

PRESENTATION

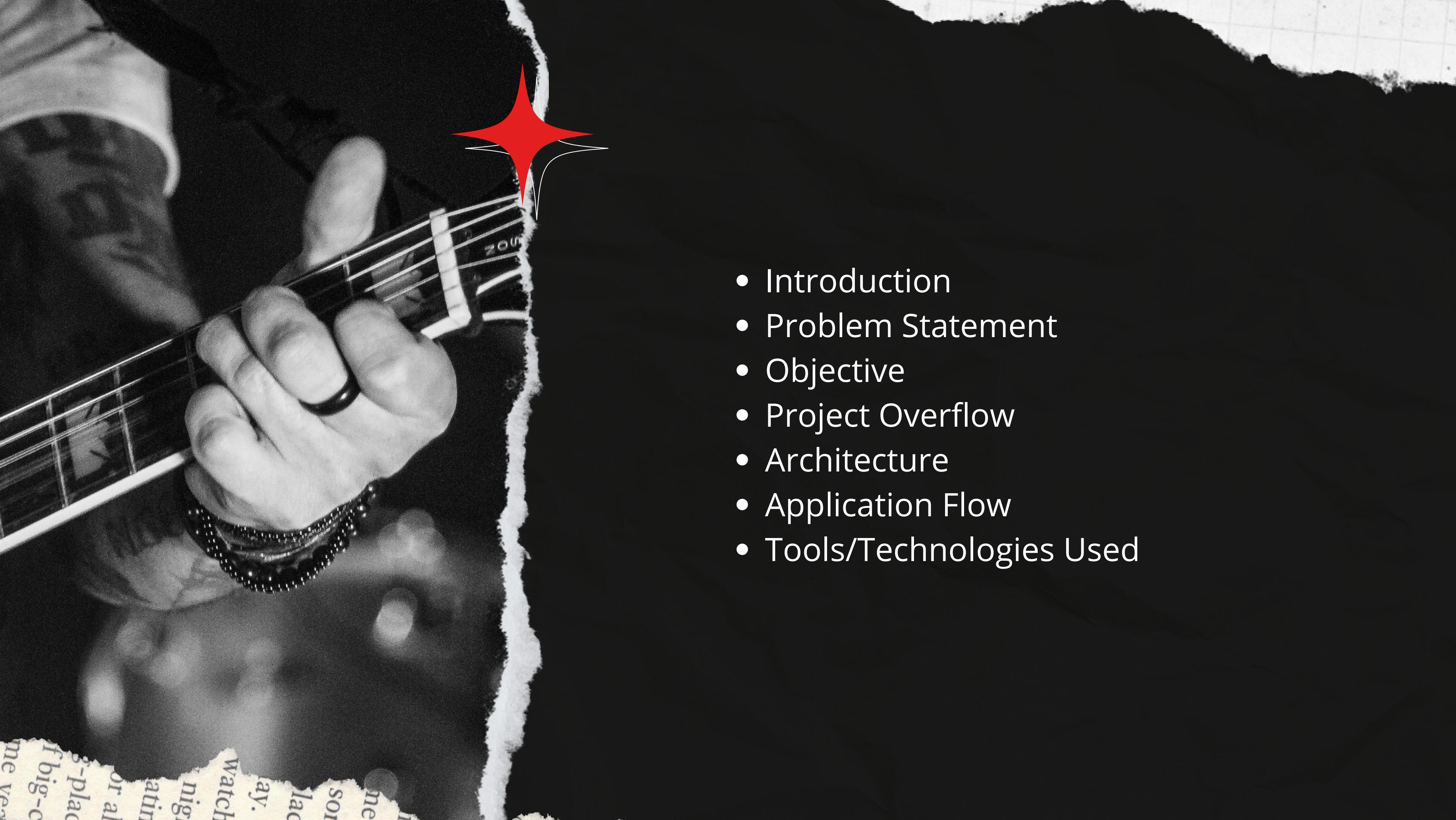
MUZIX APP

B A T C H C

OUR TEAM MEMBERS

- Ankit Kumar
- Ravindra Billakurthi
- Shaahid
- Chinnam Sowmya





- Introduction
- Problem Statement
- Objective
- Project Overflow
- Architecture
- Application Flow
- Tools/Technologies Used



INTRODUCTION

1

In this globalized world, music becomes essential, which means there's a demand for apps that provide old music and new music

3

Customer must have a better interface and ease of access through out the app and choose their desired music

2

Music plays an important role in day today's life. So we designed a safe, Efficient, Reliable application.

4

The better way of music app with lots of facilities like adding music to a new playlists, bookmarking your favourite songs and in a easier manner

PROBLEM STATEMENT

- Creating a server in Spring Boot to facilitate registration and login with MySQL as backend.
Upon login, JWT token has to be generated.
- User should be able to:
 - Search music
 - Bookmark music
 - Create playlists
- Account Manager should be able to manage user accounts.



OBJECTIVE

LOGIN

Our main aim is to provide best User Experience

So that User can log in to the app and can access their own playlists and bookmarked songs from our app by their choice

SIGN UP

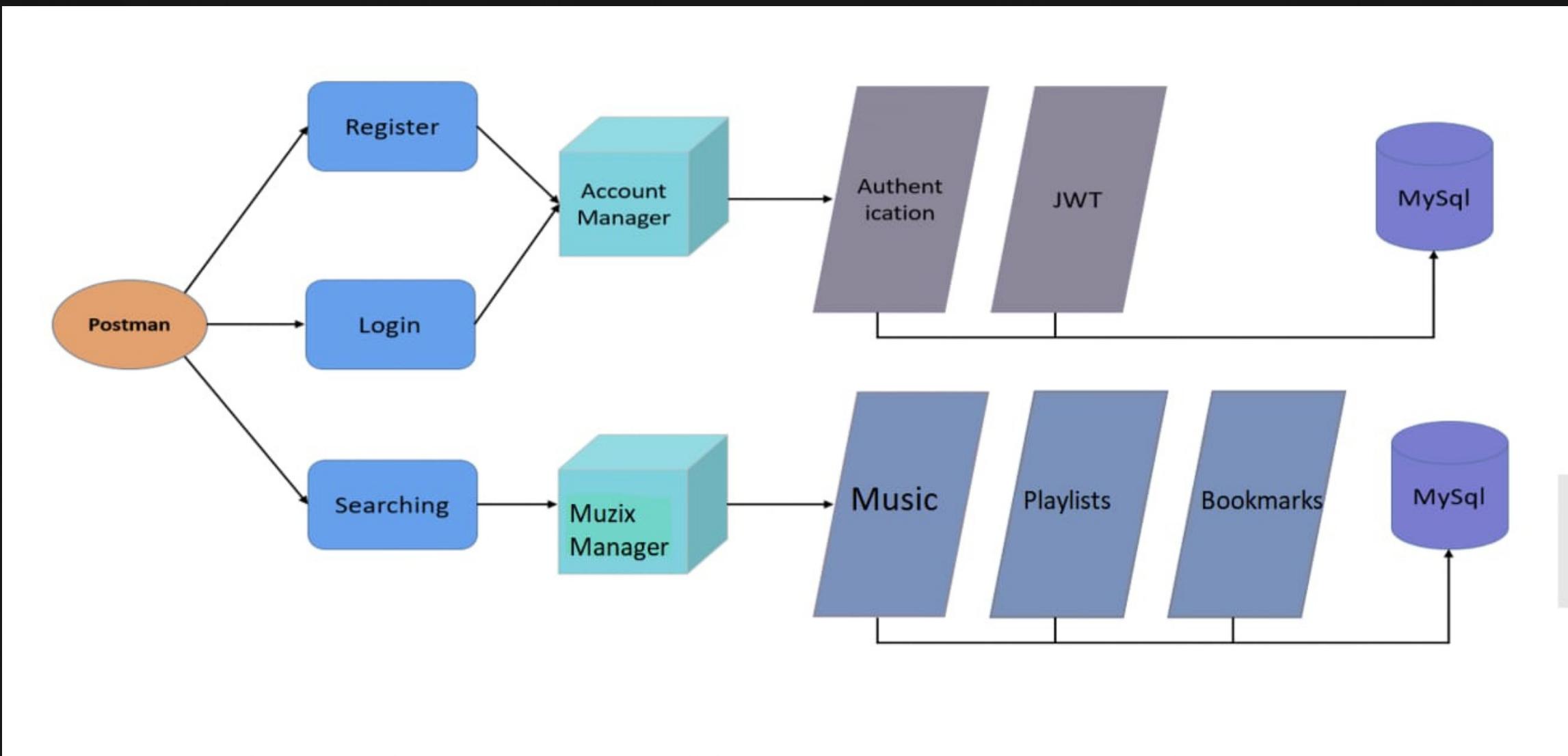
New users can sign up into the app easily and can add thier favourite songs to playlists or bookmark so that they can see those songs as soon as they logged in



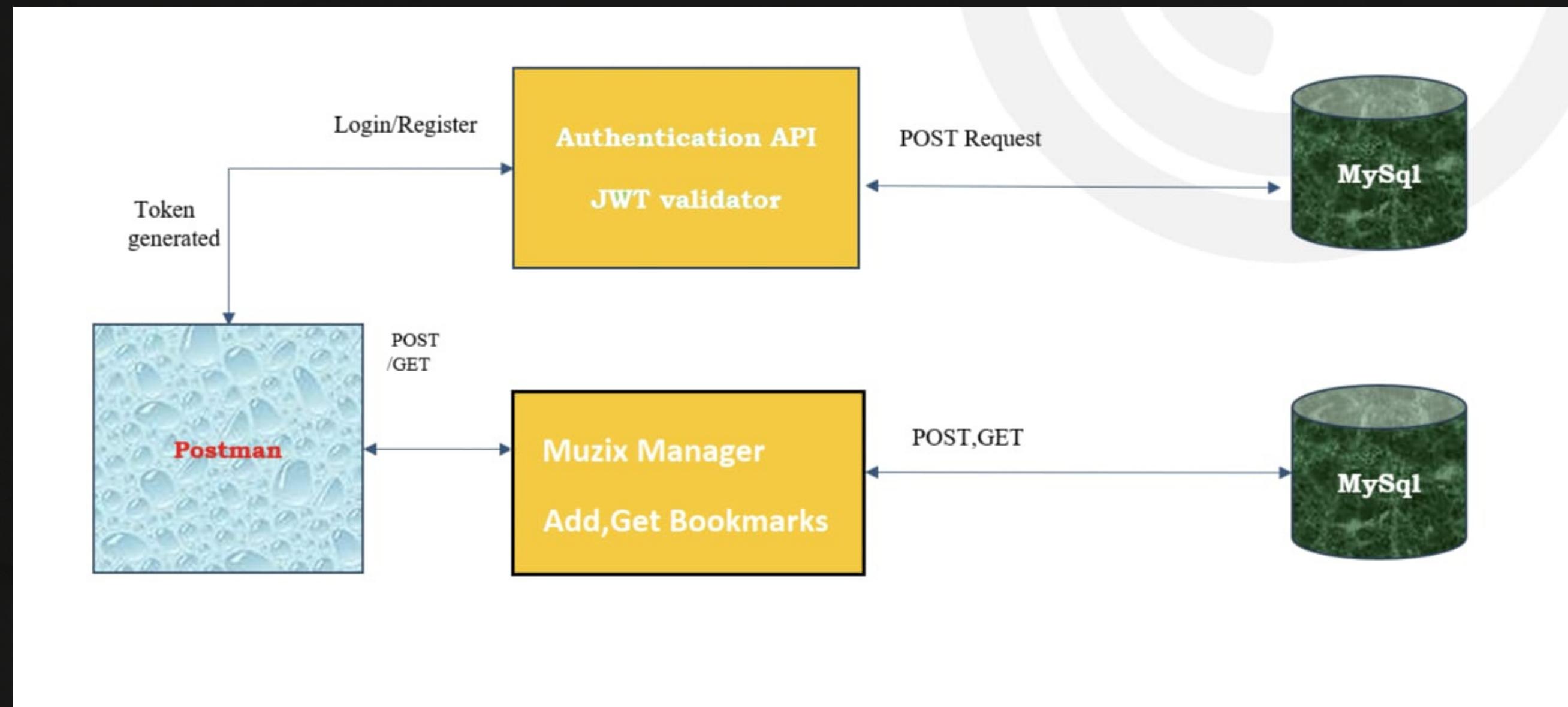
something
watching
lackth
ay.
or a
g-pl
f bi
to
me
nic
be
t
0
6

PROJECT

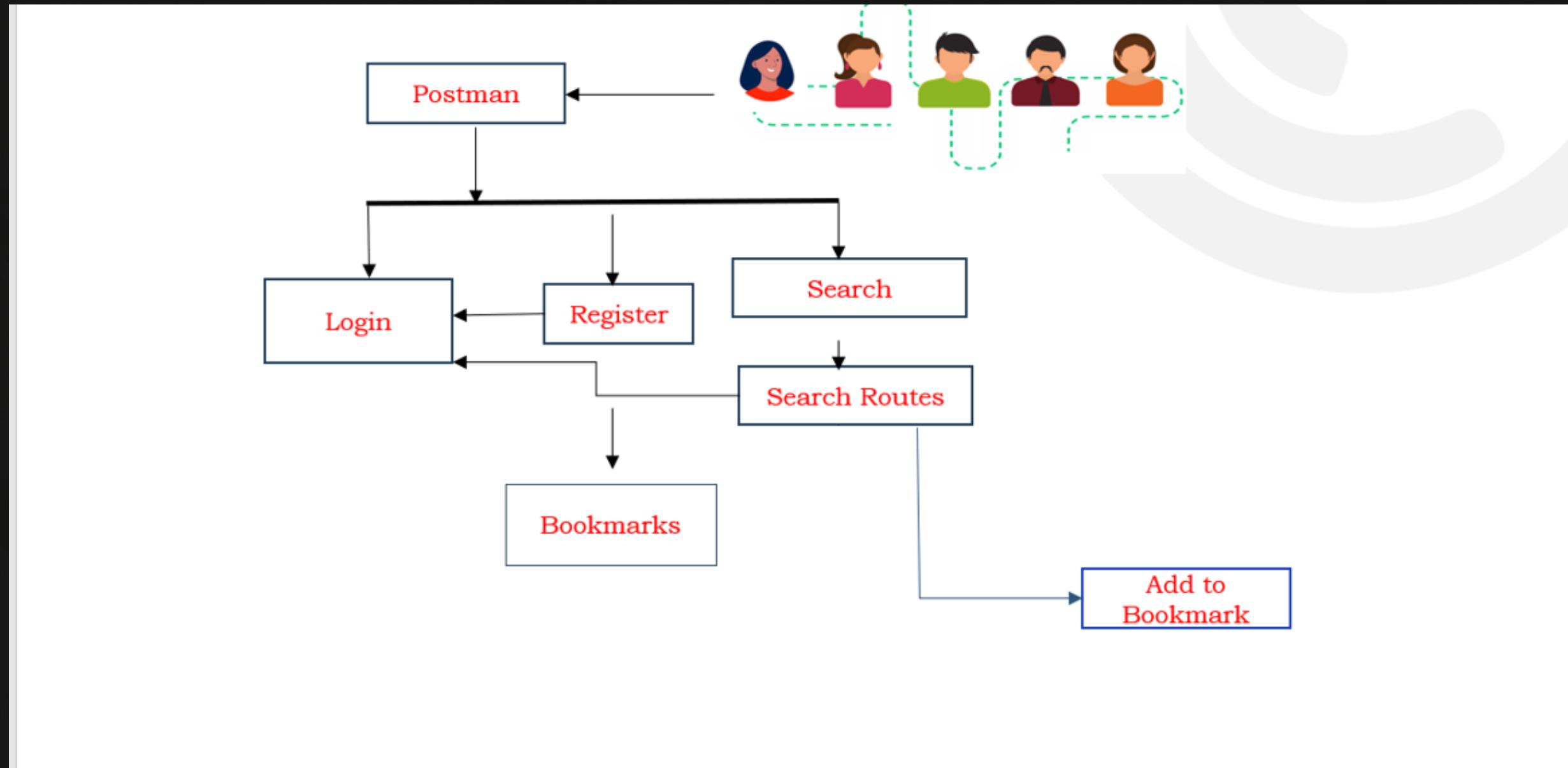
OVERFLOW



ARCHITECTURE



PROJECT FLOW



SWAGGER

localhost:2245/swagger-ui.html#/music-controller

swagger default (/v2/api-docs) api_key Explore

SPRINGBOOT-WITH-RESTAPI-SWAGGER

This is my first Swagger Application

Created by SOWMYA
See more at <http://localhost:2245/api/musics>
[Contact the developer](#)
[Apache 2.0](#)

bookmark-controller : Bookmark Controller

Show/Hide | List Operations | Expand Operations

GET	/api/bookmarks	getAllBookmark
GET	/api/bookmarks/{userId}	getBookmarksByUserId
DELETE	/api/delbookmark/{songName}	deleteBookmarkBySongName
GET	/api/getbookmark/{songName}	getBookmarkBySongName
GET	/api/getbookmarks/{id}	getBookmarkById
POST	/api/saving	saveBookmark

music-controller : Music Controller

Show/Hide | List Operations | Expand Operations

DELETE	/api/delmusic/{id}	deleteMusicById
GET	/api/getmusic/{songName}	getMusicBySongName
GET	/api/getmusics/{singerName}	getMusicBySingerName
GET	/api/getmusicsid/{songId}	getMusicBySongId

POSTMAN Screenshot

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Explore' (highlighted), a search bar ('Search Postman'), and various buttons like 'Invite', 'Upgrade', and 'Join Waitlist'. Below the header, a banner mentions 'Integration, Postman for gRPC, and Postman CLI.'

The main workspace displays a collection named 'Overview' containing a POST request to 'http://localhost:3345/authenticate' and a GET request to 'http://localhost:2245/a'. The POST request is selected.

The request details panel shows:

- Method: POST
- URL: http://localhost:3345/authenticate
- Body tab is active, showing JSON content:

```
1
2   ...
3     "user": "ankit",
4     "password": "ankit123"
```

The response panel shows:

- Status: 200 OK
- Time: 193 ms
- Size: 478 B
- Raw response content: eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhbmtpdCIsImV4cCI6MTY2Mjg20TEyMywiaWF0IjoxNjYyODMzMTEzfQ.DW7-hwAW4tndbRKo95l1xs6e9x2YWrtuFexOpwl5R0E

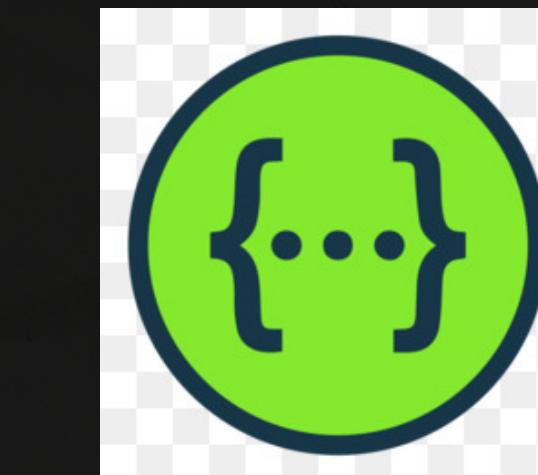
TOOLS/TECHNOLOGIES USED



Spring tools Suite



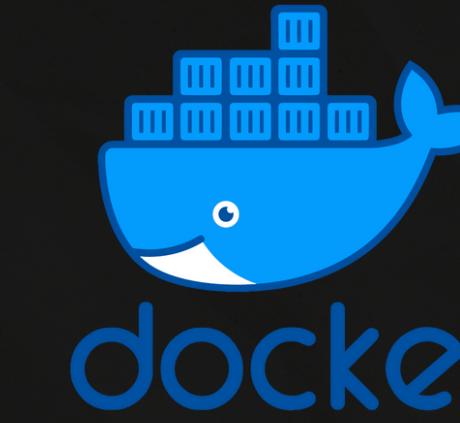
MySQL



Swagger



Postman



Docker



GITLAB



THANK YOU

B A T C H C

M U Z I X A P P