Sprint 1 Requirements Artifacts

Team 25

Requirement #1: Create Repository and setup Jira/GitHub integration (1)

* Detailed Description:
  + Initialize a public GitHub repository with main as the protected default
    - Agreed on utilizing a trunk-based development branching strategy.
  + Add the “GitHub for Jira” app so that Jira can read branches, commits, and PRs
    - Use issue keys (like E5P3-#) in branch names for linking/referencing.
* Acceptance:
  + Jira tickets show linked branches/PR and development work panel is active on the Jira Team Project page.

Requirement #2: Implement Spotify OAuth 2.0 login to authenticate users without creating new accounts (13)

* Detailed Description:
  + Authorization Code with PKCE:
    - The client is a browser on any device; backend exchanges code for tokens and stores refresh token server-side within an application session.
  + Spotify Access Scopes:
    - user-read-email, user-read-private, user-read-recently played.
* Acceptance:
  + Continuing with Spotify completes the PKCE flow, backend stores a refresh token, session cookies are set and received in requests, user is redirected to dashboard on successful login.

Requirement #3: Fetch user's username from Spotify (2)

* Detailed Description:
  + Fetch user’s username from Spotify’s API and have it within our system (Figure 1).

Requirement #4: Fetch user profile picture from Spotify (3)

* Detailed Description:
  + Fetch user’s profile picture from Spotify’s API and have it within our system (Figure 1).

Requirement #6: Setup database to store Spotify data (3)

* Detailed Description:
  + Created PostgreSQL database instance on external server, with custom default username and password settings for our 581 group.
  + Set up Cloudflare tunneling service to allow access to this database instance over the internet.
  + Created a basic test script to instantiate database connection and run simple query scripts.

Requirement #16: After logging in, show the logged-in user's dashboard by default (3)

* Detailed Description:
  + App routes enforce authentication guards. If a session cookie is valid, route to dashboard, else redirect to login.
  + On successful authentication, load user’s username and profile picture.
* Acceptance:
  + Post-login, accessing the dashboard route (‘/dashboard’) takes user to their dashboard page, without a visible flash of the login page.
  + A non-authenticated user attempting to hit the ‘/dashboard’ route results in a reroute to the login page.

Requirement #17: On application launch, display a log-in with Spotify landing page (5)

* Detailed Description:
  + A non-authenticated user navigating to the landing page (‘/login’) should see a button labeled “Login with Spotify” (Figure 2).
* Acceptance:
  + Clicking on this button should take the user to an OAuth 2.0 request page (Figure 4).

Requirement #19: Display user's username on user's dashboard (2)

* Detailed Description:
  + Post-authentication, load and display the user’s username at the top left corner of the dashboard header (Figure 1).

Requirement #20: Display user's profile picture on user's dashboard (2)

* Detailed Description:
  + Post-authentication, load and display the user’s profile picture at the top left corner of the dashboard header (Figure 1).

Requirement #22: Derive formula to calculate diversity score based on genres listened to (5)

* Detailed Description:
  + Shannon’s entropy is a measure of uncertainty, or the average amount of information contained in a variable’s outcomes.
  + By converting genres listened to by the user into a probability, we can use this formula to derive how diverse that users’ music taste is.
  + Entropy will be an impurity value between 0 and 1, with scores closer to 1 indicating strong, diverse genres and 0 indicating little genre variation.

Requirement #24: Derive formula to calculate a music taste rating based on the average of a series of listening histories (5)

* Detailed Description:
  + Using Shannon’s entropy from the diversity rating, the development team will average our scores and use that as a baseline for music taste.
  + Any user who wants their music taste rated will be given a taste score depending on how similar it is to the baseline.

# Appendix

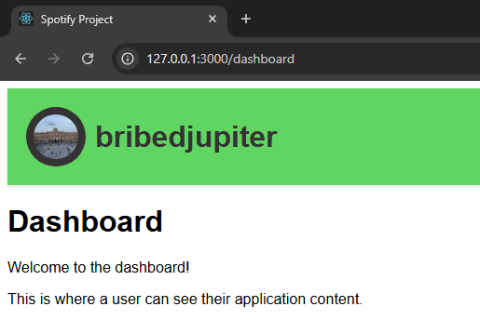


Figure 1. The current dashboard page.

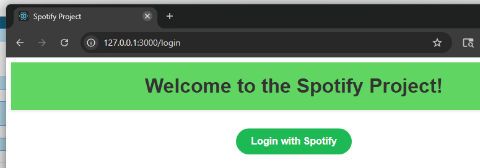


Figure 2. The login page.

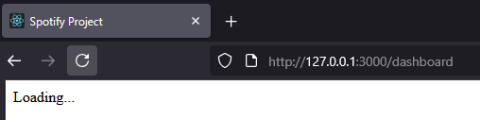


Figure 3. The loading screen while fetching data from Spotify.

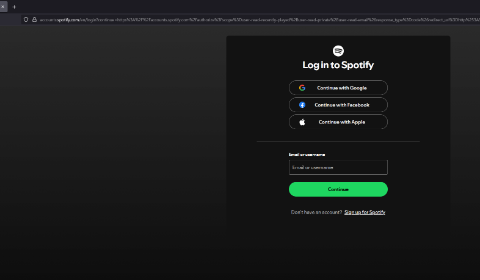


Figure 4. Spotify's authorization service's sign-in page.

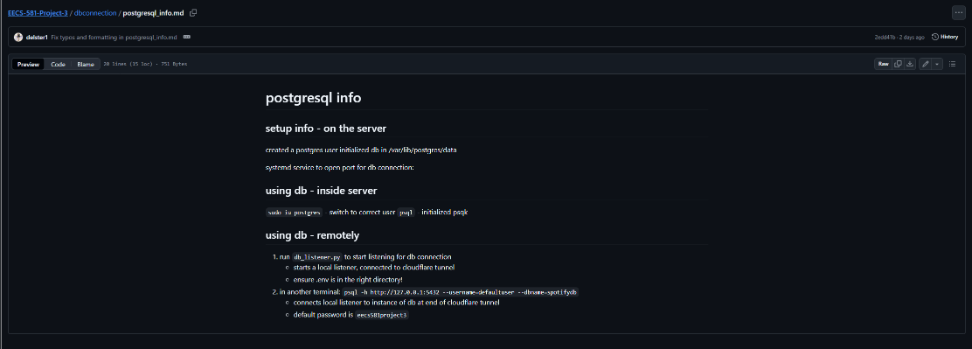


Figure 5. The database connection information.

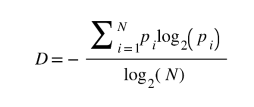


Figure 6. The genre diversity score formula.

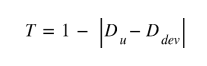


Figure 7. The music taste rating score formula.