

# Progetto Tecnologie Web

Matteo Lotto, Riccardo Tassetto, Davide Zago

## Indice

<b>1</b>	<b>Informazioni</b>	<b>2</b>
<b>2</b>	<b>Abstract</b>	<b>3</b>
<b>3</b>	<b>Utenti destinatari</b>	<b>3</b>
<b>4</b>	<b>Realizzazione</b>	<b>4</b>
4.1	Struttura . . . . .	4
4.2	Presentazione . . . . .	6
4.3	Comportamento . . . . .	7
4.3.1	PHP . . . . .	7
4.3.1.1	Gestione utente . . . . .	9
4.3.2	Javascript . . . . .	9
4.3.3	Database MySQL . . . . .	9
<b>5</b>	<b>Validazione</b>	<b>11</b>
<b>6</b>	<b>Ruoli</b>	<b>12</b>

# 1 Informazioni

il progetto è disponibile sul server del corso all' indirizzo tecweb2016/  
L'account amministratore è:

- **Utente:** admin
- **Password:** admin

Un account utente è:

- **Utente:** user
- **Password:** user

è comunque possibile registrarsi all'interno del sito.

**Referente:** davide.zago.5@studenti.unipd.it

## 2 Abstract

Il progetto si propone di realizzare il sito web di un rivenditore di componenti per PC. Lo scopo principale è quello di fornire ai clienti una interfaccia dalla quale possono visualizzare e ordinare i pezzi che vogliono acquistare, il sito quindi offre la classica funzionalità di *carrello*, dove l'utente può inserire i componenti che vorrebbe acquistare, inoltre per ogni utente è disponibile la storia degli acquisti effettuati, da questa lista l'utente è in grado di dare una valutazione e di scrivere una recensione sul prodotto. Un admin può aggiungere e rimuovere oggetti dal catalogo, e creare dei Bundle che verranno visualizzati nella Home.

## 3 Utenti destinatari

I clienti del negozio sono generalmente persone già informate su cosa vogliono acquistare e sulla funzione dei vari componenti, perciò vengono fornite le caratteristiche tecniche di ogni pezzo, gli utenti possono inoltre fare ricerche mirate sul nome di un componente oppure tramite una ricerca avanzata possono selezionare un tipo specifico di componenti (RAM, Processore, ecc...). Per non lasciare fuori gli utenti meno esperti forniamo nella home page una serie di computer preassemblati su diverse fasce di prezzo con una semplice descrizione che aiuti i clienti meno esperti a capire se una data macchina soddisfa o meno i suoi bisogni.

## 4 Realizzazione

### 4.1 Struttura

La pagina e' stata scritta in HTML 5.

Il sito e' composto da pagine php, che presentano dati e interazioni diversi a seconda del tipo di utente o delle interazioni svolte. Le nostre pagine sono:

- **home.php:** introduce gli utenti alle proposte del sito, mostrando i prodotti che sono stati più venduti nel sito, e i Bundle, insiemi di prodotti preparati dagli amministratori.
- **prodotti.php:** mostra una lista dei prodotti che sono disponibili nel sito mostrandone i vari dati, tali prodotti possono essere aggiunti al carrello se l'utente ha effettuato la login. A questa pagina si arriva anche usando la barra di ricerca in alto a destra, che porta i risultati della ricerca.
- **productdetails.php:** si accede a questa pagina cliccando sul nome di un prodotto, che ne mostra i dati in modo più dettagliato, e visualizza le recensioni lasciate dagli utenti. Inoltre contiene un pulsante per aggiungere il prodotto al carrello dell'utente (se anonimo, tale pulsante non compare).
- **carrello.php:** in questa pagina vengono visualizzati i prodotti che un utente ha aggiunto al proprio carrello, e permette la conferma degli acquisti.
- **account.php:** serve a mostrare i dati relativi ad un account, e contiene il link alla pagina *purchasehistory.php*. Tale pagina mostra tutti gli acquisti precedenti effettuati dall'utente, per un facile ricontrollo.
- **recensione.php:** è possibile accedere a questa pagina da *purchasehistory.php*, da qui gli utenti possono inserire una recensione e una valutazione di un prodotto che hanno già acquistato.
- **bundledetails.php:** mostra i dettagli dei bundle presenti nella home, insieme alla lista dei prodotti di cui è composto. La pagina permette la facile aggiunta di tutti i prodotti del bundle dentro il carrello.
- **register.php:** permette la creazione di un nuovo utente, con le i dati che vengono controllati da javascript e, eventualmente, da php.
- **login.php:** permette agli utenti di connettersi al sito. Se si connette un amministratore, viene reindirizzato alla pagina *adminmenu.php*.

- **adminmenu.php:** questa pagina presenta i link a tutte i tool di amministrazione. Questa e le pagine per gli admin, reindirizzano ad home qualora un utente non admin vi acceda.
- **adminproducts.php:** permette la creazione di nuovi prodotti, o la modifica di esistenti, tramite la pagina *modificaProd.php*.
- **adminaccounts.php:** permette un accesso alla lista di tutti gli utenti del sito, con la possibilità di promuovere o degradare gli altri utenti.  
**adminbundle.php** in questa pagina è possibile creare nuovi bundle o modificarne di esistenti, inserendo quali prodotti ne fanno parte.
- **adminpurchasehist.php:** presenta una lista di tutti gli acquisti che sono stati effettuati da tutti gli utenti del sito, per aiutare in caso di problemi con le transazioni.

La pagina 404.php serve a reindirizzare qualora si cerca di entrare in cartelle del sito, quale quella delle immagini:un index.php reindirizza l'utente a questa pagina.

Ogni pagina presenta la barra di navigazione nella parte superiore. Tale barra risulta diversa qualora il sito lo visita un utente anonimo per il quale appare l'opzione di effettuare il login. Un utente registrato, che ha accesso ad account.php e veloce accesso al carrello, o un amministratore, che ottiene accesso alla gestione del sito (adminmenu.php), alla pagina del proprio account,e relativi servizi da utente, e qualunque utente ha accesso al link di Logout.

## 4.2 Presentazione

Nel realizzare l'interfaccia grafica del sito è stato usato lo standard CSS3. I diversi file .css utilizzati ci permettono di realizzare una formattazione adatta per la visione del sito da un desktop, un mobile e per un layout di stampa. I file utilizzati sono rispettivamente:

- *style.css*;
- *small.css*;
- *print.css*.

Grazie alla funzionalità che ci mette a disposizione questa versione dello standard, ovvero le *media query*, siamo in grado di definire dei "breakpoint" e quindi di modellare lo stile di visualizzazione in base al device utilizzato, facendo riferimento direttamente alle dimensioni dello schermo.

Per la visualizzazione da desktop abbiamo deciso di visualizzare un menù orizzontale affiancato dal logo del sito e da una barra di ricerca, sfruttando la piena larghezza dello schermo.

Mentre per la visualizzazione da mobile abbiamo scelto di mostrare il menù in verticale, di non visualizzare il logo del sito e di tenere la barra di ricerca sotto al menù, in modo che sia subito visibile all'utente.

Abbiamo inoltre scelto di non utilizzare la stessa formattazione tra le pagine viste dagli utenti e quelle viste dagli amministratori. Infatti nelle pagine viste dagli utenti, come ad esempio *prodotti.php* che racchiude il catalogo dei prodotti disponibili nel nostro sito, abbiamo scelto un layout costituito da più blocchi *div* ognuno dei quali contiene un singolo prodotto, accompagnato dall'immagine e da tutte le informazioni utili all'acquisto.

In questo modo crediamo che l'utente possa trovare minore difficoltà nel cercare le informazioni di cui ha bisogno, piuttosto che trovarsi davanti ad una tabella, ricca di dati, ma di difficile interpretazione.

Diversamente, nella parte del sito dedicata agli amministratori, abbiamo pensato appunto a strutture tabellari (esempio *adminproducts.php*), presupponendo che chi amministra il sito sia in grado di muoversi all'interno della sezione e rendendo la tabella più semplice e chiara possibile.

## 4.3 Comportamento

Per definire il comportamento del sito è stato utilizzato PHP 7 e in qualche caso JavaScript.

Per memorizzare i dati è stato utilizzato un database relazionale MySQL.

### 4.3.1 PHP

PHP è stato utilizzato per la gestione del sito lato server, è stato utile per:

- la visualizzazione dinamica dei dati memorizzati nel database;
- la gestione degli utenti che effettuano il *login*;
- l'inserimento e il controllo di input di utenti all'interno del database;
- l'inserimento di dati nel database da parte degli admin.

Abbiamo raggruppato le parti ricorrenti in ogni pagina in dei file php contenuti nella cartella *general* chiamati:

- *Header.php*;
- *Meta.php*;
- *Footer.php*.

questi file vengono poi richiamati nelle varie pagine attraverso il comando *require*.

La gestione della connessione al database è affidata alla classe *DBAccess* contenuta in *connessione.php*, questa classe oltre ad aprire la connessione al database contiene diverse funzioni che si occupano di interrogare con linguaggio SQL il database:

- **createUser:** Serve per la registrazione degli utenti, controlla che i valori inseriti abbiano una definita lunghezza minima e massima, che il nome utente non sia già registrato e inserisce nel database il nuovo utente;
- **checkUser:** Se l'utente inserito è registrato permette la login, controlla anche se l'utente è un admin o meno;
- **alterAdminright:** toglie i privilegi di amministratore ad un admin;
- **getUserlist:** ritorna una array associativo contenente i dati di tutti gli utenti;

- **getP:** ritorna un array associativo contenente i dati di tutti i prodotti nel database;
- **ricerca:** questa funzione ritorna un array associativo contenente tutti i prodotti che in almeno uno dei campi nome, categoria o descrizione contiene una stringa data;
- **ricercaAvanzata:** ritorna un array associativo contenente i dati di tutti i prodotti di uno specifico tipo in ordine di prezzo;
- **createProduct:** inserisce nel database un nuovo prodotto con i parametri dati;
- **getProddata:** ritorna una array associativo contenente i dati di uno specifico prodotto;
- **modifyProduct:** permette di modificare le informazioni riguardanti uno specifico prodotto contenuto nel database;
- **getBestselling:** ritorna un array associativo contenente i 6 prodotti più venduti;
- **getfullPH:** ritorna un array associativo contenente tutto lo storico dei prodotti venduti;
- **getPH:** ritorna un array associativo contenente lo storico degli acquisti di uno specifico utente;
- **acquista:** inserisce nello storico degli acquisti i prodotti presenti nel carrello di uno specifico utente;
- **eliminaC:** se nel carrello di un dato utente vi è un dato prodotto con quantità 1 questo viene rimosso dal carrello, se la quantità è maggiore di 1 invece la quantità viene diminuita di 1;
- **aggiungiC:** aggiunge un prodotto al carrello di un dato utente, se il prodotto è già presente viene incrementata la quantità;
- **getCarrello:** ritorna un array associativo contenente i prodotti all'interno del carrello di uno specifico utente;
- **getProdReview:** ritorna un array associativo contenente tutte le recensioni di un dato prodotto;
- **aggiungiR:** inserisce nel database una recensione fatta da un utente per un dato prodotto;



- **getLatestBundles:** ritorna un array associativo contenente gli ultimi bundle creati;
- **getB:** ritorna un array associativo contenente i nomi di tutti i bundle;
- **getPB:** ritorna un array associativo contenente l'id di tutti i prodotti di un dato bundle;
- **aggiungiPB:** inserisce un prodotto in un dato bundle;
- **rimuoviPB:** rimuove un prodotto da un dato bundle;
- **creaB:** crea un nuovo bundle;
- **getBundlepartsdata:** ritorna un array associativo contenente le informazioni di tutti i prodotti contenuti in un dato bundle;

#### 4.3.1.1 Gestione utente

Per gestire gli utenti abbiamo utilizzato la variabile `$_SESSION{login_user}` nella quale viene mantenuto l'username degli utenti che effettuano il login fino a quando questi chiudono il browser, grazie a questa variabile l'username dei vari utenti è noto in ogni pagina e ci è dunque possibile mostrare a ognuno il proprio carrello e le informazioni sul proprio account come lo storico degli acquisti; inoltre l'utilizzo di questa variabile ha reso semplice il logout che viene effettuato attraverso i comandi `session_unset` e `session_destroy`.

#### 4.3.2 Javascript

Javascript è stato utilizzato per controllare a lato client se gli input degli utenti sono corretti o meno.

#### 4.3.3 Database MySQL

I dati relativi al nostro sito sono stati salvati in un database relazionale SQL, queste sono le tabelle che costituiscono il database:

- **Account:** username,password,email,admin,datacreazione;
- **Prodotto:** id,nome,categoria,descrizione,valutazione,prezzo;
- **PurchaseHistory:** idordine,compratore,prodotto,data,quantità;
- **Carrello:** username,prodotto,quantità;
- **Recensione:** username,prodotto,review,voto,data;

- **Bundles** nome, descrizione, data;
- **Bundleparts:** bundle, pezzo.

nel database sono inoltre presenti 2 *Trigger*:

- **AggiornaVoto:** questo *Trigger* aggiorna il voto di un prodotto dopo che un utente inserisce la propria valutazione su di esso;
- **AggiornaVotoEdit:** questo *Trigger* aggiorna il voto di un prodotto dopo che un utente modifica la propria valutazione su di esso.

## 5 Validazione

Per validare il codice html e css abbiamo utilizzato *<https://validator.w3.org/>*

Per validare il codice css abbiamo utilizzato *<https://jigsaw.w3.org/css-validator/>*

il sito è stato testato sui browser:

- Chrome (v.);
- Edge (v.);
- Firefox (v.).

## 6 Ruoli

- **Matteo Lotto:** stesura di codice PHP e Javascript, realizzazione del database e della struttura html;
- **Riccardo Tassetto:** realizzazione della struttura html, dei fogli di stile css e della funzione di ricerca;
- **Davide Zago:** stesura di codice PHP e realizzazione della struttura html.