CHAPTER- 1

INTRODUCTION

1.1 ABOUT THE ORGANISATION

4u software solutions is a leading solution provider for internet-based applications, software development, web designing, web development, SEO Services, Digital Marketing, Mobile Application development etc, established in year of 2012 having its Registered office at Satya Vihar, Bhubaneswar. The Software Firm has been promoted by some highly dedicated young software professional to provide total IT services under one roof. The company is expertise in Software Development, web designing, Web hosting, Mobile Application Development, Internet Marketing and other IT enabled services. It uses not only the latest technology but also it uses the technology as per the client's requirement.

Services Provided by them:

- Web Development
- Web Design
- Web hosting
- Software Development
- Mobile Application Development
- Internet Marketing

1.2 ABOUT THE PROJECT

The project Audio on Demand is basically Music Recommendation System or an audio streaming web application. Music recommender system is a system which learns from the users past listening history and recommends them songs which they would probably like to hear in future.

The Application allows the user to stream various genres of music. And at the same time, it will recommend music by analysing the user behaviour and the genre, user usually listen.

Currently, there are many music streaming services, like Pandora, Spotify, etc. which are working on building high-precision commercial music recommendation systems. These companies generate revenue by helping their customers discover relevant music and charging them for the quality of their recommendation service. Thus, there is a strong thriving market for good music recommendation systems.

The users' can access this application both in Desktop and Mobiles. The Application provides for a multifunctionality player which meant to play the audio as well as provides controls including play, pause, next, prev, volume control, audio motion effect and seek control. There is also a profile section where the user can sign up and restore all kind of history, playlist.

The Project uses the python and PHP as programming language. As python is open sourced, easy to understand, highly readable and has a powerful syntax it has been used. And backend functionality is accomplished using php. The Django Framework is used to designed the web pages and Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models. On the other hand, php has been implemented inside the CodeIgniter as it is faster as well secured.

The Recommend part is designed with Artificial Intelligence and machine learning tools using the Anaconda environment. The Recommendation engine is fully based on the client-based filtering model.

1.3 PLAN OF THE REPORT

Chapter 2 gives a detailed description about the problem definition, need of the project and feasibility analysis including operational, technical and Economic feasibility. Chapter 3 Provides the Introduction of the Software requirement specifications. It includes the Purpose, Scope, software, hardware specification and user requirements (functionalities) including Functional, Non-Functional and Performance Requirements. Chapter 4 contains the required analysis information with UML diagrams. Chapter 5 describes the module description, programming languages that has been used and Database design. Chapter 6 gives information about the implementation and testing including Unit, Integration, functional and System Testing. Chapter 7 gives information about the Conclusion and Foreseeable enhancements.

1.4 PLAN OF THE PROJECT

Planning plays vital role in any types of project. This project is also planned in such a manner and it has been illustrated as follows.

- 27-12-18 to 05-01-19. R & D of Audio Player
- 07-01-19 to 20-01-19. Audio Player Module
- 21-01-19 to 27-01-19. R & D of Content Module
- 30-01-19 to 24-02-19. Content Module
- 25-02-19 to 10-03-19. R & D of Encoding Module
- 11-03-19 to 25-03-19. Encoding Module
- 26-03-19 to 04-04-19. R & D of Recommendation Module
- 05-04-19 to 24-04-19. Recommendation Engine Module
- 26-04-19. Testing of the Project

CHAPTER - 2

PROBLEM DEFINITION AND FEASIBILITY ANALYSIS

INTRODUCTION

This Chapter describes the problem definition, need of the application and the feasibility analysis including Technical, Schedule, Operational and Economic feasibility and also the profitability.

2.1 PROBLEM DEFINITION

The purpose of our System is to design a project based on Audio on Demand Application which is intended to support Music Streaming.

In previous days people used to download music from the internet to their computer and listen them at their own convenience. However, the problem with this is one needs more storage space to store the audio files. In case of radio also people need to wait for what radio stations are playing.

Today's consumers want control over their listening experience. That means listening to what they want, when they want, on the device that's most convenient to them at the time. In particular, the younger audience prefers to listen to audio on demand rather than sit passively and let radio stations choose what to play them or download music to from internet and store them in their device.

2.2 NEED OF THE APPLICATION

Audio on demand can solve this problem most effectively. In audio streaming, people don't anymore need to download audio files to their computer. They can easily choose the songs from the website and listen. Also, with the integration of the recommendation engine, people will get recommended music as per their preference.

2.3 FEASIBILITY ANALYSIS

A feasibility study is an analysis that takes all of a project's relevant factors into account—including economic, technical, and scheduling considerations—to ascertain the likelihood of completing the project successfully.

Three key considerations involved in the feasibility analysis are:

- Technical feasibility
- Schedule Feasibility
- Operational feasibility
- Economic feasibility

2.3.1 Technical Feasibility

This project is technically feasible as all technologies used in this project are all easily available; all the python, php and Machine learning libraries used in this project are open source libraries easily available over the internet. So, the project is technically feasible.

2.3.2 Schedule Feasibility

This project is feasible as this project was completed in available time, the schedule that was decided before the start of the project has been strictly followed and hence the project was completed in the given time.

2.3.3 Operational Feasibility

The tool is a culmination of Audio file processing and Audio streaming techniques which can be easily accessed, the graphical user interface provides a very friendly interactive front end that makes it much easy for the user to understand and use the software. The project aims to do the recommend of music to the users as per their behaviour, to do so we used the machine learning tools of content-based recommendation system. So, it is operationally feasible.

2.3.4 Economic Feasibility

This project was economically feasible because we have used only freely available open source libraries which were available at free of cost, hence economically not much amount has been spent for this project.

CHAPTER - 3

SOFTWARE REQUIREMENT SPECIFICATION

INTRODUCTION

A software requirements specification (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements. It includes the use cases of how user is going to interact with software system. The software requirement specification document consists of all necessary requirements required for project development.

3.1 PURPOSE

The overall objective is to obtain a clear understanding about the need of the users and requirement that the website has to satisfy. Analysis involves activities like elicitation of user requirements, identifying the scope and feasibility of the system, identifying end users of the system and to perform various heuristic estimations. Information is organized and presented as the system requirement specification.

3.2 SCOPE

- In my project application, a user will get different taste of music (Genres).
- User will also get the information of the song.
- Depends upon the speed of the network, user will get the quality of the song.
- User will get the recommendation of music.

3.3 SYSTEM REQUIREMENTS

The tool requires the following hardware, software requirements at the developer and the following user characteristics:

Hardware: -

1. Intel® Core i5 CPU3.0 GHZ or above
2. 4GB RAM
3. 500GB Hard Disk Drive (HDD)
4. USB port 3.0
5. Monitor
6. Windows

Software: -

1. XAMPP Server
2. Anaconda
3. Django Framework
4. CodeIgniter
5. ATOMS
6. Brackets Text Editor

3.4.1 User Characteristics

- The user of this system must have this following characteristic:
- User must be a computer literate.
- As the interface of the system is English, the use must be well aware of English. User should be well aware of how to interact with a music portal

3.5 FUNCTIONALITIES

3.5.1 Functional Requirements

This project gives four major functionalities such as

- Playing Audio Content
- Audio Encoding
- Recommended Content
- Authentication and Storing

- **Playing Audio Content**

In the Playing Audio Content, the user will get various operations such as playing the audio file as per the user demand, the player has the various controls like

  - Playing
  - Next/Prev
  - Volume Control
  - Seek Control
  - Tempo Control

- **Audio Encoding**

  For Audio encoding there is function called FFMpeg is used. By using this command any format of audio file can be converted into a single format audio file as well as it also provides all the information about the file.

- **Recommended Content**

  The Content-Based Recommendation system has been used to recommend the content to the user while in the application. The Content-Based Recommender relies on the similarity of the items being recommended. The basic idea is that if you like an item, then you will also like a “similar” item. It generally works well when it's easy to determine the context/properties of each item. A content-based recommender works with data, as an instance music genre for the music dataset. Based on that data, the system makes suggestions to the user. As the user provides more inputs or takes actions on the recommendations, the engine becomes more and more accurate.

- **Authentication**

  The Authentication includes the functionalities such as

  - User Registration
  - User Login

3.5.2 Non-Functional Requirements

- Pages should load within a second.
- Data should load without page refreshment.
- It should work very fast without any buffer.

3.5.3 Performance Requirement

- The Audio quality should be good.
- The UI should be friendly.
- Recommendation should be relevant.

.

CHAPTER - 4

SYSTEM ANALYSIS

INTRODUCTION

This Chapter describes the UML diagrams such as Use case, Class, Activity and Data flow diagram.

4.1 UML DIAGRAMS

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: A Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

To represent the system design, we have used the three following UML Diagrams:

- Use case Diagrams
- Class Diagrams
- Activity Diagrams
- Data Flow Diagram

4.1.1 Use case diagram

Use case diagrams are usually referred to as behaviour diagram used to describe a set of actions that some system or systems(subject) should or can perform in collaboration with one or more external users of the system(actor).

In our report we have shown Use Case diagrams, dividing the entire system into two parts, for each of which we have given a separate diagram and one to overview the entire structure. These are the Use Case Diagrams are provided, as listed below: fig I, and Fig II

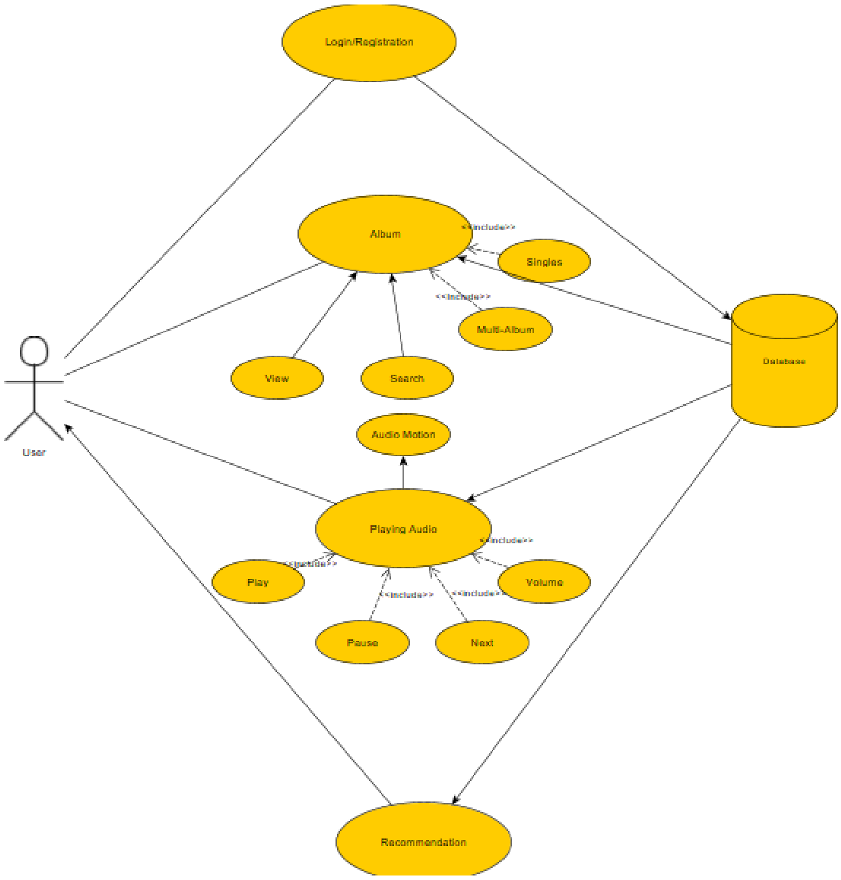


Fig I. Usecase Diagram for the Music Player

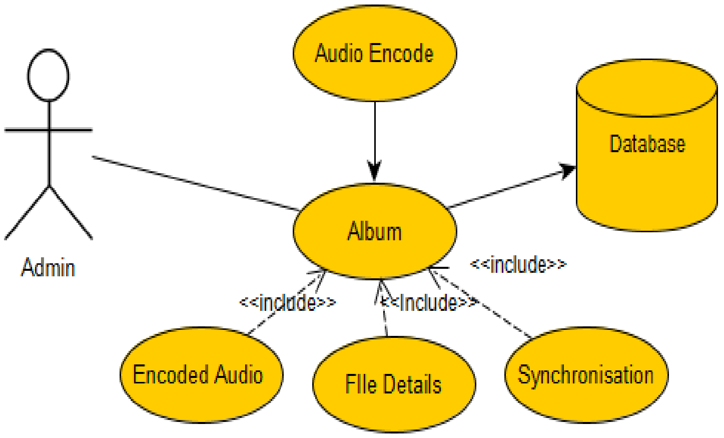


Fig II. Use case Diagram for Audio Encoding

4.1.2 Class Diagrams

A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

1. Content (UML Class Diagrams)

The content model visualizes the domain relevant information for the Web system that mainly comprises the content of the Web application. In our example the information is provided by the classes Album, Artist and Song. A UML class diagram and UML plain classes are used to model the content.

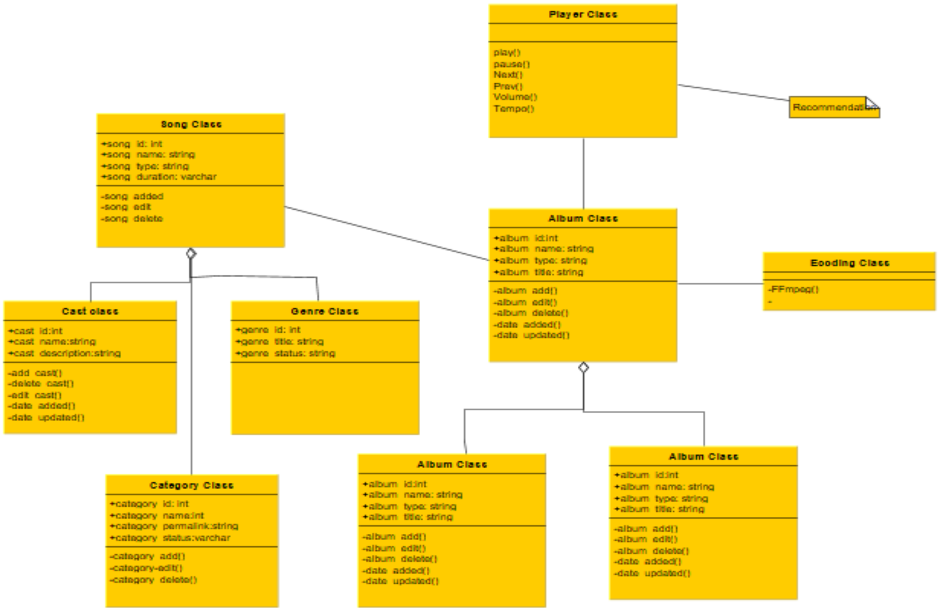
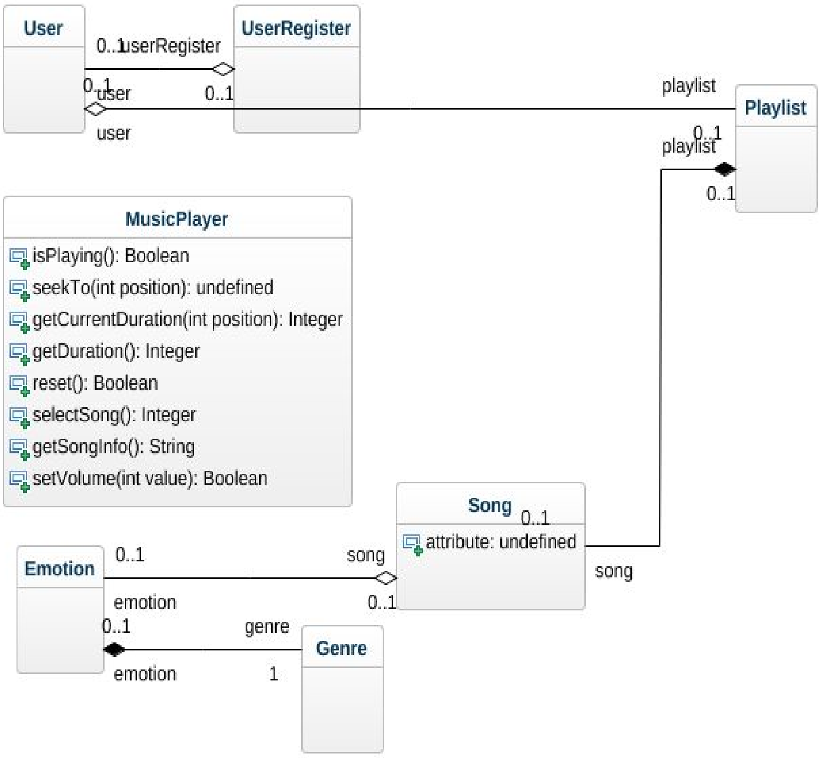


Fig III. class diagram for Content Module

2. Player Model (Class Diagrams)

A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.



4.1.3 Activity Diagrams

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join.

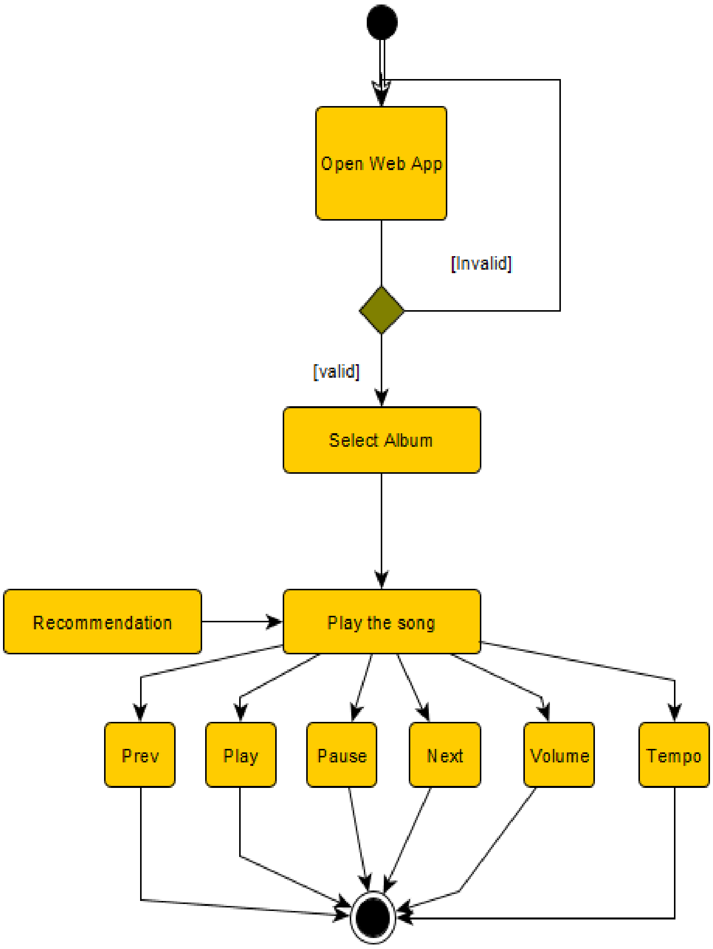


Fig IV. Activity diagram for the Audio on Demand

4.1.4 Dataflow Diagram

A data flow diagram (DFD) is a way of representing a flow of a data of a process or a system (usually an information system) The DFD also provides information about the outputs and inputs of each entity and the process itself. A data flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

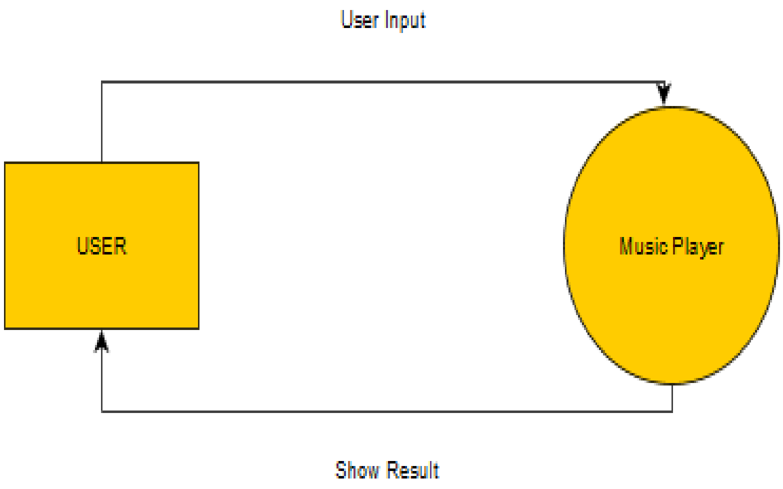


Fig V. DFD for music player

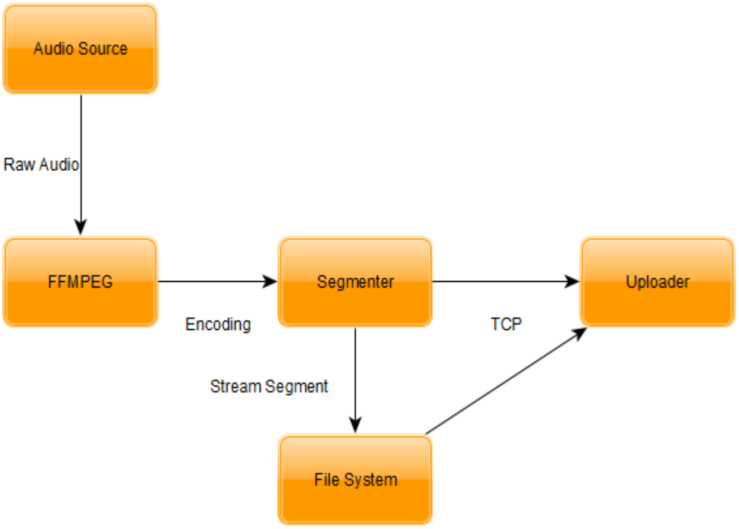


Fig VI. DFD for Audio Encoding

CHAPTER - 5

SYSTEM DESIGN

INTRODUCTION

This chapter describes the module description and Database design.

Preliminary Module Description

The Audio on Demand is an Audio streaming web application. As per the module description, The Audio player will be meant for interaction with the end-users. A distinction is made between users and registered users. A user becomes a registered user by logging in. Unregistered users can register with a username that has not been taken by another user and a freely chosen password. Every user can search for albums by their name. Other search methods are not offered. The album detail pages show the title of the album, the name of the artist, the list of songs. For a register user, A profile session will be there. All kind of activities will place inside the profile session. The register user can log out by click on the logout button.

5.1 MODULE DESCRIPTION

This project is splitted into four modules such as

1. Audio Player Module
2. Content Module
3. Encoding Module
4. Recommendation Engine Module

5.1.1 Audio Player Module

In this module an Audio Player is designed, which will control all the operations like play the audio, pause, next, seek control, volume control, Tempo control.

INPUT- User has to select a song to be played in the website.

OUTPUT- Audio Player will be appear and play the selected song. There are various controls will be shown to the user as mentioned above.

5.1.2 Content Module

It Is the core part of the project. Contents adding, content removing, modification of content and the user registration part are done in this module.

INPUT 1- There is no input for the users, Admin is only allowed to add the contents.
INPUT 2- The User can input the user name and password
OUTPUT 1- All the song categories and list of the song will be shown to the user.
OUTPUT 2- The Registration or Sign in done successfully.

5.1.3 Encoding Module

An audio encoding is used for the compression or decompression of digital audio data from a live stream media. The purpose of using an audio codec is to effectively reduce the size of an audio file without affecting the quality of the sound. This encoding module developed using FFmpeg command and php.

INPUT- For user, no input. Admin is only allowed to insert the song.
OUTPUT- Any format of song will be encoded to a single mp3 format.

5.1.4 Recommendation Engine Module

The aim of this module is to recommend contents to the users by following their behaviour in the website.
This module has developed using the Content-Based recommendation system.

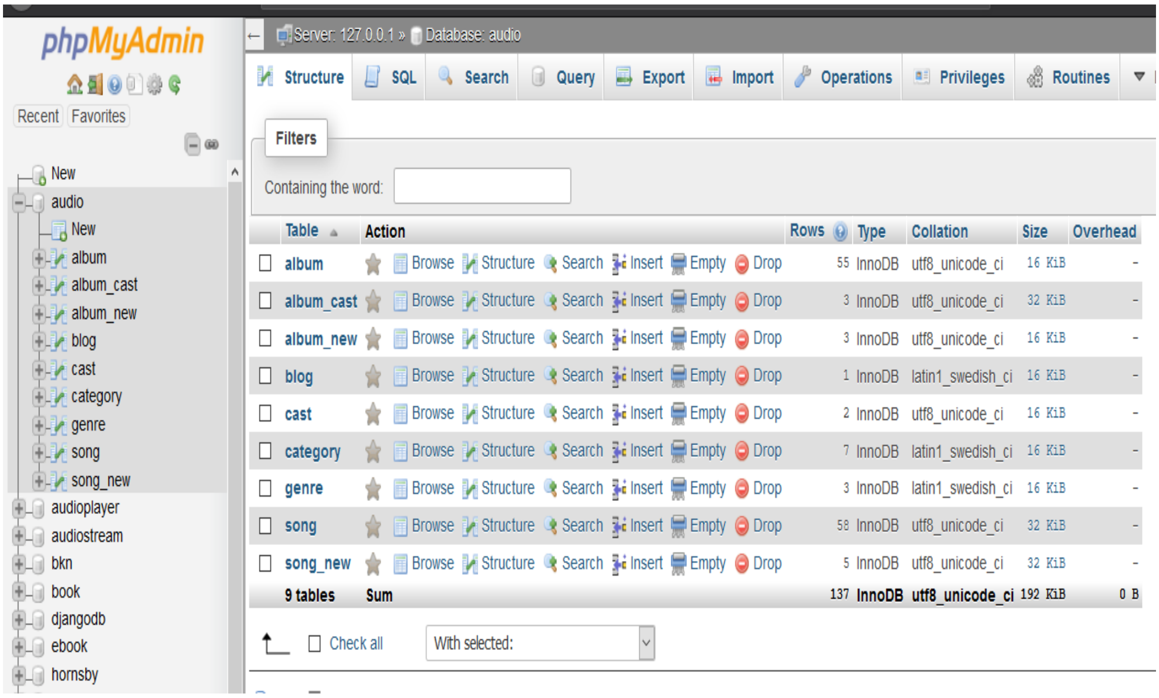
INPUT- There is no input.

OUTPUT- Recommendation of content will be appeared by analysing the user interested song.

5.3 DATABASE DESIGN

The database used for this project is **audio**, which is the most popular open-source relational database management system developed by Oracle Corporation. MySQL is one of the best RDBMS being used for developing various web-based software applications.

The MYSQL Server database is used to store all the song file, categories, albums, blogs, etc and the deign have been shown below.



Containing the word:

	Table	Action	Rows	Type	Collation	Size	Overhead
□	album	★ Browse Structure Search Insert Empty Drop	55	InnoDB	utf8_unicode_ci	16 KiB	-
□	album_cast	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_unicode_ci	32 KiB	-
□	album_new	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_unicode_ci	16 KiB	-
□	blog	★ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
□	cast	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_unicode_ci	16 KiB	-
□	category	★ Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	16 KiB	-
□	genre	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
□	song	★ Browse Structure Search Insert Empty Drop	58	InnoDB	utf8_unicode_ci	32 KiB	-
□	song_new	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8_unicode_ci	32 KiB	-
	9 tables	Sum	137	InnoDB	utf8_unicode_ci	192 KiB	0 B

□ Check all With selected:

CHAPTER 6

TESTING AND SYSTEM IMPLEMENTATION

6.1 TESTING

Testing is the process of evaluating a system or its components with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

There are four different testing methods are successfully explained below.

6.1.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. All the modules have been individually tested. These modules have been designed to satisfy particular requirements; this test was performed to check if the desired outcomes were available for several data sets. The modules were individually tested and all the test cases have passed successfully with no default occurrences.

- **Unit Testing for Audio player**

  Objective- To ensure the audio player is correctly working.

S. no	**Event**	**Action/ Attribute and Value**	**Expected Result**	**Result**
1	Verify that the music player having the all operational buttons and control bars.	Play Button, Pause Button, Next, Prev, Seek Bar, Volume control. Button: Boolean (1- Active, 0- Inactive)	All the Buttons and controls are found.	Pass

2	Verify the music player can play the music.	Select song and press the play button. Play Button: Boolean	Music playing Successfully.	Pass
3	Verify all the operations and controls bar are working.	Buttons Progress Container	Operation and controls are working successfully.	Pass
4	Verify the duration of the song.	Track_ timer	Duration showing successfully	Pass
5	Verify the tempo control.	Motion Control	Tempo customization Successfully working.	Pass

➢ **Unit testing- User Login**

Objective- To ensure the user can login by providing User ID and Password.

➢ **S.no**	**Event**	**Action / Attribute and Value**	**Expected Result**	**Result**
1	Verify that the User Id and Password enter by user and match the data in the database when user click ‘Login’ Button.	Login Id: Valid ID Password: Valid Password	Login Successfully	Pass
2	Verify the Invalid Id and Password that enter by user and match with the data store in the	Login Id: Invalid ID Password: Invalid Password	Login Failed and prompt out the error message to user.	Pass

	database when user click “Login” Button.	Or Login Id: Valid ID Password: Invalid Password		
3	Verify the situation that user doesn’t entry any value.	Login Id: null Password: null	Login Failed and prompt out the error message	Pass
4	Forgot User Id or Password (RESET)	Information Verified.	Prompt the user Id and Password sent to you mail id.	Pass

- **Unit Testing- Registration of User**

  Objective- To Ensure the Registration page is working properly

S.no	**Event**	**Action/ Attribute and Value**	**Expected Result**	**Result**
1	Registration for new user.	System user information.	Prompt Registration Successfully.	Pass
2	Registration Using Facebook	User Information from Facebook	Redirect to Home with Successful registration.	Pass
3.	Register without providing any information.	All Blanks.	Prompt require field error message	Pass
4	Register with the duplicate information(mail/mob)	Enter duplicate data.	Prompt the data already exist(mail/mob)	Pass

- **Unit Testing- Audio Encoding**

Objective- To ensure that the audio codec working properly

S.no	**Event**	**Action**	**Expected Result**	**Result**
1	Verify all kind of audio format can be played.	The ffmpeg commands are used to convert all the format into one unique format.	All kind of format played successfully	Pass
2	Verify the audio bitrate.	The ffmpeg command will help the audio to be load in different bitrate as per the user choice.	User can play with different bitrate.	Pass
3	Verify the effect of network connectivity.	The ffmpeg command will help the audio to be played in all possible network connectivity.	The bitrate changes as per the data rate.	Pass

- **Unit Testing – Recommendation System**

Objective- To ensure that the recommendation engine working properly.

S.no	Event	Action	Expected Result	Result
1	Verify the Datasets having all the required attributes.	From the dataset the engine will fetch all the information so that recommendation can be done.	The Attributes are perfect.	Pass
2	Verify the recommend contents are relevant.	When user listening a song, same type of song will recommend to the user.	By fetching the genre of the song, Recommendation will happen.	Pass

6.1.2 Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation.

Both the individual modules, Content modules and Audio encoding modules were integrated and tested together. The testing was done in three steps:

- ✓ Successful integration was verified when the system and software both functioned properly after integration
- ✓ Successful Inter passage of data verified that the integration was not only physically feasible, but also logically proper.
- ✓ The software after integration was finally tested to verify its compatibility with the hardware system.
- ✓ The tests remained successful for all the test cases with default occurrences.

6.1.3 Functional Testing:

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing. Functions are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. Each functionalities of this application are tested by providing give set of inputs in order to know the actual behaviour of the applications and thereafter comparing with the expected results as per the given specifications. Functional testing mainly involves:

- ✓ Verifying user interfaces
- ✓ Verifying end to end work flows

The functions of individual modules were separately tested to verify if they were all perfectly working. These functions were taken as units and provided required input data, which was only declared successful on obtaining desired output, within desired period of time and resources. These tests can be considered as a Unit testing if Unit testing is considered as Integration of several functions.

Following are a list of few functions tested:

- ✓ Storage and Retrieval of data to and from database
- ✓ Functioning of buttons as and when required
- ✓ The Proper function of Audio encoding.
- ✓ The proper function of Recommendation System.
- ✓ All the requirements are tested successfully with given data set with default occurrences.

6.1.4 System Testing:

System testing is a level of software testing where complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. Usually software is only one element of a larger computer-based system. Ultimately, software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

The created standalone application is tested among various systems with different Windows Operating System and for both 32 bit and 64-bit systems. The software has some basic requirement to be function able that states the SRS Document. The derive report stated the following requirements:

The tool requires the following hardware, software requirements at the developer as well as the user's end and the following user characteristics:

Hardware: -

1. Intel® Core i5 CPU3.0 GHZ or above
2. 4GB RAM
3. 500GB Hard Disk Drive (HDD)
4. USB port 3.0
5. Monitor
6. Windows

Software: -

1. XAMPP Server
2. Django Framework
3. CodeIgniter
4. ATOMS
5. Brackets Text Editor

6.1.5 Acceptance Testing:

Acceptance testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery. Formal testing with respect to user needs, requirements, and business processes conducted to determine the acceptance criteria and enable the user, customers or other authorized entity to determine whether or not to accept the system. The software since was brought up for the fulfilment of the company’s personal requirements and not marketing purpose, the Acceptance testing was successfully passed once it functioned according to the requirements of the authorities.

6.2 SYSTEM IMPLEMENTATION

How all the modules have been developed and what are the languages are used to implement and design them are discussed inside the system implementation.

6.2.1 Audio Player module

In this module the audio player has been designed and developed. This module has only the front-end part. To develop the front-end part Django, HTML5, CSS, JavaScript, jQuery, Rest API are used.

Functionalities- Play, Pause, Next, prev, Volume, Seek, Audio motion.

6.2.2 Content Module

This is the core part of the module; All the CRUD operations of content have been deployed in this module.

Front-end- To design and implement the front-end part Django, HTML5, CSS, JavaScript, Bootstrap has been used.

Back-end- PHP CodeIgniter has been used.

Functionalities- Content adding, Content modification, content deletion.

6.2.3 Encoding Module

In this module the audio encoding has been done. It helps in converting audio file format into single format. And it also provides all the information of the audio file.

Php and FFmpeg has been implemented to develop this module.

6.2.4 Recommendation engine module

In this module, the content-based filtering model has been implemented. The aim of this recommendation engine is to recommend songs to the user by following the users' preferred song.

Django and Machine learning has been implemented to develop this module.

6.2.5 The tools and Technology

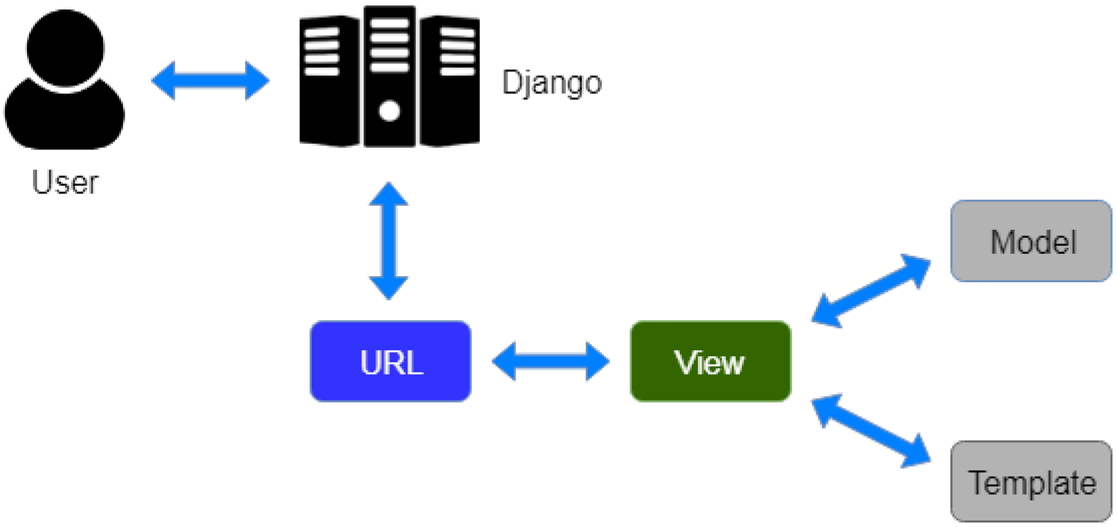
Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Django is Ridiculously fast, Reassuringly Secure and Exceedingly Scalable.

Django MVT- The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data.

See the following picture shows the Django MVT based controls flows.



Virtual Environment

A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated python virtual environments for them.

CodeIgniter

CodeIgniter is loosely based on the popular model–view–controller (MVC) development pattern. While controller classes are a necessary part of development under CodeIgniter, models and views are optional. CodeIgniter can be also modified to use Hierarchical Model View Controller (HMVC) which allows developers to maintain modular grouping of Controller, Models and View arranged in a sub-directory format.

FFmpeg

FFmpeg is the leading multimedia framework, able to decode, encode, transcode, mux, demux, stream, filter and play pretty much anything that humans and machines have created. It supports the most obscure ancient formats up to the cutting edge. No matter if they were designed by some standards committee, the community or a corporation. It is also highly portable. FFmpeg compiles, runs, and passes our testing infrastructure FATE across Linux, Mac, Windows etc. under a wide variety of build environments, machine architecture, and configurations.

• **Recommendation System (Content-Based Filtering)**

The Content-Based Recommender relies on the similarity of the items being recommended. The basic idea is that if you like an item, then you will also like a “similar” item. It generally works well when it's easy to determine the context/properties of each item. A content-based recommender works with data that the user provides, either explicitly movie/music ratings for the Movie/Music dataset. Based on that data, a user profile is generated, which is then used to make suggestions to the user. As the user provides more inputs or takes actions on the recommendations, the engine becomes more and more accurate.

Overall, here are the pros of using content-based recommendation:

* Can recommend to users with unique tastes. * Can recommend new & unpopular items. * Can provide explanations for recommended items by listing content-features that caused an item to be recommended (in this case, movie(music) genres)

However, there are some cons of using this approach:

* Finding the appropriate features is hard. * Does not recommend items outside a user’s content profile. * Unable to exploit quality judgments of other users.

CHAPTER 7

CONCLUSION AND FORESEEABLE ENHANCEMENT

6.1 CONCLUSION

In recent time, Data Science and Data Driven decision making is highly getting popular. Currently every application is upgrading to data driven as an instance from the data predictions to self-driven car, everywhere the Machine learning algorithms has been implementing. My project Audio on Demand is based on the Machine Learning Recommendation system. To design the front-end part, I have also learned and implement the html, CSS and JavaScript. For the Back-end development I also learned and implement the CodeIgniter which is an MVC system and really very easy and secured. A FFmpeg command has been implemented for audio encoding purpose. Django framework, which is very popular nowadays to develop the web application using python and ML codes. It’s really very good as I got chance to research and do some implementation of machine learning with this project: Audio on Demand, a Music Recommendation System. There are many different approaches to this problem and I got to know some algorithms in detail. By manipulating the dataset, changing the learning set and testing set, changing some parameters of the problem and analysing the result, I earn a lot of practicing skills. I have also faced a lot of problem in dealing with the dataset, how to explore it and also had difficulties in some programming details. As Machine learning was first time for me, I have gone through various websites and tutorials and Afterall the work has done.

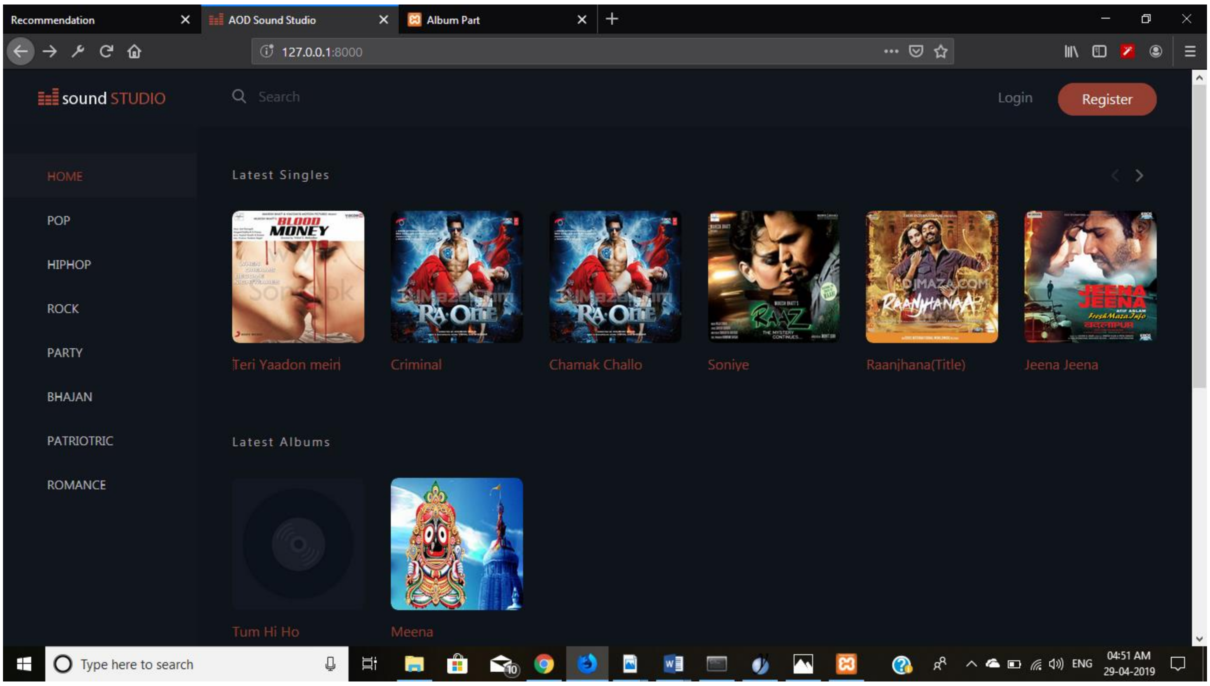
2 FORESEEABLE ENHANCEMENTS

1. Still more operations can be added to the application.
2. Other recommendation filter technology like Collaborative filter can be implement for the better recommendation.
3. Cloud Technology can be implemented on this project for storing the data.
4. This application can be more fledged so that can be used for commercial purpose.

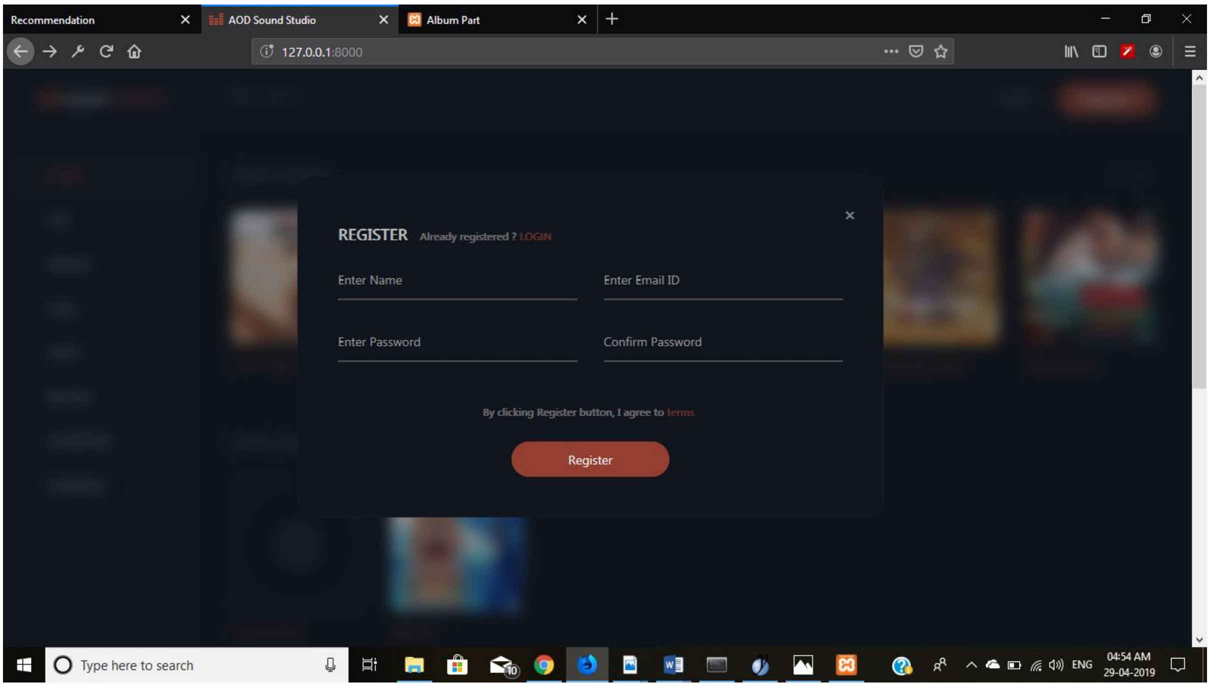
APPENDIX- A

SCREEN SHOTS

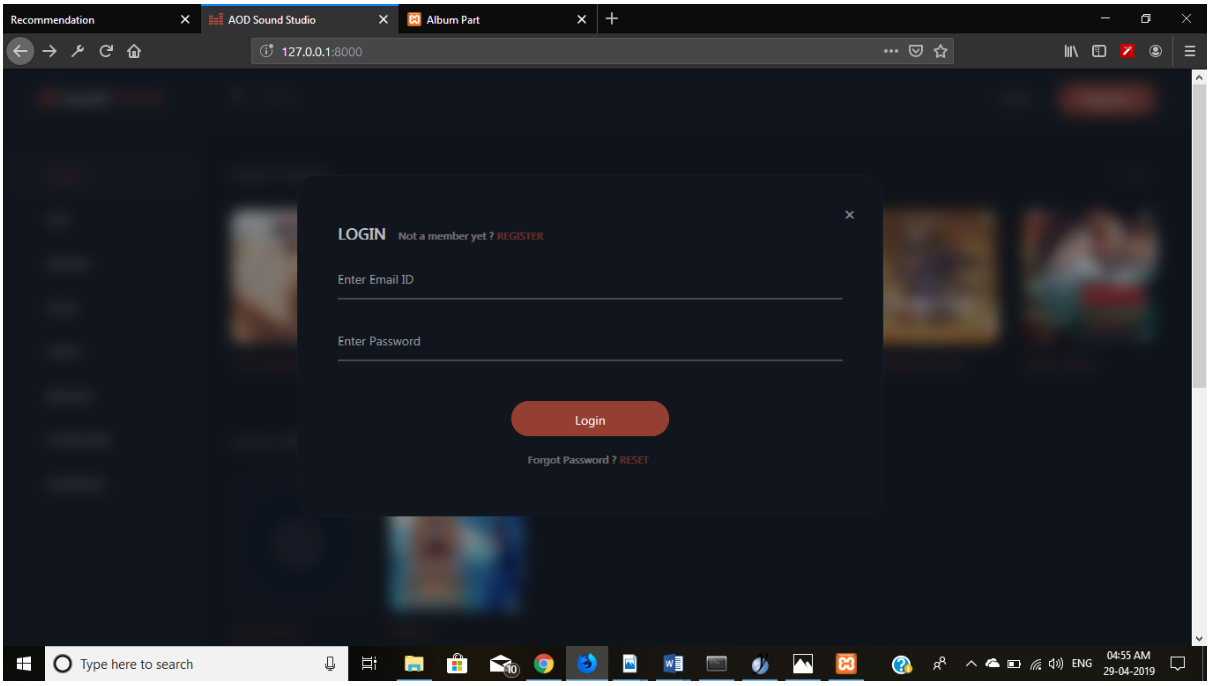
Home Page



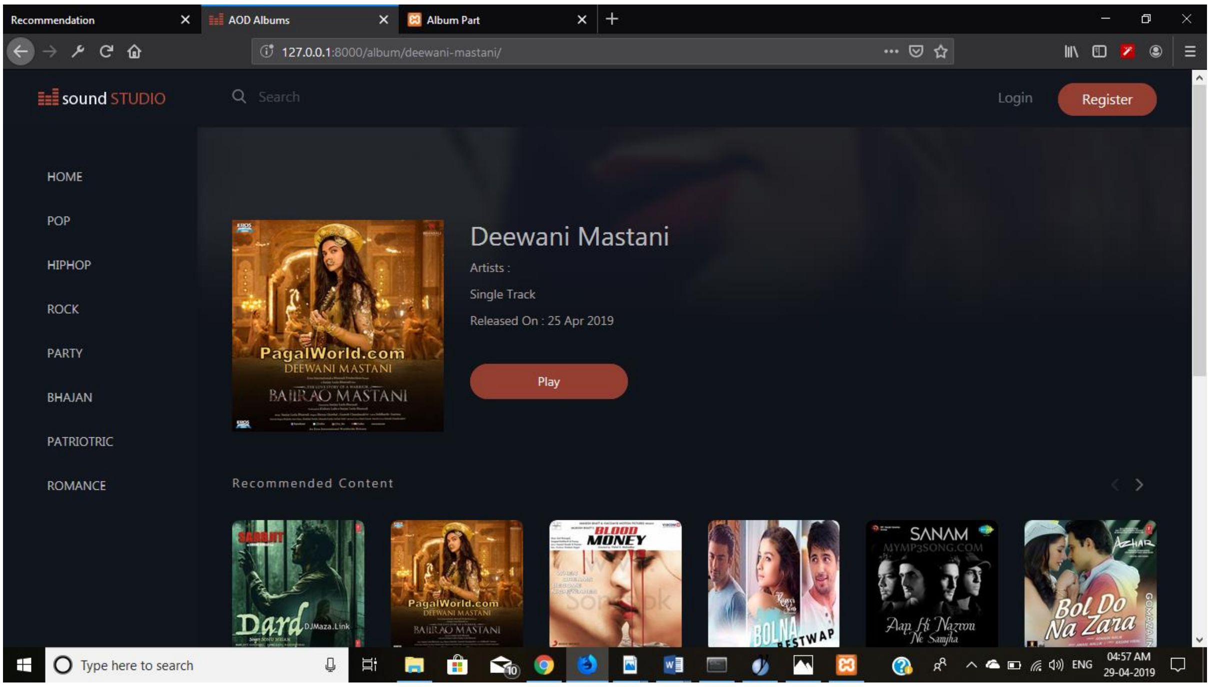
Registration Page



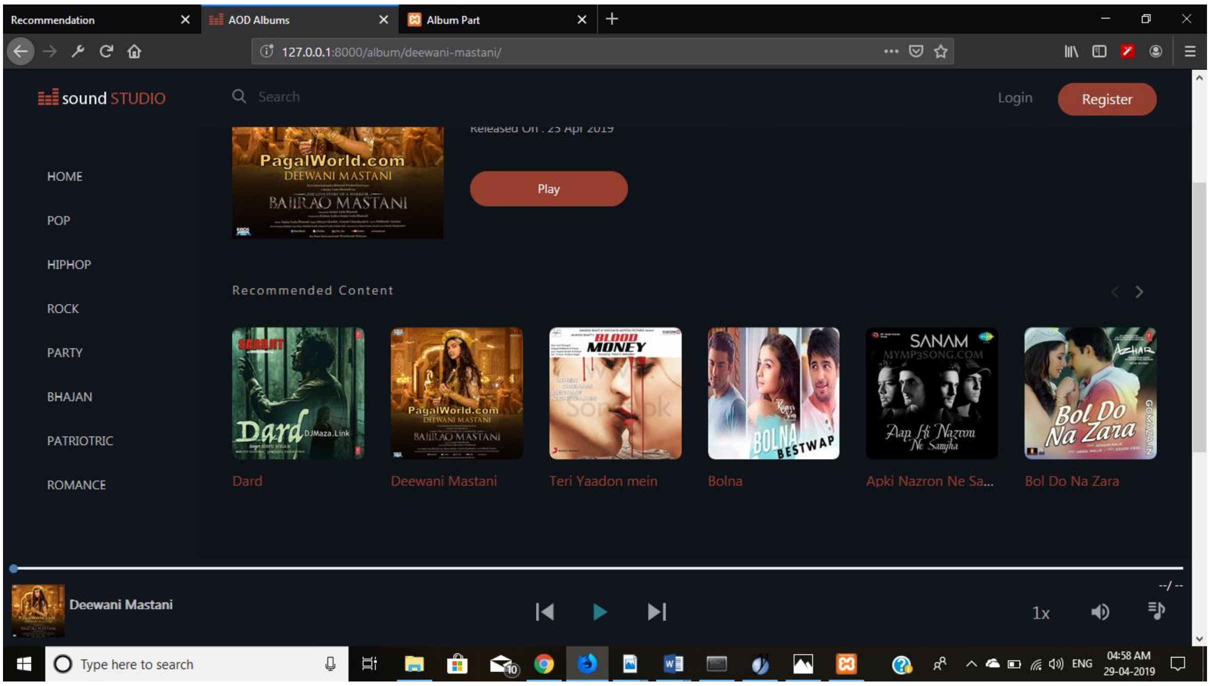
Log in Page



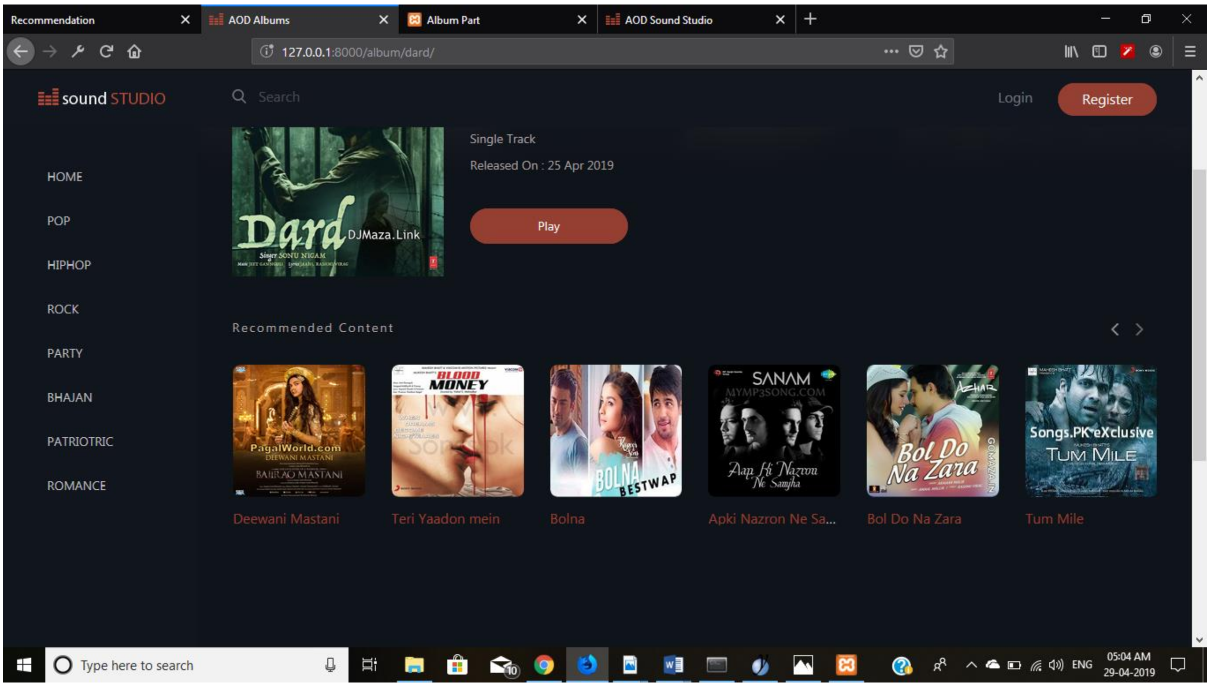
Player Part



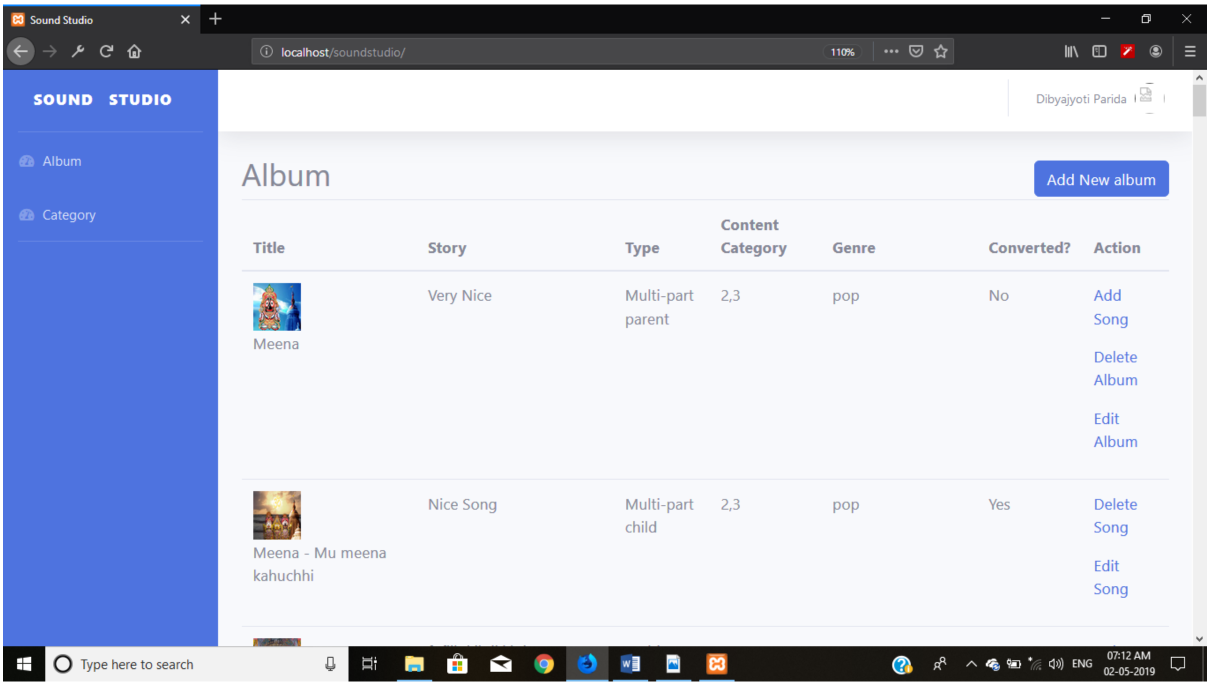
Audio Player



Recommended Content

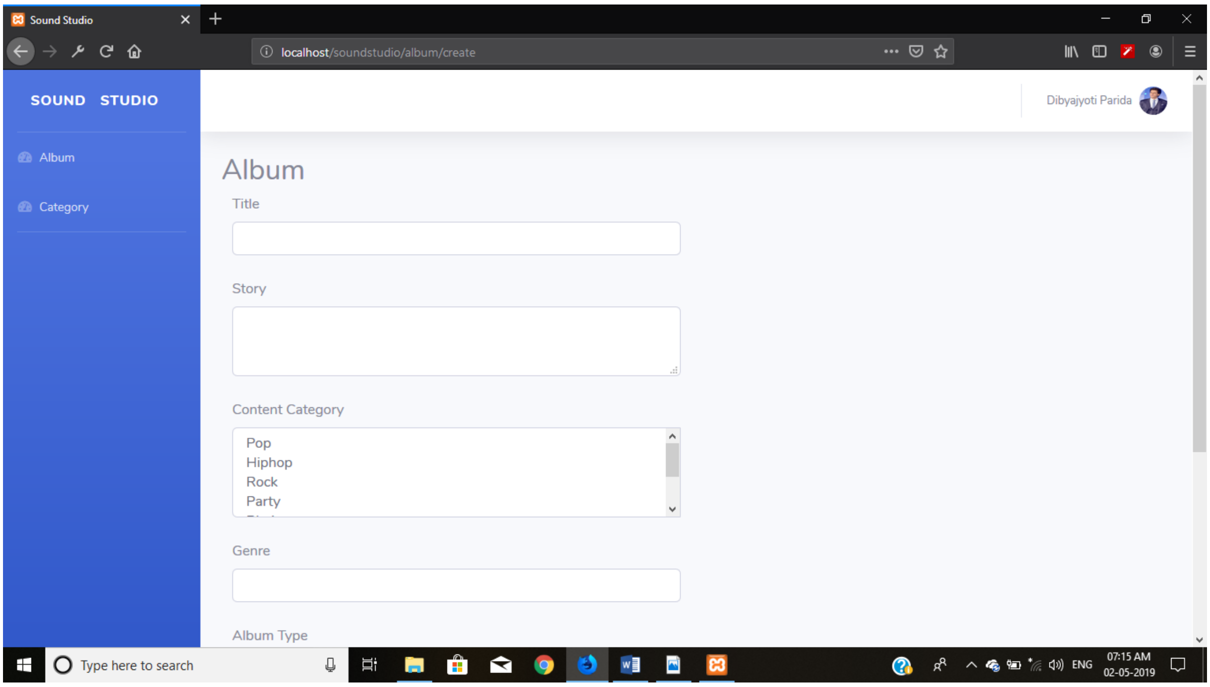


Admin Part



Album

Title	Story	Type	Content Category	Genre	Converted?	Action
Meena	Very Nice	Multi-part parent	2,3	pop	No	Add Song Delete Album Edit Album
Meena - Mu meena kahuchhi	Nice Song	Multi-part child	2,3	pop	Yes	Delete Song Edit Song



Album

Title

Story

Content Category

Pop
Hiphop
Rock
Party

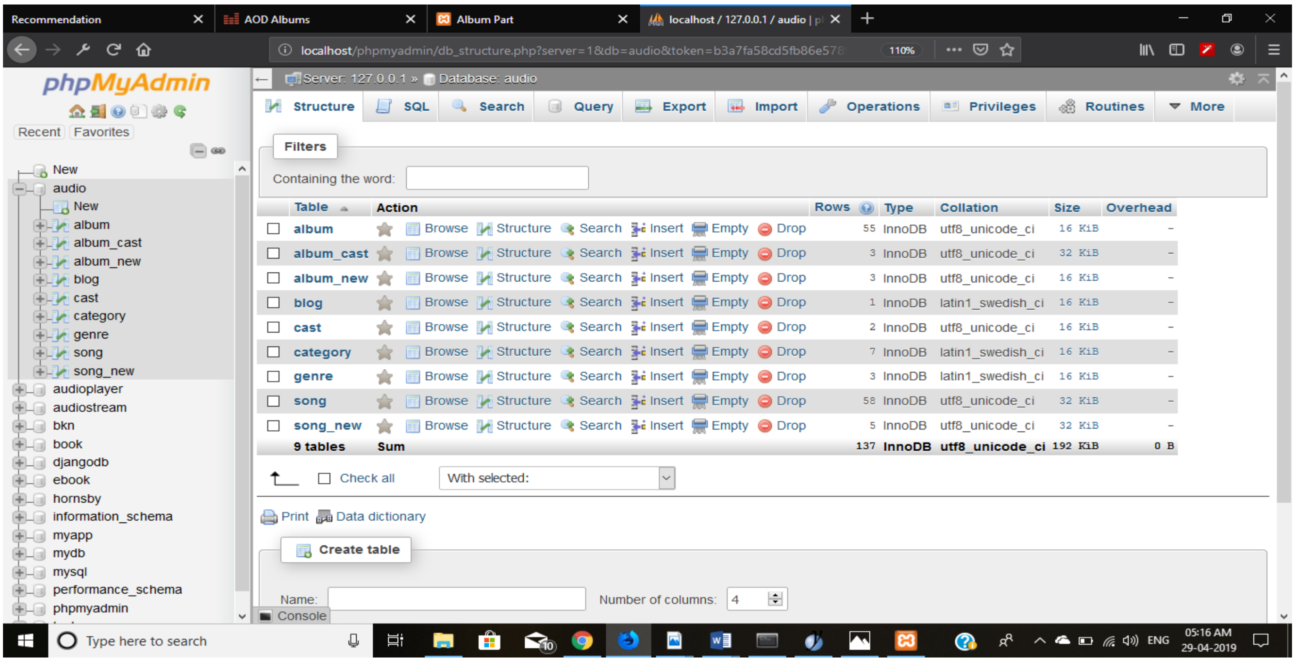
Genre

Album Type

APPENDIX- B

DATABASE DESIGN

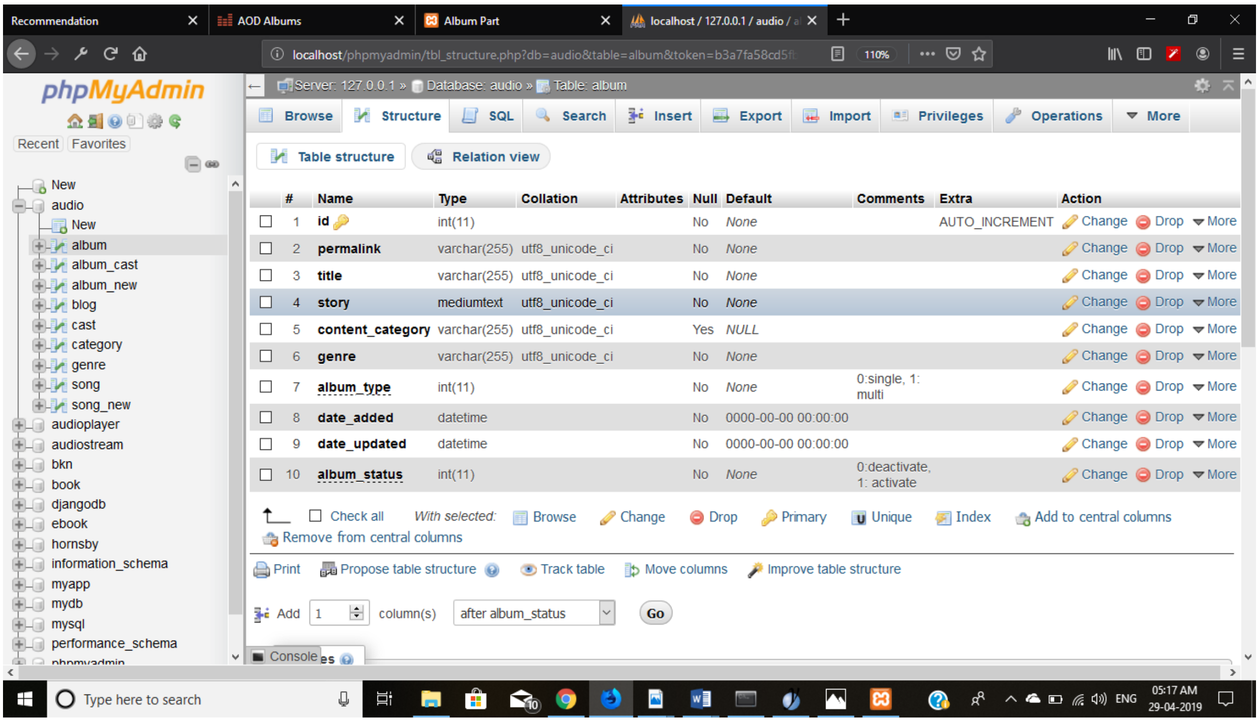
Database



Server: 127.0.0.1 » Database: audio

	Table	Action	Rows	Type	Collation	Size	Overhead
□	album	☆ Browse Structure Search Insert Empty Drop	55	InnoDB	utf8_unicode_ci	16 KiB	-
□	album_cast	☆ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_unicode_ci	32 KiB	-
□	album_new	☆ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_unicode_ci	16 KiB	-
□	blog	☆ Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
□	cast	☆ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_unicode_ci	16 KiB	-
□	category	☆ Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	16 KiB	-
□	genre	☆ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
□	song	☆ Browse Structure Search Insert Empty Drop	58	InnoDB	utf8_unicode_ci	32 KiB	-
□	song_new	☆ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8_unicode_ci	32 KiB	-
	9 tables	Sum	137	InnoDB	utf8_unicode_ci	192 KiB	0 B

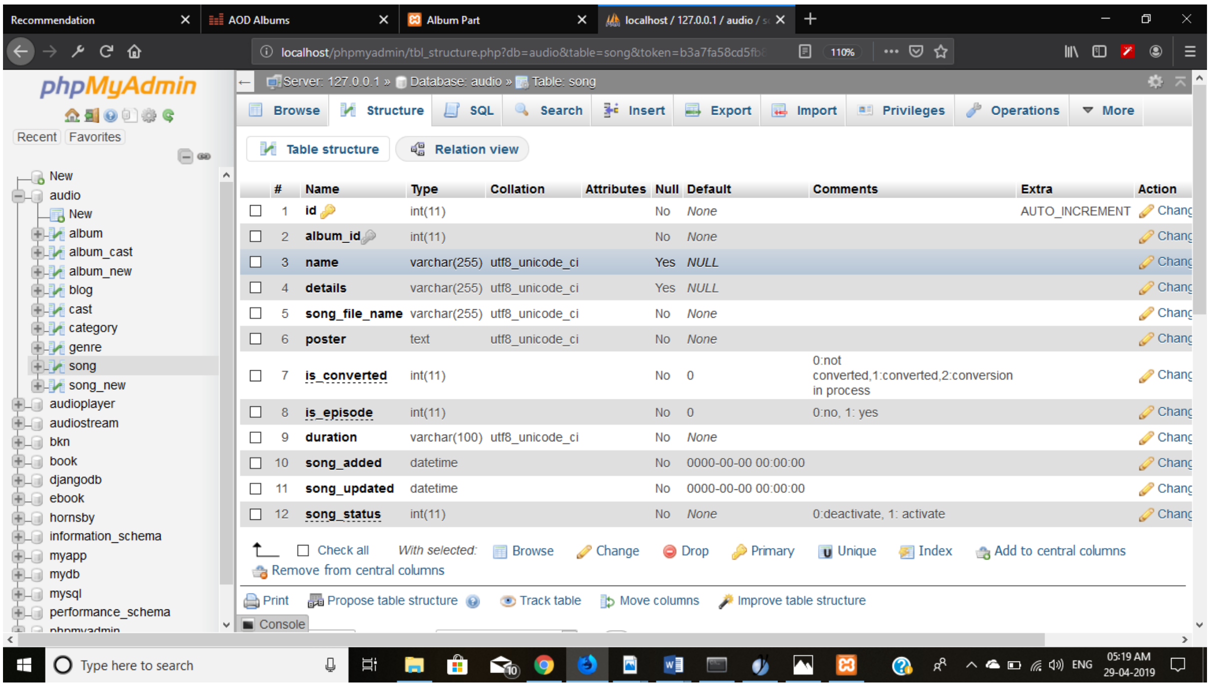
Album Table



Server: 127.0.0.1 » Database: audio » Table: album

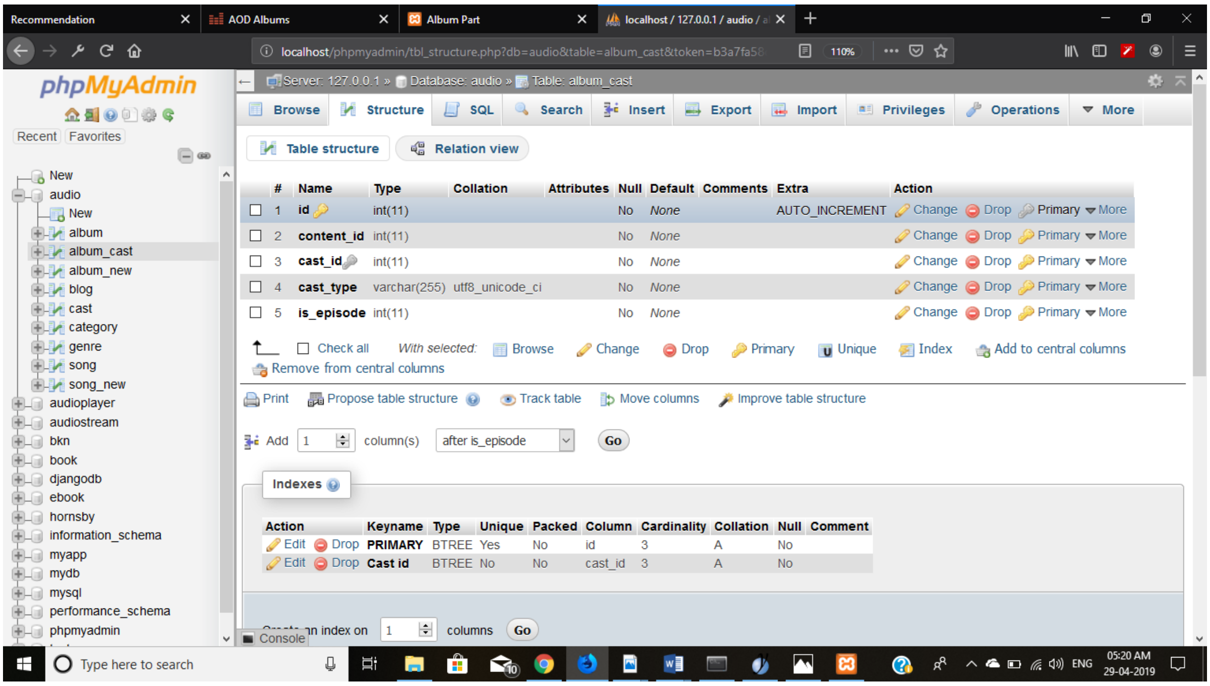
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	permalink	varchar(255)	utf8_unicode_ci		No	*None*			Change Drop More
□	3	title	varchar(255)	utf8_unicode_ci		No	*None*			Change Drop More
□	4	story	mediumtext	utf8_unicode_ci		No	*None*			Change Drop More
□	5	content_category	varchar(255)	utf8_unicode_ci		Yes	*NULL*			Change Drop More
□	6	genre	varchar(255)	utf8_unicode_ci		No	*None*			Change Drop More
□	7	album_type	int(11)			No	*None*	0:single, 1: multi		Change Drop More
□	8	date_added	datetime			No	0000-00-00 00:00:00			Change Drop More
□	9	date_updated	datetime			No	0000-00-00 00:00:00			Change Drop More
□	10	album_status	int(11)			No	*None*	0:deactivate, 1: activate		Change Drop More

Song Table



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**id**	int(11)			No	*None*		AUTO_INCREMENT	Chang
□	2	**album_id**	int(11)			No	*None*			Chang
□	3	**name**	varchar(255)	utf8_unicode_ci		Yes	*NULL*			Chang
□	4	**details**	varchar(255)	utf8_unicode_ci		Yes	*NULL*			Chang
□	5	**song_file_name**	varchar(255)	utf8_unicode_ci		No	*None*			Chang
□	6	**poster**	text	utf8_unicode_ci		No	*None*			Chang
□	7	**is_converted**	int(11)			No	0	0:not converted,1:converted,2:conversion in process		Chang
□	8	**is_episode**	int(11)			No	0	0:no, 1: yes		Chang
□	9	**duration**	varchar(100)	utf8_unicode_ci		No	*None*			Chang
□	10	**song_added**	datetime			No	0000-00-00 00:00:00			Chang
□	11	**song_updated**	datetime			No	0000-00-00 00:00:00			Chang
□	12	**song_status**	int(11)			No	*None*	0:deactivate, 1: activate		Chang

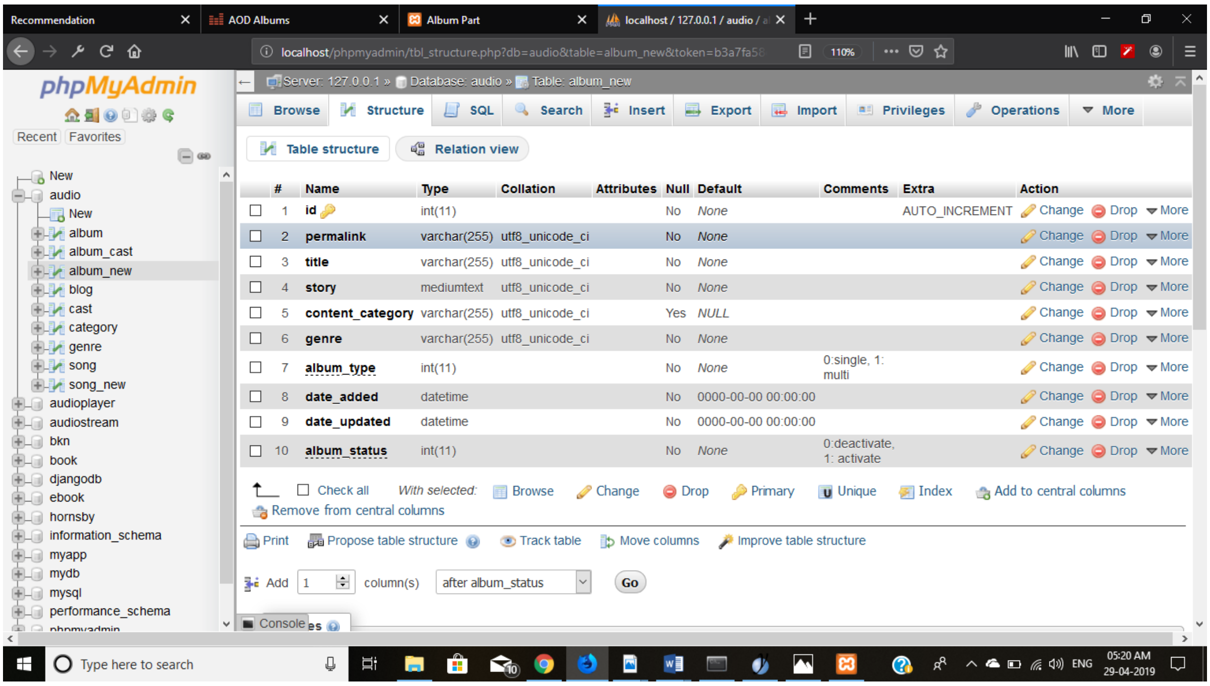
Album cast



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop Primary More
□	2	**content_id**	int(11)			No	*None*			Change Drop Primary More
□	3	**cast_id**	int(11)			No	*None*			Change Drop Primary More
□	4	**cast_type**	varchar(255)	utf8_unicode_ci		No	*None*			Change Drop Primary More
□	5	**is_episode**	int(11)			No	*None*			Change Drop Primary More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	**PRIMARY**	BTREE	Yes	No	id	3	A	No	
Edit Drop	**Cast id**	BTREE	No	No	cast_id	3	A	No	

Album new Table



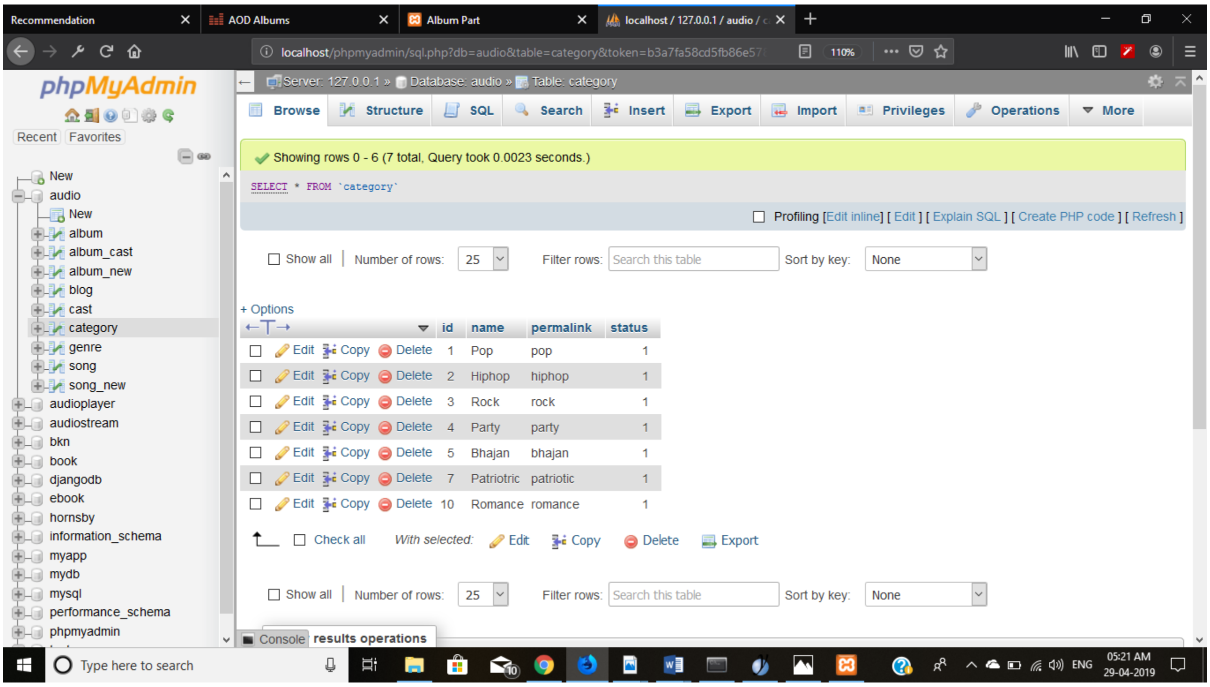
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
□	2	permalink	varchar(255)	utf8_unicode_ci		No	None			Change Drop More
□	3	title	varchar(255)	utf8_unicode_ci		No	None			Change Drop More
□	4	story	mediumtext	utf8_unicode_ci		No	None			Change Drop More
□	5	content_category	varchar(255)	utf8_unicode_ci		Yes	NULL			Change Drop More
□	6	genre	varchar(255)	utf8_unicode_ci		No	None			Change Drop More
□	7	album_type	int(11)			No	None	0:single, 1: multi		Change Drop More
□	8	date_added	datetime			No	0000-00-00 00:00:00			Change Drop More
□	9	date_updated	datetime			No	0000-00-00 00:00:00			Change Drop More
□	10	album_status	int(11)			No	None	0:deactivate, 1: activate		Change Drop More

□ Check all *With selected:* Browse Change Drop Primary Unique Index Add to central columns Remove from central columns

Print Propose table structure Track table Move columns Improve table structure

Add 1 column(s) after album_status Go

Category Table



Showing rows 0 - 6 (7 total, Query took 0.0023 seconds.)

```
SELECT * FROM `category`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

		id	name	permalink	status
□	Edit Copy Delete	1	Pop	pop	1
□	Edit Copy Delete	2	Hiphop	hiphop	1
□	Edit Copy Delete	3	Rock	rock	1
□	Edit Copy Delete	4	Party	party	1
□	Edit Copy Delete	5	Bhajan	bhajan	1
□	Edit Copy Delete	7	Patriotric	patriotic	1
□	Edit Copy Delete	10	Romance	romance	1

□ Check all *With selected:* Edit Copy Delete Export

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

SAMPLE CODES

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
<meta name="theme-color" content="#12171f" />
<!-- The above 3 meta tags *must* come first in the head; any other head content must come
*after* these tags -->
<title>AOD {{title}}</title>
{% load staticfiles %}
<link rel="shortcut icon" href="{% static 'assets/favicon.ico' %}" type="image/x-icon">
<!-- Bootstrap and vendor css -->
<link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">
<link rel="stylesheet" href="{% static 'vendor/custom-scrollbar/jquery.mCustomScrollbar.css' %}">
<link rel="stylesheet" href="{% static 'vendor/owlcarousel/owl.carousel.css' %}">
<!--Main CSS-->
<link rel="stylesheet" href="{% static 'css/main.css' %}">
<link rel="stylesheet" href="{% static 'css/responsive.css' %}">
<link rel="stylesheet" href="{% static 'css/audioplayer.css' %}">


<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
<![endif]-->
```

```
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="{% static 'js/jquery.min.js' %}"></script>
</head>
<body>
  <nav class="navbar navbar-default navbar-fixed-top background-container">
        <ul class="nav navbar-nav">
            <!--Hamburger Menu : Mobile Only-->
            <li class="hamburger">
                <a href=".if-mobile" type="button" class="hamburger-trigger" data-class="menu-visible">
                    <img src="{% static 'assets/Icons/humburger.png' %}" alt="Menu" width="20">
                </a>
            </li>
            <!--Logo-->
            <li class="brand">
                <a class="navbar-brand" href="index.html">
                    <img src="{% static 'assets/Logo.png' %}" alt="Brand Logo">
                </a>
            </li>
            <!--Search-->
            <li class="search">
                <span href=".search" class="toggle" data-class="search-visible">
                    <img src="{% static 'assets/Icons/search.png' %}" alt="" width="16">
                </span>
                <form class="">
                    <label for="search" class="hidden">Search</label>
                    <input type="search" name="s" value="" placeholder="Search" class="search-field form-control">
                </form>
            </li>
            <!--Menu -->
            <li class="pull-right open">
```

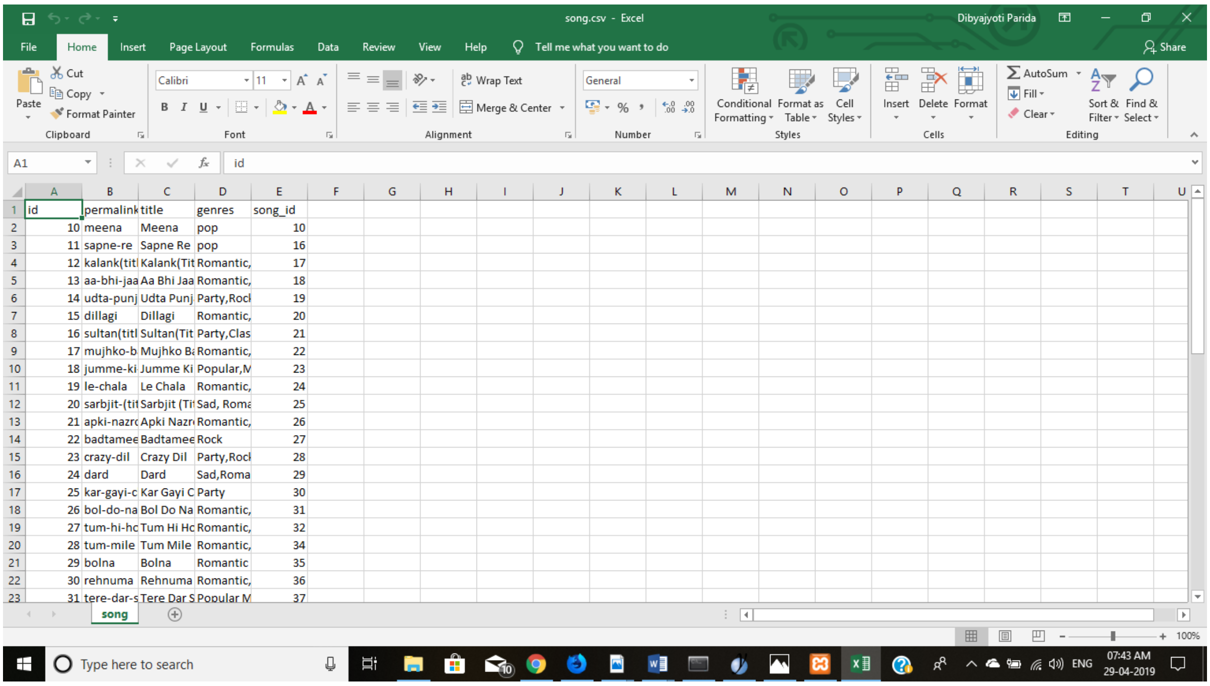
```
import json
import os
class HomePageView(TemplateView):
    template_name = "home.html"



def showAlbums(request, slug =''):
    castDetails = {}
    recommend = {}
    catlist = data.getMenu()
    result = data.getAlbumDetails(slug)
    album = result['album']
    songs = result['songs']
    title = ''
    album_type = 0
    recommendStatus = 0
    for i in album:
        title = i.title
        album_type = i.album_type
        castDetails = data.getCastDetailsOfContent(i.id, i.is_episode)
    if album_type == 0:
        arr = genre_recommendations(str(title))
        recommend = arr.to_dict('records')
        recommendStatus = 1
    return render(request, 'album.html', {"albums" : album, "recommended" :
recommend,"songs" : songs, "casts" :castDetails, "menu" : catlist, "title" : "Albums",
"detailsPage" : 1, 'recommendStatus' : recommendStatus})
def homePage(request):
    catlist = data.getMenu()
    newSingle = data.getSingle()
    newAlbum = data.getMulti()
    return render(request, 'home.html',{'menu' : catlist,'title' : 'Sound Studio', 'newSingle' :
newSingle['data'], 'newAlbum' : newAlbum['data']})
def showCategory(request, slug =''):
    catlist = data.getMenu()
    catDetails = data.getCategoryDetails(slug)
    legalCat = 0
    totalAlbums = 0
    albums = []
    if(catDetails):
        legalCat = 1
        albums = data.getAlbums(catDetails.id)
        totalAlbums =  len(list(albums))
    return render(request, 'category.html',{'albums' : albums, 'menu' : catlist,'title' : 'Sound
Studio', 'legalCat' : legalCat, 'totalAlbums' : totalAlbums})
def showCastDetails(request, slug =''):
    legalCast = 0
    totalAlbums = 0
    castname = ""
```

DATASET

song.csv



	A	B	C	D	E
1	id	permalink	title	genres	song_id
2	10	meena	Meena	pop	10
3	11	sapne-re	Sapne Re	pop	16
4	12	kalank(titl	Kalank(Tit	Romantic,	17
5	13	aa-bhi-jaa	Aa Bhi Jaa	Romantic,	18
6	14	udta-punj	Udta Punj	Party,Rock	19
7	15	dillagi	Dillagi	Romantic,	20
8	16	sultan(titl	Sultan(Tit	Party,Clas	21
9	17	mujhko-b	Mujhko Ba	Romantic,	22
10	18	jumme-ki	Jumme Ki	Popular,M	23
11	19	le-chala	Le Chala	Romantic,	24
12	20	sarbjit-(tit	Sarbjit (Tit	Sad, Roma	25
13	21	apki-nazro	Apki Nazro	Romantic,	26
14	22	badtamee	Badtamee	Rock	27
15	23	crazy-dil	Crazy Dil	Party,Rock	28
16	24	dard	Dard	Sad,Roma	29
17	25	kar-gayi-c	Kar Gayi C	Party	30
18	26	bol-do-na	Bol Do Na	Romantic,	31
19	27	tum-hi-ho	Tum Hi Ho	Romantic,	32
20	28	tum-mile	Tum Mile	Romantic,	34
21	29	bolna	Bolna	Romantic	35
22	30	rehnuma	Rehnuma	Romantic,	36
23	31	tere-dar-s	Tere Dar S	Popular M	37

BIBLIOGRAPHY

Website Reference

www.youtube.com

www.tutorialspoint.com

www.tecmint.com

www.udemy.com

www.wikipedia.com

www.medium.com

https://jquery.com/

https://getbootstrap.com/

https://www.w3schools.com/

https://www.djangoproject.com/

https://codeigniter.com/