

Esercizio di Programmazione: Sistema di Gestione Fatture in JavaScript

Obiettivo: Sviluppare un sistema di gestione fatture in JavaScript che permetta agli utenti di creare, modificare, eliminare e visualizzare fatture, utilizzando il metodo Test Driven Development (TDD) e memorizzando i dati in una variabile.

Descrizione del Progetto

Questo progetto mira a sviluppare un sistema base per la gestione delle fatture che utilizza le strutture dati in memoria per il mantenimento temporaneo dei dati durante l'esecuzione dell'applicazione. L'uso di TDD garantirà che tutte le funzioni siano ben progettate e testate prima della loro implementazione.

Tecnologie Suggeste

- **Linguaggio:** JavaScript
- **Framework di Test:** Jest per i test unitari

Milestones

1. Setup Iniziale del Progetto

- Inizializzare un progetto Node.js con Jest configurato per eseguire i test unitari.

2. Struttura dei Dati

- Definire un modello di dati per le fatture utilizzando classi o oggetti semplici in JavaScript.
- Implementare una struttura in memoria (array o oggetto) per mantenere le fatture durante la sessione.

3. Implementazione delle Funzioni CRUD

- Sviluppare funzioni per aggiungere, modificare, eliminare e recuperare le fatture dalla struttura dati in memoria.
- Implementare validazioni per verificare la correttezza dei dati inseriti.

4. Test e Ottimizzazione

- Scrivere test unitari dettagliati per ogni funzione, inclusi test per la validazione dei dati e per la logica CRUD.
- Ottimizzare la gestione dei dati per garantire la massima efficienza in termini di accesso e modifica delle informazioni.

Checklist dei Tipi di Test da Fare

• Test di Validità dei Dati

- Verifica che i campi obbligatori di una fattura non siano vuoti.
- Assicurarsi che i formati di data e numerici siano corretti.

• Test di Funzionalità CRUD

- Test per l'aggiunta di nuove fatture.
- Test per la modifica di fatture esistenti.
- Test per l'eliminazione di fatture.
- Test per il recupero e la visualizzazione di fatture.

• Test di Gestione degli Errori

- Verifica che il sistema gestisca correttamente i tentativi di salvataggio di dati non validi.
- Test per la gestione di input non validi e messaggi di errore appropriati.

Consegnabile Finale

- Un repository Git con il codice sorgente completo.
- Una suite di test unitari che verifica tutte le funzionalità.
- Documentazione su come utilizzare le funzioni del sistema.