



Burgheria Padovana

Anno accademico: 2020/2021

Indirizzo sito web:

Email referente del gruppo: davide.albiero@studenti.unipd.it

Componenti	Albiero	Davide	1193425
	Fincato	Alessandro	1201264
	Panighel	Cristiano	1201284
	Tossuto	Matteo	1193493

Utenti	Nickname	Password
Admin	admin	admin
User	user	user

Indice

1	Introduzione	3
2	Analisi	4
2.1	Studio dell'utenza finale	4
2.2	Casi d'uso	4
2.2.1	Utente generico	4
2.2.2	Amministratore	4
3	Progettazione	5
3.1	Obiettivi	5
3.2	Design	5
3.3	Database	5
4	Presentazione	6
4.1	Desktop	6
4.2	Mobile	6
4.3	Stampa	6
5	Implementazione	7
5.1	Linguaggi	7
5.1.1	HTML5 e CSS	7
5.1.2	MySQL	7
5.1.3	PHP	8
5.1.4	JavaScript	8
6	Validazione	10
7	Fase di Test e Strumenti	11
7.1	XAMPP	11
7.2	Silktide - Toolbar	11
7.3	Total Validator	11
8	Suddivisione del lavoro	12

1 Introduzione

Il progetto *Burgheria Padovana* vuole implementare un sito Internet che offra la possibilità di fornire informazioni riguardo il suo punto vendita.

Il locale è aperto già da diversi anni e ha finalmente deciso di rinnovarsi creando un proprio sito internet.

Il sito dovrà contenere informazioni riguardanti i panini che offre, suddivisi nelle varie categorie: Pollo, Manzo e Speciali. Inoltre conterrà gli eventi a cui sarà possibile partecipare, la storia del locale, gli orari di apertura e come contattare il locale.

Il sito permette ad un utente privilegiato (admin) di inserire o eliminare gli eventi e di controllare ed eliminare i commenti degli altri utenti. Gli utenti normali (user) saranno visitatori con privilegi minimi per poter inserire od eliminare i propri commenti oltre a poter visitare il sito normalmente.

Gli *user* dovranno effettuare la registrazione o il login prima di poter usufruire dei privilegi.

Inoltre deve essere garantita l'accessibilità in modo che chiunque possa navigare nel sito senza problemi e tranquillamente.

Terminato con l'accessibilità, si pone l'attenzione sull'usabilità: rispettando la separazione tra struttura, presentazione e comportamento e rispettando gli standard *W3C* per quanto riguarda *HTML* e *CSS*.

Il sito dovrà garantire una navigazione fluida agli utenti evitando disorientamento e, nel caso accadesse, fornirgli supporto per tornare al sito.

2 Analisi

2.1 Studio dell'utenza finale

Il locale Burgheria Padovana offre un prodotto internazionale, comodo, veloce e buono.

Pertanto il sito è pensato per rivolgersi ad una categoria di utenti eterogenei, ad esempio: da consumatori abituali a chi vuole provare qualcosa di nuovo, da chi ha bisogno di mangiare qualcosa al volo a chi vuole gustarsi un buon pasto con calma.

Queste categorie di utenti, con privilegi minimi, verrà denominata come *utente generico*. Mentre l'utente con privilegi verrà denominato come *amministratore*. Entrambe le tipologie potranno accedere ai loro privilegi autenticandosi tramite form di login.

Essendo un utenza finale generica, sarà necessario utilizzare un linguaggio informale, semplice e comprensibile. Allo stesso modo si andrà a creare un sito di struttura e layout semplici e simili ai modelli a cui l'*utente generico* è abituato. Si cercherà, quindi, di non rompere le convenzioni esterne e offrendo, indipendentemente dal browser o dal dispositivo utilizzato, una navigazione veloce e intuitiva.

2.2 Casi d'uso

2.2.1 Utente generico

2.2.2 Amministratore

3 Progettazione

3.1 Obiettivi

3.2 Design

3.3 Database

Una parte fondamentale del progetto è il database, in quanto è usato per contenere le informazioni che verranno poi usate dal sito.

È stato pensato di creare un database con la funzione di contenere tutti i dati relativi ai prodotti in vendita compresi i voti e i commenti, agli eventi a cui è possibile partecipare e i dati degli utenti registrati.

Come si può vedere dalla *Figura 1*, le tabelle sono sei:

- **Eventi:** contiene il titolo, la data, il luogo e la descrizione degli eventi a cui sarà possibile partecipare
- **Categoria:** contiene il nome delle categorie
- **Prodotti:** contiene il nome, l'immagine, gli ingredienti e la descrizione dei prodotti venduti
- **Utenti:** contiene l'username e la password degli utenti iscritti. Inoltre si verifica se l'utente è un admin.
- **Voti:** contiene i voti degli utenti
- **Commenti:** contiene i commenti degli utenti

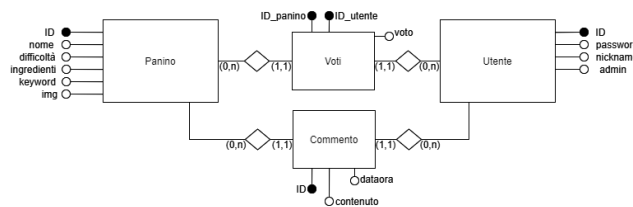


Figura 1: Schema concettuale database

Il database viene interrogato da *PHP* quando bisogna utilizzare o richiamare le informazioni. Le pagine: *Eventi*, *I nostri Burger*, *Panino* sono riempite dinamicamente tramite chiamata *PHP*.

Come spiegato nella sezione *Analisi*, esistono due tipi di utenti: *utente generico* e *amministratore*. Nel database viene utilizzato un booleano per differenziare le due tipologie.

4 Presentazione

4.1 Desktop

Si è deciso di utilizzare una dimensione massima di 1024px in modo che la pagina non risulti eccessivamente grande evitando così l'eventuale sforzo fisico dell'utente nel visionare l'intera pagina.

Caratteristiche della versione *desktop*:

- Il logo del locale è presente;
- Il menu si presenta orizzontale tra il nome del locale e il breadcrumb.

4.2 Mobile

Per facilitarne l'utilizzo e rendere accessibile il sito da *mobile* è stato deciso di apportare alcune modifiche rispetto alla versione *Desktop*.

È stato necessario utilizzare due punti di rottura per le dimensioni dello schermo:

- Un max-width di 730px
- Un max-width di 694px

In base al punto di rottura avvengono determinate modifiche allo stile delle pagine:

- **730px**
 - L'unica modifica, a questo punto di rottura, riguarda la pagina *Menu* dove la sezione verticale a sinistra del *contenutoGenerale*, usata per la selezione della categoria, viene spostata al di sopra del *contenutoGenerale*; Viene trasformata in una sezione orizzontale tenendo la linea di separazione tra questa sezione e il *contenutoGenerale*, presente anche nella versione *Desktop*. In questo modo viene facilitata la visione e l'utilizzo da *mobile*.
- **694px** A questo punto di rottura avvengono diverse modifiche:
 - Il logo del locale viene rimosso;
 - Il menu orizzontale viene trasformato in un menu a tendina(tramite *CSS*), con apertura verticale, adattandosi all'utilizzo da *mobile*;
 - La pagina *Panino* viene riadattata per un utilizzo migliore da *mobile*;
 - La pagina *Contatti* viene riadattata per un utilizzo migliore da *mobile*, verticalizzando il contenuto;
 - La pagina *gestioneEventi* viene riadattata per un utilizzo migliore da *mobile*, verticalizzando le form.

4.3 Stampa

5 Implementazione

5.1 Linguaggi

5.1.1 HTML5 e CSS

È stato deciso di utilizzare il linguaggio *HTML5* per diversi motivi:

- *HTML5* è oramai supportato dalla maggior parte dei Browser Web;
- Dopo un sondaggio proposto alla clientela è stato riscontrato che il 70% dei clienti sceglie fuori casa di mangiare da noi e quindi userebbero il telefono per visitare il sito;
- La maggior parte degli accessi al web avviene tramite mobile dove i browser sono spesso aggiornati.

Si tiene a precisare che in ogni caso viene rispettata la sintassi *XHTML* e in caso il browser non supporti *HTML5* il sito degraderà in modo elegante.

Inoltre si è scelto di utilizzare *HTML5* per la sua fluidezza e funzionalità aggiuntive. Esiste una pagina *HTML5* per ogni pagina del sito.

Come linguaggio di stile è stato usato *CSS3*.

È stata mantenuta la separazione tra struttura e presentazione utilizzando i seguenti accorgimenti:

- non sono stati usati tag di stile nelle pagine *HTML*;
- per la presentazione sono stati utilizzati file esterni senza utilizzare codice a cascata inline o embedded.

Ogni singola regola è stata valutata prima di utilizzarla, in base alla compatibilità dei browser. In alcune parti del sito sono state utilizzate le flex invece delle grid poichè supportato maggiormente.

Si è deciso di utilizzare una misura sempre in em o in percentuale.

Questo migliora l'accessibilità e garantisce una corretta visualizzazione delle pagine su tutti i formati di schermi.

I fogli di stile utilizzati sono tre:

- Un foglio style.css per le impostazioni di stile generale;
- Un foglio mobile.css per le impostazioni di stile che riguardano il mobile, separando i diversi punti di rottura;
- Un foglio print.css per le impostazioni che riguardano la stampa.

5.1.2 MySQL

Per il mantenimento e salvataggio dei dati e delle informazioni, si è deciso di utilizzare *MySQL*

Per le funzioni che necessitano l'uso di *MySQL* viene implementato l'estensione *mysqli*

Il database utilizzato è descritto nel capitolo 3: *Progettazione -> Database*.

Da notare che alcuni campi per l'inserimento hanno una grandezza maggiore di quella limite imposta all'utente. Questo per assicurarci che anche usando tag HTML, dato che in *MySQL* i simboli "<" e ">" vengono tradotti occupando più spazio, ci sia il rischio che l'input non venisse accettato dal DB portando ad un rifiuto dell'inserimento.

5.1.3 PHP

È stato deciso di elaborare tutte le pagine in *PHP*. Questa operazione è necessaria in quanto ogni pagina viene riportato il tasto di login se l'utente non è loggato, altrimenti ne viene mostrato l'username.

Il linguaggio *PHP* viene utilizzato per leggere, ricavare, scrivere ed eliminare informazioni dal database.

Il sito è strutturato in questo modo:

- file *HTML* per ogni singola pagina contenente tutte le parti statiche;
- i relativi file in *PHP*, per elaborare le parti dinamiche e ritornare la pagina completa;
- file *PHP* per la gestione di diverse funzionalità: inserimento o eliminazione di eventi, login e logout, la possibilità di votare o commentare un panino;
- file *PHP* per l'interazione con il *DB*.

In questo modo si mantiene la separazione tra struttura e comportamento, in quanto la struttura viene descritta nei file *HTML*, mentre il comportamento è gestito dai file *PHP* e *JS* come vedremo in seguito.

Nei file *PHP* vengono eseguiti controlli lato client sull'input delle form controllando che i dati inseriti siano corretti e cercando di ridurre al minimo possibili tentativi di *SQL injection*.

I metodi utilizzati per il controllo dell'input sono:

- *trim()*: elimina gli spazi prima e dopo la stringa in input;
- *htmlentities()*: converte tutti i possibili caratteri speciali in entità *HTML*;
- *strip_tags()*: elimina tutti i possibili tag *HTML*;
- inoltre viene utilizzata la funzione *mysql_real_escape_string*;
- nella query per il login viene utilizzato *BINARY* per rendere la form case sensitive;

5.1.4 JavaScript

JavaScript è un linguaggio che si occupa del comportamento del sito.

Nel nostro caso *JavaScript* viene usato per effettuare controlli sulle form e per alcune feature:

- nella pagina *Panino* viene usato per verificare che il commento da inserire rispetti alcune condizioni e in caso contrario l'utente viene avvisato;
- nella pagina *Panino* viene usato per il bottone "carica altri commenti";
- nella pagina *gestioneEventi* viene usato per controllare che i dati immessi nella form *Aggiungi evento* siano corretti, in caso contrario l'utente viene avvisato;
- nella pagina *gestioneEventi* viene usato per prendere dal *DB* le date relative all'evento che si seleziona nella form di eliminazione;

- nella pagina *gestioneCommenti* viene usato per il bottone "carica altri commenti" e per l'eliminazione dei commenti.

6 Validazione

La validazione è uno passo fondamentale del progetto, poiché serve a verificare che siano rispettati gli standard *W3C* per quanto riguarda *HTML* e *CSS*.

Infatti rispettare gli standard assicura un codice pulito, corretto e che favorisca l'accessibilità e l'usabilità.

Un sito che rispetta gli standard assicura:

- un sito più accessibile e usabile;
- un alto livello di compatibilità tra i diversi browser, rendendo minime o nulle le differenze di visualizzazione del sito, sempre considerando la possibile versione del browser utilizzato dall'utenza a cui ci si riferisce;
- essendo valido non dovrebbe contenere errori e quindi la navigazione risulta più veloce e fluida;
- un sito valido e senza errori non viene interpretato dal browser "a modo suo" ma rispetta "il volere" degli sviluppatori comportandosi come previsto da questi ultimi;
- un sito valido e senza errori viene indicizzato in modo positivo dal browser.

Per validare il sito sono stati utilizzati i seguenti strumenti:

- **W3C HTML Validator**

Indirizzo sito web: <https://validator.w3.org/>

È un servizio gratuito di *W3C* che consente di validare le pagine *HTML*. La validazione può avvenire in tre modi: attraverso il caricamento del file da validare, facendo copia e incolla del codice da validare o inserendo l'indirizzo della pagina se quest'ultima si trova online.

Se il codice non è valido, viene segnalato il numero di errori, il loro tipo e a quale riga e colonna della pagina sono stati trovati.

- **W3C CSS Validator**

Indirizzo sito web: <https://jigsaw.w3.org/css-validator/>

È un servizio gratuito di *W3C* che consente di validare il codice *CSS*.

Anche in questo caso i metodi di validazione sono tre come nel sito precedente.

La segnalazione degli errori viene gestita nell'identico modo del sito precedente.

- In entrambi i siti, se le pagine e il codice risultano validi, riportano del codice *HTML* per poter utilizzare delle immagini, del *W3C*, che certificano la validazione del sito.

7 Fase di Test e Strumenti

7.1 XAMPP

XAMPP è una distribuzione di Apache che permette, tra le varie funzioni, l'utilizzo di *MySQL* e *PHP* di cui abbiamo usufruito.

Il loro utilizzo ci ha permesso di utilizzare in locale la parte dinamica del sito e la parte di Database, testando le richieste *PHP* e le interrogazioni *MySQL*.

7.2 Silktide - Toolbar

È un plugin del browser Google Chrome, che permette di testare la navigazione del sito attraverso l'attivazione di varie disabilità capendo se il sito risulta accessibile.

- Permette di verificare che il sito sia usufruibile e di facile lettura per persone con dislessia.
- Può simulare la visualizzazione del sito da parte di persone daltoniche verificando che sia utilizzabile da queste ultime.
- Può simulare la visualizzazione del sito da parte di persone con la cataratta e verificare che sia accessibile per loro.
- Può simulare la visualizzazione del sito da parte di persone con perdita di visione periferica o perdita di visione centrale verificando che possano accedere ed utilizzare il sito.
- Può simulare uno screen reader, verificando che sia di facile lettura a quest'ultimo e non ci siano problemi di perdita di focus, che ogni parte del sito sia capibile solo attraverso l'utilizzo di uno screen reader e che questo permetta l'uso di ogni funzione del sito.

Il fine ultimo per l'utilizzo di questo plugin è di rendere il sito accessibile a più categorie di utenti possibili cercando di prendere in considerazione tutte le disabilità che rendono complicato l'utilizzo del browser.

7.3 Total Validator

È un strumento molto potente che consente una validazione ottimale delle pagine Web, capendo se sono accessibili.

Le sue funzioni sono:

- Controlla che il codice *HTML* rispetti gli standard.
- Controlla che il codice *CSS* rispetti gli standard.
- Controlla che vengano rispettate le regole di accessibilità, verifica che siano rispettate le *WCAG* e utilizza *AIRA* test.
- Controlla che i link funzionino.

8 Suddivisione del lavoro

- **Albiero Davide**

- Progettazione iniziale
- Sviluppo codice HTML
- Sviluppo codice CSS
- Sviluppo DB
- Sviluppo codice PHP
- Sviluppo codice JS
- Validazione delle pagine
- Test di accessibilità
- Test di usabilità

- **Fincato Alessandro**

- Progettazione iniziale
- Sviluppo codice HTML
- Sviluppo codice CSS
- Sviluppo DB
- Validazione delle pagine
- Test di accessibilità
- Test di usabilità

- **Panighel Cristiano**

- Progettazione iniziale
- Sviluppo codice HTML
- Sviluppo codice CSS
- Validazione delle pagine
- Test di accessibilità
- Test di usabilità

- **Tossuto Matteo**

- Progettazione iniziale
- Sviluppo codice HTML
- Sviluppo codice CSS
- Sviluppo DB
- Sviluppo codice PHP
- Validazione delle pagine
- Test di accessibilità
- Test di usabilità
- Stesura Relazione

Queste sono le principali attività svolte dal singolo membro del gruppo. Tutti i membri hanno partecipato sia allo sviluppo del sito mobile che desktop. È stata usata una repository di *GitHub*, per permettere uno svolgimento coordinato del lavoro attraverso l'utilizzo di *Issue* e *Pull Request*. Inoltre attraverso l'uso di *GitHub* si può verificare il lavoro di ognuno dei membri. La comunicazione è avvenuta tramite un gruppo *Telegram*, tramite meeting *Zoom* e lo scambio di commenti su *GitHub*. Veniva discusso come strutturare il progetto, eventuali scelte importanti o critiche.