

## Sprint 1 Plan

Product Name: Groovo

Team Name: Groovo

Sprint Completion Date: 10/20/2025

Revision Number: Rev. 1.0

Revision Date: 10/01/2025

## Sprint Goal

Deliver foundational authentication and navigation features that enable users to sign up, log in, and access the early homepage and profile structure. This sprint focuses on building the core login/signup flow, homepage routing, and initial profile + bio functionality to establish the platform's base user experience.

## Task Listing (Organized by User Story)

### User Story 1 (3 pts)

As a user, I want to be able to sign up and log in so I can access my own profile.

1. Task : Design & Implement Login/Signup UI [1 pt]
  - a. Create /login and /signup routes using Next.js App Router.
  - b. Add form fields: username/email, password, confirm password.
  - c. Add client-side validation (required fields, password length, matching passwords).
  - d. Add form submit button + loading states.
  - e. Add navigation links between "Login" and "Sign Up."
  - f. Style to match Groovo's design system.
2. Task: Backend Authentication (Signup) [1 pt]
  - a. Implement /api/signup route.
  - b. Create user schema/model (MongoDB).
  - c. Hash password before saving.
  - d. Validate unique username/email.
  - e. Return success or error messages.
3. Task: Backend Authentication (Login) [1 pt]
  - a. Implement /api/login route.
  - b. Validate user existence.
  - c. Compare hashed passwords.
  - d. Create session or JWT.
  - e. Redirect to homepage on success.

## **User Story 2 (8 pts)**

As a user, I want to see a landing page so that I can navigate the web page and see what Groovo has to offer.

1. Task: Create Landing Page Route + Layout [1 pt]
  - a. Create the root route using [next.js](#) app router
  - b. Define the main page structure with sections for featured albums, and navigation links
2. Task: Build Global Navigation Header [1 pt]
  - a. Create a reusable header component that appears on the landing page and other main views
  - b. Add nav links for core pages such as home, browse/discover/profile and login/signup
3. Task: add a Popular Albums Preview in home page [2 pts]
  - a. Add a popular albums grid below that displays several album cards with spotify API
4. Task: Search Bar Functionality [2 pt]
  - a. Add a search bar component to the landing page
  - b. Allow users to type album or artist names into the search input and handle the submit event
  - c. Style the search bar to match existing UI
5. Task: Backend Mock API Endpoints [2 pts]
  - a. GET /api/trending → popular albums (mock response now; real Spotify later) (1)
  - b. GET /api/feature/album-of-week → single featured album (mock now; later curated/logic) (1)

## **User Story 3:**

[6 pts] As a user, I want to be able to write a bio and change my name so that I can customize my profile and express myself.

1. Task: Implement a user bio field in MongoDB — (1 pts)
  - a. Update the users collection to include a bio field.
2. Task: Build API profile update route — (1 pts)
  - a. Create an authenticated API route that allows users to update their bio and basic profile information (e.g., bio, display name, etc.).
3. Task: Implement bio input and edit functionality— (2 pts)

- a. Add an “edit profile” button to the user’s profile page
  - b. Create a form or modal allowing users to input or update their profile
  - c. Validate input length(e.g., 0-200 characters) and prevent empty submissions
  - d. Manage form state with TypeScript types(Controlled components in react/next.js?)
4. Task: Connect bio editing to backend API— (2 pts)
- a. Integrate frontend form with backend endpoints(eg, Put command)
  - b. Send updated bio via JSON body and update the MongoDB user doc through Express
  - c. Display Success/error messages to the user
  - d. Ensure changes reflect immediately on the profile page (maybe react state or refetch user data)

### **User Story 4:**

[4 pts] As a user, I want to view my profile page displaying saved albums and reviews so that I can look back at them later when I forget which albums that I listened to

1. Task : Create the profile page layout: — (1 pts)
  - a. Set up a new route(eg. /profile) and corresponding React page
  - b. Add a consistent layout(eg, header, navigation bar, main content area) from the previous pages we have created(eg. color palette, spacing.
2. Task : Implement the User profile Information section — (1 pts)
  - a. Show basic profile information such as the username, profile picture, and short bio
  - b. Create a reusable UserHeader component for this section
  - c. Handle cases where no profile picture or bio is available
3. Task: Fetch and display saved album/reviews — (2 pts)
  - a. Integrate the frontend with the backend api endpoint
  - b. Fetch and render a grid/list of saved albums and reviews(or mockup data for now?)
  - c. Display album artwork, title, artist, and a short review snippet
  - d. Implement loading and error states for data fetching

### **Team Roles**

Adam Gonzales – Developer

Carter Wong – Developer / Scrum Master

Christopher Bocanegra – Developer / Product Owner

Srikar Chunduri – Developer

Shawn Dhillon – Developer

### **Initial Task Assignments**

Srikar – User Story 1

Shawn – User Story 1

Christopher – User Story 2

Adam – User Story 2

Carter – User Story 2

### **Scrum Meeting Schedule**

Monday – 8 PM – Team Scrum

Friday – 2 - 2:45 PM – Team Scrum + TA/Tutor

Sprint Burn up Chart

Burnup Chart - Sprint 1, Groovo

