

“Kanban Board”

VERSIONE FINALE – Aggiunte e correzioni sono mostrate in **rosso**

Progettare e implementare un'applicazione web per creare e gestire Kanban boards, cioè, boards basate sull'utilizzo di schede per organizzare e prioritizzare dei progetti (per esempio, passi necessari per completare un certo progetto, l'organizzazione del vostro prossimo viaggio – vedere https://en.wikipedia.org/wiki/Kanban_board).

Una **board** Kanban è una pagina a più colonne. Ogni board ha un nome, un proprietario, e può essere condivisa con altri utenti. Ogni **colonna** nella board ha un *titolo* e una o più **schede**. Le colonne e le schede possono essere create, editate, e cancellate dinamicamente. Una board deve avere almeno 2 colonne. Una colonna può avere un numero qualsiasi di schede (incluso 0). Ogni scheda ha queste proprietà:

- Un titolo (obbligatorio)
- Una descrizione (obbligatoria)
- Dei link di interesse (**una lista di zero o più URLs**)
- Una scadenza (opzionale).

Tali proprietà possono essere inserite quando l'utente crea la scheda, e possono essere modificate più tardi, quando l'utente edita la scheda.

Le schede possono essere spostate all'interno delle colonne e tra colonne tramite un insieme di 4 bottoni presenti in ogni scheda (su/giù all'interno della colonna, e sinistra/destra verso colonne adiacenti). Le schede possono essere *archivate*, cioè, ancora presenti nella colonna, ma posizionate al fondo della colonna. Le schede archiviate mostrano solamente il titolo e la descrizione e consentono una sola azione: estrazione (che posizionerà la scheda di nuovo nella parte alta della colonna).

L'applicazione web consente agli utenti autenticati di gestire le loro boards, che sono mostrate nella pagina principale (dopo il login) e di creare nuove boards. La pagina principale lista tutte le boards (di cui l'utente è proprietario o condivise con l'utente), con il loro nome, il numero totale di schede, e il numero delle schede scadute (cioè schede che hanno una scadenza che è passata).

L'utente può svolgere varie operazioni sulla board stessa, sulle colonne e sulle carte (si veda Tabella 1). Dopo ogni modifica (Tabella 1), l'applicazione deve immediatamente salvare il cambiamento sul database del server.

Il creatore della board può condividere la board stessa con un qualsiasi altro utente **autenticato esistente**. **Quando una board è condivisa con un utente, quell'utente può fare le stesse identiche operazioni sulla board che farebbe il proprietario (incluso cancellarla)**. Una pagina specifica dell'applicazione deve listare, in due sezioni, le “boards condivise con altri” e le “boards condivise con me”.

La gestione di modifiche che avvengono contemporaneamente non è richiesta. Si assuma che una data board sia aperta solamente da un utente alla volta.

L'applicazione web, quando visitata da un utente non autenticato, deve mostrare una pagina con una "board di esempio" a sola lettura, per incentivare i visitatori a loggarsi e iniziare a creare le loro boards.

Tabella 1: Azioni disponibili sulla Kanban board

Operazione	Da...
Creare una nuova board	... la pagina principale
Cancellare una board	... la pagina principale o la pagina delle boards condivise
Condividere una board con un altro utente	... dentro la board
Creare, rinominare o cancellare una colonna	... dentro la board
Creare una scheda	... dentro la colonna
Editare una scheda	... dentro la scheda (eccetto quelle archiviate)
Spostare (su/giù/sinistra/destra), cancellare o archiviare una scheda	... dentro la scheda (eccetto quelle archiviate)
Estrarre una scheda dall'archivio	... dentro la scheda archiviata

Requisiti del progetto

- L'architettura dell'applicazione e il codice sorgente devono essere sviluppati adottando le migliori pratiche (best practices) di sviluppo del software, in particolare quelle per le single page applications (SPA) che usano React e REST.
- Il progetto deve essere realizzato come applicazione React, che interagisce con un'API REST implementata in Node+Express. Il database deve essere memorizzato in un file SQLite.
- La comunicazione tra il client ed il server deve seguire il pattern "React Development Proxy" e React deve girare in modalità "development".
- La directory radice del progetto deve contenere un file README.md e contenere due subdirectories (client e server). Il progetto deve poter essere lanciato con i comandi: "cd server; nodemon server.js" and "cd client; npm start". Uno scheletro delle directory di progetto sarà fornito.
- L'intero progetto deve essere consegnato tramite GitHub, nel repository creato da GitHub Classroom.
- Il progetto non deve includere le directories in node_modules. Esse devono essere ricreabili tramite il comando "npm install", subito dopo "git clone".
- Il progetto può usare librerie popolari e comunemente adottate (come per esempio moment.js, react-bootstrap, ecc.), se applicabili e utili.
- L'autenticazione dell'utente e l'accesso devono essere realizzati tramite token JWT, memorizzato in un cookie http-only. Non è richiesto alcun ulteriore meccanismo di protezione.
- Il database del progetto deve essere definito dallo studente e deve essere precaricato con almeno 5 utenti di esempio (la procedura di registrazione non è richiesta), e almeno 4 boards ognuna con alcune colonne e relative schede.

Contenuto del file README.md

Il file README.md deve contenere le seguenti informazioni (un template sarà disponibile nello scheletro del progetto – in generale, ogni spiegazione non dovrebbe essere più lunga di 1-2 righe):

1. Una lista delle Route dell'Applicazione React, con una breve descrizione dello scopo di ogni route
2. Una lista delle API REST offerte dal server, con una breve descrizione dei parametri e delle entità scambiate
3. Una lista delle tabelle del database, con il loro scopo
4. Una lista dei principali componenti React usati nell'applicazione
5. Uno screenshot **della pagina di editing della Kanban board** (embeddando una immagine messa nel repository)
6. Username e password dei 5 utenti di test. Identificare un utente che ha condiviso almeno una board ed ha almeno una board condivisa con lui.