

Progetto di Tecnologie Web

Anno accademico 2020/2021



Componenti

Ayoub	Maher	1187406
Denisa	Hida	1204284
Giacomo	Sassaro	1187566
Gianpiero Giuseppe	Tovo	1193350

Indirizzo sito web: <http://tecweb.studenti.math.unipd.it/gsassaro>

Email referente del gruppo: giacomo.sassaro@studenti.unipd.it

Utenti	Username	Password
Admin	admin	admin
User	user	user

Indice

1	Introduzione	2
1.1	Abstract	2
1.2	Analisi dell'utenza	2
1.3	Funzionalità	2
2	Progettazione	3
2.1	Struttura delle pagine	3
2.2	Struttura del sito	3
2.3	Attori	4
2.4	Accessibilità	5
2.4.1	Percepibile	5
2.4.2	Utilizzabile	5
2.4.3	Comprensibile	5
2.4.4	Robusto	5
2.5	Database	6
3	Realizzazione	7
3.1	HTML	7
3.2	CSS	7
3.3	Javascript	9
3.4	PHP	10
4	Validazioni	12
4.1	WCAG Color Checker	12
4.2	WebAIM WAVE Tool	12
4.3	Total Validator	12
4.4	ChromeVox	12
5	Organizzazione del lavoro	13
6	Note per l'utilizzo e l'installazione	14
6.1	Installazione	14
6.2	Credenziali	14
6.3	Popolamento	14
6.4	Connessione al database	14

1 Introduzione

1.1 Abstract

La Tana del Luppolo è un sito che nasce come vetrina per un piccolo birrificio artigianale. L'idea é quella che i clienti possano lasciare delle recensioni e dare dei voti alle birre che assaggiano ed eventualmente acquistano durante le degustazioni organizzate periodicamente dall'azienda.

1.2 Analisi dell'utenza

Il sito web si rivolge ad un utenza molto variegata, prevalentemente persone appassionate di birra o alla ricerca di nuove passioni. Tale utenza comprende dai più giovani, molto abili e ferrati nella navigazione, fino agli utenti più anziani, in genere con meno dimestichezza nel navigare su internet.

1.3 Funzionalità

Per accedere al sito, data la natura dei contenuti, viene predisposta una pagina utile ad assicurarsi che gli utenti siano maggiorenni e ad impedire quindi di proseguire in caso contrario. Superata la verifica dell'età viene quindi visualizzata una vetrina di birre consigliate, l'utente può quindi procedere: visualizzandone una descrizione più approfondita, visualizzando la lista completa delle birre disponibili, accedendo al sito oppure visualizzando i contatti dell'azienda.

La registrazione al sito permette di aggiungere e rimuovere recensioni personali delle birre, composte da una descrizione e da un voto in decimi.

Gli account amministratori, oltre a gestire le proprie recensioni, hanno la possibilità di rimuovere tutte le recensioni in modo da poter svolgere la loro funzione di moderazione.

Gli account possono essere elevati al ruolo di admin dagli amministratori del database, che si occupano inoltre dell'inserimento di nuove birre.

2 Progettazione

E' stata scelta una strategia di progettazione *Responsive Web Design*, il sito infatti é in grado di adattarsi graficamente in modo automatico in base al dispositivo con il quale viene visualizzato. Si è preferito usare XHTML5 perché ci permette di usare la specifica WAI-ARIA, discussa in seguito.

Il progetto è stato affrontato con un approccio top-down, individuando elementi generici da implementare e procedendo quindi con lo sviluppo di pagine e funzioni.

2.1 Struttura delle pagine

Sono di seguito riportati i differenti elementi della struttura delle pagine del sito.

Header L' header contiene il logo ed il nome del sito, un pulsante per visualizzare la barra di ricerca, un pulsante per accedere al profilo personale ed una barra di navigazione con i riferimenti alle principali pagine del sito.

É presente in tutte le pagine del sito.

Breadcrumb Sotto alla testata è situata la breadcrumb, che permette all'utente di orientarsi visualizzando il percorso dalla homepage fino alla pagina corrente.

La breadcrumb non è visibile nelle pagine di avviso di contenuto non trovato, di accesso negato e di conferma dell'eliminazione dell'account.

Main La sezione Main visualizza il contenuto principale e caratterizzante della pagina.

Footer Al piede della pagina è situato il footer che riporta informazioni sul progetto e i relativi certificati di validazione.

É presente in tutte le pagine del sito.

2.2 Struttura del sito

L'index page del sito reindirizza alla pagina di verifica dell'età, mentre conduce direttamente alla homepage se nelle variabili di sessione è indicato che l'utente ha già eseguito la verifica ed è maggiorenne.

La verifica viene eseguita dalla pagina *ageverification* stessa ed ogni altra pagina reindirizza a quest'ultima se rileva che nella sessione dell'utente non è presente la variabile che rappresenta l'avvenuta verifica.

Di seguito si riportano le pagine del sito e le rispettive caratteristiche.

Home L'*homepage* è la pagina che viene mostrata all'utente dopo la verifica dell'età e contiene una lista di birre in offerta o consigliate dall'azienda.

Prodotti La pagina *prodotti* visualizza tutte le birre disponibili nel database, suddivise in pagine.

La pagina viene inoltre utilizzata per visualizzare i risultati di una ricerca tramite la barra di ricerca.

Dettagli La pagina *dettagli* visualizza informazioni complete e recensioni della birra scelta dalla pagina *prodotti*.

Se l'utente ha effettuato l'accesso vengono visualizzati i controlli per aggiungere e rimuovere recensioni.

Contatti La pagina *contatti* contiene una breve descrizione dell'azienda ed il contatto telefonico e di posta elettronica dell'azienda.

Spazio utente La registrazione e l'accesso degli utenti viene gestita dalle pagine *login* e *registrazione* mentre *dettagliaccount* e *modificadati* permettono la visualizzazione e la modifica dei dati del proprio profilo.

Avvisi Il tentativo di accesso ad aree vietate o a contenuti non disponibili viene notificato dalle pagine *accessdenied* e *notfound* mentre il risultato di query viene visualizzato in sezioni della pagina dalla quale viene eseguita l'operazione.

La pagina *deleteaccount* avverte della corretta eliminazione dell'account e la pagina *logout* si occupa di terminare la sessione e reindirizzare l'utente alla homepage.

2.3 Attori

Vengono di seguito elencati i possibili tipi di utenti che possono interagire con il sito.

- **Utente non abilitato:** Se l'utente non ha una sessione attiva o ha dichiarato di essere minorenne, ogni tentativo di accesso alle pagine del sito risulta in una ridirezione alla pagina di verifica dell'età, dalla quale può proseguire solo dopo aver confermato di essere maggiorenne;
- **Utente maggiorenne e non loggato:** Un utente che ha dichiarato di essere maggiorenne e non è loggato nel sito ha accesso a tutte le pagine tranne quelle di gestione dell'account; il tentativo di accesso a quest'ultime verrà negato e segnalato da una pagina apposita. L'utente può accedere alla pagina di registrazione attraverso il link *Registrazione* nella pagina login, raggiungibile dal tasto account situato nell'header. Nelle pagine di dettagli delle birre saranno visualizzate le recensioni ma non ci sarà la possibilità di modificarle o aggiungerne di nuove;
- **Utente maggiorenne e loggato:** Gli utenti che hanno effettuato l'accesso possono visitare tutte le pagine del sito ed hanno la possibilità di aggiungere ed eliminare le proprie recensioni. Dalla pagina *account* possono inoltre visualizzare e modificare i propri dati, effettuare il logout o eliminare l'account;
- **Utente amministratore:** Un utente amministratore ha accesso a tutte le pagine del sito ed ha la possibilità di eliminare anche le recensioni degli altri utenti. Un account deve essere eletto amministratore dagli operatori del database.

2.4 Accessibilità

Seguendo le linee guida per l'accessibilità del contenuto Web (WCAG 2.1), abbiamo modificato e aggiunto specifici elementi per rendere il nostro sito accessibile a tutti i nostri utenti, tra cui anche quelli che usano tecnologie assistive.

2.4.1 Percepibile

Alternative testuali Tutti i contenuti non testuali presentati all'utente hanno un'alternativa testuale equivalente.

Adattabile Per la strategia seguita (*Responsive Web Design*) il sito non perde informazioni o struttura quando viene visualizzato in dispositivi diversi.

Distinguibile L'uso dei colori non è stato fatto solamente per presentazione ma si usano colori specifici che rappresentano emozioni, come per esempio l'uso del verde quando un'azione esegue con successo e l'uso del rosso quando succede il contrario. La rappresentazione visiva del testo e immagini contenenti testo ha un rapporto di contrasto di almeno 7:1, che ci fa superare il test WCAG AAA.

2.4.2 Utilizzabile

I controlli sono correttamente assegnati ad una label e all'interno di form sono raggruppati da un tag *fieldset*, descritto da un rispettivo tag *legend*.

Tramite l'attributo *tabindex* l'ordine di focus tramite tasto tab viene corretto o nascosto ad alcuni elementi di presentazione, come ad esempio le icone degli utenti delle recensioni, che sono quindi marcate con il valore *presentation* dell'attributo *role*.

I tag di heading sono stati utilizzati seguendo una corretta gerarchia ed in modo coerente per segnalare titoli e contenuti delle sezioni.

Nel sito sono assenti link circolari e sono presenti un *hidden link* utile allo screen reader per saltare al contenuto della pagina ed un *back-to-top-button* per tornare all'inizio della pagina.

2.4.3 Comprensibile

In aiuto ai sistemi di screen-reading sono inoltre segnalate le parole inglesi dagli attributi *xml:lang* e *lang*, le immagini principali sono dotate di un attributo *alternative text* consono e i alcuni controlli marcati con l'attributo *aria-label*.

I valori provenienti dai controlli, prima di essere processati lato server vengono validati e sanitizzati.

2.4.4 Robusto

Tutti i tag sono chiusi in modo corretto ed ogni pagina è caratterizzata da una lista adeguata di keyword.

2.5 Database

Il sito sfrutta un database molto semplice, composto da tre tabelle:

- **Utenti:** raccoglie i dati anagrafici e di accesso di tutti gli utenti registrati al sito. É stata implementato un campo *admin_flag* che viene posto a *TRUE* nel caso in cui l'account sia amministratore; la chiave primaria é un id auto-incrementate e *username* é chiave univoca, così da impedire la registrazione al sito di più utenti con lo stesso username. Abbiamo deciso di non memorizzare la data di nascita nel database in quanto un utente che visita il sito deve aver dichiarato precedentemente di essere maggiorenne, un campo *data_nascita* per gli utenti non verrebbe quindi utilizzato e risulterebbe superfluo;
- **Birre:** raccoglie i dati delle birre presenti nel sito. La chiave primaria é un id auto-incrementante, mentre il nome é chiave univoca, così da impedire l'inserimento di più birre con lo stesso nome;
- **Recensioni:** raccoglie le recensioni delle birre (testo e voto) lasciate dagli utenti. La chiave primaria é un id auto-incrementante, ed essendo l'unica chiave un utente può lasciare più recensioni della stessa birra. Il campo *utente* é chiave esterna riferita al campo *id* della tabella *Utenti* mentre il campo *birra* é la chiave esterna riferita al campo *id* della tabella *Birre*.

Il database é normalizzato.

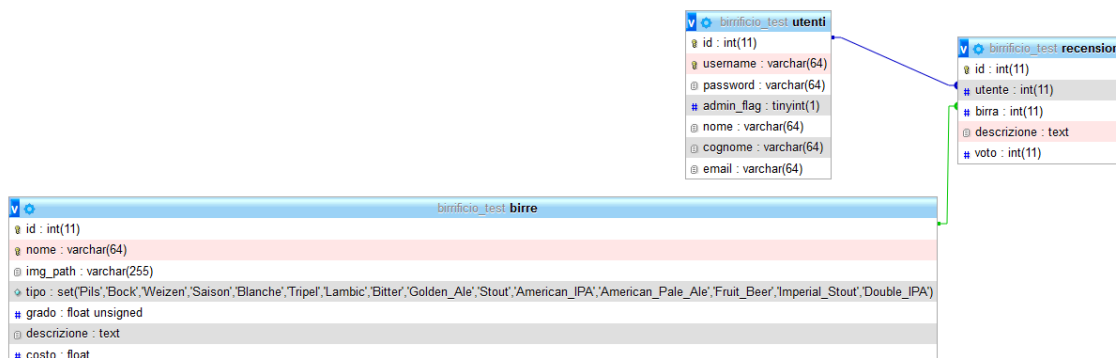


Figura 1: UML database

3 Realizzazione

Ogni pagina ha un suo scheletro in un file HTML, il quale viene aperto dal file *nome_pagina.php* tramite la funzione *file_get_contents()*, successivamente la pagina viene popolata dalla classe PHP **htmlMaker** da noi sviluppata e solo a questo punto viene mostrata all'utente.

Così facendo viene mantenuto totalmente separato il comportamento dalla struttura, rimane possibile gestire le sessioni e non sono necessarie chiamate AJAX dalle pagine.

3.1 HTML

Abbiamo cominciato scrivendo un header ed un footer che fossero unici per tutte le pagine, questi due file HTML vengono letti dalla classe *htmlMaker* e inseriti in tutte le pagine presenti nel sito. Successivamente abbiamo iniziato a scrivere il codice HTML di ogni altra pagina, a partire dalla *home*, *prodotti* e *contatti*. Man mano che le pagine prendevano forma andavamo a spostare il codice HTML scritto nella classe PHP responsabile della generazione del contenuto delle varie pagine.

3.2 CSS

Riguardo allo stile CSS, abbiamo lavorato parallelamente allo sviluppo delle pagine a partire dallo stile principale condiviso per poi procedere anche con lo stile per dispositivi mobile, la quale presentazione è gestita da un apposito foglio di stile che ottimizza la visualizzazione e dispone barra di navigazione e pulsanti di ricerca e account nel menu ad hamburger, situato nella parte destra dell'header.

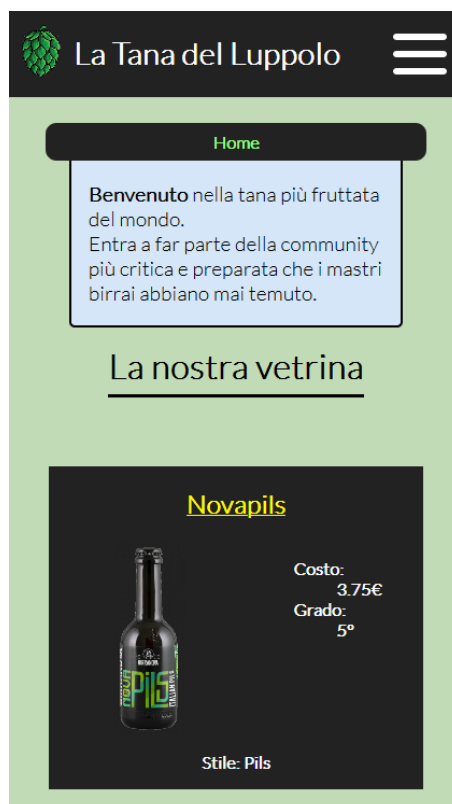


Figura 2: Il layout mobile della homepage

Infine è stato predisposto un foglio di stile per la stampa, dalla quale sono stati nascosti alcuni elementi giudicati superflui (ad esempio la barra di navigazione o i pulsanti back-to-top, ricerca e account) o modificati per migliorarne la presentazione (come la breadcrumb, le recensioni o la grandezza delle immagini principali).

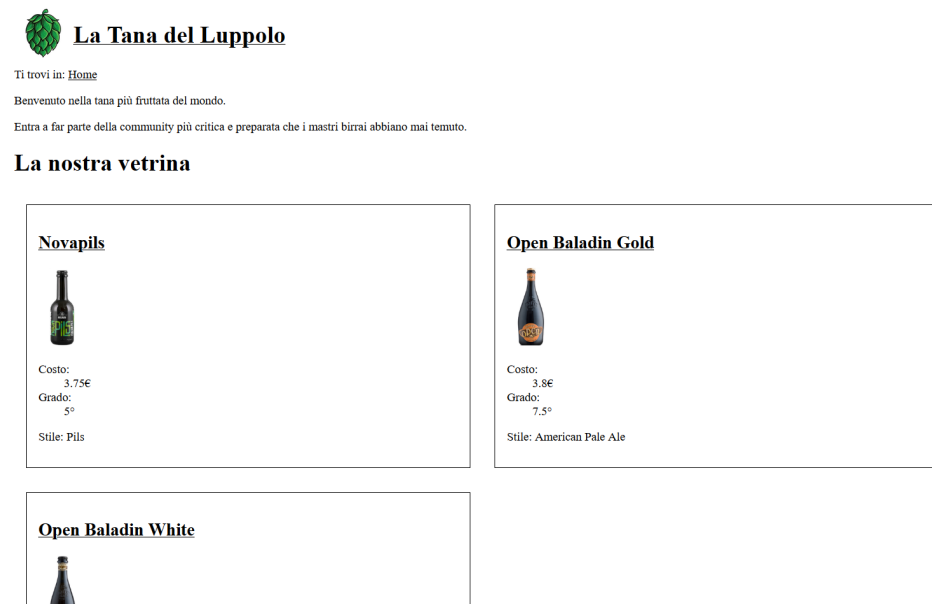


Figura 3: Il layout di stampa della pagina Prodotti

I colori dominanti del sito sono il grigio scuro ed un verde pistacchio per header, sfondi ed altri elementi di presentazione, mentre vengono utilizzati il nero, il verde ed il giallo per testi e link.



La palette è stata scelta in modo da garantire una corretta visualizzazione anche da persone affette da diversi tipi di cecità o sensibilità del contrasto ed il contenuto del sito rimane correttamente visualizzabile anche senza il caricamento dello stile CSS.



Figura 4: Vista della pagina Prodotti da persone affette da deuteranopia (cecità al verde)

3.3 Javascript

Tutta la parte di JavaScript del progetto è contenuta in un unico file `main.js` e viene invocata da un link esterno nella sezione head di ogni pagina.

Il file `main.js` è stato come segue strutturato.

Validatori I primi metodi servono per controllare la corretta compilazione dei vari campi di input testuali presenti nelle form di registrazione di un account e di modifica dei dati di un utente. Inoltre nel caso di non corrispondenza con la specifica grammatica in seguito definita si occupano di mostrare lo specifico errore.

- `validateName(str)`: controlla che il nome dell'utente corrisponda all'espressione regolare `/^[A-Z ,.'-]{3,30}$/i`. Saranno accettate quindi solamente stringhe di lunghezza compresa tra i 3 e i 30 caratteri, composte da caratteri alfabetici e i caratteri `.`, `,`, `'`, `-`;
- `validateSurname(str)`: controlla che il cognome dell'utente corrisponda all'espressione regolare `/^[A-Z ,.'-]{3,30}$/i`. Saranno accettate quindi solamente stringhe di lunghezza compresa tra i 3 e i 30 caratteri, composte da caratteri alfabetici e i caratteri `.`, `,`, `'`, `-`;
- `validateUsername(str)`: controlla che lo username dell'utente corrisponda all'espressione regolare `/^[w_]{4,30}$/i`. Saranno accettate quindi solamente stringhe di lunghezza compresa tra i 4 e i 30 caratteri composte da caratteri alfabetici e il carattere `_`;
- `validatePassword(str)`: controlla che la password dell'utente corrisponda all'espressione regolare `/^[w_]{4,30}$/i`. Saranno accettate quindi solamente stringhe di lunghezza compresa tra i 4 e i 30 caratteri composte da caratteri alfabetici e il carattere `_`;
- `validateEmail(str)`: controlla che l'email dell'utente corrisponda all'espressione regolare

`/[\S]{4,32}@[w]{4,32}((?:\.[w]+)+)?(\.(it|com|edu|org|net|eu)){1}/`

Le stringhe accettate dovranno quindi essere composte da una stringa tra i 4 e i 32 caratteri come identificativo, il carattere di at @, il dominio tra i 4 e i 32 caratteri e infine un suffisso tra (.it|.com|.edu|.org|.net|.eu).

Controlli input I metodi che seguono vengono utilizzati dalle invocazioni tramite submit delle form.

- *CheckRegistrazione()*: si occupa di recuperare il valore dei campi della form di registrazione di un nuovo utente e restituisce true se tutti i campi sono stati compilati e se rispettano le grammatiche prefissate;
- *CheckModificadati()*: si occupa di recuperare il valore dei campi della form di modifica delle informazioni di un utente, restituisce true se tutti i campi sono stati compilati e se rispettano le grammatiche prefissate;

Gestori presentazione Gli ultimi tre metodi servono solamente per la modifica della contenuto del DOM anche al fine di una migliore portabilità verso gli schermi più piccoli.

- *getSearchbar()*: Questo è un metodo trigger che viene invocato nel caso di click sull'icona di ricerca, si occupa di rendere visibile la barra di ricerca ed assegnarle il focus. Nel caso di un secondo click la barra tornerà invisibile;
- *checkWidth()*: Questo è un metodo trigger che viene invocato al caricamento di ogni pagina e al rilevamento di una modifica alla larghezza del body. Quando la larghezza del display diventa inferiore a 850px, come in un dispositivo mobile, si occupa di rimuovere le icone, la barra di navigazione e la barra di ricerca dalla parte di header visibile e aggiungerle al menù laterale a comparsa. Viceversa, nel caso in cui la larghezza dovesse superare 850px riposiziona i contenitori nella parte di header visibile ;
- *showMenu()*: Questo è un metodo trigger che viene invocato nel caso di click sull'icona del menù ad hamburger. Si occupa di mostrare in primo piano il menù nascosto con l'applicazione o la rimozione della classe di stile `open`;

3.4 PHP

Ogni pagina del sito è generata da un file PHP. Abbiamo creato due classi con metodi statici, una per la costruzione dell'HTML e una per la connessione al database, in modo da non aver bisogno di istanziare oggetti per richiamarne le funzioni:

- **htmlMaker.php**: si occupa della costruzione di codice HTML, ed é composta dai seguenti metodi:
 - *makeHead*: metodo utilizzato da tutte le pagine del sito, costruisce l'head HTML.
 - *makeHeader*: metodo utilizzato da tutte le pagine del sito, costruisce l'intestazione HTML.
 - *makeHeading*: metodo utilizzato da tutte le pagine del sito, costruisce il titolo principale.
 - *makeBreadCrumbs*: metodo utilizzato da tutte le pagine del sito, costruisce la breadcrumb.
 - *makeTornaSu*: metodo utilizzato da tutte le pagine del sito, costruisce il pulsante per tornare all'inizio della pagina.

- *makeFooter*: metodo utilizzato da tutte le pagine del sito, costruisce il piè di pagina.
 - *makeBanner*: metodo utilizzato dalla *home* per costruire il banner di presentazione del sito.
 - *listBeers*: metodo utilizzato da *home* e *prodotti*, costruisce la lista con le birre presenti nel sito.
 - *beerInfo*: metodo utilizzato dalla pagina di dettagli di ogni birra, costruisce la parte di dettagli, e nel caso l'utente sia loggato anche la form per l'invio di una recensione.
 - *beerReview*: metodo utilizzato dalla pagina di dettagli di ogni birra, costruisce la lista di recensioni.
 - *makeNotFound*: costruisce una pagina di errore nel caso in cui il contenuto cercato dall'utente non sia presente nel sito.
 - *makeDeleteAccount*: costruisce la pagina che viene restituita quando si elimina un account.
 - *makeAccessdenied*: costruisce una pagina di errore nel caso in cui l'utente non abbia i permessi di vedere una determinata pagina.
 - *userInfo*: metodo utilizzato da *dettagli account*, costruisce una lista con tutte le informazioni dell'account in cui si è loggati.
- **dbConnection.php**: si occupa dell'interazione tra il sito e il database, possiede delle costanti contenenti i dati per stabilire la connessione (username, password, host e nome del database) ed è composta dai seguenti metodi:
 - *openDBConnection*: apre la connessione con il database, è un metodo privato che viene richiamato solo dalle altre funzioni della classe. Nel caso di errore lancia un'eccezione;
 - *collapse*: viene utilizzato per tutte le interrogazioni al database che restituiscono un risultato di una sola colonna, utilizzando questo metodo l'input fornito in forma di lista di array viene compresso in un'array monodimensionale. È un metodo privato che può essere richiamato solamente da altri metodi della classe;
 - *query*: metodo utilizzato per interrogare il database, accetta come parametro la query da inviare ed un valore booleano da impostare a *true* nel caso in cui si voglia utilizzare il metodo *collapse*. Questo metodo apre la connessione al database utilizzando *openDBConnection*, invia la query e nel caso di errori lancia un'eccezione; se il booleano è posto a *true* utilizza il metodo *collapse* per comprimere il risultato della query. Infine chiude la connessione al database e restituisce i dati ricevuti tramite l'interrogazione.
 - *command*: metodo utilizzato per inviare al database comandi (insert, update, delete, alter, drop, create), l'unico parametro accettato è il comando da inviare. Questo metodo apre la connessione al database utilizzando *openDBConnection*, invia il comando, nel caso di errori lancia diverse eccezioni ed infine chiude la connessione.

Per ogni pagina del sito viene prelevata la struttura dal suo file HTML, controllata l'inizializzazione e la valorizzazione della variabile dell'età, successivamente viene controllata la variabile di sessione relativa all'account nel caso in cui quella determinata pagina abbia comportamenti diversi a seconda del tipo di utente che la visualizza. Nelle pagine raggiungibili da più fonti, viene verificata la pagina di provenienza dell'utente e quindi controllata la presenza di parametri *GET* o *POST* nella richiesta. Infine con una serie di *str_replace()* viene aggiornato lo scheletro della pagina inizialmente caricato con i vari metodi presenti in *htmlMaker.php*; solo a questo punto la pagina è pronta per la visualizzazione.

4 Validazioni

4.1 WCAG Color Checker

L'estensione¹ per il browser Firefox non rileva problemi tra i colori scelti ed i livelli di contrasto.

4.2 WebAIM WAVE Tool

L'estensione² per il browser Firefox non segnala problemi e valuta positivamente l'utilizzo dei tag, gerarchia degli heading e label dei controlli.

4.3 Total Validator

Tutte le pagine del sito sono state validate attraverso l'estensione per browser di Total Validator⁴ e Total Validator Test³. Il codice sorgente generato da PHP di ogni pagina è stato validato con standard *XHTML5*, così da poter utilizzare i tag più recenti forniti da HTML5, mantenendo una struttura in stile XHTML. Inoltre è stato verificato che le pagine garantiscano un'accessibilità conforme agli standard *WCAG21 AAA*.

4.4 ChromeVox

È stata usata l'estensione ChromeVox⁵ di Google Chrome per testare il funzionamento dello screen reader sul sito. ChromeVox interpreta le pagine come gli altri screen reader ed è facile da usare per i sviluppatori ed ovviamente da chi ne ha bisogno. Con l'uso di questo tipo di screen reader si è verificato che tutte le funzionalità del contenuto di una pagina sono utilizzabili tramite un'interfaccia di tastiera senza richiedere tempi specifici per la pressione dei singoli tasti.

Links

1. WCAG Color Checker <https://addons.mozilla.org/it/firefox/addon/wcag-contrast-checker>
2. WebAIM WAVE Tool <https://addons.mozilla.org/it/firefox/addon/wave-accessibility-tool/>
3. Total Validator Test <https://www.totalvalidator.com/products/index.html>
4. Total Validator Extension <https://www.totalvalidator.com/products/extension.html>
5. ChromeVox <https://chrome.google.com/webstore/detail/screen-reader/kgejglhpjiefppelpmljglcjbhoiplfn>

5 Organizzazione del lavoro

Il lavoro tra i membri del gruppo é stato ripartito nel seguente modo:

- **Ayoub Maher:**

- Lavoro sui file HTML;
- Creazione e gran parte dello sviluppo JS;
- Lavoro sui file CSS (CSS mobile);
- Lavoro sui file PHP;
- Stesura di alcune parti della relazione.

- **Denisa Hida:**

- Lavoro sui file HTML;
- Lavoro sul file JS;
- Lavoro sui file CSS;
- Lavoro sui file PHP;
- Controllo con screen reader;
- Stesura di alcune parti della relazione.

- **Giacomo Sassari:**

- Lavoro sul file JS;
- Lavoro sui file CSS;
- Creazione e gran parte dello sviluppo PHP;
- Validazione del codice;
- Creazione e stesura di alcune parti della relazione.

- **Gianpiero Giuseppe Tovo:**

- Creazione e gran parte dello sviluppo HTML;
- Lavoro sui file CSS;
- Lavoro sui file PHP;
- Controlli sull'accessibilità;
- Stesura di alcune parti della relazione.

6 Note per l'utilizzo e l'installazione

6.1 Installazione

I requisiti minimi per il corretto funzionamento del sito sono:

- **Server HTTP**, si consiglia APACHE;
- **PHP versione 7.0 o superiore**;
- **DBMS compatibile**, si consiglia MariaDB;

6.2 Credenziali

Come riportato nella tabella ad inizio pagina le credenziali per l'accesso al sito sono:

- **Amministratore**:
 - Username: admin;
 - Password: admin.
- **Utente**:
 - Username: user;
 - Password: user.

Nel sito sono registrati altri utenti, che sono stati utili per il debugging del sito, é comunque possibile registrare altri utenti attraverso la pagina di registrazione.

6.3 Popolamento

Sono state inserite nel sito un numero relativamente basso di birre, mentre le recensioni non hanno un senso in quanto generate randomicamente da un tool. Abbiamo deciso di non perdere troppo tempo nel popolamento del database in quanto il sito é utilizzato puramente a scopo didattico.

6.4 Connessione al database

Le credenziali per la connessione al database vanno modificate nel file *dbConnetion.php* dentro la cartella PHP.

- **HOST_DB**: indirizzo ip del DBMS;
- **USERNAME**: nome utente per l'accesso al DBMS;
- **PASSWORD**: password per l'accesso al DBMS;
- **DB_NAME**: nome del database a cui si vuole collegarsi.