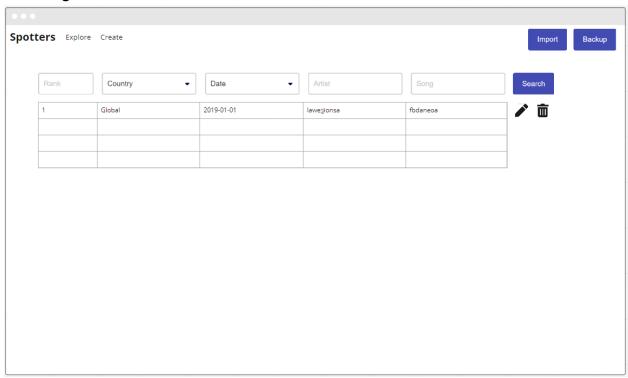
Sprint 3 Artifact

Members: Thet Zaw , Marco Baez Pulido , Jessie Lu , George Orduno Galicia , Yong Hwei Toon

Features:

- Feature 1: as a user, I want to insert a new song to the chart
- Feature 2: as a user, I want to update an existing song on the chart
- **Feature 3**: as a user, I want to delete an existing song on the chart
- Feature 4: as a user, I want to import and back up my data

GUI design:



Explore page



Create page

Test cases:

- Feature 1 Test Cases: as a user, I want to insert a new song to the chart
 - Test Case 1: artist name and song name empty
 Correct Output: error. those parameters need to be filled
 - Test Case 2: all parameters are valid
 <u>Correct Output</u>: inserts new song (with artist name) to the chart for that specific location, date, and position.
- Feature 2 Test Cases: as a user, I want to update an existing song on the chart
 - **Test Case 1**: artist name and song name empty Correct Output: error. those parameters need to be filled
 - **Test Case 2**: all parameters are valid <u>Correct Output</u>: updates an existing song (with artist name) in the chart for that specific location, date, and position.
- Feature 3 Test Cases: as a user, I want to delete an existing song on the chart
 - Test Case 1: as a user, I click the delete button next to a searched song
 <u>Correct Output</u>: website sends a request to the backend and deletes the song
 from the chart (data.csv) for that specific location, time, and position.
- Feature 4 Test Cases: as a user, I want to import and back up my data
 - Test Case 1: as a user, I exit
 Correct Output: CSV files are modified and data is backed up

To-do list:

Done list of last sprint

- Design GUI to search for songs from the Spotify's Daily Top 200 charts [finished by Thet Zaw and verified by everyone]
- Backend program to search for a record in csv file
 [finished by Jessie Lu and verified by everyone]
- Backend program to convert a record into a class
 [finished by David and verified by everyone]
- Research and Debugging
 - everyone
- misc: "logic.js"
 - alternative implementation of data structures, parsers, and search functions to improve search() from O(n) to O(1) time: "logic.js"
 - implemented by David and not verified
 - verify() to validate artist and song name user inputs
 - implemented by David, George and not verified
 - input date format converter (string to X/365)
 - implemented by Marco and not verified

To-do for next sprint

- GUI to update/insert/delete a song
 - Acceptance criteria: the user has a UI to input information
- GUI to import/backup the data
 - Acceptance criteria: the user can interact with the buttons which sends a request to the back-end
- Backend method to delete song from chart
 - Acceptance criteria: The method removes a row from in-memory database
- Backend method to insert and update song from chart
 - Acceptance criteria: The method inserts/updates a row in in-memory database
- Backend method to update chart without refresh
 - Acceptance criteria: table shows when search button is pressed without refreshing the page
- Backend method import/backup data
 - Acceptance criteria: the method stores in-memory database in a csv file when backing up and the method stores the csv file as in-memory database when importing