N Music

Documentation

Group Members:

John Galarza Calderon, Tobin Antony, Rajat Rajan

Table of Contents

Introduction	3
Project Details	3
Database	3
Backend	4
Frontend	10
Sprint	10
User Stories	10
Screenshots	13
GitHub Link	21

Introduction

The purpose of this capstone project is to combine all topics learned to create a functional full stack web application. It utilizes various technologies for each main component. Frontend will utilize Bootstrap framework with Thymeleaf and JavaScript. Backend uses Spring framework and MySQL for database.

Project Details

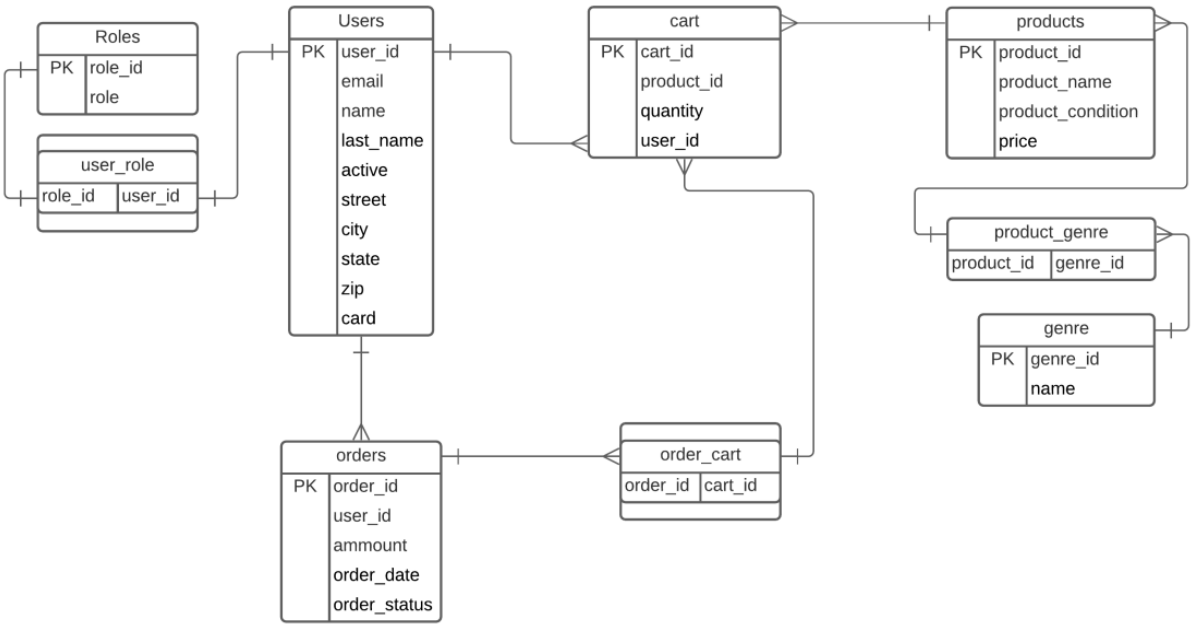
N Music is a web application that lets users search for and purchase music related products. These products include songs, albums, instruments, cd's, and records. The users will be able to navigate our catalog for desired products. To add products to the cart, the user must have an account and must be logged in; if they do not have an account they can register by following the link in the login page. Once the user is finished with shopping, they can complete the transaction in the cart page. The application also has administrator privileges for users with an admin role. Administrators can add, edit, or delete categories, genres, and products. Any changes will be stored in the database and the website will be updated immediately.

Database

MySQL Database was used for this project. N Music has the following table structure:

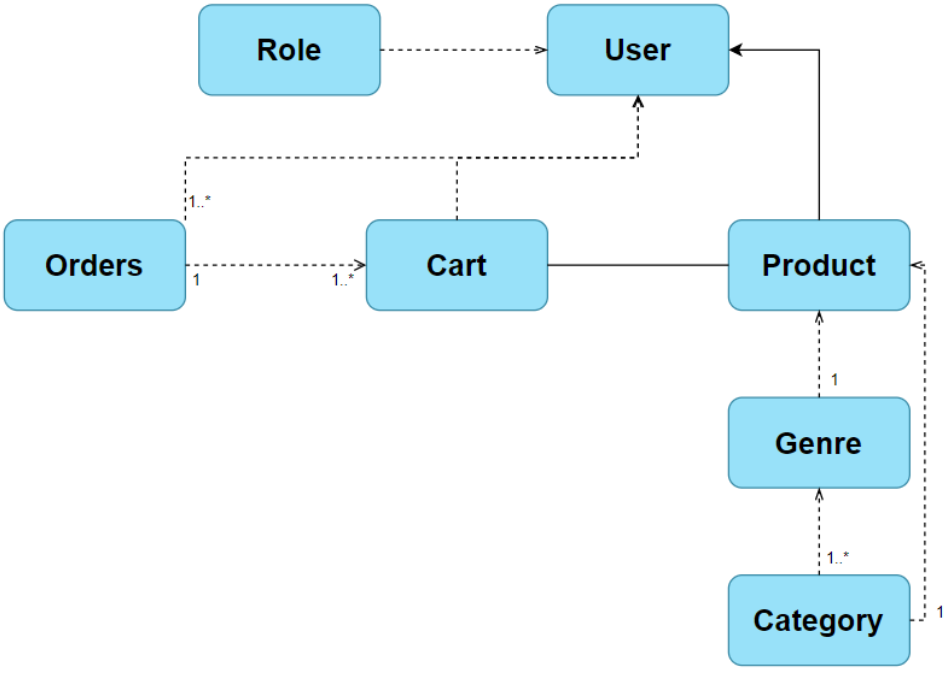
- User Table: To store the details of the user using the website.
- User Role: To store the user role. Mainly used to distinguish between admin and regular user.
- Products: To store the details of the products. It stores the details of the music like name, price, image etc.
- Cart: This table stores the details of the product added by the user to the cart. It stores data like quantity etc.
- Orders: This table stores details of purchased products.
- Orders_Cart: Since the order has a one-to-many relationship with cart items. This table is used for mapping the order with items in the cart.
- Category: This table is used to store the various categories of the product.
- Genre: This table is used to store the genre of the products.
- Genre_Category: Since genre has various categories. This table maps the relation between them.

The following is the entity relationship diagram.

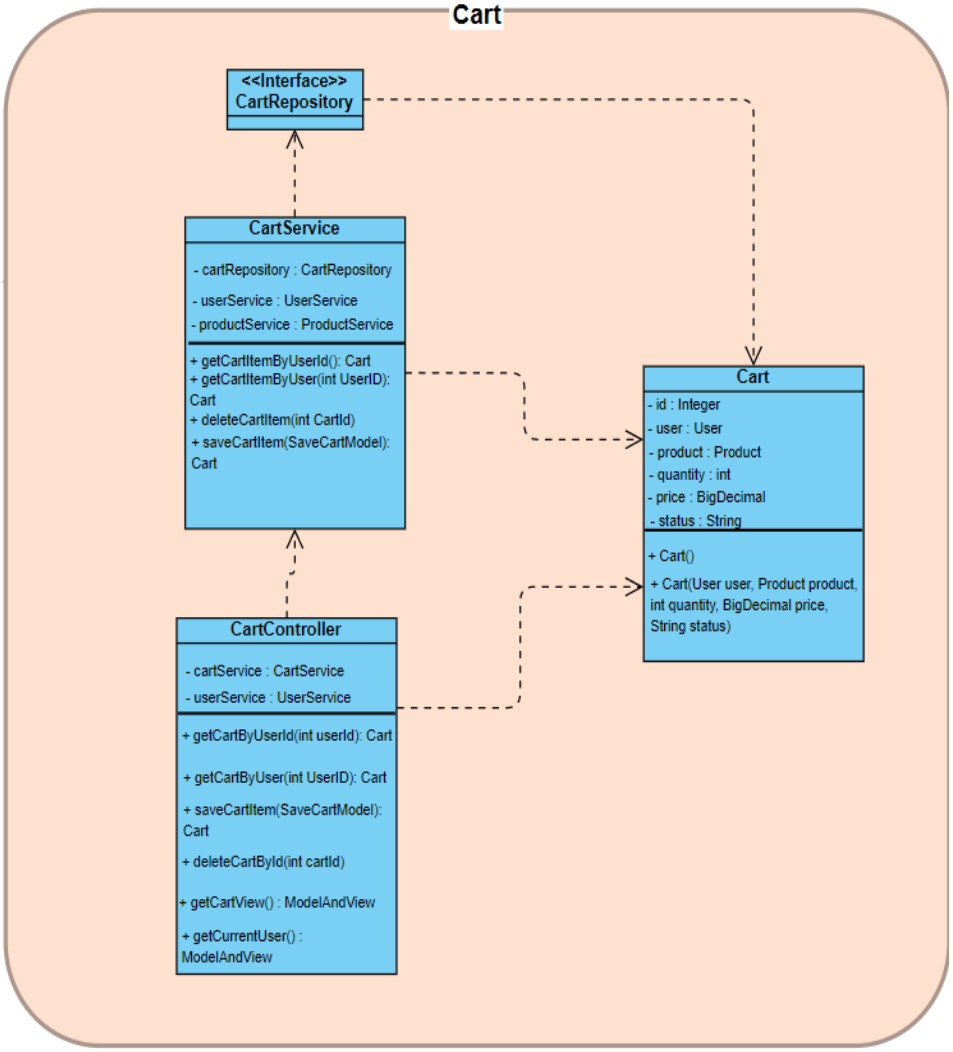


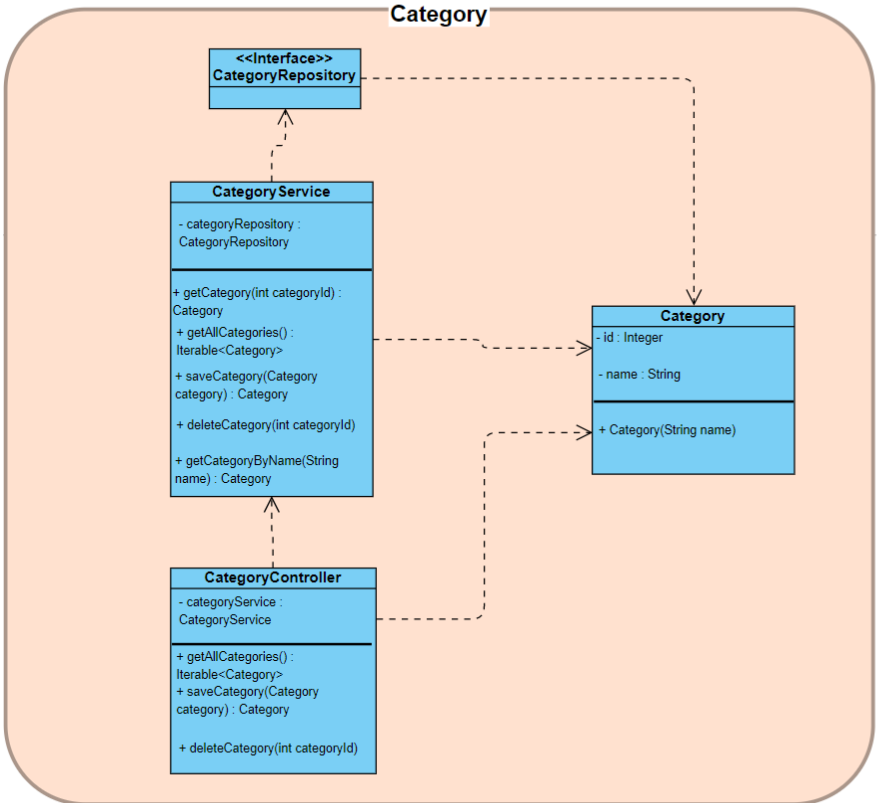
Backend

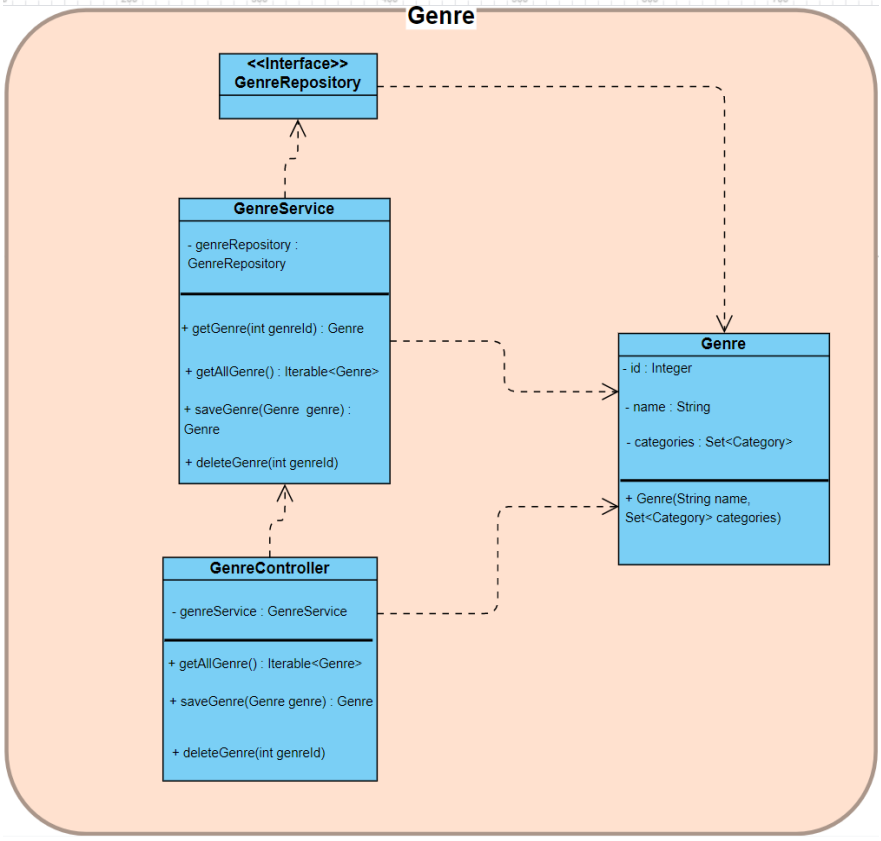
Spring Framework was used for the structuring of our project. By doing so, this project consists of entities, controllers, repositories, and services. The following diagram shows all entities used in this project and their relationships.

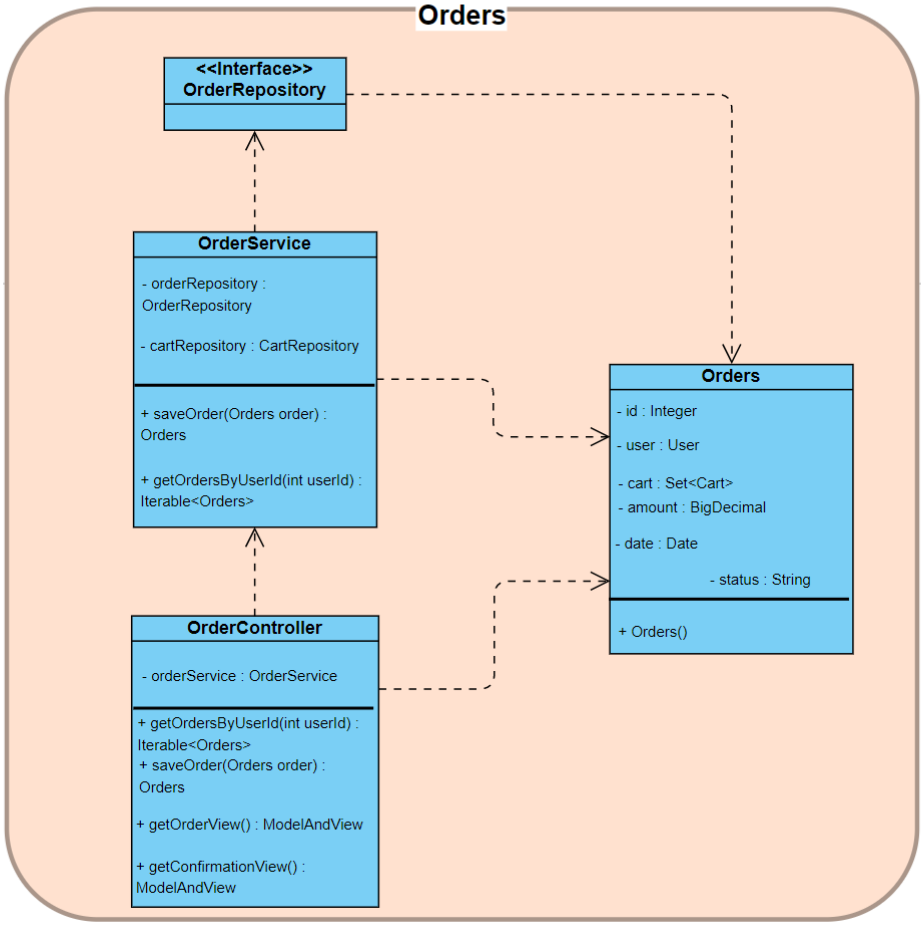


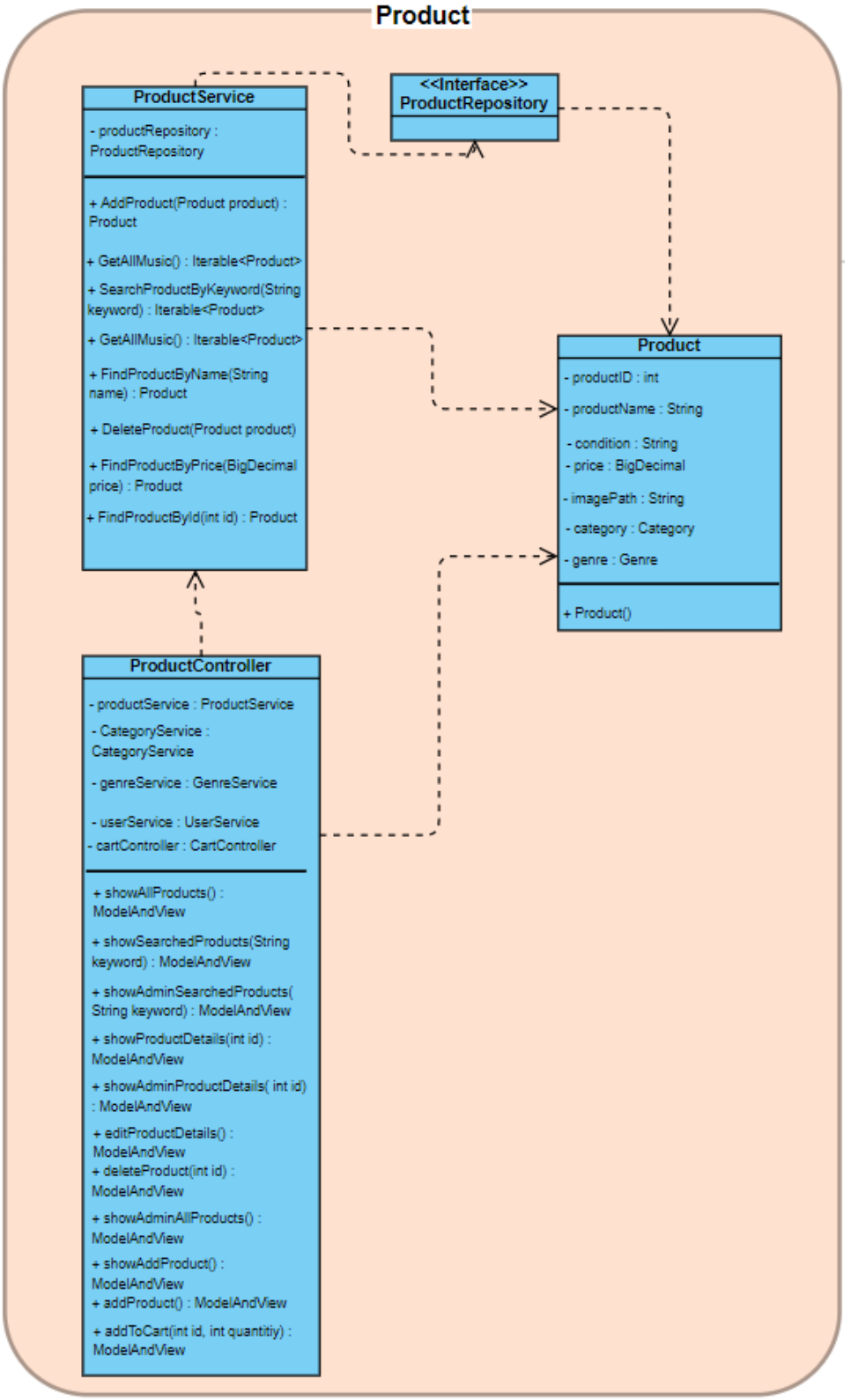
Each entity has its own repository, service, and controller. The repository extends either CrudRepository or JpaRepository, giving them the ability to perform crud functions for database manipulation. The service contains methods that use these functions to perform some action like add or delete. Finally the controller can call the functions from the service and handle frontend actions as needed. The following are the class diagrams for each entity.

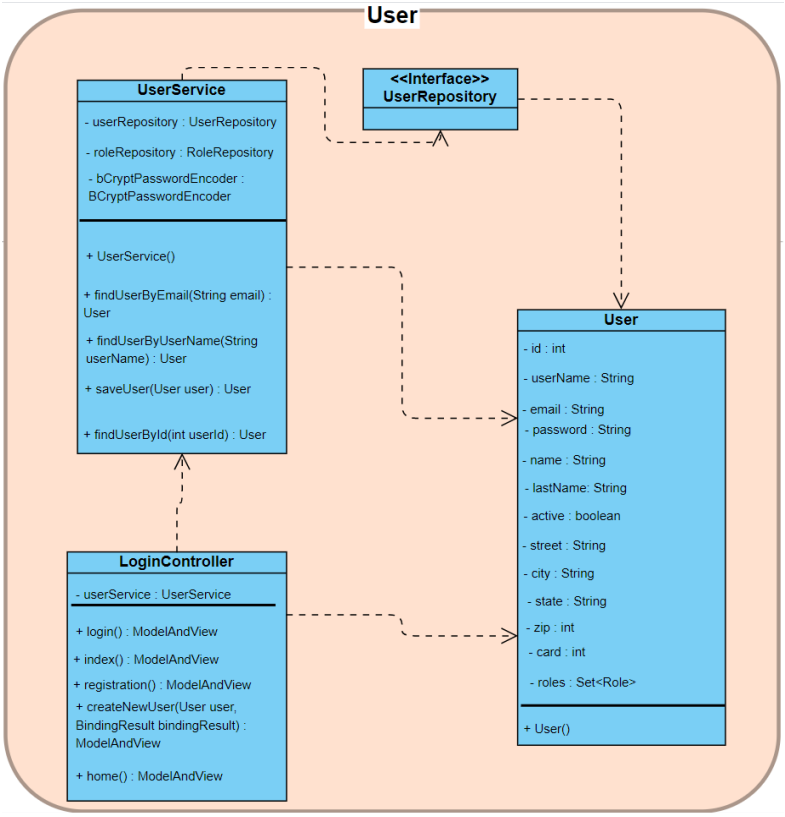


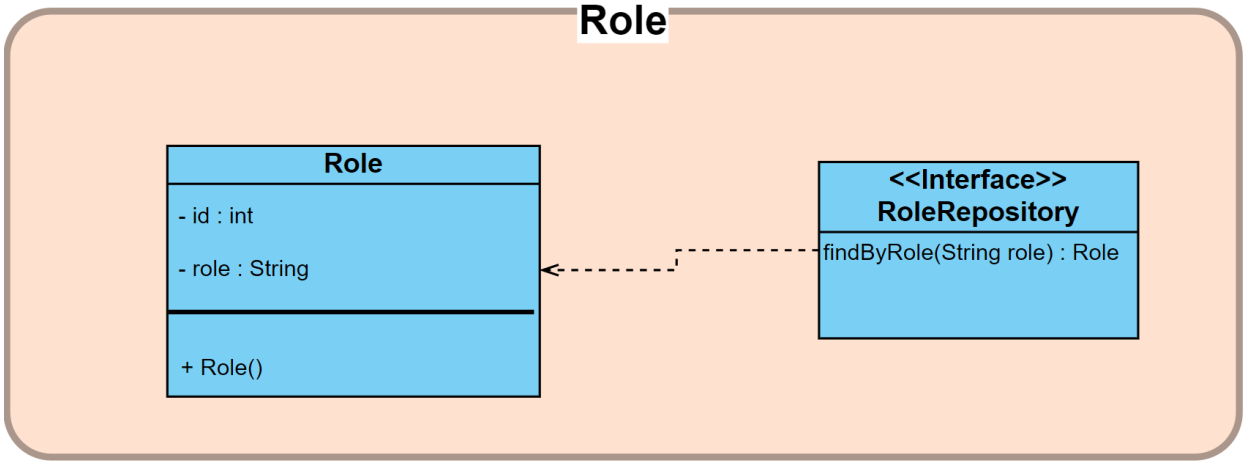

Category
<<Interface>>
CategoryRepository
CategoryService
- categoryRepository : CategoryRepository
+ getCategory(int categoryId) : Category
+ getAllCategories() : Iterable<Category>
+ saveCategory(Category category) : Category
+ deleteCategory(int categoryId)
+ getCategoryByName(String name) : Category
Category
- id : Integer
- name : String
+ Category(String name)
CategoryController
- categoryService : CategoryService
+ getAllCategories() : Iterable<Category>
+ saveCategory(Category category) : Category
+ deleteCategory(int categoryId)


Genre
<<Interface>>
GenreRepository
GenreService
- genreRepository : GenreRepository
+ getGenre(int genreId) : Genre
+ getAllGenre() : Iterable<Genre>
+ saveGenre(Genre genre) : Genre
+ deleteGenre(int genreId)
Genre
- id : Integer
- name : String
- categories : Set<Category>
+ Genre(String name, Set<Category> categories)
GenreController
- genreService : GenreService
+ getAllGenre() : Iterable<Genre>
+ saveGenre(Genre genre) : Genre
+ deleteGenre(int genreId)


Orders
<<Interface>>
OrderRepository
OrderService
- orderRepository : OrderRepository
- cartRepository : CartRepository
+ saveOrder(Orders order) : Orders
+ getOrdersByUserId(int userId) : Iterable<Orders>
Orders
- id : Integer
- user : User
- cart : Set<Cart>
- amount : BigDecimal
- date : Date
- status : String
+ Orders()
OrderController
- orderService : OrderService
+ getOrdersByUserId(int userId) : Iterable<Orders>
+ saveOrder(Orders order) : Orders
+ getOrderView() : ModelAndView
+ getConfirmationView() : ModelAndView


Product
ProductService
- productRepository : ProductRepository
+ AddProduct(Product product) : Product
+ GetAllMusic() : Iterable<Product>
+ SearchProductByKeyword(String keyword) : Iterable<Product>
+ GetAllMusic() : Iterable<Product>
+ FindProductByName(String name) : Product
+ DeleteProduct(Product product)
+ FindProductByPrice(BigDecimal price) : Product
+ FindProductById(int id) : Product
<<Interface>>
ProductRepository
Product
- productID : int
- productName : String
- condition : String
- price : BigDecimal
- imagePath : String
- category : Category
- genre : Genre
+ Product()
ProductController
- productService : ProductService
- CategoryService : CategoryService
- genreService : GenreService
- userService : UserService
- cartController : CartController
+ showAllProducts() : ModelAndView
+ showSearchedProducts(String keyword) : ModelAndView
+ showAdminSearchedProducts(String keyword) : ModelAndView
+ showProductDetails(int id) : ModelAndView
+ showAdminProductDetails(int id) : ModelAndView
+ editProductDetails() : ModelAndView
+ deleteProduct(int id) : ModelAndView
+ showAdminAllProducts() : ModelAndView
+ showAddProduct() : ModelAndView
+ addProduct() : ModelAndView
+ addToCart(int id, int quantitiy) : ModelAndView


User
UserService
- userRepository : UserRepository
- roleRepository : RoleRepository
- bCryptPasswordEncoder : BCryptPasswordEncoder
+ UserService()
+ findUserByEmail(String email) : User
+ findUserByUserName(String userName) : User
+ saveUser(User user) : User
+ findUserById(int userId) : User
<<Interface>>
UserRepository
User
- id : int
- userName : String
- email : String
- password : String
- name : String
- lastName: String
- active : boolean
- street : String
- city : String
- state : String
- zip : int
- card : int
- roles : Set<Role>
+ User()
LoginController
- userService : UserService
+ login() : ModelAndView
+ index() : ModelAndView
+ registration() : ModelAndView
+ createNewUser(User user, BindingResult bindingResult) : ModelAndView
+ home() : ModelAndView


Role
Role
- id : int
- role : String
+ Role()
<<Interface>>
RoleRepository
findByRole(String role) : Role

N Music also utilizes the Spring Security dependency to handle user authentication. Spring Security lets us assign which pages are accessible by everyone and which ones are restricted by user roles.

Frontend

The front end of the application was done using HTML, CSS, Bootstrap, Thymeleaf, Javascript, and Jquery.

The overall structure was created using HTML and HTML elements.

CSS and Bootstrap were used to give the website a visual appeal. Both css style elements and Bootstrap classes were used.

Thymeleaf: thymleaf was used in screen to implement communication with the backend framework.

Javascript and Jquery: Javascript was used in some screens to manipulate the html page dynamically, for validation and to communicate through the backend using rest API's. It was also used to create dynamic elements.

Sprint

The application was developed using agile methodology and was done in sprint. After the sprint planning meeting we identified the following as the various user stories.

User Stories

1. As a developer, I want the application to use Spring Security, so that user authentication is handled securely.
2. As a developer, I want the application to have a login/registration page, so that users can sign into or register their account.
3. As an administrator, I want an admin page, so that I can see and navigate between the admin options.
4. As a user, I want to see a products page, so that I can view all products.
5. As a developer, I want a products details page, so that users can view the details of a product and add it to their cart.
6. As an administrator, I want a genre and categories page so that I can add, edit, or delete genres and categories.

7. As a user, I want a cart page so that I can view what is in my cart and begin the transaction.
8. As a developer, I want an order page, so that the user can enter mailing and billing information and complete the transaction.

The above user stories were divided among ourselves.

Sprint Backlog

User Story	Assignee	Status
Implement Spring Security	John	BackLog
Coding and Testing of the User Login Page	John	BackLog
Coding and Testing of Admin Page.	John	BackLog
Coding and Testing of the Product Screen	Rajat	BackLog
Coding and Testing of the Product Details screen	Rajat	BackLog
Coding and Testing of the Genre and Category Screen	Tobin	BackLog
Coding and Testing of the Cart Screen	Tobin	BackLog

Coding and Testing of the Order Screen	Tobin	BackLog

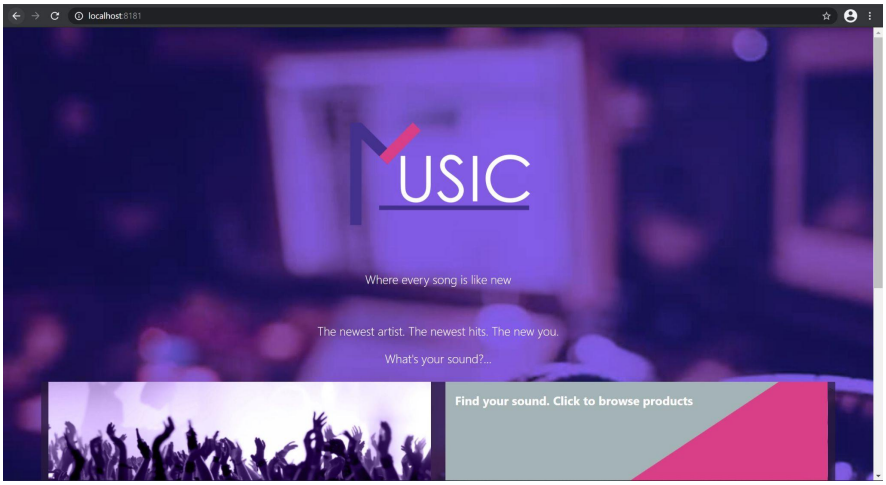
We had meetings daily and discussed the various concerns and roadblocks. And after a week of intense agile development. The status of the tasks were as follows:

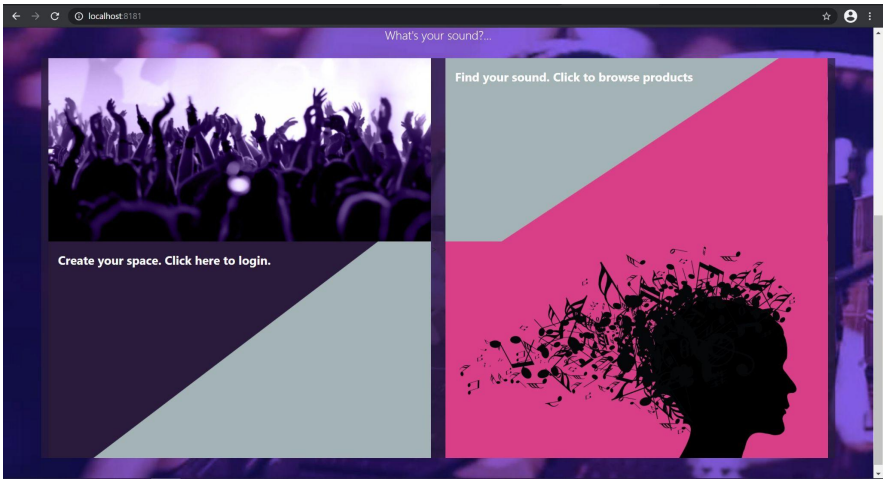
User Story	Assignee	Status
Implement Spring Security	John	Completed
Coding and Testing of the User Login Page	John	Completed
Coding and Testing of Admin Page.	John	Completed
Coding and Testing of the Product Screen	Rajat	Completed
Coding and Testing of the Product Details screen	Rajat	Completed
Coding and Testing of the Genre and Category Screen	Tobin	Completed

Coding and Testing of the Cart Screen	Tobin	Completed
Coding and Testing of the Order Screen	Tobin	Completed

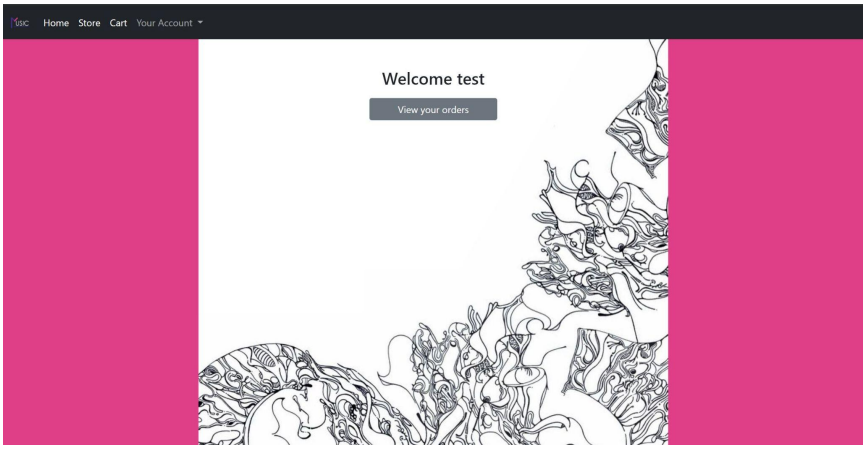
Screenshots

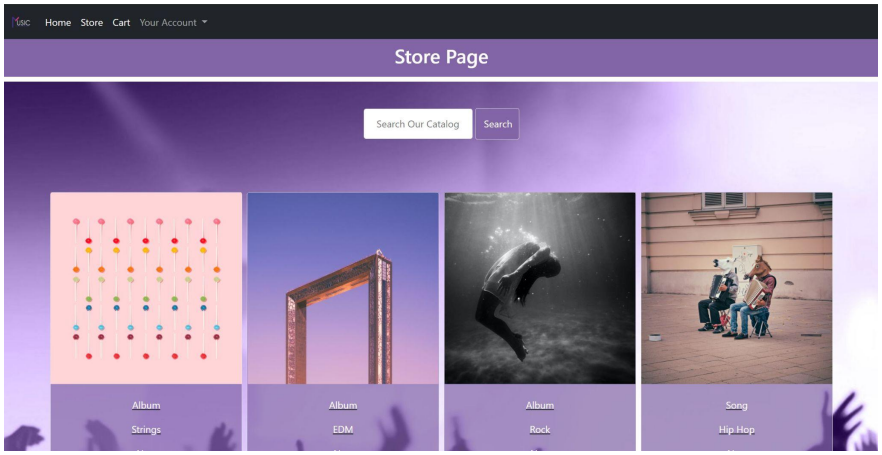
Landing Page

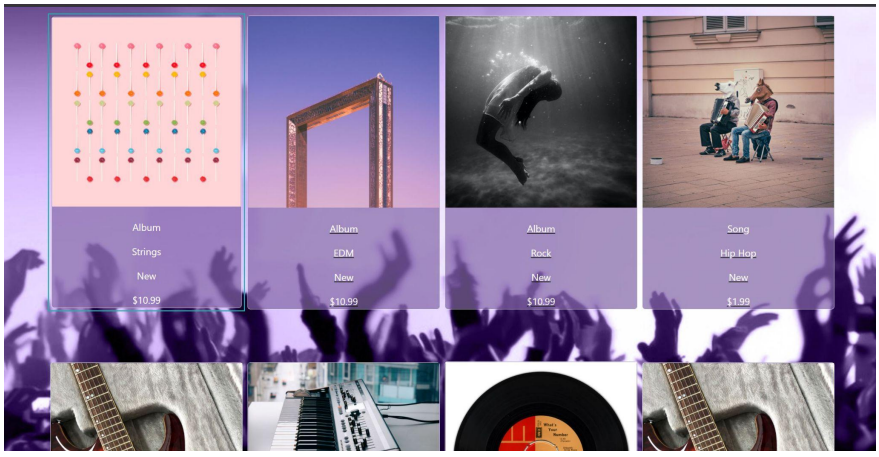


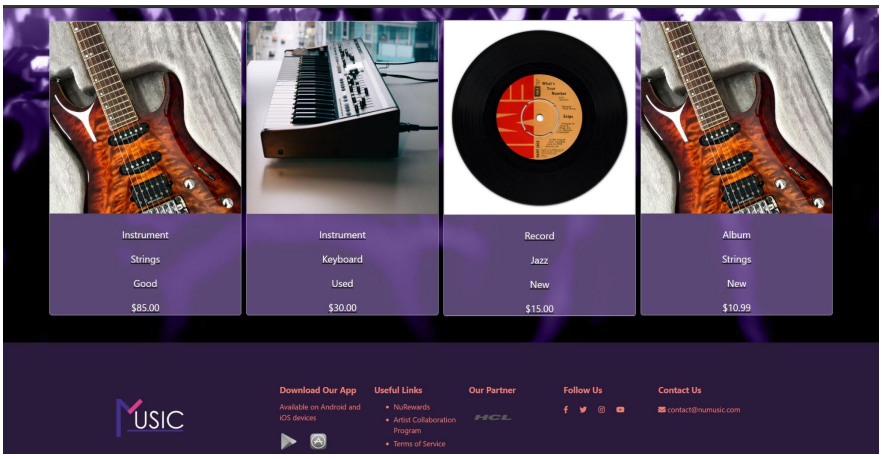


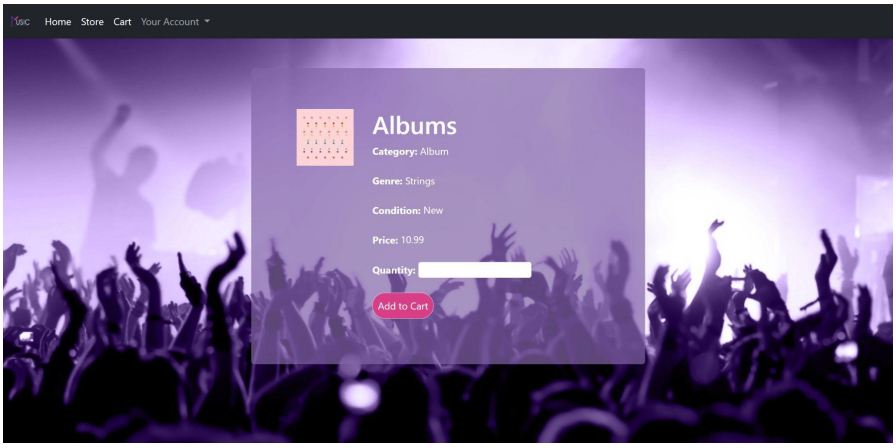
User Pages

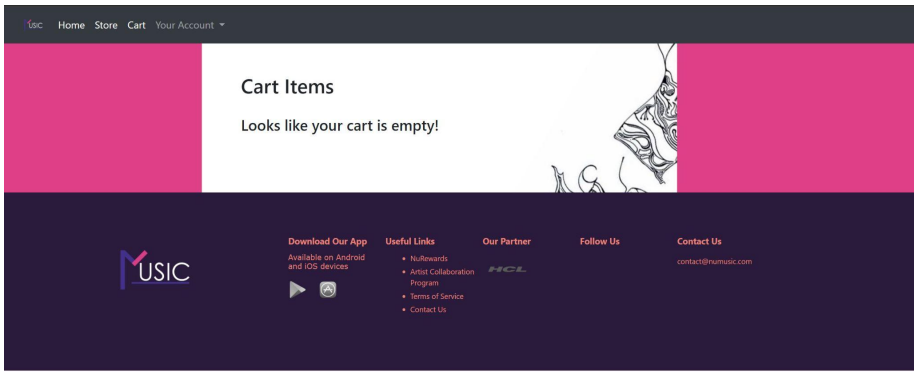

Home
Store
Cart
Your Account
Welcome test
View your orders

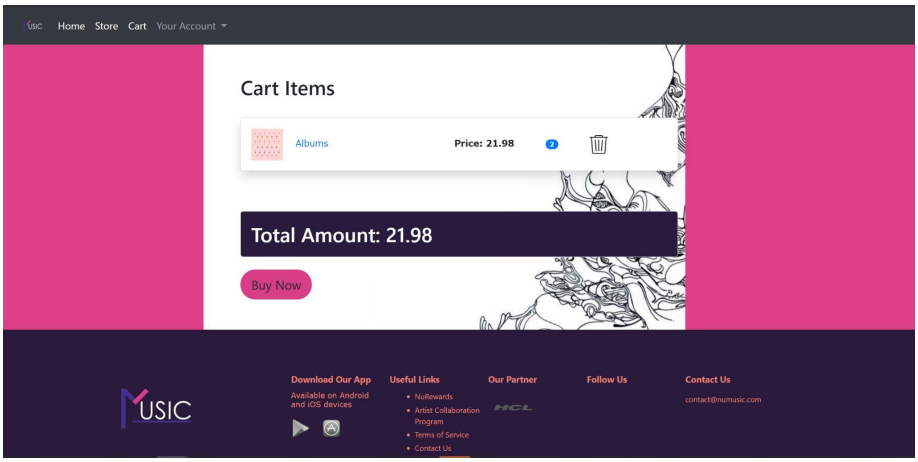

Home
Store
Cart
Your Account
Store Page
Search Our Catalog
Search
Album
Strings
Album
EDM
Album
Rock
Song
Hip Hop

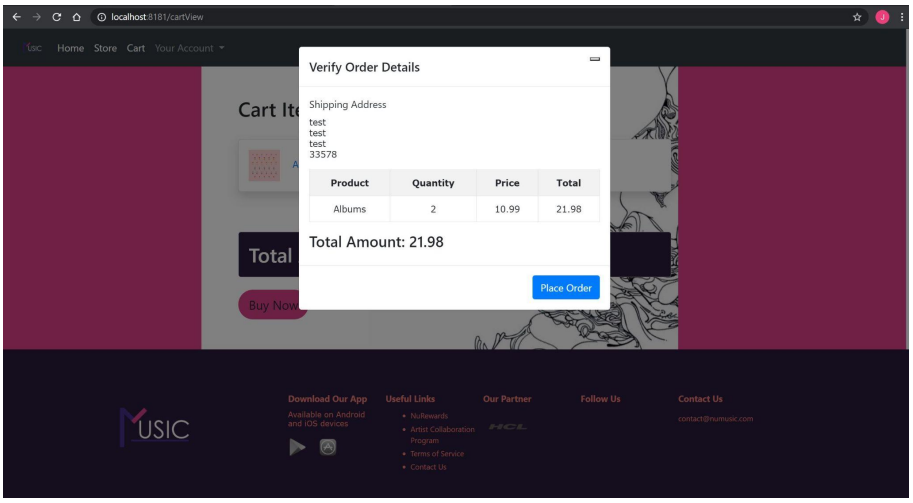

Album
Strings
New
$10.99
Album
EDM
New
$10.99
Album
Rock
New
$10.99
Song
Hip Hop
New
$1.99

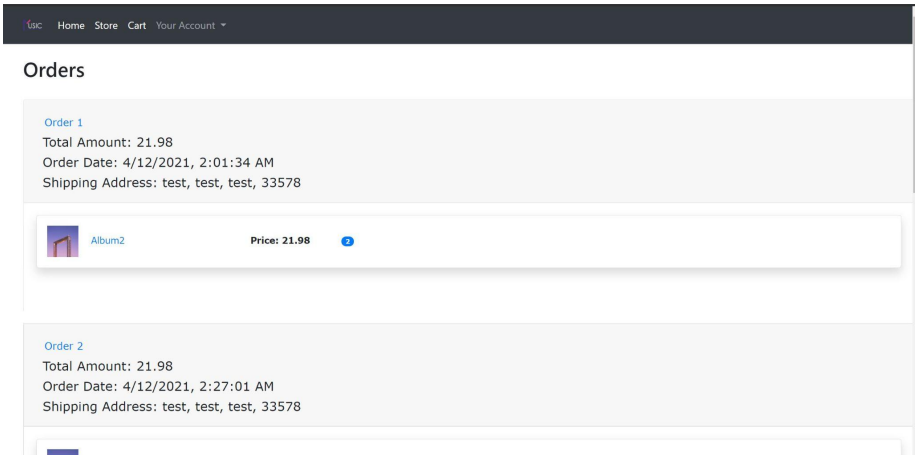

Instrument
Strings
Good
$85.00
Instrument
Keyboard
Used
$30.00
Record
Jazz
New
$15.00
Album
Strings
New
$10.99
MUSIC
Download Our App
Available on Android and iOS devices
Useful Links
NuRewards
Artist Collaboration Program
Terms of Service
Our Partner
HCL
Follow Us
Contact Us
contact@numusic.com


Home
Store
Cart
Your Account
Albums
Category: Album
Genre: Strings
Condition: New
Price: 10.99
Quantity:
Add to Cart

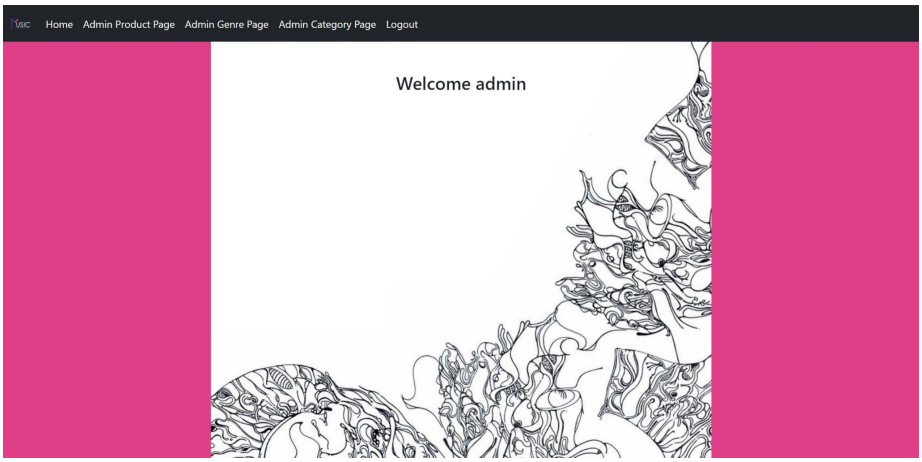

Home Store Cart Your Account
Cart Items
Looks like your cart is empty!
Download Our App
Available on Android and iOS devices
Useful Links
NuRewards
Artist Collaboration Program
Terms of Service
Contact Us
Our Partner
HCL
Follow Us
Contact Us
contact@numusic.com

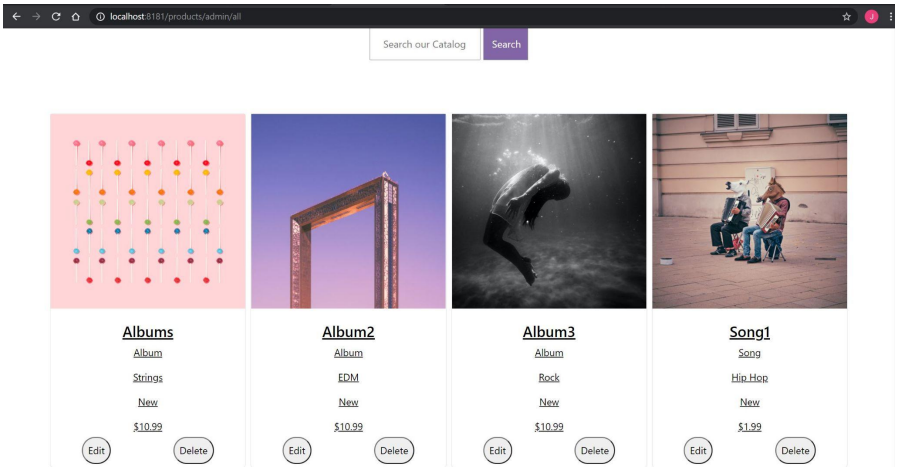

Home Store Cart Your Account
Cart Items
Albums
Price: 21.98
Total Amount: 21.98
Buy Now
Download Our App
Available on Android and iOS devices
Useful Links
NuRewards
Artist Collaboration Program
Terms of Service
Contact Us
Our Partner
HCL
Follow Us
Contact Us
contact@numusic.com

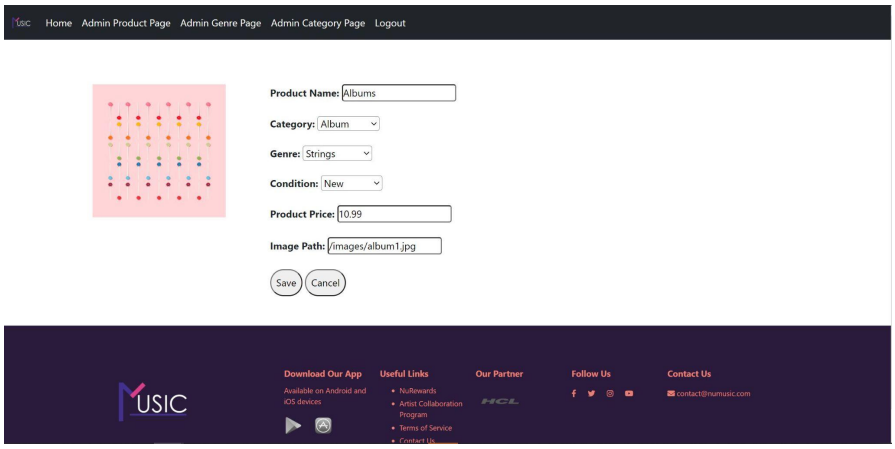

localhost:8181/cartView
Home Store Cart Your Account
Verify Order Details
Shipping Address
test
test
test
33578
Product Quantity Price Total
Albums 2 10.99 21.98
Total Amount: 21.98
Place Order
Total
Buy Now
Download Our App
Available on Android and iOS devices
Useful Links
NuRewards
Artist Collaboration Program
Terms of Service
Contact Us
Our Partner
Follow Us
Contact Us
contact@numusic.com

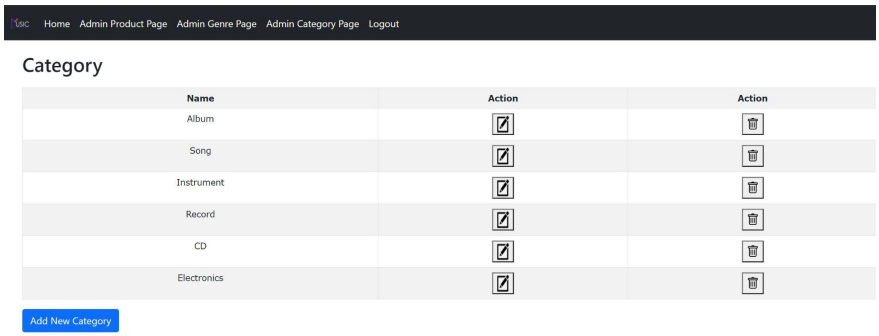

Home Store Cart Your Account
Orders
Order 1
Total Amount: 21.98
Order Date: 4/12/2021, 2:01:34 AM
Shipping Address: test, test, test, 33578
Album2
Price: 21.98
2
Order 2
Total Amount: 21.98
Order Date: 4/12/2021, 2:27:01 AM
Shipping Address: test, test, test, 33578

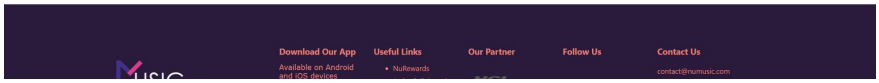
Admin Pages

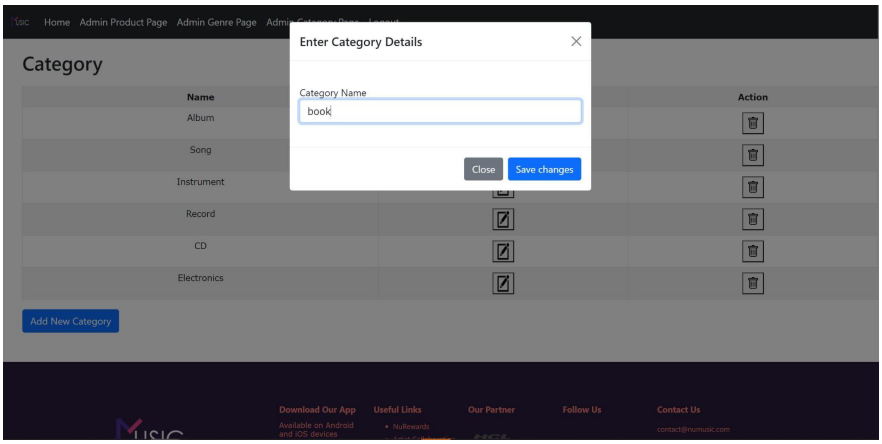

Home
Admin Product Page
Admin Genre Page
Admin Category Page
Logout
Welcome admin

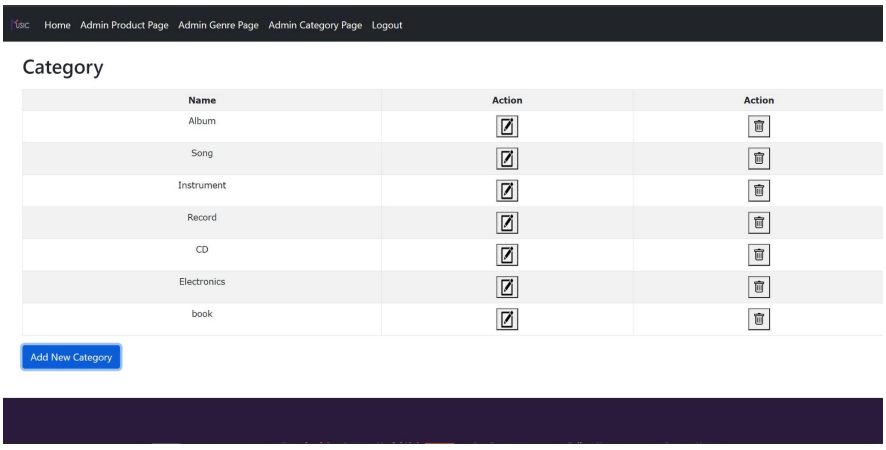

localhost:8181/products/admin/all
Search our Catalog
Search
Albums
Album
Strings
New
$10.99
Edit
Delete
Album2
Album
EDM
New
$10.99
Edit
Delete
Album3
Album
Rock
New
$10.99
Edit
Delete
Song1
Song
Hip Hop
New
$1.99
Edit
Delete

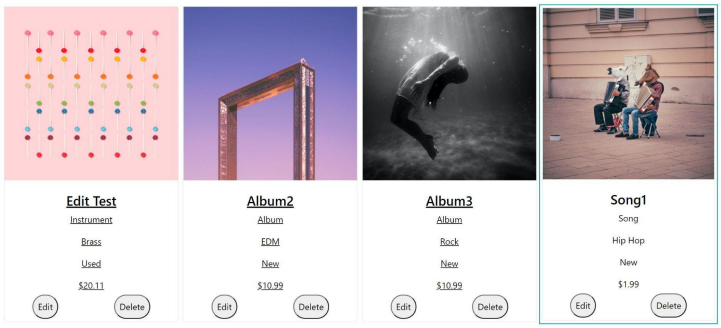

Home
Admin Product Page
Admin Genre Page
Admin Category Page
Logout
Product Name: Albums
Category: Album
Genre: Strings
Condition: New
Product Price: 10.99
Image Path: /images/album1.jpg
Save
Cancel
MUSIC
Download Our App
Available on Android and iOS devices
Useful Links
NuRewards
Artist Collaboration Program
Terms of Service
Our Partner
HCL
Follow Us
Contact Us
contact@numusic.com

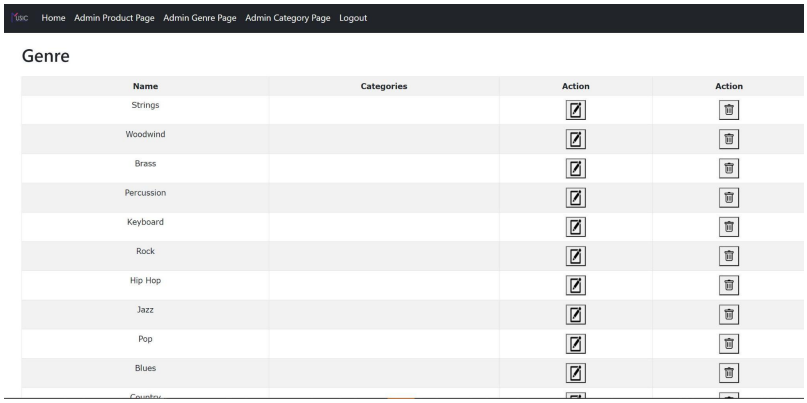

Home
Admin Product Page
Admin Genre Page
Admin Category Page
Logout
Category
Name
Action
Action
Album
Song
Instrument
Record
CD
Electronics
Add New Category

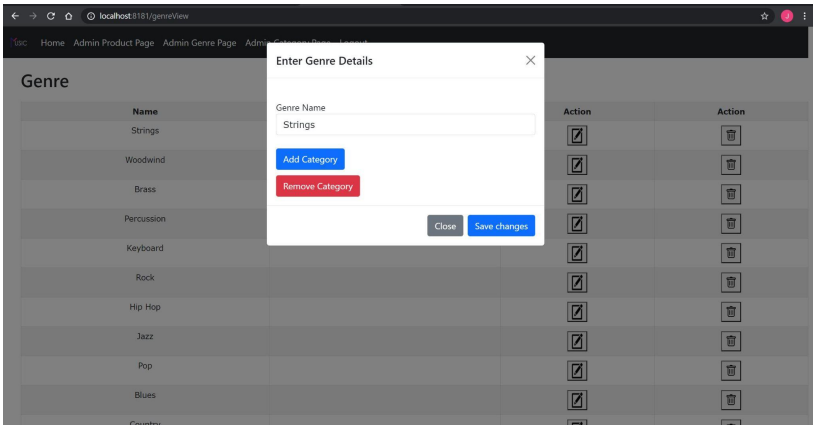

Download Our App
Useful Links
Our Partner
Follow Us
Contact Us

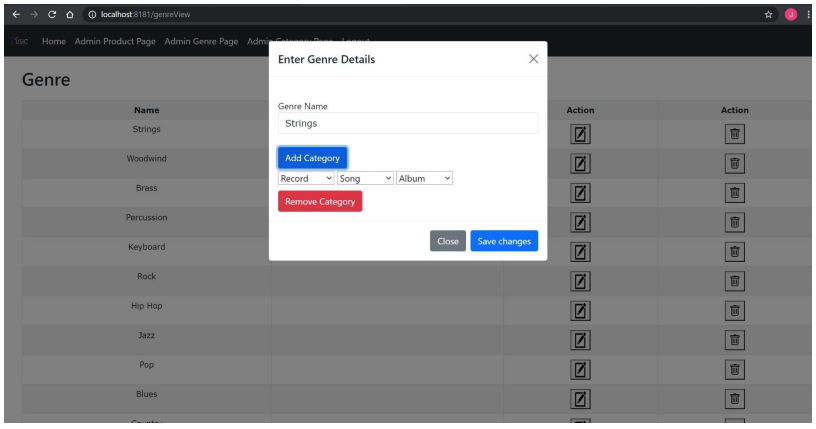

Enter Category Details
Category Name
book
Close
Save changes
Category
Add New Category


Home
Admin Product Page
Admin Genre Page
Admin Category Page
Logout
Category
Name
Action
Action
Album
Song
Instrument
Record
CD
Electronics
book
Add New Category


Edit Test
Instrument
Brass
Used
$20.11
Edit
Delete
Album2
Album
EDM
New
$10.99
Edit
Delete
Album3
Album
Rock
New
$10.99
Edit
Delete
Song1
Song
Hip Hop
New
$1.99
Edit
Delete


Home
Admin Product Page
Admin Genre Page
Admin Category Page
Logout
Genre
Name
Categories
Action
Action
Strings
Woodwind
Brass
Percussion
Keyboard
Rock
Hip Hop
Jazz
Pop
Blues


localhost:8181/genreView
Home
Admin Product Page
Admin Genre Page
Genre
Enter Genre Details
Genre Name
Strings
Add Category
Remove Category
Close
Save changes


localhost:8181/genreView
Home
Admin Product Page
Admin Genre Page
Genre
Enter Genre Details
Genre Name
Strings
Add Category
Record
Song
Album
Remove Category
Close
Save changes

GitHub Link

https://github.com/tug05734/capstone-music-store.git