**Project pitch**

***HeardIT***

|  |
| --- |
| **Date : 03/03/2024** |
| **Author : Mihail Vasilev** |

**Project description**

HeardIT is a music sharing platform where you will be able to discover and listen and learn how to play to your favorite songs from a plethora of indie bands. The website focuses on small to medium music creators and people interested in learning how to play the songs on their instruments. The application provides several features that allow users to have an interactive experience combining the listening and learning aspect in one easy-to-find place.

Songwriters will have the ability to upload their tracks, the chords/tabs, lyrics and any other information that will allow their fans to be able to not only listen to the songs but also learn to play them.

Fans will be able to experience these features easily through the interactive user interface and will be able to leave comments, likes and make playlists with their favorite songs.

This way HeardIT will be a place where music creators and fans will have the opportunity to interact, share their experience, learn and get closer to the aft of music together.

**Features and Challenges**

In this section I will explain the features of the application, the priority of each feature and the challenges that will arise during the implementation of each feature. The features will be created in order of priority with the highest one starting from **1.**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Challenges** | **Priority score** |
| Song streaming | No experience in streaming services | **1** |
| Account management   * Create * Update * Login and logout * User types – creator, admin, fan | GDPR security issues | **1** |
| Upload/Delete tracks | Transfer and storage of large files | **1** |
| Upload extra information   * Chords/tabs * Lyrics * Extra information |  | **2** |
| Advanced search |  | **2** |
| Comments section and likes |  | **3** |
| Playlist creation |  | **3** |
| Statistics |  | **4** |

**Non-functional requirements**

1. Performance
   * Response time:
   * Scalability:
   * Load Handling:
2. Reliability
   * Uptime:
   * Fault tolerance:
   * Data management:
3. Security
   * Data encryption
   * Authorization and authentication
4. Scalability
   * Database scalability:
   * User number growth:
5. Usability
   * UI design:
6. Maintainability
   * Code maintainability:
   * Scalable architecture:
7. Testing:
   * Comprehensive testing:
   * Load testing:

**Technologies**