

# CHAPTER 1

## INTRODUCTION

MelodyStream is a web application that aims to make it easy for users to access high-quality audio and video content. The platform provides a reliable and user-friendly interface for music and video streaming, content uploading, and content verification. It is built on the MERN stack, which is known for its flexibility and scalability.

The key features of MelodyStream include user authentication, content streaming, content uploading, content verification by a designated verifier, search and discovery, playlists and favorites, and user roles and permissions. All of these features work together to create a comprehensive music and video streaming platform that caters to the needs of different user types such as admin, verifier, and user.

MelodyStream's technical architecture is designed for reliability, scalability, and security. It uses cloud-based hosting and MongoDB database, which ensures that the platform can handle a large number of users and content. Additionally, the MERN stack provides the flexibility and scalability needed for a music and video streaming platform.

The implementation of MelodyStream has resulted in a user-friendly and reliable music and video streaming platform. Users can easily access high-quality content, while content providers can be sure that their content is verified and meets the platform's standards. The content verification process has been well-received, as it ensures that all uploaded content is authentic and of high quality.

MelodyStream accommodates different user types such as admin, verifier, and user. The admin has full access to the platform and can manage users, content, and verifiers. Verifiers are responsible for verifying uploaded content, while users can browse and stream content, upload their own content.

## 1.1 Objectives

The main objective of MelodyStream is to provide a reliable platform for music and video streaming that offers a high-quality user experience. Additionally, the platform aims to provide a content verification process that ensures the authenticity and quality of user-generated content, as well as user authentication and authorization to ensure security and privacy for users. To achieve this, MelodyStream has been designed to accommodate the needs of different user types, including admin, verifier, and user. The platform also aims to foster a sense of community among users and provide a platform for sharing and discovering new music and video content. To ensure scalability and flexibility, the platform has been built using the MERN stack and has implemented cloud-based hosting and a MongoDB database for reliability and security. Additionally, user feedback and usage statistics will be continually collected and analyzed to improve the platform's functionality and user experience. Ultimately, MelodyStream aims to become a leading music and video streaming platform that offers a comprehensive user experience and high-quality content.

## CHAPTER 2

### LITERATURE SURVEY

A survey of video streaming techniques by S. S. Sahoo and S. K. Lenka (2011). This paper provides a comprehensive overview of the various video streaming techniques used in different scenarios, such as live streaming and on-demand streaming. The authors discuss the challenges faced by video streaming platforms and the methods used to overcome them.

An overview of video streaming protocols" by M. Al-Tahat and A. Al-Mahasneh (2015). This paper provides a detailed review of the most commonly used video streaming protocols, such as HTTP-based protocols, RTMP, and RTP. The authors compare the features, performance, and limitations of these protocols and discuss their suitability for different use cases.

A comparative study of video streaming platforms by S. Mukherjee et al. (2018). This paper presents a comparative study of various video streaming platforms, such as YouTube, Netflix, and Amazon Prime Video. The authors analyze the features, content, user experience, and revenue models of these platforms and identify the factors that contribute to their success.

Adaptive bitrate streaming for video delivery over the internet by Y. Liu and H. Lu (2011). This paper discusses the adaptive bitrate streaming (ABS) technique, which is used to deliver video content over the internet while adapting to the varying network conditions. The authors describe the principles of ABS and evaluate its effectiveness in improving the quality of video streaming.

Video streaming over wireless networks: A survey by N. Saxena and V. Gupta (2016). This paper provides an overview of the challenges and solutions for video streaming over wireless networks. The authors discuss the various wireless technologies used for video streaming and the techniques used to mitigate the effects of packet loss, delay, and jitter.

Quality of experience in video streaming: Challenges and solutions by M. Zink et al (2014). This paper focuses on the quality of experience (QoE) in video streaming, which is a critical factor in the success of video streaming platforms. The authors discuss the challenges in measuring and improving QoE and the solutions proposed by researchers and industry

practitioners.

Security issues in video streaming over the internet by A. Kumar and R. S. Raw (2017). This paper discusses the security issues in video streaming over the internet, such as content piracy, copyright infringement, and cyber attacks. The authors describe the different security mechanisms used by video streaming platforms and evaluate their effectiveness in ensuring the security of the content and the users.

Social media-based video streaming platforms: A review by A. Aggarwal et al. (2021). This paper reviews the emerging trend of social media-based video streaming platforms, such as Facebook Live, Instagram Live, and TikTok. The authors discuss the features, content, user engagement, and monetization strategies of these platforms and the challenges they face in ensuring the quality and security of the content.

Multimedia content delivery in 5G networks: Opportunities and challenges by X. Zhou et al (2020). This paper discusses the opportunities and challenges of multimedia content delivery in 5G networks, which promise higher bandwidth, lower latency, and better user experience. The authors describe the key features of 5G networks and the multimedia delivery techniques that can benefit from them.

Blockchain-based video streaming platforms: A review by S. K. Ray and A. Ghosh (2021). This paper reviews the emerging trend of blockchain-based video streaming platforms, which aim to address the issues of content ownership, copyright protection,

A survey of cloud-based video streaming by S. S. Sahoo and S. K. Lenka (2014). This paper provides a survey of cloud-based video streaming technologies and architectures. The authors discuss the various cloud-based video streaming platforms and their features, advantages, and limitations. They also present a comparative analysis of cloud-based video streaming and traditional video streaming techniques.

Quality assessment of video streaming services: A survey by Y. Liu et al. (2016). This paper presents a survey of quality assessment techniques for video streaming services. The authors discuss the subjective and objective quality assessment methods and the challenges in evaluating the quality of video streaming services. They also review the existing quality

assessment models and metrics and their applicability to different scenarios.

Video streaming using peer-to-peer networks: A survey by J. Li et al. (2010). This paper provides a survey of video streaming using peer-to-peer (P2P) networks. The authors discuss the various P2P video streaming architectures and the challenges in achieving high-quality video streaming over P2P networks. They also review the existing P2P video streaming systems and their features, limitations, and performance.

A survey of multimedia streaming over wireless networks by C. Qian et al. (2017). This paper presents a survey of multimedia streaming over wireless networks, including audio and video streaming. The authors discuss the challenges in multimedia streaming over wireless networks, such as network congestion, packet loss, and signal interference. They also review the existing multimedia streaming techniques and protocols and their effectiveness in improving the quality of multimedia streaming over wireless networks.

A survey of adaptive video streaming techniques by M. Amiri et al. (2016). This paper provides a survey of adaptive video streaming techniques, including client-based and server-based adaptation. The authors discuss the principles of adaptive video streaming and the challenges in designing adaptive video streaming systems. They also review the existing adaptive video streaming techniques and their performance, scalability, and flexibility.

Video streaming over software-defined networks: A survey by F. J. Villanueva et al. (2018). This paper presents a survey of video streaming over software-defined networks (SDNs). The authors discuss the advantages of SDNs for video streaming, such as improved network management, QoS, and security. They also review the existing SDN-based video streaming systems and their features, challenges, and opportunities.

A survey of real-time video streaming techniques for mobile devices by S. Jiang et al. (2013). This paper provides a survey of real-time video streaming techniques for mobile devices, including smartphones and tablets. The authors discuss the challenges in real-time video streaming over mobile networks, such as limited bandwidth and battery life. They also review the existing real-time video streaming techniques and their performance, efficiency, and usability.

Video streaming for virtual reality: A survey" by F. Xu et al. (2019). This paper presents a survey of video streaming for virtual reality (VR) applications. The authors discuss the challenges in video streaming for VR, such as high bandwidth requirements and low latency. They also review the existing VR video streaming techniques and their features, limitations, and performance.

## CHAPTER 3

### METHODOLOGY

#### 3.1 Proposed System

MelodyStream is a web application developed using the MERN stack, which comprises MongoDB, Express.js, React.js, and Node.js. The proposed system is designed to provide users with a platform to stream audio and video content seamlessly. It also allows users to upload and post their own videos for others to view.

The system is designed to have three different user roles – Admin, Verifier, and User. Admins have complete control over the system, including managing user accounts, verifying videos uploaded by users, and adding new content to the platform. Verifiers are responsible for checking and approving videos uploaded by users, ensuring that they are appropriate for the platform. Users can browse and stream content on the platform and also upload their own videos for others to view.

#### 3.2 Technologies Used

The following technologies were used in developing MelodyStream:

##### 3.2.1 MongoDB

MongoDB is a NoSQL database used to store data in JSON-like documents. It was used to store data for MelodyStream, including user accounts, videos, and other content.

##### 3.2.2 Express.js

Express.js is a web application framework for Node.js that provides a range of features for building web applications. It was used to create the backend for MelodyStream, handling HTTP requests and responses and interacting with the database.

##### 3.2.3 React.js

React.js is a JavaScript library used for building user interfaces. It was used to create the front-end of MelodyStream, providing users with an intuitive and responsive interface for browsing and streaming content.

##### 3.2.4 Node.js

Node.js is a JavaScript runtime environment used to build server-side applications. It was used to run the backend of MelodyStream, handling requests from the front-end and interacting with the database.

### **3.3 Working of Modules**

The system comprises several modules that work together to provide users with a seamless experience. These modules include:

#### **3.3.1 User Authentication**

The user authentication module is responsible for ensuring that only authorized users can access the system. It requires users to provide valid login credentials before they can access the system. The authorization module ensures that users can only perform actions within their roles.

#### **3.3.2 Content Management**

The content management module is responsible for managing the content on the platform. It includes adding new content to the platform, removing inappropriate content, and updating existing content.

#### **3.3.3 Video Verification**

The video upload module allows users to upload their own videos to the platform. The verification module ensures that uploaded videos are appropriate for the platform and meet the necessary standards.

#### **3.3.4 Streaming**

The streaming module is responsible for streaming audio and video content to users. It uses a range of technologies, including HTML5 video and audio tags, to provide users with a seamless streaming experience.

In conclusion, the methodology section outlines the approach taken to develop MelodyStream, the technologies used, and the working of different modules. This section provides a detailed overview of the system's design and functionality and serves as a foundation for the rest of the case study report.

### **3.4 Data Flow Diagram**

DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. It is a graphical tool, useful for communicating with users, managers and other personnel. It is useful for analyzing existing as well as proposed system.



### 3.4.1 Level 0

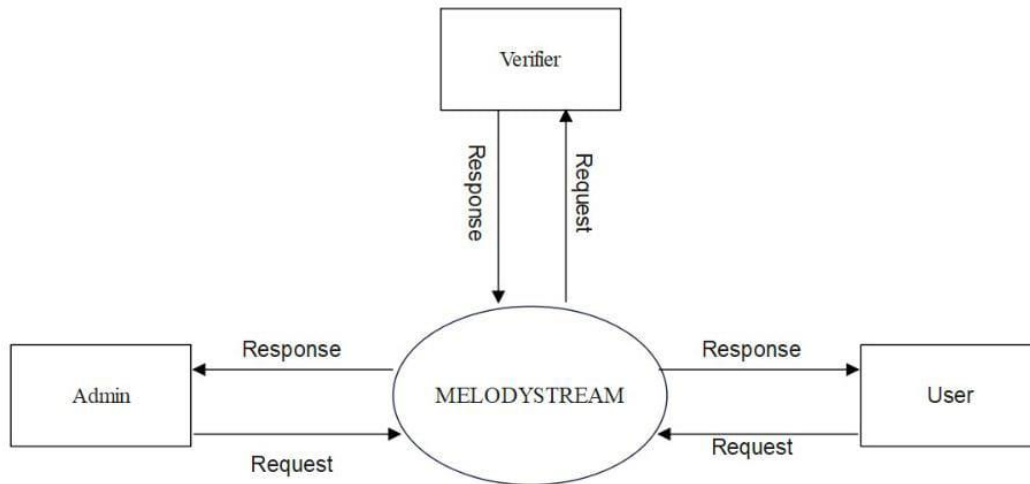


Figure 3.1: Level 0 DFD

### 3.4.2 Level 1 for Admin

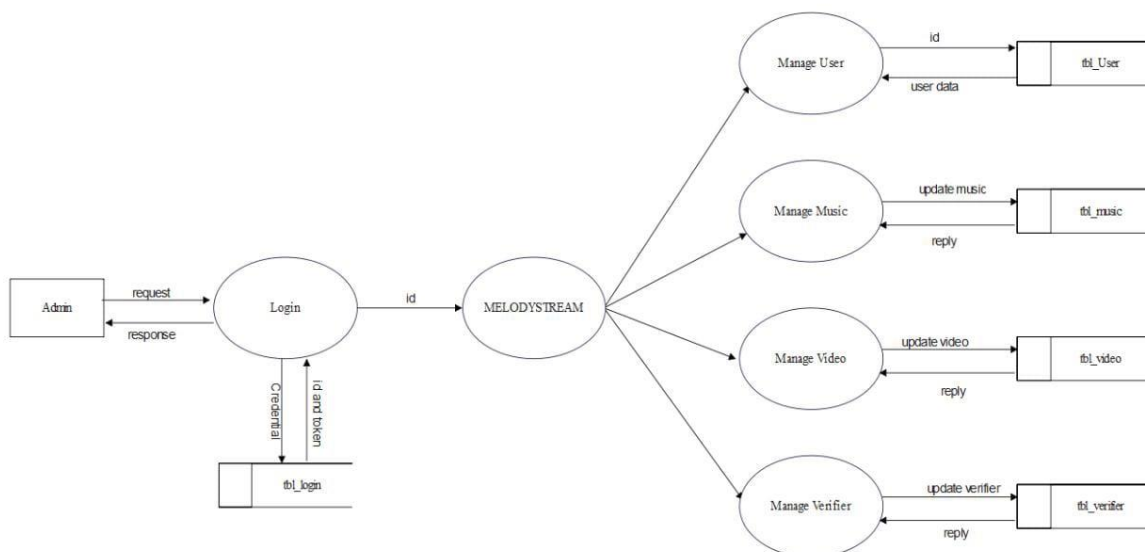


Figure 3.2: Level 1 DFD for Admin

### 3.4.3 Level 1 for User

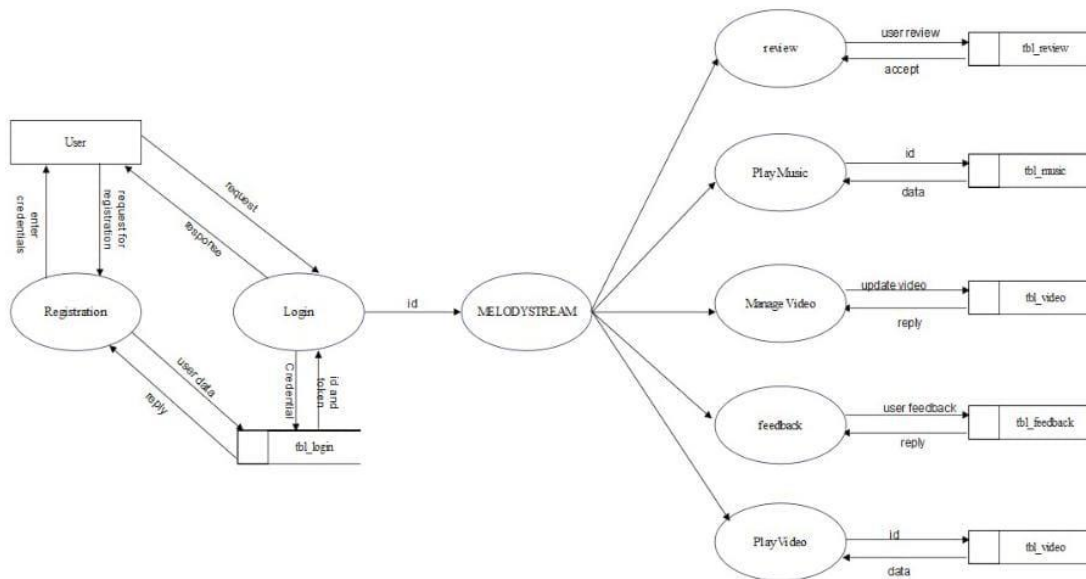


Figure 3.3: Level 1 DFD for User

### 3.4.4 Level 1 for Verifier

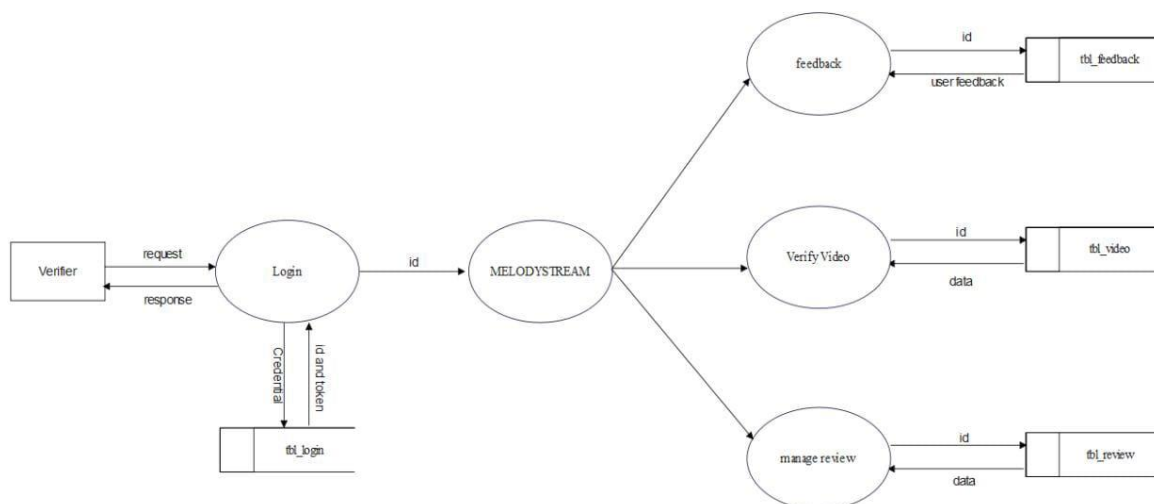


Figure 3.4: Level 1 DFD for Verifier

## **CHAPTER 4**

### **RESULTS AND DISCUSSION**

Testing is the major quality measures employed during the software development. After the coding phase, computer programs available are executed for testing purpose. Testing not only has to uncover errors introduced during coding, but also locates errors committed during previous phase. Thus the aim testing is to uncover requirements design or coding errors in the program.

- Testing is a process of executing a program with intension of finding an error.
- A good test case is on that has a highest probability of finding an as yet undiscovered error.
- A successful testing is one that covers an as yet undiscovered error

The main objective is to design tests that systematically uncover different classes of errors and to do so with minimum amount of time and effort. Testing demonstrate that software functions appear to be working according to specification, that performance requirements appears to have been met. Data collected as testing is conducted provide a good indication of software reliability and some indication of software quality as a whole. But there is one thing that testing cannot do: Testing cannot show the absence of defects it can only show that software defects as present.

#### **4.1 Test Plan**

A test plan is a systematic approach to test a system. The plan typically contains a detailed understanding of what the eventual workflow will be. Normally testing of any large system will be in two parts.

- The functional verification and validation against the requirement specification
- Performance evaluation against the indicated requirements

Testing activity is involved right from the beginning of the project. At the very first stage of testing, the goals and objectives are set. This simplifies the limits or boarders of testing process. Before testing, the tester should plan what kind of data he is giving for test. Give data inputs as functional, boundary, stress, performance, usability values etc.

Characteristics of a Good Test:

- Tests are likely to catch bugs
- No redundancy
- Not too simple or too complex

Test Cases A specific set of steps and data along with expected results of a particular test objective. A test case should only test one limited subset of a feature or functionality. Test case documents for each functionality/testing area will be written, reviewed and maintained separately in excel sheets. In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same. Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if error test cases are not yet written, it is OK for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any, is expected to trigger errors.

## **4.2 Implementation**

Implementation is the process of having the system personnel check out and put new equipment to use, train the users to use the new system and construct any file that are needed to see it. The final and important phases in the system life cycle are the implementation of the new system. System implementation refers to the steps necessary to install a new system to put into operation. The implementation has different meaning, ranging from the conversion of a basic application to complete replacement of computer system. Implementation includes all these activities that take place to convert from old system to new one. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The methods of implementation and time scale adopted are found out initially. The system is tested properly and at the same time the users are trained in the new procedure. Proper implementation is essential to provide a reliable system to meet organizational requirements. Successful implementations may not guarantee improvement in the organization involves the following things:

- Careful planning
- Investigation of the system and constraint

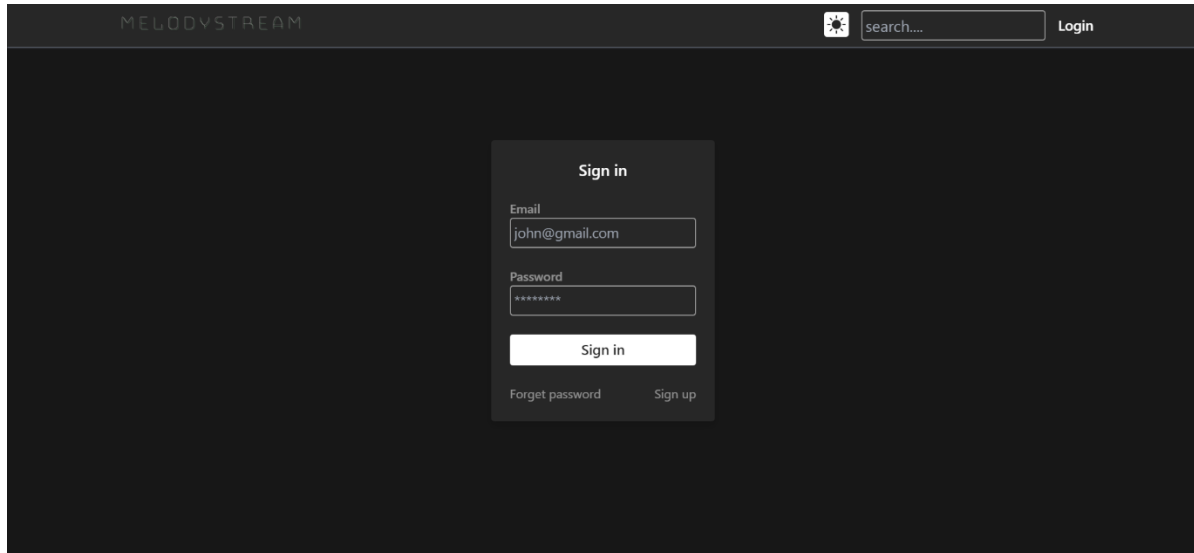
- Design the methods to achieve the changeover .
- Train the staff in the changed phase
- Evaluation of change over method Implementation methods

There are several methods for handling the implementation and consequent conversation from the old to new automated system. The most secure for this conversation is to run the old and new system in parallel. This method offers high security but the cost for maintaining the two systems in parallel is very high. Another method is direct cut over the existing system to automated system. The change may take place within a week or within a day. Implementation Phase It includes a description of all activities that must occur to implement the new system and put into operation. It consists of the following steps:

- List all files required for the implementation.
- Identify all data required to build new files during the implementation
- List all new document and procedure that go to the new system.

## 4.3 Outputs

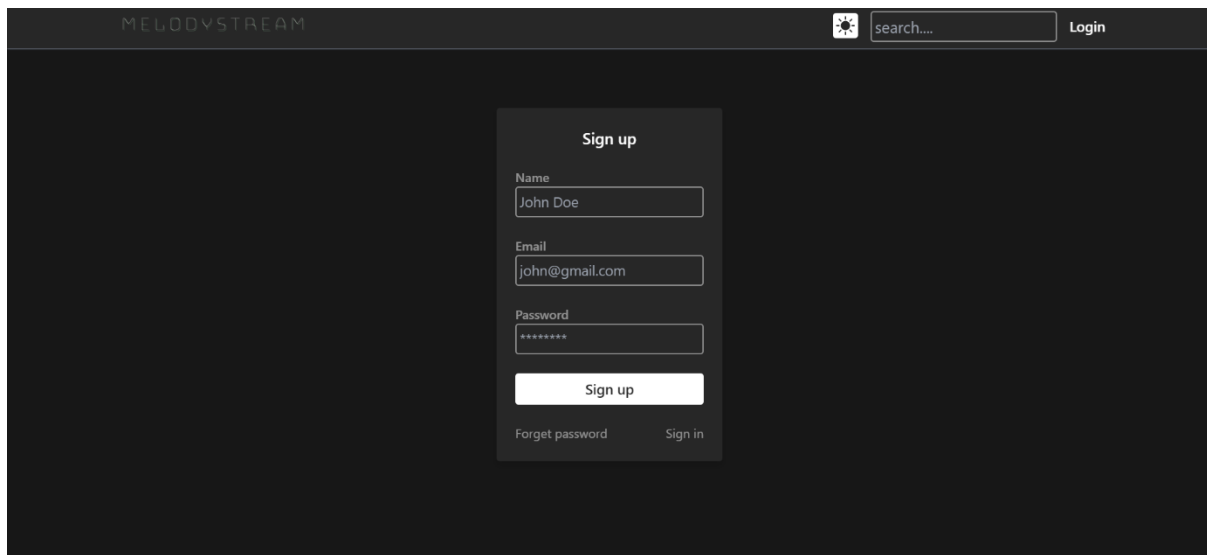
### 4.3.1 User Login



The screenshot shows the MELODYSTREAM login page. At the top, there is a dark header with the site name 'MELODYSTREAM' on the left, a search bar with a magnifying glass icon and the text 'search....' in the center, and a 'Login' button on the right. The main content area is dark gray. In the center, there is a white 'Sign in' form. The form has a title 'Sign in', an 'Email' field with the text 'john@gmail.com', a 'Password' field with six asterisks, a white 'Sign in' button, and two links at the bottom: 'Forget password' and 'Sign up'.

Figure 4.1: Login Page

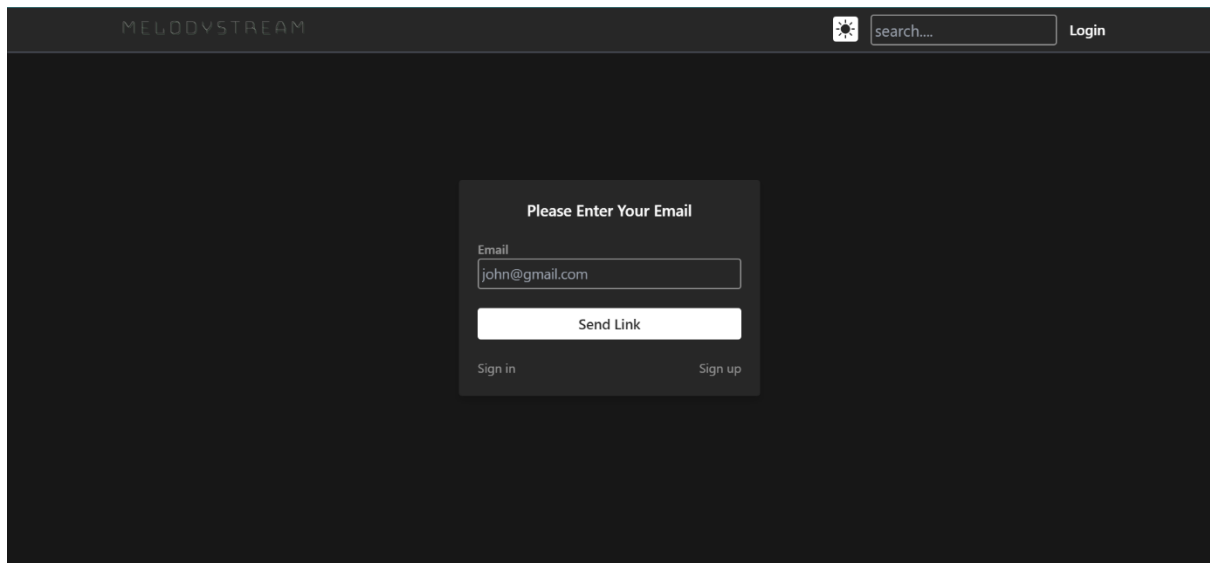
### 4.3.2 User Registration



The screenshot shows the MELODYSTREAM registration page. At the top, there is a dark header with the site name 'MELODYSTREAM' on the left, a search bar with a magnifying glass icon and the text 'search....' in the center, and a 'Login' button on the right. The main content area is dark gray. In the center, there is a white 'Sign up' form. The form has a title 'Sign up', a 'Name' field with the text 'John Doe', an 'Email' field with the text 'john@gmail.com', a 'Password' field with six asterisks, a white 'Sign up' button, and two links at the bottom: 'Forget password' and 'Sign in'.

Figure 4.2: Registration Page

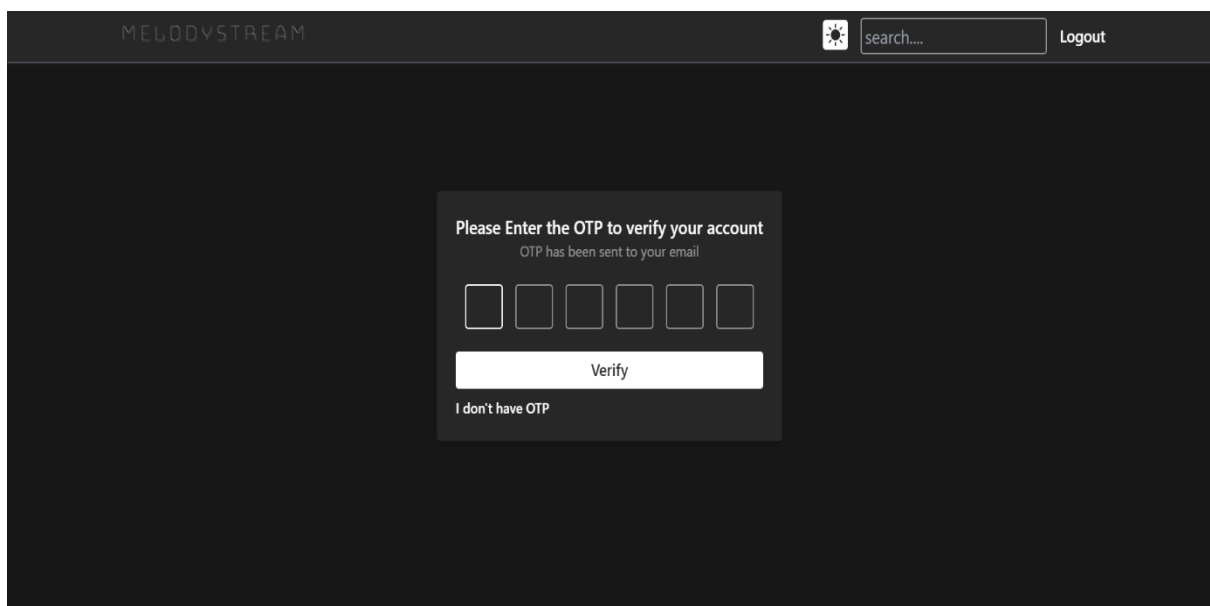
### 4.3.3 Forgot Password



The screenshot shows the 'Forgot Password' page of the MELODYSTREAM application. The page has a dark theme. At the top, there is a header with the 'MELODYSTREAM' logo on the left, a search bar with a magnifying glass icon and the text 'search....' in the center, and a 'Login' button on the right. The main content area is dark gray and contains a white-bordered box with the title 'Please Enter Your Email'. Inside this box, there is an 'Email' label, a text input field containing 'john@gmail.com', a 'Send Link' button, and two links at the bottom: 'Sign in' and 'Sign up'.

Figure 4.3: Forgot Password Page

### 4.3.4 Email Verification



The screenshot shows the 'Email Verification' page of the MELODYSTREAM application. The page has a dark theme. At the top, there is a header with the 'MELODYSTREAM' logo on the left, a search bar with a magnifying glass icon and the text 'search....' in the center, and a 'Logout' button on the right. The main content area is dark gray and contains a white-bordered box with the title 'Please Enter the OTP to verify your account'. Below the title, it says 'OTP has been sent to your email'. There are six empty square input fields for the OTP digits. Below these fields is a 'Verify' button. At the bottom of the box, there is a link that says 'I don't have OTP'.

Figure 4.4:Email Verification Page

## CHAPTER 5

### CONCLUSION

Based on the analysis of the MelodyStream case study, it can be concluded that the platform has been successful in addressing the challenges faced by users in accessing high-quality music and video content. The platform's user-friendly interface, reliable streaming capabilities, and content verification features have created a comprehensive platform that accommodates the needs of its different user types.

The use of the MERN stack for development has allowed for a high degree of flexibility and scalability, ensuring reliability and security for its users. The development approach have also been effective in identifying and resolving issues early in the development process, resulting in a robust and functional platform.

This project has the potential for further improvements and enhancements, such as personalized playlists and live streaming, which could enhance the user experience and increase engagement.

The MelodyStream project is an important and relevant contribution to the field of music and video streaming platforms. The successful implementation of this project demonstrates the power of the MERN stack for web development. The project has the potential for growth and development in the future, and it has the potential to create a unique community of music and video lovers, where they can interact and share their passion with each other.



## 5.1 Future Enhancement

- **Personalized Recommendations:** By implementing personalized recommendations based on user preferences, listening history, and behavior, the platform can offer a more personalized experience for its users.
- **Social Sharing:** By incorporating social sharing features, users can easily share their favorite music and videos with friends and followers on various social media platforms, increasing the platform's visibility and engagement.
- **Live Streaming:** Incorporating live streaming capabilities would allow users to watch and participate in live events and concerts, creating a more immersive experience.
- **Gamification:** By incorporating gamification features such as badges, achievements, and rewards, the platform can increase engagement and encourage users to explore more of the platform's features.
- **Virtual Reality:** The integration of virtual reality technology can create an immersive and interactive experience for users, allowing them to attend virtual concerts or music festivals from the comfort of their own homes.

## REFERENCES

1. Bradshaw, T. (2019). How to Build a Music Streaming Website. Medium. <https://medium.com/swlh/how-to-build-a-music-streaming-website-2996a0a6be0f>
2. D'Cruz, M. (2018). Building a music streaming service from scratch. Hackernoon. <https://hackernoon.com/building-a-music-streaming-service-from-scratch-5d1f7c9dc9c8>
3. Express. (2021). Express.js Guide. <https://expressjs.com/>
4. Hennessy, M. (2021). The Beginner's Guide to ReactJS. FreeCodeCamp. <https://www.freecodecamp.org/news/the-beginners-guide-to-reactjs/>
5. Koisuru, K. (2020). Building a scalable audio streaming platform with Node.js. Medium. <https://medium.com/@koisuru/building-a-scalable-audio-streaming-platform-with-node-js-aa461a12a977>
6. Kross, J. (2019). Building a music streaming app with React Native. Medium. <https://medium.com/react-native-training/building-a-music-streaming-app-with-react-native-4a317f22df53>
7. Li, S. (2019). Building a Music Streaming Web App with React. LogRocket. <https://blog.logrocket.com/building-a-music-streaming-web-app-with-react/>
8. MongoDB. (2021). MongoDB Documentation. <https://docs.mongodb.com/>
9. Node.js. (2021). Node.js Documentation. <https://nodejs.org/en/docs/>
10. React. (2021). React Documentation. <https://reactjs.org/docs/getting-started.html>
11. Richardson, L. (2020). RESTful Web Services. O'Reilly Media.
12. Sisodia, N. (2020). Building a Music Streaming Website. GeeksforGeeks. <https://www.geeksforgeeks.org/building-a-music-streaming-website/>
13. Sovit, R. (2019). How to Build a Music Streaming App like Spotify. Apiumhub. <https://apiumhub.com/tech-blog-barcelona/how-to-build-a-music-streaming-app-like-spotify/>
14. Stack Overflow. (2021). Stack Overflow. <https://stackoverflow.com/>
15. Udemy. (2021). Udemy. <https://www.udemy.com/>
16. W3Schools. (2021). W3Schools. <https://www.w3schools.com/>
17. YouTube. (2021). YouTube. <https://www.youtube.com/>
18. Zameer, S. (2019). Building a music streaming app using React Native. Medium. <https://medium.com/@saadzameer/building-a-music-streaming-app-using-react-native-d6c5f5ae5d6e>

19. Zerín, N. (2020). How to Build a Music Streaming App like Spotify: Features, Cost, and Tech Stack. Cleveroad. <https://www.cleveroad.com/blog/how-to-build-a-music-streaming-app-like-spotify>
20. Zhang, H. (2021). Best Practices for RESTful APIs. Auth0. <https://auth0.com/blog/restful-api-best-practices-rest-api-security/>