
Riggio Simone



SWITCH360
SCUOLA-LAVORO

PCTO

12/05/2024

1. INTRODUZIONE

Questo progetto nasce dall'esigenza di creare una piattaforma unica sia per studenti che professori che agevoli la gestione dell'alternanza scuola lavoro sotto ogni punto di vista. Fornirà la visualizzazione completa delle varie aziende permettendo allo studente di scegliere le proprie preferite. Inoltre, lo studente potrà recensire le aziende in cui ha già svolto l'alternanza (scrivendo un commento e lasciando un voto, in modo tale che gli studenti futuri potranno decidere quale azienda "scegliere"). Nel sito vi sarà la possibilità di compilare il diario di bordo, scrivendo l'esperienza fatta durante la giornata.

Anche il responsabile (account amministratore) PCTO sarà in grado di visualizzare le aziende, tra cui quelle scelte dallo studente, e avrà la possibilità di assegnare lo studente ad una azienda. Quest'ultimo avrà anche l'accesso al diario di bordo degli studenti della sua classe, in modo tale da poter visualizzare l'esperienza fatta. Il responsabile potrà riassegnare uno studente (che non si è trovato bene in una azienda) ad un'altra azienda.

1.1 OBIETTIVI

1. Creare una piattaforma web intuitiva e user-friendly per studenti e professori.
2. Consentire agli studenti di visualizzare in modo completo le varie aziende disponibili per l'alternanza scuola-lavoro.
3. Implementare un sistema di recensioni delle aziende, permettendo agli studenti di condividere le proprie esperienze.
4. Fornire agli studenti la possibilità di compilare un diario di bordo giornaliero per registrare le loro esperienze durante l'alternanza.
5. Consentire al responsabile PCTO di visualizzare le aziende disponibili e quelle scelte dagli studenti.
6. Implementare la funzionalità per il responsabile PCTO di assegnare gli studenti alle aziende.
7. Consentire al responsabile PCTO di accedere ai diari di bordo degli studenti assegnati alla sua classe.

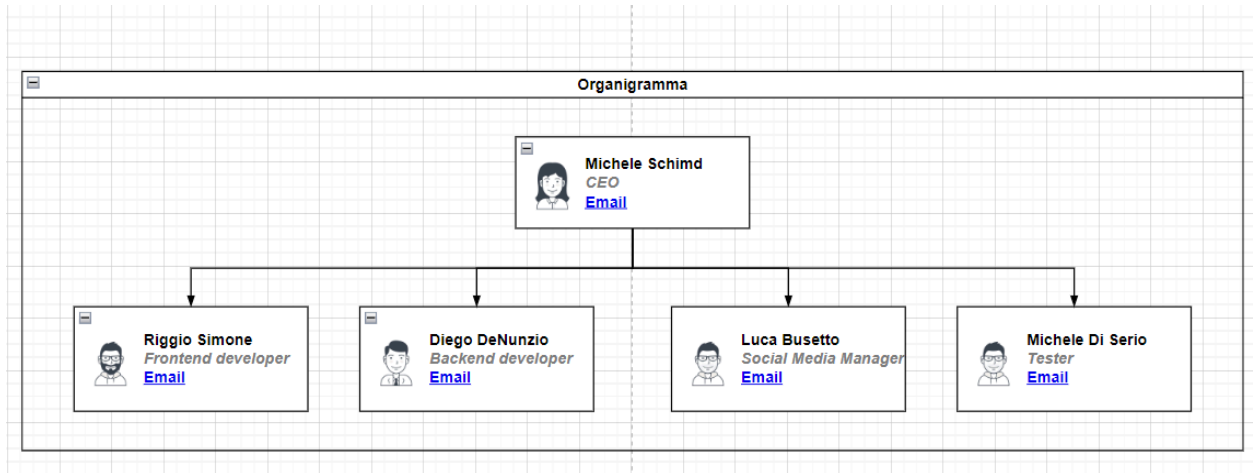
2 GLOSSARIO

1. **Alternanza Scuola-Lavoro:** Programma che prevede la combinazione di periodi di studio in aula e periodi di formazione pratica in azienda, al fine di favorire l'acquisizione di competenze professionali da parte degli studenti.
2. **Azienda:** Impresa o organizzazione che partecipa al programma di alternanza scuola-lavoro, offrendo opportunità di stage e formazione pratica agli studenti.
3. **Backend:** Parte del sistema software che gestisce la logica di funzionamento dell'applicazione, l'elaborazione dei dati e le comunicazioni con il database.
4. **Bug:** Errore o malfunzionamento in un programma software che causa risultati inaspettati o indesiderati.
5. **Compilare il Diario di Bordo:** Registrare le esperienze e le attività svolte durante la giornata di alternanza scuola-lavoro.
6. **Database:** Raccolta organizzata di dati che possono essere facilmente accessibili, gestiti e aggiornati. In questo progetto, il database supporta la piattaforma e contiene informazioni su aziende, studenti, recensioni e diari di bordo.
7. **Feedback:** Risposte e opinioni fornite dagli utenti sulla piattaforma, utilizzate per migliorare il sistema.
8. **Fat Free Framework (F3):** Framework PHP leggero utilizzato per lo sviluppo del backend della piattaforma, che facilita la gestione delle richieste HTTP, la manipolazione dei dati e la sicurezza.
9. **Front-end:** Parte del sistema software con cui gli utenti interagiscono direttamente, solitamente sviluppata utilizzando HTML, CSS e JavaScript.
10. **HTML (HyperText Markup Language):** Linguaggio di programmazione utilizzato per creare la struttura delle pagine web.
11. **PHP (Hypertext Preprocessor):** Linguaggio di scripting lato server utilizzato per lo sviluppo web, particolarmente adatto per creare pagine web dinamiche.
12. **Piattaforma:** Sistema software sviluppato per la gestione dell'alternanza scuola-lavoro, che include funzionalità per studenti, professori e responsabili PCTO.
13. **PCTO (Percorsi per le Competenze Trasversali e per l'Orientamento):** Programma che offre agli studenti esperienze di apprendimento pratico in contesti lavorativi reali.
14. **Project Manager:** Persona responsabile della pianificazione, esecuzione e chiusura del progetto, coordinando le attività e gestendo le risorse.
15. **Recensione:** Valutazione scritta fornita dagli studenti sulle aziende in cui hanno svolto l'alternanza scuola-lavoro, includendo commenti e voti.
16. **Responsabile PCTO:** Persona che gestisce e coordina le attività relative ai percorsi di alternanza scuola-lavoro all'interno di una scuola.

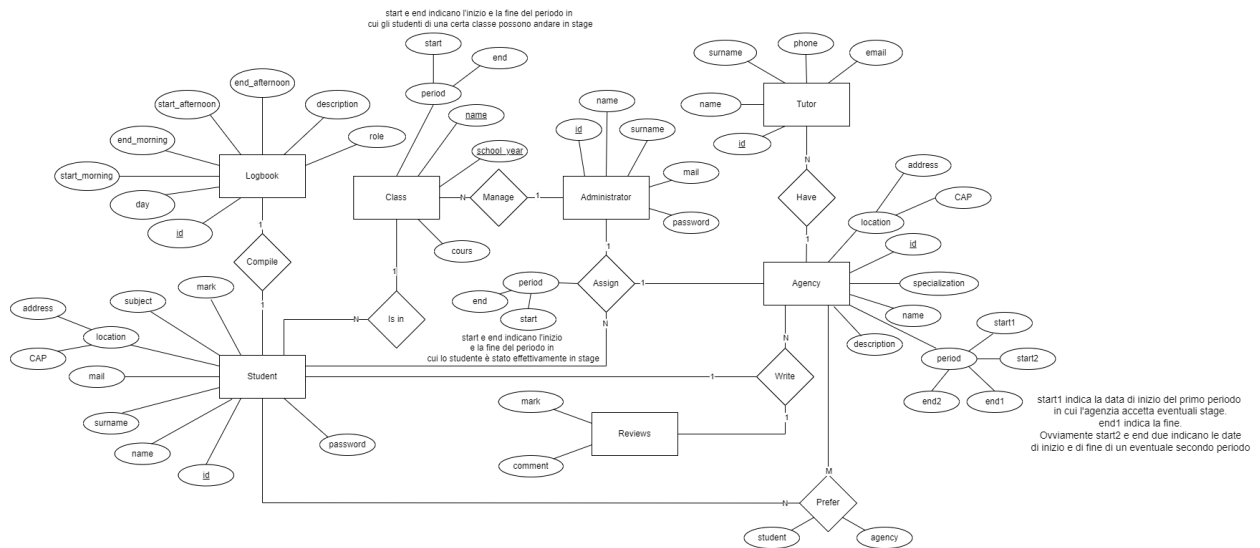
-
17. **Schema ER (Entity-Relationship):** Modello utilizzato per rappresentare le relazioni tra entità in un database.
 18. **Schema Logico:** Rappresentazione dettagliata della struttura del database, che include tabelle, campi e relazioni tra le tabelle.
 19. **Server:** Computer o sistema che fornisce risorse, dati, servizi o programmi a altri computer, noti come client, su una rete.
 20. **Sprint:** Ciclo di lavoro breve e ripetitivo utilizzato nello sviluppo agile, di solito della durata di 1-2 settimane, durante il quale viene sviluppato e testato un set specifico di funzionalità.
 21. **Stakeholder:** Persona o gruppo interessato al successo del progetto, inclusi utenti finali, finanziatori, sviluppatori e manager.
 22. **Tailwind CSS:** Framework CSS altamente personalizzabile utilizzato per creare interfacce utente moderne e responsive.
 23. **Test di Integrazione:** Test che verificano l'interazione tra diversi componenti del sistema per assicurarsi che funzionino correttamente insieme.
 24. **Test di Regression:** Test eseguiti per verificare che le modifiche al codice non abbiano introdotto nuovi errori o malfunzionamenti nei componenti esistenti.
 25. **Test di Unità:** Test che verificano il corretto funzionamento di singole componenti o unità di codice in isolamento.
 26. **Test Funzionali:** Test che verificano che il software esegua le funzioni previste e soddisfi i requisiti definiti dagli utenti finali.
 27. **Test e Debug:** Fase del progetto durante la quale la piattaforma viene sottoposta a test approfonditi per identificare e correggere errori o problemi.
 28. **Visual Studio Code:** Ambiente di sviluppo integrato (IDE) altamente configurabile, utilizzato per lo sviluppo del codice sorgente della piattaforma.
 29. **Vulnerabilità:** Debolezza nel sistema che può essere sfruttata per causare danni o accessi non autorizzati.
 30. **UX/UI (User Experience/User Interface):** Disciplina che si occupa della progettazione dell'interfaccia utente e dell'esperienza dell'utente, assicurando che l'uso della piattaforma sia intuitivo e soddisfacente.

3. ORGANIZZAZIONE PROGETTO

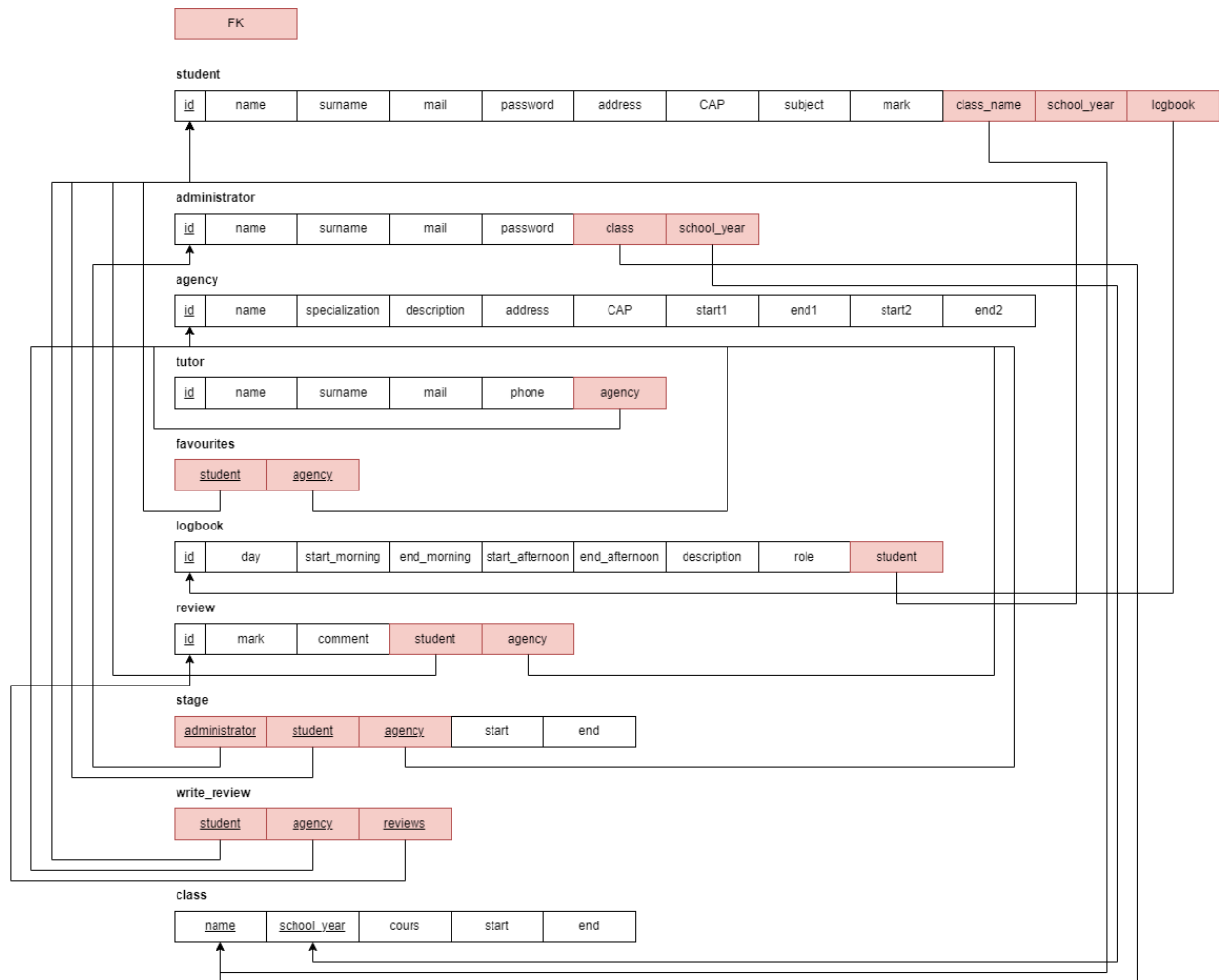
3.1 ORGANIGRAMMA



3.2 DIAGRAMMA ER



3.3 SCHEMA LOGICO



3.4 FASI DI LAVORO

Si è preferito dividere il progetto in fasi tra cui:

Analisi dei Requisiti

La fase di analisi dei requisiti coinvolge la raccolta e l'analisi approfondita delle esigenze richieste per la piattaforma. Questo include la definizione dei requisiti funzionali e non funzionali, la valutazione dei vincoli e la creazione di documentazione dettagliata.

Progettazione Database

Durante questa fase, verrà progettata la struttura del database che supporterà la piattaforma tra cui lo Schema ER e lo Schema Logico. Questo include la definizione delle tabelle, delle relazioni tra di esse, la progettazione dei campi e la valutazione delle query necessarie per l'interazione con i dati.

Progettazione Software

Per lo sviluppo software si è utilizzato l'ambiente di sviluppo Visual Studio Code e linguaggio html/php per quanto riguarda il front-end (con l'implementazione del framework Tailwind), mentre per il backend e la parte di server rest si è utilizzato Fat Free.

Test e Debug

Durante questa fase, la piattaforma verrà sottoposta a test approfonditi per identificare e correggere eventuali errori o problemi. Questo include test di unità, test di integrazione e test funzionali per assicurare che la piattaforma funzioni correttamente e risponda alle esigenze degli utenti.

4. DESCRIZIONE DEI PROCESSI GESTIONALI

4.1 ASSUNZIONI, DIPENDENZE E VINCOLI

Assunzioni

- Gli utenti (studenti, professori, responsabili PCTO) avranno una conoscenza di base dell'uso delle piattaforme web.
- Le aziende collaboranti forniranno informazioni accurate e aggiornate.

Dipendenze

- Disponibilità dei dati aziendali.
- Collaborazione delle scuole e delle aziende partecipanti.
- Continuità del supporto tecnologico e delle infrastrutture.

Vincoli

- Budget limitato di €200.000.
- Scadenze temporali per il completamento delle diverse fasi del progetto.
- Normative legali e regolamentari riguardanti la privacy e la gestione dei dati degli utenti.

4.2 MECCANISMO DI MONITORAGGIO E CONTROLLI

Per assicurare che il progetto proceda secondo i piani, saranno implementati i seguenti meccanismi di monitoraggio e controllo:

- Revisioni periodiche del progetto: Incontri regolari per valutare lo stato di avanzamento e risolvere eventuali problemi.
- Report di stato: Report settimanali e mensili che dettagliano il progresso rispetto agli obiettivi.
- Strumenti di gestione del progetto: Utilizzo di software di gestione del progetto come Jira o Trello per tracciare le attività e le scadenze.
- Feedback continuo: Raccolta di feedback dagli utenti durante le fasi di sviluppo e test.

4.3 TABELLA RISCHI

Numero	Descrizione del Rischio	Probabilità	Impatto	Livello di Rischio	Azioni di Mitigazione
1	Ritardi nello sviluppo a causa di requisiti non chiari	Alta	Alto	Alto	Effettuare incontri regolari con gli stakeholder per chiarire i requisiti. Documentare e convalidare i requisiti prima di iniziare lo sviluppo.
2	Mancata disponibilità delle risorse umane necessarie	Media	Alto	Medio	Avere un piano di riserva per assicurare la disponibilità delle risorse. Considerare l'outsourcing di determinate attività in caso di necessità.
3	Problemi di compatibilità con i browser e i dispositivi	Media	Medio	Medio	Effettuare test approfonditi su una varietà di browser e dispositivi durante lo sviluppo. Utilizzare strumenti di compatibilità per identificare e risolvere tempestivamente i problemi.
4	Vulnerabilità della sicurezza dei dati	Media	Alto	Alto	Implementare misure di sicurezza robuste come crittografia dei dati, autenticazione multi-fattore e monitoraggio continuo. Effettuare regolarmente test di

					penetrazione per individuare e correggere le vulnerabilità.
5	Modifiche dei requisiti durante lo sviluppo	Alta	Medio	Alto	Adottare metodologie agili per gestire le modifiche dei requisiti in modo efficiente. Mantenere una comunicazione continua con gli stakeholder per comprendere e gestire le modifiche richieste.
6	Problemi di performance dell'applicazione	Media	Medio	Medio	Effettuare test di performance durante lo sviluppo per identificare e risolvere tempestivamente eventuali problemi. Ottimizzare il codice e l'infrastruttura per garantire prestazioni ottimali.
7	Perdita di dati a causa di errori umani o guasti hardware	Bassa	Alto	Medio	Implementare procedure di backup regolari e sicure per proteggere i dati. Utilizzare strumenti di monitoraggio per rilevare tempestivamente eventuali guasti hardware e intervenire prontamente.
8	Risorse finanziarie insufficienti per completare il progetto	Media	Alto	Medio	Monitorare attentamente il budget e le spese. Identificare anticipatamente eventuali scostamenti dal budget e adottare misure correttive. Valutare la possibilità di

					richiedere finanziamenti aggiuntivi se necessario.
9	Manutenzione e supporto post-implementazione non adeguati	Media	Medio	Medio	Pianificare un adeguato budget e risorse per la manutenzione e il supporto post-implementazione. Offrire formazione e documentazione dettagliate per gli utenti finali. Monitorare e rispondere prontamente alle richieste di assistenza e supporto.
10	Inesperienza nel team di sviluppo con le tecnologie richieste	Alta	Medio	Alto	Fornire formazione e supporto al team per acquisire competenze necessarie. Considerare l'assunzione di risorse con esperienza nel caso in cui la formazione non sia sufficiente.

5. DESCRIZIONE DEI PROCESSI TECNICI

5.1 METODI STRUMENTI E TECNICHE

I metodi, strumenti e tecniche utilizzati nel progetto includono:

Metodi di sviluppo agile

Per garantire una flessibilità ottimale e una risposta rapida ai cambiamenti delle esigenze del progetto, viene adottato un approccio di sviluppo agile. Questo metodo prevede:

- Iterazioni brevi e frequenti: Sviluppo in cicli brevi chiamati sprint (di solito di 1-2 settimane), durante i quali viene sviluppato un insieme di funzionalità specifiche.
- Feedback continuo: Revisione e adattamento costante basato sul feedback degli utenti e degli stakeholder, che consente di migliorare continuamente il prodotto.
- Collaborazioni frequenti: Team di progetto composto da sviluppatori, tester, project manager e stakeholder che lavorano insieme per raggiungere gli obiettivi del progetto.
- Pianificazione adattiva: Capacità di adattare il piano di progetto in base alle nuove esigenze e priorità emerse durante lo sviluppo.

Strumenti di sviluppo

Gli strumenti utilizzati per lo sviluppo del progetto sono selezionati per massimizzare l'efficienza e la qualità del codice prodotto:

- Visual Studio Code: Ambiente di sviluppo integrato (IDE) altamente configurabile, ideale per lo sviluppo web. Supporta estensioni per HTML, PHP, JavaScript, e CSS, che facilitano il coding e il debugging.
- Fat Free Framework (F3): Un framework PHP leggero e potente utilizzato per il backend, che offre una serie di funzionalità avanzate per la gestione delle richieste HTTP, la manipolazione dei dati e la sicurezza.
- Tailwind CSS: Un framework CSS altamente personalizzabile che permette di creare interfacce utente moderne e responsive in modo rapido e efficiente.

Tecniche di testing

Per garantire che la piattaforma funzioni correttamente e risponda alle esigenze degli utenti, vengono implementate diverse tecniche di testing:

-
- Test di Unità: Test automatici focalizzati su singoli componenti o unità di codice. Ogni funzione o metodo viene testato in isolamento per assicurarsi che funzioni come previsto.
 - Test di Integrazione: Test che verificano l'interazione tra diversi componenti del sistema. Questi test assicurano che i vari moduli funzionino correttamente quando integrati.
 - Test Funzionali: Test che verificano che il software esegua le funzioni e le operazioni previste dagli utenti finali. Questi test includono scenari realistici basati sui requisiti del progetto.
 - Test di Regressione: Eseguiti dopo modifiche al codice per garantire che le nuove implementazioni non abbiano introdotto errori nei componenti già esistenti.
 - Test di Usabilità: Valutazioni per assicurarsi che l'interfaccia utente sia intuitiva e facile da usare per gli utenti finali.

6. PIANIFICAZIONE DEL LAVORO, RISORSE UMANE E BUDGET

La pianificazione del lavoro sarà suddivisa nelle fasi già elencate e descritte. Le risorse umane coinvolte includono sviluppatori, tester e un project manager. Il budget totale stimato per il progetto è di €200.000.

6.1 DETTAGLIO COSTI

- Affitto locale: €1.000 al mese (€2.000 totale)
- Affitto server: €2.000 al mese (€4.000 totale)
- Sviluppatori: €60.000
- Tester: €10.000
- Project Manager: €40.000
- Hardware: €10.000
- Altro (spese generali, formazione, etc.): €20.000
- Spese Legali: €20.000
- Deposizione logo e marchio: €5.000 (da mantenere nel tempo)

6.2 RISORSE NECESSARIE

Per la realizzazione del progetto sono necessarie diverse risorse, sia umane che materiali:

Risorse Umane

Un team composto da sviluppatori front-end e back-end, tester per assicurare la qualità del prodotto, un project manager per coordinare le attività e gestire il progetto, e un esperto di UX/UI per garantire che l'interfaccia utente sia intuitiva e accessibile.

Infrastrutture Tecnologiche

Server affidabili per ospitare la piattaforma, strumenti di sviluppo e testing come Visual Studio Code, Fat Free Framework, Tailwind CSS, JUnit/PHPUnit, Selenium e Postman.

Materiali

Hardware necessario per il team di sviluppo, inclusi computer, strumenti di testing e reti sicure.

Budget Finanziario

Una gestione attenta del budget di €200.000 per coprire tutti i costi previsti, inclusi stipendi, affitti, hardware, software, spese legali e altri costi operativi.

6.3 PIANIFICAZIONE – DEADLINES E MILESTONE

Il progetto aveva come termine il 16/05 ed è iniziato circa 3 mesi fa. Di seguito è riportata una pianificazione dettagliata delle scadenze e delle milestones raggiunte:

Inizio Progetto: 15/02

- **Fase di Analisi dei Requisiti: 15/02 - 28/02**
 - Raccolta e analisi delle esigenze
 - Definizione dei requisiti funzionali e non funzionali
 - Creazione della documentazione dettagliata
 - **Milestone 1 (Fine Analisi dei Requisiti): 28/02**
- **Progettazione Database: 01/03 - 10/03**
 - Definizione delle tabelle e delle relazioni
 - Creazione dello Schema ER e dello Schema Logico
 - Valutazione delle query necessarie
 - **Milestone 2 (Progettazione Database Completa): 10/03**
- **Progettazione Software: 11/03 - 25/03**
 - Sviluppo dell'architettura del sistema

- Implementazione del front-end con HTML/PHP e Tailwind CSS
- Sviluppo del backend con Fat Free Framework
- **Milestone 3 (Progettazione Software Completa): 25/03**
- **Sviluppo Software: 26/03 - 30/04**
 - Codifica delle funzionalità principali
 - Integrazione dei moduli frontend e backend
 - **Milestone 4 (Funzionalità Principali Implementate): 15/04**
- **Test e Debug: 01/05 - 10/05**
 - Esecuzione di test di unità, integrazione, funzionali e di regressione
 - Correzione di eventuali bug e problemi identificati
 - **Milestone 5 (Test e Debug Completati): 10/05**
- **Revisione Finale: 11/05 - 15/05**
 - Distribuzione della piattaforma sui server di produzione
 - Revisione finale e verifica del sistema in ambiente live
 - Raccolta del feedback iniziale dagli utenti
 - **Milestone 6 (Deploy e Revisione Finale Completati): 15/05**

Termine del Progetto: 16/05

LOC = 6500 righe di codice

7. CONCLUSIONI

In conclusione, il progetto di sviluppo di una piattaforma unica per la gestione dell'alternanza scuola-lavoro rappresenta un passo significativo verso l'innovazione e l'efficienza nella gestione delle esperienze formative degli studenti. L'obiettivo principale di creare un sistema intuitivo e user-friendly è stato raggiunto grazie all'uso di tecnologie moderne e approcci di sviluppo agili.

L'implementazione di sistemi di sicurezza robusti e test approfonditi ha assicurato che la piattaforma sia non solo funzionale, ma anche sicura e affidabile. Il progetto, completato entro i tempi previsti e rimanendo nei limiti del budget, dimostra come una pianificazione dettagliata e una gestione efficace delle risorse siano cruciali per il successo di progetti complessi.

Con la piattaforma ormai operativa, il prossimo passo sarà monitorare e rispondere ai feedback degli utenti per garantire che la piattaforma continui a soddisfare le loro esigenze e rimanga al passo con le evoluzioni tecnologiche e normative.

