sessione sarà disponibile anche per il "footer.php", e alle pagine che servirà l'utilizzo di una variabile di sessione.
Una volta effettuato il logout, la sessione viene distrutta e viene rigenerato l'id per una prossima sessione.

Interrogazione del db

Il database viene interrogato quando:

- Si effettua il login. Viene ricercato se l'utente è registrato nella tabella 'utenti'.
- Si effettua una registrazione. Viene richiesto l'username e l'e-mail per i controlli sugli input dalla tabella 'utenti'.
- Vengono ricercati degli eventi. Si accede alla tabella 'eventi'.
- Un'utente ha effettuato il login e accede al suo carrello. Viene richiamata la tabella 'carrello', invece quando accede ai biglietti acquistati, ovvero lo storico degli ordini, viene interrogata la tabella 'ordini'.
- Viene concluso un acquisto. Si accede alla tabella 'carrello'.
- Un'utente ha effettuato il login e vuole aggiungere un biglietto al carrello. Viene recuperata la quantità dalla tabella 'carrello'.
- Viene caricato il carrello. Si accede alla tabella 'eventi' per recuperare le informazioni e, se l'utente ha effettuato il login, si accede alla tabella 'carrello'.
- Viene cambiata una quantità dal carrello. Si richiede il prezzo e la quantità dalle tabelle 'eventi' e 'carrello', per il calcolo del prezzo totale (la quantità solo per l'utente che ha effettuato il login).
- Viene eliminato un biglietto dal carrello di un'utente che ha effettuato il login. Si interroga la tabella 'carrello' per il numero totale di biglietti presenti nel carrello.

Validazione dei dati di input

- **Lato client:**
  Quando un utente cerca di effettuare il login, i dati in input vengono validati controllando che non siano vuoti e nel caso viene restituito un errore sotto al campo. Invece se un utente cerca di effettuare una registrazione, vengono controllati che non siano vuoti e si effettua anche un controllo che l'e-mail sia in un formato valido attraverso un'espressione regolare e nel caso vengono restituiti gli errori sotto al

campo. Nel carrello si controlla anche che la quantità sia un numero da 1 a 5, con un'espressione regolare, restituendo l'opportuno errore.

- **Lato server:**
  Ogni volta che come parametro si passa una quantità, si effettua un ulteriore controllo che sia un numero da 1 a 5, con un'espressione regolare. Quando si ricerca un evento tramite la barra di ricerca delle categorie e della data, vengono controllati i parametri delle categorie che siano delle stringhe e che le date siano nel formato valido, attraverso delle espressioni regolari. Quando si effettua il login, vengono sanificate le stringhe dell'username e della password attraverso un filtro che elimina i caratteri dannosi. Quando si effettua la registrazione, viene validata l'e-mail attraverso un filtro e vengono sanificate le stringhe dell'username e della password attraverso un filtro che elimina i caratteri dannosi (restituendo gli opportuni errori). Quando si effettua una ricerca dalla barra presente nel menù di navigazione, la stringa viene sanificata attraverso un filtro che elimina i caratteri dannosi.

Sicurezza

Tutti i dati che provengono da input, sono validati o sanificati con gli opportuni filtri o espressioni regolari per catturare tutti i caratteri dannosi che possono essere inseriti.
Quando si effettuano interrogazioni/inserimenti/eliminazioni all'interno del database, si utilizza la funzione 'quote' di php che sostituisce le stringhe con stringhe equivalenti che usano sequenze di escape.

Presentazione

Lo stile del sito è effettuato secondo uno standard simile per tutte le pagine, con un'intestazione ed un piè di pagina sempre presenti ed integrati da un "top.html". L'intestazione ed il piè di pagina sono responsive, in quanto vengono utilizzate delle flexbox in cui i vari elementi si ridimensionano e si spostano l'uno sotto l'altro quando la schermata si ridimensiona. L'intestazione, quando viene ridimensionata la schermata, si modifica e viene visualizzato un pulsante per aprire un menù laterale. Anche il corpo centrale della pagina è effettuato secondo il metodo delle flexbox, in modo che quando la pagina viene ridimensionata, gli elementi si spostano uno sotto l'altro e si ridimensionano a seconda della grandezza della finestra. I colori delle varie sezioni sono in contrasto tra di loro, in modo da dare una chiara separazione delle varie sezioni, anche per quanto riguarda il contenuto generato dall'utente. Il font è identico per tutte le pagine e i colori dei font variano a seconda della sezione, in modo da essere leggibili. Alcune sezioni sono più strette rispetto alle altre, per garantire una maggiore leggibilità. Per ogni contenuto è stato definito il margine e il padding opportuno, in modo che non si sovrapponga ad altro contenuto e per essere allineato in modo che sia leggibile.

Front-end

Separazione presentazione/contenuto/comportamento (stile unobtrusive)

La separazione è stata effettuata in modo completo, in quanto la presentazione (CSS) è separata dal contenuto (HTML) di tutte le pagine, includendo il file "style.css", dove c'è la presentazione per tutte le pagine HTML. Anche il comportamento (JS) è stato diviso, in quanto tutte le pagine hanno un proprio file js, che verrà incluso nel contenuto della pagina. In più è presente un file js comune a più pagine. È stata utilizzata la libreria jQuery.

Soluzione cross-platform

Il sito è usufruibile da PC.

Organizzazione file e cartelle di progetto

All'interno della cartella "ProgettoTWeb", ci sono cinque cartelle che suddividono tutto il materiale:

- Css, dove è presente il foglio di stile "style.css"
- Html, dove sono presenti tutte le pagine che contengono HTML.
- Img, dove sono presenti tutte le immagini del sito.
- Js, dove sono presenti tutti i file javascript.
- Php, dove sono presenti, sotto opportune cartelle, tutti gli script esclusivamente PHP.

Soluzioni html/css/javascript degne di nota

- Quando l'utente genera gli eventi che desidera ricercare, viene ritornato un oggetto JSON alla chiamata ajax nel file js. La funzione 'listaEventi', nel file js, crea tutte le sezioni del biglietto. L'ultima sezione, dove è presente il pulsante di aggiunta al carrello, è stata creata in modo che la funzione effettui un loop sull'oggetto JSON, definendo l'evento del click di ogni singolo pulsante in modo dinamico passando i parametri opportuni ad un'altra funzione. Ciò viene fatto per ogni singolo biglietto e ogni singolo pulsante. In questo modo viene inserito il biglietto corretto nel carrello. Successivamente crea anche un'intestazione della sezione dei biglietti, con i filtri che sono stati scelti e definisce il drag and drop.

- Login, registrazione e sezione privata dell'utente (a schermata intera) sono stati creati, con il menu a discesa, in modo che quando l'utente clicca su uno di questi link, si rendono visibili due sezioni che precedentemente erano invisibili. Una è la sezione in cui è disegnata la freccia, che è stata possibile disegnarla grazie ai bordi nel file css, e l'altra è la sezione dove contiene tutti gli input e il pulsante. In questo modo l'utente ha a disposizione queste funzionalità nella stessa pagina.
- Tutti gli input che sono delle liste di selezione, sono stati personalizzati in modo da renderle, come stile, in linea con il sito. Questo è stato reso possibile modificando lo stile dei select, ma soprattutto è stata creata l'icona con le due frecce, sfruttando i selettori '::before' e '::after' e definendo i bordi all'interno dei selettori, nel file css.
  Infine è stata modificata anche la loro posizione in modo da allinearli correttamente.

Back-end e comunicazione front/back-end

Architettura generale classi/funzioni php

Le funzioni PHP sono organizzate in cartelle differenti in base alle loro logiche funzionali.

Schema del db

Tabella	Colonne
Carrello	idEvento, quantita, username, data
Eventi	id, titolo, descrizione, data, prezzo, categoria, sottocategoria, citta, luogo, ora, url
Ordini	id, idEvento, quantita, username, data
Utenti	email, username, password

Descrizione delle funzioni remote

- **addAcquisto.php,** inserisce tutti i biglietti del carrello nella tabella ordini, ovvero completa l'acquisto. La funzione è richiamata senza parametri e in caso di successo viene resa invisibile la sezione dei biglietti e del prezzo in modo da visualizzare un feedback sull'acquisto completato. Successivamente viene svuotato il carrello.
- **addCartNoLogin.php,** quando l'utente effettua il login, inserisce tutti i biglietti del carrello all'interno del database, per l'utente connesso. La funzione è richiamata senza parametri e in caso di successo l'utente è reindirizzato alla homepage.
- **addToCart.php,** inserisce un biglietto all'interno del carrello. Se l'utente non ha effettuato il login, utilizzerà una variabile array di sessione, altrimenti il database. La funzione è richiamata con il metodo "POST" in cui vengono passati come parametri l'ID dell'evento e la quantità. La funzione restituisce 0 in caso sia raggiunto il limite di 5 biglietti, 1 in caso di successo e una stringa di errore sulla quantità nel caso non sia in un formato valido. In caso di successo, a seconda della risposta, viene visualizzato un feedback sotto il pulsante premuto per aggiungere il biglietto al carrello e, se aggiunto, modificato il numero di biglietti presenti nel carrello.
- **changeQuantity.php,** quando l'utente modifica una quantità nel carrello, questa funzione modifica la quantità nell'array di sessione, se l'utente non è connesso, oppure nel database. La funzione è richiamata con il metodo "POST" in cui viene passato il parametro con l'ID dell'evento e la quantità. La funzione restituisce una stringa di errore altrimenti nulla. In caso di successo, se la risposta è una stringa, viene visualizzato l'errore e disabilitato il pulsante per concludere l'ordine, altrimenti viene richiamata una funzione per aggiornare il dettaglio sui prezzi.
- **common.php,** contiene le funzioni php comuni a tutti gli altri script.
- **controlSearchEvento.php,** controlla se è definita la variabile di sessione 'searchEvento', ovvero se l'utente ha ricercato tramite la barra di ricerca nell'intestazione. La funzione è richiamata senza parametri. Restituisce la stringa con ciò che ha digitato l'utente, se è definita, altrimenti 0. In caso di successo, viene definita una variabile globale e richiamata una funzione che visualizzerà i risultati.
- **getBigliettiAcquistati.php,** recupera tutti gli ordini di un utente. La funzione è richiamata senza parametri. Restituisce un oggetto JSON con sintassi:

```
{
 "ordini": [
  {
   "ordine": [
        {"id": "1", "quantita": "1", "dataOrdine": "2021-01-03", "titolo": "Pinguini Tattici Nucleari",
         "descrizione": "", "data": "2021-10-04", "prezzo": "39,10 €", "citta": "Torino", "luogo": "Pala
         Alpitour", "ora": "21:00", "url": "../img/pinguini.jpg"}
   ]
  }
```

}
In caso di successo, richiamerà una funzione per leggere l'oggetto JSON e visualizzare tutti gli ordini.

- **getCarrello.php,** recupera tutti i biglietti del carrello di un utente. La funzione è richiamata senza parametri. Restituisce un oggetto JSON con sintassi:

```
{
  "carrello": [
    {"id": "3", "titolo": "Pinguini Tattici Nucleari", "descrizione": "", "data": "2021-10-04", "prezzo":
     "39,10 €", "citta": "Torino", "luogo": "Pala Alpitour", "ora": "21:00", "url": "../img/pinguini.jpg"}
  ]
}
```

In caso di successo, richiamerà una funzione per leggere l'oggetto JSON e visualizzare tutti i biglietti, nonché il prezzo totale.
- **getList.php,** recupera tutti gli eventi cercati da un utente. La funzione è richiamata con il metodo "POST" in cui vengono passati i parametri della categoria, sottocategoria (se selezionata) e le due date (se selezionate). Restituisce un oggetto JSON con sintassi:

```
{
  "eventi": [
    {"id": "3", "titolo": "Pinguini Tattici Nucleari","descrizione": "", "data": "2021-10-04", "prezzo": "39,10 €",
"citta":
      "Torino", "luogo": "Pala Alpitour", "ora": "21:00", "url": "../img/pinguini.jpg"}
  ]
}
```

Nel caso di errori nei parametri passati, perché non validi, viene restituito un oggetto JSON vuoto:

```
{
  "eventi": [
  ]
}
```

In caso di successo, richiamerà una funzione per visualizzare tutti gli eventi ricercati.
- **getUser.php,** controlla se l'utente esiste. La funzione è richiamata con il metodo "POST" in cui vengono passati come parametri l'username e la password. La funzione ritorna 1, se l'utente esiste, altrimenti 0. In caso di successo, se la risposta è 1 richiamerà una funzione che farà l'aggiunta del carrello all'utente appena connesso e reindirizzerà alla home, altrimenti viene rilasciato un riscontro di errore.
- **logout.php,** distrugge la sessione corrente, rigenera l'ID di sessione e reindirizza alla home.
- **refreshQuantity.php,** viene utilizzata quando si modifica la quantità nel carrello e quando viene caricato il carrello, per aggiornare il dettaglio sui prezzi e selezionare le quantità dei biglietti. La funzione è chiamata senza parametri. Restituisce un oggetto JSON con sintassi:

```
{
  "totale": [
    {"prezzo": "78,20", "quantita": "2", "quantitaSelect": "2"}
  ]
}
```

In caso di successo, richiamerà una funzione che leggerà l'oggetto JSON e modificherà il prezzo totale, il numero totale di biglietti e selezionerà le quantità dei biglietti.
- **register.php,** effettua la registrazione di un nuovo utente. La funzione è richiamata con il metodo "POST" in cui vengono passati i parametri dell'e-mail, username e password. In caso di input invalidi ritornerà una stringa con l'errore, altrimenti 1. In caso di successo, se la risposta è una stringa, vengono suddivisi i casi di errore e visualizzati come feedback all'utente, altrimenti se la risposta è 1 viene effettuato il login automatico richiamando "getUser.php".
- **removeFromCart.php,** effettua la rimozione di un biglietto (o di tutti i biglietti) dal carrello. La funzione è chiamata con il metodo "POST" in cui viene passato come parametro l'ID dell'evento, se si rimuove solo un biglietto, invece se si svuota il carrello viene passato un parametro 'all=true'. La funzione ritorna il numero di biglietti presenti nel carrello. In caso di successo, viene richiamata una funzione che aggiornerà il carrello e il numero degli elementi nel carrello.
- **searchEvento.php,** definisce una variabile di sessione per un evento ricercato dalla barra di ricerca nell'intestazione. La funzione viene richiamata con metodo "POST" in cui viene passato come parametro il nome dell'evento ricercato. In caso di successo si viene reindirizzati alla pagina degli eventi, a meno che non si è già sulla pagina e quindi vengono visualizzati i risultati.
- **selectedOption.php,** definisce una variabile di sessione con codice HTML per selezionare l'elenco a discesa della categoria in automatico (dovuto ad un click sulla categoria nel piè di pagina). La funzione viene richiamata con metodo "POST" in cui viene passato come parametro il codice HTML per la select. In caso di successo, si viene reindirizzati alla pagina degli eventi con l'elenco a discesa già selezionato sulla categoria precedentemente cliccata e i risultati degli eventi per quella categoria.