

# MUSIC RECOMMENDER SYSTEM

*A Report submitted in  
The partial fulfillment for the degree of*

**B. Tech.**

**Computer Science and Engineering**

**By**

**Vinay Panwar**

**Vaishnavi Kaushal**

**Rupali Wanare**

**Pursued in**

**Institute of Engineering and Technology**



The logo of Sage University Indore is a red shield with a yellow border. Inside the shield, there is a yellow flame-like symbol at the top center, flanked by two yellow stars. Below the flame, the word "SAGE" is written in large, bold, yellow letters. Underneath "SAGE", the word "UNIVERSITY" is written in smaller, white, bold letters. At the bottom of the shield, the word "INDORE" is written in yellow. A yellow banner with a red border is draped across the bottom of the shield, containing the text "Sage University Indore" in red.

# **MUSIC RECOMMENDER SYSTEM**

## CERTIFICATE

This is to certify that the project report entitled **Music Recommender System** submitted by **Vinay Panwar, Vaishnavi Kaushal, Rupali wanare** to the INSTITUTE OF ENGINEERING AND TECHNOLOGY, INDORE, in partial fulfillment for the award of the degree of **B. Tech in (COMPUTER SCIENCE AND ENGINEERING)** is a *bona fide* record of project work carried out by him/her under my/our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree or diploma.

<Signature>

Dr. Sachin Patel

Supervisor

Institute of Engineering and Technology

<Signature>

Prof. Sadhna Pandey

Co-supervisor

Institute of Engineering and Technology

Counter signature of HOD with seal

## DECLARATION

I declare that this project report titled **Music Recommender System** submitted in partial fulfillment of the degree of **B. Tech in (COMPUTER SCIENCE ENGINEERING)** is a record of original work carried out by me under the supervision of Dr. Sachin Patel, and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Date

Signature

Vinay Panwar

Vaishnavi Kaushal

Rupali Wanare

## ACKNOWLEDGEMENTS

On this great occasion of accomplishment of our project on the Music Recommender System, we would like to sincerely express our gratitude to Dr. Sachin Patel, who has been supported through the completion of this project.

We would also be thankful to our HOD Dr. Rajat Bhandari of the Institute of Engineering and Technology, Sage University Indore, for providing all the required facilities to complete this project.

I take this opportunity to thank Dr. Sanjeev Agarwal Director – of SAGE UNIVERSITY, Dean –Of students Activities, and other faculty members who helped in preparing the guidelines.

Finally, as one of the team members, I would like to thank all my group members for their support and coordination, I hope we will achieve more in our future endeavors.

Signature

Vinay Panwar

Vaishnavi Kaushal

Rupali Wanare

## ABSTRACT

The Recommender System is a system that provides customers with more reliable and friendly results ( in this case it's a song). We learned about many recommender systems and their working and found out how most recommender system which provides the friendliest result. We used the traditional method with some improvement and try to build a system which generate more profound result.

As the recommender systems were the wide area to start, so we choose more sorted form of result which was song. Music recommendation also included many genres of songs like Hindi songs or English songs, so we preferred Hindi songs, and created a system which gives a list of songs based on some input parameter.

For more user-friendliness, we used feedbacks to provide user more accurate service. We are working to build more upgrade in our system including with movies, books, and more products to recommend.

## LIST OF TABLES

<b>S.no.</b>	<b>Title</b>	<b>Page No.</b>
1	Hardware Requirements	10
2	Software Requirements	10

## Table of contents

CHAPTER 1: INTRODUCTION.....	9
CHAPTER 2: SPECIFICATION REQUIREMENTS.....	10
2.1 HARDWARE REQUIREMENTS.....	10
2.2 SOFTWARE REQUIREMENTS.....	10
CHAPTER 3: TECHNOLOGY USED.....	11
3.1 PYTHON MODEULE .....	11
3.1.1 TKINTER .....	11
3.1.2 PANDAS .....	11
3.1.3 PYGLET .....	11-12
3.1.4 TIME.....	12
3.2 API .....	12
3.2.1 SPOTIPY .....	12
3.3 VS-CODE .....	13
CHPATER 4: SDLC MODEL.....	14
4.1 USED MODEL.....	14
CHAPTER 5: FLOW DIAGRAM.....	15
CHAPTER 6: FUNCTIONAL SPECIFICATION.....	16
6.1 USER MODULE.....	16
CHAPTER 7: UI .....	17-19
CHAPTER 8: TESTING.....	20
8.1 SOFTWARE TESTING.....	20
8.2 STRATEGIES .....	20
8.2.1 UNIT TESTING.....	20
8.2.2 INTEGRATION TESTING.....	20-21
8.2.3 SYSTEM TESTING.....	21
8.2.4 ACCEPTANCE TESTING.....	22
CHAPTER 9: FUTURE UPGRADES .....	23
CHAPTER 10: CONCLUSION .....	24
CHAPTER 11: REFERENCES .....	25



## CHAPTER 1: INTRODUCTION

The Recommender system is a system that helps an algorithm or device to guess something for their respective users. There are many different types of recommender systems available in the tech industry and some of them are working positively as others are in the process of continuing to innovate to customize the product more and more.

With this idea in our mind, we choose to work on a recommender system that helps the user to understand their day-to-day requirements and make their decisions with clearer information in mind. As we all knew that recommender would be a very deep field to work in as today's user requirements and daily life activities are much more complex than we looked at first when we choose to work on all the areas. But after the realization of the problem, we choose to work with the music recommendation system.

How are these music applications recommending songs to their user, we got to learn, this is all based on the user's mood. I mean most of the time user is in a happy mood or natural mood or could be angry. There was so much vast majority of possibilities, so we tried to modularize them into small tasks(modules) we divided the task into three major phases and worked with and will continue to work on it.

Our problem was so basic to recommend any kind of music to a user but which song to refer which user at first looked much more complicated.

This model/UI Help user understand their moods and also let them decide which song they want to listen to feel their emotions.

## CHAPTER 2: SPECIFICATION REQUIREMENTS

Requirement are the environments , services and technologies which were used to build the system. There are two types of requirements need to develop Music recommender system.

### 2.1 HARDWARE REQUIREMENTS

Hardware requirements include all the device which were get used in process of developing the recommender system. This include such as storage, processing unit, display, and audio speaker.

Requirements: Laptop or pc with below configuration	
RAM	8GB
ROM	1TB
PROCESSOR	i3 or successor
SPEAKER	Dual
Display	720p display or successor
Touch	Enable
Keyboard and mouse or touchpad	standard
Processor speed	2.5 ghtz

### 2.2 SOFTWARE REQUIREMENTS

Software requirement includes all the service which were requirement to develop an running UI, which made the whole difference in the process of development. There were some modules, and api required to develop the entire system.

Software requirements	
Python module	Tkinter, file, pandas, random, Spotipy
API	Spotipy
IDEs	Visual studio code, jupyter notebook, terminal environment
Access	Spotify
Application for documentation	PowerPoint, Word

## CHAPTER 3: TECHNOLOGY USED

Many modern technology and practices had been used to develop a well functioning software that helps the people day routine become more crystal clear.

### 3.1 PYTHON MODULES

A python module is a file containing python definitions and statements. A module can define functions, classes, and variables. Grouping related code into a modules makes the code easier to understand and use. There are many different modules available in python i.e. numpy, pandas, matplotlib, scikit-learn, seaborn.

We used most of the modules of python while developing the system.

#### 3.1.1 TKINTER

Tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with Tkinter is the fastest and easiest way to create GUI applications. Creating a GUI using Tkinter is an easy task.

To create a Tkinter GUI :-

- a. Import Tkinter module using (import Tkinter)
- b. Create the main window (via creating reference)
- c. Add content to GUI
- d. Apply trigger events.

#### 3.1.2 PANDAS

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

#### 3.1.3 PYGLET

**Pyglet** is easy to use but powerful library for developing visually rich GUI applications like games, multimedia, etc on Windows, Mac OS, and Linux. This library is created purely in Python and it supports many features like

windowing, user interface event handling, Joysticks, OpenGL graphics, loading images, and videos, and playing sounds and music.

We used **pyglet** for playing songs.

### 3.1.4 TIME

This module provides various time-related functions. For related functionality, see also the `DateTime` and `calendar` modules. The Python time module provides many ways of representing time in code, such as objects, numbers, and strings. It also provides functionality other than representing time, like waiting during code execution and measuring the efficiency of your code.

To use all the module's you can use the command written below

Pip install module-name

## 3.2 API

APIs are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols. For example, the weather bureau's software system contains daily weather data. The weather app on your phone "talks" to this system via APIs and shows you daily weather updates on your phone.

API stands for Application Programming Interface. In the context of APIs, the word Application refers to any software with a distinct function. The interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses. Their API documentation contains information on how developers are to structure those requests and responses.

API architecture is usually explained in terms of client and server. The application sending the request is called the client, and the application sending the response is called the server. So in the weather example, the bureau's weather database is the server, and the mobile app is the client.

### 3.2.1 SPOTIPY

Spotipy is a lightweight Python library for the Spotify Web API. With *Spotipy* you get full access to all of the music data provided by the Spotify platform. Our API is what is commonly known as a RESTful API. It is an interface that programs can use to retrieve and manage Spotify data over the internet. The Web API uses the same HTTP protocol that's used by every internet browser.

### 3.3 VS-CODE

Visual Studio Code is a free, lightweight but powerful source code editor that runs on your desktop and the web and is available for Windows, macOS, Linux, and Raspberry Pi OS. It comes with built-in support for JavaScript, TypeScript, and Node.js and has a rich ecosystem of extensions for other programming languages (such as C++, C#, Java, Python, PHP, and Go), runtimes (such as .NET and Unity), environments (such as Docker and Kubernetes), and clouds (such as Amazon Web Services, Microsoft Azure, and Google Cloud Platform).

Aside from the whole idea of being lightweight and starting quickly, Visual Studio Code has IntelliSense code completion for variables, methods, and imported modules; graphical debugging; linting, multi-cursor editing, parameter hints, and other powerful editing features; snazzy code navigation and refactoring; and built-in source code control including Git support. Much of this was adapted from Visual Studio technology.





## CHAPTER 4: SDLC MODEL

The Software Development Life Cycle (SDLC) is a structured process that enables the production of high-quality, low-cost software, in the shortest possible production time. The goal of the SDLC is to produce superior software that meets and exceeds all customer expectations and demands. The SDLC defines and outlines a detailed plan with stages, or phases, that each encompasses its own process and deliverables. Adherence to the SDLC enhances development speed and minimizes project risks and costs associated with alternative methods of production.

### 4.1 USED MODEL

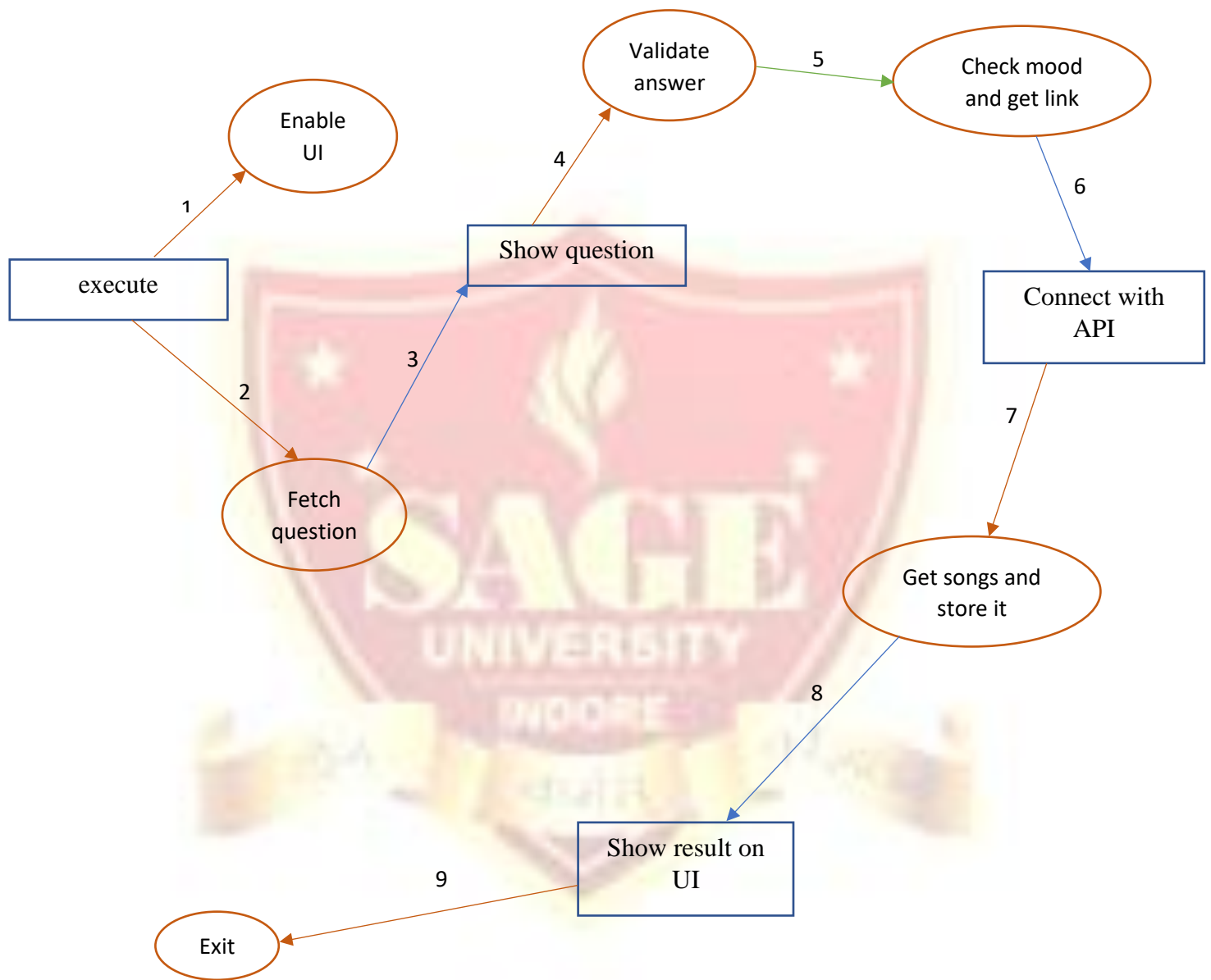
We decided to use the Agile strategy of Software development as we knew this was the only software development life cycle process that deals with ongoing adaption and accepts new changes in the process and product. We are a student and don't know at which point of time we get to update the process or are in process. We started with the identification of the problem statement to actually finding the problem which we were going to solve and help innovate the solution. We came up with a recommendation system problem but even then that was too large to deal with so we decided to move with the music recommendation system. After deciding the problem we started to work on solutions and came up with numerous solutions like an application or web app but in the end, we decided to build a UI for users to simplify their decision tree at some level in life.

The Agile model helped us adapt more changes, and gain more insight into widget board.

The agile SDLC model is a combination of iterative and incremental process models with a focus on process adaptability and customer satisfaction by rapid delivery of working software products. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross-functional teams working simultaneously on various areas –

1. Planning
2. Requirements Analysis
3. Design
4. Coding
5. Unit Testing and
6. Acceptance Testing.

## CHAPTER 6: FLOW DIAGRAM



## CHAPTER 6: FUNCTIONAL SPECIFICATION

A functional specification is a formal document used to describe a product's intended capabilities, appearance, and interactions with users in detail for software developers. The functional specification is a kind of guideline and continuing reference point as the developers write the programming code.

The method of preparing the specifications before the product is known as the "write the manual first" approach, serving as an outline of the finished program.

Typically, the functional specification for an application program with a series of interactive windows and dialogs with a user would show the visual appearance of the user interface (UI) and describe each of the possible user input actions and the program response actions.

A functional specification may also contain formal descriptions of user tasks, dependencies on other products, and usability criteria. Many companies have guides for developers that describe what topics any product's functional specification should contain.

### 6.1 USER MODULE

The **Music Recommender system** is a GUI application that uses web-based API to provide more reliable results. This needs Only the End-user to use the GUI. This model doesn't need a standalone account for this UI, it only gets to execute to use the UI.

Spotify accounts would work for the UI to get more customized and reliable results.

### END USER MODULE

This module is of end user who are going to interact with the system. End user is the one who use the software to better help them at decision making.



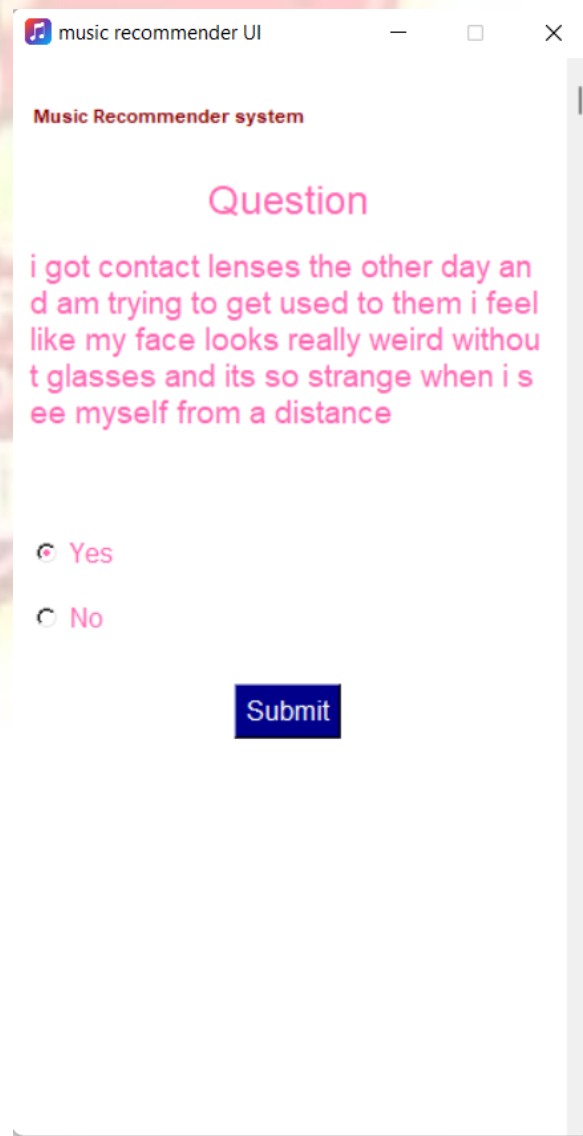
## CHAPTER 7: UI (USER INTERFACE)

The user interface is the point at which human users interact with a computer, website, or application. The goal of effective UI is to make the user's experience easy and intuitive, requiring minimum effort on the user's part to receive the maximum desired outcome.

UI is the layer by which users interact with the model (software), it includes all device through by user interact with the system such as mouse, keyboard, touchpad, display, monitor, and other form of ways.

As a system UI, right now we have 3 distinct window on UI, this windows are named as Home window, suggestion window, and play window.

### Home window

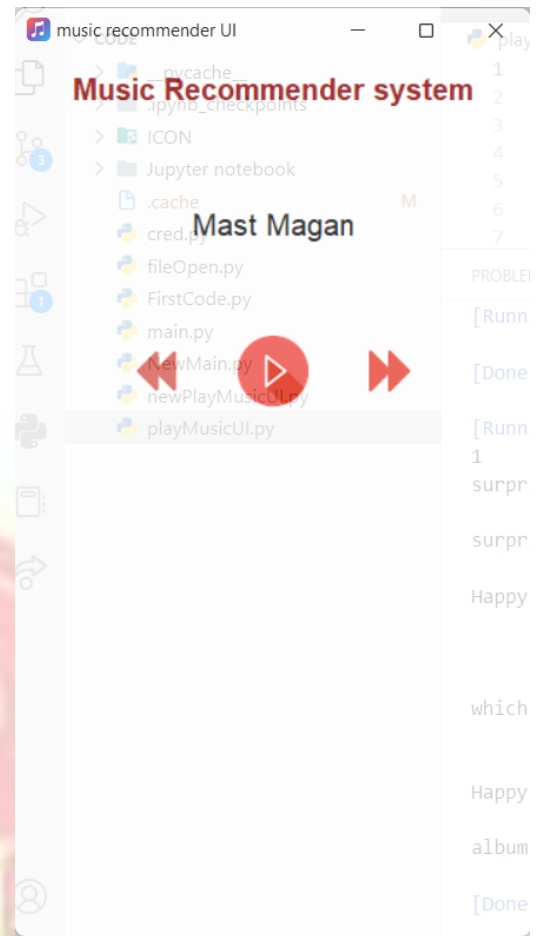


The screenshot shows a web browser window titled "music recommender UI". Inside the window, the text "Music Recommender system" is displayed at the top. Below it, a question is presented in pink text: "i got contact lenses the other day and am trying to get used to them i feel like my face looks really weird without glasses and its so strange when i see myself from a distance". At the bottom of the question, there are two radio button options: "Yes" and "No". A blue "Submit" button is located at the bottom right of the form.

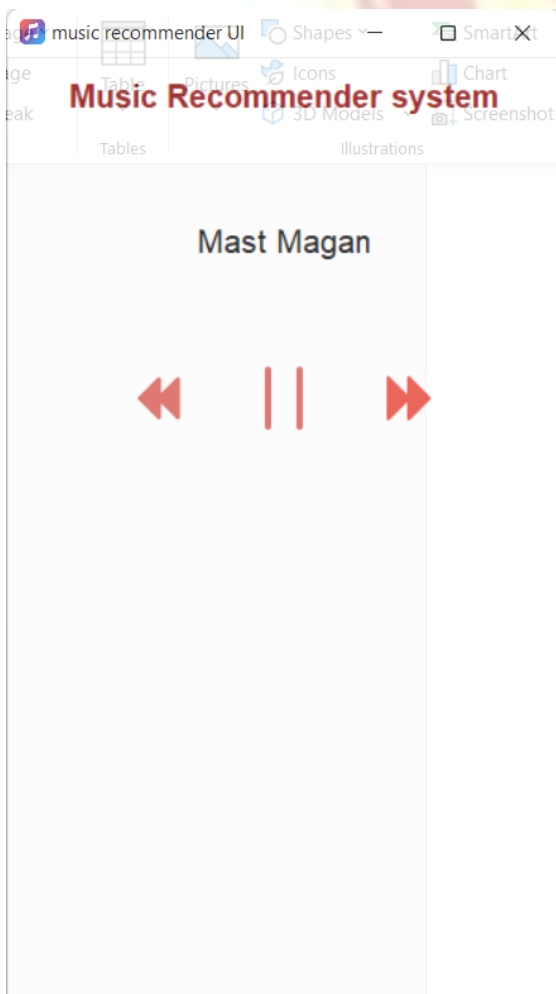


**Suggestion window**

## Play window



## Next window



## CHAPTER 8: TESTING

Testing is an approach to verify and validate whether the product is in well condition to use or not. This ensures users, the product is of good quality and, use it without any worry.

### 8.1 SOFTWARE TESTING

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that the software product is Defect free. It involves the execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps, or missing requirements in contrast to actual requirements.

### 8.2 STRATEGIES

Testing strategies include all the approaches and practices which are used to ensure the software and ensure the quality of it. There are multiple practices available to ensure the quality of Software.

#### 8.2.1 UNIT TESTING

A unit test is a way of testing a unit - the smallest piece of code that can be logically isolated in a system. In most programming languages, that is a function, a subroutine, a method, or a property. The isolated part of the definition is important.

Unit testing is done in three ways.

1. Black box testing → This testing technique is used in covering the unit tests for input, user interface, and output parts.
2. White box testing → This technique is used in testing the functional behavior of the system by giving the input and checking the functionality output including the internal design structure and code of the modules.
3. Gray box testing → This technique is used in executing the relevant test cases, test methods, and test functions, and analyzing the code performance for the modules.

#### 8.2.2 INTEGRATION TESTING

Integration testing is the second level of the software testing process that comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

Unit testing uses modules for testing purposes, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.

This level of testing can be achieved via different approaches some of them are :-

1. **Top-down approach** → The top-down approach holds that the top-level module of the software must go through the testing process before any other modules can be tested. For a top-down approach to be effected, the top-level part of the module is first fully developed. After the completion of development, the testing of the top level is carried out. Right after the completion of the testing, the sub-module of the software is merged with the top-level module which has completed its testing.
2. **Mixed approach** → In the mixed integration approach, there is no preference for either the top-level module or the bottom-level module. The integration can be carried out at any point in time, provided the modules have gone through the development process.
3. **Bottom-up approach** → For the bottom-up approach, preference is given to the sub-modules. They are first developed and go through testing before any other module in the system. The other part of the software can follow through with its testing just after the sub-module testing is completed.
4. **Big bang approach** → In the big bang approach, every module is integrated as its development is complete. So, at any point in time, any complete module can be integrated into the whole software system. This is a very risky approach, as there could be setbacks upon error discovery during the testing process. The setback would occur due to the difficulty of discovering which module, in particular, holds this error, and it could prove costly to trace the error and delay the software development process.

### 8.2.3 SYSTEM TESTING

System Testing is a type of software testing that is performed on a completely integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested.

Entire system can be tested via various approaches, some of them are as below.

1. **Performance testing** → Performance Testing is a type of software testing that is carried out to test the speed, scalability, stability, and reliability of the software product or application.
2. **Load Testing** → Load Testing is a type of software Testing which is carried out to determine the behavior of a system or software product under extreme load.
3. **Stress Testing** → Stress Testing is a type of software testing performed to check the robustness of the system under varying loads.
4. **Scalability Testing** → Scalability Testing is a type of software testing which is carried out to check the performance of a software application or system in terms of its capability to scale up or scale down the number of user request loads.

### 8.2.4 ACCEPTANCE TESTING

Performance Testing is a type of software testing that ensures software applications perform properly under their expected workload. It is a testing technique carried out to determine system performance in terms of sensitivity, reactivity, and stability under a particular workload.

There are many ways to ensure that an application is performing well or not, Some of them are.....

1. **Load Testing** → It checks the product's ability to perform under anticipated user loads. The objective is to identify performance congestion before the software product is launched in the market.
2. **Stress Testing** → It involves testing a product under extreme workloads to see whether it handles high traffic or not. The objective is to identify the breaking point of a software product.
3. **Endurance Testing** → It is performed to ensure the software can handle the expected load over a long period of time.
4. **Volume Testing** → In volume testing, a large number of data is saved in a database and the overall software system's behavior is observed. The objective is to check the product's performance under varying database volumes.
5. **Scalability Testing** → In scalability testing, software applications' effectiveness is determined by scaling up to support an increase in user load. It helps in planning capacity addition to your software system.
6. **Spike Testing** → It tests the product's reaction to sudden large spikes in the load generated by users.



## CHAPTER 9: FUTURE UPGRADES

We are looking forward to enhancing the suggestion more customizable, and also including more suggestible services such as movies, books, and products.

Enhancing the software workload in terms of users is our main priority.

We are working more on the UI part of the system to make the software friendly to users and easily usable by anyone. As we have a pretty good idea about the appearance of software is everything when it comes to users.

We also made a git repository so everyone can use it and help us with more upgrades in it.

The reference is given in the reference section.



## CHAPTER 10: CONCLUSION

Recommending systems are more user reliable as it recommends more customized result. Customized results mean more familiarity and satisfaction to users.

As per the growing needs of user and lack of time, it requires our more effort to give the optimal solutions for the problems. Our team created a widget board which help users to get more insightful brief about their recommendation and customized result. We are working on widget project which help user to get a recommendation for movies, music, and books. Currently we worked only for music recommendation.





## CHAPTER 11: REFERENCES

We referenced many sites and books to get help to create more customized and user-reliable results.

Here are the list of web.

1. [www.google.com](http://www.google.com)
2. <https://www.geeksforgeeks.org/performance-testing-software-testing/>
3. <https://www.geeksforgeeks.org/system-testing/>
4. <https://www.guru99.com/software-testing-introduction-importance.html>
5. <https://app.grammarly.com/ddocs/1723435934>
6. <https://spotipy.readthedocs.io/en/2.21.0/>
7. <https://aws.amazon.com/what-is/api/>
8. <https://docs.python.org/3/library/time.html>
9. <https://www.geeksforgeeks.org/python-time-module/>
10. <https://www.geeksforgeeks.org/introduction-to-pyglet-library-for-game-development-in-python/>
11. <https://www.geeksforgeeks.org/introduction-to-pandas-in-python/>
12. <https://developer.spotify.com/dashboard/login>

Here is the github repo reference to use it.

<https://github.com/vinay-panwar/Major-Project-I/tree/master>