

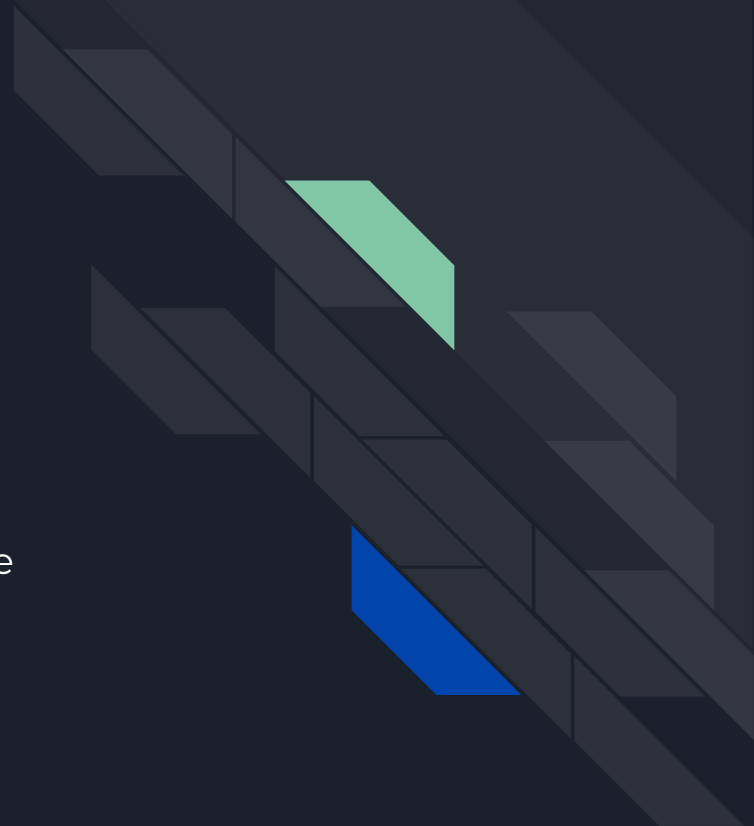


# Progetto Ingegneria del Software

# Distribuzione del team

Nel nostro progetto, io e mio fratello ci siamo suddivisi il lavoro in due macro-aree: **front end** e **back end**. Ognuno di noi ha sviluppato la propria parte in modo autonomo, per poi integrarle nella fase finale.

La progettazione del **database**, invece, è stata svolta in modo collaborativo e sincrono. Abbiamo lavorato insieme per analizzare e strutturare al meglio i dati, adottando un approccio simile a un brainstorming, in modo da esplorare diverse prospettive e trovare soluzioni ottimali.





# Evidenze delle cerimonie

Abbiamo adottato un approccio **AGILE/SCRUM** per organizzare e gestire lo sviluppo del progetto in modo efficace.

All'inizio, durante lo **Sprint Planning**, abbiamo definito le funzionalità principali e suddiviso le attività tra frontend e backend, assegnando priorità e stimando i tempi di sviluppo.

Durante il processo, ci siamo confrontati attraverso incontri periodici simili ai **Daily Stand-up**, per aggiornare lo stato di avanzamento, risolvere eventuali problemi e coordinare l'integrazione tra le parti.

Al termine di ogni fase, abbiamo effettuato una "**Sprint Review**", verificando il corretto funzionamento delle funzionalità implementate e apportando miglioramenti quando necessario.



# Oggetto del progetto

Il nostro obiettivo era sviluppare un sistema di gestione delle **FAQ** per il sito web, permettendo agli utenti di consultare rapidamente le risposte alle domande più frequenti.

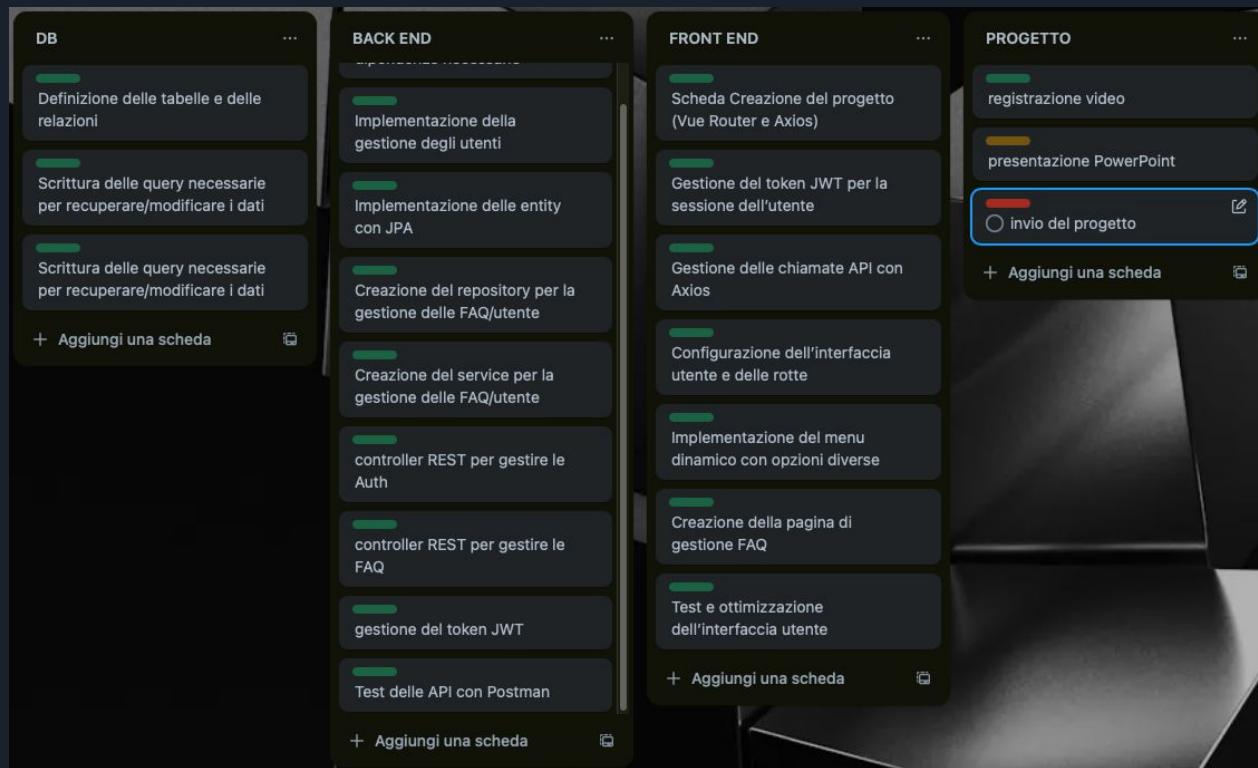
Durante la progettazione, ci siamo chiesti se fosse necessaria una figura amministrativa per la gestione delle FAQ. Per questo motivo, abbiamo deciso di implementare anche un sistema di **autenticazione**, seppur in modo rapido e non ottimizzato come avrebbe richiesto un progetto specifico per la gestione degli accessi.

Il sistema distingue due tipologie di utenti:

- **Utente standard:** può visualizzare le FAQ, anche senza essere registrato.
- **Amministratore:** ha accesso a un menu dedicato che gli consente di **aggiungere** o **rimuovere** FAQ dal database.

Questa suddivisione garantisce un controllo più strutturato sulle informazioni pubblicate, consentendo agli utenti di accedere facilmente ai contenuti e all'amministratore di mantenere aggiornato il database delle FAQ.

# Gestione del progetto con Trello





# Disegno architetturale scelto

Abbiamo adottato un'**architettura ibrida**, combinando il **pattern MVC (Model-View-Controller)** nel backend con un'**architettura a due livelli (Two-Tier Architecture)** per separare frontend e backend.

- **Backend (Spring Boot - MVC):**

Il backend segue il pattern **MVC**, in cui:

- **Model:** rappresenta le entità del database tramite **JPA**.
- **Controller:** espone le API REST per la gestione delle FAQ.
- **Service:** contiene la logica di business, separando il codice di gestione dei dati dalla logica dell'applicazione.
- **Repository:** gestisce l'accesso al database con **Spring Data JPA**.

- **Frontend (Vue.js - REST API):**

Il frontend è un'applicazione separata che comunica con il back end tramite **Axios** per ottenere e modificare i dati.

La UI è dinamica e gestisce la visualizzazione delle FAQ con un approccio **reattivo**.



# Funzionalità Implementate

## 1. Visualizzazione delle FAQ (Utente e Ospite)

- Le FAQ sono visibili **anche senza autenticazione** (inizialmente, viene mostrato **solo l'elenco delle domande**)
- Cliccando su una domanda, la risposta viene visualizzata (il caricamento delle FAQ avviene tramite chiamata API al backend)

## 2. Autenticazione e Ruoli Utente

Esistono **due tipologie di utenti** (l'autenticazione è basata su **JWT - JSON Web Token**):

- **Utente normale** (registrato o ospite): può **solo visualizzare** le FAQ.
- **Admin**: ha permessi speciali per **gestire** le FAQ.

## 3. Gestione delle FAQ (Admin)

- **Aggiunta di una nuova FAQ**
- **Eliminazione di una FAQ**



# Dettaglio di una funzionalità

## Funzionalità: Aggiunta di una nuova FAQ

Dal lato **back end**, è stata sviluppata un'API REST con Spring Boot che gestisce la richiesta di aggiunta. L'admin invia una nuova domanda e risposta attraverso una chiamata HTTP **POST**, che viene validata e salvata nel database MySQL tramite JPA. Il sistema utilizza **Spring Security** con autenticazione JWT per garantire che solo gli amministratori possano accedere a questa funzione.

Dal lato **frontend**, è stata creata un'interfaccia in Vue.js con un **form interattivo** dove l'admin può inserire la domanda e la risposta. Una volta confermati i dati, viene inviata una richiesta API al back end utilizzando **Axios**.

Se l'operazione va a buon fine, la nuova FAQ viene immediatamente aggiornata nella lista visibile agli utenti, senza bisogno di ricaricare la pagina.





# Feedback della vostra esperienza.

L'intero progetto è stato un'esperienza stimolante e formativa. La suddivisione del lavoro tra frontend e backend ci ha permesso di sviluppare in parallelo, migliorando l'organizzazione e la comprensione delle tecnologie utilizzate.

Nonostante alcune sfide, come l'integrazione delle API e la gestione dell'autenticazione, siamo riusciti a risolverle con successo, affinando il nostro approccio.

Siamo molto soddisfatti del risultato finale: un sistema funzionale, ben strutturato e facilmente espandibile.

Questo progetto ci ha permesso di consolidare le nostre competenze e di collaborare in modo efficace, rendendo l'intero processo estremamente gratificante!