A Project Report on

Developing Smart ML Based Recommendation System

Submitted in partial fulfillment of the requirements for the award
of the degree of

Bachelor of Engineering

in

Information Technology

by

Sakshi Naik(17104059)
Sayali Phowakande(17104060)
Arjun Rajput(17104068)

Under the Guidance of

Prof.Apeksha Mohite
Prof.Geetanjali Kalme

Department of Information Technology
NBA Accredited
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2021-2022

Approval Sheet

This Project Report entitled ***"Developing Smart ML Based Recommendation System"*** Submitted by ***Sakshi Naik(17104059), Sayali Phowakande(17104060), Arjun Rajput(17104068)***is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Prof.Geetanjali Kalme
Co-Guide

Prof.Apeksha Mohite
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Developing Smart ML Based Recommendation System"*** submitted by ***Sakshi Naik(17104059), Sayali Phowakande (17104060), Arjun Rajput(17104068)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***,to the University of Mumbai,is a bonafide work carried out during academic year 2021-2022.

Prof.Geetanjali Kalme Co-Guide	Prof.Apeksha Mohite Guide
Prof. Kiran Deshpande Head Department of Information Technology	Dr. Uttam D.Kolekar Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Acknowledgement

We have great pleasure in presenting the report on **Developing Smart ML Based Recommendation System** We take this opportunity to express our sincere thanks towards our guide **Prof.Apeksha Mohite** & Co-Guide **Prof.Geetanjali Kalme** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Sakshi Naik:
17104059:

Sayali Phowakande:
17104060:

Arjun Rajput:
17104068:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Sakshi Naik (17104059)

Sayali Phowakande (17104060)

Arjun Rajput (17104068)

Date:

Abstract

Sometimes, music plays an important role in our life. Whether you are sad or happy, music plays an important factor as it expresses your mind, also the importance of music in your life will depend on your personal experience. This Recommendation System is developed for those users who express their feeling and prefer listening as well as viewing music videos depending on their choice. The recommendation system will filter out the contents depending upon user choice with similar data. For this Recommendation System, two techniques are used which are Collaborative Filtering and Content-based filtering. Considering the issues for some users while searching they can play music with help of a voice assistant.

Contents

List of Figures

List of Abbreviations

ML:	Machine Learning

Chapter 1

Introduction

For some people, music has an important role in life. Music is one of the solutions for many problems as it enhances mood, can make excited, can also feel relaxing and calm. Basically, it helps you from reducing stress, depression, pain. Considering the rapid development of mobile devices and the internet has made it possible to get more closer to the music by music player system. The reason behind most portable music systems is that music can be played whenever and where ever. The increase in the number of music available exceeds the listening capacity of a single individual. Therefore it is sometimes difficult to choose from millions of music. The solution for this issue is that there should be a good recommender system that can provide user music recommendations. This system will basically recommend music to users according to analyzing the most popular, highly rated and users preference this will help users to get personalized results. By considering all factors we are developing a smart recommendation system that will be convenient for music listeners as well as there would be an increase of audience in the music field. This Smart recommendation system is an advancement to the basic music player as it can recommend music to users by personalizing according to their choice and demands. Besides recommending music audio, this system also provides a recommendation of videos for 'now playing' music i.e While listening to the music, the music player screen while also provides a recommendation of music in video format. To make it more convenient voice assistants would be integrated with this smart recommendation system to have a hand-free experience. This System even helps out those users who are unable to type manually by using Voice Assistant. Voice assistant is basically implemented to have a hands-free experience for users. Voice-assistant would also result to recommend music on basis of the user's request.This would be beneficial for those are busy in other activities and wish to have music played simultaneously. In this Recommendation System, the resulting Recommendation is be provided based on audience preference or request. Besides this, the system will also recommend music based on categories like Latest Music, Top Hit Music that will introduce users to new music. For processing, this recommendation machine learning algorithms need to implement. This algorithm helps to instruct the data set and provide desirable output depending upon the specific algorithm used. The most common approach towards recommendation systems has been the Content-based Technique and Collaborative Technique. Most of the recommendation systems use collaborative filtering techniques as recommended music based on a community of users, their preferences, and their browsing behavior. Whereas in Content-Based filtering is a technique in which music is recommended on the basis of users' similar preferences and by knowledge accumulated of the user by studying users' recent played, favorite music, search music, etc. Besides recommending music this system will also provide video recommendations i.e while playing music the system will also provide Video option recommendations which will be useful for those users who want to play videos instead of audio, this can be implemented by navigating users to another page which will view video content. Another feature of the smart recommendation system is integrating the system with voice assistants. Voice-Assistant will also act as an interface for some users. This feature would be useful for those users who are willing to work simultaneously while playing music, which means that requesting voice assistant through command would give results. Basically, voice assistance is integrated to have a hand-free experience.

Chapter 2

Literature Review

2.1 Video Recommendation System Based on Human Interest

In year 2019 author Shainee Jain,Tejaswi Pawar, Heth Shah,Omkar Morye and Bhushan Patil, has published paper Video Recommendation System Based on Human Interest. This system is develop for teenager user who are likely to watch videos on mobile. This paper proposes a video recommendation system that collects the reaction of the users for various videos which helps to know its relevance. Based on the viewers' watching history or browsing, the system is capable of recommending videos to the users. In this system Hybrid System is used which combination of Content Base filtering and Collaborative filtering.

2.2 Music Recommendation using Collaborative Filtering and Deep Learning

In year 2019 Anand Neil Arnold and Vaira Muthu S has published paper Music Recommendation using Collaborative Filtering and Deep Learning.This Music Recommendation system recommends user music as well as videos depending on user preference. Here Collaborative filtering is used in which existing history of the user and recommends music from other user's history which are similar.

2.3 Artificial Intelligence-based Voice Assistant

In the year 2020 S Subhash, Prajwal N Srivatsa, S Siddesh, A Ullas Santhosh B publish paper Artificial Intelligence-based Voice Assistant.This system is basically an intelligent personalized assistant which can perform mental tasks like turning on/off smartphone applications with the help of the Voice User interface (VUI) which is used to listen and process audio commands. For this, the PyCharm library was installed from python packages. As a result, this system gives output on voice commands like playing songs on video, searching any location, and google search output.

2.4 Music Video Recommendation Based on Link Prediction Considering Local and Global Structures of a Network

In this literature the authors Yui Matsumoto; Ryosuke Harakawa; Takahiro Ogawa; Miki Haseyama publish the paper in the year 2019 which was Music Video Recommendation Based on Link Prediction Considering Local and Global Structures of a Network in this system they have implemented a novel method based on LP-LG SN for recommending music and videos. In this, they have the construction of a network by collaborative use of the multi-model feature. As a result, it can work well in real-world applications. In the future, this application will introduce a framework to fuse prediction which can control the effect of local and global structure-based

2.5 FARMER'S ASSISTANT using AI Voice Bot

In the year 2021 authors Kiruthiga Devi M; Divakar M S; Vimal Kumar V; Martina Jaincy D E; Kalpana R A; Sanjai Kumar R M published a paper titled FARMER'S ASSISTANT using AI Voice Bot. The main purpose of this application is to develop a mobile application that can assist farmers depending on two techniques voice bot and suggestion bot. The multi-language response was generated depending on the farmer's queries. These queries were responded to by a multi-linguistic bot which was implemented using Google translator,pysttsx3, and Google search engines. This mobile application can improve increase in agriculture production and suggest farmer for progress in better farming practice.

2.6 Mobile App Recommendation System Using Machine learning Classification

In the year 2020, the authors Jisha R C, Amrita J M, Aswini R Vijay, Indhu G S had published a paper titled Mobile App Recommendation System Using Machine learning Classification. This application is basically developed Web Crawling and by using a clustering algorithm. Here Web Crawling is used to record information about websites and the Clustering algorithm is used to collect clusters on basis of Popularity and security aspects. The basic aim behind this application is to provide a simple recommendation system. Basically, this application shows how the rating, permission, and size of the application are being considered. Web crawling is implemented to extract users' rating permission and application size. This application was built on an android application that would give more efficient and accurate results.

Chapter 3

Objective

3.1 Compatibility

To develop a Cross platform application, i.e Developing single application that can be run on different operating system.

3.2 Feasibility

To build a hand-free mobile application by integrating it with Voice assistant,which can make application more convenient.

3.3 Regularity

To keep a track of frequently played music by user.

3.4 Usability

To provide recommendations based on recorded information of users' preferences and suggesting video link of played music so that even videos can be watched.

3.5 Serviceability

To deliver a set of playlist from analyzing the current and future popularity of music ,artist and genres.

Chapter 4

Project Design

4.1 Existing System

Most of the music system recommend music based on user profile.This is achieved based on analysing users profiles.Analysing user profile can achieved recommendation by applying two techniques.This techniques can be Content based filtering and collaboration based or something only one technique can be used.Recommendation on bases of this can not more innovative.Some system do not consider user preferences while recommending this practice couldn't help system to filter out user preference's playlist.This system recommends by using tags.

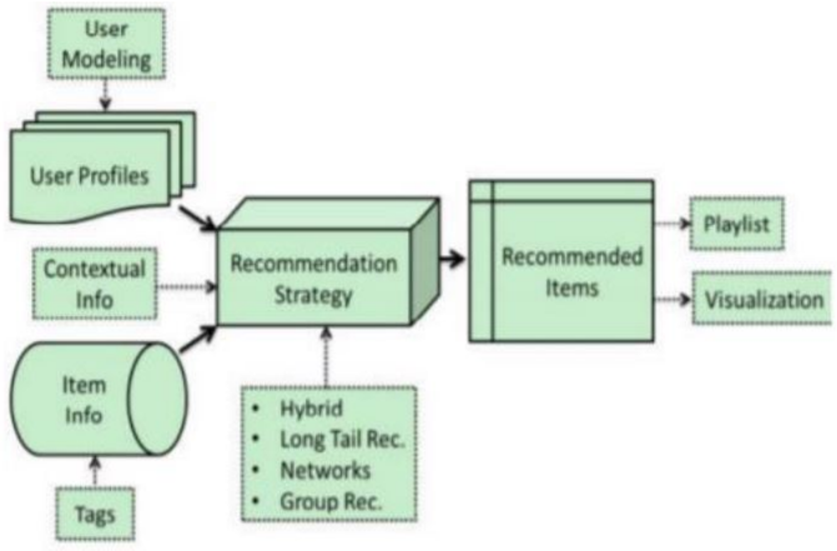


Figure 4.1: Existing System.

4.2 Proposed System Architecture

The main goal of our application is to recommend users with the latest, preferred, and previously played music along with the video link. This is can be implemented by applying machine learning filtering algorithms which are collaborative and content-based filtering. This algorithm provides music based on user history and by collecting other user preferences. Recommendation is achieved on bases of hybrid recommendation technique Following are the modules which are been considered while implementing this application.

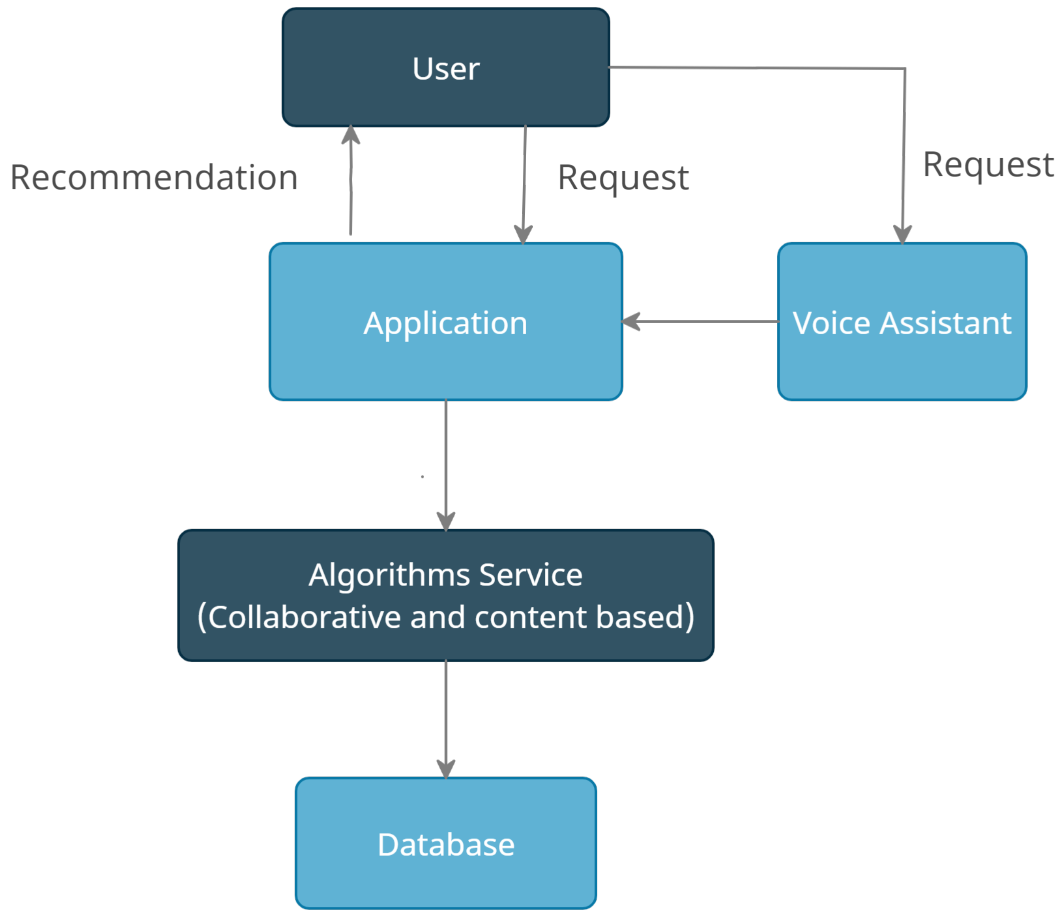


Figure 4.2: Block diagram.

4.2.1 User

The user module is the targeted module that will request recommendations by interacting with the application or with a voice assistant to get music recommendations.

4.2.2 Application

This module is the main interaction with the user module which consists of a main application wherein music is recommended and played according to the user. This is a module where the user interacts the most to get recommendations.

4.2.3 Algorithms Service

This module consist of a Machine learning mechanism is which will recommend music by using algorithms like content-based and collaboration filtering.

4.2.4 Database

This module consists of a collection of user details and a music playlist which would be pushed towards user's dashboard depending upon the algorithm.

4.2.5 Voice assistant

This module is implemented to perform hand-free use of application wherein user can command to assistant and assistant future send the request to the application.

4.3 UML

4.3.1 Use-Case Diagram

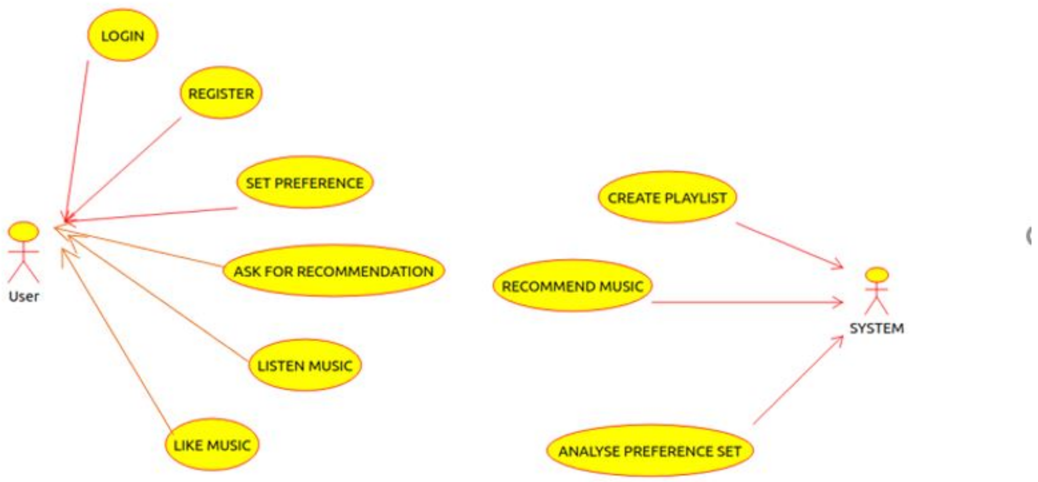


Figure 4.3: Use Case Diagram

In this project there are two modules. One is system and another one is user module. Users need to first register themselves whatever information asked by this application. After that users can login using email id and password. User can see multiple music in system. If the users want to like the music from the play list they can like them and store for future references.All the analysed music from user can be recommended to user. Here system create playlist for users,after analysing users profiles it push recommendation.

4.3.2 Activity Diagram

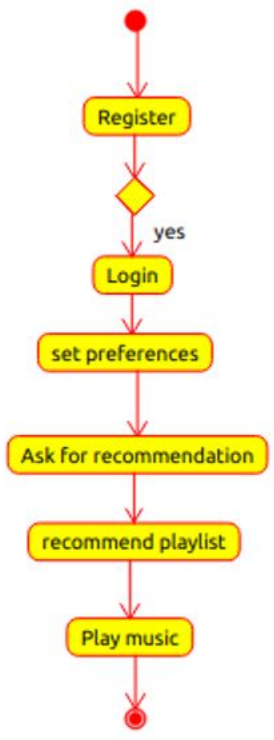


Figure 4.4: Activity Diagram.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. In this activity diagram, we will discuss the flow of project. The user will first register themselves into system by setting preferences,After registration user can login to the system.If user do not have login credential they need to first register themselves. after login user can set preference and ask for recommendation and Play music.

4.3.3 Class Diagram

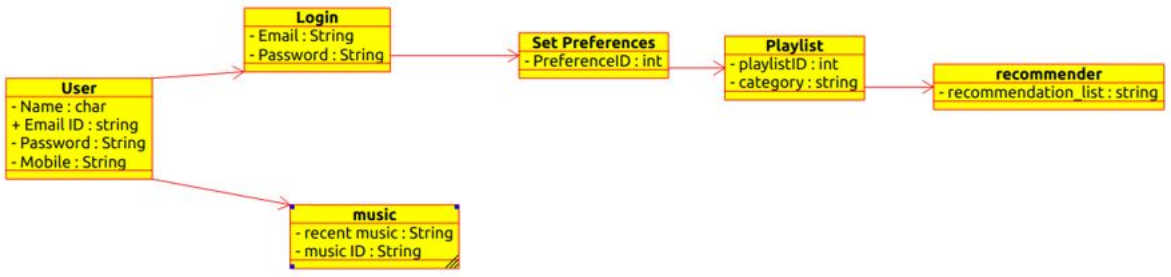


Figure 4.5: Class diagram.

Class diagram is a static diagram. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system.

Chapter 5

Project Implementation

```
import 'package:flutter/material.dart';
// ignore: unused_import
import 'package:flutter_signin_button/flutter_signin_button.dart';
// ignore: unused_import

import 'package:srs_ml/SignIn.dart';

import 'MyLogin.dart';

// ignore: use_key_in_widget_constructors
class Welcome extends StatefulWidget {

  @override
  _WelcomeState createState() => _WelcomeState();
}

class _WelcomeState extends State<Welcome> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      // ignore: avoid_unnecessary_containers
      body: Container(
        child: Column(

          children: <Widget>
          [
            SizedBox(height:45.0),
            // ignore: avoid_unnecessary_containers, sized_box_for_whitespace
            Container(
              height: 300,
              child: Image(image: AssetImage("images/Welcome.JPG"),
              fit: BoxFit.contain,
              ),
            ),
            SizedBox(height:20),
            RichText(
              text: TextSpan(
                text:'Welcome to Music App',style: TextStyle(fontSize:25.0,fontWeight
                color:Colors.pink),
                // ignore: prefer_const_literals_to_create_immutables

              )
            ),
            SizedBox(height:10.0),
            Text('Let go world on!',style: TextStyle(color:Colors.pink),),
            SizedBox(height:30.0),


            Row( mainAxisAlignment: MainAxisAlignment.center,
              children: <Widget>[

                // ignore: deprecated_member_use
                RaisedButton(
                  padding:EdgeInsets.only(left:30,right:30),
                  onPressed: (){
                  Navigator.push(
                    context,
                    MaterialPageRoute(builder: (context) => MyLogin()),
                      );
                        },
                  child: Text('Login',style: TextStyle(
                    fontSize: 20,
                    fontWeight:FontWeight.bold,
                    color:Colors.white,
                  ),),
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(10.0),

                  ),
                  color:Colors.pink
                  ),
                  SizedBox(width:20.0),
                  // ignore: deprecated_member_use
                  RaisedButton(
                   padding:EdgeInsets.only(left:30,right:30),
                  onPressed: (){Navigator.push(
                    context,
                    MaterialPageRoute(builder: (context) => SignIn()),
                      );
                      },
                  child: Text('Sign Up!',style: TextStyle(
                    fontSize: 20,
                    fontWeight:FontWeight.bold,
                    color:Colors.white,
                  ),),
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(10.0),

                  ),
                  color:Colors.pink
                  ),
                  SizedBox(height: 30.0),


              ],
            ),
            SignInButton(
                    Buttons.Google,
                       text: "Sign up with Google",
                        onPressed: (){Navigator.push(
                    context,
                    MaterialPageRoute(builder: (context) => SignIn()),
                      );
                      },
              )

          ],
          ),
      ),

    );
  }
}
```

Login

```
// ignore: duplicate_ignore
// ignore: file_names
// ignore_for_file: prefer_const_constructors, file_names, no_logic_in_create_state,

import 'package:flutter/material.dart';
import 'package:srs_ml/Welcome.dart';
import 'package:firebase_auth/firebase_auth.dart';

import 'SignUp.dart';

class Loginmy extends StatefulWidget {
  const Loginmy({Key? key}) : super(key: key);

  @override
  _LoginmyState createState() => _LoginmyState();
}

// ignore: non_constant_identifier_names
class _LoginmyState extends State<Loginmy> {
  // ignore: unused_field
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  late String _email,_password;

   checkAuthentification() async {
    _auth.authStateChanges().listen((user) {
      if (user != null) {
        print(user);

        Navigator.pushReplacementNamed(context, "/");
      }
    });
   }

   @override
  void initState() {
    super.initState();
    // ignore: unnecessary_this
    this.checkAuthentification();
  }

  login() async {
    if (_formKey.currentState!.validate()) {
      _formKey.currentState!.save();

      try {
        await _auth.signInWithEmailAndPassword(
            email: _email, password: _password);
      } catch (e) {

        print('Error is $e');
      }
    }
  }

  showError(String errormessage) {
    showDialog(
        context: context,
        builder: (BuildContext context) {
          return AlertDialog(
            title: Text('ERROR'),
            content: Text(errormessage),
            actions: <Widget>[
              FlatButton(
                  onPressed: () {
                    Navigator.of(context).pop();
                  },
                  child: Text('OK'))
            ],
          );
        });
  }

  navigateToSignUp() async {
    Navigator.push(context, MaterialPageRoute(builder: (context) => SignUp()));
  }




  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        // ignore: avoid_unnecessary_containers
        child: Container(
        child: Column(
          children: <Widget>[
            // ignore: sized_box_for_whitespace
            Container(
              height: 400,
              child: Image(
                image: AssetImage("images/login.jpg"),
                fit: BoxFit.contain,
              ),
            ),
            SizedBox(height:20),
            RichText(
              text: TextSpan(
                text:'Welcome Back',style: TextStyle(fontSize:25.0,fontWeight: FontWe
                color:Colors.pink ),
                // ignore: prefer_const_literals_to_create_immutables

              )
            ),
            Container(
              child: Form(
                key: _formKey,
                child: Column(

                  children: <Widget>[
                    Container(


                      child: TextFormField(
                          validator: (input) {
                            if (input!.isEmpty) return 'Enter Email';
                          },
                          decoration: InputDecoration(
                              labelText: 'Email',
                              prefixIcon: Icon(Icons.email)),
                          onSaved: (input) => _email = input!),
                    ),
                    Container(
                      child: TextFormField(
                          validator: (input) {
                            if (input!.length < 6)
                              // ignore: curly_braces_in_flow_control_structures
                              return 'Provide Minimum 6 Character';
                          },
                          decoration: InputDecoration(
                            labelText: 'Password',
                            prefixIcon: Icon(Icons.lock),
                          ),
                          obscureText: true,
                          onSaved: (input) => _password = input!),
                    ),
                    SizedBox(height: 20),
                    // ignore: deprecated_member_use
                    RaisedButton(
                      padding: EdgeInsets.fromLTRB(70, 10, 70, 10),
                      onPressed: (){},
                      child: Text('LOGIN',
                          style: TextStyle(
                              color: Colors.white,
                              fontSize: 20.0,
                              fontWeight: FontWeight.bold)),
                      color: Colors.pink,
                      shape: RoundedRectangleBorder(
                        borderRadius: BorderRadius.circular(20.0),
                      ),
                    )
```

```
                ],
              ),
            ),
          ),

        ],
      ),
    ),
  )




  );
 }
}
```

Registration

```
// ignore: duplicate_ignore
// ignore: file_names
// ignore_for_file: file_names, prefer_const_constructors, unused_field

import 'package:flutter/material.dart';
import 'package:srs_ml/MyLogin.dart';
class SignIn extends StatefulWidget {
  const SignIn({ Key? key }) : super(key: key);

  @override
  _SignInState createState() => _SignInState();

}

class _SignInState extends State<SignIn> {
  get _formKey => null;
  late String _name, _email, _password;
  var _value = false;
  var _value1 = false;
  var _value2 = false;
  var _value3 = false;
  var _value4 = false;
  var _value5 = false;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(


        // ignore: avoid_unnecessary_containers
        child: Container(
        child: Column(
          children: <Widget>[
            // ignore: sized_box_for_whitespace
            Container(
              height: 400,
              child: Image(
                image: AssetImage("images/login.jpg"),
                fit: BoxFit.contain,


              ),
            ),
            SizedBox(height:20),
            RichText(
              textAlign: TextAlign.left,
              text: TextSpan(
                text:'Create Your Account!',style: TextStyle(fontSize:25.0,fontWeight
                color:Colors.pink ),
                // ignore: prefer_const_literals_to_create_immutables

              )
            ),
            Container(
              child: Form(
                key: _formKey,
                child: Column(

                  children: <Widget>[
                     Container(


                      child: TextFormField(
                          validator: (input) {
                            if (input!.isEmpty) return 'Enter Your Name';
                          },
                          decoration: const InputDecoration(
                              labelText: 'Name',
                              prefixIcon: Icon(Icons.person),
                          ),
                          onSaved: (input) => _name = input!),
                    ),
          Container(


           child: TextFormField(
             validator: (input) {
              if (input!.isEmpty) return 'Enter Email';
             },
             decoration: const InputDecoration(
               labelText: 'Email',
               prefixIcon: Icon(Icons.email)),
             onSaved: (input) => _email = input!),
          ),
          Container(
           child: TextFormField(
             validator: (input) {
              if (input!.length < 6)
               // ignore: curly_braces_in_flow_control_structures
               return 'Provide Minimum 6 Character';
             },
             decoration: InputDecoration(
              labelText: 'Password',
              prefixIcon: Icon(Icons.lock),
             ),
             obscureText: true,
             onSaved: (input) => _password = input!),
          ),
          SizedBox(height: 20),
          RichText(
       textAlign: TextAlign.left,
       text: TextSpan(
        text:'Set Your Preferences Here!',style: TextStyle(fontSize:25.0,font
        color:Colors.pink ),
        // ignore: prefer_const_literals_to_create_immutables

       )
      ),
      SizedBox(height: 20),
      Container(
       child: SwitchListTile(
        title:Text('Party Music'),
        subtitle:Text('Will recommend Party Music'),
        activeColor: Colors.pink,
        value: _value,
        onChanged: (value){
         setState(() {
          _value=value;
```

```
          });
        },
        )
),

SizedBox(height: 20),
    Container(
      child: SwitchListTile(
        title:Text('Classical Music'),
        subtitle:Text('Will recommend Classical Music'),
        activeColor: Colors.pink,
        value: _value2,
        onChanged: (value){
          setState(() {
            _value2=value;
          });
        },
        )
),
SizedBox(height: 20),
    Container(
      child: SwitchListTile(
        title:Text('K-Pop'),
        subtitle:Text('Will recommend K-Pop Music'),
        activeColor: Colors.pink,
        value: _value3,
        onChanged: (value){
          setState(() {
            _value3=value;
          });
        },
        )
),
SizedBox(height: 20),
    Container(
      child: SwitchListTile(
        title:Text('Bollywood Hits'),
        subtitle:Text('Will recommend Bollywood Hits'),
        activeColor: Colors.pink,
        value: _value4,
        onChanged: (value){
          setState(() {
            _value4=value;
          });
        },
        )
),
SizedBox(height: 20),
    Container(
      child: SwitchListTile(
        title:Text('Old is Gold'),
        subtitle:Text('Will recommend Old Music'),
        activeColor: Colors.pink,
        value: _value1,
        onChanged: (value){
          setState(() {
            _value1=value;
          });
        },
        )
),
SizedBox(height: 20),
    Container(
      child: SwitchListTile(
        title:Text('Peaceful Music'),
        subtitle:Text('Will recommend Silent peaceful music'),
        activeColor: Colors.pink,
        value: _value5,
        onChanged: (value){
          setState(() {
            _value5=value;
          });
        },
        )
),
            // ignore: deprecated_member_use
            RaisedButton(
              padding: EdgeInsets.fromLTRB(70, 10, 70, 10),
              onPressed: (){Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => MyLogin()),
              );
              },
              child: Text('Create!',
                  style: TextStyle(
                      color: Colors.white,
                      fontSize: 20.0,
                      fontWeight: FontWeight.bold)),
              color: Colors.pink,
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(20.0),
              ),
            ),
            SizedBox(height:20),
                    ],
                  ),
                ),
              ),

            ],
          ),
        ),
      )





      );


    }
  }
```

Playlist

```
package com.google.firebase.codelabs.recommendations

import android.os.Bundle
import android.util.Log
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.lifecycle.ViewModelProvider
import com.google.firebase.codelabs.recommendations.adapters.FilterType
import com.google.firebase.codelabs.recommendations.adapters.ItemClickListener
import com.google.firebase.codelabs.recommendations.adapters.MoviesAdapter
import com.google.firebase.codelabs.recommendations.data.Movie
import com.google.firebase.codelabs.recommendations.databinding.FragmentMovieListBind
import com.google.firebase.codelabs.recommendations.viewmodels.LikedMoviesViewModel


/**
 * A fragment containing the list of available movies recognized in our model.
 */
class MovieListFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val binding = FragmentMovieListBinding.inflate(inflater, container, false)
        context ?: return binding.root

        val viewModel: LikedMoviesViewModel = ViewModelProvider(requireActivity()).ge
        val movieClickListener = object : ItemClickListener() {
                         override fun onLike(movie: Movie) {
                            viewModel.onMovieLiked(movie)
                        }

                        override fun onRemoveLike(movie: Movie) {
                            viewModel.onMovieLikeRemoved(movie)
                        }
        }
        val adapter = MoviesAdapter(movieClickListener, FilterType.NONE)
        binding.list.adapter = adapter
        viewModel.movies.observe(viewLifecycleOwner) {
            adapter.submitList(it.toList())
            adapter.notifyDataSetChanged()
        }
        setHasOptionsMenu(true)
        return binding.root
    }
}
```

Liked

```
package com.google.firebase.codelabs.recommendations

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import androidx.lifecycle.ViewModelProvider
import com.google.firebase.codelabs.recommendations.adapters.FilterType
import com.google.firebase.codelabs.recommendations.adapters.ItemClickListener
import com.google.firebase.codelabs.recommendations.adapters.MoviesAdapter
import com.google.firebase.codelabs.recommendations.data.Movie
import com.google.firebase.codelabs.recommendations.databinding.FragmentLikedMoviesBi
import com.google.firebase.codelabs.recommendations.viewmodels.LikedMoviesViewModel
import java.lang.Exception


/**
 * Fragment showing the list of movies the user has liked.
 */
class LikedMoviesFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        val binding = FragmentLikedMoviesBinding.inflate(inflater, container, false)
        context ?: return binding.root

        val viewModel: LikedMoviesViewModel = ViewModelProvider(requireActivity()).ge
        val movieClickListener = object : ItemClickListener() {
            override fun onLike(movie: Movie) {
                throw Exception("Movie was already liked")
            }

            override fun onRemoveLike(movie: Movie) {
                viewModel.onMovieLikeRemoved(movie)
            }
        }
        val adapter = MoviesAdapter(movieClickListener, FilterType.LIKED)
        binding.list.adapter = adapter

        viewModel.movies.observe(viewLifecycleOwner) {
            adapter.submitList(it.toList())
            adapter.notifyDataSetChanged()
        }
        setHasOptionsMenu(true)
        return binding.root
    }
}
```

Chapter 6

Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input. In this application actual functionality is recommend music. In this application user can see music recommendation according to their preferences. Unit testing checks that whether the music appear on the screen or not. User can select their song as per the choice. Unit testing checks that whether the recommendation of songs or the system is running well or not.

Chapter 7

Result

Home

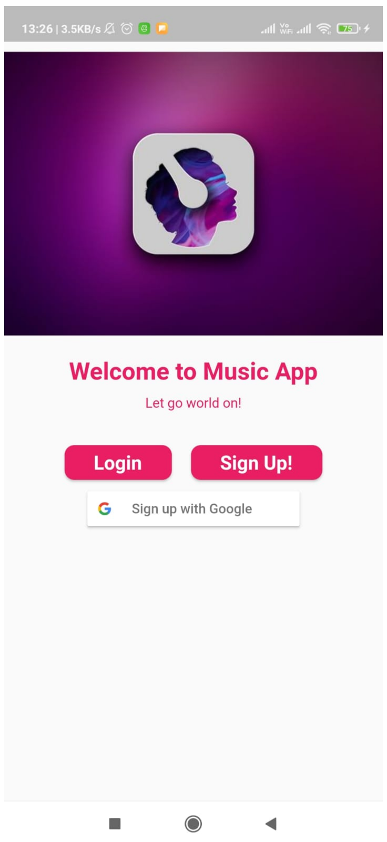


Figure 7.1: Home page.

This is a home page page for the system were users can Register themselves or if registered previous can login through it. Here the login tab is used for those users who have already registered themselves.The Sign Up tabs is used to registered user by providing details to system.Here Sign Up in Google functionality is provided for Gmail users for authentication.

Login

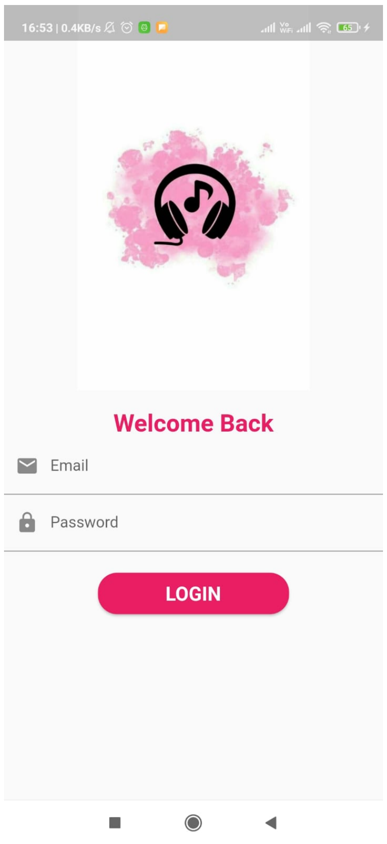


Figure 7.2: Login page.

This module consist of Login page where user can provide authenticate email and authenticated password which was provided by users while registration.If user is not been register earlier they need to register themselves before login.

Sign Up/Registration

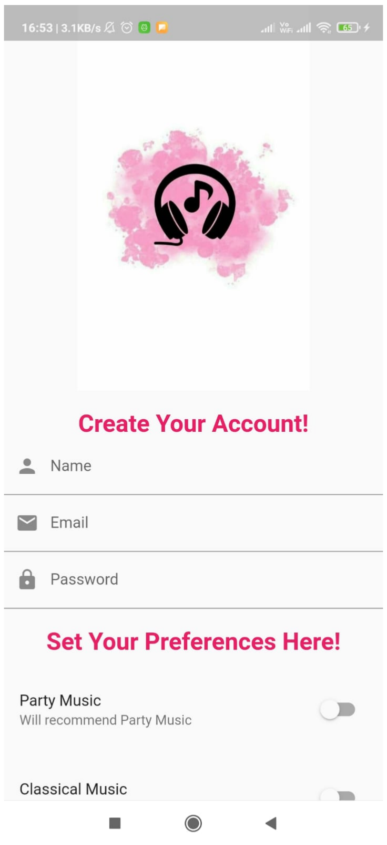
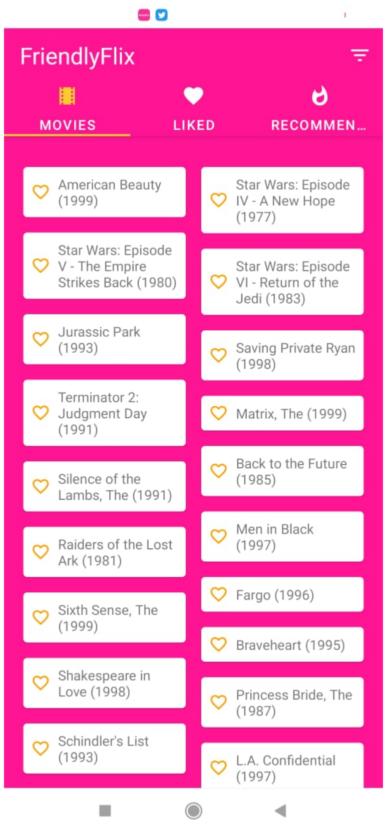


Figure 7.3: Sign Up.

This module consist of Sign Up page which can be also known as registration page in this page user need to provide specific information.While registration users get register to system and can login after this.While registering even the preferences where tried to collect from user.

Playlist


FriendlyFlix

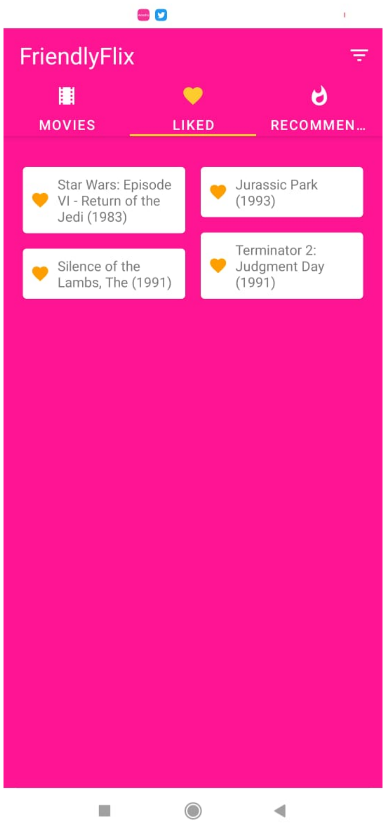
MOVIES
LIKED
RECOMMEN...

American Beauty (1999)
Star Wars: Episode IV - A New Hope (1977)
Star Wars: Episode V - The Empire Strikes Back (1980)
Star Wars: Episode VI - Return of the Jedi (1983)
Jurassic Park (1993)
Saving Private Ryan (1998)
Terminator 2: Judgment Day (1991)
Matrix, The (1999)
Silence of the Lambs, The (1991)
Back to the Future (1985)
Raiders of the Lost Ark (1981)
Men in Black (1997)
Sixth Sense, The (1999)
Fargo (1996)
Braveheart (1995)
Shakespeare in Love (1998)
Princess Bride, The (1987)
Schindler's List (1993)
L.A. Confidential (1997)

Figure 7.4: Playlist.

This module consist of playlist,this is the first page after user gets login into the system .This page consist of multiple music list.Future in this module user can like the music and and store in liked section.

Playlist

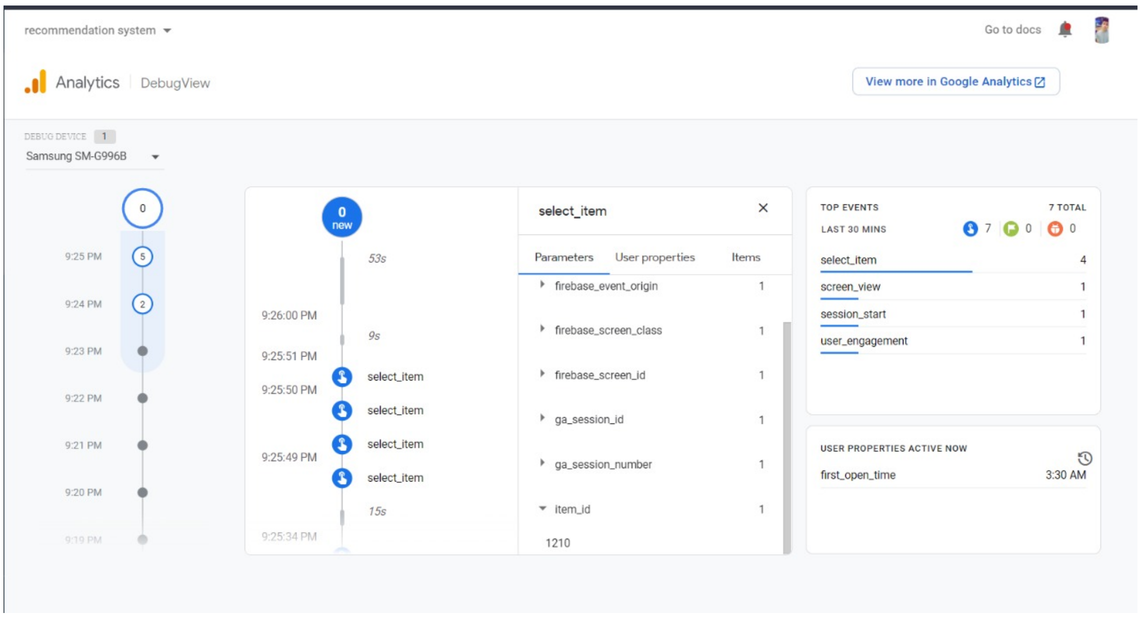


Star Wars: Episode VI - Return of the Jedi (1983)
Jurassic Park (1993)
Terminator 2: Judgment Day (1991)
Silence of the Lambs, The (1991)

Figure 7.5: Liked Music.

This module consist of liked or favorite playlist where user have liked the music from playlist ,this module can help user to get ease access from playlist as they have selected this music as favorite once.

DebugView



9:26:00 PM	
9:25:51 PM	
9:25:50 PM	select_item
	select_item
9:25:49 PM	select_item
	select_item
9:25:34 PM	

select_item

Parameters	User properties	Items
firebase_event_origin		1
firebase_screen_class		1
firebase_screen_id		1
ga_session_id		1
ga_session_number		1
item_id		1
1210		

TOP EVENTS — LAST 30 MINS — 7 TOTAL

select_item	4
screen_view	1
session_start	1
user_engagement	1

USER PROPERTIES ACTIVE NOW

first_open_time	3:30 AM

Figure 7.6: Debug View.

This is Debug View of the application which is used for analysis by firebase analytic.This Debug View shows tracking data being sent by user.Whenever the user selects an item from the screen this views monitors the events.

Chapter 8

Conclusions and Future Scope

By applying the knowledge and skill set, we are determined towards building a completely user interactive system that would be useful for every music listeners. According to the project objective the system was suppose to be implemented as a cross-platform application that will be compatible with multiple operating systems. This system was basically proposed a recommendation system with hybrid technique using ML as this system will recommend music to users depending upon preferences, recently played, and ratings of other users, Along with the music recommendation, a video link will also be suggested for those users who are interested to watch music in video format. To make it more innovative Voice assistants was suppose to be integrated for a hand-free and personalized experience to users.Considering this we have develop a system Cross platform application using Kotlin.Kotlin intends to be a software development kit for creating cross-platform mobile applications. Apart from this user can liked the music from the playlist,which can ease for further use.All of this data analyzing is monitor on firebase analytics.Talking about the scope of project this system can be used by music listeners however it exceeds the listening capacity of a single individual. It can be also used to keep a track of frequently liked music by user.

Bibliography

[1] Shainee Jain; Tejaswi Pawar; Heth Shah; Omkar Morye; Bhushan Patil," Video Recommendation System Based on Human Interest", 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)(IEEE).

[2] Sheela Kathavate, " Music Recommendation System using Content and Collaborative Filtering Methods" for IJERT : Volume 10, Issue 02 (February 2021).

[3] Anand Neil Arnold, Vairamuthu S.," Music Recommendation using Collaborative Filtering and Deep Learning"for International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-7, May, 2019.

[4] S Subhash; Prajwal N Srivatsa; S Siddesh; A Ullas; B Santhosh" Artificial Intelligence-based Voice Assistant"for 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)

[5] K. D. M, D. M S, V. K. V, M. Jaincy D E, K. R A and S. Kumar R M, "FARMER'S ASSISTANT using AI Voice Bot," 2021 3rd International Conference on Signal Processing and Communication (ICPSC), 2021, pp. 527-531, doi: 10.1109/ICSPC51351.2021.9451760.

[6] Xiangpo Li,"Research on the Application of Collaborative Filtering Algorithm in Mobile E-Commerce Recommendation System"for"2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)"

[7] "Yui Matsumoto; Ryosuke Harakawa; Takahiro Ogawa; Miki Haseyama","Music Video Recommendation Based on Link Prediction Considering Local and Global Structures of a Network" for "IEE Journal"

[8] Dr. Jagendra Singh,"Collaborative Filtering based Hybrid Music Recommendation System" for "Third International Conference on Intelligent Sustainable Systems [ICISS 2020]"

[9] S. Chang, A. Abdul, J. Chen and H. Liao, "A personalized music recommendation system using convolutional neural networks approach," 2018 IEEE International Conference on Applied System Invention (ICASI), 2018, pp. 47-49, doi: 10.1109/ICASI.2018.8394293.

[10] M. Sunitha and T. Adilakshmi, "Mobile based music recommendation system," 2016 International Conference on Inventive Computation Technologies (ICICT), 2016, pp. 1-4, doi: 10.1109/INVENTIVE.2016.7830183.

[11] Wu, Xia, Zhu, Yanmin, T1 - "A Hybrid Approach Based on Collaborative Filtering to Recommending Mobile Apps", DO - 10.1109/ICPADS.2016.0011

[12] R. C. Jisha, J. M. Amrita, A. R. Vijay and G. S. Indhu, ”Mobile App Recommendation System Using Machine learning Classification,” 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp. 940-943, doi: 10.1109/ICCMC48092.2020.ICCMC-000174.

[13] JOUR, Bae Joonho, Park Jinkyoo, Choi Jeonghye ”THE RECOMMENDER SYSTEM FOR MOBILE APPS” DO - 10.15444/GMC2018.10.07.03 Url - https://www.researchgate.net/publication/ 326691334THERECOMMENDERSYSTEMFORMOBILEAPPS

[14] R. Obeidat, R. Duwairi and A. Al-Aiad, ”A Collaborative Recommendation System for Online Courses Recommendations,” 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), 2019, pp. 49-54, doi: 10.1109/Deep-ML.2019.00018.

[15] Zheng Chen; Xueyue Liu; Li Shang,”Improved course recommendation algorithm based on collaborative filtering” for ”2020 International Conference on Big Data and Informatization Education (ICBDIE)”

[16] Saadman Shahid Chowdury, Atiar Talukdar, Ashik Mahmud, Tanzilur Rahman”Domain specific Intelligent personal assistant with bilingual voice command processing”IEEE

Appendices

Appendix-A: Android Studio Download and Installation

1. Download android studio from https://developer.android.com/studio/

2. To install the android studio,Click the install tab. The default install locations are: Windows:C:/Program Files/android

3. After the install is complete, the setup wizard downloads and installs additional components, including the Android SDK.
4.To install Kotlin language,Select kotlin while opening new project.

Appendix-B: Firebase Setup

1.Go to https://firebase.google.com/ and navigate to 'Go to Console'
2.Sign Up to your gmail account.
3. After signup create your new project on https://console.firebase.google.com/

Publication

Paper entitled **“Developing Smart ML Based Recommendation System”** is presented at **“3rd International Conference On Emerging Technologies In Data Mining And Information Security”** by **“Sakshi Naik”,“Sayali Phowakande”, “Arjun Rajput”, “Prof.Apeksha Mohite”, “Prof.Geetanjali Kalme”.**