

HiveMind

DAVIDE DI PIERRO | N86004755

Descrizione del Progetto

La piattaforma di condivisione HiveMind è realizzata come Single Page Application (SPA) sviluppata per gestire e visualizzare dati dinamici attraverso un'interfaccia utente moderna e responsiva. Utilizza Angular per il frontend e Node.js con Express per il backend. Il frontend comunica con il backend tramite API RESTful per ottenere e manipolare dati.

Backend (Node.js con Express)

Tecnologie Principali:

- **Node.js:** Piattaforma server-side basata su JavaScript per l'esecuzione di codice JavaScript.
- **Express:** Framework web per Node.js che semplifica lo sviluppo di API RESTful e applicazioni web.
- **Swagger:** Utilizzato per la documentazione delle API mediante Swagger UI Express.

ORM (Object-Relational Mapping) e database:

- **Sequelize:** ORM per Node.js che supporta diversi database relazionali come PostgreSQL tramite pg e SQLite tramite sqlite3 (Entrambi sono inclusi nel progetto ma la migrazione da SQLite a PostgreSQL si è rivelata più ostica del previsto e di conseguenza abbandonata).

Middleware e Utility:

- **Cors:** Middleware per la gestione delle richieste HTTP CORS.
- **Dotenv:** Per il caricamento delle variabili di ambiente da file .env
- **JsonWebToken** (jsonwebtoken): Utilizzato per l'autenticazione basata su token JWT.
- **Morgan:** Middleware per il logging delle richieste HTTP.
- **Nodemon:** Utilizzato in ambiente di sviluppo per il riavvio automatico dell'applicazione dopo le modifiche al codice.

Frontend (Angular)

Tecnologie Principali:

- **Angular:** Framework per lo sviluppo di SPA, utilizzando TypeScript per il coding.

Componenti UI e Librerie:

- **Angular Material:** Libreria di componenti UI per Angular.
- **Ngx-Infinite-Scroll** e **Ngx-Masonry:** Utilizzati per la gestione del caricamento infinito e per la disposizione dinamica dei contenuti.
- **Ngx-Toastr:** Libreria per la gestione delle notifiche.

- **Tailwind CSS:** Framework CSS utility-first per la costruzione rapida di interfacce moderne e responsive.

Docker

Il progetto utilizza Docker Compose per l'orchestrazione dei container necessari per lo sviluppo e il deployment:

Backend Container:

- Utilizza **Node.js** per eseguire l'applicazione Express.

Frontend Container:

- Angular viene compilato e distribuito con **Node.js** e usa **Nginx** per servire l'applicazione.