

# Relazione PROGETTO TECNOLOGIE WEB

2022/2023

Liquindoli Michele

## TEMA DEL SITO:

La pagina è a Tema Super Mario e il mondo che lo circonda.

In particolare è uno shop online che permette agli utenti di acquistare i prodotti che vede nella homepage.

Il sito web si presenta con un indirizzamento alla pagina di Login in cui vi sono 2 bottoni, uno per la registrazione degli utenti comuni e uno per la registrazione per gli amministratori.

Una volta entrato nella homepage, la barra di navigazione superiore abilita diverse sezioni tra cui, il logo, Shop, About us, Contatcts, un Icona che si riferisce al carrello, il profilo e il tasto di logout.

Molti di questi tasti nella navbar sono delle ancore ad elementi che si trovano nella homepage.php, mentre il carrello e il profilo ridirigono ad una pagina specifica per i suoi dati. Il tasto di Logout semplicemente effettua la disconnessione dell'account.

## FUNZIONALITÀ DEL SITO:

Generale:

Le pagine del sito web *necessitano la il login prima di essere visualizzate.*

Nel caso in cui, un utente che non ha fatto accesso, provi ad accedere a queste pagine mediante l'inserimento del loro nome mediante URL, si viene lo stesso ridiretti nella sezione del login.

Una scelta implementativa del sito è che alla creazione dell'amministratore, ne viene creato un account associato di tipo cliente, in modo tale da permette anche all'amministratore di acquistare prodotti tramite il suo sito web.

L'amministratore, se acceduto mediante il suo account dipendete, avrà la sezione "profilo" con le diverse funzioni di gestione del sito web come: Aggiungere un amministratore, aggiungere/rimuovere/modificare un prodotto.

Sign in e Login:

Se non abbiamo un account ci troveremo davanti alla pagina di login.

In questa pagina troveremo un semplice form che richiede le credenziali di accesso per il sito web.

Se non si dispone di un account, si può cliccare il tasto "Sign in" per essere indirizzato ad una pagina dedicata alla registrazione, il quale al click del tasto "register" creerà un account per l'utente.

Entrambe le sezioni, di login e sign in, sono gestite mediante controlli per i campi da compilare client-side, ad esempio il controllo che venga inserita una email nella forma corretta o i requisiti minimi per una password sicura.

Nel caso in cui qualche dato non sia inserito un'apposita notifica sarà visualizzata e non si procederà all'invio dei dati al server.

Nel momento in cui ci si riesce a registrare correttamente, viene mostrato un piccolo messaggio e subito dopo si viene ridiretti alla sezione login.

Nel login vi sono disponibili 2 tasti: Login e Login Amministratore.

Anche qui se l'accesso va a buon fine viene mostrato un messaggio e si viene ridiretti nella homepage.

La sessione di quel determinato utente viene impostata quando le credenziali del login.

Il click del tasto Logout, segna la chiusura della sessione, da cui si viene ridiretti nella pagina di login, notificando l'utente con un messaggio nel mentre che la pagina di login viene caricata.

Gestione del contenuto generato dall'utente:

L'utente già nella homepage può generare contenuto attraverso la sezione "shop" che presenta, sotto lo slide show, una barra orizzontale con dei tasti cliccabili, che rappresentano le categorie.

Questi tasti permettono di modificare il contenuto della sezione sottostante, che è dedicata alle card dei prodotti.

Al click della categoria vengono caricati dinamicamente i prodotti, sotto forma di card che ha la struttura di: foto del prodotto, il suo titolo, la sua descrizione e un tasto per aggiungerlo al carrello. Questo viene effettuato mediante una richiesta asincrona Ajax che interroga il database restituendo tutti i prodotti che fanno parte di quella determinata categoria. Lo stesso accade quando si vuole accedere al carrello.

## FRONT-END:

Organizzazione file e cartelle:

Per organizzare al meglio le componenti del sito web, ho optato un'organizzazione dei file in cartelle specifiche in base alla loro funzione così da evitare lo stile intrusivo.

Sono presenti 5 cartelle:

- Assets, contiene altre sottocartelle organizzative con le immagini e icone che sono utilizzare nel sito web
- Css: Sono presenti tutti i fogli di stile, anche qui organizzati per nome riferendosi a ciò che vanno a stilizzare (ad esempio troveremo un foglio di stile "top.css" che si riferisce al "top.html"), questi rendono anche abbastanza responsive il sito web attraverso le media query.
- Html: contiene i file puramente html (in alcuni di essi ci sono dei piccoli frammenti di php che servono necessariamente al controllo della sessione corrente)
- Js: contiene tutti i file javascript che regolano il comportamento della pagina e che permettono di renderlo un sito web dinamico
- Php: Qui troveremo i file con estensione ".php" i quali, alcuni di essi si riferiscono alle pagine che verranno mostrate nel browser e aggiornate con il contenuto dinamico (ad esempio "homepage.php") ma anche i file che processano le richieste asincrone effettuate dai file javascript per la richiesta di dati dal database e per i controlli server-side.

Cross-Plattform:

Il sito web funziona anche su altri browser, ovviamente con leggere differenze estetiche che non cambiano il risultato finale e non alterano la navigazione accessibile per l'utente.

Inoltre si è riusciti a rendere il sito web responsive, dunque abbastanza godibile ed usabile sui browser di schermi di dimensioni diverse a quelli dei classici pc.

Soluzioni degne di nota:

Delle soluzioni degne di nota possono essere la gestione dei prodotti mostrati.

Inizialmente si vede una sezione che contiene diversi tasti che rappresentano le categorie di prodotti che sono presenti nel database. Queste categorie possono essere più di quelle che sono mostrate nella sezione, dunque attraverso delle proprietà Css (overflow-x), questa sezione delle categorie presenta una *scrollbar* che permette di mostrare anche le categorie nascoste aiutandosi con la barra di scorrimento. Ad ogni categoria cliccata infine viene aggiornato il contenuto della sezione sottostante, che inizialmente contiene tutti i prodotti disordinati, con i prodotti che fanno parte di quella categoria.

## COMUNICAZIONE FRONT-END E BACK-END:

Architettura delle funzioni:

In generale, i nomi dei file php sono il più significativi possibile su cosa gestiscono. Infatti ho optato per creare più file php che possano essere chiamati singolarmente

dalle richieste asincrone di Ajax in modo tale ognuna di esse gestisca solo una situazione. (ad esempio: "get\_data\_from\_cart.php").

In database è formato da 4 tabelle:

- **Cliente:** Contiene i dati che sono stati inseriti al momento della registrazione del cliente
  - Nome
  - Cognome
  - Email → chiave primaria
  - Password
- **Dipendente:** Contiene i dati che sono stati inseriti al momento della registrazione del dipendente/amministratore
  - Nome
  - Cognome
  - Email → chiave primaria
  - Password
- **Carrello:** Contiene i dati associati tra cliente e Titolo del prodotto, con la quantità acquistata e la somma del costo unitario per la quantità
  - Cliente → Chiave primaria Composta con "Titolo"
  - Titolo → Chiave primaria Composta con "Cliente"; → chiave esterna su "Prodotti.Titolo"
  - Quantita
  - Somma
- **Prodotti:** Contiene i dati che definiscono un particolare prodotto
  - Categoria
  - Titolo → chiave primaria
  - Url\_foto
  - Descrizione
  - Prezzo

Descrizione funzioni remote:

Le funzioni remote che sono utilizzate restituiscono i dati strutturati in json quando si tratta di dati che verranno utilizzati per creare contenuto dinamico. Questo avviene attraverso la codifica mediante la funzione `json_encode`, su ciò che vogliamo ritornare.

Quasi tutte le funzioni remote restituiscono il risultato della query fatta sul DB, codificata in json. Dato che le query vengono ricavate in forma di array associativo, l'intero array viene convertito in json e restituito, dunque sta all'utilizzatore effettivo dei dati ricevuti a gestire l'accesso ai campi corretti.

Da diverse funzioni sono utilizzati diversi metodi di invio (POST / GET) dei dati al server, la scelta è stata a discrezione del pericolo che si poteva correre nel mostrare i dati inviati nell'URL oppure no.

Non tutte le richieste ajax necessariamente trasmettono dei dati al server, anzi spesso vengono fatte chiamate solo per ricevere dati dal database, ma la maggior parte delle funzioni che inviano dei dati lo fanno attraverso il formato json.

Una possibile sintassi del valore restituito può avere forma:

```
{
  [
    {"Categoria":"Bowser"},
    {"Categoria":"Daisy"},
    {"Categoria":"Donkey Kong"},
    {"Categoria":"Goomba"},
    {"Categoria":"Luigi"},
    {"Categoria":"Mario"},
    {"Categoria":"Peach"},
    {"Categoria":"Poster"},
    {"Categoria":"Toad"},
    {"Categoria":"Toadette"},
    {"Categoria":"Wario"},
    {"Categoria":"Yoshi"}
  ]
}
```

Eventuali condizioni di errore della richiesta Ajax vengono gestite con una funzione apposita chiamata `ajaxfail`, mentre eventuali valori di ritorno non corretti vengo gestiti parte "success" di Ajax, la quale si occuperà di mostrare un messaggio relativo all'errore ottenuto. Ad esempio Viene mostrato un errore con un messaggio descrittivo se cerco di effettuare l'accesso con dei dati che superano i controlli client-side, ma non quelli server-side che riguardano controlli sul database, dunque con un account che non è registrato nel database.