Sprint 4 Plan

Product Name: Groovo

Team Name: Groovo

Sprint Completion Date: [12/1/2025]

Revision Number: [e.g., Rev. 1.0]

Revision Date: [11/17/2025]

Sprint Goal

Deliver social-sharing and recommendation features that enhance user engagement, while empowering moderators with essential content-management tools. This sprint will focus on enabling shareable profiles, highlighting an "Album of the Week," and implementing moderator controls for removing inappropriate content.

Task Listing (Organized by User Story)

User Story 1:

[3 pts] As a user, I would like to share my profile and reviews with a link so that I can share it to anyone I want to outside of the website

1. Task: Implement a "Share Profile" button on the user's profile page that copies the profile URL to the clipboard [1 pts]
    a. Add a share button UI element to the profile page layout
    b. Implement clipboard functionality to copy the URL
    c. Add a visual confirmation that says "Link Copied!"
    d. Style the share button to match the app's design
    e. Add hover/focus states for accessibility
2. Task: Add a "share review" button on each review component that copies the review's sharable link to the clipboard [2 pts]
    a. Add a share button inside the review component UI
    b. Add a confirmation message that says "Link Copied!" when the button is clicked
    c. Add hover, focus states for accessibility
    d. Test the share button inside different views (profile page, feed page, album page) to ensure it works everywhere

Total for User Story 1: 3 pts

User Story 2:

[5 pts] As a user, I would like to see an album of the week in my homepage so that I can get a recommendation.

1. Task: **Backend** Scraper: Get Top 5 Albums from Billboard[2 pts]
    a. Create a Node script/module (e.g., scripts/fetchBillboardTopAlbums.js or services/billboardService.ts).
    b. Fetch the HTML from https://www.billboard.com/charts/billboard-200/.
    c. Parse top 5 entries:
        i. album name
        ii. artist name
        iii. chart position (1–5)
    d. Save results in a weekly_charts or similar collection
    e. Add basic error handling + logging.
    f. Document how to run the script manually (e.g., npm run fetch:billboard).
        i. add a lazy “refresh if older than 7 days” later
2. Task: Spotify Matching & Enrichment[2 pts]
    a. Implement a matchBillboardAlbumToSpotify({ title, artist }) helper:
        i. Use Spotify Search endpoint: GET /v1/search?type=album&q=${title} ${artist}
    b. Define matching logic:
        i. Use the first result where type === "album" and artist name are similar (case-insensitive)
    c. For each Billboard album, store:
        i. spotifyAlbumId
        ii. spotifyUrl (external_urls.spotify)
        iii. imageUrl (largest or medium-size cover)
    d. Save enriched data back in DB so future API calls read from Mongo, not Spotify each time.
3. Task: **Frontend** - Display “Top Albums This Week” on Homepage[1 pts]
    a. Create useTopAlbumsWeek() hook that calls /api/feature/top-albums-week and exposes data, isLoading, isError.
    b. Build a <TopAlbumsSection /> component:
        i. Renders 5 album cards with cover, album name, artist, position badge (#1, #2, etc.)

Total for User Story 2: 5 pts

User Story 3:

[6 pts] As a moderator, I would like to be able to remove comments/posts of other users so I can keep the feed and comments clean from anything vulgar.

1. Task: Define backend user roles[1 pts]
    a. Create role field on user creation

    b. Create and set admin roles to allow moderation allocation
    c. Give moderators crud on all objects
    d. Default users to read only or create on objects
2. Task: Setup the backend to allow soft Delete/Patch on objects[3 pts]
    a. User role validation component
    b. Create Delete calls on comments, reviews, and feed objects
        i. Remove from feed with a deleted timestamp and deleted by with user id or username, and deletion reason
    c. Create Patch calls to edit a user's role to allow admins to allocate moderators
3. Task: Create a ui for moderators to cleanly delete objects and allocate other moderators[2 pts]
    a. Create user-based, specific ui to allow only moderators' visibility to moderation elements as a reusable;e components on all objects
    b. Delete button tied to objects integrating in the backend Delete calls on all objects
        i. Confirmation screen/check
        ii. Deletion reason field with default options like spam, harassment, and inappropriate content as a dropdown
    c. Promotion button on a users profile to elevate their permissions (set role to moderator)
        i. Demotion button to remove moderation roles, show the demotion reason and removed by admin user

Total for User Story 3: 6 pts

User Story 4:

[5 pts] (Replacing US4) As a user, I would like to receive personalized album recommendations based on the albums I've reviewed or liked, so I can discover new music that matches my taste.

1. Create a backend module to generate a "taste vector" for the user based on their activity.
2. Generate actual album recommendations using Spotify's recommendation system but adding a custom scoring system
3. Display recommended albums prominently on the homepage.

Total for User Story 4: 5 pts

ADDED TO BACKLOG, CAN FINISH THIS ONCE THE QUARTER IS OVER

User Story 5:

[5 pts] As a user, I would like to see upcoming concerts/performances for artists I've

reviewed or followed in my feed so I can check if there is a concert near me to go to of my favorite artist.

1. Task: Fetch upcoming concerts from Ticketmaster API[2 pts]
    a. Set up API integration with Ticketmaster to fetch events by artist name or Spotify artist ID.
    b. Handle authentication, API rate limits, and request errors.
    c. Normalize the event data for frontend use (event name, date, venue, city, ticket link).
2. Task: Build ConcertFeed component [2 pts]
    a. Create a React component that lists upcoming concerts for followed/reviewed artists.
    b. Include event details: artist, date, venue, city, and a link to buy tickets.
3. Task: Connect feed to user data[1 pts]
    a. Fetch the list of artists the user has reviewed or is following from MongoDB.
    b. Query Ticketmaster for events for all relevant artists and merge results.
    c. Handle empty states (no upcoming events) gracefully.

Total for User Story 5: 5 pts

Team Roles

Team Member	Role(s)
Adam Gonzales	Developer
Carter Wong	[Developer / Scrum Master]
Christopher Bocanegra	[Developer / Product Owner]
Srikar Chunduri	[Developer / Scrum Master / Product Owner]
Shawn Dhillon	[Developer]

Initial Task Assignments

Team Member	User Story	Initial Task
Carter Wong	User Story 6	[Task Description]
Srikar Chunduri	User Story 4	[Task Description]
Adam Gonzales	User Story 2	[Task Description]

Christopher Bocanegra	User Story 1, User Story 5 Task 2	Work on adding the buttons on user story 1 and the design for user story 5
Shawn Dhillon	User Story 3, User Story 4	

Initial Burnup Chart

Scrum Meeting Schedule

Day	Time	Type / Attendees
Mon	8 pm	[Team Scrum]
Wed	9 am	[Team Scrum]
Fri	9 am	[Team Scrum]
Mon	2 - 2:45 pm	[TA/Tutor visit]