



UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di Laurea in Informatica

Integrazione di un sistema di gestione meeting per una piattaforma di recruiting

Relatore: Prof.ssa Daniela Micucci

Tesi di Laurea di:
Simone Lesinigo
Matricola 899540

Anno Accademico 2023-2024

Indice

Introduzione	1
1 Tecnologie utilizzate	2
1.1 Frontend	2
1.2 Backend	3
2 Approccio Proposto	5
2.1 Problema e Obiettivo	5
2.1.1 Caratteristiche dei Meeting	5
2.2 Metodologia	6
3 Step2: Text style	7
3.1 Bold	7
3.2 Italic	7
3.3 Underline	7
3.4 mixed	7
3.5 emph	7
4 Elements of the document	8
4.1 Images	8
4.2 Tables	8
4.3 List	8
4.4 Equations	9
4.5 Code Snippet	9
4.5.1 verbatim	9
4.5.2 listings	9
4.5.3 algpseudocode	10
4.5.4 algorithm	10
4.6 href	10
5 Positioning and Size	12
5.1 package adjustbox	12
5.2 Figure environment	12
6 Advanced images and table	14
6.1 Double figures	14
6.2 Wrapped images	14
6.2.1 Wrapped Tables	15

7	Additional concepts	16
7.1	Terms	16
7.2	Acronym	16
8	Multiple files management	17
8.1	Multiple file 1	17
8.2	Multiple file 1	17
A	Euristiche di Nielsen	19

Introduzione

Il campo del recruiting è in continua evoluzione, con una crescente necessità di strumenti che facilitino in modo semplice ed efficiente l'incontro tra domanda e offerta di lavoro. In questo contesto, la gestione dei colloqui di lavoro rappresenta una fase cruciale e complessa del processo di selezione.

L'obiettivo di questa esperienza di stage, svoltasi presso l'azienda Nesecom SRLS, è stato quello di integrare un sistema per la gestione dei colloqui di lavoro tramite meeting all'interno di un sito web in sviluppo, RisUma.

RisUma, acronimo di RISorse UMAne, è pensato per essere una piattaforma di recruiting intuitiva, che consente agli utenti e alle aziende di trovare efficacemente le opportunità di lavoro e le figure professionali richieste, gestendo l'intero processo di selezione su un'unica piattaforma.

Questo progetto è ancora nelle fasi iniziali del suo sviluppo, ma punta a eguagliare e superare le più famose piattaforme di recruiting grazie alle sue comode funzionalità.

Capitolo 1

Tecnologie utilizzate

Nel corso dello sviluppo del progetto sono state impiegate diverse tecnologie moderne, per garantire un'applicazione robusta, scalabile e facilmente manutenibile. La parte del progetto a me assegnata ha richiesto uno sviluppo full-stack, coinvolgendo sia il frontend che il backend.

1.1 Frontend

Il frontend rappresenta la parte dell'applicazione con cui l'utente interagisce direttamente. È stato sviluppato utilizzando **React** insieme a **TypeScript**, con l'obiettivo di garantire un'esperienza utente fluida e intuitiva.

Per gestire i pacchetti e le dipendenze, è stato utilizzato **npm** (Node Package Manager).

Inoltre, il design dell'interfaccia utente è stato standardizzato utilizzando il **MUI Theme** di Material-UI.

- **React:** React è una libreria JavaScript open source, sviluppata da Facebook, per la costruzione di interfacce utente. Utilizzando un approccio basato sui componenti, React consente di creare interfacce utente modulari e riutilizzabili. La sua capacità di aggiornare e mostrare efficientemente solo i componenti necessari in risposta ai cambiamenti rende React particolarmente adatto per lo sviluppo di applicazioni interattive e ad alte prestazioni. Inoltre, React permette di gestire lo stato delle applicazioni in modo prevedibile e scalabile, facilitando lo sviluppo di applicazioni complesse. [7]
- **TypeScript:** TypeScript è un linguaggio di programmazione open source sviluppato da Microsoft, che estende JavaScript aggiungendo tipi statici. L'adozione di TypeScript permette di ridurre significativamente gli errori durante lo sviluppo, grazie al controllo statico dei tipi che individua potenziali problemi prima ancora dell'esecuzione del codice. Inoltre, TypeScript facilita la manutenzione del codice in progetti di grandi dimensioni, migliorando la leggibilità e la documentazione attraverso la tipizzazione esplicita. [8]
- **MUI Theme (Material-UI):** Material-UI è una libreria di componenti React che implementa le linee guida del **Material Design** di Google.

Utilizzare Material-UI permette di sviluppare interfacce utente coerenti e professionali senza dover creare e stilizzare i componenti da zero. La libreria offre una vasta gamma di componenti pre-stilizzati e altamente personalizzabili, che facilitano la creazione di un design uniforme e accattivante. Inoltre, Material-UI supporta nativamente la creazione di temi, consentendo di personalizzare l'aspetto dell'applicazione in modo centralizzato.

- **npm:** Node Package Manager è il gestore di pacchetti predefinito per l'ecosistema JavaScript, utilizzato per installare e gestire le dipendenze necessarie per lo sviluppo delle applicazioni. Npm facilita l'integrazione di librerie e strumenti di terze parti, permettendo agli sviluppatori di accedere rapidamente a un vasto repository di pacchetti open source. Questo strumento è fondamentale per mantenere aggiornate le dipendenze del progetto e per gestire le versioni delle librerie utilizzate.

1.2 Backend

Il backend rappresenta la parte dell'applicazione che gestisce la logica di business, l'elaborazione dei dati e la comunicazione con il database. È responsabile del funzionamento lato server dell'applicazione, elaborando le richieste degli utenti e restituendo le risposte appropriate al frontend.

Il backend dell'applicazione è stato sviluppato utilizzando **Java** con il framework **Spring Boot**. Inoltre, viene utilizzato **Maven** come strumento di gestione delle dipendenze.

Per la gestione della persistenza dei dati è stato utilizzato **Java Persistence API (JPA)**, che permette di interfacciarsi con il database in modo efficiente e strutturato. Il database utilizzato si basa su **PostgreSQL 14**.

- **Java:** Java è un linguaggio di programmazione ad alto livello, orientato agli oggetti e a tipizzazione statica, sviluppato da Sun Microsystems (ora di proprietà di Oracle). È noto per la sua robustezza, sicurezza e portabilità, grazie alla Java Virtual Machine (JVM) che permette di eseguire il codice Java su qualsiasi piattaforma. Java è ampiamente utilizzato nello sviluppo di applicazioni enterprise, sistemi embedded, applicazioni mobili e web. La sua vasta ecosistema di librerie e strumenti lo rende una scelta ideale per lo sviluppo backend. [6]
- **Spring Boot:** Spring Boot è un framework open source per lo sviluppo di applicazioni Java basato su Spring, che semplifica il processo di creazione di app web e microservizi. Spring Boot offre una configurazione automatica dei componenti, riducendo significativamente il tempo necessario per avviare un progetto. Fornisce anche una serie di funzionalità integrate, come la gestione della sicurezza e il logging che facilitano lo sviluppo e la manutenzione delle applicazioni. [5]
- **Maven:** Maven è uno strumento di build automation e gestione delle dipendenze per progetti Java sviluppato dalla Apache Software Foundation. Utilizzando un modello basato su pom.xml (Project Object Model), Maven facilita la gestione del ciclo di vita del progetto, dall'inizializzazione alla distribuzione. Consente di integrare facilmente le librerie di terze

parti e di gestire le versioni delle dipendenze, migliorando la coerenza e la riproducibilità del build. [2]

- **Java Persistence API (JPA):** La Java Persistence API (JPA) è una specifica standard per la gestione della persistenza dei dati nelle applicazioni Java. JPA offre un modo standardizzato per mappare le classi Java agli oggetti del database, permettendo di eseguire operazioni CRUD (Create, Read, Update, Delete) in modo semplice e intuitivo.

Inoltre fornisce un linguaggio per effettuare query SQL, chiamato **JPQL** (Java Persistence Query Language), che è indipendente dal DBMS utilizzato.

Utilizzando JPA, gli sviluppatori possono concentrarsi sulla logica di business senza doversi preoccupare dei dettagli specifici dell'interazione con il database, migliorando la produttività e la manutenibilità del codice. [1]

- **PostgreSQL:** PostgreSQL è un sistema open source di database relazionale a oggetti sviluppato da Michael Stonebraker. È famoso per la sua affidabilità e per l'integrità dei dati, oltre alla community che continua a sostenerlo continuando a sviluppare nuove soluzioni. [3]

Capitolo 2

Approccio Proposto

2.1 Problema e Obiettivo

Il problema principale affrontato riguarda:

- Per le **aziende**: ottimizzazione dell'efficienza nella gestione dei meeting per i colloqui di lavoro, fornendo uno strumento adeguato e completo di tutte le funzionalità necessarie. L'obiettivo principale è migliorare l'esperienza dei recruiter, che potrebbero trovarsi a gestire un elevato numero di meeting.
- Per gli **utenti**: eliminare la necessità per i candidati di segnare manualmente gli appuntamenti, offrendo loro uno strumento che ricordi in modo comodo e automatico tutti i colloqui programmati, facilitandone l'accesso.

2.1.1 Caratteristiche dei Meeting

Un ulteriore obiettivo riguarda le proprietà che devono caratterizzare un meeting:

- Non è richiesto avere un account per partecipare al meeting, ma sarà sufficiente inserire un nome a propria scelta al momento dell'accesso.
- Nessuno può accedere al meeting prima di un quarto d'ora rispetto all'orario di inizio prestabilito o dopo che sia trascorso l'orario di fine.

2.2 Metodologia

Di seguito sono riportati i passi seguiti per la realizzazione di questo progetto. Si vuole far notare che RisUma è un'idea del direttore aziendale, e pertanto non vi è un cliente con cui confrontarsi. Inoltre, poiché non sono state disponibili direttive scritte ma solo alcune indicazioni orali, tutte le scelte sono state prese a mia completa discrezione.

1. **Servizio:** il primo passo è stato scegliere quale servizio di terze parti utilizzare per le chiamate online. Si cercava un servizio che:
 - (a) avesse le caratteristiche richieste
 - (b) si integrasse bene con il progetto (API disponibili, compatibilità con le tecnologie utilizzate...)
 - (c) fosse facilmente usabile anche dagli utenti meno esperti

Per tali motivi sono state prese in considerazione le piattaforme più note, quali Cisco Webex, Google Meet e Zoom.

2. **Interfaccia utente:** è stato scelto il design della pagina frontend. Basandosi anche sulle 10 euristiche di Jakob Nielsen, si sono perseguiti i seguenti obiettivi:
 - (a) Adattare il sistema al mondo reale. (Euristica 2)
 - (b) La pagina doveva essere facile da navigare, riducendo al minimo il carico cognitivo per l'utente. (Euristica 6)
 - (c) Optare per uno stile minimalista e intuitivo. (Euristica 7)
 - (d) Implementare funzionalità di feedback per guidare l'utente in caso di errori o problemi durante l'utilizzo della pagina. (Euristiche 8 e 9)
3. **Integrazione delle API:** sono state integrate le API per consentire le operazioni CRUD sui meeting, sviluppando contemporaneamente le parti di frontend e backend secondo il seguente ordine:
 - (a) **Creazione:** implementazione della funzionalità per creare nuovi meeting, gestendo i parametri necessari come data, ora e invitati.
 - (b) **Recupero:** realizzazione della logica per recuperare e visualizzare i meeting di un utente.
 - (c) **Modifica:** implementazione della possibilità di modificare i parametri dei meeting già esistenti.
 - (d) **Eliminazione:** implementazione della funzionalità per annullare meeting programmati.

Questa parte è stata cruciale nel corso della mia esperienza di stage in quanto ha costituito il nucleo essenziale della pagina che ho progettato e integrato.

4. **Test:** Il sistema è stato sottoposto da parte mia e di alcuni colleghi a dei test per garantire che tutte le funzionalità operassero correttamente.
5. **Deploy:** È stato eseguito un deploy su un server per testare le prestazioni in un ambiente più realistico rispetto a quello locale.

Capitolo 3

Step2: Text style

3.1 Bold

Example of bold **text**.

3.2 Italic

Example of italic *text*.

3.3 Underline

Example of underlined text.

3.4 mixed

Example of mixed ***word***

3.5 emph

Example of emph *word*

Example of emph word

Capitolo 4

Elements of the document

4.1 Images



Figura 4.1: Example image

Reference to the image 4.1.

4.2 Tables

Reference to the table 4.1.

4.3 List

Example of an unordered list:

- List entries start with the `\item` command.

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

Tabella 4.1: Example table

- Individual entries are indicated with a black dot, a so-called bullet.
- The text in the entries may be of any length.

Example of an ordered list:

1. This is the first entry in our list.
2. The list numbers increase with each entry we add.

Example of nested list:

1. First level item
2. First level item
 - (a) Second level item
 - (b) Second level item
 - i. Third level item
 - ii. Third level item
 - A. Fourth level item
 - B. Fourth level item

4.4 Equations

This is an example of an in-line equation: $E = MC^2$

This is an example of an display mode equation:

$$E = MC^2 \tag{4.1}$$

This is a reference to the Equation 4.1.

4.5 Code Snippet

4.5.1 verbatim

Text enclosed inside `\texttt{verbatim}` environment
is printed directly
and all `\LaTeX{}` commands are ignored.

4.5.2 listings

```
import numpy as np

def incmatrix(genl1, genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int) #dummy variable

    #compute the bitwise xor matrix
```

```

M1 = bitxormatrix(genl1)
M2 = np.triu(bitxormatrix(genl2),1)

for i in range(m-1):
    for j in range(i+1, m):
        [r,c] = np.where(M2 == M1[i,j])
        for k in range(len(r)):
            VT[(i)*n + r[k]] = 1;
            VT[(i)*n + c[k]] = 1;
            VT[(j)*n + r[k]] = 1;
            VT[(j)*n + c[k]] = 1;

        if M is None:
            M = np.copy(VT)
        else:
            M = np.concatenate((M, VT), 1)

VT = np.zeros((n*m,1), int)

return M

```

4.5.3 algpseudocode

```

i ← 10
if i ≥ 5 then
    i ← i − 1
else
    if i ≤ 3 then
        i ← i + 2
    end if
end if

```

4.5.4 algorithm

reference to algorithm 1

4.6 href

For further references see Something Linky or go to the next url: <http://www.overleaf.com>

Algorithm 1 An algorithm with caption

Require: $n \geq 0$

Ensure: $y = x^n$

$y \leftarrow 1$

$X \leftarrow x$

$N \leftarrow n$

while $N \neq 0$ **do**

if N is even **then**

$X \leftarrow X \times X$

$N \leftarrow \frac{N}{2}$

▷ This is a comment

else if N is odd **then**

$y \leftarrow y \times X$

$N \leftarrow N - 1$

end if

end while

Capitolo 5

Positioning and Size

5.1 package adjustbox



5.2 Figure environment



Figura 5.1: h image



Figura 5.2: Top image



Figura 5.3: ! image

Capitolo 6

Advanced images and table

6.1 Double figures

Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque...



(a) Caption1



(b) Caption 2

Figura 6.1: Caption for this figure with two images

Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

6.2 Wrapped images

Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat.



Figura 6.2: Caption1

Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

6.2.1 Wrapped Tables

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Tabella 6.1: A wrapped table going nicely inside the text.

Header-1	Header-1	Header-1
2	3	5
2	3	5
2	3	5

Capitolo 7

Additional concepts

7.1 Terms

The Latex typesetting markup language is specially suitable for documents that include maths. Formula are rendered properly an easily once one gets used to the commands.

7.2 Acronym

Given a set of numbers, there are elementary methods to compute its **gcd!** (**gcd!**), which is abbreviated gcd. This process is similar to that used for the lcm.

Capitolo 8

Multiple files management

8.1 Multiple file 1

Hello world!



Figura 8.1: U14

Hello, here is some text without a meaning...

8.2 Multiple file 1

Bibliografia

- [1] V. Racca. “Introduzione a JPA e mapping delle relazioni.” (2020), indirizzo: <https://www.vincenzoracca.com/blog/framework/jpa/jpa-relations/>. (Data di accesso: 25.06.2024).
- [2] A. Gatu. “Cos’è Maven, a cosa serve e come si usa.” (2024), indirizzo: <https://www.nextre.it/cose-maven-si-usa/>. (Data di accesso: 25.06.2024).
- [3] Geekandjob. “Cos’è PostgreSQL.” (), indirizzo: <https://www.geekandjob.com/wiki/postgresql>. (Data di accesso: 25.06.2024).
- [4] S. Lesinigo, *Appunti del corso di Interazione Uomo-Macchina*, Università degli Studi di Milano-Bicocca, Docente: Cabitza Federico Antonio Niccolò Amedeo, Anno Scolastico 2023-2024.
- [5] Microsoft. “Che cos’è Spring Boot di Java?” (), indirizzo: <https://azure.microsoft.com/it-it/resources/cloud-computing-dictionary/what-is-java-spring-boot>. (Data di accesso: 25.06.2024).
- [6] Wikipedia. “Java (linguaggio di programmazione).” (), indirizzo: [https://it.wikipedia.org/wiki/Java_\(linguaggio_di_programmazione\)](https://it.wikipedia.org/wiki/Java_(linguaggio_di_programmazione)). (Data di accesso: 24.06.2024).
- [7] Wikipedia. “React (web framework).” (), indirizzo: [https://it.wikipedia.org/wiki/React_\(web_framework\)](https://it.wikipedia.org/wiki/React_(web_framework)). (Data di accesso: 25.06.2024).
- [8] Wikipedia. “TypeScript.” (), indirizzo: <https://it.wikipedia.org/wiki/TypeScript>. (Data di accesso: 25.06.2024).

Appendice A

Euristiche di Nielsen

Delineate da Jakob Nielsen in collaborazione con Rolf Morich nel 1994, sono un insieme di principi generali ampiamente utilizzati ancora oggi che supportano la progettazione di sistemi interattivi usabili [4]:

1. Visibilità dello stato del sistema
2. Corrispondenza tra sistema e mondo reale
3. Controllo e libertà dell'utente
4. Consistenza e standard
5. Riconoscimento piuttosto che ricordo
6. Flessibilità ed efficienza d'uso
7. Estetica e minimalismo
8. Prevenzione degli errori
9. Aiuto nel rilevare e correggere gli errori
10. Aiuto e documentazione