

Obiettivi generali dell'esercizio

L'esercizio si divide in due parti: sviluppo backend e sviluppo frontend.

Obiettivo principale dello sviluppo backend è la definizione di un endpoint API che restituisca il contenuto di un file CSV in formato JSON.

Obiettivo principale dello sviluppo frontend è la creazione di una single-page application che mostri, seguendo un design definito, i dati forniti dalla API di cui sopra.

Risultato

L'esercizio deve essere consegnato come codice sorgente, con istruzioni di "compilazione". Il risultato del build dell'applicazione sono dei file statici, da servire su browser tramite un Web Server.

Sviluppo Backend

L'esercizio può essere svolto in due modalità:

1) Conosci e usi tecnologie **Serverless**?

- Framework serverless (in ordine di preferenza)
 - **Serverless Framework, AWS SAM**
- Linguaggio di programmazione (in ordine di preferenza)
 - **Node.js**
 - **PHP**
- Cloud services (AWS)
 - **Lambda, API Gateway**

2) Alternativamente, puoi svolgerlo così

- Linguaggio di programmazione (in ordine di preferenza)
 - **Node.js**
 - Framework (in ordine di preferenza): **ExpressJS, Koa.js**
 - **PHP**
 - Framework (in ordine di preferenza): **CakePHP, Symfony, Laravel, Zend**
- Web Server (in ordine di preferenza)
 - **Nginx**
 - **Apache**
 - **Node/PHP built-in web server**
- Specifiche:
 - la business logic:
 - legge il CSV come stream, evitando il caricamento dell'intero file

- pagina i risultati a gruppi di 100
- la API è una route http del tipo: /api/flyers?page=1&limit=100

Sviluppo Frontend

- Linguaggi di programmazione e markup: Javascript, HTML, CSS (Sass)
- Possibilità di usare uno dei seguenti framework / librerie Javascript: React, Backbone
- Possibilità di usare una delle seguenti librerie UI:
 - Material Design Lite: <https://getmdl.io/>
 - Material UI (React): <https://material-ui-next.com/>
 - Semantic UI: <https://semantic-ui.com/>
 - Semantic UI React: <https://react.semantic-ui.com/>
 - Bootstrap: <https://getbootstrap.com/>
 - React Bootstrap: <https://react-bootstrap.github.io/>
- Specifiche:
 - utilizzare le View del framework PHP scelto
 - chiamata lato client alle API /api/flyers sviluppate in backend
 - gestione di empty state, loading state, error state
 - funzionalità di “salva nei preferiti” sfruttando il localStorage del browser
 - layout responsive che segua il mockup allegato
 - compatibilità con versioni di Internet Explorer >= 11

Allegati

- Flyers Data:
 - CSV: da usare nello sviluppo della API Backend
- Mockup:
 - iPhone
 - iPhone - Open Menu
 - iPad

Flyers Data

I dati nel CSV corrispondono a una piccola parte dei dati dei volantini di ShopFully.
In particolare:

- id: id del volantino
- title: titolo del volantino
- start_date: data di inizio dell'offerta del volantino
- end_date: data di scadenza dell'offerta del volantino (suggerimento: non si dovrebbero mostrare volantini scaduti)
- is_published: flag booleano (esempio: non mostriamo i volantini che abbiano un valore di is_published uguale a 0)
- retailer: nome del retailer di riferimento
- category: nome della categoria del volantino

Suggerimenti

- Preferisci eseguire entrambe le parti backend e frontend, piuttosto che completarne una sola “alla perfezione”
- Una volta scelto il framework, attieniti alle sue specifiche per eseguire l’esercizio e basati sulle documentazioni ufficiali
- Scrivi funzioni astratte e riutilizzabili, prevedendo una possibile crescita dell’applicazione
- Abbi un occhio di riguardo alle performance
- Una volta scelto il framework UI, guarda tutti suoi componenti (buttons, cards, etc.) e utilizza tutti quelli che, secondo te, servono alla realizzazione del layout
- Usa immagini di diverse dimensioni per le card dei volantini, prendi ispirazione dai volantini della home page di doveconviene.it

Nice to have

- Il codice HTML scritto o generato dal DOM è pulito, semanticamente corretto e conforme alle regole del W3C
- Il codice è leggibile e documentato
- L’applicazione segue le principali regole e best-practice per l’accessibilità
- Se il localStorage non è supportato, usa i cookies come fallback
- Gli elementi della griglia di Flyer vengono mostrati in lazy loading man mano che l’utente scrolla la pagina