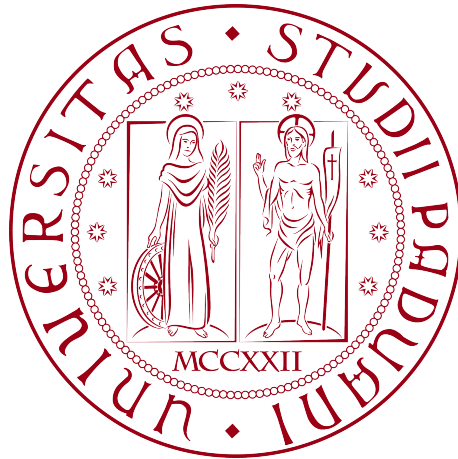


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



***SviluppAbile: un'estensione per sviluppare pagine
web accessibili con l'aiuto di un chatbot***

Tesi di Laurea

Relatrice

Prof.ssa Ombretta Gaggi

Laureanda

Elena Chilese

Matricola 2008074

ANNO ACCADEMICO 2024-2025

“Dedicata ad Elena, che mai si è arresa...”

Ringraziamenti

Per prima cosa desidero ringraziare la professoressa Ombretta Gaggi, relatrice della mia tesi e tutor del mio stage, per l’assistenza e il sostegno fornitimi durante tutto il percorso di stage e nella stesura del lavoro.

Un sincero grazie va anche a Salvatore Gatto per il supporto e l’aiuto durante lo stage.

Con profondo affetto ringrazio i miei genitori e, in particolar modo, mia mamma, per la pazienza e la forza con cui mi ha sempre sostenuta nei momenti più difficili. Un ringraziamento va anche ai miei zii e soprattutto a mio nonno Adriano, il mio angelo custode, per l’affetto che non mi ha mai fatto mancare.

Un grazie speciale a Michele, che mi è stato accanto fin dall’inizio del percorso universitario, diventando non solo un prezioso punto di riferimento in ogni occasione, ma anche il mio compagno di avventure ed il mio migliore amico.

Ringrazio di cuore tutti i miei amici e i compagni di università (in particolare Matteo, Elena ed Erica) per la loro vicinanza ed il sostegno dimostrato in questi anni.

Desidero esprimere la mia gratitudine inoltre ai colleghi (e miei ex professori) dell’ITE Fusinieri, che hanno saputo trasmettermi passione e competenze fondamentali indispensabili per il raggiungimento di questo traguardo universitario.

Padova, Settembre 2025

Elena Chilese

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di trecento ore svolto presso il Dipartimento di Matematica “Tullio Levi-Civita” dell’Università di Padova.

Il prodotto finale dello stage è un’estensione web *Chrome-based_G* per aiutare gli sviluppatori web nella creazione di pagine web accessibili.

Gli obiettivi del progetto sono:

- Studio dell’accessibilità web;
- Studio di Manifest V3;
- Studio delle possibili *API_G* per l’integrazione dell’ *IA_G*;
- Creazione di un’estensione web che integri un *chatbot_G IA_G* per analizzare il codice sorgente della pagina web ed offrire risposte adeguate.

Tabella dei contenuti

1	Introduzione	1
1.1	L'idea del progetto	1
1.2	Analisi delle soluzioni già presenti	3
1.3	Organizzazione del testo	3
2	Analisi dei requisiti	5
2.1	Obiettivo del progetto	5
2.2	Casi d'uso	5
2.2.1	Attori	5
2.2.2	Elenco e descrizione dei casi d'uso	6
	UC0: Visualizzazione prompt iniziale	6
	UC1: Analisi assistita	6
	UC2: Modalità sviluppo guidato	7
	UC3: Blocco estensione su pagine di sistema	9
	UC4: Gestione errori di comunicazione con l'IA	9
2.3	Tracciamento dei requisiti	9
2.3.1	Requisiti obbligatori	10
2.3.2	Requisiti desiderabili	10
2.3.3	Requisiti facoltativi	11
3	Tecnologie	12
3.1	Linguaggi	12
3.1.1	HTML	12
3.1.2	CSS	13
3.1.3	Prism.js	14
3.1.4	JavaScript	14
3.2	Tecnologie e strumenti	16

TABELLA DEI CONTENUTI

3.2.1	Chrome Extension	16
3.2.1.1	Manifest V3	16
3.2.2	Ollama	19
3.2.3	GitHub	20
3.2.4	VSCoDe	21
3.2.5	Total Validator	22
3.2.6	WAVE	22
3.2.7	Lighthouse	23
4	Sviluppo del progetto di stage	24
4.1	Web design	24
4.2	PoC	24
4.3	Prodotto finale	25
4.3.1	Guida all'utilizzo	26
4.3.1.1	Avvio dell'estensione	26
4.3.1.2	Analisi assistita	27
4.3.1.3	Modalità guidata	30
4.3.2	Interazione con l'AI	32
4.3.3	Filtraggio risposta generata	34
4.4	Problematiche riscontrate	35
4.5	Problematiche riscontrate	36
5	Test	38
5.1	Test "Analisi assistita"	38
5.1.1	F1-score	38
5.1.2	Sito web: SudokuWorld	39
5.1.2.1	Total Validator	39
5.1.2.2	Lighthouse	40
5.1.2.3	SviluppAbile	40
5.1.2.3.1	F1-score	46

TABELLA DEI CONTENUTI

5.1.3	Sito web: Dolce Risveglio	47
5.1.3.1	Total Validator	47
5.1.3.2	Lighthouse	48
5.1.3.3	Sviluppabile	48
5.1.3.3.1	F1-score	50
5.1.4	Sito web: E-lixirium	51
5.1.4.1	Total Validator	52
5.1.4.2	Lighthouse	52
5.1.4.3	SviluppAbile	53
5.1.4.3.1	F1-score	54
5.1.5	Sito web: Corsa Ideale	55
5.1.5.1	Total Validator	56
5.1.5.2	Lighthouse	56
5.1.5.3	SviluppAbile	57
5.1.5.3.1	F1-score	59
5.1.6	Sito web: BookOverflow	60
5.1.6.1	Total Validator	60
5.1.6.2	Lighthouse	60
5.1.6.3	SviluppAbile	60
5.1.6.3.1	F1-score	62
5.1.7	Sito web: LuzzAuto	62
5.1.7.1	Total Validator	63
5.1.7.2	Lighthouse	63
5.1.7.3	SviluppAbile	64
5.1.7.3.1	F1-score	65
5.2	Test “Modalità guidata”	66
5.3	Lista WCAG citate	71

TABELLA DEI CONTENUTI

6 Conclusioni	73
6.1 Raggiungimento degli obiettivi	73
6.1.1 Misurazione quantitativa dei requisiti soddisfatti	74
6.2 Sviluppi futuri	74
6.3 Consuntivo finale	75
6.4 Competenze acquisite	75
6.5 Valutazione personale	77
Bibliografia	ii

Elenco delle figure

3.1	Logo HTML5	13
3.2	Logo CSS3	14
3.3	Logo JavaScript	15
3.4	Logo Ollama	20
3.5	Logo GitHub	21
3.6	Logo VSCode	21
3.7	Logo Total Validator	22
3.8	Logo WAVE	23
3.9	Logo Lighthouse	23
4.1	Logo dell'estensione <i>SviluppAbile</i>	26
4.2	Pulsante per il cambio modalità diurna-notturna	26
4.3	Popup di avvio dell'estensione <i>SviluppAbile</i>	27
4.4	Modalità: analisi assistita	28
4.5	Modalità: analisi assistita - modalità notturna	28
4.6	Modalità: analisi assistita dopo un'interazione	29
4.7	Modalità: sviluppo guidato, pagina iniziale	30
4.8	Modalità: sviluppo guidato - modalità notturna	31
4.9	Modalità: sviluppo guidato, dopo alcune interazioni	32
5.1	Analisi di Total Validator sul sito web <i>SudokuWorld</i>	39
5.2	Analisi di Lighthouse sul sito web <i>SudokuWorld</i>	40
5.3	Analisi di Total Validator sul sito web <i>Dolce Risveglio</i>	47
5.4	Analisi di Lighthouse sul sito web <i>Dolce Risveglio</i>	48
5.5	Analisi di Total Validator sul sito web <i>E-lixirium</i>	52
5.6	Analisi di Lighthouse sul sito web <i>E-lixirium</i>	53
5.7	Analisi di Total Validator sul sito web <i>Corsa Ideale</i>	56
5.8	Analisi di Lighthouse sul sito web <i>Corsa Ideale</i>	57

5.9	Analisi di Total Validator sul sito web <i>BookOverflow</i>	60
5.10	Analisi di Lighthouse sul sito web <i>BookOverflow</i>	60
5.11	Analisi di Total Validator sul sito web <i>LuzzAuto</i>	63
5.12	Analisi di Lighthouse sul sito web <i>LuzzAuto</i>	64
5.13	Pagina HTML base per effettuare i test della modalità guidata	66

Elenco delle tabelle

2.1	Tabella del tracciamento dei requisiti obbligatori	10
2.2	Tabella del tracciamento dei requisiti desiderabili	11
2.3	Tabella del tracciamento dei requisiti facoltativi	11
3.1	Confronto modelli Ollama utilizzati	19
5.1	Tabella riassuntiva analisi <i>SudokuWorld</i> tramite <i>SviluppAbile</i>	46
5.2	Tabella riassuntiva analisi <i>Dolce Risveglio</i> tramite <i>SviluppAbile</i>	51
5.3	Tabella riassuntiva analisi <i>E-lixirium</i> tramite <i>SviluppAbile</i>	55
5.4	Tabella riassuntiva analisi <i>Corsa Ideale</i> tramite <i>SviluppAbile</i>	59
5.5	Tabella riassuntiva analisi <i>BookOverflow</i> tramite <i>SviluppAbile</i>	62
5.6	Tabella riassuntiva analisi <i>LuzzAuto</i> tramite <i>SviluppAbile</i>	65

Capitolo 1

Introduzione

L'accessibilità web rappresenta un aspetto fondamentale per garantire che tutte le persone, indipendentemente dalle loro abilità o disabilità, possano utilizzare i contenuti digitali in modo efficace. Le linee guida [WCAG_G](#) (*Web Content Accessibility Guidelines*) sono uno standard internazionale, esse definiscono i criteri tecnici per migliorare l'accessibilità dei siti web, intervenendo su aspetti quali la struttura semantica, la navigazione tramite tastiera, il contrasto cromatico dei colori e la compatibilità con tecnologie assistive.

Un sito accessibile non solo migliora l'esperienza d'uso per persone con disabilità visive, motorie o cognitive, ma estende anche la fruibilità a un pubblico più ampio, contribuendo a un web più inclusivo e universale.

Nonostante ciò, spesso la realizzazione pratica di siti accessibili incontra difficoltà dovute a scarsa conoscenza delle best-practices dello sviluppo web accessibile, delle normative in questione e della scarsa diffusione di strumenti automatizzati efficaci.

1.1 L'idea del progetto

Il progetto sviluppato durante il mio stage si inserisce all'interno di un'iniziativa molto più ampia intitolata *“Supporting Accessibility Auditing and HTML Validator using Large Language Models”* a cui partecipano e collaborano docenti e stagisti dell'Università di Padova e dell'Università di Bologna.

Tale progetto nasce dall'esigenza di affrontare il problema dell'accessibilità web, diritto fondamentale per tutti i cittadini (in particolare per le persone con disabilità) che devono poter accedere alle informazioni sul web senza barriere.

Nonostante le normative nazionali e internazionali (come la *Direttiva Europea sull'Accessibilità Web* e l' *European Accessibility Act*) impongano requisiti chiari, una buona

parte dei siti web presenta ancora numerose criticità di accessibilità. [17]

In questo contesto, il progetto mira a valutare, testare e sfruttare le capacità dei *Large Language Models* (LLM_G) per supportare l’audit dell’accessibilità e la validazione del codice $HTML_G$.

L’obiettivo finale è sviluppare strumenti innovativi che utilizzino modelli di *intelligenza artificiale* $_G$, come $ChatGPT_G$, per fornire un supporto interattivo agli sviluppatori, aiutandoli a identificare e correggere problematiche di accessibilità in modo più efficace e comprensibile rispetto ai metodi tradizionali. Attraverso questa integrazione, si intende migliorare la qualità e soprattutto la chiarezza dei risultati e favorire una cultura più diffusa sull’accessibilità digitale.

Inoltre, si vuole anche valutare se gli LLM siano effettivamente in grado di svolgere questo tipo di lavoro.

L’estensione “*SviluppAbile*” è un primo esempio di strumento per lo sviluppo guidato di pagine web accessibili. Essa nasce dall’esigenza di supportare gli sviluppatori web nel creare pagine accessibili in modo semplice e interattivo, sfruttando le potenzialità dell’*intelligenza artificiale*.

L’estensione *Chrome-based* sviluppata offre due modalità principali:

- una modalità di analisi assistita, che permette di ispezionare il DOM_G di una pagina web e di porre domande specifiche sull’accessibilità, ottenendo risposte chiare e contestualizzate;
- una modalità di sviluppo guidato, in cui l’utente riceve suggerimenti di codice accessibile in tempo reale, visualizzati in una colonna centrale e pronti per essere copiati o scaricati.

Questo duplice approccio consente di integrare nel flusso di lavoro quotidiano degli sviluppatori un valido supporto basato sull’*IA*, con l’obiettivo di facilitare il rispetto delle linee guida $WCAG$ e migliorare la qualità complessiva del codice.

1.2 Analisi delle soluzioni già presenti

Sul mercato esistono diversi strumenti e *plugin* dedicati alla verifica dell'accessibilità delle pagine web, come Total Validator (vedi paragrafo 3.2.5), WAVE (vedi paragrafo 3.2.6) e Lighthouse (vedi paragrafo 3.2.7), che offrono principalmente funzionalità di scansione automatica e generazione di report.

Questi strumenti, tuttavia, forniscono spesso una panoramica statica dei problemi riscontrati; indicano chiaramente dove si trova l'errore, ma i risultati sono generalmente formulati in modo formale e destinati ad un esperto di accessibilità. Di conseguenza, l'utente non specialista deve interpretare autonomamente le informazioni, con il rischio di applicare soluzioni non sempre pienamente conformi alle linee guida.

Alcuni software integrano suggerimenti o tutorial, ma raramente utilizzano modelli di *intelligenza artificiale* in grado di interagire con lo sviluppatore in linguaggio naturale, rispondendo a domande specifiche o guidandolo direttamente nella scrittura del codice.

“SviluppAbile” si distingue per l'integrazione di una chat *IA* che funge da assistente personale, e per la duplice modalità operativa che combina l'analisi in tempo reale con un supporto diretto allo sviluppo, colmando così una lacuna importante nelle soluzioni attualmente disponibili.

1.3 Organizzazione del testo

Il documento è diviso in sei capitoli e illustra in maniera dettagliata l'esperienza di stage svolta.

Il *secondo capitolo* illustra le informazioni raccolte in fase di analisi del progetto tramite requisiti e casi d'uso.

Il *terzo capitolo* approfondisce le tecnologie e gli strumenti utilizzati per la realizzazione del progetto.

Il [quarto capitolo](#) descrive lo sviluppo dell'estensione, evidenziando le scelte progettuali e le soluzioni implementative.

Il [quinto capitolo](#) presenta i test effettuati, volti a valutare l'effettiva utilità dell'estensione rispetto agli strumenti tradizionali di validazione del codice accessibile.

Il [sesto capitolo](#) riassume i risultati ottenuti e le possibili evoluzioni future del progetto.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- Gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- Per i termini riportati nel glossario viene utilizzata la seguente nomenclatura: *Application Programming Interface_G*;
- I termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*;
- Gli esempi di codice, le domande poste all'IA e le righe di codice utilizzano il formato `monospace`.

Capitolo 2

Analisi dei requisiti

2.1 Obiettivo del progetto

L'obiettivo principale del progetto è la realizzazione di un'estensione web interattiva con chatbot integrato, pensata per supportare gli sviluppatori nell'individuazione e nella correzione degli errori di accessibilità presenti nelle pagine web.

Il progetto si propone di rispondere alle seguenti esigenze:

- Fornire un linguaggio chiaro e facilmente comprensibile, anche per chi non ha conoscenze approfondite delle linee guida di accessibilità;
- Offrire un supporto immediato e contestuale durante lo sviluppo del codice;
- Facilitare il miglioramento della qualità e dell'accessibilità dei siti web, promuovendo buone pratiche di progettazione inclusiva.

2.2 Casi d'uso

2.2.1 Attori

Un attore rappresenta un elemento esterno al sistema, che può essere una persona o un altro sistema software, che interagisce con esso per il raggiungimento di un determinato obiettivo. L'attore non coincide quindi con un singolo utente specifico, ma con una categoria di utenti che condividono lo stesso ruolo nell'utilizzo del sistema.

Nel caso dell'estensione *SviluppAbile*, l'attore identificato è lo sviluppatore web, interessato a migliorare la qualità del codice prodotto e a garantire che i siti realizzati siano accessibili ad un pubblico ampio e diversificato.

2.2.2 Elenco e descrizione dei casi d'uso

UC0: Visualizzazione prompt iniziale

Attori principali: Sviluppatore web.

Precondizioni: L'utente ha avviato l'estensione.

Descrizione: L'estensione offre la possibilità di scegliere tra l'analisi assistita e la modalità guidata.

Postcondizioni: L'estensione è pronta per visualizzare il risultato della modalità scelta.

UC1: Modalità analisi assistita

Attori principali: Sviluppatore web.

Precondizioni: L'utente ha selezionato la modalità di analisi assistita.

Descrizione: L'estensione esegue la scansione del DOM della pagina web e mostra il codice sul pannello a sinistra. Nel pannello a destra è visibile la chat vuota con due domande suggerite e un campo di testo per l'inserimento di domande.

Postcondizioni: L'utente visualizza il DOM e la chat che potrà utilizzare.

UC1.1: Utilizzo delle domande suggerite

Attori principali: Sviluppatore web.

Precondizioni: L'utente ha avviato la modalità di analisi assistita.

Descrizione: L'utente clicca su una domanda suggerita e la invia all'IA.

Postcondizioni: L'utente riceve una risposta contestualizzata, vede eventuali righe pertinenti del DOM evidenziate e nella chat appaiono delle ulteriori domande suggerite correlate.

UC1.2: Inserimento domanda

Attori principali: Sviluppatore web.

Precondizioni: L'utente ha aperto la chat in modalità di analisi assistita.

Descrizione: L'utente inserisce manualmente una domanda nel form dedicato della chat e la invia all'IA.

Postcondizioni: L'utente riceve una risposta contestualizzata, vede eventuali righe pertinenti del DOM evidenziate e nella chat appaiono delle ulteriori domande suggerite correlate.

UC2: Modalità sviluppo guidato

Attori principali: Sviluppatore web.

Precondizioni: L'utente ha selezionato la modalità di sviluppo guidato.

Descrizione: L'estensione mostra il DOM della pagina, il pannello centrale con il codice generato e la chat interattiva.

Postcondizioni: L'utente dispone di uno spazio di sviluppo assistito in cui ricevere sia chiarimenti teorici sia frammenti di codice pronti per l'uso.

UC2.1: Utilizzo delle domande suggerite

Attori principali: Sviluppatore web.

Precondizioni: L'utente ha avviato la chat in modalità sviluppo guidato.

Descrizione: L'utente clicca su una domanda suggerita e la invia all'IA.

Postcondizioni: L'utente riceve una risposta contestualizzata, con eventuale blocco di codice visualizzato nel pannello centrale e suggerimenti per approfondire l'analisi.

UC2.2: Inserimento domanda

Attori principali: Sviluppatore web.

Precondizioni: L'utente ha aperto la chat in modalità di sviluppo guidato.

Descrizione: L'utente inserisce manualmente una domanda nel form dedicato della chat e la invia all'IA.

Postcondizioni: L'utente riceve una risposta contestualizzata, con eventuale blocco di codice visualizzato nel pannello centrale e suggerimenti per approfondire l'analisi.

UC2.2: Copia di un blocco di codice

Attori principali: Sviluppatore web.

Precondizioni: L'IA ha generato almeno un blocco di codice in risposta a una domanda dell'utente.

Descrizione: L'estensione visualizza il blocco di codice numerato; l'utente può copiarlo negli appunti per utilizzarlo immediatamente nel proprio progetto.

Postcondizioni: Il frammento selezionato è disponibile negli appunti dell'utente.

UC2.3: Download del codice generato

Attori principali: Sviluppatore web.

Precondizioni: L'estensione ha generato uno o più blocchi di codice in modalità guidata.

Descrizione: L'utente clicca il pulsante *“Scarica codice”*; l'estensione concatena tutti i blocchi generati e produce un file HTML scaricabile.

Postcondizioni: L'utente ottiene un file contenente tutti i suggerimenti di codice prodotti dall'IA.

UC3: Blocco estensione su pagine di sistema

Attori principali: Sviluppatore web.

Precondizioni: L'utente tenta di avviare l'estensione in una pagina di sistema del browser (es. `chrome://`, `edge://`, `about:`, nuova scheda).

Descrizione: L'estensione rileva che la pagina corrente appartiene alle aree riservate del browser, non accessibili alle estensioni. L'estensione apre un popup informativo che comunica all'utente l'impossibilità di utilizzare *SviluppAbile* in quella pagina.

Postcondizioni: L'utente è informato del blocco e comprende che l'estensione può essere utilizzata solo su pagine web ordinarie.

UC4: Gestione errori di comunicazione con l'IA

Attori principali: Sviluppatore web.

Precondizioni: L'utente ha inviato una richiesta all'IA tramite chat (in modalità analisi assistita o sviluppo guidato).

Descrizione: L'estensione gestisce risposte non valide o errori di parsing JSON provenienti dal server dell'IA notificando l'utente.

Postcondizioni: L'utente è consapevole dell'errore e può ripetere la richiesta o verificare la configurazione del sistema.

2.3 Tracciamento dei requisiti

Dall'analisi dei requisiti e degli UC effettuata sono emersi dei requisiti di diverso tipo, ai quali è stato associato un codice identificativo per distinguerli. Si farà riferimento ai requisiti secondo la seguente classificazione:

- **RO** per i requisiti obbligatori, vincolanti in quanto obiettivo primario richiesto dal committente;

- **RD** per i requisiti desiderabili, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto;
- **RF** per i requisiti facoltativi, rappresentanti valore aggiunto non strettamente competitivo.

Le sigle precedentemente indicate saranno seguite da un trattino e da una coppia sequenziale di numeri, per identificare il singolo requisito in maniera univoca.

2.3.1 Requisiti obbligatori

Tabella 2.1: Tabella del tracciamento dei requisiti obbligatori

Requisito	Descrizione	Fonti
RO-01	L'estensione deve offrire un prompt iniziale che consenta all'utente di scegliere tra modalità analisi assistita e modalità sviluppo guidato.	UC0
RO-02	In modalità analisi assistita, l'estensione deve permettere di visualizzare il DOM numerato della pagina e di porre domande all'IA.	UC1
RO-03	In modalità sviluppo guidato, l'estensione deve mostrare il DOM numerato, il pannello centrale con i suggerimenti di codice e la chat laterale.	UC2
RO-04	L'estensione deve impedire l'avvio su pagine di sistema del browser (es. chrome://, about:blank, nuova scheda).	UC3
RO-05	L'estensione deve gestire eventuali errori di comunicazione con l'IA mostrando messaggi esplicativi all'utente.	UC4

2.3.2 Requisiti desiderabili

Tabella 2.2: Tabella del tracciamento dei requisiti desiderabili

Requisito	Descrizione	Fonti
RD-01	L'estensione deve proporre domande suggerite per facilitare l'interazione con l'IA in entrambe le modalità.	UC1.1, UC2.1
RD-02	In modalità analisi assistita, l'estensione deve evidenziare nel codice le righe pertinenti alla risposta dell'IA.	UC1.2
RD-03	In modalità sviluppo guidato, l'IA deve generare suggerimenti di codice accessibile numerati e separati in blocchi.	UC2
RD-04	L'utente deve poter scaricare un file HTML contenente tutti i blocchi di codice generati.	UC2.3

2.3.3 Requisiti facoltativi

Tabella 2.3: Tabella del tracciamento dei requisiti facoltativi

Requisito	Descrizione	Fonti
RF-01	L'estensione deve supportare una modalità "notte" per migliorare la leggibilità e ridurre l'affaticamento visivo.	
RF-02	L'estensione deve supportare il cambio della lingua dell'interfaccia dall'italiano all'inglese.	
RF-03	L'estensione deve mantenere la cronologia delle interazioni con l'IA nella sessione corrente.	

Capitolo 3

Tecnologie

3.1 Linguaggi

3.1.1 HTML

HTML (*HyperText Markup Language*) è il linguaggio di marcatura standard utilizzato per strutturare i contenuti delle pagine web. Non si tratta di un linguaggio di programmazione, ma di uno strumento che consente di definire gli elementi presenti su una pagina (come titoli, testi, immagini, link, *form_G*) assegnando loro un significato semantico tramite l'uso di tag.

Introdotta nel 1991 da Tim Berners-Lee, *HTML* si è evoluta nel tempo attraverso diverse versioni, fino a diventare uno standard globale supervisionato dal *World Wide Web Consortium* (*W3C_G*).

La struttura di un file *HTML* si basa su una serie di elementi, rappresentati attraverso tag, che permettono di definire la natura e la funzione dei contenuti all'interno di una pagina web. Ogni tag descrive una tipologia specifica di contenuto o un comportamento associato, contribuendo all'organizzazione semantica del documento. I tag sono delimitati da parentesi angolari (< >) e, nella maggior parte dei casi, sono utilizzati in coppie: un tag di apertura e uno di chiusura, quest'ultimo contraddistinto da una barra (</tag>). Alcuni elementi, tuttavia, non necessitano di chiusura e sono detti auto-chiudenti, come ad esempio per l'inserimento di immagini, o
 per i ritorni a capo. Questa struttura gerarchica e nidificata consente al *browser_G* di interpretare correttamente i contenuti, garantendo coerenza tra presentazione e significato.

La versione attuale, HTML5 (logo in figura 3.1), è stata rilasciata in modo stabile

nel 2014. HTML5 rappresenta un'evoluzione del linguaggio di *markup*_G con l'obiettivo di ridurre la dipendenza da *plugin* esterni, introducendo nuove funzionalità native. Tra le principali novità vi sono gli elementi `<canvas>`, `<video>` e `<audio>`, nuovi tag semantici per strutturare i contenuti (come articoli, intestazioni, sezioni) e controlli avanzati per i *form* (email, *URL*_G, numeri, date, ricerca). Vengono inoltre introdotti strumenti per la memorizzazione locale dei dati sul client tramite `localStorage` e `sessionStorage`.



Figura 3.1: Logo HTML5

3.1.2 CSS

CSS (*Cascading Style Sheets*) è un linguaggio utilizzato per definire la presentazione dei documenti *HTML* e *XML*_G. Introdotto nel 1996 da Håkon Wium Lie, consente di separare la struttura dei contenuti dalla loro formattazione, migliorando la chiarezza del codice e facilitandone la manutenzione.

I fogli di stile permettono di specificare, tramite regole composte da selettori, proprietà e valori, l'aspetto degli elementi *HTML*: layout, colori, font, margini, spaziature, effetti visivi, animazioni e molto altro. Le regole di stile possono essere inline (tramite tag `<style>` o con attributi `style`) oppure riportate in file esterni `.css` riutilizzabili. La prima soluzione non mantiene però una netta separazione tra struttura e contenuto, non rispettando quindi le best-practice del settore.

L'attuale versione, CSS3 (logo in figura 3.2), è modulare e introduce importanti funzionalità come `flexbox`, `grid`, media query per il responsive design, transizioni, trasformazioni e nuovi selettori.



Figura 3.2: Logo CSS3

3.1.3 Prism.js

Prism.js è una libreria JavaScript leggera ed estensibile progettata per evidenziare la sintassi del codice sorgente in modo chiaro e leggibile, rispettando gli standard moderni del web. Supporta un'ampia gamma di linguaggi di programmazione e di *markup*, ed è facilmente personalizzabile tramite temi e *plugin*.

Grazie alla sua efficienza e semplicità di integrazione, viene utilizzata in milioni di siti web per migliorare la leggibilità del codice, rendendolo più comprensibile sia a sviluppatori che a utenti. E' disponibile in diverse varianti, sia per il white-mode che per il dark-mode.

Nella mia estensione, Prism.js è stato integrato per colorare il codice sorgente del *DOM*, facilitando l'analisi e la comprensione della struttura *HTML*.

3.1.4 JavaScript

JavaScript (logo in figura 3.3) è un linguaggio di programmazione dinamico, interpretato ed orientato agli oggetti, progettato originariamente per rendere le pagine web interattive. È stato sviluppato nel 1995 da Brendan Eich per Netscape con il nome LiveScript, e successivamente rinominato in JavaScript per motivi di marketing, pur non avendo legami diretti con il linguaggio Java. La sua standardizzazione è gestita da *ECMA International*, sotto il nome ECMAScript.

Concepito per essere eseguito direttamente all'interno del *browser*, JavaScript consente di scrivere *script*_G incorporati nel codice *HTML*, eseguibili senza compilazione

al momento del caricamento della pagina. Questi *script* permettono di manipolare il *DOM* (Document Object Model), reagire agli eventi dell'utente (come click, input da tastiera o movimenti del mouse), modificare dinamicamente il contenuto della pagina e gestire funzionalità come validazioni, messaggi, animazioni, e interazioni *asincrone*. Oltre alla manipolazione di elementi visivi, JavaScript permette di sfruttare numerose *API* messe a disposizione dal *browser*, dalla gestione del mouse e del touch, alla manipolazione delle immagini, fino alla gestione locale dei dati attraverso meccanismi come il `LocalStorage`. Questa versatilità lo rende uno strumento fondamentale per lo sviluppo di applicazioni web moderne.

JavaScript si è evoluto da semplice linguaggio *client-side* a tecnologia completa e versatile, oggi utilizzabile anche su *server* (ad esempio con *Node.js*) e in ambienti esterni ai *browser* grazie all'uso di diversi motori JavaScript i quali permettono l'esecuzione di codice *JS* con alte prestazioni, rendendo il linguaggio adatto anche a contesti *backend*, applicazioni desktop e mobile.

Oggi JavaScript rappresenta uno dei tre pilastri fondamentali del web, insieme a *HTML* e *CSS*, ed è completamente integrato con essi. Esistono anche linguaggi alternativi (come TypeScript o CoffeeScript) che vengono compilati in JavaScript, offrendo funzionalità aggiuntive mantenendo la compatibilità con l'ambiente JavaScript standard.



Figura 3.3: Logo JavaScript

3.2 Tecnologie e strumenti

3.2.1 Chrome Extension

Le estensioni di Chrome sono piccoli programmi software che consentono di arricchire e personalizzare il *browser*, aggiungendo funzionalità aggiuntive o migliorando quelle già esistenti.

Grazie ad esse è possibile modificare l'interfaccia utente, automatizzare attività ripetitive, integrare servizi esterni e rendere più efficiente la navigazione sul web.

La distribuzione delle estensioni avviene principalmente tramite il *Chrome Web Store*, ma durante le fasi di sviluppo è possibile installarle manualmente utilizzando la modalità sviluppatore disponibile nel *browser*.

3.2.1.1 Manifest V3

Manifest V3 (*MV3_G*) è la specifica più recente per la creazione di estensioni per Chrome e altri *browser* basati su *Chromium*.

Rispetto alla precedente Manifest V2, l'ultima versione introduce diverse modifiche per migliorare la sicurezza, le prestazioni e la privacy delle estensioni [5].

Questa nuova versione è una risposta alle criticità emerse con il Manifest V2, e introduce cambiamenti strutturali significativi nel modo in cui le estensioni interagiscono con il *browser* e le pagine web.

E' importante notare che Manifest V3 non è retrocompatibile con le estensioni V2 senza modifiche sostanziali e che la versione è tuttora in evoluzione, poiché alcune funzionalità sono ancora in fase di implementazione.

Il file `manifest.json` in *MV3* continua a rappresentare il cuore dell'estensione, definendo metadati fondamentali come nome, versione, permessi richiesti e *script* da eseguire. Tuttavia, rispetto a *MV2_G*, *MV3* impone restrizioni più rigide su quali *API* possono essere utilizzate e introduce un modello di esecuzione più efficiente e sicuro.

Gli elementi principali di un'estensione basata su Manifest V3 sono:

- **Manifest:** come nelle versioni precedenti, il file `manifest.json` contiene tutte le informazioni necessarie per il funzionamento dell'estensione, inclusi i permessi richiesti. Tra i permessi più rilevanti vi sono:
 - `downloads`: permette all'estensione di avviare e gestire download di file, ad esempio per salvare contenuti generati o analizzati;
 - `scripting`: consente l'esecuzione di *script* personalizzati nelle pagine web, utile per analizzare o modificare il *DOM*;
 - `activeTab`: garantisce l'accesso temporaneo alla scheda attiva dopo un'interazione dell'utente con l'estensione;
 - `tabs`: fornisce informazioni sulle schede del *browser* e permette di interagire con esse, come ottenere *URL* o titoli;
 - `storage`: consente di memorizzare e recuperare dati locali, ad esempio impostazioni o risultati di analisi.
- **Service Worker:** in Manifest V3, i tradizionali background *script* sono sostituiti da service worker. Questi sono *script* che vengono eseguiti in background ma solo quando necessario, rispondendo a eventi come le richieste di rete o azioni dell'utente. A differenza dei background *script* di *MV2*, i service worker non sono persistenti e vengono sospesi quando inattivi, riducendo così l'utilizzo di risorse e migliorando le prestazioni complessive. I service worker non possono accedere direttamente al *DOM* delle pagine, ma comunicano con i content *script* tramite messaggi.
- **Content Script:** vengono iniettati direttamente nelle pagine web e possono interagire con il *DOM*. In *MV3*, anche i content *script* sono soggetti a restrizioni più severe per evitare conflitti con il contenuto della pagina e migliorare la sicurezza.

Il Manifest V3 introduce modifiche significative nell'ecosistema delle estensioni per *browser*, con effetti sia positivi sia problematici.

Aspetti positivi

Tra i vantaggi più evidenti vi è il blocco del codice remoto, che impedisce alle estensioni di scaricare ed eseguire codice non verificato dopo l'installazione. Questa misura contribuisce a ridurre alcuni rischi di sicurezza, proteggendo gli utenti da esecuzioni di codice potenzialmente dannoso.

Aspetti critici

Il nuovo standard non elimina completamente i rischi: le *API* continuano a consentire la raccolta di dati sensibili (come la cronologia di navigazione) lasciando alcune vulnerabilità inalterate.

Dal punto di vista dello sviluppo, *MV3* comporta limitazioni importanti: l'*API* `webRequest`, utilizzata per modificare dinamicamente le richieste di rete, è stata sostituita da `declarativeNetRequest`, più rigida e meno flessibile, riducendo le possibilità di innovazione.

Inoltre, le pagine background persistenti vengono sostituite dai service worker, che presentano vincoli sulle *API* disponibili e problemi di compatibilità con funzioni avanzate (come *WebSockets_G*, *WebAssembly_G* o la localizzazione delle estensioni).

La migrazione a *MV3* si è dimostrata complessa e molte funzionalità di estensioni preesistenti rischiano di rompersi o di subire un peggioramento delle prestazioni. Infine, il processo decisionale di Google è stato percepito come scarsamente collaborativo, poiché l'implementazione del nuovo standard è stata imposta senza un reale dialogo con la comunità degli sviluppatori [6].

In sintesi, sebbene il blocco del codice remoto rappresenti un progresso in termini di sicurezza, Manifest V3 introduce restrizioni che limitano innovazione, privacy e performance, portando l'*Electronic Frontier Foundation* a definirlo un atto ostile verso gli utenti, frutto di una posizione dominante di Google nell'ecosistema delle estensioni web.

3.2.2 Ollama

Ollama (logo in figura 3.4) è una piattaforma che consente di eseguire localmente modelli di *intelligenza artificiale* avanzati, sviluppati in collaborazione con OpenAI e altri partner, mantenendo alte prestazioni anche senza l'uso del *cloud computing*_G. Oltre ai modelli gpt-oss-20B e gpt-oss-120B, progettati rispettivamente per scenari a bassa *latenza*_G o per applicazioni generali ad alto livello di ragionamento, Ollama supporta numerosi altri modelli *open weight*_G, tra cui Mistral (utilizzato solo all'inizio a causa delle ridotte capacità del mio pc personale), Llama 3.1:8B e Llama 3.2:3B, che ho utilizzato nel mio progetto. Nella tabella sottostante 3.1 vengono messi a confronto i vari modelli di Ollama utilizzati nei test dell'estensione *SviluppAbile*, sulla base di parametri quali la capacità di ragionamento complesso, la comprensione linguistica, la conoscenza generale, la velocità di risposta e i requisiti hardware.

Tabella 3.1: Confronto modelli Ollama utilizzati

Versione	Llama 3.1:8b	Llama 3.2:3b	Mistral
Ragionamento complesso	Molto elevato, ottimo su coding, Q&A, compiti multilingue complessi	Buono ma inferiore ai modelli più grandi, migliore per compiti rapidi	Limitato rispetto a Llama 8b, pensato per compiti specifici
Comprensione linguistica	Alta precisione e coerenza, anche in testi lunghi	Discreta, ma può generare errori grammaticali o essere meno coerente in testi lunghi	Buona solo in testi brevi e semplici
Conoscenza generale	Ampia copertura di argomenti	Più ridotta per via del minor numero di parametri	Limitata
Velocità di risposta	Più lenta su hardware modesto	Rapida con HW adeguato	Molto rapida

– continua nella pagina successiva

Tabella 3.1 – continua dalla pagina precedente

Versione:	Llama 3.1:8b	Llama 3.2:3b	Mistral
Requisiti hardware	Almeno 8 GB RAM + richiede <i>GPU_G</i> di fascia media/alta	Almeno 4 GB RAM	Almeno 2 GB RAM

La piattaforma integra funzionalità avanzate (function calling, navigazione web integrata, esecuzione di codice Python, output strutturati), accesso completo al processo di ragionamento del modello, regolazione dello sforzo computazionale e possibilità di fine-tuning per personalizzare le prestazioni. Grazie al supporto nativo per la *quantizzazione_G* in formato MXFP4, Ollama riesce a ridurre drasticamente il consumo di memoria, permettendo l'esecuzione di modelli molto grandi anche su sistemi con risorse limitate. La licenza Apache 2.0 garantisce libertà di utilizzo e personalizzazione, mentre l'ottimizzazione per *GPU* NVIDIA GeForce RTX e RTX PRO assicura alte prestazioni in locale. .



Figura 3.4: Logo Ollama

3.2.3 GitHub

GitHub (logo in figura 3.5) è una piattaforma di hosting basata su *Git_G*, il sistema di controllo di versione distribuito ideato da Linus Torvalds. Lanciata nel 2008, GitHub si è affermata come uno degli strumenti principali per la collaborazione nello sviluppo software, consentendo a sviluppatori di tutto il mondo di condividere, modificare e mantenere progetti in modo coordinato.

La piattaforma offre funzionalità avanzate per il versionamento del codice, la gestione dei rami (branching), il tracciamento delle modifiche e la revisione collaborativa tramite

pull request.

Oltre a essere uno strumento tecnico, GitHub ha favorito la nascita di una vera e propria comunità *open source*_G, nella quale il codice è accessibile, riutilizzabile e migliorabile da chiunque.

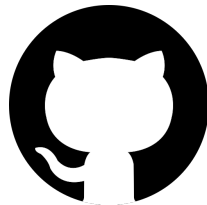


Figura 3.5: Logo GitHub

3.2.4 VSCode

Visual Studio Code (logo in figura 3.6), conosciuto semplicemente come VSCode, è un editor di codice sorgente sviluppato da Microsoft e distribuito gratuitamente. Introdotto nel 2015, ha rapidamente conquistato una posizione di rilievo tra gli strumenti preferiti dagli sviluppatori grazie alla sua combinazione di leggerezza, flessibilità e ricchezza di funzionalità.

Compatibile con i principali sistemi operativi (Windows, macOS e Linux), VSCode supporta evidenziazione della sintassi per numerosi linguaggi, suggerimenti intelligenti e completamento automatico del codice, strumenti di *debugging*_G integrati e gestione del controllo di versione tramite *Git*.

Inoltre, grazie a un vasto marketplace di estensioni, può essere facilmente personalizzato per adattarsi a diversi flussi di lavoro e tecnologie, rendendolo un ambiente di sviluppo versatile e ampiamente diffuso.



Figura 3.6: Logo VSCode

3.2.5 Total Validator

Total Validator (logo in figura 3.7) è uno strumento di convalida web che permette di verificare l'accessibilità secondo le linee guida WCAG 2.0, 2.1 e 2.2, la conformità alla Section 508 statunitense e alle specifiche ARIA.

Consente inoltre di controllare la validità del codice HTML rispetto a numerosi standard, incluso l'HTML5, e la correttezza dei CSS in conformità con molte specifiche del W3C.

Tra le sue funzioni principali rientrano anche la rilevazione di link interrotti e la verifica ortografica multilingue. Lo strumento permette di analizzare pagine offline, protette da autenticazione o generate dinamicamente tramite JavaScript, risultando così particolarmente flessibile.

È disponibile per Windows, macOS e Linux e, dal suo lancio nel 2005, è stato adottato da milioni di siti web, comprese istituzioni pubbliche e governative.



Figura 3.7: Logo Total Validator

3.2.6 WAVE

WAVE (*Web Accessibility Evaluation Tool*) è uno strumento sviluppato da WebAIM per supportare la valutazione dell'accessibilità dei siti web (logo in figura 3.8). Consente di individuare automaticamente molte violazioni delle WCAG e, al tempo stesso, facilita l'analisi manuale da parte degli sviluppatori.

È disponibile come servizio online e come estensione per i principali browser, permettendo di testare anche pagine locali o protette. I risultati vengono visualizzati direttamente sulla pagina tramite icone e pannelli che segnalano errori, avvisi e caratteristiche accessibili, fornendo indicazioni utili per le correzioni.

Per contesti più complessi, WAVE offre inoltre un'API che ne consente l'integrazione nei processi di sviluppo e di testing automatizzato.



Figura 3.8: Logo WAVE

3.2.7 Lighthouse

Lighthouse (logo in figura 3.9) è uno strumento open-source sviluppato da Google per valutare la qualità delle pagine web. Esegue controlli automatici su performance, accessibilità, SEO e progressive web app, restituendo report dettagliati con punteggi e suggerimenti pratici.

Si può utilizzare tramite Chrome DevTools, riga di comando (CLI), modulo Node o interfaccia web, consentendo anche audit su pagine protette o locali. Lighthouse quindi supporta gli sviluppatori nel miglioramento della velocità di caricamento, dell'usabilità e della visibilità dei siti.



Figura 3.9: Logo Lighthouse

Capitolo 4

Sviluppo del progetto di stage

4.1 Web design

L'estensione è stata progettata con un approccio *desktop-first_G*, in considerazione del target principale costituito da sviluppatori web che operano prevalentemente su schermi di grandi dimensioni.

Tale scelta consente di privilegiare la chiarezza e l'ampiezza degli spazi di lavoro, garantendo una disposizione ottimale dei pannelli e delle funzionalità principali.

Particolare attenzione è stata posta all'accessibilità cromatica, mediante uno studio accurato dei contrasti tra testo, sfondo ed elementi interattivi, al fine di assicurare leggibilità anche in condizioni visive differenti.

È stata inoltre implementata la possibilità di personalizzare l'esperienza visiva tramite un pulsante dedicato alla selezione della modalità giorno/notte, che permette all'utente di adattare l'interfaccia alle proprie preferenze e alle condizioni ambientali di utilizzo. Nonostante l'approccio iniziale privilegi i dispositivi desktop, il layout rimane responsivo grazie a soluzioni flessibili che mantengono l'usabilità anche su schermi di dimensioni ridotte. Questa scelta assicura un'esperienza coerente ed accessibile in diversi contesti d'uso.

4.2 PoC

L'obiettivo iniziale era la realizzazione di un *Proof of Concept_G* (*PoC_G*) che permettesse di verificare la fattibilità tecnica dell'estensione proposta. In questa fase preliminare, l'attenzione era rivolta principalmente a due aspetti fondamentali: il recupero del codice sorgente della pagina web e l'integrazione con un'*API* di *intelligenza artificiale*, necessaria per avviare i primi test di analisi e generazione di suggerimenti.

Il *PoC* non teneva conto di aspetti come la scalabilità, l'ottimizzazione delle performance o la gestione di casi d'uso complessi, ma si limitava a dimostrare che le funzionalità di base potessero essere effettivamente implementate.

La struttura iniziale era volutamente semplice in quanto conteneva solamente funzioni basilari nei file:

- `script.js`: responsabile dell'avvio dell'estensione;
- `analysis.js`: deputato all'estrazione del codice sorgente della pagina web e alla sua messa a disposizione per ulteriori elaborazioni;
- `service_worker.js`: gestisce le richieste verso il modello di *intelligenza artificiale*, inviando il codice recuperato e ricevendo in risposta i suggerimenti generati.

Questa versione sperimentale costituiva una base minima ma solida su cui successivamente è stato possibile costruire le funzionalità avanzate, affinare l'interfaccia e integrare logiche più complesse per la gestione dei risultati.

È importante sottolineare che Chrome, per ragioni di sicurezza, blocca di default l'esecuzione di chiamate esterne da parte delle estensioni, impedendo quindi il corretto funzionamento dell'integrazione con Ollama. Per avviare correttamente l'estensione è necessario aprire Chrome da terminale e disabilitare temporaneamente alcuni vincoli di sicurezza.

Ecco il comando utilizzato:

```
chrome.exe --disable-web-security --user-data-dir="C:_temp_chrome"
```

4.3 Prodotto finale

In questo paragrafo viene illustrato il risultato concreto del progetto: l'estensione web *SviluppAbile* (logo in figura 4.1), uno strumento progettato per supportare gli sviluppatori nell'individuazione e nella correzione degli errori di accessibilità nelle pagine web.

Di seguito, viene descritta l'estensione realizzata e vengono analizzate le funzionalità implementate, mettendo in evidenza l'integrazione con il *chatbot* e le modalità di interazione con l'utente.



Figura 4.1: Logo dell'estensione *SviluppAbile*

4.3.1 Guida all'utilizzo

La presente sezione illustra le funzionalità dell'estensione attraverso una dimostrazione pratica guidata, supportata da screenshot dell'interfaccia per mostrare concretamente le modalità di utilizzo. Ogni sottosezione è dedicata a una specifica funzionalità. Si tenga presente che l'estensione è disponibile anche in modalità notturna, cliccando l'apposito pulsante in alto a destra (vedi figura 4.2).



Figura 4.2: Pulsante per il cambio modalità diurna-notturna

4.3.1.1 Avvio dell'estensione

All'avvio, cliccando sull'icona dell'estensione *SviluppAbile* (figura 4.1) nella barra del *browser*, viene visualizzato un popup (figura 4.3) che consente di scegliere tra due modalità operative:

- **Analisi assistita:** visualizza il *DOM* della pagina web e mette a disposizione una chat in cui l'utente può porre domande sull'accessibilità e ricevere risposte contestualizzate, con evidenziazione delle porzioni di codice rilevanti;
- **Modalità guidata:** presenta, oltre alla chat, un pannello in cui verranno inseriti frammenti di codice accessibile e scaricabili in formato *HTML*.



Figura 4.3: Popup di avvio dell'estensione *SviluppAbile*

Questa scelta iniziale permette all'utente di adattare lo strumento al proprio flusso di lavoro: analizzare passo-passo una pagina già esistente o sviluppare in maniera assistita un nuovo contenuto.

4.3.1.2 Analisi assistita

Nella modalità di analisi assistita (vedi figure 4.4 e 4.5), l'interfaccia dell'estensione si presenta suddivisa in due pannelli principali: sulla sinistra viene visualizzato il *DOM* della pagina web analizzata, mentre sulla destra compare la finestra della chat. Quest'ultima, inizialmente vuota, offre già due domande (*"Come posso migliorare l'accessibilità della pagina web?"* e *"Il mio codice rispetta le linee guida WCAG?"*), pensate per orientare l'utente nel primo utilizzo e stimolare un'interazione immediata. Oltre a tali suggerimenti, è disponibile un campo di testo in cui l'utente può inserire manualmente le proprie domande o richieste di chiarimento.

CAPITOLO 4. SVILUPPO DEL PROGETTO DI STAGE

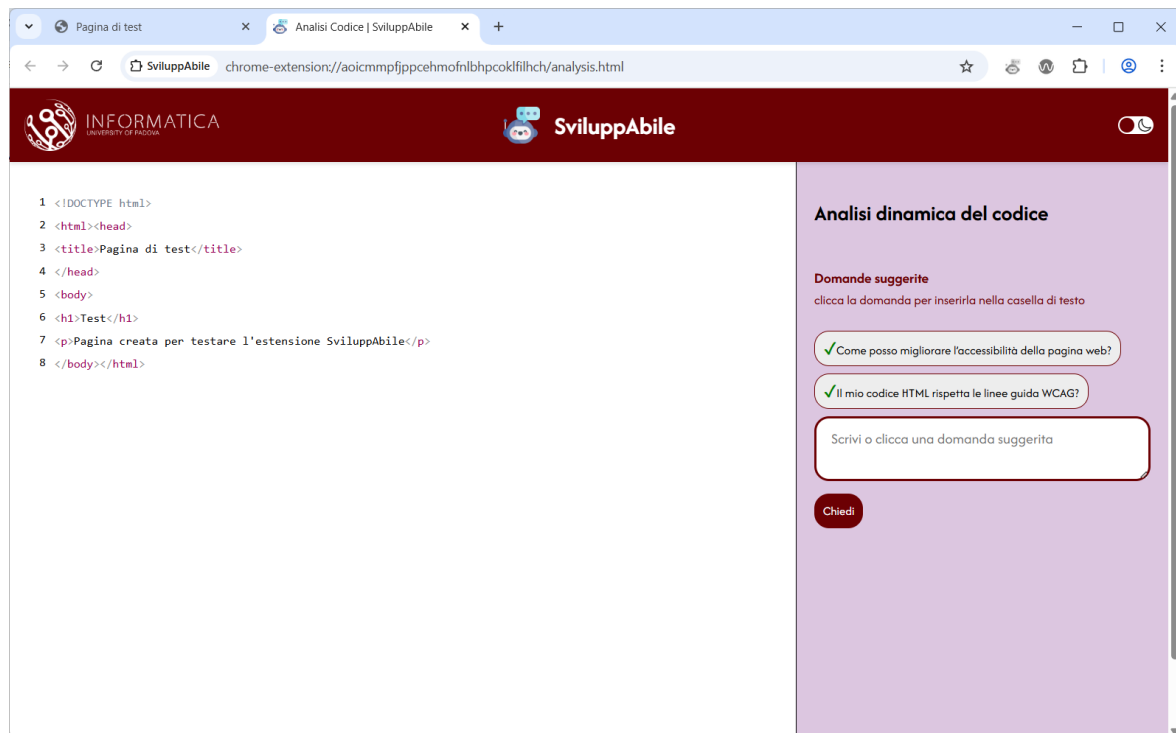


Figura 4.4: Modalità: analisi assistita

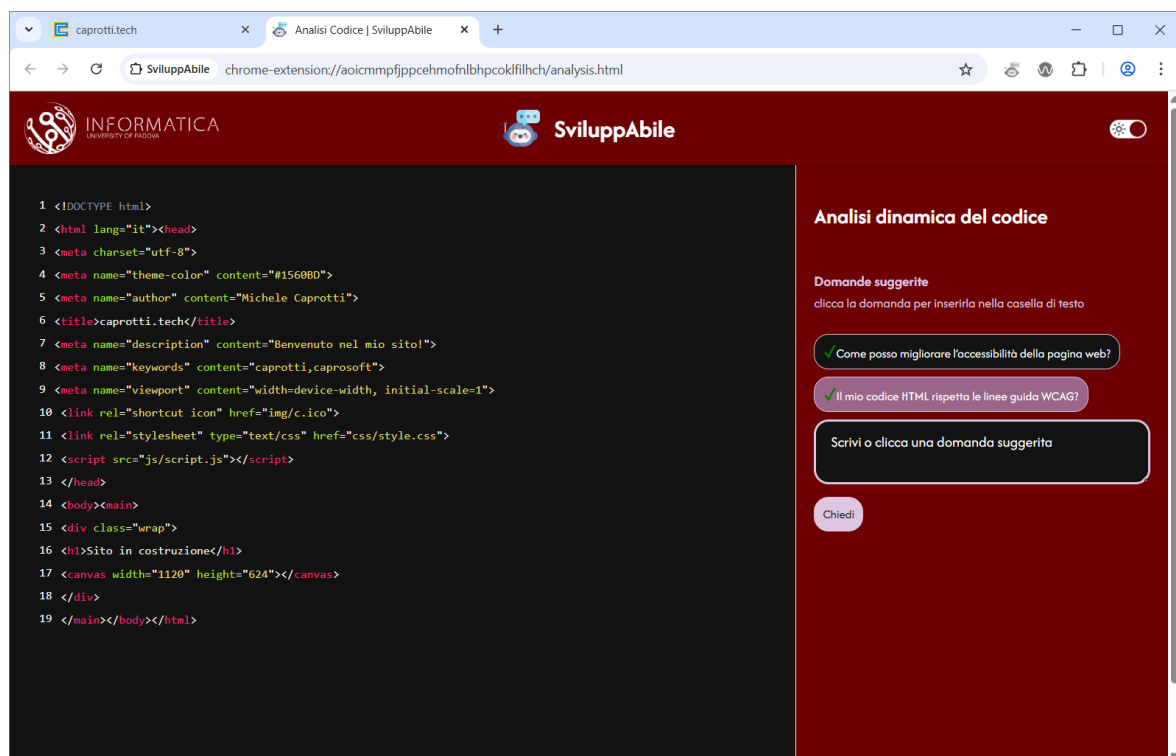


Figura 4.5: Modalità: analisi assistita - modalità notturna

L'interazione vera e propria inizia quando l'utente formula la prima domanda: a questo punto il *chatbot* genera una risposta in linguaggio naturale, adattata al contesto del *DOM* analizzato. Per facilitare la comprensione delle spiegazioni, l'estensione è in grado di evidenziare automaticamente le porzioni di codice rilevanti all'interno del *DOM*, così da collegare visivamente l'errore o la soluzione proposta con la parte effettiva di codice interessata.

Ad ogni scambio successivo, il sistema arricchisce la conversazione proponendo una o due nuove domande suggerite, collegate al tema trattato nella risposta precedente. Questa dinamica consente di mantenere la chat scorrevole e guidata, riducendo lo sforzo cognitivo dell'utente e favorendo un approccio progressivo alla comprensione delle problematiche di accessibilità della pagina web (vedi figura 4.6).

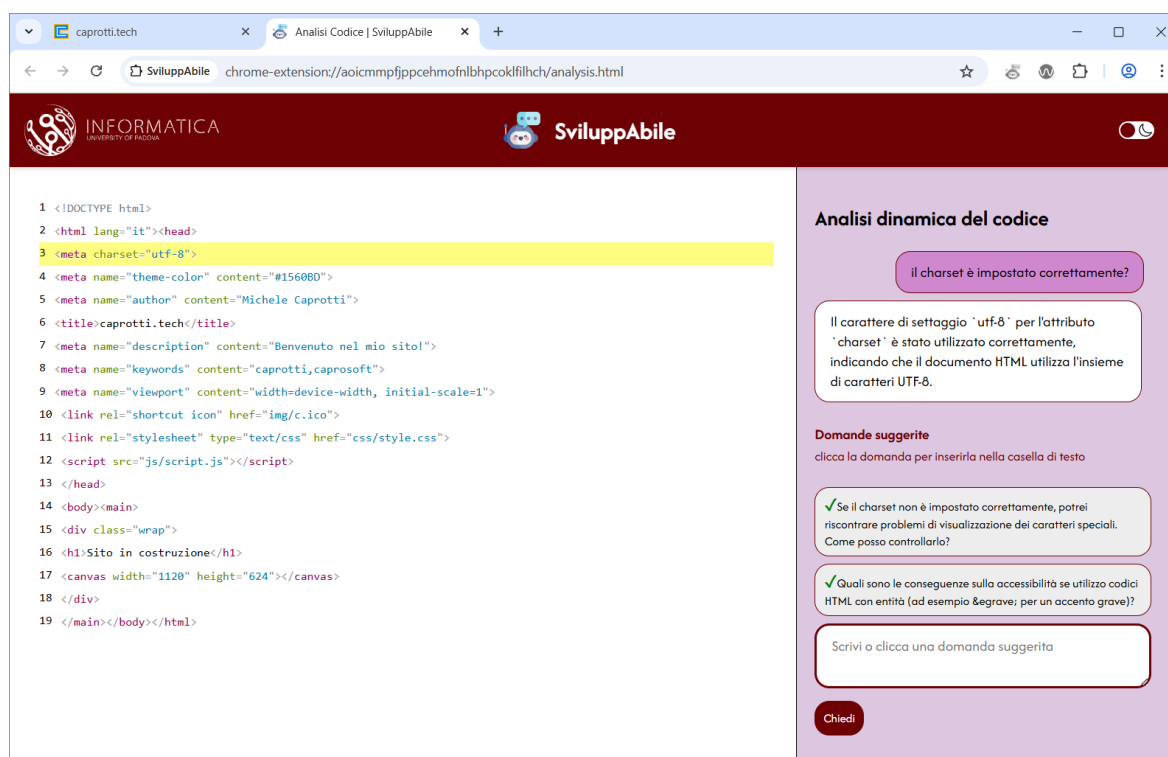


Figura 4.6: Modalità: analisi assistita dopo un'interazione

4.3.1.3 Modalità guidata

La modalità di sviluppo guidato (vedi figure 4.7 e 4.8) si differenzia dall'analisi assistita per la presenza di un'interfaccia composta da tre pannelli distinti. Sulla sinistra rimane visibile il *DOM* della pagina mentre sulla destra è presente la finestra della chat. La novità è costituita dal pannello centrale, dedicato alla visualizzazione dei frammenti di codice suggeriti dal *chatbot*.

Inizialmente, quest'area centrale appare vuota e contiene soltanto un messaggio in grigio, con caratteri a richiamo "informatico", che riporta la dicitura **eventuale codice suggerito**. Come nella modalità assistita, anche qui la chat propone due domande iniziali suggerite per avviare l'interazione: *"Come si crea un <head> che contenga tutti gli elementi utili per una pagina accessibile?"* e *"Come rendo accessibili le immagini?"*.

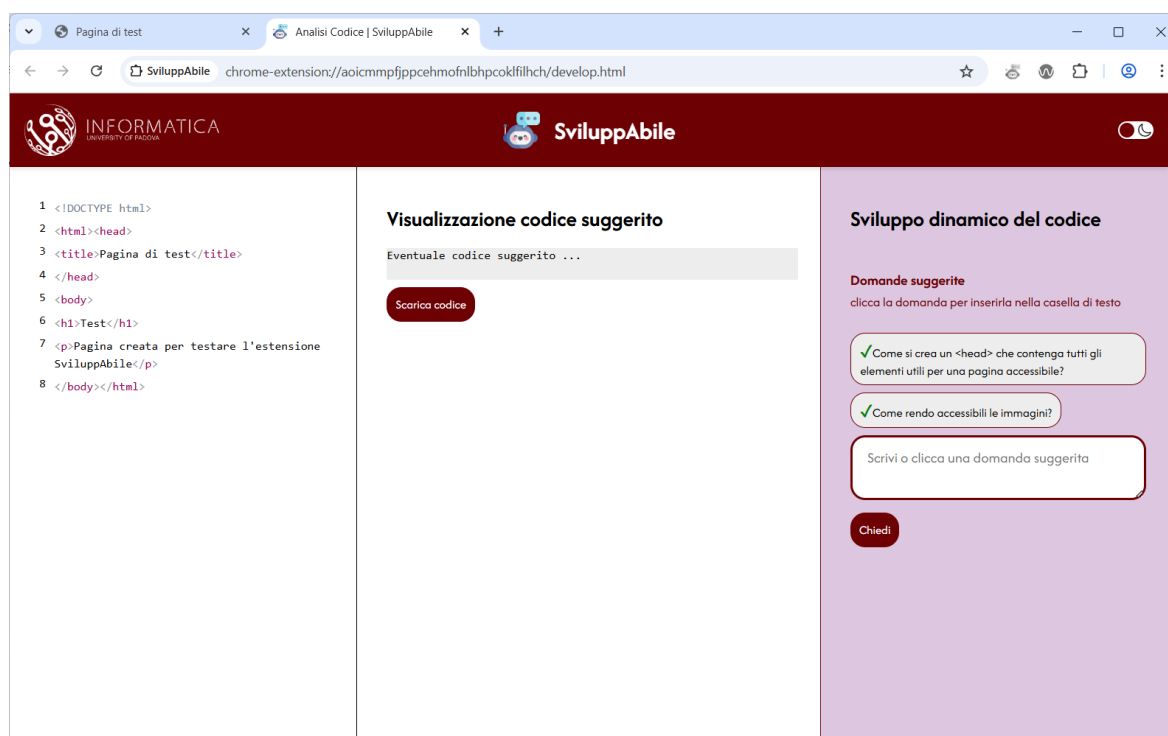


Figura 4.7: Modalità: sviluppo guidato, pagina iniziale

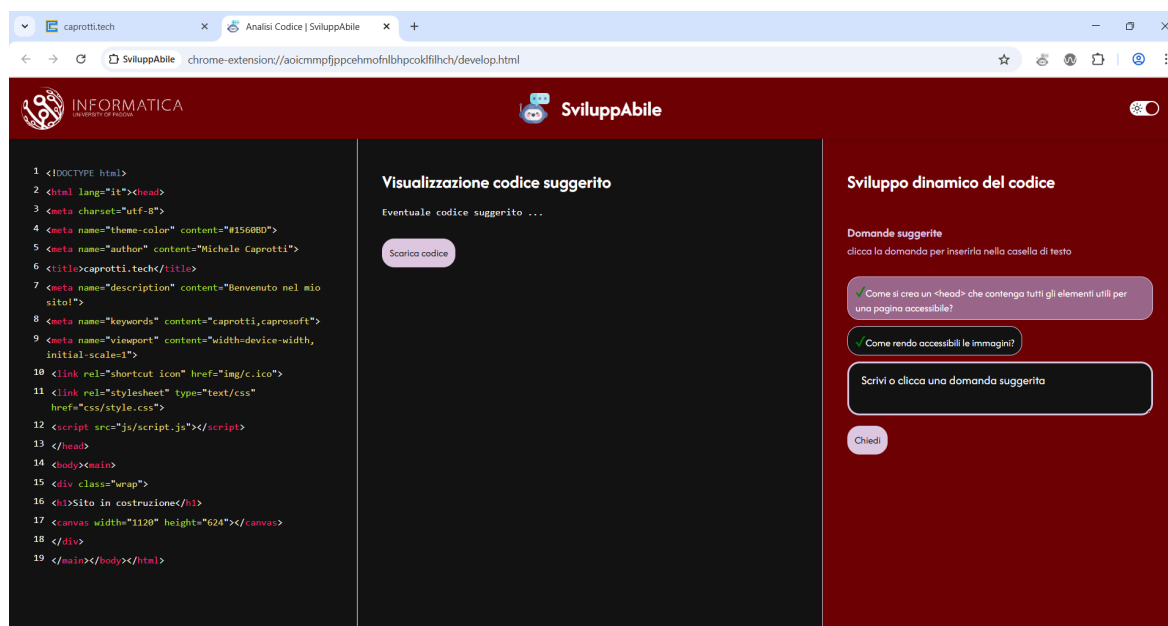


Figura 4.8: Modalità: sviluppo guidato - modalità notturna

Quando l'utente seleziona una domanda, il *chatbot* non solo produce una risposta testuale, ma genera anche un blocco di codice che viene immediatamente visualizzato nel pannello centrale. Questa caratteristica rende la modalità guidata particolarmente utile in fase di sviluppo, poiché consente di osservare concretamente le soluzioni suggerite e di integrarle facilmente nel proprio progetto.

Per agevolare l'utilizzo pratico, ogni codice prodotto può essere scaricato direttamente tramite l'apposito pulsante "Scarica codice" (vedi figura 4.9). L'estensione genera automaticamente un file *HTML* contenente i blocchi di codice proposti, consentendo allo sviluppatore di salvare e riutilizzare le soluzioni in maniera rapida e strutturata.

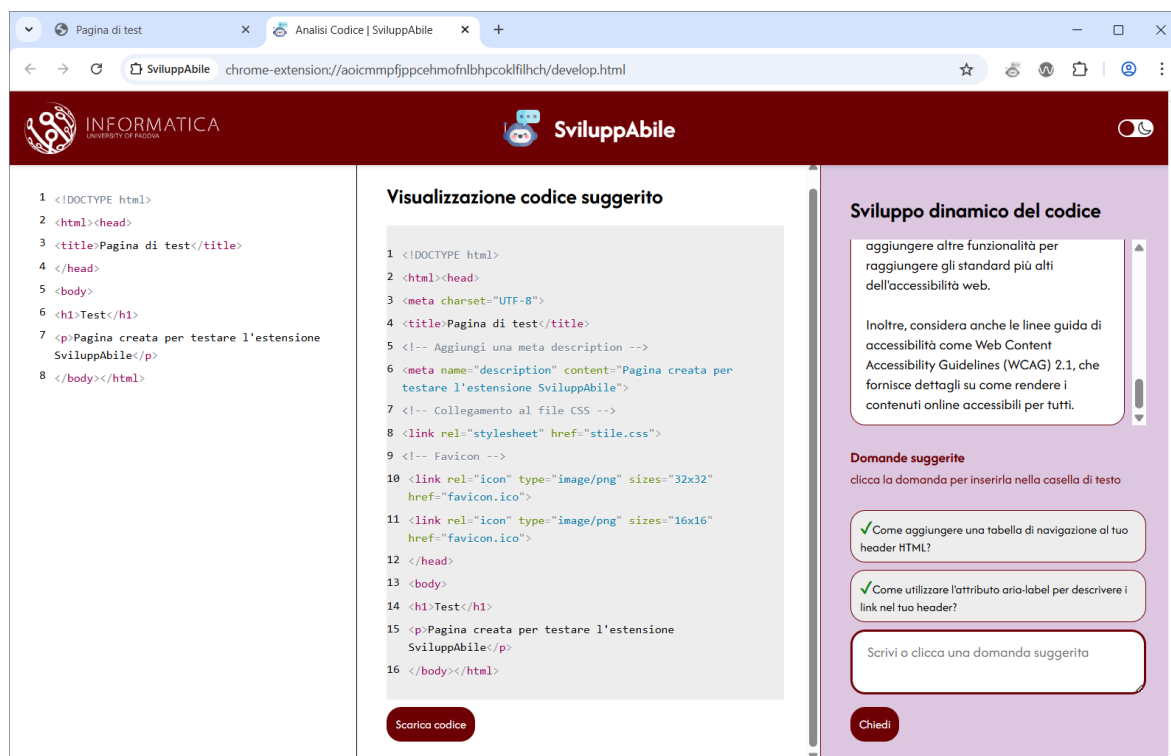


Figura 4.9: Modalità: sviluppo guidato, dopo alcune interazioni

4.3.2 Interazione con l'AI

L'estensione sviluppata prevede un'integrazione diretta con Ollama (come descritto in precedenza). Questa scelta garantisce all'utente il pieno controllo sui dati, riducendo i rischi legati alla trasmissione di informazioni sensibili verso servizi esterni e permettendo un utilizzo anche in assenza di connessione Internet stabile. L'interazione con l'*intelligenza artificiale* avviene attraverso chiamate HTTP ad un endpoint locale, al quale vengono inviati prompt costruiti dinamicamente in base alle esigenze del flusso operativo. Il codice 4.1 mostra tale interazione.

```

1  async function inviaPrompt(prompt) {
2    const res = await ↵
      ↵ fetch('http://localhost:11434/api/generate', {
3      method: 'POST',
4      headers: { 'Content-Type': 'application/json' },
5      body: JSON.stringify({

```

```
6     model: "llama3.1:8b",
7     prompt,
8     stream: false
9   })
10  });
11  return await res.json();
12 }
```

Listing 4.1: Funzione di interazione con Ollama

Sono stati definiti tre casi d'uso principali per la generazione dei prompt:

- l'invio congiunto del *DOM* della pagina e della domanda dell'utente, utile per ricevere spiegazioni e indicazioni sulle righe che verranno poi evidenziate in quanto utili alla comprensione della risposta. Tale prompt è visibile nel codice 4.2;

```
1  const promptPrincipale =
2  'Sei un assistente che analizza codice HTML. Rispondi ←
   ← alla domanda in modo chiaro ma conciso, usando ←
   ← al massimo 5-6 frasi. ' +
3  'Evita ripetizioni o spiegazioni troppo generiche. ' +
4  'Alla fine della risposta, su una nuova riga, scrivi ←
   ← le righe eventualmente utilizzate per la ←
   ← risposta nel seguente formato:\n\n' +
5  '##RIGHE##\n{"righe": [elenco_di_numeri_di_riga]}\n\n' +
6  'Codice HTML:\n${codice}\n\nDomanda: ${domanda}';
```

Listing 4.2: Prompt per la generazione di risposta e righe da evidenziare

- l'invio della sola domanda per generare ulteriori domande più approfondite da consigliare all'utente, come visibile nel codice 4.3;

```
1  const promptSuccessiva =
2  'Suggerisci 1 o 2 domande piu' specifiche ←
   ← sull'accessibilita' o sull'analisi del codice, ' +
3  'partendo dalla seguente domanda:\n\n' +
```

```
4      'Rispondi solo con il seguente formato JSON:\n\n' +  
5      '##DOMANDA##\n{ "domande": ["prima domanda", ↵  
        ↵ "seconda...", "terza..."] }\n\n' +  
6      'Domanda iniziale: ${domanda}';
```

Listing 4.3: Prompt per la generazione di domande successive

- l'invio del *DOM* insieme alla richiesta di modifica, scenario in cui l'*IA* produce sia una risposta argomentata sia un blocco di codice pronto per essere inserito nel pannello centrale della pagina nella modalità di sviluppo guidato (vedi codice 4.4).

```
1      const promptCodice =  
2      'Sei un assistente che aiuta a rendere accessibile ↵  
        ↵ il codice HTML. ' +  
3      'Rispondi in massimo 5-6 frasi chiare e tecniche. ' +  
4      'Se suggerisci del codice, racchiudilo tra i ↵  
        ↵ marcatori \'##CODICE##\' come mostrato di ↵  
        ↵ seguito:\n\n' +  
5      '##CODICE##\n<codice HTML da inserire o ↵  
        ↵ modificare>\n##FINECODICE##\n\n' +  
6      'Codice HTML:\n${codice}\n\nDomanda: ${domanda}';
```

Listing 4.4: Prompt per la generazione di risposta e codice *HTML* accessibile

Questa diversificazione consente di mantenere un approccio modulare e adattabile, rendendo l'assistente in grado di rispondere a necessità differenti con un unico modello sottostante.

4.3.3 Filtraggio risposta generata

Un aspetto fondamentale del funzionamento dell'estensione riguarda il filtraggio e la rielaborazione della risposta generata dall'intelligenza artificiale. Le risposte restituite da Ollama, infatti, non vengono mostrate direttamente all'utente, ma sono sottoposte

ad un processo di *parsing* e di pulizia.

In primo luogo, l'estensione distingue le diverse tipologie di output attese: la risposta vera e propria alla domanda inserita, le eventuali domande suggerite e gli eventuali blocchi di codice generati da visualizzare e/o scaricare. A tal fine vengono utilizzati marcatori testuali inseriti nel prompt (ad esempio `##DOMANDE##` o `##CODICE##`), che consentono di individuare con precisione le sezioni rilevanti all'interno della risposta.

Una volta ricevuto l'output, funzioni dedicate come `estraiRigheDaRisposta` ed `estraiDomandeSuggerite` applicano espressioni regolari per isolare le parti utili, scaricando elementi ridondanti o formattazioni non necessarie.

Il filtraggio consente anche di separare le informazioni in blocchi distinti, in modo che ciascun contenuto possa essere mostrato nella pagina web nel pannello appropriato (ad esempio, suggerimenti testuali nella chat e codice sorgente nel riquadro centrale). Questo approccio riduce il carico cognitivo per l'utente, che non si trova di fronte a una risposta grezza e complessa, ma ad un output strutturato e facilmente navigabile. Inoltre la possibilità di visualizzare alcune righe di codice evidenziate consente una comprensione più immediata del codice sorgente analizzato rendendo più intuitivo il processo di revisione del codice.

4.4 Problematiche riscontrate

Durante lo sviluppo del progetto sono emerse alcune criticità di natura tecnica. In primo luogo, la scelta dell'*IA* da integrare si è rivelata complessa: le *API* gratuite disponibili online presentavano forti limitazioni legate al numero di token, rendendole inadeguate per un utilizzo continuativo. Successivamente è stata valutata l'adozione di Ollama, la cui installazione, tuttavia, non è risultata possibile sul computer personale a causa delle risorse hardware insufficienti. L'ostacolo è stato superato grazie alla disponibilità di un computer fornito dall'università, che ha consentito l'utilizzo stabile della piattaforma.

Un'ulteriore problematica si è manifestata dopo l'integrazione dell'*IA* nell'estensione: l'invio delle richieste tramite la chat restituiva sistematicamente il messaggio **"Errore durante l'elaborazione: Risposta vuota dal server"**. Attraverso un'attenta fase di *debug*, supportata dall'inserimento di log per tracciare l'esecuzione, è stato possibile individuare la causa: non un malfunzionamento dell'*IA* o dell'estensione, bensì le restrizioni imposte da Chrome, che limita l'interazione delle estensioni con *API* esterne, bloccando di fatto la corretta trasmissione delle richieste.

Il problema è stato infine risolto avviando Chrome da terminale con parametri specifici che disabilitano i blocchi di sicurezza predefiniti, permettendo così all'estensione di comunicare correttamente con l'*IA*.

4.5 Problematiche riscontrate

Durante lo sviluppo del progetto sono emerse alcune criticità di natura tecnica. In primo luogo, la scelta dell'*IA* da integrare si è rivelata complessa: le *API* gratuite disponibili online presentavano forti limitazioni legate al numero di token, rendendole inadeguate per un utilizzo continuativo. Successivamente è stata valutata l'adozione di Ollama, la cui installazione, tuttavia, non è risultata possibile sul computer personale a causa delle risorse hardware insufficienti. L'ostacolo è stato superato grazie alla disponibilità di un computer fornito dall'università, che ha consentito l'utilizzo stabile della piattaforma.

Un'ulteriore problematica si è manifestata dopo l'integrazione dell'*IA* nell'estensione: l'invio delle richieste tramite la chat restituiva sistematicamente il messaggio **"Errore durante l'elaborazione: Risposta vuota dal server"**. Attraverso un'attenta fase di *debug*, supportata dall'inserimento di log per tracciare l'esecuzione, è stato possibile individuare la causa: non un malfunzionamento dell'*IA* o dell'estensione, bensì le restrizioni imposte da Chrome, che limita l'interazione delle estensioni con *API* ester-

ne, bloccando di fatto la corretta trasmissione delle richieste. Il problema è stato infine risolto avviando Chrome da terminale con parametri specifici che disabilitano i blocchi di sicurezza predefiniti, permettendo così all'estensione di comunicare correttamente con l'*IA*.

Infine, è emersa una limitazione legata alla gestione della memoria dell'*IA*: idealmente, il codice sorgente della pagina avrebbe potuto essere caricato una sola volta al momento della scelta della modalità, inviando in seguito soltanto le domande per migliorare tale codice.

Tuttavia, Ollama non dispone di una capacità di memoria sufficiente a mantenere lo stato della conversazione in assenza del ricaricamento completo del *DOM*. Ciò comporta la necessità di reinviare l'intero contenuto ad ogni interazione, con un impatto significativo sull'efficienza complessiva.

Capitolo 5

Test

5.1 Test “Analisi assistita”

Per effettuare i test dell'estensione sono stati utilizzati alcuni siti realizzati per l'edizione 2025 del concorso *Accattivante Accessibile*. I test hanno preso in considerazione gli errori individuati da TV insieme a quelli segnalati dalla docente referente e sono stati messi a confronto con gli errori rilevati da *SviluppAbile*. Infine, ho confrontato i risultati elaborando un resoconto tabellare per ciascun sito analizzato e, successivamente, ho applicato la metrica F_1 -score per ottenere un report oggettivo.

5.1.1 F1-score

La F_1 -score è una metrica utilizzata per valutare modelli di classificazione, sia binari che multi-classe. Essa combina in un unico valore la *precision* (quanto le predizioni positive sono corrette) e il *recall* (quanti casi positivi reali vengono identificati), calcolati come:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}$$

L' F_1 -score è definita come la media armonica di tali valori:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

Questo significa che nel calcolo verranno utilizzati i tre dati rilevati nel presente lavoro: "errori trovati" come TP (veri positivi), "errori non trovati" come FN (falsi negativi), e FP (falsi positivi). In questo modo, l' F_1 -score riflette in modo equilibrato sia la

capacità di individuare correttamente errori reali, sia il controllo sui falsi allarmi.

L' F_1 -score assume valori compresi tra 0 e 1. Il valore minimo, 0, indica prestazioni pessime (almeno una delle due metriche è nulla), mentre il valore massimo, 1, corrisponde a prestazioni perfette (precisione e recall entrambi pari a 1).

Valori prossimi a 1 indicano quindi un buon funzionamento del sistema, mentre valori vicini a 0 segnalano performance insoddisfacenti.

5.1.2 Sito web: SudokuWorld

Di seguito vengono riportati alcuni dei test effettuati sul sito [SudokuWorld](https://caa.studenti.math.unipd.it/amonaco/Sudokuworld/pages/home.php).

5.1.2.1 Total Validator

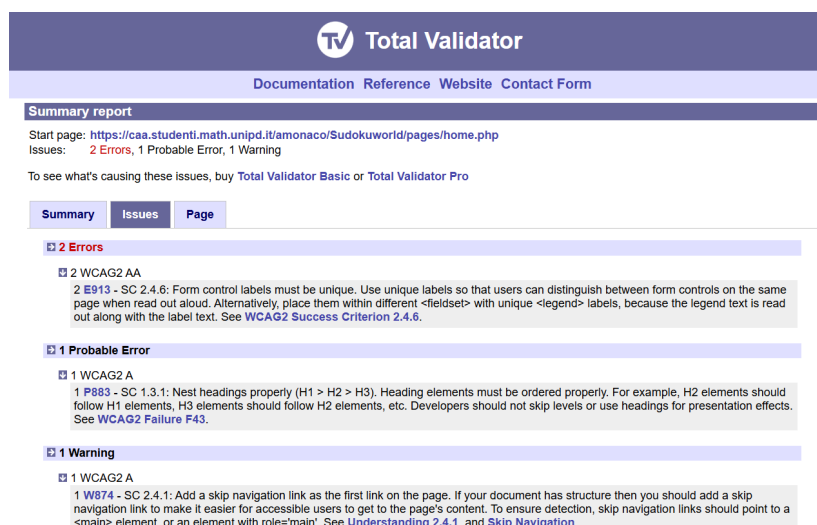


Figura 5.1: Analisi di Total Validator sul sito web *SudokuWorld*

L'analisi (vedi figura 5.1) ha evidenziato due errori principali: etichette dei controlli dei form non univoche e intestazioni nidificate in modo non corretto. È stato inoltre segnalato un warning relativo all'assenza di un link di navigazione rapida.

5.1.2.2 Lighthouse

Il report generato dallo strumento Lighthouse ha restituito un punteggio di 100/100 nella sezione Accessibility, indicando che, secondo le metriche automatiche adottate, non sono stati rilevati errori o problematiche di conformità (vedi figura 5.2).

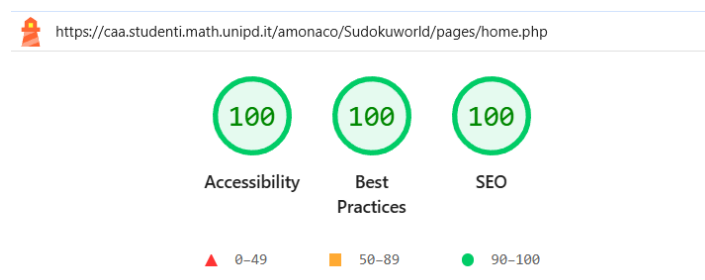


Figura 5.2: Analisi di Lighthouse sul sito web *SudokuWorld*

5.1.2.3 SviluppAbile

Di seguito vengono riportati alcuni esempi di domande poste all'estensione *SviluppAbile* e delle relative risposte ricevute.

(pagina `home.php`)

Il mio codice HTML rispetta le linee guida WCAG?

Il tuo codice HTML sembra essere in parte conforme alle linee guida WCAG (Web Content Accessibility Guidelines), ma potrebbe migliorare ulteriormente per garantire una maggiore accessibilità. Alcune aree che richiedono ulteriore attenzione includono:

- L'attributo 'aria-label' non è utilizzato in modo coerente nei controlli di navigazione.
- Le etichette dei pulsanti "Cerca" e "Profilo" potrebbero essere più descrittive.

Tuttavia, il tuo codice include molti elementi positivi come l'utilizzo di 'alt' per immagini, la presenza di un titolo ('<title>') e di metatag di descrizione ('<meta name="description">'). Questi sono passaggi fondamentali per migliorare la comprensione del contenuto da parte dei motori di ricerca e degli assistenti.

Per garantire una maggiore accessibilità, considera di applicare ulteriormente le linee guida WCAG 2.1, che offrono standard e raccomandazioni per creare esperienze web più inclusive.

La pagina non rispetta la norma W3C WCAG 2.1 che richiede di nidificare correttamente le intestazioni. Non sono presenti elementi HTML con attributo "h1", mentre ci sono intestazioni con grado inferiore.

La situazione descritta viola la regola WCAG 1.3.1, che richiede la corretta gerarchia di intestazioni ($H1 > H2 > H3$). In questo caso, l'h2 non è diretto al figlio dell'H1.

Vi è inoltre la presenza di due etichette (label) con il valore "Cerca" all'interno della stessa pagina HTML, ma con attributi id diversi ("searched-product" e "for"). Secondo la specifica SC 2.4.6, le etichette dei form control devono essere uniche.

Vi sono errori riguardanti i link?

Il codice HTML contiene due carousel circolari:

1. Il primo è definito come "specialProductsCarousel" e si trova nella sezione '<main>'.
2. Il secondo è chiamato "bestSellersCarousel" ed è anch'esso presente all'interno della stessa sezione.

Quanto ai controlli sui link visitati, non sono implementati controlli specifici per la traccia dei link visitati sul sito web.

(pagina sudoku.php)

Il mio codice HTML rispetta le linee guida WCAG?

Il tuo codice HTML non rispetta pienamente le linee guida WCAG (Web Content Accessibility Guidelines) per diverse ragioni:

1. ****La navigazione è troppo complessa****: la barra di navigazione contiene molti elementi e link, potrebbe essere difficile per gli utenti disabili utilizzare l'interfaccia.
2. ****L'accessibilità dei form è insufficiente****: non sono presenti attributi accessibili per i campi di input e submit, rendendo la navigazione dei form difficoltosa per gli utenti che usano dispositivi di assistenza.
3. ****La lettura dei testi in basso risulta difficile****: alcuni elementi come il timer sono

troppo piccoli per essere letti facilmente dagli schermi con alto contrasto o da dispositivi di assistenza.

4. ****Non ci sono alternative al linguaggio naturale****: le immagini contengono solo icone e simboli, che potrebbero non essere compresi da utenti analfabetici.

Aggiungere attributi accessibili ai campi input e submit:

* Modifica il codice HTML come segue:

```
<form class="d-flex align-items-center my-2 my-lg-0 ms-auto"
action="/amonaco/Sudokuworld/pages/search.php" method="GET">
    <label for="searched-product" class="visually-hidden">Cerca</label>

    <input class="form-control me-2" type="search" name="searched-product"
id="searched-product" placeholder="Cerca" value="" aria-label="Search"
/>

    <button class="btn btn-outline-success" type="submit" aria-label="Submit
the form">Cerca</button>

</form>
```

Aggiungere tag alt alle immagini: assicurati di aggiungere il contenuto alternativo per ogni immagine.

* Modifica il codice HTML come segue:

```
<a class="nav-link" href="/amonaco/Sudokuworld/pages/sudoku.php">

    <i class="bi bi-grid-3x3-gap-fill"></i>

</a>
```

Aggiungere tag aria: per fornire informazioni aggiuntive alle pagine web, puoi utilizzare gli attributi aria.

* Modifica il codice HTML come segue:

```
<h1 class="container border border-black p-2 mb-4 rounded text-center" id="sudokuTitle"
role="heading" aria-level="1">!! Sudoku !!</h1>
```

Controlla l'accessibilità con strumenti di testing: prima della pubblicazione, assicura-

ti di utilizzare strumenti come Lighthouse o WAVE per valutare la tua pagina web e risolvere eventuali problemi.

Ricorda che un sito web accessibile è più facile da utilizzare per tutti gli utenti, compresi quelli con disabilità.

DOMANDE SUGGERITE:

"Quale è il criterio WCAG 2.1 che riguarda la navigazione con tastiera?"

"Come posso utilizzare gli strumenti di analisi del codice per identificare le aree dell'accessibilità da migliorare nel mio sito web HTML?"

"Come posso utilizzare gli strumenti di analisi del codice per identificare le aree dell'accessibilità da migliorare nel mio sito web HTML?"

Per identificare le aree dell'accessibilità da migliorare nel tuo sito web HTML, puoi utilizzare gli strumenti di analisi come W3C Validator e Lighthouse. Il W3C Validator verifica la correttezza del codice HTML e fornisce suggerimenti per migliorare l'accessibilità. Lighthouse è uno strumento open-source che valuta l'accessibilità, le prestazioni e la sicurezza del tuo sito web.

Resoconto finale dei test

Di seguito viene illustrato il processo di test effettuato da *SviluppAbile* confrontando i risultati ottenuti dagli altri strumenti di validazione e integrando anche un controllo manuale basato sulle Web Content Accessibility Guidelines (WCAG 2.1). Lo studio è stato articolato sulle principali pagine del sito, inclusa l'area riservata.

Pagina home.php

L'estensione *SviluppAbile* ha rilevato la mancanza di una gerarchia corretta delle intestazioni: non è presente un elemento `<h1>`, mentre compaiono direttamente intestazioni di livello inferiore. Questo comportamento viola il criterio *1.3.1 - Info and Relationships*, che prescrive una corretta nidificazione semantica. Inoltre, sono state

individuare etichette duplicate per i campi di ricerca, contrarie al criterio [2.4.6 - *Headings and Labels*](#), ed un utilizzo incoerente di aria-label nei controlli di navigazione, che può generare confusione negli screenreader. L'analisi manuale ha permesso di osservare ulteriori criticità, tra cui la mancanza di un meccanismo di bypass (skip link) in violazione del criterio [2.4.1 - *Bypass Blocks*](#).

Pagina sudoku.php

In questa pagina è stato rilevato che i form risultano privi di etichette accessibili e che i pulsanti non sono sufficientemente descrittivi, in contrasto con i criteri [3.3.2 - *Labels or Instructions*](#) e [4.1.2 - *Name, Role, Value*](#). È stata inoltre segnalata l'assenza di testi alternativi per le icone, che viola il criterio [1.1.1 - *Non-text Content*](#). Alcuni elementi testuali, come il timer, presentano dimensioni ridotte e difficilmente leggibili, con possibili ricadute sul criterio [1.4.4 - *Resize Text*](#). Un'osservazione manuale ha evidenziato che i contrasti cromatici non sempre soddisfano il rapporto minimo richiesto dal criterio [1.4.3 - *Contrast \(Minimum\)*](#) e che non è garantita la piena navigabilità da tastiera ([2.1.1](#)).

Pagina search.php

La pagina di ricerca consente agli utenti di filtrare e visualizzare i prodotti disponibili. L'estensione *SviluppAbile* ha rilevato alcune criticità principali: assenza di intestazione <h1>, che viola il criterio [1.3.1](#), etichette duplicate nei campi di ricerca ([2.4.6](#)) e uso incoerente di aria-label nei controlli di navigazione.

L'analisi tramite strumenti di accessibilità ha evidenziato ulteriori problematiche: mancata navigabilità completa da tastiera ([2.1.1](#)), testi alternativi mancanti per le icone o immagini illustrative ([1.1.1](#)) e che il contrasto di alcuni elementi testuali non soddisfa il rapporto minimo previsto ([1.4.3](#)).

Pagine product.php?id=...

Le schede prodotto (analizzando quelle relative alla tazza "I love sudoku" e alla maglia

“commit sudoku”) hanno mostrato ulteriori problematiche. *SviluppAbile* ha rilevato la mancanza di testi alternativi per le immagini (1.1.1), l’assenza di titoli significativi nei tag <title> (2.4.2 - *Page Titled*) e l’utilizzo non semantico del markup per i prezzi scontati (1.3.1). Sono stati inoltre segnalati link privi di indicazioni chiare di stato focus o visited (2.4.7 - *Focus Visible*) e la mancanza di attributi ARIA utili a descrivere pulsanti generici come “Aggiungi al carrello” (4.1.2). Quest’ultimo errore rappresenta un falso negativo, in quanto il pulsante è definito correttamente. L’analisi tramite strumenti di accessibilità ha messo in luce ulteriori difetti: contrasti di colore insufficienti (1.4.3), assenza di skip link (2.4.1) e mancanza di landmark semantici (1.3.1).

Pagina carrello.php

Sebbene l’estensione non abbia segnalato particolari criticità nella struttura della pagina, un’analisi manuale rivela che i prodotti nel carrello non sono accompagnati da testi alternativi adeguati per le immagini (1.1.1), né da descrizioni sufficientemente univoche per i pulsanti di azione, come “Rimuovi” o “Procedi al pagamento”. La mancanza di etichette descrittive viola i criteri 2.4.6 e 3.3.2.

Pagina login.php

L’accesso alla pagina di login con credenziali di test ha permesso all’estensione di segnalare alcune criticità: i campi di input risultano privi di etichette esplicite e i pulsanti sono generici, compromettendo la fruibilità per chi utilizza tecnologie assistive (3.3.2 e 4.1.2). Inoltre viene segnalata l’assenza di attributi `aria-label` sui controlli di autenticazione. Quest’ultimo errore in realtà è un falso negativo, derivante dalla capacità ridotta di analisi dell’IA alla base dell’estensione.

L’analisi manuale ha messo in luce ulteriori problematiche non rilevate dallo strumento: la mancanza di messaggi di errore in caso di inserimento di credenziali errate (3.3.1 - *Error Identification*), link come “Registrati” privi di testo descrittivo contestuale (2.4.4 - *Link Purpose (In Context)*), e la mancanza di informazioni semantiche aggiuntive che

faciliterebbero la navigazione da tastiera ([2.1.1](#)).

Conclusioni

Il confronto tra i risultati di *SviluppAbile* e l'analisi tramite strumenti di convalida (integrata con un'analisi manuale) basati sulle WCAG mostra come l'estensione sia efficace nell'individuare errori comuni e strutturali – quali la mancanza di testi alternativi, etichette duplicate, heading disordinati e pulsanti privi di descrizione – ma non riesca a rilevare altri aspetti fondamentali per l'accessibilità, come i contrasti cromatici e la presenza di skip link. Inoltre l'estensione suggerisce modifiche non conformi, come ad esempio attributi alt non idonei.

Questo dimostra che uno strumento automatico, pur costituendo un valido supporto, deve al giorno d'oggi ancora essere integrato con una verifica manuale alla luce delle linee guida ufficiali del W3C.

5.1.2.3.1 F1-score

Nella tabella [5.1](#) vengono riassunti i risultati ottenuti.

Tabella 5.1: Tabella riassuntiva analisi *SudokuWorld* tramite *SviluppAbile*

Errori trovati	Errori non trovati	Falsi positivi
<ul style="list-style-type: none">• $h1 > h2 > h3$• Etichette dei form non univoche• Errori con gli alt• Errore button "registra-ti"• Link circolari• Manca controllo link già visitati	<ul style="list-style-type: none">• Contrasti dei colori• Skip navigation link	<ul style="list-style-type: none">• Tag ARIA (suggerimenti errati)• Suggerimenti alt errati

– continua nella pagina successiva

Tabella 5.1 – continua dalla pagina precedente

Errori trovati	Errori non trovati	Falsi positivi
6	2	2

L' F_1 -score calcolato è $F_1 = 0.75$, un valore piuttosto buono che indica una capacità complessiva soddisfacente di individuare gli errori reali, pur con la presenza di alcuni mancati rilevamenti e di falsi positivi. Ciò suggerisce che l'estensione riesce a fornire risultati utili e affidabili, anche se rimane margine per ulteriori miglioramenti nella precisione e nella completezza dell'analisi.

5.1.3 Sito web: Dolce Risveglio

Di seguito vengono riportati alcuni dei test effettuati sul sito web [Dolce Risveglio](#).

5.1.3.1 Total Validator

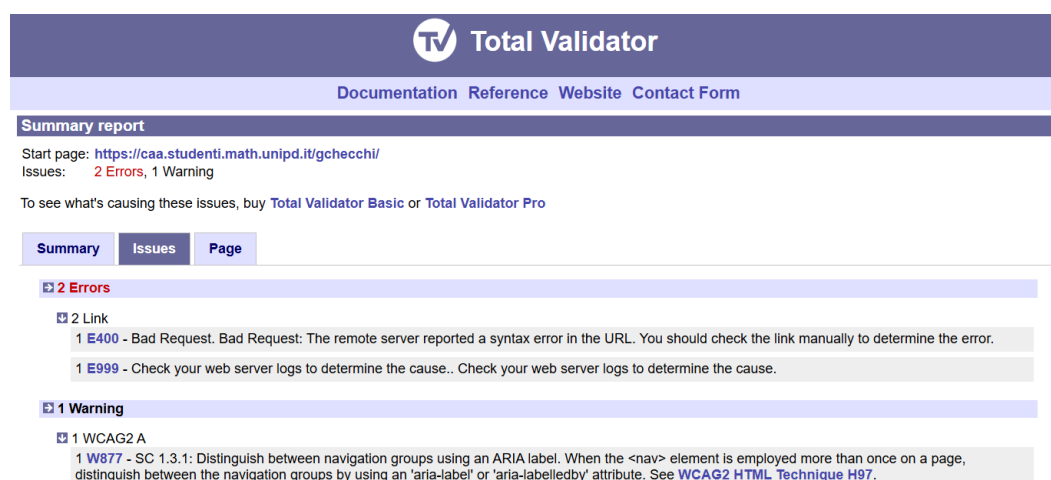


Figura 5.3: Analisi di Total Validator sul sito web *Dolce Risveglio*

L'analisi (vedi figura 5.3) ha rilevato 2 errori relativi ai link e 1 warning. Gli errori comprendono un URL con richiesta non valida e un problema lato server da verificare nei log. Il warning riguarda la necessità di distinguere tra più aree di navigazione tramite un'etichetta ARIA.

5.1.3.2 Lighthouse

Il report ha restituito un punteggio di 100/100 nella sezione *Accessibility*, indicando che, secondo le metriche automatiche adottate, non sono stati rilevati errori o problematiche di conformità (vedi figura 5.4).

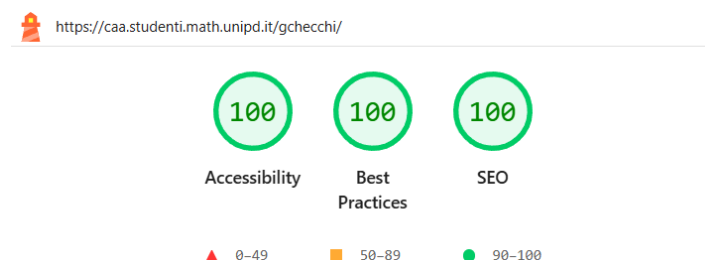


Figura 5.4: Analisi di Lighthouse sul sito web *Dolce Risveglio*

5.1.3.3 Sviluppabile

Di seguito viene illustrato il processo di test effettuato da *SviluppAbile* confrontando i risultati ottenuti dagli altri strumenti di validazione e integrando anche un controllo manuale basato sulle Web Content Accessibility Guidelines (WCAG 2.1). Lo studio è stato articolato sulle principali pagine del sito.

Pagina home.php

La pagina presenta una panoramica della caffetteria, con sezioni dedicate alla storia del locale, ai prodotti offerti e link al menù. L'estensione ha rilevato alcune criticità principali: immagini prive di testi alternativi (*1.1.1 - Non-text Content*), uso di interazioni senza gerarchia coerente (*1.3.1 - Info and Relationships*) e suggerimenti per aggiungere label ed id negli elementi di input (*2.4.6 - Headings and Labels*).

L'analisi tramite strumenti automatici ha evidenziato ulteriori problemi: contrasto insufficiente in alcune aree di testo o link (*1.4.3*) e l'assenza di alcuni alt (*1.1.1*).

Pagina menu.php

La pagina presenta il menù della caffetteria, con sezioni dedicate a bevande, pietanze

e dolci, strutturate in liste e tabelle descrittive (`<d1>`). L'estensione *SviluppAbile* ha rilevato alcune criticità principali: immagini dei piatti prive di testi alternativi significativi (1.1.1), bottoni per la selezione dei menu senza chiara indicazione dello stato attivo tramite attributi ARIA (4.1.2 - *Name, Role, Value*) e suggerimenti per aggiungere label agli elementi interattivi (2.4.6). L'estensione suggerisce inoltre una modifica delle tabelle descrittive; tale modifica è purtroppo errata in quanto l'IA attualmente non è in grado di creare correttamente tabelle accessibili.

Ulteriori controlli con strumenti automatici hanno evidenziato problemi come contrasti insufficienti tra testo e sfondo in alcune aree del menù (1.4.3) e assenza di skip link funzionanti per navigare rapidamente tra le sezioni del menù (2.4.1).

Pagina contatti.php

La pagina consente agli utenti di inviare richieste o informazioni tramite un modulo di contatto, presentando campi per nome, email, telefono e messaggio. L'estensione ha rilevato alcune criticità principali: immagini con alt poco chiari (1.1.1), campi del modulo senza indicazioni ARIA aggiuntive per facilitare la navigazione e la comprensione (4.1.2) e suggerimenti per aggiungere label più descrittive agli elementi interattivi presenti nel footer (2.4.6).

L'analisi tramite strumenti automatici ha evidenziato ulteriori problemi: contrasto insufficiente tra testo e sfondo in alcune aree (1.4.3), assenza di landmark semantici principali (`<main>` e `<nav>`) (1.3.1) e mancanza di skip link alternativi per navigare rapidamente al contenuto principale (2.4.1).

Pagina prenotazioni.php

La pagina `prenotazioni.php` permette agli utenti di prenotare un tavolo, specificando nome, email, telefono, giorno, orario e tavolo desiderato tramite un modulo strutturato. L'estensione ha rilevato alcune criticità principali: immagini prive di testi alternativi (1.1.1), campi del modulo senza indicazioni ARIA aggiuntive per facilitare la comprensione e l'interazione (4.1.2) e un errore riguardante l'uso improprio degli attributi

required e aria-required nello stesso tag, che può generare errori.

L'analisi tramite strumenti automatici non ha evidenziato ulteriori problemi.

Conclusioni

Il confronto tra i risultati di *SviluppAbile* e le verifiche effettuate con strumenti automatici mostra che l'estensione è efficace nell'individuare errori di struttura e contenuto, come testi alternativi mancanti, etichette non descrittive e incongruenze tra attributi `required` e `aria-required`. Tuttavia, l'estensione non rileva completamente problematiche relative al contrasto cromatico, landmark semantici, skip link e piena navigabilità da tastiera.

Per ottenere una valutazione completa dell'accessibilità del sito, è necessario integrare l'uso di *SviluppAbile* con controlli aggiuntivi basati sulle WCAG, assicurando che tutti i form abbiano label univoche, id coerenti e indicazioni ARIA appropriate.

5.1.3.3.1 F1-score

Nella tabella [5.2](#) vengono riportati i risultati ottenuti utilizzando l'estensione *SviluppAbile*.

Tabella 5.2: Tabella riassuntiva analisi *Dolce Risveglio* tramite *SviluppAbile*

Errori trovati	Errori non trovati	Falsi positivi
<ul style="list-style-type: none">• Alt non corretti delle img• Etichette dei form non univoche• Suggerisce l'aggiunta di label e id per gli elementi input• Errore required e aria-required utilizzati insieme	<ul style="list-style-type: none">• Alcuni alt mancanti	<ul style="list-style-type: none">• Suggerisce un eccessivo uso di ARIA
4	1	1

L' F_1 -score calcolato è $F_1 = 0.8$, un risultato positivo che evidenzia una buona capacità di individuazione degli errori effettivi, con un numero limitato di falsi positivi e pochi elementi mancanti.

5.1.4 Sito web: E-lixirium

Di seguito vengono riportati alcuni dei test effettuati sul sito web [E-lixirium](#).

5.1.4.1 Total Validator

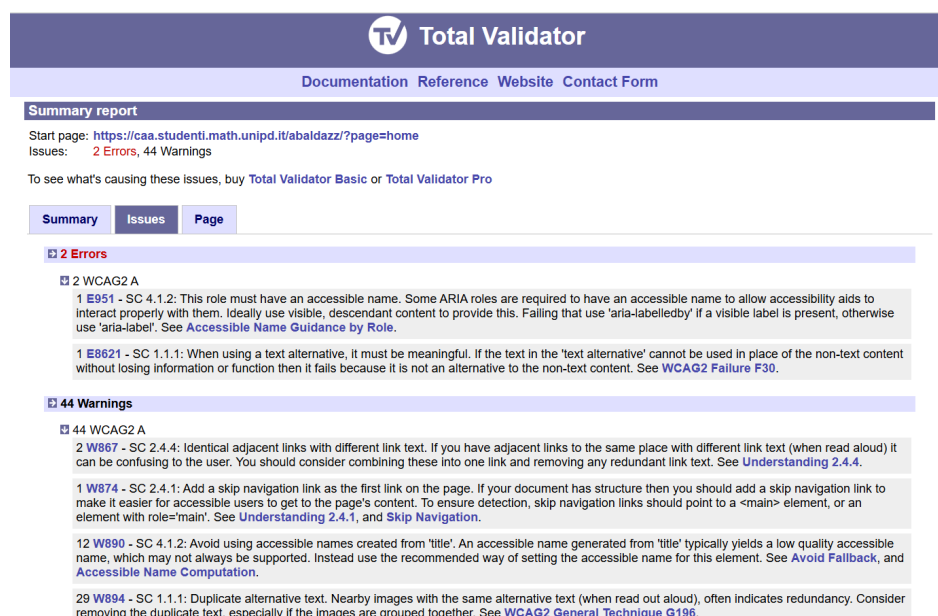


Figura 5.5: Analisi di Total Validator sul sito web *E-lixirium*

L'analisi (vedi figura 5.5) ha rilevato 2 errori principali e 44 warning. Gli errori riguardano principalmente elementi con ruoli ARIA a cui non è stato associato un nome accessibile e testi alternativi non significativi. Tra i warning, i più frequenti sono duplicazioni di testo alternativo per immagini, nomi accessibili generati da attributi `title`, link adiacenti con testi diversi e l'assenza di link di navigazione rapida.

5.1.4.2 Lighthouse

Il report ha restituito un punteggio di 96/100 nella sezione *Accessibility* (vedi figura 5.6). Il problema segnalato riguarda il contrasto insufficiente tra colori di sfondo e primo piano, che può ridurre la leggibilità per alcuni utenti.

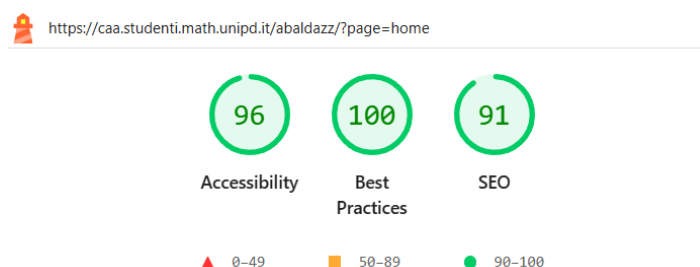


Figura 5.6: Analisi di Lighthouse sul sito web *E-lixirium*

5.1.4.3 SviluppAbile

Di seguito viene illustrato il processo di test effettuato da *SviluppAbile* confrontando i risultati ottenuti dagli altri strumenti di validazione e integrando anche un controllo manuale basato sulle Web Content Accessibility Guidelines (WCAG 2.1). Lo studio è stato articolato sulle principali pagine del sito.

Pagina home

La pagina contiene sezioni dedicate al logo, alla navigazione e alla presentazione di categorie e prodotti. L'estensione ha rilevato alcune criticità principali: immagini dei prodotti prive di testi alternativi significativi (*1.1.1 - Non-text Content*), link adiacenti con testi diversi ma comportamento simile (*2.4.4*) ed elementi della navbar nascosti tramite `display:none` invece di `visibility:hidden` (*2.1.1 - Keyboard*).

L'analisi tramite strumenti automatici ha evidenziato ulteriori problemi: assenza di skip link per saltare la navigazione (*2.4.1 - Bypass Blocks*) e link duplicati o circolari nella barra di navigazione (*2.4.4*). Un ulteriore problema non rilevato da *SviluppAbile* riguarda il logo con link circolare verso la homepage anche quando l'utente è già nella homepage stessa (*2.4.4 - Link Purpose (In Context)*).

Pagina products

La pagina mostra l'elenco completo dei prodotti disponibili nel negozio *E-lixirium*. Come per la homepage, si riscontrano criticità di accessibilità significative: le immagini dei prodotti utilizzano come testo alternativo il nome del file (*1.1.1*), le stelle di va-

lutazione dei prodotti hanno alt duplicati o poco descrittivi ([1.1.1](#)), e non è presente uno skip link per saltare la navigazione ([2.4.1](#)).

Gli ulteriori problemi rilevati dagli strumenti automatici sono gli stessi rilevati nella homepage.

Pagina about

La pagina **about** presenta informazioni sull'azienda e la sua missione, con testo descrittivo e un'immagine illustrativa. Dal punto di vista dell'accessibilità, si riscontrano le stesse criticità delle pagine precedenti: l'immagine ha un testo alternativo generico (**About E-lixirium**) che non descrive il contenuto visivo in dettaglio ([1.1.1](#)).

La struttura della pagina non prevede skip link per saltare la navigazione principale ([2.4.1](#)) e non tutti i link sono chiaramente descrittivi nel contesto ([2.4.4](#)). Inoltre, l'estensione avvisa che le informazioni contenute nella pagina sono incomplete rispetto a ciò che ci si aspetta da tale pagina (mancanza di informazioni di contatto per esempio).

Conclusioni

L'analisi complessivamente evidenzia una serie di criticità ricorrenti, in particolare relative ai testi alternativi delle immagini, alla gestione della navigazione e al contrasto visivo degli elementi interattivi. Le problematiche riscontrate potrebbero ostacolare l'accesso al sito da parte di utenti con disabilità visive o che utilizzano ausili tecnologici. Vengono anche consigliati miglioramenti su alcune pagine del sito web, come l'aggiunta di informazioni per gli utenti e di testi alternativi più descrittivi.

5.1.4.3.1 F1-score

Nella tabella [5.3](#) vengono riportati i risultati ottenuti utilizzando l'estensione *Svilup-pAbile*.

Tabella 5.3: Tabella riassuntiva analisi *E-lixirium* tramite *SviluppAbile*

Errori trovati	Errori non trovati	Falsi positivi
<ul style="list-style-type: none">• Errori di gestione della navigazione• "display: none" da sostituire con "visibility: hidden"• Link adiacenti identici con testi diversi• Testo alternativo duplicato	<ul style="list-style-type: none">• Skip navigation link• Link circolari nella navbar• Logo con reindirizzamento nella home anche in home stesso	
4	3	0

L' F_1 -score calcolato è $F_1 = 0.73$, un risultato positivo che evidenzia come l'estensione abbia individuato correttamente la maggior parte degli errori presenti, pur lasciando tre problemi non rilevati. Non sono stati generati falsi positivi, il che indica un buon equilibrio tra precisione e completezza dell'analisi.

5.1.5 Sito web: Corsa Ideale

Di seguito vengono riportati alcuni dei test effettuati sul sito web [Corsa Ideale](#).

5.1.5.1 Total Validator

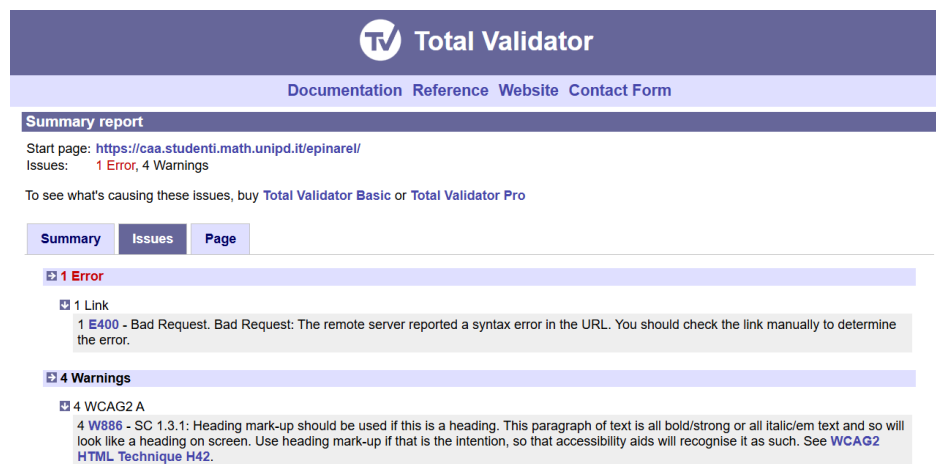


Figura 5.7: Analisi di Total Validator sul sito web *Corsa Ideale*

L'analisi ha evidenziato un solo errore riguardante i link, dovuto a una richiesta non valida (*Bad Request*), e quattro avvisi relativi all'uso improprio della marcatura tipografica (testi interamente in grassetto o corsivo che dovrebbero essere definiti come intersezioni). Questi warning non compromettono direttamente l'accessibilità, ma indicano buone pratiche di struttura da rispettare per garantire una corretta interpretazione da parte degli strumenti assistivi.

5.1.5.2 Lighthouse

Il report ha restituito un punteggio di 100/100 nella sezione *Accessibility*, indicando che, secondo le metriche automatiche adottate, non sono stati rilevati errori o problematiche di accessibilità (vedi figura 5.8). Il problema segnalato riguarda il contrasto insufficiente tra colori di sfondo e primo piano, che può ridurre la leggibilità per alcuni utenti.

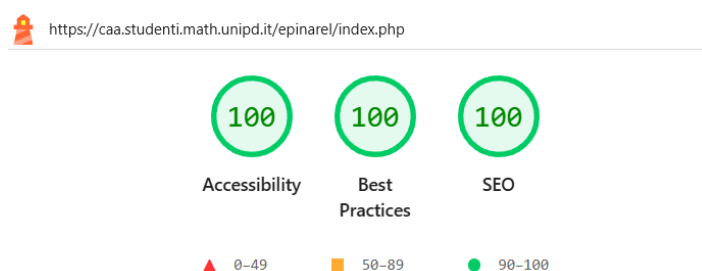


Figura 5.8: Analisi di Lighthouse sul sito web *Corsa Ideale*

5.1.5.3 SviluppAbile

Di seguito viene illustrato il processo di test effettuato da *SviluppAbile* confrontando i risultati ottenuti dagli altri strumenti di validazione e integrando anche un controllo manuale basato sulle Web Content Accessibility Guidelines (WCAG 2.1). Lo studio è stato articolato sulle principali pagine del sito.

Pagina home

La pagina presenta il sito **CorsaIdeale**, con sezioni dedicate alla navigazione, alla presentazione delle ultime uscite, ai servizi offerti e ai ruoli dei corridori. Dal punto di vista dell'accessibilità, si riscontrano le seguenti criticità principali: immagini dei prodotti e del logo prive di testi alternativi significativi ([1.1.1 - Non-text Content](#)), attributi `id` senza valore descrittivo o duplicati ([2.4.4 - Link Purpose \(In Context\)](#)) ed etichette `label` non univoche ([3.3.2 - Labels or Instructions](#)), .

La struttura della pagina prevede uno skip link per saltare la navigazione ([2.4.1](#)), ma non tutti i link della navbar sono chiaramente descrittivi nel contesto ([2.4.4](#)). Inoltre, non vengono utilizzati attributi ARIA aggiuntivi per migliorare la fruibilità tramite lettori di schermo, e alcune immagini decorative o informative non hanno alt appropriato ([1.1.1](#)). Questi due ultimi errori segnalati rappresentano in realtà falsi positivi; come visto anche nei test precedenti *SviluppAbile* tende a correggere erroneamente `alt` e ARIA.

Non sono stati riscontrati problemi relativi al contrasto dei colori ([1.4.3 - Contrast](#)

(*Minimum*)), rilevabili invece dagli altri strumenti di controllo dell'accessibilità.

Pagina lista

La pagina mostra l'elenco delle scarpe con funzionalità di ricerca, filtro e ordinamento. L'estensione *SviluppAbile* ha rilevato alcune criticità principali: assenza di testo alternativo significativo per le immagini delle scarpe e per il logo della navbar (*1.1.1*), utilizzo di `onclick` su interi blocchi cliccabili senza alternative da tastiera (*2.1.1 - Keyboard*). L'estensione rileva inoltre che i nomi degli attributi sono scritti in maiuscolo anziché in minuscolo (*4.1.1 - Parsing*).

L'analisi tramite strumenti di accessibilità ha evidenziato ulteriori problematiche: mancanza di landmark semantici coerenti come `<main>` e `<nav>` per ciascun blocco funzionale (*1.3.1*) ed etichette ridondanti o generiche nei pulsanti di filtro (*2.4.6 - Headings and Labels*).

Pagina chi-siamo

La pagina presenta informazioni sull'azienda, i valori e il team di CorsaIdeale. L'estensione ha rilevato alcune criticità principali: immagini senza testo alternativo significativo, compreso il logo e le foto dei membri del team (*1.1.1*), uso di `onclick` sui link del menu mobile senza alternative da tastiera (*2.1.1*).

L'analisi tramite strumenti automatici ha evidenziato ulteriori problematiche: mancanza di landmark semantici coerenti e completi per ciascun blocco di contenuto (*1.3.1*), etichette generiche o mancanti per alcuni link e pulsanti (*2.4.6*) e testi alternativi vuoti o poco descrittivi per le icone e immagini illustrative (*1.1.1*).

Pagina registrati

La pagina consente agli utenti di registrarsi ed accedere alle funzionalità del sito *CorsaIdeale*. L'estensione *SviluppAbile* ha rilevato alcune criticità principali: immagini del logo e del pulsante "torna su" prive di testo alternativo significativo (*1.1.1*) e campi del form con placeholder utilizzati come unica indicazione (*2.4.6*). Viene inoltre segnalato

l'errore di chiusura di un tag `p` con un tag `h3` ([1.3.1 - Info and Relationships](#)).

Conclusioni

L'analisi complessiva delle principali pagine di *CorsaIdeale* evidenzia criticità ricorrenti, in particolare relative ai testi alternativi delle immagini, alla gestione della navigazione e all'accessibilità dei form e dei link. Purtroppo l'estensione *SviluppAbile* non è in grado di controllare i contrasti dei colori in maniera esaustiva e suggerisce spesso errori non presenti di heading scorretti.

5.1.5.3.1 F1-score

Nella tabella [5.4](#) vengono riportati i risultati ottenuti utilizzando l'estensione *SviluppAbile*.

Tabella 5.4: Tabella riassuntiva analisi *Corsa Ideale* tramite *SviluppAbile*

Errori trovati	Errori non trovati	Falsi positivi
<ul style="list-style-type: none">• Alt immagini vuoti• Attributi id senza valore valido• Chiusura tag <code>p</code> erroneamente con tag <code>h3</code>• Label non univoca• Nomi attributi in maiuscolo anziché in minuscolo	<ul style="list-style-type: none">• Contrasto colori	<ul style="list-style-type: none">• Tag di heading usati in maniera scorretta
5	1	1

L' F_1 -score calcolato è $F_1 = 0.83$, un risultato positivo che evidenzia come l'estensione abbia rilevato correttamente la maggior parte degli errori presenti, lasciando un solo problema non individuato e generando un solo falso positivo.

5.1.6 Sito web: BookOverflow

Di seguito vengono riportati alcuni dei test effettuati sul sito [BookOverflow](#), sito web vincitore del concorso.

5.1.6.1 Total Validator



Figura 5.9: Analisi di Total Validator sul sito web *BookOverflow*

Come visibile in figura 5.9, non vi sono errori di accessibilità.

5.1.6.2 Lighthouse

Il report ha restituito un punteggio di 100/100 nella sezione *Accessibility*, indicando che, secondo le metriche automatiche adottate, non sono stati rilevati errori o problematiche di accessibilità (vedi figura 5.10).

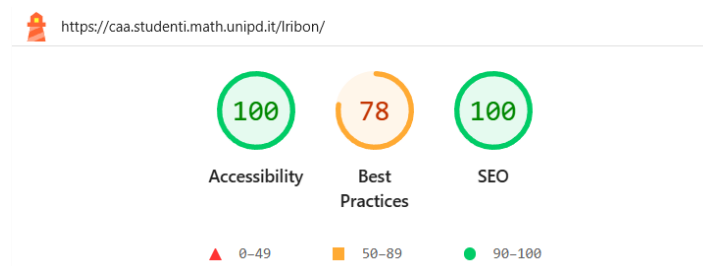


Figura 5.10: Analisi di Lighthouse sul sito web *BookOverflow*

5.1.6.3 SviluppAbile

Di seguito viene illustrato il processo di test effettuato da *SviluppAbile* confrontando i risultati ottenuti dagli altri strumenti di validazione e integrando anche un controllo

manuale basato sulle Web Content Accessibility Guidelines (WCAG 2.1). Lo studio è stato articolato sulle principali pagine del sito.

Pagina home

La pagina introduce la piattaforma e mostra i libri più scambiati. L'estensione *SviluppAbile* segnala un uso inappropriato del tabindex sugli elementi interattivi (2.1.1).

L'analisi ha confermato invece che tutti gli elementi interattivi sono correttamente accessibili, dall'analisi manuale invece si può notare che sembra non esservi un contrasto sufficiente tra link visitati e non visitati (1.4.3).

Pagina esplora

La pagina consente agli utenti di scoprire nuovi libri, visualizzare i titoli più popolari e accedere a contenuti personalizzati dopo il login. L'estensione *SviluppAbile* ha segnalato falsi positivi per uso inappropriato del tabindex sugli elementi interattivi (2.1.1). L'analisi manuale ha confermato che tutti i link e i pulsanti sono etichettati in modo chiaro e tutte le immagini decorative o informative dispongono di testo alternativo appropriato (1.1.1, 1.3.1, 2.4.6). Inoltre, i contenuti personalizzati richiedono il login, ma i messaggi indicano chiaramente come accedere, senza compromettere l'accessibilità.

Pagina come-funziona

La pagina illustra il funzionamento della piattaforma *BookOverflow*, spiegando agli utenti i vantaggi dello scambio libri e guidandoli passo passo attraverso la creazione della libreria, la selezione dei desideri e le modalità di spedizione. L'estensione *SviluppAbile* non segnala alcun tipo di errore.

Conclusioni

In sintesi, la pagina risulta pienamente accessibile; le criticità segnalate da *SviluppAbile* costituiscono falsi positivi dovuti alle ristrette capacità di analisi dell'IA alla base dell'estensione.

5.1.6.3.1 F1-score

Nella tabella 5.5 vengono riportati i risultati ottenuti utilizzando l'estensione *SviluppAbile*.

Tabella 5.5: Tabella riassuntiva analisi *BookOverflow* tramite *SviluppAbile*

Errori trovati	Errori non trovati	Falsi positivi
	<ul style="list-style-type: none">• Contrasto insufficiente link visitati/non	<ul style="list-style-type: none">• Tabindex non adeguato
0	1	1

L' F_1 -score calcolato è $F_1 = 0$, valore che indica prestazioni non soddisfacenti in questo caso specifico. Ciò evidenzia la necessità di ulteriori miglioramenti nell'identificazione degli errori, soprattutto per quanto riguarda i falsi positivi.

5.1.7 Sito web: LuzzAuto

Di seguito vengono riportati alcuni dei test effettuati sul sito web [LuzzAuto](#).

5.1.7.1 Total Validator

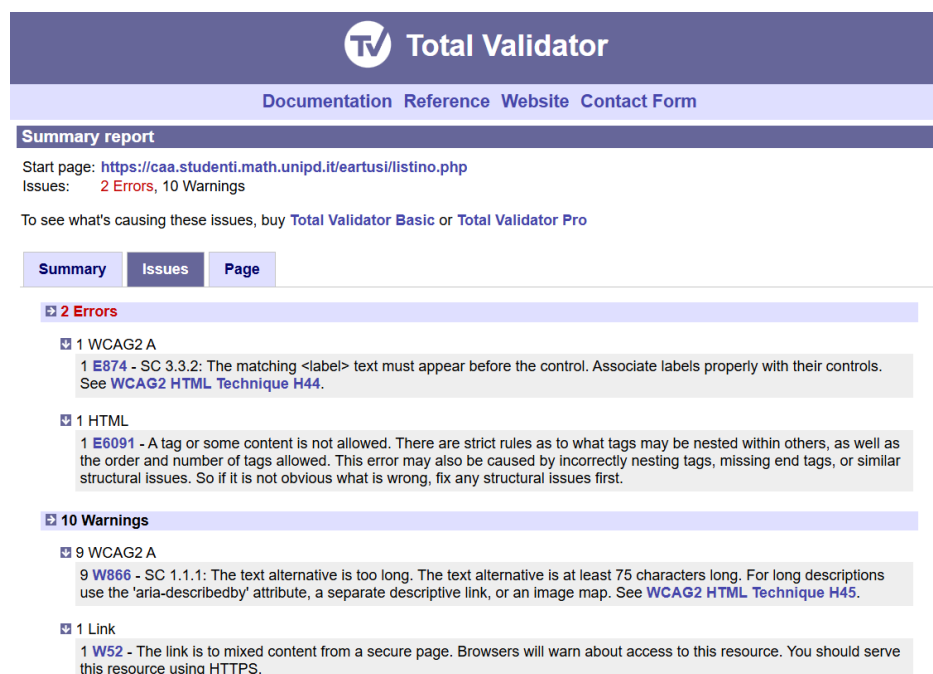


Figura 5.11: Analisi di Total Validator sul sito web *LuzzAuto*

Nella pagina `home.php` non sono presenti errori di accessibilità, ma solamente un "Warning" riguardante testi alternativi (`alt`) troppo lunghi. Nella pagina `listino.php` (come visibile in figura 5.11), sono segnalati due errori principali: uno relativo all'associazione corretta delle etichette ai rispettivi controlli e uno di struttura HTML, dovuto all'uso di tag non consentiti o a un errato annidamento degli stessi.

5.1.7.2 Lighthouse

Il report ha restituito un punteggio di 100/100 nella sezione *Accessibility*, indicando che, secondo le metriche automatiche adottate, non sono stati rilevati errori o problematiche di accessibilità (vedi figura 5.12).

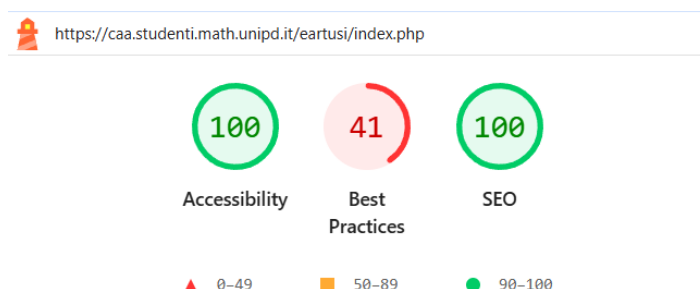


Figura 5.12: Analisi di Lighthouse sul sito web *LuzzAuto*

5.1.7.3 SviluppAbile

Di seguito viene illustrato il processo di test effettuato da *SviluppAbile* confrontando i risultati ottenuti dagli altri strumenti di validazione e integrando anche un controllo manuale basato sulle Web Content Accessibility Guidelines (WCAG 2.1). Lo studio è stato articolato sulle principali pagine del sito.

Pagina home

L'analisi della pagina tramite l'estensione *SviluppAbile* non ha rilevato errori di accessibilità. L'unico avviso segnalato riguarda testi alternativi (`alt`) troppo lunghi per alcune immagini, ma queste descrizioni dettagliate non compromettono la fruibilità dei contenuti per utenti con disabilità visive (*1.1.1 - Non-text Content*).

Pagina about

L'analisi della pagina non ha evidenziato errori di accessibilità e sono stati correttamente individuati gli elementi interattivi come link e pulsanti, che risultano etichettati e navigabili tramite tastiera (*2.1.1*), garantendo la conformità ai requisiti WCAG relativi all'accessibilità da tastiera e alla comprensibilità dei contenuti testuali. La pagina presenta una struttura semantica chiara (*1.3.1 - Info and Relationships*), immagini corredate da testi alternativi descrittivi (*1.1.1*) e informazioni accessibili sulle sezioni del sito.

Pagina test_drive

L'analisi della pagina non ha rilevato errori di accessibilità.

Conclusioni

L'analisi complessiva delle pagine principali tramite *SviluppAbile* mostra che l'estensione è efficace nell'individuare problemi come testi alternativi troppo lunghi e garantire la navigabilità da tastiera degli elementi interattivi. Tuttavia, alcune criticità relative alla struttura HTML, all'uso corretto dei tag semantici e alla compatibilità con lettori di schermo non vengono completamente rilevate. È pertanto consigliabile integrare l'uso di *SviluppAbile* con controlli manuali e verifiche basate sulle WCAG 2.1 ([1.1.1](#), [1.3.1](#), [2.1.1](#), [4.1.1](#)) per ottenere una valutazione completa dell'accessibilità del sito.

5.1.7.3.1 F1-score

Nella tabella [5.6](#) vengono riportati i risultati ottenuti utilizzando l'estensione *SviluppAbile*.

Tabella 5.6: Tabella riassuntiva analisi *LuzzAuto* tramite *SviluppAbile*

Errori trovati	Errori non trovati	Falsi positivi
<ul style="list-style-type: none">Le etichette dei controlli non sono state associate correttamente ai rispettivi elementi.	<ul style="list-style-type: none">Tag o contenuto non consentito.	
1	1	0

L' F_1 -score calcolato è $F_1 \approx 0.67$, un valore intermedio che evidenzia un discreto equilibrio tra errori rilevati e mancati. Pur non trattandosi di un risultato ottimale, esso mostra la capacità dell'estensione di individuare correttamente una parte significativa dei problemi, lasciando spazio a ulteriori miglioramenti.

5.2 Test “Modalità guidata”

Per effettuare i test utili a dimostrare l’efficacia di *SviluppAbile*, ho creato una semplice pagina *HTML* (vedi figura 5.13).

```
1 <!DOCTYPE html>
2 <html><head>
3 <title>Pagina di test</title>
4 </head>
5 <body>
6 <h1>Test</h1>
7 <p>Pagina creata per testare l'estensione
  SviluppAbile</p>
8 </body></html>
```

Figura 5.13: Pagina HTML base per effettuare i test della modalità guidata

Come si crea un <head> che contenga tutti gli elementi utili per una pagina accessibile?

Per creare un head accessibile, assicurati di includere i seguenti elementi:

1. ****Metatag Titolo (title)**:** Utilizzato dal browser come titolo della finestra della pagina.
2. ****Metatag Descrizione (description)**:** Descrive il contenuto della pagina e appare nella barra degli strumenti del browser e nei risultati dei motori di ricerca.
3. ****Metatag Chiave (keywords)**:** Non più importante come in passato, ma puoi utilizzarla se necessario per le ricerche.

Inoltre, assicurati di utilizzare un markup semantico corretto per le sezioni della pagina.

Ad esempio, puoi utilizzare ‘<header>’, ‘<nav>’, ‘<main>’, ‘<section>’, ‘<article>’, ‘<footer>’ in luogo di semplici div.

Infine, assicurati di aggiungere un elemento ‘alt’ a tutte le immagini con una descrizione del contenuto dell’immagine.

CODICE SUGGERITO:

```
1 <head>
2 <title>Pagina di test</title>
3 <meta charset="UTF-8">
4 <meta name="viewport" content="width=device-width, initial-scale=1.0">
5 <meta name="description" content="Pagina creata per testare l'estensione
SviluppAbile">
6 <meta name="keywords" content="test,sviluppo accessibile">
7 </head>
```

Come posso ora creare dei paragrafi che rispettino le linee guida BEM?

Per creare paragrafi in base alle linee guida BEM, puoi utilizzare la sintassi seguente:

1. Definisci un classico `'block__element'` per il contenuto del paragrafo: `'class="block__element"`
2. Utilizza `'block__element-modificatore'` per aggiungere modificatori: `'class="block__element-modificatore"`

In questo modo, i paragrafi saranno distinti e facili da identificare tramite la tecnica BEM.

Nota: Assicurati di sostituire `'-modificatore'` con il nome del modificatore effettivo nel tuo progetto.

CODICE SUGGERITO:

```
1 <h1 class="block__title">Titolo</h1>
2 <p class="block__text -grande">Testo di esempio con un modificatore per il
colore.</p>
3 <p class="block__text -rosso">Testo di esempio con altro contenuto.</p>
```

Come posso inserire immagini accessibili?

Inserire immagini accessibili significa garantire che le informazioni contenute nelle immagini siano anche disponibili ai utenti con disabilità visiva. Ciò può essere fatto utilizzando l'attributo 'alt' all'interno dell'elemento '':

In questo esempio, l'attributo 'alt' contiene una descrizione breve e concisa della foto. Se il browser non è in grado di visualizzare l'immagine (ad esempio, se si utilizza un lettore schermo), la descrizione contenuta nell'attributo 'alt' verrà visualizzata al posto dell'immagine.

Inoltre, per immagini decorative o grafici che non contengono informazioni importanti, è consigliabile utilizzare l'attributo 'alt=""', in modo da eliminare il testo alternativo dalla navigazione.

CODICE SUGGERITO:

```
1 
```

Come si crea una table rispettando le linee guida WCAG?

Per creare una tabella accessibile secondo le linee guida WCAG, è importante fornire un'esperienza utente inclusiva e facilitare l'accessibilità a tutte le persone, in particolare alle persone con disabilità visive che utilizzano strumenti di lettura ad alta voce o tastiera. Ecco alcuni passaggi per creare una tabella accessibile:

1. **Istruzioni per l'utilizzo della tabella**: fornire un'introduzione sulle informazioni presentate nella tabella e quali sono le colonne principali, aiutando gli utenti che utilizzano strumenti di lettura ad alta voce.
2. **Tabella semantica con role="presentation" impostato su false**: questo ti permetterà a un assistente di lettura di comprendere il contenuto della tabella e fornirlo correttamente all'utente.

Nota che ogni elemento della tabella (celle, intestazioni) deve avere un id o un attri-

buto aria-*, per fornire una descrizione dell'elemento e aiutare gli utenti con disabilità visive a comprendere il contenuto.

CODICE SUGGERITO:

```
1 <table>
2 <caption>Esempio di tabella accessibile</caption>
3 <thead>
4 <tr role="row">
5 <th scope="col" id="nome">Nome</th>
6 <th scope="col" id="età">Età</th>
7 </tr>
8 </thead>
9 <tbody>
10 <tr role="row">
11 <td headers="nome">Mario Rossi</td>
12 <td headers="età">32 anni</td>
13 </tr>
14 <tr role="row">
15 <td headers="nome">Giovanni Bianchi</td>
16 <td headers="età">35 anni</td>
17 </tr>
18 </tbody>
19 </table>
```

Come posso utilizzare gli attributi ARIA per migliorare l'accessibilità degli elementi interattivi?

Gli attributi ARIA (Accessible Rich Internet Applications) sono stati sviluppati per rendere le applicazioni web accessibili a tutti, nonostante eventuali disabilità. Sono utilizzati per aggiungere funzionalità di navigazione e controllo agli elementi interattivi come i bot-

toni e gli elementi dropdown.

Per migliorare l'accessibilità degli elementi interattivi tramite ARIA, ti suggerisco di utilizzare le seguenti strategie:

1. Utilizza il role di un elemento (ad esempio 'button', 'menuitem') per specificarne la funzione.
2. Aggiungi gli attributi aria-label e aria-describedby per fornire una descrizione dell'elemento interattivo.
3. Utilizza l'attributo aria-haspopup per indicare se un elemento è un menu o una lista.

In questo modo, l'elemento interattivo sarà più facile da navigare e controllare tramite strumenti assistivi come screen reader e mouse navigator.

Per ulteriori informazioni sugli attributi ARIA, ti consiglio di consultare la documentazione ufficiale dell'W3C.

CODICE SUGGERITO:

```
1 <!DOCTYPE html>
2 <html><head>
3 <title>Pagina di test</title>
4 </head>
5 <body>
6 <!-- Utilizzo del role e degli attributi ARIA per il bottone -->
7 <button role="button" aria-label="Clicca per eseguire l'azione"> Esegui azione
</button>
8 <!-- Utilizzo del role e degli attributi ARIA per il menu -->
9 <ul role="menu">
10 <li role="presentation">
11 <a href="#" aria-label="Opzione 1" aria-haspopup="true">Opzione 1</a>
12 <ul role="sottomenu">
13 <li role="presentation"><a href="#" aria-label="Sottopagina 1.1">Sottopagina
```



```
1.1</a></li>
14 <li role="presentation"><a href="#" aria-label="Sottopagina 1.2">Sottopagina
1.2</a></li>
15 </ul></li></ul>
16 </body></html>
```

5.3 Lista WCAG citate

- 1.1.1 - Non-text Content
- 1.3.1 - Info and Relationships
- 1.4.3 - Contrast (Minimum)
- 1.4.4 - Resize Text
- 2.1.1 - Keyboard
- 2.4.1 - Bypass Blocks
- 2.4.2 - Page Titled
- 2.4.4 - Link Purpose (In Context)
- 2.4.6 - Headings and Labels
- 2.4.7 - Focus Visible
- 2.4.8 - Location
- 2.4.9 - Link Purpose (Link Only)
- 3.3.1 - Error Identification
- 3.3.2 - Labels or Instructions
- 4.1.1 - Parsing

- 4.1.2 - Name, Role, Value

Capitolo 6

Conclusioni

Al termine dello stage, si può affermare che l'obiettivo principale del progetto è stato in parte raggiunto. È stata infatti realizzata un'estensione web, denominata *SviluppAbile*, capace di analizzare le pagine web e assistere gli sviluppatori nell'individuazione e correzione degli errori di accessibilità. L'attività ha richiesto più fasi, tra cui la progettazione dell'architettura dell'estensione, l'implementazione delle diverse funzionalità ed una consistente fase di testing finale.

Particolare attenzione è stata posta all'integrazione del chatbot, concepito per fornire spiegazioni chiare e in linguaggio naturale, così da rendere i contenuti accessibili anche a chi non ha conoscenze avanzate delle linee guida WCAG per l'accessibilità web.

6.1 Raggiungimento degli obiettivi

Il progetto *SviluppAbile* ha raggiunto pienamente l'obiettivo primario di realizzare uno strumento interattivo per il supporto allo sviluppo accessibile. Sono state implementate le funzionalità principali: la visualizzazione del codice sorgente delle pagine web e la possibilità di interagire con un chatbot integrato per analizzare la pagina web, ricevere chiarimenti e suggerimenti personalizzati.

Un ulteriore traguardo è stato il completamento della modalità guidata, che affianca all'analisi automatica un ambiente di sviluppo interattivo, composto da tre pannelli, in cui l'utente può sperimentare correzioni assistite del codice e scaricare direttamente i blocchi proposti in formato HTML. Questa funzionalità ha permesso di unire l'aspetto correttivo con quello formativo, rendendo lo strumento utile sia come validatore che come supporto didattico.

L'architettura basata esclusivamente su tecnologie web standard, conforme a Manifest V3, ha garantito la compatibilità con i browser moderni e la facilità di distribuzione

dell'estensione. In sintesi, il sistema ha dimostrato di soddisfare gli obiettivi stabiliti: fornire uno strumento intuitivo, accessibile e formativo, capace di supportare in tempo reale lo sviluppatore nella produzione di pagine web più accessibili.

Per quanto riguarda invece la qualità e la completa correttezza delle risposte e del codice generato, è emerso che l'IA talvolta segnala errori inesistenti o, al contrario, non rileva alcune criticità reali. Inoltre il codice proposto non sempre risulta pienamente conforme alle best-practice in materia di accessibilità. Pertanto, il miglioramento non riguarda l'estensione in sé, bensì la scelta del modello di intelligenza artificiale e il suo continuo affinamento.

6.1.1 Misurazione quantitativa dei requisiti soddisfatti

Il confronto con i requisiti inizialmente definiti (vedi paragrafo 2.3) evidenzia quanto segue:

- Requisiti obbligatori: 5/5 implementati (100%)
- Requisiti desiderabili: 4/4 soddisfatti (100%)
- Requisiti facoltativi: 1/3 rispettati (33%)

Quindi il tasso di completamento complessivo è di 10/12 requisiti (83%).

6.2 Sviluppi futuri

L'estensione web *SviluppAbile* costituisce una base su cui innestare ulteriori miglioramenti: grazie alla struttura modulare, sarà possibile introdurre nuove funzionalità senza compromettere la stabilità del sistema.

Un primo ambito di sviluppo riguarda l'integrazione di modelli di intelligenza artificiale più avanzati (come *ChatGPT* o sistemi analoghi), in grado di fornire risposte più precise, complete e soprattutto più rapide. Questo permetterebbe di aumentare l'affidabilità del chatbot e di garantire un supporto sempre più accurato agli sviluppatori.

Dal punto di vista dell'interfaccia utente, potrebbe essere introdotta una scansione automatica iniziale del DOM, che mostri fin da subito eventuali errori in un pannello dedicato, prendendo spunto da strumenti consolidati come *WAVE*.

Un'evoluzione significativa riguarda inoltre l'integrazione di tecniche di *Computer Vision*, che consentirebbero di estendere l'analisi anche agli aspetti visivi della pagina, come il contrasto cromatico, la leggibilità dei testi e le proporzioni degli elementi grafici. Altri sviluppi possibili includono il collegamento con validatori esterni (*WAVE*, *Lighthouse*), l'introduzione di un versionamento dei suggerimenti generati per facilitare il confronto tra diverse soluzioni e l'ottimizzazione delle prestazioni tramite caching o caricamento progressivo del DOM. In prospettiva, *SviluppAbile* potrebbe evolvere in un vero e proprio laboratorio per l'accessibilità, capace di coniugare analisi automatica, supporto interattivo e verifica visiva, offrendo agli sviluppatori uno strumento ancora più completo e formativo.

6.3 Consuntivo finale

Lo sviluppo dell'estensione *SviluppAbile* ha richiesto un totale di 300 ore, in linea con il monte ore preventivato. Il tempo è stato principalmente suddiviso tra lo studio del linguaggio JavaScript e l'apprendimento dello sviluppo di estensioni per browser, l'integrazione dell'*intelligenza artificiale* per l'analisi del DOM, il ripasso delle best-practice per l'accessibilità web e le attività finali di test per verificare il corretto funzionamento e l'affidabilità dell'estensione.

6.4 Competenze acquisite

Il progetto di stage ha rappresentato un'importante occasione di crescita tecnica e professionale, permettendo di acquisire competenze trasversali fondamentali. Dal punto di vista dello sviluppo web, l'esperienza ha consolidato la mia conoscenza di HTML,

sia nella struttura delle pagine che nella gestione degli elementi interattivi, e delle tecniche per produrre codice accessibile e semanticamente corretto. L'uso di strumenti di validazione del codice, come Total Validator, ha permesso di comprendere a fondo le problematiche legate all'accessibilità e di applicare concretamente le linee guida WCAG durante lo sviluppo.

Particolare attenzione è stata posta all'integrazione di funzionalità complesse in un ambiente modulare e compatibile con Manifest V3, comprendendo lo sviluppo di pannelli interattivi, la comunicazione tra script di background e content script e l'uso iniziale delle API di Chrome. L'implementazione del chatbot ha richiesto la selezione di un'IA utilizzabile, la cui scelta è ricaduta su Ollama, nonché la generazione dinamica di domande suggerite, il filtraggio delle risposte prodotte e la sincronizzazione con il codice visualizzato nei pannelli.

Il progetto ha avuto un carattere sperimentale, quindi non tutte le funzionalità da sviluppare erano definite fin dall'inizio; di conseguenza, non tutte le soluzioni ipotizzate sono state realizzabili, soprattutto a causa della complessità tecnica e dei vincoli di tempo. Questa esperienza ha comunque permesso di sviluppare capacità di problem solving, adattamento e gestione autonoma delle priorità, affrontando le difficoltà tipiche della sperimentazione.

Lo stage ha richiesto la gestione autonoma di circa 300 ore, pianificando in modo indipendente le attività giornaliere e settimanali, senza orari prefissati come in un contesto aziendale, e organizzando con cura il lavoro e le priorità per portare avanti l'intero progetto, affrontando così la sfida di un'organizzazione completamente autonoma dei tempi di lavoro.

Infine, il progetto ha permesso di sviluppare un approccio critico all'accessibilità digitale, comprendendo a fondo le linee guida WCAG e le strategie per guidare gli sviluppatori nel miglioramento della qualità dei propri siti.

In sintesi, l'esperienza ha combinato conoscenze teoriche e pratiche, fornendo strumenti concreti per la realizzazione di applicazioni web interattive, performanti e inclusive.

6.5 Valutazione personale

Lo svolgimento di questo progetto di stage ha rappresentato la mia prima vera esperienza di ricerca, un'occasione preziosa che mi ha permesso di andare oltre la semplice applicazione delle conoscenze acquisite durante il percorso universitario. Da un lato, ho potuto consolidare competenze tecniche già note e sperimentarne di nuove, dall'altro ho avuto l'opportunità di affrontare problemi a me sconosciuti con un approccio critico e metodologico, tipico delle attività di ricerca. Particolarmente stimolante è stata la possibilità di coniugare aspetti teorici, come i principi di accessibilità e le normative WCAG, con aspetti pratici legati alla realizzazione di uno strumento effettivamente utile agli sviluppatori. Ho trovato gratificante anche la libertà progettuale concessa, che mi ha spinto a sperimentare soluzioni non sempre funzionanti e a trasformare le difficoltà incontrate in momenti di crescita personale e professionale.

In conclusione, considero questa esperienza di stage un passaggio fondamentale del mio percorso formativo, tanto da rivalutare la possibilità di proseguire con una laurea magistrale (in *Intelligenza Artificiale* oppure in *Programming Languages, Systems and Algorithms*), anche se a tempo parziale.

Bibliografia

Siti

- [1] URL: <https://kinsta.com/it/blog/creare-estensione-chrome/>.
- [2] URL: <https://www.levelaccess.com/resources/must-have-wcag-2-1-checklist/>.
- [3] URL: <https://wcag.it/>.
- [4] URL: <https://www.w3.org/WAI/fundamentals/>.
- [5] URL: <https://developer.chrome.com/docs/extensions/develop/migrate/what-is-mv3?hl=it> (cit. a p. 16).
- [6] URL: <https://www.eff.org/deeplinks/2021/12/googles-manifest-v3-still-hurts-privacy-security-innovation> (cit. a p. 18).
- [7] URL: <https://developer.chrome.com/docs/extensions/develop/migrate/api-calls?hl=it>.
- [8] URL: <https://www.w3schools.com/html/>.
- [9] URL: <https://www.html.it/guide/guida-html/>.
- [10] URL: <https://www.01net.it/le-principali-novita-di-html5/>.
- [11] URL: <https://www.css3.info/>.
- [12] URL: <https://prismjs.com/>.
- [13] URL: <https://ollama.com/blog/gpt-oss>.
- [14] URL: <https://ollama.com/library/mistral>.
- [15] URL: <https://ollama.com/library/llama3>.
- [16] URL: <https://www.microsoft.com/it-it/>.
- [17] URL: <https://digital-strategy.ec.europa.eu/it/policies/web-accessibility-directive-standards-and-harmonisation> (cit. a p. 2).

- [18] URL: <https://www.treccani.it/vocabolario/parser/>.
- [19] *Application Programming Interface*. URL: https://www.treccani.it/enciclopedia/api_%28Lessico-del-XXI-Secolo%29/.
- [20] *Asincrono*. URL: <https://www.treccani.it/vocabolario/asincrono/>.
- [21] *Backend*. URL: https://it.wikipedia.org/wiki/Front-end_e_back-end.
- [22] *Browser*. URL: <https://www.treccani.it/vocabolario/browser/>.
- [23] *Chatbot*. URL: https://it.wikipedia.org/wiki/Chat_bot.
- [24] *ChatGPT*. URL: <https://www.treccani.it/enciclopedia/eol-chatgpt/>.
- [25] *Client-side*. URL: https://it.wikipedia.org/wiki/Lato_client.
- [26] *Cloud computing*. URL: [https://www.treccani.it/vocabolario/cloud-computing_res-4d54e944-8996-11e8-a7cb-00271042e8d9_\(Neologismi\)/](https://www.treccani.it/vocabolario/cloud-computing_res-4d54e944-8996-11e8-a7cb-00271042e8d9_(Neologismi)/).
- [27] *Debugging*. URL: <https://www.treccani.it/vocabolario/debugging/>.
- [28] *Desktop-first*. URL: <https://hvdig.co.uk/web-agency/mobile-first-vs-desktop-first>.
- [29] *Document Object Model*. URL: https://it.wikipedia.org/wiki/Document_Object_Model.
- [30] *eXtensible Markup Language*. URL: <https://it.wikipedia.org/wiki/XML>.
- [31] *Form*. URL: [https://www.treccani.it/vocabolario/form_\(Neologismi\)/](https://www.treccani.it/vocabolario/form_(Neologismi)/).
- [32] *GIT*. URL: [https://it.wikipedia.org/wiki/Git_\(software\)](https://it.wikipedia.org/wiki/Git_(software)).
- [33] *Graphics Processing Unit*. URL: <https://www.treccani.it/vocabolario/url/>.
- [34] *Intelligenza Artificiale*. URL: [https://www.treccani.it/enciclopedia/intelligenza-artificiale_\(Enciclopedia-della-Matematica\)/](https://www.treccani.it/enciclopedia/intelligenza-artificiale_(Enciclopedia-della-Matematica)/).
- [35] *Large Language Model*. URL: [https://www.treccani.it/vocabolario/neo-modello-linguistico-di-grandi-dimensioni_\(Neologismi\)/](https://www.treccani.it/vocabolario/neo-modello-linguistico-di-grandi-dimensioni_(Neologismi)/).
- [36] *Latenza*. URL: <https://www.treccani.it/vocabolario/latenza/>.

- [37] *Markup*. URL: https://it.wikipedia.org/wiki/Linguaggio_di_markup.
- [38] *Open source*. URL: <https://www.treccani.it/vocabolario/open-source/>.
- [39] *Open Weight*. URL: <https://invezz.com/it/notizie/2025/08/05/openai-rilascia-i-modelli-open-weight-cosa-sono-e-perche-cambia-tutto/>.
- [40] *Plugin*. URL: [https://it.wikipedia.org/wiki/Plugin_\(informatica\)](https://it.wikipedia.org/wiki/Plugin_(informatica)).
- [41] *Proof of Concept*. URL: https://it.wikipedia.org/wiki/Proof_of_concept.
- [42] *Quantizzazione*. URL: <https://www.treccani.it/vocabolario/quantizzazione/>.
- [43] *Script*. URL: <https://www.treccani.it/vocabolario/script/>.
- [44] *Server*. URL: <https://www.treccani.it/vocabolario/server/>.
- [45] *Transmission Control Protocol*. URL: https://it.wikipedia.org/wiki/Transmission_Control_Protocol.
- [46] *Uniform Resource Locator*. URL: <https://www.treccani.it/vocabolario/url/>.
- [47] *WebAssembly*. URL: <https://it.wikipedia.org/wiki/WebAssembly>.
- [48] *WebSocket*. URL: <https://it.wikipedia.org/wiki/WebSocket>.
- [49] *World Wide Web*. URL: https://it.wikipedia.org/wiki/World_Wide_Web.
- [50] *World Wide Web Consortium*. URL: https://it.wikipedia.org/wiki/World_Wide_Web_Consortium.