Fullstack Task Manager

[Il Tuo Nome]

30 settembre 2024

Indice

1 Descrizione del Progetto

Questo progetto è un'applicazione web full-stack che permette agli utenti di effettuare il login e visualizzare messaggi di successo o errore in base alle credenziali inserite. L'obiettivo principale è stato quello di:

- Implementare un **frontend** con **React** e **TypeScript**.
- Creare un **backend** con **Flask** in **Python**.
- Utilizzare **Docker** e **Docker Compose** per containerizzare l'applicazione.
- Gestire le rotte con **React Router**.
- Fornire istruzioni chiare per configurare e avviare il progetto su qualsiasi macchina con Docker installato.

2 Tecnologie Utilizzate

2.1 Frontend

- React
- TypeScript
- React Router DOM

2.2 Backend

- Python
- Flask
- PyJWT (per la gestione dei token JWT)
- Flask-CORS (per gestire le richieste cross-origin)

2.3 Containerizzazione

- Docker
- Docker Compose

3 Prerequisiti

Per eseguire questo progetto, è necessario avere installato:

- **Docker**: Installazione di Docker
- **Docker Compose**: Installazione di Docker Compose

4 Struttura del Progetto

La struttura delle directory del progetto è la seguente:

```
fullstack-task-manager/
 backend/
    app.py
    requirements.txt
    Dockerfile
 frontend/
    src/
       App.tsx
       index.tsx
       components/
```

```
        Login.tsx
        SuccessPage.tsx
        FailurePage.tsx
    package.json
    Dockerfile
docker-compose.yml
README.md
```

5 Istruzioni per l'Installazione

Segui questi passaggi per configurare il progetto su un altro computer:

5.1 1. Clona il repository

Apri il terminale e clona il repository:

```
git clone https://github.com/tuo-username/fullstack-task-manager.git
cd fullstack-task-manager
```

Assicurati di sostituire `tuo-username` con il tuo nome utente GitHub o l'URL corretto del repository.

5.2 2. Costruisci e avvia i container Docker

Esegui il seguente comando nella directory principale del progetto:

```
docker-compose up --build
```

Questo comando:

- **Costruisce** le immagini Docker per il frontend e il backend.
- **Avvia** i container per il frontend e il backend.
- **Configura** la rete Docker per consentire la comunicazione tra i container.

6 Come Avviare l'Applicazione

Dopo aver eseguito `docker-compose up -build`, i servizi saranno in esecuzione sui seguenti indirizzi:

- **Frontend**: http://localhost:3000
- **Backend**: Il backend è accessibile dal container frontend tramite il nome del servizio Docker `backend`.

Nota: Il backend non è esposto direttamente all'esterno, ma comunica con il frontend attraverso la rete Docker interna.

7 Come Utilizzare l'Applicazione

1. **Accedi al frontend** aprendo un browser web e navigando su http://localhost:3000.
2. **Pagina di Login**:
   - Inserisci le credenziali:
     - **Username**: `user`
     - **Password**: `password`
   - Clicca sul pulsante **"Login"**.
3. **Risultato**:
   - **Credenziali Corrette**:
     - Verrai reindirizzato a una pagina con il messaggio *"Le tue credenziali sono corrette"*.
   - **Credenziali Errate**:
     - Verrai reindirizzato a una pagina con il messaggio *"Le tue credenziali non sono corrette"*.

8 Troubleshooting

Se incontri problemi durante l'installazione o l'esecuzione dell'applicazione, prova le seguenti soluzioni:

8.1 1. Verifica che Docker sia in esecuzione

Assicurati che Docker sia installato correttamente e che il demone Docker sia in esecuzione.

8.2 2. Verifica i container in esecuzione

Esegui il comando:

```
docker ps
```

Dovresti vedere i container per il frontend e il backend in esecuzione.

8.3 3. Controlla i log dei container

Per visualizzare i log del frontend:

```
docker-compose logs frontend
```

Per visualizzare i log del backend:

```
docker-compose logs backend
```

8.4 4. Porte in uso

Assicurati che le porte **3000** (frontend) e **5000** (backend) non siano già in uso da altre applicazioni sul tuo computer.

8.5 5. Problemi di cache

Se incontri problemi con la cache di Docker, puoi eseguire:

```
docker system prune --all --force
```

Attenzione: Questo comando rimuove tutte le immagini, i container e le reti non utilizzate.

8.6 6. Riavviare i container

Se hai apportato modifiche al codice, devi riavviare i container:

```
docker-compose down
docker-compose up --build
```

9 Possibili Miglioramenti

Per approfondire le tue competenze e migliorare l'applicazione, puoi considerare le seguenti implementazioni:

- **Gestione dello stato globale**: Integra Redux o la Context API di React.
- **Interazione con un database**: Aggiungi un database come PostgreSQL per memorizzare utenti e task.
- **Autenticazione avanzata**: Implementa la registrazione degli utenti e l'autenticazione JWT completa.
- **UI/UX migliorata**: Utilizza librerie come Material-UI o Ant Design per migliorare l'interfaccia utente.
- **Test automatizzati**: Aggiungi test unitari e di integrazione per frontend e backend.
- **Deployment**: Distribuisci l'applicazione su un servizio cloud come Heroku o AWS.

10 Autore

Progetto sviluppato da **[Il Tuo Nome]**.

Se hai domande o necessiti di assistenza, non esitare a contattarmi.

Grazie per aver utilizzato il Fullstack Task Manager!

Note Finali

Questo progetto è stato creato per scopi educativi, con l'obiettivo di apprendere e mettere in pratica tecnologie come React, Docker, Python e TypeScript. Sentiti libero di esplorare il codice, apportare modifiche e utilizzare questo progetto come base per sviluppi futuri.