



Università degli Studi di Padova
Corso di laurea in Informatica



Jomp - Un salto nel lavoro

Relazione sul progetto di Tecnologie Web
a.a. 2017/2018

Componenti del gruppo

Sara Feltrin	sara.feltrin1@studenti.unipd.it	9999999
Lisa Parma	lisa.parma@studenti.unipd.it	1121284
Silvia Bazziato	silvia.bazziato@studenti.unipd.it	1096809

Sito web:/sfeltrin

	e-mail	password
Utente	utente@utente.com	utente01
Azienda	azienda@azienda.com	azienda01
Amministratore	admin@admin.com	admin

Indice

1	Introduzione	2
1.1	Abstract	2
1.2	Utenti destinatari	2
2	Progettazione	2
2.1	Database	2
3	Struttura	3
3.1	scjdissd	3
3.1.1	Nor chiu iuhhiu	3
4	Presentazione	3
4.1	Scelta CCS 3	3
4.2	desktop.css	3
4.3	print.css	4
5	Comportamento	4
5.1	PHP	4
5.1.1	Strutturazione pagine	4
5.1.2	Ricerca	4
5.1.3	Registrazione	4
5.1.4	Login	4
5.1.5	Area personale utente	4
5.1.6	Area personale aziende	4
5.1.7	Area personale amministratore	4
5.2	JavaScript	4
5.2.1	Validazione form registrazione utenti e aziende	5
5.2.2	Ridimensionamento header	5
5.2.3	Evidenziare la pagina in cui ci si trova nel menù	5
5.2.4	Menù a tendina	6
6	Accessibilità	6
6.1	scjdissd	6
6.1.1	Nor chiu iuhhiu	6
7	Divisione del lavoro	6

1 Introduzione

1.1 Abstract

Il progetto Jomp si propone di implementare un sito internet per facilitare la ricerca di lavoro da parte degli utenti e la ricerca di personale da parte delle aziende. Il sito dispone di una semplice e intuitiva barra di ricerca in cui si possono filtrare le proposte di lavoro per titolo dell'annuncio, luogo di lavoro e tipologia di impiego, dando luogo all'elenco di annunci corrispondenti. Il sito è stato sviluppato rispettando gli standard W3C, la separazione tra struttura, presentazione, comportamento e le regole di accessibilità richieste.

1.2 Utenti destinatari

Come menzionato precedentemente, il sito è rivolto a chi cerca o offre un lavoro, ragion per cui il target è definito da soggetti adulti (la registrazione può avvenire solo se si è maggiorenni). Al sito possono accedere diverse categorie di utenti:

- gli utenti registrati che possono salvare gli annunci per loro più interessanti e vedere informazioni più dettagliate riguardo all'annuncio;
- le aziende registrate che possono pubblicare i loro annunci di lavoro;
- gli utenti non registrati che posso ricercare annunci senza però poterli salvare o vedere i dettagli dell'offerta di lavoro (le aziende non registrate non possono aggiungere annunci di lavoro);
- l'amministratore che gestisce il sito può rimuovere utenti, aziende e annunci.

2 Progettazione

2.1 Database

Il sito, per funzionare correttamente, è collegato(?) ad un database costruito in linguaggio SQL. È stato scelto questo linguaggio in quanto tutti i componenti del gruppo lo conoscono e ne hanno dimestichezza. All'interno del database vi sono diverse tabelle, le principali sono Utenti, Aziende e Annunci, e ognuna di esse si appoggia ad altre tabelle minori come Consultazioni, Tipo, OrarioLavoro, ContrattoLavoro. Le ultime tre, in particolare, raccolgono la tipologia di caratteristiche che l'annuncio può assumere e si è scelto di inserirle in una tabella per evitare che l'utente possa modificare la voce nel menù a tendina (?).

3 Struttura

3.1 scjdisd

rvaefvafdgvaergvaergvae

3.1.1 Nor chiu iuhhiu

fgsbfgsb sfg sfv

4 Presentazione

La presentazione è stata fatta in CSS3 tenendola completamente separata dalla struttura attraverso due file esterni, uno per gestire la visualizzazione in un qualsiasi schermo e un'altro per gestire la stampa.

4.1 Scelta CCS 3

È stato deciso di utilizzare l'ultimo standard di CSS dato che, essendo uscito nel 2008, tutte le sue caratteristiche aggiuntive rispetto al CSS2 sono ben supportate in tutti i browser.

In particolare le novità che sono state utili sono:

- **Nuove pseudo-classi:** *:first-child*, *:last-child*, e *:nth-child(n)* sono state utilizzate per gestire il breadcrumb;
- **Nuove proprietà:** sono state largamente utilizzate le nuove proprietà *border-radius* e *box-shadow*;
- **Transizioni:** sono state modificate delle transizioni con *transition-property* e *transition-duration* per rendere meno forte il cambio di visualizzazione per l'utente;

4.2 desktop.css

Questo file raccoglie tutte le regole di stile utilizzate per la corretta presentazione di ogni elemento delle pagine web del sito. Sono stati sviluppati gerarchicamente tutti gli elementi in modo da non dover ripetere regole di stile per i discendenti.

Inizialmente sono state applicate delle regole per "azzerare" i valori di default del box-model che, per ogni browser, possono risultare diversi.

Come buona prassi, sono state individuate delle *classi* standard per elementi simili che ritroviamo in diverse pagine (come i form) e sono stati successivamente specializzati attraverso i diversi *id* per ottenere strutturazione simile ma stile diverso (form per registrazione, form per modifica dati, form per ricerca ecc..).

Per adattare la visualizzazione del sito a tutti i dispositivi e a tutti gli schermi è stato deciso

di utilizzare delle media query con dei determinati break-point per adattare il contenuto sia a schermi molto grande sia a schermi piccoli come gli smartphone. Questa soluzione è stata preferita all'utilizzare un secondo file solo per i cellulari in quando il sito è stato costruito in maniera responsive usando grandezze relative come *em* per gli elementi la cui dimensione dipendeva dal testo e percentuali per gli altri. In questo modo il sito risulta naturalmente molto più flessibile rispetto alle dimensioni dello schermo e le uniche differenze da apportare in una versione mobile sono cambiamenti di posizione di intere strutture o la loro eliminazione.

4.3 print.css

Questo foglio di stile si applica automaticamente quando un utente vuole stampare la pagina. Sono stati tolti tutti gli elementi visivi non strettamente necessari, quindi tutti i colori e le immagini di background o di presentazione; abbiamo mantenuto solo i bordi dei form colorati. Abbiamo rimosso anche il menu principale dall'header e il menu laterale delle aree personali in quanto non ne abbiamo ritenuta fondamentale la visualizzazione su una pagina stampata. Una caratteristica di questo file è la presenza di un forte raggruppamento dei tag che utilizzano la stessa regola. La parte iniziale sarà quindi un elenco di tag seguiti dalla regola che li accomuna. Questa decisione peggiora la leggibilità dei fogli di stile, ma ne alleggerisce notevolmente il peso.

5 Comportamento

5.1 PHP

Dichiamo in generale che lo abbiamo utilizzato per tutto e che è figo. oppure non scriviamo nulla qui

5.1.1 Strutturazione pagine

Abbiamo usato funzioni comuni a tutte le pagine

5.1.2 Ricerca

5.1.3 Registrazione

5.1.4 Login

5.1.5 Area personale utente

5.1.6 Area personale aziende

5.1.7 Area personale amministratore

5.2 JavaScript

Il linguaggio di scripting JavaScript è stato utilizzato per migliorare l'usabilità del sito. Si è tenuto conto del fatto che JavaScript può non funzionare su tutti i dispositivi che possono accedere al sito web (per non compatibilità o per preferenze dell'utente) così sono state assicurate funzioni equivalente server side con PHP.

In particolare con JavaScript sono state aggiunte le seguenti funzionalità:

5.2.1 Validazione form registrazione utenti e aziende

Prima di inviare i dati raccolti al server per controllarli con PHP, sono stati aggiunti dei controlli campo per campo che se non soddisfatti non permettono di inviare i dati al server. I controlli sono eseguiti *onBlur* su ogni input e permettono di controllare lunghezza massima e minima, se un certo dato soddisfa una certa espressione regolare (definite per e-mail e sito internet aziende) e se la ripetizione della password corrisponde alla prima inserita. Questi controlli real-time sono di grande aiuto all'utente per sapere subito se i dati inseriti corrispondono alla richiesta o c'è la necessità di modificarli.

Inoltre è servito anche un controllo per la gestione dell'input type *data*, tipo di input di HTML5 non gestito correttamente in tutti i browser. Nei browser in cui si vede l'input come stringa preformata (*gg/mm/aaaa*) o come calendario viene effettuato un controllo solo per vedere se l'input è stato immesso, negli altri browser dove invece si vede un normale input di testo viene eseguito il controllo anche sul formato dei dati immessi, che deve essere del tipo *aaaa/mm/gg* per aderire al type *data*.

5.2.2 Ridimensionamento header

Quando avviene uno scroll nelle pagine del sito, l'*header* (che per proprietà CSS risulta fisso) viene ridimensionato così da occupare meno spazio nella pagina visibile.

Per fare ciò attraverso JavaScript viene aggiunta dinamicamente la classe *small* all'*header* quando l'offset verticale della pagina risulta maggiore dell'offset iniziale cambiandone le proprietà grafiche. Quando si torna ad inizio pagina questa classe viene tolta facendo tornare l'*header* con le proprietà iniziali.

5.2.3 Evidenziare la pagina in cui ci si trova nel menù

Nel menù principale è stato scelto di evidenziare la pagina in cui ci si trova attraverso JavaScript così da evidenziare le differenze che ci sono state tra questa tecnica e quella che è stata fatta per i menù dentro le pagine personale in cui viene evidenziato ciò tramite PHP.

La funzione, molto semplice, prevede solamente un confronto tra l'attributo *href* dentro i tag *li* del menù e l'URL della pagina corrente. Quando viene trovata una corrispondenza viene aggiunta una classe che ne modifica le proprietà grafiche tramite CSS. Rispetto alla soluzione PHP di passare una variabile con la pagina corrente alla funzione che stampa il menù e di gestire molti sottocasi (uno per pagina) risulta una soluzione molto più veloce e semplice. Il suo limite però è nel fatto che JavaScript può essere disabilitato/non andare, mentre PHP no.

Dato che sotto l'*header* è presente il *breadcrumb* con la pagina corrente, anche in caso di disattivazione di JavaScript non si rischia che l'utente non capisca più in che pagina si trova.

5.2.4 Menù a tendina

Anche per il menù a tendina si è utilizzata la tecnica di aggiungere una classe ad un secondo menù che permettesse di cambiare l'attributo *display*. L'aggiunta di questa classe è data dall'evento *onClick* di un elemento (immagine menu ad hamburger) che è resa visibile da una media query solamente sotto gli 840px, quindi non solamente per mobile ma anche per finestre browser ristrette.

6 Accessibilità

6.1 scjdisd

rvaefvafdgvaergvaergvae

6.1.1 Nor chiu iuhhiu

fgsbfgsb sfg sfv

7 Divisione del lavoro

Lisa Parma:

- Struttura e comportamento pagine che non richiedono una sessione attiva;
- Login e registrazione utente (*login.php*, *UserSignIn.php*);
- Pagine personali utente:

-
- *UtDashboard.php*
 - *UtCercaAnnuncio.php*
 - *UtAnnunciSalvati.php*
 - *UtModificaDati.php*
 - CSS (file: *desktop.css*);
 - JavaScript (file: *javascript.js*).

Sara Feltrin:

- Struttura e comportamento pagine che non richiedono una sessione attiva;
- Login e registrazione azienda e login amministratore (*login.php*, *CompanySignIn.php*);
- Pagine personali azienda:
 - *AzDashboard.php*
 - *AzPubblicaAnnuncio.php*
 - *AzResocontoAnnunci.php*
 - *AzModificaDati.php*
- Pagine personali amministratore:
 - *AdminDashboard.php*
 - *AdminModUtenti.php*
 - *AdminModAziende.php*
 - *AdminModAnnunci.php*
- Database (file: *database.sql*).

Silvia Bazzoato:

- CSS per la stampa (file: *print.css*);
- pagine *sectionPrivacy.php*, *sectionTermCond.php*.