Programmazione Web e Mobile

Rocco Amico - Matricola N. 12190A

Progetto:

MetricON nasce con l'idea di fornire statistiche chiave di un dato canale o di un dato video su Youtube utilizzando la API di Google "Youtube Data API v3".
Il progetto vuole abbattere la barriera di difficoltà tecnologica che all'utente comune comporterebbe un utilizzo diretto delle API, favorendo l'usabilità da interfaccia web e riducendo ad un unico click quello che si otterrebbe con molteplici chiamate API.
Il progetto è disponbile online al seguente link: https://www.roccoamico.it/metricon

Architettura

```
├── client
│   ├── channel
│   │   ├── channel.css
│   │   ├── channel.html
│   │   └── channel.js
│   ├── home
│   │   ├── home.css
│   │   └── home.html
│   └── video
│       ├── video.css
│       ├── video.html
│       └── video.js
└── server
    └── server.js
```

Logica

API Key

Registrandosi al programma sviluppatori di Google, è possibile chiedere l'accesso alle API, in particolare quella di nostro interessa è la Youtube Data API v3: viene fornito un token che dovrà essere utilizzato in tutte le chiamate.
Per comodità nel progetto la API Key è stata inserita come costante nel server.js (chiaramente è stata generata con un account usa e getta creato ad hoc), mentre nel progetto deployato online la API Key è definita come una variabile d'ambiente all'interno di DigitalOcean.

Backend

All'interno del server.js con l'utilizzo del framework web Express sono definite le route del sito per servire i file statici e vengono effettuate le chiamate API.
Oltre ad Express, per effettuare le chiamate utilizziamo Axios, che tramite XMLHttpRequest chiama la API (utilizzando l'apposito token) e ottiene indietro i dati (in formato JSON) che poi andremo a servire nel Frontend.
Questo file si occupa inoltre di servire il webserver (in locale alla porta 3000).

Frontend

All'interno della cartella client troviamo tre cartelle per le tre pagine che verranno servite:

- la home,
- la pagina per visualizzare le statistiche di un video,
- la pagina per visualizzare le statistiche di un canale.

channel.js e video.js si occupano di prendere come input il parametro inserito dall'utente dall'interfaccia web per fare una richiesta al server.js (con fetch) e, solo se la risposta da esito positivo, mostrare graficamente (utilizzando chart.js) i dati ottenuti.
In caso contrario verrà visualizzato un messaggio di errore.

Stile
Il sito utilizza congiuntamente una classe di Bootstrap e i fogli di stile per definire il layout grafico e la presentazione degli elementi.

Testing:
I parametri di input di /video sono stati testati con
- https://www.youtube.com/watch?v=6i1uPfPzXlc
- https://www.youtube.com/watch?v=WDmgVlwjFto
- https://www.youtube.com/watch?v=bef8QLNHubw

I parametri di input di /channel sono stati testati con
- queiduesulserver
- pewdipie
- Davie504

In entrambi gli input non è possibile fare una richiesta con input vuoto ed entrambi restituiscono un errore in caso di input non valido (canale non esistente, url video non esistente).

Inoltre è stato stato provato questo payload per triggerare una vulnerabilità XSS
- *<h1>Hello, <script>alert(1)</script>!</h1>*

con esito negativo in quanto c'è una validazione dell'input che non ne permette l'esecuzione.

Versioning e hosting:
Il progetto sfrutta git per il versionamento, è stato deployato su DigitalOcean e collegato alla zona DNS di OVH per il dominio roccoamico.it.

Istruzioni per l'esecuzione in locale:
npm install
npm run start