



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	4
4	Implementazione	4
4.1	Linguaggi	4
4.1.1	XHTML 1.0 Strict e HTML5	4
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. 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. Tutto il codice redatto è stato scritto secondo le raccomandazioni W3C, accertando poi che siano state rispettate, validando HTML e CSS con i rispettivi tool di W3C.
- **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;
- Assenza di link circolari;
- 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 ben strutturato, che faciliti l'individuazione del contenuto di interesse;
 - 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

Si è deciso di strutturare il sito con un layout a colonna singola, come illustrato in Figura 1. Questo modello permette una buona mantenibilità, dato che anche in caso si effettuino modifiche al contenuto della pagina la sua presentazione rimane uniforme, avendo a disposizione l'intera larghezza della finestra. Dimostra di essere anche un layout molto flessibile in quanto la transizione dalla presentazione su desktop a quella su dispositivi mobili risulta fluida.

L'Header, che contiene logo e nome del ristorante, fondamentalmente varia solo in dimensione nella transizione tra i due layout.

Il Footer invece contiene informazioni generali sul ristorante, la dichiarazione del Copyright, e l'eventuale immagine di certificata validazione delle pagine XHTML 1.0 Strict. Per garantire una lettura migliore, restringendo la finestra si passa ad una visualizzazione da doppia a singola colonna.

La principale differenza tra i layout scelti sta nella posizione e presentazione del menù di navigazione.

Nei device con finestre di visualizzazione sufficientemente larghe si è scelto di disporre le voci del menù in riga, creando quindi una barra di navigazione. I link alle varie pagine principali del sito sono disposte a partire da sinistra, mentre quelli riguardanti pagine per gli utenti e le operazioni a loro disposizione si trovano a partire da destra.

Nei dispositivi più piccoli si è deciso invece di creare un menù espandibile, raggiungibile premendo sull'icona ad hamburger nell'angolo in alto a destra, che mostra tutte le voci disponibili, per rispettare il comportamento più comune che un utente medio si aspetta utilizzando uno smartphone.

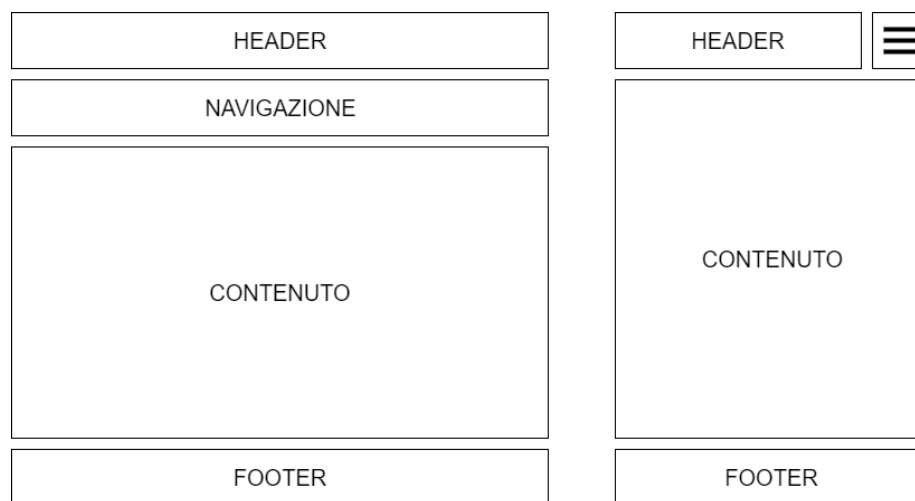


Figura 1: Schema del layout per desktop e mobile.

3.3 Accessibilità

4 Implementazione

4.1 Linguaggi

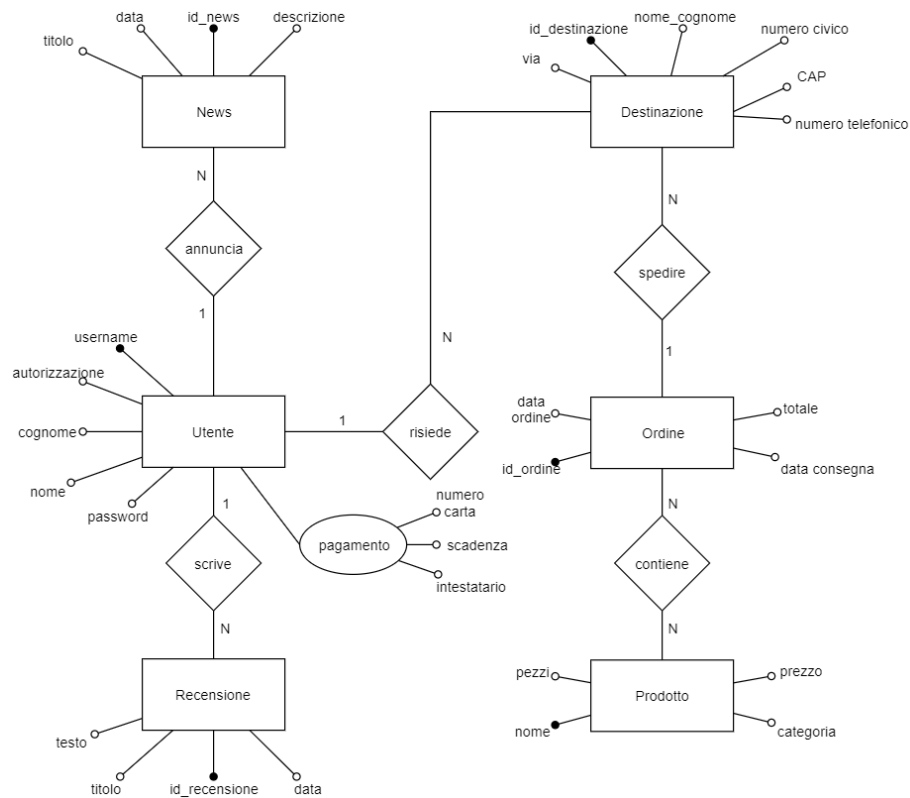
4.1.1 XHTML 1.0 Strict e HTML5

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:



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>