



Sushi Nakamura

Progetto di Tecnologie Web A.A. 2019/2020

Informazioni sul gruppo

Membri	Dindinelli Alessandro - 1170457
	Frison Nicolò - 1147682
	Giardina Mirco - 1136663
	Tommasin Alessandro - 1189293

Indice

1	Introduzione	2
1.1	Abstract	2
2	Analisi	2
2.1	Analisi dell'utenza	2
2.2	Casi d'uso	2
2.2.1	Utente non autenticato	2
2.2.2	Utente generico autenticato	2
2.2.3	Amministratore	2
3	Progettazione	2
3.1	Obiettivi	2
3.2	Layout	3
3.3	Accessibilità	3
4	Implementazione	3
4.1	Linguaggi	3
4.1.1	XHTML 1.0 Strict e HTML5	3
4.1.2	CSS	4
4.1.3	PHP	4
4.1.4	SQL	4
4.1.5	JavaScript	5
5	Fase di test	5
5.1	Strumenti usati	5
5.1.1	W3C HTML Validator	5
5.1.2	W3C CSS Validator	6
5.1.3	TotalValidator	6
5.1.4	SonarCloud	6
6	Organizzazione del lavoro	6

1 Introduzione

1.1 Abstract

Il sito web **Sushi Nakamura** è stato sviluppato per permettere all'omonimo ristorante di Padova un mezzo per promuovere sè stesso ed il suo nuovo servizio di take away. Nel sito è possibile reperire tutte le informazioni riguardanti i contatti ed i prodotti che possono essere acquistati. Inoltre l'amministratore ha la possibilità di inserire, rimuovere e modificare eventuali articoli in vendita e news che possono essere visualizzate dagli utenti.

2 Analisi

2.1 Analisi dell'utenza

2.2 Casi d'uso

I casi d'uso possono essere riassunti sotto le seguenti categorie:

2.2.1 Utente non autenticato

2.2.2 Utente generico autenticato

2.2.3 Amministratore

3 Progettazione

3.1 Obiettivi

Gli obiettivi principali perseguiti durante la progettazione del sito sono i seguenti:

- **Separazione tra struttura, presentazione e comportamento:** Obiettivo fondamentale, in quanto raggiungerlo permette di soddisfare più agevolmente anche gli altri punti. La struttura è stata realizzata con documenti in XHTML 1.0 Strict dove possibile, così da garantire una maggiore retrocompatibilità con vecchi browsers, ed HTML5 dove si sono ritenute necessarie le funzionalità aggiuntive permesse dal linguaggio. Tutto il codice redatto è stato scritto secondo le raccomandazioni W3C, accertando che fossero state rispettate tramite validazione. La presentazione è stata sviluppata con fogli di stile CSS linkati, mentre il comportamento con script esterni realizzati in Javascript e PHP. In questo modo la struttura non dovrà cambiare, anche a seguito di modifiche alla presentazione del sito.
- **Accessibilità:** Il sito deve poter essere fruibile agevolmente dal maggior numero di utenti possibile, compresi quelli con differenti tipi di disabilità. Per garantire una buona accessibilità, alcune misure adottate sono:

- Uso dei `tabindex` nel menù di navigazione;
- Testo alternativo per le immagini;
- Testi e link con buoni livelli di contrasto;
- Uso dell'attributo `lang` per testi non in italiano;
- **Fluidità:** Il sito deve poter essere consultabile tramite varie tipologie di dispositivi, tra cui PC desktop, tablet e smartphone. Bisogna quindi garantire una buona adattabilità alle differenti dimensioni di schermo.
- **Fruibilità:** Per realizzare un sito navigabile intuitivamente si sono seguite alcune linee guida comuni nel web, come ad esempio:
 - Sfruttare un layout a tre pannelli (per riferimento vedere il capitolo 3.2 Layout METTERE UN RIFERIMENTO?);
 - Agevolare lo scroll tramite link relativi per raggiungere diversi punti di una stessa pagina;
 - Mantenere colorazioni diverse per link visitati e non;

3.2 Layout

3.3 Accessibilità

4 Implementazione

4.1 Linguaggi

4.1.1 XHTML 1.0 Strict e HTML5

Come già detto, per la realizzazione del sito si è optato per avere sia pagine in XHTML 1.0 Strict, che in HTML5. L'uso di XHTML 1.0 garantisce una migliore retrocompatibilità con browsers più datati ed una migliore aderenza del codice agli standard di correttezza, ma HTML5 permette di sfruttare alcune funzionalità aggiuntive che abbiamo sfruttato ad esempio in vari form per inserire input di tipo *date*, utilizzare *placeholder* o per rendere dei campi *required* in modo da chiarire e migliorare l'usabilità del sito. In entrambi i casi per ottenere codice pulito si sono seguiti i seguenti principi:

- Mantenere la separazione tra struttura e presentazione, usando file separati linkati nell'header;
- Ogni `<tag>` aperto deve essere poi chiuso `</tag>` (oppure `<tag />`);
- I tag di intestazione vanno usati in base all'importanza di ciò che racchiudono, e non in base all'estetica base html;

In generale, per garantire un codice corretto, sono state seguite le linee guida che sono state illustrate durante il corso di Tecnologie Web e quelle dello standard W3C. La verifica del codice è stata poi effettuata utilizzando il tool di validazione W3C Validator¹, la piattaforma SonarCloud² integrata con GitHub ed il software di validazione di siti Total Validator.³

Si è prestata attenzione al ruolo dei metatag nell'header, il cui buon uso migliora anche l'indicizzazione del sito da parte dei motori di ricerca, aspetto fondamentale per un'attività commerciale che voglia ricevere un'adeguata visibilità nel web.

Si è deciso di fare uso anche di tabelle, rendendole più agevolmente accessibili tramite l'uso di adeguati *summary*, e di *scope* su colonne e righe di interesse.

4.1.2 CSS

4.1.3 PHP

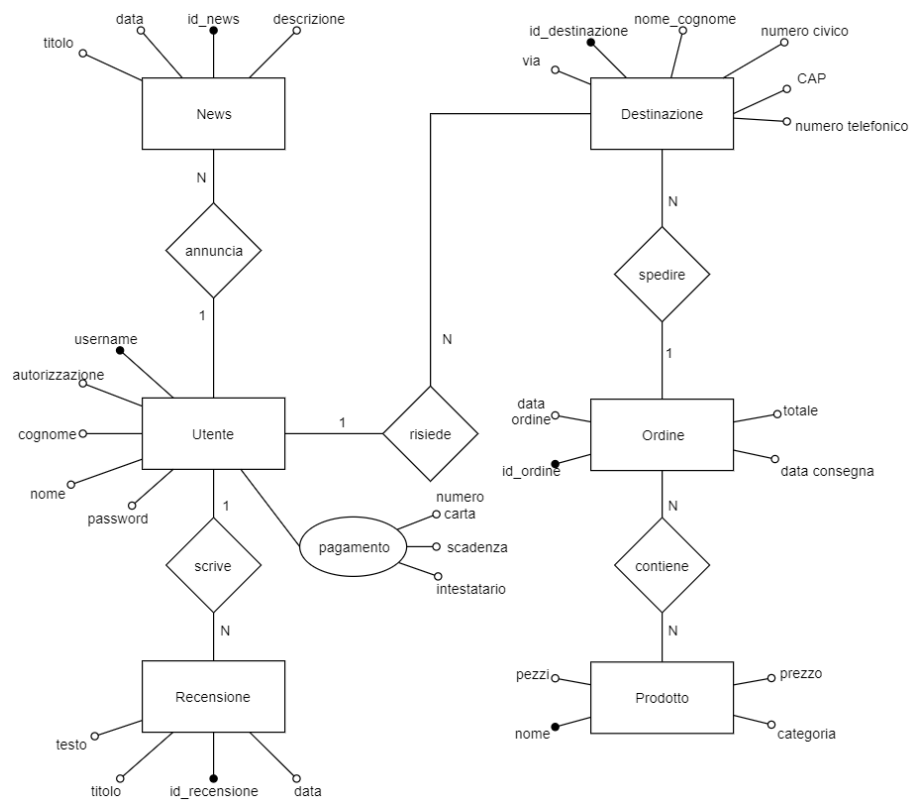
4.1.4 SQL

Sql è stato usato per codificare il database. Si rimanda al file *creazione_database.sql* nella cartella *Database* [\[url\]](#) della repository per il file di costruzione del database. Di seguito il diagramma ER del database:

¹Riferimento al sito di [W3C Validator](#).

²Riferimento al sito di [SonarCloud](#).

³Riferimento al sito di [Total Validator](#).



4.1.5 JavaScript

5 Fase di test

5.1 Strumenti usati

5.1.1 W3C HTML Validator

Le pagine html sono state validate usando il validatore fornito dall'organizzazione W3C per garantire la corretta visualizzazione del contenuto della pagina senza fare entrare i browser in **Quirks Mode**. É stato usato anche per validare il risultato delle pagine php incollando il risultato ottenuto facendo eseguire lo script php.

5.1.2 W3C CSS Validator

5.1.3 TotalValidator

5.1.4 SonarCloud

Servizio integrato con GitHub per la verifica del codice nella repository. Ad ogni push veniva fatta un **analisi statica** del codice alla ricerca di problemi e vulnerabilità come ad esempio un problema comune è stata la ripetizione rindondate di regole css. Questa fase di test era bloccante, ovvero perchè il codice venisse aggiunto dovevano prima essere risolti i problemi.

6 Organizzazione del lavoro

Il progetto è stato suddiviso in modo tale che ogni membro avesse la possibilità di creare sia alcune pagine HTML che il relativo CSS, facendo da verificatore nelle pagine degli altri membri. Lo stesso può essere detto per quanto concerne PHP, JavaScript e la creazione ed il popolamento del database. La sviluppo può essere seguito nella repository di GitHub utilizzata:

<https://github.com/Mirco469/ProgettoSushi>