

FCA

Teconologie utilizzate:

- Javascript
- Python 3.8
- Amazon AWS - DynamoDB
- Amazon AWS - Lambda Function
- Amazon AWS API-Gateway

DESCRIZIONE PROGETTO

Il layer va importante in ogni file “.html” dove lo si vuole implementare.

Ogni click effettuato dall’utente verrà catturato e inserito nel database, con informazioni aggiuntive, quali:

- Hash, generato dallo snapshot dell’elemento cliccato + il corrispettivo testo.
- Coordinate X e Y dell’elemento rispetto alla pagina web.
- Testo dell’elemento cliccato.
- Dominio di riferimento
- ID univoco
- Type dell’elemento

Prima del salvataggio dei dati il programma chiede al database se in base al corrispettivo hash dell’elemento appena cliccato, ha un suggerimento per il prossimo click.

Se si l’elemento corrispondente al “prossimo click” suggerito verrà evidenziato, altrimenti ritornerà una risposta vuota.

Una volta al giorno nella console AWS si eseguirà una Lambda Function, questa scansionerà la tabella contenente tutti i click effettuati fin’ora e genererà una tabella “Suggerimenti”.

Attualmente il programma funziona correttamente solo se si mantiene fissa la dimensione della finestra.

Config.js

Questo file contiene la configurazione per la connessione a DynamoDB.

misc.js

Questo file contiene le seguenti funzioni:

- **getHash** : genera un hash univoco prendendo come parametro una stringa
- **idTime**: ritorna un id univoco in base ai millisecondi.
- **checkText**: Se il testo della lettura OCR del canvas è vuoto, il record non viene salvato nel database.
- **checkInput**: prende l'elemento HTML premuto, se nella lista, il record non viene salvato.
- **screenshot**: screenshot globale della pagina web.

entrypoint.js

Questo file contiene le seguenti funzioni:

- **test**: Prende l'event appena cliccato e lo passa nella funzione checkInput, se = 1 ritorna e si ferma.
- **event_screen**: genera uno screenshot dell'elemento appena cliccato utilizzando la libreria html2canvas
- **modify**: Prende in input la foto generata da "event_screen" ed esegue la lettura OCR tramite la libreria "tesseract", se il testo è vuoto ritorna altrimenti, genera l'hash, e subito dopo lo passa al metodo getHint(per cercare se esiste un corrispettivo suggerimento) e salva il record nel Database.
- **getHint**: Effettua una chiamata Ajax su una corrispettiva API creata su AWS passando in input l'hash, la funzione lambda della corrispettiva, cercherà nella tabella "hint" se c'è un suggerimento da ritornare all'utente, altrimenti ritorna statusCode : 400.
Se il suggerimento esiste la funzione ritorna statusCode:200 e le corrispettive coordinate per evidenziare l'elemento da suggerire attraverso il metodo "light".
- **light**: Prende in input delle coordinate X e Y, cercando il corrispettivo elemento nella pagina web e lo evidenzia per 3 secondi.

STRUTTURA DATABASE.

Per il momento si ha la struttura seguente:

tabella_test:

La seguente tabella utilizza il dominio del sito come chiave di partizione e l'id univoco come chiave di sorting.

L'hash invece è utilizzato come "Index Secondario" come appoggio alla scansione e alle query da eseguire nella tabella.

hint:

La seguente tabella utilizza solo una chiave di partizione "click" e contiene il "next_click" e le sue relative coordinate.

Lambda Function:

UpdateNewTab:

La seguente lambda scansiona tutta la tabella "tabella_test" ed esegue un accoppiamento e parsing dei dati e li passa ad una funzione upload che ha il compito di andare a creare la tabella "hint"

GetHint:

La seguente funzione lambda si aspetta semplicemente un hash in input e in base ad esso se è presente nella tabella "hint" ritorna le coordinate del NEXT click, altrimenti ritorna una response: statusCode:400

IMPLEMENTAZIONI FUTURE

Per il momento il programma funziona correttamente se si mantiene fissa una WindowSize.

I suggerimenti sono per tutti gli utenti e non c'è distinzione di path dentro al sito.

In futuro si vuole implementare un metodo per utilizzare il programma con qualsiasi larghezza dello schermo e di dare i suggerimenti in base ad un possibile utente che sia Loggato in un possibile sito.

In caso si volesse utilizzare il layer per più siti web, andrebbe ampliato il parser dei dati prendendo come riferimento ad un certo punto anche il dominio.

Layer da inserire in ogni pagina web.

La pagina "blank.html" specifica gli script da importare se si vuole utilizzare il layer ove la struttura sia

-Cartella_prova

---js

---blank.html