

Relazione Tirocinio Curricolare

Alex Bruno

Matricola: s248220

Corso di Laurea: Ingegneria Informatica

Azienda: GetApper Fragments s.r.l.

Periodo: 27/10/2023 - 30/10/2024

Indice

1. **Introduzione**
 - 1.1. Contesto del tirocinio
 - 1.2. Obiettivi e motivazioni
2. **Descrizione dell'azienda**
 - 2.1. Storia e missione
 - 2.2. Struttura organizzativa
3. **Obiettivi del tirocinio**
 - 3.1. Obiettivi specifici
 - 3.2. Competenze da acquisire
 - 3.3. Aspettative iniziali
4. **Attività svolte**
 - 4.1. Progetti principali
 - 4.2. Tecnologie e strumenti utilizzati
 - 4.3. Sfide affrontate e soluzioni adottate
5. **Esplorazione Dettagliata dei Progetti**
 - 5.1. BMT - Business Management Tool
 - 5.1.1. Richiesta 1: Supporto ai Promemoria
 - 5.1.2. Richiesta 2: Form nuovo promemoria
 - 5.2. Sale Assistant
 - 5.2.1. Task 1: Creazione Form per Generazione Istruzioni da Link
 - 5.2.2. Task 2: Aggiunta Chip Stato nella organizationList
 - 5.2.3. Task 3: Logout lato admin
 - 5.2.4. Task 4: Creazione dialog per Assistente Attivo
 - 5.2.5. Task 5: Generazione in tempo reale delle risposte di OpenAI
6. **Competenze acquisite**
 - 6.1. Competenze tecniche
 - 6.2. Competenze trasversali
7. **Riflessioni personali**
 - 7.1. Valutazione dell'esperienza
 - 7.2. Confronto tra aspettative e realtà
 - 7.3. Lezioni apprese
8. **Conclusioni**
 - 8.1. Sintesi dei punti chiave
9. **Bibliografia**
 - 9.1. Riferimenti, materiali e fonti consultate
 - 9.2. JavaScript e Frameworks
 - 9.3. Backend e Database
 - 9.4. Gestione delle Dipendenze e Versionamento
 - 9.5. Apprendimento e Pratica
 - 9.6. Risorse Aggiuntive
 - 9.7. Documentazione Tecnica
 - 9.8. Materiale fornito dall'azienda
 - 9.9. Video di approfondimento

1. Introduzione

La presente relazione ha l'obiettivo di documentare e analizzare l'esperienza di tirocinio svolta in qualità di web developer presso GetApper Fragments s.r.l. . Questo periodo ha rappresentato un'opportunità significativa per integrare le conoscenze teoriche acquisite durante il percorso accademico con esperienze pratiche nel contesto lavorativo.

L'azienda, riconosciuta per lo sviluppo di applicazioni web particolarmente sfidanti dal punto di vista tecnologico e prestazionale, soprattutto per ciò che concerne lo sviluppo software, la consulenza IT e la formazione di sviluppatori, si è distinta per la sua capacità di creare soluzioni digitali all'avanguardia. Lavorare all'interno di un ambiente dinamico e stimolante ha permesso di affrontare sfide concrete e di applicare le competenze tecniche in modo efficace.

Gli obiettivi prefissati per il tirocinio includevano l'acquisizione di competenze pratiche specifiche nell'ambito dello sviluppo web, la comprensione delle dinamiche di lavoro in team e la capacità di affrontare e risolvere problemi reali. Tali obiettivi sono stati perseguiti attraverso un approccio pratico e orientato ai risultati, incoraggiando un continuo scambio di idee e feedback tra i membri del team.

Durante il tirocinio, ho avuto l'opportunità di apprendere nuove tecnologie e linguaggi di programmazione, oltre a collaborare a progetti reali, contribuendo attivamente alla creazione di varie componenti di una web app e allo sviluppo di un sistema di gestione. Questa esperienza non solo ha permesso un miglioramento delle competenze tecniche, ma ha anche facilitato lo sviluppo di abilità interpersonali e di lavoro in team, fondamentali nel contesto ingegneristico.

La relazione è strutturata in diverse sezioni che esploreranno la descrizione dell'azienda, gli obiettivi stabiliti, le attività svolte, le competenze acquisite, nonché le riflessioni personali e le conclusioni relative all'intera esperienza di tirocinio. Attraverso questa analisi, si intende fornire una visione chiara del valore formativo e professionale del tirocinio presso GetApper Fragments s.r.l. .

1.1 Contesto del tirocinio

Il tirocinio ha avuto luogo in un periodo caratterizzato da una crescente domanda di soluzioni digitali. Questo contesto ha reso fondamentale l'acquisizione di competenze pratiche e teoriche nel campo dello sviluppo web. La scelta di intraprendere un tirocinio in questo ambito si è rivelata giusta, poiché ha fornito un'opportunità unica per comprendere le dinamiche e le necessità di un settore in continua evoluzione.

1.2 Obiettivi e motivazioni

Gli obiettivi del tirocinio sono stati definiti in base alle necessità personali di sviluppo professionale e alle aspettative del settore. In particolare, ho mirato a:

- **Acquisire competenze tecniche:** Approfondire le conoscenze relative ai linguaggi di programmazione e agli strumenti di sviluppo web, tra cui HTML, CSS, JavaScript, TypeScript e framework come React e Node.js.
- **Sviluppare competenze trasversali:** Migliorare le capacità di comunicazione, collaborazione e problem solving all'interno di un team di lavoro.
- **Comprendere il ciclo di vita dello sviluppo software:** Approfondire le fasi di progettazione, sviluppo, testing e deployment.

La motivazione principale alla base della scelta di questo tirocinio è stata la volontà di coniugare le competenze tecniche con l'esperienza pratica. Questo approccio ha contribuito a una formazione completa, preparandomi ad affrontare le sfide future nel campo dello sviluppo e ingegneria del software.

2. Descrizione dell'azienda

2.1. Storia e missione

GtApper Fragments s.r.l. si è proposta di offrire soluzioni ad hoc, mirando a soddisfare le esigenze specifiche dei propri clienti. L'azienda opera in un contesto altamente competitivo, dove la rapidità di adattamento e la capacità di anticipare e seguire le tendenze del mercato sono fondamentali per il successo. A tal fine, GetApper Fragments si impegna a comprendere profondamente il settore dei suoi clienti e le loro sfide, consentendo di sviluppare prodotti su misura, in grado di integrare funzionalità avanzate e tecnologie all'avanguardia.

La missione dell'azienda è quella di diventare un punto di riferimento nel panorama tecnologico, fornendo prodotti e servizi che non solo rispondano alle aspettative dei clienti, ma che le superino. Ciò è anche possibile grazie a un costante investimento in ricerca e formazione.

2.2. Struttura organizzativa

La struttura organizzativa di GetApper Fragments s.r.l. è caratterizzata da un modello flessibile e collaborativo, pensato per adattarsi alle esigenze di un mercato in continua evoluzione. Il team è composto da sviluppatori esperti, come nello sviluppo front-end, il back-end, e l'integrazione di intelligenza artificiale. Questa diversità di competenze consente una gestione più efficace dei progetti, garantendo che ogni aspetto dello sviluppo venga affrontato con attenzione e efficienza.

La comunicazione tra i membri del team avviene in modo costante e diretto, grazie all'utilizzo di tecnologie moderne come Slack, che facilita il dialogo immediato e la condivisione di idee, e GitHub, che permette una gestione collaborativa del codice. Questi strumenti non solo migliorano l'efficienza del lavoro quotidiano, ma consentono anche una rapida risoluzione dei problemi, poiché ogni membro del team può contribuire attivamente a discutere e affrontare le sfide tecniche che emergono durante lo sviluppo.

In sintesi, la struttura organizzativa di GetApper Fragments s.r.l. non è solo un modo di lavorare, ma un vero e proprio approccio che valorizza la collaborazione.

3. Obiettivi del tirocinio

3.1. Obiettivi specifici

Gli obiettivi del tirocinio sono stati definiti per permettere l'apprendimento e il migliore contributo all'interno dell'azienda, tenendo conto delle necessità specifiche del team e delle aspettative di crescita personale. Tra i principali obiettivi si sono distinti:

- **Acquisire competenze pratiche nella programmazione web:** Durante il tirocinio, è stata fornita l'opportunità di lavorare su vari aspetti dello sviluppo web, inclusi front-end e back-end. Questo ha permesso di comprendere a fondo le dinamiche della programmazione, dall'interazione con l'utente alla gestione dei dati nel server.
- **Collaborare attivamente alla realizzazione di progetti:** Partecipare a progetti reali ha rappresentato un aspetto cruciale dell'esperienza di tirocinio. Lavorando a stretto contatto con il team, è stato possibile applicare le conoscenze teoriche in situazioni pratiche, contribuendo attivamente alla scrittura di codice, il quale in seguito è stato integrato nel progetto stesso.
- **Comprendere il funzionamento di un ambiente lavorativo tecnologico:** L'esperienza in azienda ha fornito una visione chiara delle dinamiche quotidiane di un ambiente lavorativo, inclusi processi, metodologie di sviluppo e l'importanza della collaborazione tra i membri del team.

3.2. Competenze da acquisire

Durante il tirocinio, si è mirato a sviluppare competenze sia tecniche sia trasversali, essenziali per una carriera futura nel settore.

Competenze tecniche da acquisire:

- **Padronanza dei principali linguaggi di programmazione e strumenti del settore:** Tra gli obiettivi principali, vi era quello di acquisire familiarità con linguaggi come JavaScript e TypeScript, oltre a framework come React e tecnologie NoSQL come MongoDB. Il tirocinio si proponeva di fornire le basi per comprendere e applicare le migliori pratiche di programmazione e gli standard tecnologici comunemente utilizzati nell'industria.
- **Applicazione dei principi CRUD:** Comprendere e applicare le operazioni fondamentali di creazione, lettura, aggiornamento e cancellazione (CRUD) è stato un altro obiettivo chiave. L'acquisizione di questa competenza è stata mirata alla costruzione di applicazioni web interattive e alla gestione delle basi di dati in modo efficace.
- **Implementazione dell'architettura MVC:** Si mirava anche a sviluppare la capacità di utilizzare il modello architetturale Model-View-Controller (MVC). Apprendere l'MVC ha facilitato l'organizzazione del codice e ha consentito di comprendere la separazione delle responsabilità tra la logica di business, l'interfaccia utente e la gestione dei dati, rendendo il codice più modulare e gestibile.

- **Gestione dello stato con Redux:** Un obiettivo formativo era l'implementazione di Redux, una libreria per la gestione centralizzata dello stato nelle applicazioni JavaScript. Acquisire familiarità con Redux ha permesso di capire come strutturare e mantenere il flusso di dati in modo efficace, migliorando prevedibilità e debugging all'interno di contesti complessi.
- **Familiarità con Git e GitHub per il versionamento del codice:** L'utilizzo di Git e GitHub era tra le competenze essenziali da sviluppare per la gestione delle versioni e la collaborazione. La capacità di comprendere e lavorare con Git ha migliorato l'integrazione continua e la gestione delle modifiche nel codice, promuovendo una gestione collaborativa e tracciabile dei progetti.

Competenze trasversali da acquisire:

- **Abilità relazionali e capacità di lavorare in team:** La capacità di comunicare efficacemente e collaborare all'interno di un team era una competenza chiave da sviluppare. Il tirocinio ha fornito l'opportunità di interagire con colleghi e collaboratori, condividere idee e soluzioni e formulare domande mirate per favorire una comprensione approfondita delle problematiche e delle soluzioni.
- **Problem solving:** L'obiettivo formativo includeva anche lo sviluppo della capacità di identificare e risolvere i problemi in modo sistematico. Affrontare le sfide in un ambiente dinamico richiedeva un approccio orientato alla risoluzione pratica e innovativa dei problemi, includendo l'utilizzo di risorse online e la documentazione tecnica per un apprendimento autonomo ed efficace.

3.3. Aspettative iniziali

Le aspettative iniziali includevano:

- **Una integrazione nel team di lavoro:** Si sperava di diventare parte attiva del gruppo, contribuendo in modo significativo ai progetti.
- **Possibilità di contribuire in modo significativo ai progetti:** Ci si aspettava di ricevere incarichi sfidanti e stimolanti, che avrebbero permesso di applicare le proprie competenze e imparare dai successi e dai fallimenti.
- **Ricevere un feedback costruttivo:** L'auspicio era quello di ricevere feedback regolari e utili dai professionisti esperti dell'azienda, per poter migliorare continuamente e adattare il proprio approccio alle esigenze del team e dei progetti.
- **Acquisire competenze tecnologiche:** Ci si aspettava di sviluppare una conoscenza pratica e approfondita delle tecnologie utilizzate in azienda, in particolare dei linguaggi e degli strumenti chiave come JavaScript, TypeScript, React e MongoDB, per prepararsi adeguatamente per le sfide future nel settore dello sviluppo web.

4. Attività svolte

4.1. Progetti principali

Durante il periodo di tirocinio, ho avuto l'opportunità di partecipare a diversi progetti significativi, tra cui:

BMT - Business Management Tool:

Nel progetto Business Management Tool (BMT), ho partecipato allo sviluppo di un sistema di gestione interno per GetApper, pensato per semplificare l'organizzazione e l'aggiornamento delle informazioni aziendali in tempo reale. Il sistema è stato progettato per consentire agli utenti di monitorare e gestire attività, promemoria e risorse, supportando un'organizzazione più efficiente e facilitando il rispetto delle scadenze.

Il mio ruolo ha coperto sia il frontend che il backend, permettendomi di contribuire a un'ampia gamma di funzionalità. Ho utilizzato React per creare un'interfaccia utente reattiva e intuitiva, insieme a Redux per la gestione dello stato dell'applicazione, e Node.js e MongoDB per configurare un backend robusto e scalabile. Tra le principali funzionalità implementate:

- **Sistema di Promemoria:** Ho sviluppato un modulo che consente di impostare promemoria personalizzati per attività specifiche, associabili a utenti responsabili. Questo sistema include notifiche automatiche per garantire che le scadenze siano rispettate e gli utenti restino aggiornati.
- **Form per l'Inserimento delle Attività:** Ho creato un modulo per l'aggiunta di nuove attività, con campi configurabili per la scelta del responsabile, la strategia di promemoria, e altre informazioni chiave. Questo form utilizza selettori avanzati per facilitare la scelta dei collaboratori e dei progetti coinvolti.

Grazie a questo progetto, ho potuto approfondire l'integrazione tra frontend e backend in un sistema di gestione aziendale complesso, lavorando su soluzioni tecniche che migliorano la produttività e l'organizzazione aziendale.

Sale Assistant:

Nel progetto Sale Assistant, ho contribuito allo sviluppo di un assistente virtuale avanzato per supportare le aziende nel processo di vendita e nella gestione delle interazioni con i clienti. L'obiettivo era creare un sistema intuitivo e proattivo, capace di rispondere in modo dinamico alle richieste degli utenti e gestire efficacemente i dati e le comunicazioni, favorendo la generazione di lead e il supporto personalizzato.

Il mio ruolo ha riguardato sia il frontend che il backend, utilizzando React e Redux per costruire un'interfaccia intuitiva e reattiva, e Node.js con MongoDB per garantire una gestione dati affidabile. Ho sviluppato alcune funzionalità chiave, tra cui:

- **Sistema di Stato e Notifiche:** Ho implementato una funzione di stato per monitorare l'attività e la disponibilità dell'assistente, includendo notifiche che avvisano l'utente di eventuali azioni necessarie o dell'opportunità di richiedere supporto.

- **Risposte in tempo reale:** Utilizzando le API di OpenAI con funzionalità di streaming, ho integrato una risposta dinamica e parola per parola, che simula una conversazione naturale e coinvolgente per l'utente.

Questa esperienza mi ha permesso di comprendere meglio i sistemi di assistenza virtuale, offrendo un'interazione fluida tra frontend e backend e migliorando l'esperienza utente grazie a un chatbot efficiente e versatile.

Questi progetti non solo hanno richiesto competenze tecniche, ma hanno anche richiesto la collaborazione con altri membri del team, in particolare durante l'integrazione con il codice scritto precedentemente da loro, affinché tutte le parti funzionassero in modo sinergico.

4.2. Tecnologie e strumenti utilizzati

Nel contesto dello sviluppo web, è fondamentale la conoscenza e l'utilizzo di un'ampia varietà di tecnologie. Durante il mio tirocinio presso GetApper Fragments s.r.l., ho avuto l'opportunità di lavorare con diverse tecnologie e strumenti, ciascuno dei quali ha contribuito significativamente alla realizzazione dei progetti. Di seguito sono descritte in dettaglio le tecnologie chiave utilizzate.

2.1 Linguaggi di programmazione

HTML, CSS e JavaScript: Questi tre linguaggi rappresentano le fondamenta dello sviluppo web. HTML (HyperText Markup Language) è utilizzato per strutturare il contenuto delle pagine web, mentre CSS (Cascading Style Sheets) consente di gestire l'aspetto visivo. JavaScript, d'altra parte, è un linguaggio di programmazione client-side essenziale per l'interattività delle applicazioni web. Durante il tirocinio, ho implementato questi linguaggi per sviluppare interfacce utente reattive e ottimizzare l'esperienza dell'utente.

Framework JavaScript: Ho utilizzato **React**, un popolare framework JavaScript sviluppato da Facebook, per costruire componenti dell'interfaccia utente. React consente un approccio basato sui componenti, favorendo la riusabilità e la manutenibilità del codice. Questa tecnologia ha reso più efficiente il processo di sviluppo, consentendo di gestire in modo efficace lo stato delle applicazioni.

2.2 Backend e database

Node.js: Per il lato server, ho utilizzato Node.js, un ambiente di esecuzione JavaScript. Node.js permette di costruire applicazioni web scalabili e performanti, grazie alla sua architettura non bloccante e al modello di I/O asincrono. Durante il tirocinio, ho implementato API RESTful utilizzando Node.js, facilitando la comunicazione tra il frontend e il backend.

MongoDB: Come sistema di gestione di database, ho impiegato MongoDB, un database NoSQL altamente scalabile. La struttura dei dati basata su documenti di MongoDB consente una grande flessibilità e facilita l'inserimento di dati non strutturati. Durante il progetto, ho gestito operazioni CRUD (Create, Read, Update, Delete) per la manipolazione dei dati degli utenti e delle applicazioni.

2.3 Strumenti di sviluppo e versionamento

Git: Ho utilizzato Git come sistema di controllo della versione. Git è essenziale per la gestione delle modifiche al codice sorgente, consentendo la collaborazione tra più sviluppatori e garantendo la tracciabilità delle versioni. Grazie a Git, è stato possibile gestire in modo efficiente il versionamento del codice e monitorare lo stato di avanzamento dei progetti, specialmente nelle fasi di sviluppo collaborativo.

GitHub: Ho utilizzato GitHub per ospitare il codice e agevolare la revisione del lavoro. GitHub ha fornito un ambiente strutturato per la gestione del repository, con strumenti di pull request, issue tracking e workflow automatizzati. La piattaforma ha facilitato la comunicazione con il team e permesso una maggiore trasparenza nell'implementazione delle funzionalità, agevolando anche l'integrazione continua e il controllo delle modifiche apportate da ciascun membro del team.

Ambienti di sviluppo: Ho lavorato con editor di codice come WebStorm, un ambiente di sviluppo integrato (IDE) molto diffuso tra gli sviluppatori web. Questo strumento offre numerosi plugin e funzionalità che hanno semplificato il processo di sviluppo, come il supporto per linting, formattazione automatica e debug, permettendo di ottimizzare il flusso di lavoro e mantenere la qualità del codice.

Yeoman: Come strumento di scaffolding, Yeoman è stato utile per generare rapidamente strutture di progetto standardizzate. Questo ha permesso di risparmiare tempo e mantenere una coerenza nel setup dei nuovi moduli o componenti dell'applicazione, migliorando la struttura e l'organizzazione del codice.

4.3. Sfide affrontate e soluzioni adottate

Durante il tirocinio, ho affrontato diverse sfide che hanno messo alla prova le mie capacità tecniche e relazionali. Alcune delle principali sfide includevano:

- **Adattamento a nuovi linguaggi e tecnologie:** Durante il tirocinio, ho dovuto affrontare la curva di apprendimento associata all'uso di nuovi linguaggi di programmazione e framework, come TypeScript e Next.js. Questo ha richiesto un impegno significativo nell'apprendimento delle sintassi, delle best practices e delle peculiarità di ciascuno. Ho dedicato tempo allo studio di documentazione e tutorial online, nonché alla pratica diretta attraverso progetti assegnati, il che ha arricchito notevolmente il mio bagaglio tecnico.

- **Integrazione delle API:** L'integrazione di API esterne ha presentato delle difficoltà, specialmente in termini di gestione degli errori e della documentazione. Per risolvere questi problemi, ho dedicato tempo alla comprensione delle specifiche delle API e ho collaborato con i membri più esperti del team per ottenere feedback e suggerimenti.
- **Debugging e ottimizzazione:** Durante lo sviluppo, sono emersi bug imprevisti e problemi di prestazioni. Ho imparato a utilizzare strumenti di debugging e profili di prestazioni, lavorando in modo sistematico per identificare e risolvere i problemi.
- **Navigazione delle revisioni del codice:** Partecipare alle revisioni del codice è stata una sfida che mi ha aiutato a capire l'importanza di scrivere codice di alta qualità. Ho imparato a ricevere critiche costruttive e a dare feedback utili agli altri. Ho anche appreso come le revisioni del codice possano portare a soluzioni più robuste e meglio progettate.
- **Collaborazione in un ambiente remoto:** Lavorare in un team distribuito ha presentato sfide legate alla comunicazione e alla coordinazione. Ho dovuto imparare a utilizzare strumenti di collaborazione online come Slack e GitHub per rimanere aggiornato sui progressi del progetto e per gestire le mie attività.

Queste esperienze mi hanno insegnato l'importanza di affrontare le sfide in modo proattivo e di collaborare con il team per trovare soluzioni efficaci.

5. Esplorazione Dettagliata dei Progetti

Nella sezione successiva verrà spiegato il processo di sviluppo delle richieste/task. Il codice relativo alle varie sezioni può essere consultato tramite il link al repository sottostante, dove le strutture delle cartelle seguono fedelmente quello del routing del progetto. In particolare, sarà presente solo ed esclusivamente il codice da me sviluppato.

Link Repository:

<https://github.com/AlexBruno1/Tirocinio-Getapper-s.r.l..git>

5.1. BMT - Business Management Tool

5.1.1 Richiesta 1: Supporto ai Promemoria

La prima richiesta assegnata nel progetto **Business Management Tool (BMT)** si è concentrata sulla creazione di un sistema di promemoria per la gestione delle attività degli utenti. L'obiettivo principale era sviluppare una funzionalità che consentisse di aggiungere promemoria personalizzati, ciascuno dei quali poteva essere associato a un responsabile specifico, identificato tramite un `userId`. Questa funzionalità era essenziale per garantire che le scadenze e i compiti importanti venissero monitorati e gestiti in modo efficace, aumentando la produttività e l'organizzazione del team.

Funzionalità Principali

Struttura del Promemoria

Ogni promemoria includeva un'azione definita tramite un **enum**, permettendo così una categorizzazione chiara e facilmente espandibile per supportare nuove operazioni in futuro. Questo approccio ha garantito che il sistema potesse evolversi senza difficoltà, rispondendo alle esigenze mutevoli degli utenti. Le azioni sono state progettate per avere parametri variabili, consentendo una maggiore flessibilità nella configurazione del promemoria.

Inoltre, ogni promemoria necessitava di una scadenza e di un intervallo temporale per l'invio di email di promemoria, garantendo così che gli utenti fossero costantemente aggiornati sulle scadenze imminenti. Questa funzionalità ha migliorato l'affidabilità del sistema, assicurando che gli utenti ricevessero notifiche tempestive per le loro attività.

Esempi di Promemoria

Un esempio di azione era il **"Sollecito pagamento fattura non incassata"**, in cui il promemoria sarebbe stato inviato ogni due settimane fino al raggiungimento della scadenza. Questa funzionalità ha permesso agli utenti di tenere sotto controllo i pagamenti in sospeso, contribuendo a una gestione finanziaria più efficace.

Un altro esempio includeva il **"Rinnovo certificato SSL"**, in cui l'email di promemoria sarebbe stata inviata solo alla scadenza specificata. Questo ha dimostrato l'importanza di ricordare agli utenti scadenze critiche per la sicurezza dei loro progetti, evitando potenziali interruzioni nei servizi.

Implementazione

Il processo di implementazione è stato suddiviso in tre passaggi principali:

1. **Creazione del Modello Backend:** È stato creato un modello **Activity** che teneva traccia delle attività da svolgere. Questo modello includeva campi cruciali come **userIds**, **title**, **reminderStrategy**, e **params**, fornendo così una struttura solida per gestire i promemoria. La scelta di questi campi è stata fondamentale per assicurare che ogni attività potesse essere descritta in modo dettagliato e gestita in modo efficace.
2. **Sviluppo dell'API:** È stata generata un'API **POST /activities** utilizzando **GeNYG** per consentire l'aggiunta di nuove attività. Questa API accettava tutti i campi del modello come payload, tranne il campo **status**, che veniva impostato automaticamente sul backend. Questo approccio ha semplificato l'interazione con il database, garantendo che le attività fossero correttamente registrate senza complicazioni.
3. **Creazione del Form di Inserimento:** È stato realizzato un dialogo/form sul frontend per l'inserimento delle nuove attività. I campi del form corrispondevano ai dati richiesti dall'API, inclusi i selettori per il responsabile, la descrizione del task e le strategie di promemoria. Questa interfaccia utente è stata progettata per essere intuitiva e facile da utilizzare, consentendo agli utenti di inserire rapidamente e con precisione le informazioni necessarie.

Conclusioni

L'implementazione del sistema di promemoria nel Business Management Tool (BMT) ha rappresentato un passo fondamentale verso una gestione più curata delle attività aziendali, soprattutto grazie alla categorizzazione flessibile e alla possibilità di configurare promemoria personalizzati per ogni scadenza e operazione.

La struttura modulare, con azioni configurabili tramite enum e una gestione scalabile delle strategie di notifica, ha garantito al progetto una notevole capacità di adattamento a nuove esigenze. Inoltre, l'integrazione fluida tra il backend e l'interfaccia utente ha reso l'esperienza dell'utente intuitiva, riducendo il tempo necessario per configurare e monitorare i promemoria.

In sintesi, questa funzionalità contribuirà a migliorare la puntualità del team, aumentando l'efficienza dei processi interni.

5.1.2 Richiesta 2: Form nuovo promemoria

La seconda richiesta assegnata nel progetto **Business Management Tool (BMT)** si è concentrata sulla creazione di un modulo per l'inserimento di nuovi promemoria, denominato

ActivityForm. Questo modulo è stato progettato per fornire agli utenti un'interfaccia intuitiva e completa per la selezione e l'impostazione di tutti gli elementi necessari a configurare un promemoria in modo efficiente, migliorando così l'esperienza utente e ottimizzando il processo di gestione delle attività.

Funzionalità Principali

Struttura del Form

Il modulo consente di specificare vari aspetti del promemoria, tra cui:

- **userIds:** un array di **ObjectId** per selezionare uno o più utenti responsabili, garantendo così che le attività siano assegnate in modo chiaro e tracciabile.
- **title:** un campo di testo per la descrizione del task, permettendo agli utenti di fornire informazioni dettagliate e specifiche sulle attività da svolgere.
- **reminderStrategy:** un selettore per definire la strategia di promemoria, consentendo agli utenti di scegliere come e quando ricevere le notifiche relative al promemoria.
- **reminderTime:** un campo di testo per specificare il tempo del promemoria, fornendo maggiore flessibilità nella pianificazione delle notifiche.
- **deadline:** un selettore di data per impostare una scadenza, assicurando che gli utenti possano fissare chiaramente le date entro cui completare le attività.

Selettori e Campi di Input

Il modulo include diversi tipi di input per facilitare la selezione degli elementi, rendendo il processo di configurazione dei promemoria intuitivo e accessibile:

- **Selettore Utenti:** un selettore multiuso (FormMultiUserSelector) consente la scelta di uno o più utenti, facilitando la collaborazione e l'assegnazione dei compiti.
- **Campo di Testo:** utilizzato per inserire la descrizione del task (FormTextField), fornendo un modo semplice e diretto per comunicare i dettagli delle attività.
- **Selettore Strategia:** un campo a discesa per selezionare la strategia di promemoria (FormSelect), offrendo agli utenti diverse opzioni per personalizzare le loro notifiche.
- **Campo di Testo per il Tempo:** consente di specificare il tempo per il promemoria (FormTextField), dando libertà di scelta su come gestire le scadenze.
- **Selettore di Data e Ora:** un picker per impostare la scadenza (FormDateTimePicker), rendendo la selezione della data e dell'ora semplice e precisa.

In aggiunta, per gestire gli altri array, sono stati implementati selettori multiuso per lavoratori, progetti, commesse e clienti. Questi sono stati sviluppati utilizzando componenti esistenti o nuovi, come **FormMultiWorkerSelector**, **FormMultiJobSelector**, **FormMultiProjectSelector**, e **FormMultiCustomerSelector**, rendendo il modulo altamente versatile e adattabile alle diverse esigenze degli utenti.

Implementazione

Il processo di implementazione del **ActivityForm** ha seguito questi passaggi:

1. **Generazione del Form:** Utilizzando il generatore *yo g-next*, è stato creato un nuovo modulo **ActivityForm**, strutturato in modo da includere tutti i campi e i selettori necessari per la configurazione di un promemoria completo. Questa fase ha richiesto una pianificazione attenta per garantire che il modulo fosse intuitivo e facile da usare.
2. **Integrazione dei Selettori:** Sono stati integrati e configurati i selettori per garantire che gli utenti potessero facilmente scegliere i responsabili, i progetti, e le commesse correlati al promemoria. Questa integrazione ha semplificato il processo di creazione, permettendo agli utenti di collegare facilmente le attività a risorse e collaboratori pertinenti.
3. **Validazione e Gestione degli Errori:** Durante lo sviluppo, è stata posta particolare attenzione alla validazione dei dati inseriti nel modulo. Sono stati implementati meccanismi di controllo per assicurare che gli utenti forniscano informazioni complete e corrette prima di inviare la richiesta per creare un nuovo promemoria. Questo approccio ha contribuito a migliorare l'affidabilità del sistema, riducendo il rischio di errori e assicurando che tutte le informazioni necessarie fossero raccolte in modo efficace.

Conclusioni

L'implementazione del modulo ActivityForm ha apportato notevoli miglioramenti nella gestione delle attività all'interno del Business Management Tool (BMT). Gli utenti sono in grado di configurare promemoria dettagliati e personalizzati, ottimizzando così il monitoraggio delle scadenze e la distribuzione delle responsabilità.

L'integrazione di selettori versatili e campi specifici ha reso il modulo altamente adattabile alle esigenze del team, facilitando la selezione di collaboratori, progetti e risorse pertinenti in modo chiaro e strutturato. Inoltre, la validazione dei dati ha aumentato l'affidabilità, contribuendo a ridurre gli errori e a migliorare l'esperienza utente.

In sintesi, l'ActivityForm è un modulo che si configura come una risorsa per sostenere la produttività degli utenti e del team e favorire una collaborazione strutturata e responsabile.

5.3. Sale Assistant

5.2.1 Task 1: Creazione Form per Generazione Istruzioni da Link

Nel primo task, ho collaborato alla realizzazione di un form dedicato alla generazione di istruzioni a partire da un link. Questo form offriva agli utenti un mezzo rapido per ottenere informazioni specifiche basate sul contenuto del link, migliorando l'efficienza nell'interazione con l'assistente.

Funzionalità Principali

- **Input del Link:** Il form è stato progettato per accettare un URL come input. Gli utenti potevano semplicemente incollare il link e premere un pulsante per generare le istruzioni corrispondenti.

Implementazione

L'implementazione è stata suddivisa in tre fasi:

1. Progettazione del Form

- **Interfaccia con Material-UI:** Per la creazione del form, ho utilizzato Material-UI, che ha permesso di strutturare un'interfaccia moderna e coerente con il resto dell'applicazione. Il form includeva un campo di input per l'URL e un pulsante per inviare la richiesta. Ho utilizzato i componenti **TextField** per il campo di input e **Button** per l'invio, garantendo una UI intuitiva e reattiva.

2. Sviluppo dell'API di Elaborazione

- **Endpoint per la Generazione delle Istruzioni:** Ho sviluppato un endpoint API **POST /generate-instructions** per gestire la logica di elaborazione. L'API riceveva il link, estraeva informazioni chiave e restituiva una risposta strutturata contenente le istruzioni generate. Per garantire l'efficienza, l'API utilizzava una logica di parsing e analisi ottimizzata per interpretare il contenuto del link in modo rapido.

3. Integrazione del Frontend con il Backend

- **Chiamata API con Redux:** Ho utilizzato Redux per gestire lo stato del form. Questo ha permesso di mantenere separati i dati e la logica di stato della richiesta, migliorando la leggibilità del codice e la coerenza nell'applicazione.

Conclusioni

Questa funzionalità ha migliorato l'efficienza dell'assistente, consentendo agli utenti di ottenere informazioni rapide tramite un semplice link. L'uso di Material-UI ha permesso di sviluppare un'interfaccia coerente e facilmente navigabile, mentre Redux ha facilitato la gestione delle chiamate API e l'aggiornamento dello stato globale.

5.2.2 task 2: Aggiunta Chip di Stato nella OrganizationList

In questo task, mi sono concentrato sull'integrazione di un chip di stato nella lista delle organizzazioni per offrire agli utenti un elemento visivo chiaro e immediato sullo stato di ciascuna organizzazione. Questo componente ha migliorato l'usabilità dell'interfaccia, permettendo agli utenti di comprendere rapidamente lo stato operativo delle organizzazioni.

Funzionalità Principali

- **Chip di Stato:** Ogni organizzazione nella lista ora include un chip che rappresenta visivamente il suo stato, come "Attivo," "In lavorazione" o "Errore." I colori e le etichette dei chip sono stati scelti per garantire una comprensione rapida e intuitiva dello stato corrente di ciascuna organizzazione, migliorando la leggibilità e la chiarezza dell'interfaccia.
- **Aggiornamento Dinamico:** I chip di stato si aggiornano in tempo reale in risposta a modifiche dello stato delle organizzazioni. Ciò consente agli utenti di vedere informazioni sempre attuali senza la necessità di ricaricare la pagina.

Implementazione

1. Modifica del Modello di Dati

- **Logica di Stato:** La logica del backend è stata arricchita con controlli per aggiornare il campo **status** in base alle attività correnti e alle interazioni con l'organizzazione, garantendo così la coerenza dei dati.

2. Aggiornamento dell'Interfaccia con Material-UI

- **Utilizzo dei Chip di Material-UI:** Per i chip di stato, ho utilizzato il componente **Chip** di Material-UI, che permette una personalizzazione completa dei colori e dello stile. Il colore del chip cambiava a seconda del valore del campo **status**, utilizzando uno schema cromatico intuitivo: ad esempio, verde per "Attivo," grigio per "In lavorazione," e rosso per "Errore."
- **Integrazione nella OrganizationList:** I chip sono stati integrati direttamente nella **OrganizationList**, posizionati accanto al nome di ciascuna organizzazione. Questo ha permesso di mantenere l'interfaccia pulita e organizzata, mantenendo comunque visibili le informazioni più importanti.

3. Gestione degli Aggiornamenti con Redux

- **Redux per la Gestione dello Stato:** Redux è stato utilizzato per memorizzare e aggiornare lo stato di ciascuna organizzazione a livello globale, facilitando il flusso di dati tra il backend e il frontend. Le modifiche di stato, come un cambio da "In lavorazione" a "Attivo," venivano tracciate tramite azioni Redux che aggiornavano il valore **status** per l'organizzazione specifica.

Conclusioni

L'aggiunta dei chip di stato ha reso l'interfaccia più informativa e intuitiva, migliorando l'esperienza utente. Grazie all'integrazione di Material-UI e Redux, l'interfaccia risponde in modo fluido alle variazioni di stato, mentre l'uso del WebSocket assicura che gli utenti dispongano sempre di informazioni aggiornate.

5.2.3 task 3: Logout Lato Admin

Il terzo task consisteva nell'implementazione della funzionalità di logout per gli utenti con privilegi di amministratore. Questa funzionalità era fondamentale per garantire la sicurezza e la privacy degli amministratori, permettendo loro di disconnettersi in modo sicuro e proteggere le informazioni sensibili della piattaforma.

Funzionalità Principali

- **Logout Sicuro:** La funzionalità di logout garantiva che le sessioni amministrative venissero invalidate, prevenendo accessi non autorizzati ai dati sensibili una volta che l'utente si disconnetteva. L'invalidazione della sessione ha permesso di eliminare eventuali token o cookie di autenticazione residui, proteggendo il sistema da potenziali rischi di sicurezza.
- **Feedback Visivo:** Dopo il logout, agli utenti veniva fornito un feedback visivo, un breve alert, che confermava il completamento della procedura. Questo elemento di

conferma era importante per garantire che gli amministratori percepissero chiaramente che la loro sessione era stata chiusa con successo.

Implementazione

L'implementazione è stata strutturata in tre fasi principali:

1. Sviluppo della Logica di Logout nel Backend

- **Endpoint di Logout:** Ho implementato un endpoint API nel backend per la terminazione della sessione. Questo endpoint gestiva la logica di disconnessione, eliminando la sessione admin.

2. Interfaccia Utente

- **Pulsante di Logout:** Nella dashboard dell'amministratore, ho aggiunto un pulsante di logout facilmente visibile, utilizzando i componenti di Material-UI per uno stile coerente con il resto dell'interfaccia. Il pulsante era accessibile dalla barra di navigazione superiore per permettere un accesso rapido e intuitivo alla funzione di logout.
- **Trigger della Funzione:** Al clic sul pulsante, veniva invocata una funzione, la quale richiama sia la API che Redux, e che gestiva il flusso di logout, aggiornando lo stato dell'applicazione per riflettere la disconnessione dell'utente.

3. Feedback Utente

- **Aggiornamento dello Stato con Redux:** Ho utilizzato Redux per aggiornare lo stato globale dell'applicazione, assicurandomi che i componenti sensibili all'autenticazione, come la dashboard e altre aree riservate, fossero correttamente reindirizzati alla pagina di login subito dopo il logout.

Conclusioni

Questa funzionalità ha migliorato la sicurezza della piattaforma, garantendo agli amministratori la possibilità di terminare la loro sessione in modo sicuro. L'uso di Material-UI ha consentito di rendere il feedback visivo semplice, mentre Redux ha facilitato la gestione dello stato di autenticazione e del flusso di logout, garantendo una transizione fluida tra la disconnessione e la pagina di login.

5.2.4 Task 4: Creazione Dialog per Assistente Attivo

il quarto task riguardava lo sviluppo di un componente di dialog, utilizzato per fornire agli utenti informazioni aggiuntive sull'assistente attivo e permettere loro di contattare i gestori in caso di necessità. Questa funzione puntava a migliorare l'interazione e la trasparenza con gli utenti, senza intraprendere una conversazione diretta, ma offrendo opzioni di contatto immediate. In particolare, durante la creazione di un nuovo assistente, nel caso in cui esso fosse figurato come "in lavorazione", invece che "attivo", il dialog si sarebbe aperto in automatico per fornire un aiuto tempestivo.

Funzionalità Principali

Il dialog presentava uno spazio informativo con dettagli sullo stato dell'assistente, rendendo

visibile, ad esempio, se l'assistente era attivo, in pausa, o disponibile solo per consultazioni limitate. Inoltre, il dialog includeva due opzioni di contatto, mail e telefono, che rispettivamente, tramite due pulsanti, permetteva all'utente di richiedere assistenza o ulteriori informazioni da parte dei gestori, rendendo più accessibile il supporto.

Implementazione

1. Frontend

- **Material-UI per il Dialog:** Il componente dialog è stato costruito utilizzando Material-UI (MUI), una libreria che fornisce componenti React con uno stile professionale e facilmente integrabile. Material-UI è stata scelta per la sua flessibilità e capacità di personalizzazione.
- **Gestione Automatica dell'Apertura:** L'apertura automatica del dialog è stata implementata grazie a una combinazione di Redux e Material-UI. Il componente monitorava il valore dello stato dell'assistente. Quando lo stato era impostato su "in lavorazione", il dialog si apriva senza l'intervento dell'utente.
- **Personalizzazione dell'Interfaccia:** All'interno del dialog erano presenti:
 - Un messaggio informativo sullo stato corrente dell'assistente, con una descrizione delle eventuali limitazioni.
 - Due pulsanti per contattare i gestori.
 - Il layout era progettato con componenti tipografici e di tooltip con icons di Material-UI per mantenere l'interfaccia intuitiva e coerente con lo stile dell'applicazione.

2. Gestione dello Stato con Redux

- **Stato dell'Assistente:** Redux è stato utilizzato per gestire lo stato dell'assistente a livello globale, consentendo al dialog di rispondere dinamicamente alle modifiche.

Conclusioni

Questo componente di dialog ha arricchito l'esperienza dell'utente, offrendo un accesso rapido e ben organizzato a informazioni importanti, insieme alla possibilità di contattare i gestori. L'implementazione ha sfruttato la flessibilità di Material-UI e una struttura di comunicazione backend efficiente per mantenere la trasparenza e l'interattività con l'utente.

5.2.5 Task 5: Generazione in Tempo Reale delle Risposte di OpenAI

L'ultimo task ha riguardato l'integrazione della generazione di risposte in tempo reale tramite le API di OpenAI, migliorando significativamente l'esperienza di interazione per gli utenti dell'assistente. Questa funzionalità ha permesso di rispondere in modo dinamico, generando risposte più dettagliate e contestualizzate a partire dalle domande degli utenti.

Funzionalità Principali

- **Risposte Dinamiche:** Attraverso l'integrazione delle API di OpenAI, l'assistente è stato configurato per elaborare le domande degli utenti in tempo reale, generando risposte dinamiche e pertinenti.
- **Visualizzazione Graduale delle Risposte:** Implementando un effetto di visualizzazione progressiva, le risposte venivano mostrate parola per parola, migliorando il realismo e il coinvolgimento durante la conversazione.

Implementazione

L'implementazione di questa funzionalità è stata suddivisa nelle seguenti fasi:

1. **Configurazione delle API di OpenAI**
 - **Integrazione delle Chiavi API:** Ho configurato le API di OpenAI nel backend, includendo la registrazione dell'account per ottenere le chiavi API e le configurazioni necessarie per le chiamate. La chiave API è stata gestita in modo sicuro tramite variabili di ambiente per garantire la sicurezza delle credenziali.
2. **Logica di Streaming per la Visualizzazione Graduale**
 - **Gestione della Risposta Graduale:** Per ottenere un'esperienza di streaming della risposta, ho utilizzato una logica di visualizzazione progressiva che permetteva di mostrare le risposte di OpenAI parola per parola. Questo approccio è stato implementato in React attraverso l'utilizzo di Redux e un sistema di gestione dello stato che aggiornava il contenuto in tempo reale.
 - **Implementazione Redux:** Redux è stato usato per centralizzare lo stato della risposta, con un'azione dedicata per aggiornare gradualmente il testo mostrato all'utente. La visualizzazione dinamica ha migliorato l'interattività, conferendo una sensazione di conversazione in tempo reale.
3. **Test dell'Interfaccia Utente**
 - **Verifica della Visualizzazione Graduale:** Ho condotto una serie di test per assicurarmi che la risposta si visualizzasse correttamente in tempo reale, verificando che ogni parola venisse visualizzata progressivamente senza interruzioni.

Conclusioni

Questa integrazione ha notevolmente migliorato l'esperienza d'uso dell'assistente, permettendo agli utenti di ricevere risposte dettagliate e immediate. L'uso di Redux per gestire lo stato delle risposte e di Material-UI per il rendering ha garantito un'interfaccia coerente e una risposta rapida ai comandi degli utenti, creando un'interazione coinvolgente e naturale.

6. Competenze acquisite

6.1. Competenze tecniche

Il tirocinio ha portato a un significativo sviluppo delle competenze tecniche in vari ambiti:

- **Sviluppo web avanzato:** Ho acquisito una padronanza consolidata dei linguaggi di programmazione front-end (JavaScript, TypeScript) e back-end (Node.js), migliorando notevolmente le mie abilità nella scrittura di codice pulito e mantenibile, con particolare attenzione all'uso di componenti riutilizzabili e all'ottimizzazione delle performance nelle applicazioni React.
- **Framework e librerie:** L'esperienza con React, Redux, Next.js e altre librerie moderne ha ampliato la mia comprensione delle architetture basate su JavaScript. Ho acquisito una conoscenza approfondita dei principi della programmazione asincrona, che mi ha permesso di sviluppare applicazioni reattive e di migliorare l'esperienza utente.
- **Implementazione del modello CRUD:** Durante il tirocinio, ho sviluppato una solida conoscenza delle operazioni CRUD (Create, Read, Update, Delete), essenziali per la gestione e manipolazione dei dati all'interno di applicazioni web. La capacità di implementare CRUD mi ha permesso di creare flussi di dati interattivi e di costruire componenti modulari per gestire in modo efficace le esigenze dinamiche degli utenti.
- **Architettura MVC (Model-View-Controller):** Ho acquisito competenze nell'implementazione dell'architettura MVC, separando logicamente la gestione dei dati (Model), la presentazione dell'interfaccia utente (View) e il controllo del flusso applicativo (Controller). Questa struttura mi ha permesso di organizzare il codice in modo più efficiente, facilitando il riutilizzo dei componenti e rendendo il progetto più scalabile e manutenibile.
- **Gestione dello stato e Redux Toolkit:** L'utilizzo di Redux Toolkit mi ha permesso di gestire in modo efficace lo stato globale dell'applicazione, migliorando la coerenza dei dati e semplificando il debug e la manutenzione del codice.
- **Database NoSQL (MongoDB):** Lavorare con MongoDB mi ha permesso di approfondire le differenze tra database relazionali e NoSQL e di apprezzare la flessibilità e scalabilità di quest'ultimo per progetti con requisiti di dati variabili. Ho imparato a progettare schemi di dati efficienti per rispondere a esigenze di rapidità nell'accesso e manipolazione.
- **Integrazione di API e AI (OpenAI):** L'implementazione di un sistema di generazione di risposte in tempo reale tramite API di OpenAI ha affinato le mie competenze in ambito di intelligenza artificiale e automazione, permettendomi di comprendere come configurare e ottimizzare richieste verso sistemi esterni per ridurre latenza e migliorare la precisione delle risposte.
- **Sicurezza e autenticazione:** Ho appreso tecniche per migliorare la sicurezza dell'applicazione, implementando autenticazione e autorizzazione a livello dell'API e migliorando la gestione delle sessioni per proteggere l'accesso ai dati sensibili.

6.2. Competenze trasversali

In aggiunta alle competenze tecniche, il tirocinio ha anche contribuito allo sviluppo di competenze trasversali fondamentali:

- **Lavoro di squadra:** Ho appreso l'importanza della comunicazione e della collaborazione all'interno di un team multidisciplinare. La condivisione delle idee e il feedback reciproco sono stati elementi chiave per il successo dei progetti, permettendomi di migliorare la qualità delle soluzioni proposte.
- **Problem solving:** Ho migliorato la mia capacità di affrontare e risolvere problemi in modo critico e creativo, sia in situazioni di stress che in contesti di collaborazione. L'approccio analitico sviluppato mi ha aiutato a trovare soluzioni innovative e pratiche per superare le sfide tecniche.
- **Adattabilità e flessibilità:** Lavorare in un ambiente dinamico mi ha permesso di sviluppare una maggiore flessibilità, adattandomi rapidamente a nuove tecnologie, metodologie di lavoro e cambiamenti nei requisiti di progetto, mantenendo sempre una visione orientata agli obiettivi.
- **Comunicazione efficace:** Ho migliorato le mie capacità di comunicazione, sia verbale che scritta, adattandole al contesto tecnico e all'interazione con colleghi di diversa esperienza. Questo ha facilitato la condivisione di conoscenze e ha reso il lavoro più armonioso e produttivo.
- **Autonomia e responsabilità:** Durante il tirocinio, mi è stata data l'opportunità di lavorare autonomamente su diverse attività. Questa esperienza mi ha permesso di sviluppare un senso di responsabilità e di imparare a gestire il mio lavoro in modo indipendente, assicurando che ogni fase del progetto fosse completata con attenzione e precisione.
- **Apprendimento continuo:** Ho adottato un approccio orientato all'apprendimento continuo, utilizzando risorse come documentazione tecnica, corsi online e il feedback del team per migliorare costantemente le mie competenze, restando aggiornato sulle ultime tecnologie e metodologie.

7. Riflessioni personali

7.1. Valutazione dell'esperienza

L'esperienza di tirocinio presso GetApper Fragments s.r.l. si è rivelata estremamente formativa e arricchente. Ho avuto l'opportunità di apprendere nuove tecnologie e di applicare le conoscenze teoriche acquisite durante il mio percorso di studi, confrontandomi con le sfide e le complessità del lavoro reale nel campo dello sviluppo web. La possibilità di collaborare con professionisti esperti mi ha fornito una prospettiva sulle best practices del settore e sui processi di sviluppo software.

In particolare, ho apprezzato il supporto ricevuto dai miei tutor. Questo ambiente collaborativo ha facilitato il mio apprendimento e ha reso il mio contributo ai progetti più significativo.

7.2. Confronto tra aspettative e realtà

Le aspettative iniziali riguardavano principalmente l'acquisizione di competenze tecniche e la comprensione del funzionamento di un team di lavoro. In effetti, ho avuto modo di approfondire entrambi gli aspetti, superando alcune delle sfide previste. Tuttavia, ho anche scoperto l'importanza di competenze trasversali, come la comunicazione e la gestione del tempo, che non avevo considerato in modo sufficientemente approfondito all'inizio del tirocinio.

In generale, l'esperienza ha superato le mie aspettative, offrendomi un'opportunità concreta di crescita professionale e personale. Ho imparato a valutare l'importanza della collaborazione e della flessibilità in un ambiente di lavoro dinamico.

7.3. Lezioni apprese

Tra le lezioni più significative apprese durante il tirocinio, meritano particolare menzione:

- **Importanza della formazione continua:** Nel settore tecnologico, caratterizzato da un'evoluzione rapida e costante, la capacità di apprendere e adattarsi rappresenta una competenza cruciale. Ho compreso come il successo nel lungo termine richieda un impegno costante nell'aggiornamento sulle nuove tecnologie e metodologie, e nel mio caso, ho avuto modo di approfondire le ultime innovazioni di JavaScript, strumenti di AI, e database NoSQL. Questa esperienza mi ha reso consapevole di quanto la formazione sia una risorsa continua, utile per migliorare le proprie capacità tecniche e rimanere competitivi.
- **Valore del feedback:** Il feedback, sia ricevuto che fornito, è fondamentale per il miglioramento continuo. Durante il tirocinio, ho imparato a cercare attivamente feedback sui miei progetti e ad accoglierlo come un'opportunità di crescita. Ho

appreso l'importanza di dare attenzione alle osservazioni dei colleghi, utilizzandole per migliorare le implementazioni e adattarmi meglio alle esigenze dell'azienda.

- **Gestione delle sfide legate all'inesperienza:** Durante il tirocinio, mi sono trovato a dover lavorare su tecnologie nuove per me e a rispondere a elevate aspettative. Ho appreso che l'inesperienza può essere gestita con successo attraverso un uso strategico della documentazione e ponendo domande mirate ai colleghi più esperti. Adottare queste tecniche si è dimostrato efficace nel mantenere un equilibrio tra la qualità del lavoro svolto e il rispetto delle richieste, trasformando l'inesperienza in una motivazione per migliorare.

8. Conclusioni

8.1. Sintesi dei punti chiave

Il tirocinio presso GetApper Fragments s.r.l. è stato un'esperienza trasformativa, sia sul piano personale che professionale, permettendomi di applicare le conoscenze teoriche acquisite durante gli studi e di sviluppare nuove competenze fondamentali per il mondo del lavoro.

Ho approfondito competenze tecniche specifiche nel campo dello sviluppo web, in particolare con tecnologie come Next.js, React, Redux, e MongoDB, migliorando le mie capacità di sviluppo sia sul lato front-end che back-end. L'opportunità di lavorare con un sistema di gestione complesso e di contribuire allo sviluppo di un sales assistant basato su AI mi ha permesso di esplorare l'applicazione pratica dell'intelligenza artificiale e di comprendere le potenzialità dei database NoSQL per la gestione di dati non strutturati.

Oltre alle competenze tecniche, ho consolidato abilità trasversali, tra cui il lavoro di squadra e la comunicazione efficace all'interno di un team multidisciplinare, dove il confronto costruttivo e il feedback reciproco sono stati essenziali per il successo dei progetti. La gestione delle specifiche e la capacità di adattamento a nuove tecnologie e metodologie hanno rappresentato sfide cruciali, che mi hanno spinto a trovare soluzioni pratiche, come l'uso metodico della documentazione e il confronto continuo con i colleghi.

Infine, l'esperienza ha confermato l'importanza dell'integrazione tra teoria e pratica: mentre la teoria fornisce le basi necessarie, la pratica quotidiana insegna ad affrontare le complessità reali del lavoro su progetti di valore concreto. Inoltre, ho compreso il valore di un apprendimento continuo in un settore come quello tecnologico, dove il progresso rapido e l'innovazione costante rendono necessario aggiornarsi e affinare continuamente le proprie competenze.

In sintesi, il tirocinio mi ha fornito un quadro completo delle dinamiche lavorative di un'azienda tecnologica moderna, preparandomi per le sfide future e fornendomi una base solida su cui costruire il mio percorso professionale.

9. Bibliografia

9.1. Riferimenti, materiali e fonti consultate

HTML

- W3Schools. (n.d.). HTML Tutorial. Consultato da <https://www.w3schools.com/html/>
- Mozilla. (n.d.). MDN Web Docs: HTML. Consultato da <https://developer.mozilla.org/en-US/docs/Web/HTML>

CSS

- W3Schools. (n.d.). CSS Tutorial. Consultato da <https://www.w3schools.com/css/>
- Mozilla. (n.d.). MDN Web Docs: CSS. Consultato da <https://developer.mozilla.org/en-US/docs/Web/CSS>

JavaScript

- W3Schools. (n.d.). JavaScript Tutorial. Consultato da <https://www.w3schools.com/js/>
- Mozilla. (n.d.). MDN Web Docs: JavaScript. Consultato da <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

TypeScript

- TypeScript. (n.d.). TypeScript - A typed superset of JavaScript that compiles to plain JavaScript. Consultato da <https://www.typescriptlang.org/>
- YouTube. (2021). Understanding TypeScript - 2023 Edition [Video]. Fireship. Consultato da https://www.youtube.com/watch?v=ahCwqrYpluM&ab_channel=Fireship

9.2 JavaScript e Frameworks

- **React**
React. (n.d.). React - A JavaScript library for building user interfaces. Consultato da <https://react.dev/>
- **Redux**
Redux. (n.d.). Redux - A Predictable State Container for JS Apps. Consultato da <https://redux.js.org/>
- **Material-UI**
Material-UI. (n.d.). Material-UI Documentation. Consultato da <https://mui.com/>
- **Yup**
Yup. (n.d.). Yup - A JavaScript schema builder for value parsing and validation. Consultato da <https://github.com/jquense/yup>
- **Moment.js**
Moment.js. (n.d.). Moment.js - A JavaScript date library for parsing, validating, manipulating, and formatting dates. Consultato da <https://momentjs.com/>
- **Axios**
Axios. (n.d.). Axios - Promise based HTTP client for the browser and Node.js. Consultato da <https://axios-http.com/>

9.3 Backend e Database

- **Node.js**
Node.js. (n.d.). Node.js - JavaScript runtime built on Chrome's V8 JavaScript engine. Consultato da <https://nodejs.org/>
- **MongoDB**
MongoDB. (n.d.). MongoDB - The Database for Modern Applications. Consultato da <https://www.mongodb.com/>

9.4 Gestione delle Dipendenze e Versionamento

- **npm**
npm. (n.d.). npm - A package manager for JavaScript. Consultato da <https://www.npmjs.com/>
- **Git**
Git. (n.d.). Git - The Simple Guide. Consultato da <https://git-scm.com/doc>

9.5 Apprendimento e Pratica

- **HackerRank**
HackerRank. (n.d.). HackerRank - Code Practice, Certification, and Interview Preparation. Consultato da <https://www.hackerrank.com/>
- **YouTube**
YouTube. (n.d.). React.js Crash Course [Video]. Traversy Media. Consultato da <https://www.youtube.com/watch?v=seXcB4x4ea8>

9.6 Risorse Aggiuntive

- **Wikipedia**
Wikipedia. (n.d.). List of HTTP status codes. Consultato da https://en.wikipedia.org/wiki/List_of_HTTP_status_codes
- **GetApper Academy**
GetApper Academy. (n.d.). Full Stack Web Development with Next.js, GitHub, CircleCI, Sentry, and AWS Amplify. Consultato da https://academy.getapper.com/p/fullstack-web-development-with-nextjs-github-circleci-sentry-and-aws-amplify?coupon_code=LUCKYBOY&product_id=4966974
- **GetApper**
GetApper. (n.d.). Generator G-Next [Source code]. GitHub. Consultato da <https://github.com/getapper/generator-g-next>

9.7 Documentazione Tecnica

- **Yeoman**
Yeoman. (n.d.). Yeoman - A Modern Workflow for Web Apps. Consultato da <http://yeoman.io/>
- **WebStorm**
WebStorm. (n.d.). WebStorm: The Smartest JavaScript IDE. Consultato da <https://www.jetbrains.com/webstorm/>

- **CircleCI**
CircleCI. (n.d.). CircleCI - Continuous Integration and Delivery. Consultato da <https://circleci.com/>
- **CodeRabbit**
CodeRabbit. (n.d.). CodeRabbit - AI-Powered Development Assistant. Consultato da <https://coderabbit.ai/>

9.8 Materiale fornito dall'azienda

- **JavaScript & TypeScript:** Presentazione - <https://docs.google.com/presentation/d/1GdWtnTXBZurmDtBCj8OIYxnBsOnJQKdddUtVCNT99ww/edit#slide=id.p>
- **ReactJS:** Presentazione - https://docs.google.com/presentation/d/1foY6sWeCDbQmmcg0U3g_uSfcB5thC879OXuc9vecxhA/edit#slide=id.gbb4df79cee_0_38
- **REST API Server Development:** Presentazione - https://docs.google.com/presentation/d/19a2yF37oO0Q94QMIQ8Lc4HEQzXFaxMgCGfDrvrPvJhA/edit#slide=id.g7171efbf99_0_76
- **Git:** Presentazione - https://docs.google.com/presentation/d/1a6KdrzGwnMGbgCqy302i2_xlKluyeovWsRJFBmgVsbY/edit#slide=id.g73064b1f49_0_183
- **Video corso Fullstack Web development con NextJs, NodeJS & ReactJS:** Corso completo - https://academy.getapper.com/p/fullstack-web-development-with-nextjs-github-circleci-sentry-and-aws-amplify?coupon_code=LUCKYBOY&product_id=4966974

9.9 Video di approfondimento

- JavaScript avanzato - <https://www.youtube.com/watch?v=GaN5VzqnRvM&t=12s>
- Introduzione a TypeScript - <https://www.youtube.com/watch?v=be2499W-HZE>
- Fullstack Web Development - <https://www.youtube.com/watch?v=ECeLB9kHuro&t=1126s>
- Gestione API REST con Node.js - <https://www.youtube.com/watch?v=lamRWwTRX0A&t=49s>
- Uso di Git in progetti collaborativi - <https://www.youtube.com/watch?v=IJPenRYCAj4&t=1012s>