

Sprint Review (Retrospective) Meeting

4/11/2024 – 17/11/2024

Group ID: **02**

Project Name: Beatify

Prepared by: **Nguyễn Duy Bảo**

Date: 16/11/2024

Attendees:

22127152 – **Lê Gia Huy**

22127114 – **Phạm Hà Hiếu**

22127115 – **Trần Trung Hiếu**

22127203 – **Võ Ngọc Khoa**

22127031 – **Nguyễn Duy Bảo**

Sprint Outcomes

1. What Went Well

- Completed the authentication pages, playlist routes, and homepage layouts made the platform more polished and functional.
- Reviewing and applying design feedback ensured the platform looks cohesive and professional, with a consistent layout across pages.
- Implementing song playing functionality giving users an engaging and interactive music experience.

2. What Went Wrong

- When trying to reuse these layouts in other areas, they broke down, leading to unexpected bugs and frustrating behavior.
- Fixing frontend bugs ate into the time that could have been spent on new features or polishing existing ones.
- Curators managing songs ran into edge cases that hadn't been accounted for during the initial design of the routes, which delayed testing and debugging.
- Developing the playback feature had its hurdles, especially around managing state for "previous," "next," and "shuffle" functionalities.

3. Problem Analysis

- The issue lies in the lack of foresight when developing components. Building them to be reusable from the start would have avoided these bugs.
- Relying on quick fixes during the initial development stage led to technical debt, which had to be paid off with time-consuming debugging.

4. Improvements

- Start designing components with reusability in mind. Modular components can be used across different parts of the app without breaking.
- Use shared libraries or centralized design systems to make layouts more consistent and easier to apply.
- Create a set of guidelines for developing and reusing components so everyone on the team is on the same page.
- Begin testing new features and integrations earlier to catch bugs before they escalate into larger issues.

5. Lessons Learned

- Taking the time to develop reusable components and features upfront can save hours of debugging and redesign later.
- Regular reviews and feedback loops are essential to keeping the platform cohesive and professional.
- Features like playback functionality require more thorough planning to avoid performance or state management issues.