Università della Svizzera italiana

Facoltà di scienze informatiche

Web Atelier
Prof. Cesare Pautasso
Masiar Babazadeh
Vincenzo Ferme
Vasileios Triglianos

Assignment 6

Continuous shuffle playback

Given the track list of the AtelierBeats library view, the goal is to play back all songs one after the other. So when one song finishes playing, the next one should start automatically. The playback order can be sequential (following the visible track list) or random (shuffle). The user may toggle between playback modes. Keep track in the database of the current playback mode as part of the user profile together with the name of the song currently playing so that when the application is re-opened the last mode is remembered.

Assignment 7

Signup and login

A user should be able to signup on AtelierBeats. The required fields should be first name, last name, username, email and password. Client-side and server-side validation of the fields is required. After the signup process, where a new user profile is added to the database, the user should be able to login. After a successful login, users get access to the main page of your application. Users that are not authenticated should not have access to the main page of your application and get redirected back to the login page. After 10 minutes of inactivity, user authentication should expire.

Assignment 8

Unified Search

Users should be able to search across all your entities from a single search bar. The search bar should support auto completion to quickly display search results right under the input field. The search should be performed among all of your model's entities (Tracks, Artists, Albums, Playlists) and should display first the results from the entity you are currently browsing. Assume for example, that a user is currently browsing the artist(s) view. Searching for a term in the search bar should display results from the artist collection, if any, on top of your rendered results list. If a user is currently viewing a playlist, the top of the results list should contain entries from the Track model that are also members of the current playlist.

Clicking on a result of the search has different effects based on the entity:

- Tracks:
  - While on the Library view: show the track on the library, hiding all the other ones
  - While on a Playlist view: go to the position in the table where the track is and change the style of the row to highlight the track
- Artist: Show the corresponding artist page
- Album: Show the corresponding album page
- Playlist: Open the corresponding playlist

Università
della
Svizzera
italiana

**Facoltà
di scienze
informatiche**

Web Atelier
Prof. Cesare Pautasso
Masiar Babazadeh
Vincenzo Ferme
Vasileios Triglianos

Assignment 9

Playlist editor

Your application should support *full playlist editing*: Users should be able to *create*, *modify*, *rename* and *delete* persistent playlists. To modify the content of a playlist, a user should be able to drag and drop tracks to the playlist, remove tracks from the playlist and also re-order tracks in the playlist. Changes to the state of the playlists should be saved to the server immediately as they happen.
Moreover you have to implement *playlist sharing*: A user should be able to follow playlists of other users and store them in his/her account. The User Interface should distinguish owned playlists from followed ones (which cannot be changed). To implement the sharing feature you will need to extend the user profile in the database/API and provide a seed function to create two user profiles.

Assignment 10

New Track Upload

A user should be able to upload a new track. The form for the upload should contain, among others, 'artist' and 'album' fields. When a user starts typing in one of these fields they should get suggestions based on the existing artists or albums. If the typed information doesn't match any of the existing records, a new model instance should be created (artist or album depending on the field). The upload should also upload the media file (mp3) to the server.
An upload can start with either:

1) a button to "add new track"
2) dropping a file from the finder in the page.

The upload of the file should start in the background while the user is filling out the data that is later saved and associated with the file once the upload completes successfully and the user submits the upload form.