**Listen Up**

**Research Document**

Mohammad Nazibul Kabir Khan

4263308

Contents

[Introduction 3](#_Toc104989158)

[Research Questions 4](#_Toc104989159)

[Problem: Data to be stored 4](#_Toc104989160)

[Major Question: What data should the software store? 4](#_Toc104989161)

[Minor Questions 4](#_Toc104989162)

[1: What user data should be stored? 4](#_Toc104989163)

[2: What details of main component need to be stored? 4](#_Toc104989164)

[3: How to determine relevancy of the data stored? 4](#_Toc104989165)

[Problem: Quality of the code 5](#_Toc104989166)

[Major Question: How to determine quality of code? 5](#_Toc104989167)

[Minor Questions 5](#_Toc104989168)

[1: What is SonarQube? 5](#_Toc104989169)

[2: How to use SonarQube? 5](#_Toc104989170)

[3: How to connect SonarQube with git using yaml file? 5](#_Toc104989171)

[Problem: Future of this application and improvements it needs 6](#_Toc104989172)

[Major Question: What is the future of the application? 6](#_Toc104989173)

[Minor Questions 6](#_Toc104989174)

[1: How this application is better than its competitor? 6](#_Toc104989175)

[2: How can this application be extended? 6](#_Toc104989176)

[Problem: Coping up with new knowledge and platform. 7](#_Toc104989177)

[Major Question: What are new things that was learned? 7](#_Toc104989178)

[Minor Questions 7](#_Toc104989179)

[1: How to use React and connect the front-end to back-end? 7](#_Toc104989180)

[2: What are the functionalities of JavaScript that are important to learn? 7](#_Toc104989181)

[3: How to use axios? 7](#_Toc104989182)

[4: What are DTOs? 7](#_Toc104989183)

[5: Why DTOs? 7](#_Toc104989184)

[6: How to convert DTO to model and vice versa? 7](#_Toc104989185)

[Conclusion 8](#_Toc104989186)

# Introduction

This document is about the research done to get answers of list of questions. These questions are related to the understanding new things and getting new ideas to be used in ListenUp project.

# Research Questions

## Problem: Data to be stored

### Major Question: What data should the software store?

First thing to worry about when creating a software solution is what data it should store. At the beginning of the creation of website, questions like these arises: *‘What data should be stored for making a secure environment?’, ‘What data is relevant to the website?’* and more. After doing **competitive analysis** on Spotify some information about the data collection is gathered. For user personal information, ListenUp website will store their email and username. And for high security encoded version of their password will also be stored. User will be able to login with their email and password.

Other important data should be collected to make website function relevantly. ListenUp is a website about songs. People can make playlist and add songs in their playlists. Again, **competitive analysis** is done on Spotify and **user requirements are explored** to get what data should be stored. ListenUp is storing all song info and the info of playlist that were made.

After **exploring user requirements** and going through **data analytics** we are validating relevancy of data stored.

#### Minor Questions

##### 1: What user data should be stored?

Methods:

* Competitive analysis
* Explore user requirements
* Security test

When users are told to make an account at the startup of ListenUp website. User are told to put in their username, email, and password. Email and encoded version of the password is stored to make user’s environment secure and restricts access to act on user’s playlists and liked songs. Conclusion is made after doing **competitive analysis** to any other website with login requirements. After **security test**, encoded password is stored in database rather than the raw password to protect against hackers.

##### 2: What details of main component need to be stored?

Methods:

* Competitive analysis
* Explore user requirements
* Data Analytics

Currently software is storing song info data, artist info data, user info data(incudes their playlists, liked playlists and liked songs).

##### 3: How to determine relevancy of the data stored?

Methods:

* Competitive analysis
* Explore user requirements
* Data Analytics

Data usage is checked throughout the whole application.

## Problem: Quality of the code

### Major Question: How to determine quality of code?

#### Minor Questions

##### 1: What is SonarQube?

Method: Literature study

SonarQube is a software system which supplies with automated testing of bugs, code smell and quality of codes.

##### 2: How to use SonarQube?

Method: Literature study

Going through slides and researching through Google.

##### 3: How to connect SonarQube with git using yaml file?

Method: Literature study

The localhost:9000(SonarQube) has the tutorial for it.

## Problem: Future of this application and improvements it needs

### Major Question: What is the future of the application?

#### Minor Questions

##### 1: How this application is better than its competitor?

Methods:

* Benchmark test
* Available product analysis

It is not.

##### 2: How can this application be extended?

Methods:

* Brainstorm
* Explore user requirements

Currently application can only do simple crud functionalities. In near future it will be able to perform interesting actions.

## Problem: Coping up with new knowledge and platform.

### Major Question: What are new things that was learned?

#### Minor Questions

##### 1: How to use React and connect the front-end to back-end?

Methods:

* Literature study
* Other people’s code review

##### 2: What are the functionalities of JavaScript that are important to learn?

Method: Literature study

Researched the whole W3School. Currently important to learn how to operate with arrays and some of the basic functionalities

##### 3: How to use axios?

Method: Literature study

Going through demos and samples of code.

##### 4: What are DTOs?

Method: Literature study

##### 5: Why DTOs?

Method: Literature study

Increases security and reduces data usage.

##### 6: How to convert DTO to model and vice versa?

Method: Literature study

Going through demos and samples of code.

# Conclusion

…….