COMMENTI CODICE ANA:

Feed:

- Feed.html:

```
107: <div id="posts"></div>
108: <p id="message"></p>
```

Idea di base: Se ci sono notti, si riempirà dinamicamente la zona a riga 107 con le loro informazioni, altrimenti apparirà un messaggio, riga 108. Ovviamente l'apparizione di uno dei due elementi (div o p), esclude l'altro, per questo si utilizzeranno le funzioni hide e show in feed.js .

- feed.js

All'inizio mi salvo come variabile globale l'username dell'utente, in modo che possa utilizzarlo nelle funzioni che riempiono il feed e controllano i bottoni save/saved e upvote/upvoted.

Obiettivo: caricare dinamicamente i post con le info delle notti create dagli utenti. Ogni post avrà:

- foto e nome degli elementi, anche artista o autore nel caso di libri e canzoni o album
- bottone more infos → porta ad una pagina che mostra le informazioni complete sulla specifica notte;
- Tasti save/saved e upvote/upvoted → se l'utente che sta visitando il feed ha salvato o ha messo mi piace a una o più notti che si trovano nel feed, queste avranno rispettivamente il bottone save settato a saved e il bottone upvoted settato ad upvoted, e non saranno clicccabili .In caso contrario, l'utente potrà cliccare su uno dei due bottoni, la scritta cambierà e diventerà non cliccabile e il db verrà aggiornato di conseguenza. Per fare ciò, ci saranno quattro funzioni:
  - FindSavedNights() / FindUpvotedNights;
  - Save_function() / Upvote_function() ;

Problema:

L'ideale sarebbe caricare le notti salvate e piaciute PRIMA di caricare i post, in modo da settare le scritte save/saved e upvote/upvoted in maniera adeguata, ma js è asincrono.

Soluzione: Per questo motivo si utilizza il costrutto async await (**righe 107-117**). In questo modo posso salvare le notti salvate e piaciute dall'utente e controllare l'id della notte per ogni post e decidere cosa scrivere.

Riga 140:

Gli elementi nel post verranno distribuiti dinamicamente secondo lo schema di una matrice: il primo elemento (elem1) andrà nella posizione (1,1), il secondo (elem2) nella posizione (1,2) e così via. Per sapere in quale posizione dovrà andare ogni elemento è importante tenersi il conto del numero di elementi. Ogni elemento, con foto e nome, verrà messo in un apposito container di nome "elem" + numero_elementi_trovati_fino_a_quel_punto, in modo da determinare la sua posizione.

FinalStep:

- finalStep.js:
  Righe 50-56:

```
$(document).ready(function () {
  var url = window.location.href ;
  var find = /\?/;
  if(find.test(String(url).toLowerCase()) == true){
    sendData();
  }
});
```

Dopo aver fatto il login a Twitter, questo rimanda l'utente alla pagina FinalStep.html e aggiunge all'url dei parametri che vanno passati al server in modo che si possano usare per pubblicare sul profilo dell'utente. Dal momento che non è detto che se l'utente si trova su FinalStep.html è perché ha fatto il login con Twitter ed è stato reindirizzato lì , controllo che nell'url ci siano quei parametri che aggiunge Twitter. In particolare, mi basta sapere se c'è un punto interrogativo. In caso affermativo, mando i dati al server, altrimenti no.

Home_login:

- home_login.js:
  Righe 14-19:
  Se l'utente accede tramite Google, non è a conoscenza di username e password. Quest'ultima, in particolare, è utile per eliminare il profilo o per cambiare la password stessa, quindi è importante comunicare queste due informazioni all'utente.

Righe 39-45:

Home login è, come dice il nome stesso, la home alla quale si accede dopo aver fatto il login o la registrazione. Ci si può arrivare in vari modi:

1. Cliccando su "Home" o "JustAPerfectNight" da qualsiasi pagina disponibile dopo il login o il signUp.
2. Dopo aver fatto l'accesso con Google, che sia questo login o registrazione.

Vale, dunque, lo stesso discorso fatto per finalStep.js. Se l'utente accede tramite Google, quest'ultimo aggiungerà all'url dei codici che serviranno per accedere ai dati (e-mail) dell'utente e che quindi dovranno essere mandati al server. Altrimenti, questi dati non andranno, ovviamente, mandati. Per questo motivo c'è lo stesso controllo effettuato in finalStep.js: vedere se sono stati aggiunti dati all'url e se c'è bisogno di mandarli al server.

MyPn/PostProfile/Post:

- **Idea di base:** tutte queste pagine hanno lo stesso scopo: mostrare il post completo, ovvero contenente tutti i dettagli riguardo alle varie notti. La struttura è la seguente: c'è uno scheletro html di base che contiene le sezioni, vuote, di tutte le varie informazioni disponibili per ogni elemento. La sezione dell'elemento x verrà riempita con le informazioni riguardanti l'elemento stesso se questo è presente nella notte, nascosta altrimenti.
- **In particolare:**
  - MyPn è la pagina che verrà visualizzata tramite il link mandato per email o postato su Twitter. Quindi in myPn.js si prende l'id della notte dall'url e si manda una richiesta al server, il quale restituirà un oggetto contente la notte

completa, quindi tutte le informazioni dettagliate per ogni elemento. Nella onload della richiesta, quando questa ha avuto successo, si usa quell'oggetto per riempire dinamicamente la pagina.

- PostProfile e Post sono le pagine che verranno visualizzate rispettivamente cliccando sul bottone more infos dal profilo o dal feed. Dal momento che in entrambe le pagine vengono fornite le informazioni sulle notti complete facendo le richieste al server e queste vengono salvate nel local storage, non serve fare una ulteriore richiesta al server per riempire PostProfile e Post, ma basta salvarsi l'id del post sul quale viene cliccato il bottone "more infos", trovare la notte con quell'id nel local storage e usare quelle informazioni direttamente.

Profile:

- Profile.html:

```
125     <p id="message"></p>
126     <!--codice my nights qui-->
127     <div id="nights_section">
128     </div>
129     <!--codice my nights qui-->
130     <div id="saved_section">
131     </div>
132     <!--codice my nights qui-->
133     <div id="favourites_section">
134     </div>
```

Idea di base: Se ci sono notti, si riempiranno dinamicamente le zone nelle righe 127, 130 o 133 con le loro informazioni, altrimenti apparirà un messaggio, riga 125. La sezione da riempire viene decisa in base a quale link tra My nights, My saved nights e My upvoted nights è stato cliccato. Di default My nights parte selezionato. Ovviamente l'apparizione di uno dei due elementi (div o p), esclude l'altro, per questo si utilizzeranno le funzioni hide e show in profile.js . Le dinamiche sono simili a feed.js, escluse le considerazioni sui bottoni save/saved e upvote/upvoted, che non sono presenti qui.