Eötvös Loránd University Faculty of Informatics

Department of Computer Algebra

Melodious: Web application for learning/teaching musical instruments

Supervisor:

Harrison-Juhász Zsófia

Lecturer

Author:

Akimzhanov Zheenbek

Computer Science BSc

Budapest, 2023

The recent pandemic resulted in education/work transitioning into online format through special web pages or applications which brought a realisation to many people that knowledge and experience can be gained not only by actually attending in person. All the online platforms and services finally got acknowledged by a bigger number of people.

Personally, I started learning guitar by watching YouTube tutorials and by doing some readings in different pages. I noticed that most of the musical instrument tutors offer their

online courses through their own external web pages. So the problem is when it comes to learning musical instruments remotely, there are not great varieties of courses bundled in one place. So I thought: "Why is there not any platform specifically dedicated for different music tutors to upload courses for people to buy if interested?". So I decided just to do that.

The basic features of the web application are going to be:

- Being able to register the user
- Being able to both publish and view others' content
- The format of the lessons may include video, text, note, sheets/tabs, etc.
- The musical instruments are categorised into different sections and can be easily navigated through
- Tutors and learners can communicate
- The progress is saved throughout the course

The application can be extended way further with the features uniquely related to instruments

For the implementation ReactJS, JavaScript and Java will be used

Contents

Chapter 1

Introduction

1.0 Embracing the Digital Transformation in Education

In the transformative wake of the 21st century, the global education landscape has witnessed remarkable changes, especially catalyzed by the Covid-19 pandemic. This sudden, yet enduring shift from traditional classroom settings to virtual learning environments has not only necessitated change but also highlighted the benefits of online learning, ushering in a new era of education.

1.1 Striking a Chord in the World of Music Education

Among the various fields embracing this transformation, music education has carved a unique niche in the digital world. Nevertheless, while online resources for music learning are abundant, a comprehensive platform that seamlessly integrates learning, teaching, and community interaction remains conspicuously absent. This gap is what "Melodious," our Online Music Education Platform, aspires to fill.

1.2 Melodious: An Innovative Symphony of Learning and Teaching

Melodious serves as an innovative ICT solution, extending beyond the conventional framework of knowledge transfer. It embodies an open marketplace for music learning, fostering an interactive and collaborative environment. This platform encourages users not only to partake in a wide array of music courses but also to contribute to the shared pool of knowledge by creating and publishing their own courses.

1.3 Hitting the Right Notes: The Advantages of Melodious

The advantages of Melodious are multifaceted. For learners, it allows self-paced learning in a comfortable environment, encompassing a broad spectrum of music genres, instruments, and styles. For educators, it opens up an opportunity to reach a global audience, share their expertise, and make a significant contribution to the world of music education.

1.4 Harmonizing User Experience: The Unique Features of Melodious

Melodious stands out with its user-centric approach. It embeds numerous interactive features like course and user reviews, keyword-based search functions, and personalized sections such as "My learnings" and "My courses". These features create an optimized user experience, while private messaging and media sharing capabilities encourage communication and collaboration among users.

1.5 Creating a Melodious Community: Fostering Engagement Through User Interaction

One unique feature that further enriches the community aspect of Melodious is the ability for users to follow each other. This fosters a sense of community, allowing users to track their favorite educators or learners, thereby enhancing mutual growth and engagement within the platform.

1.6 From Notes to Symphony: The Evolution of Melodious

The comprehensive nature of Melodious positions it as more than just an educational platform. It evolves into a global community where music enthusiasts can learn, teach, interact, and grow together. This thesis offers a detailed account of Melodious's development and operation, underscoring the significant value it brings to online music education.

Chapter 2

User Documentation

This user documentation aims to provide detailed instructions and explanations, helping users effectively navigate and utilize the various features and functionalities of Melodious.

2.0 Short Description of the Solved Problem

Melodious addresses the need for a comprehensive and interactive online music education platform. By providing a marketplace for music learning, Melodious offers a solution that seamlessly integrates learning, teaching, and community interaction.

2.1 Summary of Used Methods and Tools

Melodious was developed using modern technologies and tools. The site's logic was initially sketched using Figma for conceptualization. For the backend, Java and the Spring framework were used, including the configuration of security features. MySQL was employed as the database to store the platform's data. ReactJS was utilized for the client-side, ensuring a modern and responsive user interface. Additionally, Postman was used for testing the implemented APIs. The application is hosted on an Ubuntu-based server.

2.2 Getting Started

To access Melodious, visit the platform's domain: https://melodiousacademy.com. Upon landing on the homepage, you will find a promotional page that highlights the platform's features, testimonials, and available courses. To get started, click on the "Get Started" button, which will direct you to the sign-up/sign-in page. Create a username and password to register your account. After successfully logging in, you will be redirected to the dashboard page.

2.3 Functionality and Features

2.3.1 Navigation Bar

The navigation bar at the top of the page provides easy access to various sections:

Dashboard: Provides an overview of the user's activities and statistics.

My Courses: Allows users to manage their published and draft courses.

My Learnings: Provides tabs for taken courses, finished courses, saved courses, and archived courses.

Explore: Enables users to browse and discover published courses by category and recommendations.

Chats: Facilitates user-to-user communication through private messaging.

Storage: Stores files saved within course content or from media shared in chats.

2.3.2 My Courses

Under the "My Courses" section, users will find two tabs:

Published Courses: Displays courses published by the user, including course preview cards with relevant information.

Draft Courses: Shows draft courses that the user is currently working on. Users can edit or publish these courses.

2.3.3 My Learnings

The "My Learnings" section includes tabs for different categories of courses:

Taken Courses: Displays courses that the user has enrolled in.

Finished Courses: Shows courses that the user has completed.

Saved Courses: Contains courses that the user has saved for future access.

Archived Courses: Includes courses that the user has archived.

2.3.4 Explore

The "Explore" page allows users to browse published courses categorized by music types. It also includes a recommended courses section based on user preferences and interactions, as well as courses from followed users.

2.3.5 Chat

The "Chats" feature enables users to engage in private messaging, including the ability to send text and media to other users. Users can filter chats by username or content.

2.3.6 Storage

The "Storage" section stores files saved within course content or from media shared in chats. Various file types, including videos, images, PDF files, and notes, are supported.

2.4 Workflows and Use Cases

2.4.1 Course Creation and Editing

Users can create new courses or edit existing draft courses, specifying course details, curriculum, and content.

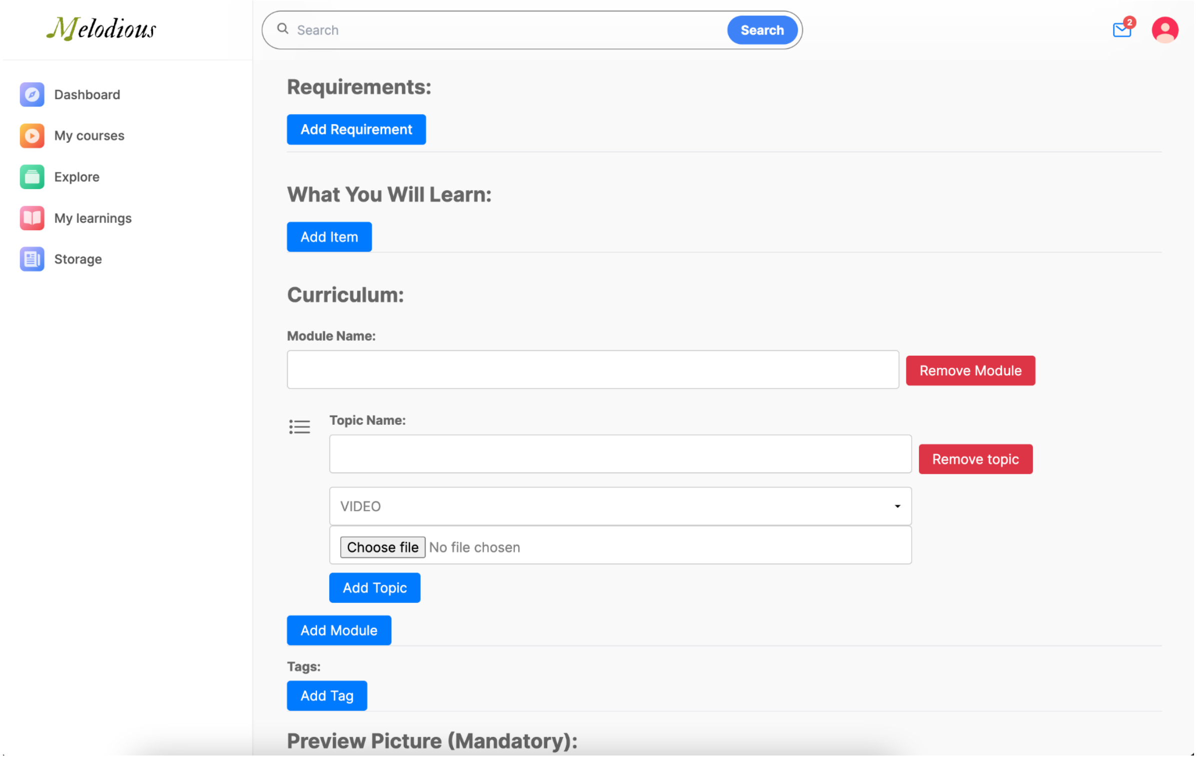
Course creation involves setting the course's name, short and long descriptions, price, requirements, and what users will learn.

The curriculum consists of modules, and each module can contain multiple topics.

Topics can include files, quizzes, or text-based content, each of which can be edited.

Users can set tags for the course, aiding search and categorization.

Preview images and promo videos can be added to enhance course presentation.



Requirements:

What You Will Learn:

Curriculum:

Module Name:

Topic Name:

VIDEO

No file chosen

Tags:

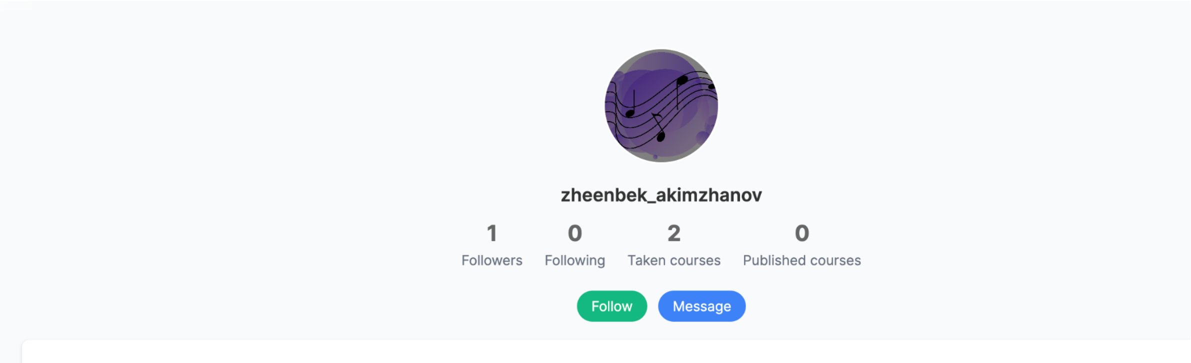
Preview Picture (Mandatory):

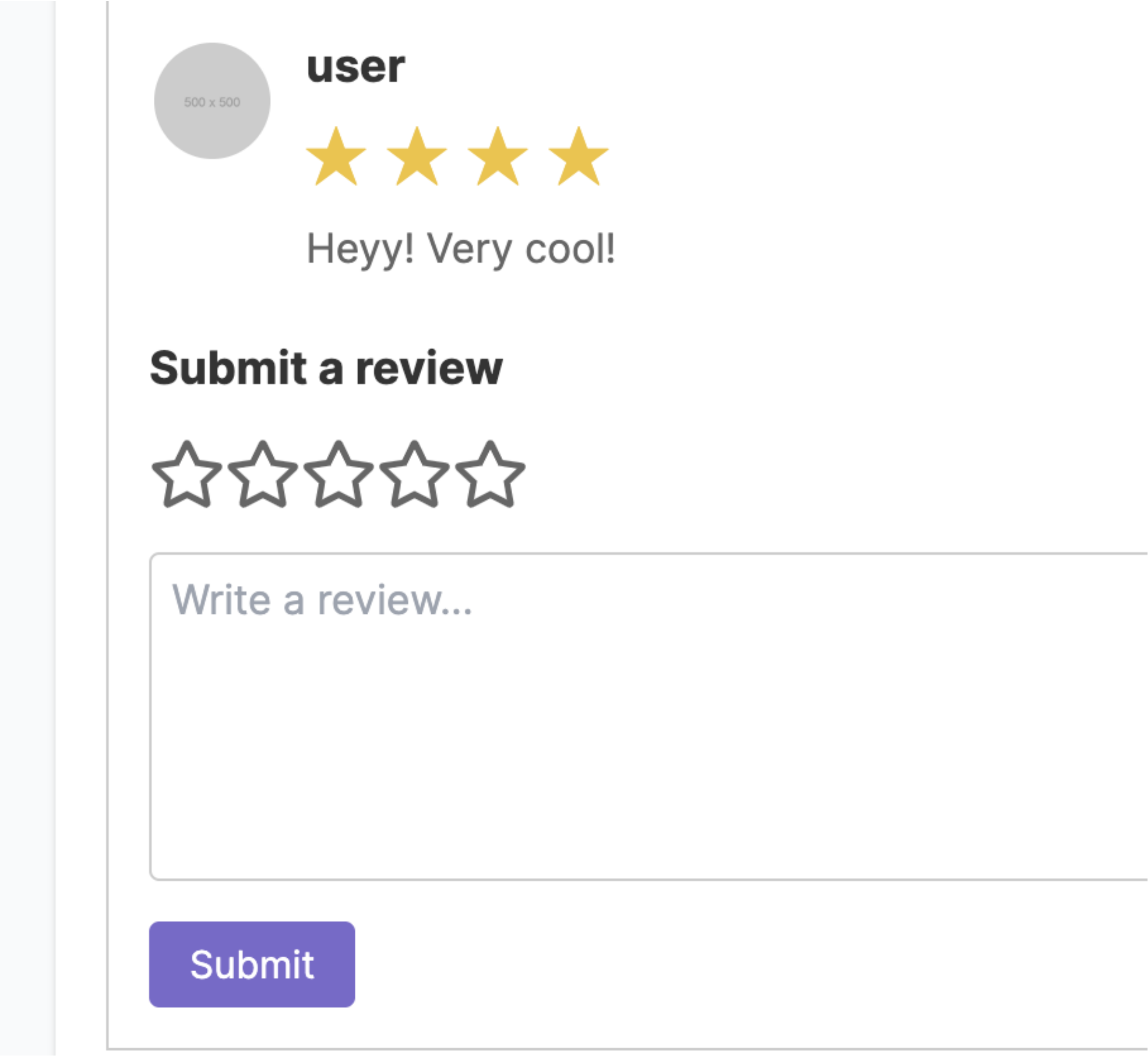
2.4.2 User Profile and Reviews

Users can view and edit their profiles, including uploading a profile picture, adding their first and last names, and providing an "About Me" section.

Profiles display user information, such as followers, followings, courses taken, and courses published.

Users can receive reviews/ratings from others and leave reviews for courses or other users.





Heyy! Very cool!

2.4.3 Course Enrollment and Content

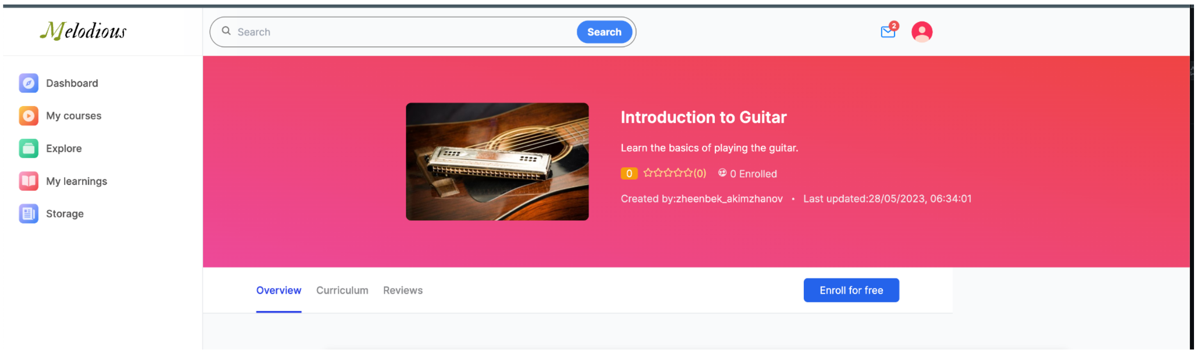
Users can enroll in courses by clicking on the "Enroll" or "Buy" button on the course description page.

Enrolled courses appear in the "Taken Courses" tab.

Course content pages provide a navigation panel for easy traversal and display the content itself, which can include videos, images, text, quizzes, etc.

Progress through the course is facilitated by arrow buttons, allowing users to move forward or backward.

Upon completing all quizzes, a course is marked as finished and is moved to the "Finished Courses" tab.

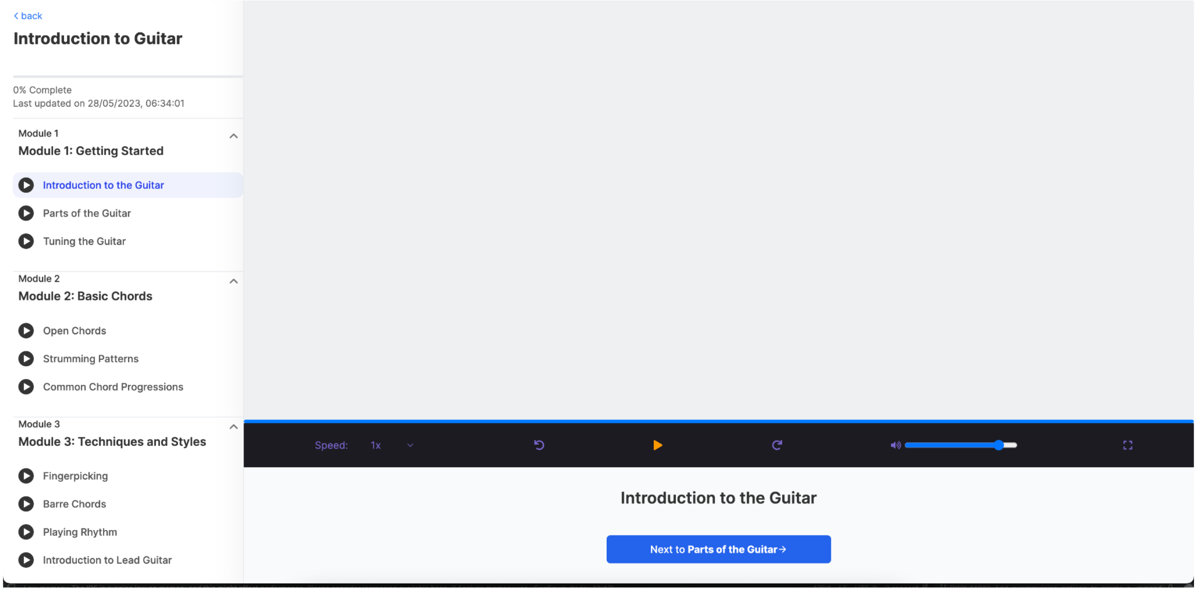


Introduction to Guitar

Learn the basics of playing the guitar.

0 ☆☆☆☆☆(0) 0 Enrolled

Created by:zheenbek_akimzhanov • Last updated:28/05/2023, 06:34:01



Introduction to Guitar

0% Complete
Last updated on 28/05/2023, 06:34:01

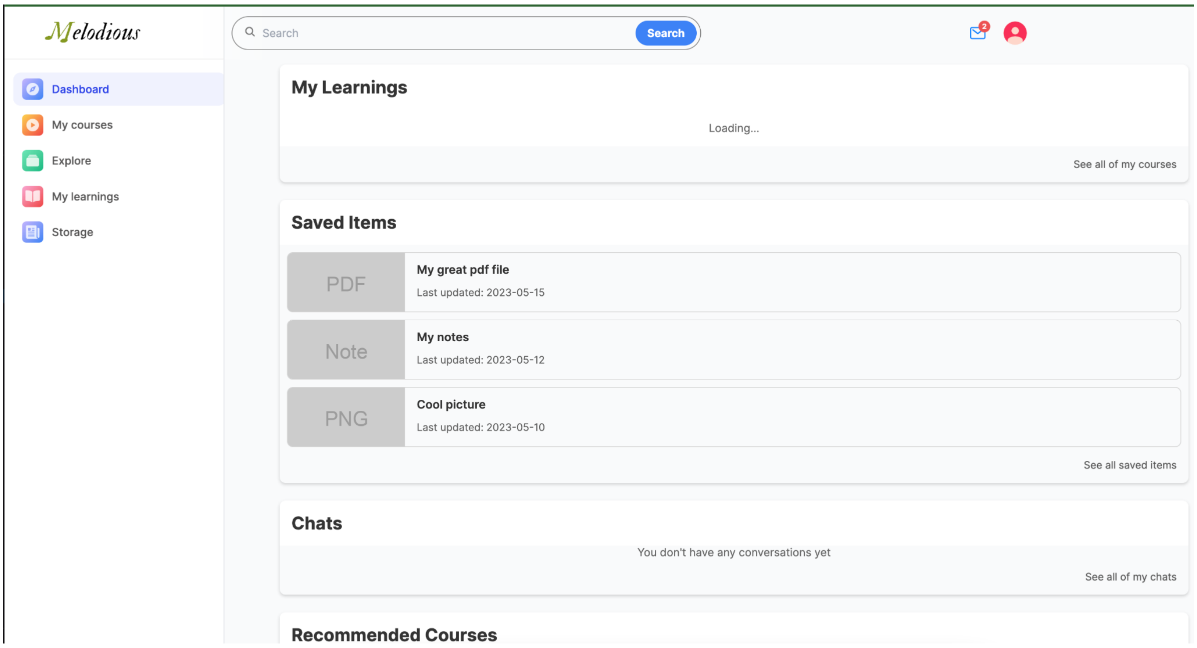
Module 1
Module 1: Getting Started
- Introduction to the Guitar
- Parts of the Guitar
- Tuning the Guitar

Module 2
Module 2: Basic Chords
- Open Chords
- Strumming Patterns
- Common Chord Progressions

Module 3
Module 3: Techniques and Styles
- Fingerpicking
- Barre Chords
- Playing Rhythm
- Introduction to Lead Guitar

Introduction to the Guitar

2.6 Dashboard



My Learnings

Loading...

See all of my courses

Saved Items

PDF	My great pdf file Last updated: 2023-05-15
Note	My notes Last updated: 2023-05-12
PNG	Cool picture Last updated: 2023-05-10

See all saved items

Chats

You don't have any conversations yet

See all of my chats

Recommended Courses

The dashboard provides an overview of the user's activities, including quick access to essential features:

"My Learnings" block: Displays up to three courses from the "Taken Courses" category. Clicking on the block will navigate to the "Taken Courses" tab to see all taken courses.

"Creating a Course" block: Shows up to three courses from the "Draft Courses" category. Clicking on the block provides links to view all draft courses or create a new course.

"Saved Items" block: Includes up to three items from the "Storage" section. Clicking on the block allows users to view all saved items.

"Chats" block: Displays up to three recent conversations, including message previews, chat names, and profile pictures. Clicking on the block directs users to the "Chats" section to see all conversations.

Chapter 3

Developer documentation

To install and run the application with your own machine:

```
git clone https://github.com/johnnyCake1/musicEducationPlatform
```

Run the backend:

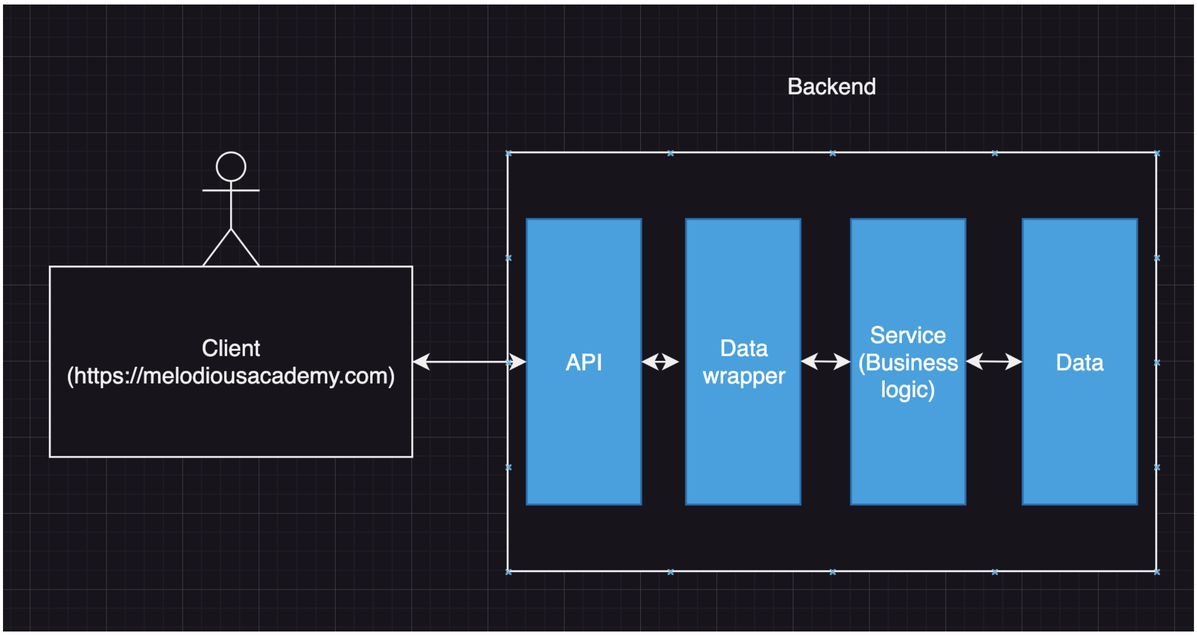
```
mvn clean install && mvn spring-boot:run
```

Run the client:

```
npm install && npm start
```

3.0 System Overview

Melodious is a robust online music education platform that leverages a standard controller <-> service <-> data layers logic to ensure efficient and scalable operations. The platform follows a client-server architecture, with a ReactJS-based frontend client communicating with a Java-based backend server.



The backend architecture adheres to the principles of separation of concerns, with controllers responsible for defining concrete APIs for clients to send and receive data. These APIs are implemented using the controller layer, which handles incoming requests and routes them to the appropriate service layer. The service layer contains the business logic of the application, handling data processing, validation, and interactions with the data layer.

Data management within Melodious is facilitated by the data layer, which utilizes the MySQL database management system for efficient storage and retrieval of user profiles, course information, reviews, and messaging. The data layer ensures secure and reliable data management, supporting the seamless functioning of the platform.

To guarantee the security of user data and interactions, Melodious incorporates Spring Security, a robust authentication and authorization framework. Each user is assigned a separate JSON Web Token (JWT) access token upon successful authentication. This access token is required to make requests to the server, and it is included in the headers of each request. This approach ensures that only authorized users can access and manipulate data within the platform.

Both the client and backend of Melodious are deployed on an Ubuntu server, which exposes the client application at the domain https://melodiousacademy.com. This deployment ensures that the platform is accessible to users over the internet, providing a seamless and user-friendly experience.

By employing a standard controller <-> service <-> data layers logic, utilizing Spring Security for authentication and authorization, and deploying the client and backend to an Ubuntu server, Melodious offers a secure, scalable, and efficient online music education platform.

3.1 Design and Architecture

Melodious follows a well-defined and modular design and architecture, ensuring scalability, maintainability, and extensibility. The system is designed to provide a seamless user experience and efficient handling of various functionalities.

Modular Architecture:

Melodious adopts a modular architecture, with distinct components responsible for specific functionalities. This modular approach allows for independent development, testing, and maintenance of different parts of the system. Key components include:

Controller Layer: Handles incoming requests from clients and defines concrete APIs for data exchange. It routes requests to the appropriate service layer for processing.

Service Layer: Contains the business logic of the application. It implements the core functionalities, such as course management, user interactions, authentication, and authorization. The service layer interacts with the data layer for data retrieval and manipulation.

Data Layer: Utilizes the MySQL database management system to store and retrieve data. It provides efficient and reliable data management, ensuring the persistence of user profiles, course information, reviews, and other essential data.

Backend Technologies:

The backend of Melodious is built using Java and the Spring framework. Java provides a robust and scalable programming language, while the Spring framework offers a wide range of tools and features for building enterprise-grade applications. It facilitates dependency injection, MVC architecture, and integrates seamlessly with other frameworks and libraries. Spring Security is utilized to ensure secure authentication and authorization processes.

Frontend Technology:

The frontend of Melodious is developed using ReactJS which enables the creation of interactive, responsive, and dynamic user experiences. React components encapsulate specific functionality and can be easily composed to build complex user interfaces. This approach promotes code reuse, modularity, and a consistent user experience throughout the application. ReactJS efficiently

Styling and CSS:

Melodious adopts a structured approach to styling by utilizing CSS and CSS-in-JS solutions, such as styled-components or CSS modules. The styling is kept separate from the components, ensuring a clear separation of concerns. This separation allows for easier management and modification of styles without impacting the underlying component logic. It also promotes code readability and enables efficient collaboration between designers and developers.

3.2 Development Workflows

Melodious adopts a set of development workflows that promote collaboration, version control, and efficient development practices. These workflows enable developers to work effectively and ensure smooth integration of new features and bug fixes into the application. Key components of the development workflows include Git version control and the use of IntelliJ IDEA Ultimate as the integrated development environment (IDE).

Git Version Control:

Melodious utilizes Git, a distributed version control system, to manage the source code and facilitate collaboration among developers. Git allows for efficient tracking of changes, branching, merging, and resolving conflicts. Developers can work on separate branches, create pull requests, and review code changes before merging them into the main codebase. This workflow ensures a controlled and organized approach to feature development, bug fixing, and code collaboration.

IntelliJ IDEA Ultimate:

Melodious leverages IntelliJ IDEA Ultimate as the primary integrated development environment (IDE) for Java development. IntelliJ IDEA Ultimate offers a wide range of features that enhance the development process. It provides comprehensive Java development tools, including code completion, debugging, refactoring, and testing capabilities. Additionally, IntelliJ IDEA Ultimate offers built-in database management features, enabling seamless interaction with the MySQL database. The IDE also includes robust version control management, allowing developers to commit, push, and pull code changes directly from the IDE. Moreover, IntelliJ IDEA Ultimate supports remote server development, facilitating the deployment and testing of the application on the Ubuntu server.

Development Environment Setup:

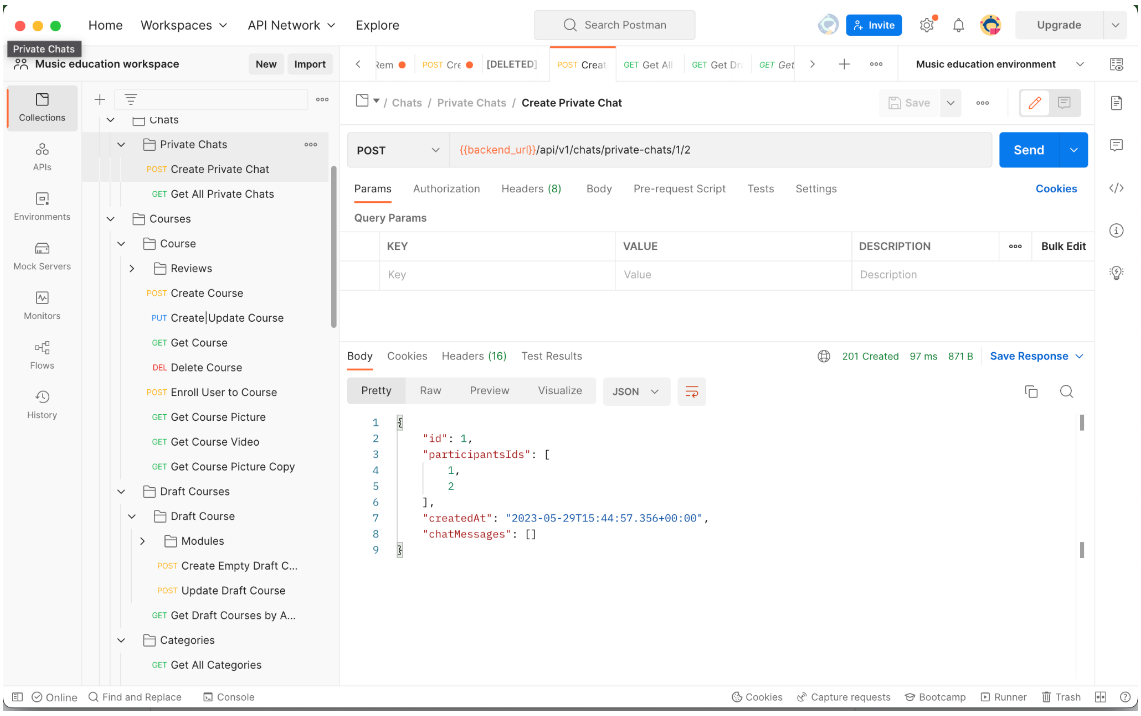
Developers working on Melodious set up their development environment by installing IntelliJ IDEA Ultimate and configuring it for the project. They clone the Git repository and set up the necessary dependencies and libraries required for the development and testing of the application. IntelliJ IDEA Ultimate provides seamless integration with Git, allowing developers to manage branches, view changes, and collaborate on code within the IDE.

3.3 Testing and Quality Assurance

Melodious recognizes the importance of testing and quality assurance to deliver a reliable and stable application. While the project may not have extensive automated tests or end-to-end testing in place, it incorporates manual testing procedures and leverages the testing capabilities provided by the Postman environment.

Testing with Postman:

Melodious leverages the Postman environment for API testing and validation. The platform's APIs are thoroughly tested using Postman's robust features. Developers utilize the Postman environment to send API requests, provide different input parameters, and validate the responses. Test cases are designed to cover a wide range of scenarios, including successful requests, error handling, and edge cases. The Postman environment allows for detailed examination of API responses, status codes, and response payloads to ensure correctness and compliance with expected results.



```
POST {{backend_url}}/api/v1/chats/private-chats/1/2
```

KEY	VALUE	DESCRIPTION
Key	Value	Description

201 Created 97 ms 871 B

```
{
    "id": 1,
    "participantsIds": [
        1,
        2
    ],
    "createdAt": "2023-05-29T15:44:57.356+00:00",
    "chatMessages": []
}
```

Manual Testing:

Melodious employs manual testing techniques to validate the functionality and user experience of the application. During manual testing, developers and quality assurance personnel systematically execute test cases, simulate user interactions, and verify the expected behavior of various features and functionalities. Manual testing includes both positive and negative test scenarios, edge cases, and user workflows to ensure comprehensive coverage.

3.4 Extensibility and Customization:

Melodious is designed to be highly extensible and customizable, allowing for future enhancements, integration of new features, and adaptation to specific requirements. The platform provides various avenues for extension and customization, empowering developers and administrators to tailor the application according to their needs. Here are some ways in which Melodious can be extended and customized:

Custom Course Modules and Content Types:

Melodious supports the creation of custom course modules and content types. Developers can expand the platform's capabilities by introducing new module types or content types, such as interactive quizzes, exercises, additional multimedia formats, or external content integrations. This flexibility enables the platform to accommodate diverse teaching methods and cater to different learning styles.

Integration with External Services:

Melodious can be extended by integrating with external services and APIs. For example, developers can integrate payment gateways to enable course purchasing and monetization. Integration with popular music streaming services or notation software can enhance the learning experience by incorporating real-time music playback or interactive exercises. By leveraging APIs and third-party services, Melodious can seamlessly integrate additional functionalities into the platform.

Customization of User Interface (UI) and User Experience (UX):

Melodious offers customization options for the user interface to align with specific branding guidelines or user preferences. Developers and administrators can modify the color schemes, typography, logos, and other UI elements to create a unique visual identity. Additionally, customization of the user experience allows for tailoring the navigation flow, user interactions, and user feedback mechanisms to optimize the platform's usability and align with specific user requirements.

User Role and Permission Customization:

Melodious provides the flexibility to customize user roles and permissions. Administrators can define new user roles or modify existing ones to suit specific organizational or educational hierarchies. This customization allows for granular control over user access, privileges, and responsibilities within the platform. Different user roles can have different permissions, such as creating courses, managing course content, or moderating user-generated content.

Localization and Internationalization:

> Melodious can be extended to support multiple languages and localization options. By integrating localization frameworks or translation services, the platform can adapt its content and user interface to different regions and languages. This customization feature enables the platform to cater to a global user base and facilitate localized learning experiences.

Plug-in Architecture and APIs:

> Melodious can incorporate a plug-in architecture that allows developers to create and integrate custom functionality through APIs. By defining well-documented APIs and providing developer resources, Melodious can encourage the development of third-party plug-ins and extensions that enhance the platform's capabilities. This extensibility empowers the community to contribute and expand the functionality of Melodious.

By enabling custom course modules, integrating with external services, customizing the user interface and user experience, customizing user roles and permissions, supporting localization and internationalization, and embracing a plug-in architecture, Melodious offers extensibility and customization options that cater to evolving needs and requirements. These customization avenues ensure the platform can adapt and grow, providing a personalized and tailored music education experience for its users.

Conclusion:

Throughout the development of Melodious, I have achieved significant milestones in addressing the problem statement and contributing to the field of music education and online learning. This project has provided an innovative ICT solution that enables users to engage in music learning, teaching, and collaboration within a seamless and interactive platform. Reflecting on the challenges I faced and the lessons I learned, as well as the impact of

Melodious, it becomes evident that the platform has made valuable contributions in the realm of music education.

By embracing the digital transformation in education, Melodious has successfully connected music learners and educators, creating a virtual marketplace for music learning and fostering a sense of community. The platform offers a diverse range of courses and empowers users to share their expertise and engage in a collaborative learning environment.

Throughout the development process, I encountered challenges such as designing an intuitive user interface, ensuring data security and privacy, and incorporating features that enhance the learning experience. Through diligent effort and the utilization of a comprehensive technology stack, I successfully addressed these challenges, resulting in a platform that provides a smooth and engaging user experience.

The impact of Melodious extends beyond its technical achievements. It has empowered music enthusiasts and educators worldwide, providing access to quality music education. The user-centric approach, rich functionality, and interactive features of the platform have contributed to the growth of a vibrant online music learning community. Melodious has enabled learners to explore a wide range of music genres, instruments, and styles, while educators have been able to reach a global audience and share their expertise.

Melodious serves as an example of the value that online learning platforms can offer in the post-Covid era, where traditional classroom settings are no longer the sole means of education. By providing a convenient and flexible learning experience, Melodious responds to the evolving needs and preferences of music learners, enabling them to pursue their passion for music regardless of geographical or time constraints. The extensibility and customization options of the platform further enhance its value, allowing for future expansion and adaptation to changing user requirements.

In conclusion, my work on Melodious has achieved significant milestones in addressing the practical problem of accessing quality music learning resources online. The impact of the platform is evident in the positive experiences of its users and the growth of its community. Melodious serves as a testament to the potential of technology in the field of education and provides inspiration for the future of music learning and online education as a whole.

Bibliography

React Documentation: The official documentation for ReactJS, which provides comprehensive guides, tutorials, and API references for developing web applications using React. [Link: https://reactjs.org/docs/]

Spring Documentation: The official documentation for the Spring Framework, which includes guides, tutorials, and reference documentation for developing Java applications using Spring. [Link: https://spring.io/docs]

MySQL Documentation: The official documentation for MySQL, which provides detailed information on database management, SQL queries, and database administration. [Link: https://dev.mysql.com/doc/]

Figma Documentation: Figma is a popular design tool. The official documentation provides tutorials, guides, and resources for creating interactive designs and wireframes. [Link: https://www.figma.com/resources/docs/]

Postman Documentation: Postman is an API development and testing tool. The documentation offers detailed information on how to create and test APIs using Postman. [Link: https://learning.postman.com/docs/]

Intellij IDEA Documentation: Intellij IDEA is an integrated development environment for Java. The documentation provides information on using the IDE's features, debugging, and version control integration. [Link: https://www.jetbrains.com/idea/documentation/]

Git Documentation: The official documentation for Git, which provides guides and reference material for version control and collaborative development. [Link: https://git-scm.com/doc]

Modern JavaScript Explained for Dinosaurs: A beginner-friendly guide to modern JavaScript development, covering ES6 features and concepts. [Link: https://medium.com/the-node-js-collection/modern-javascript-explained-for-dinosaurs-f695e9747b70]

Clean Code: A Handbook of Agile Software Craftsmanship by Robert C. Martin: A book that emphasizes the importance of writing clean, maintainable, and high-quality code. It provides practical advice and principles for software development. [Link: https://www.oreilly.com/library/view/clean-code/9780136083238/]

Designing Data-Intensive Applications by Martin Kleppmann: A comprehensive book that explores the principles, patterns, and best practices for designing and building data-intensive applications. [Link: https://dataintensive.net/]