

Software Project

**Adam Gallagher**

**N00211418**

Software Project

Online shop for bidding on second hand guitars

Year 2 2022-23

DL836 BSc (Hons) in Creative Computing

Link to resources created as part of the project.

|  |  |
| --- | --- |
| GitHub | <https://github.com/y2-SW-project/swproject23-AdamGallagher27> |
| Video | [Meeting with Adam Gallagher (Student)-20230420\_194611-Meeting Recording.mp4](https://iadt-my.sharepoint.com/:v:/g/personal/n00211418_iadt_ie/ET0_794FC_FDp8b1Sgdtgf8BOhWEKl28GBTl7LCGxf6b-Q?e=g3c3QV) |
| Figma | <https://www.figma.com/file/J2qKXjUh1eIxAHhf3QvflH/SoftwareProject?node-id=0%3A1&t=zXbYRD4nmdVdNlAD-1> |

Table of Contents

[1 Introduction 1](#_Toc96009471)

[2 Business Concept 2](#_Toc96009472)

[2.1 Business Idea 2](#_Toc96009473)

[2.2 Business model 2](#_Toc96009474)

[2.3 Market Research 2](#_Toc96009475)

[2.4 Marketing/Advertising 2](#_Toc96009476)

[2.5 Suppliers 2](#_Toc96009477)

[2.6 Competitors 2](#_Toc96009478)

[2.7 Employees 2](#_Toc96009479)

[2.8 Environmental Impact 2](#_Toc96009480)

[3 Requirements 3](#_Toc96009481)

[3.1 Introduction 3](#_Toc96009482)

[3.2 Requirements gathering 3](#_Toc96009483)

[3.2.1 Similar applications 3](#_Toc96009484)

[3.2.2 Interviews 3](#_Toc96009485)

[3.3 Requirements modelling 3](#_Toc96009486)

[3.3.1 Functional requirements 3](#_Toc96009487)

[3.3.2 Non-functional requirements 3](#_Toc96009488)

[3.3.3 Use Case Diagrams 4](#_Toc96009489)

[3.4 Feasibility 4](#_Toc96009490)

[4 Web application Design 6](#_Toc96009491)

[4.1 Layout 6](#_Toc96009492)

[4.2 Interaction 6](#_Toc96009493)

[4.3 Colour schemes 6](#_Toc96009494)

[4.4 Font choices 6](#_Toc96009495)

[4.5 Wireframes 6](#_Toc96009496)

[5 Database Design 7](#_Toc96009497)

[5.1 Description 7](#_Toc96009498)

[5.2 Business Reporting Requirements 7](#_Toc96009499)

[5.3 Textual Representation of Data-Set 7](#_Toc96009500)

[5.4 Business Rules 8](#_Toc96009501)

[5.5 Entity Relationship Diagram 8](#_Toc96009502)

[5.6 Tables 9](#_Toc96009503)

[5.7 Database Dictionary 10](#_Toc96009504)

[6 System Design/ Architecture Overview 11](#_Toc96009505)

[6.1 Introduction 11](#_Toc96009506)

[6.2 Model View Controller 11](#_Toc96009507)

[6.3 User Authenticaion 11](#_Toc96009508)

[6.4 Routing 11](#_Toc96009509)

[6.5 Templating 11](#_Toc96009510)

[7 Testing 12](#_Toc96009511)

[7.1 Introduction 12](#_Toc96009512)

[7.2 Functional Testing 12](#_Toc96009513)

[7.2.1 Login/Registration 13](#_Toc96009514)

[7.2.2 Navigation 13](#_Toc96009515)

[7.2.3 Calculation 13](#_Toc96009516)

[7.2.4 CRUD 13](#_Toc96009517)

[7.2.5 Discussion of Functional Testing Results 14](#_Toc96009518)

[7.3 User Testing 14](#_Toc96009519)

[7.4 Conclusion 14](#_Toc96009520)

[8 Project Management 15](#_Toc96009521)

[8.1 Introduction 15](#_Toc96009522)

[8.2 Project Phases 15](#_Toc96009523)

[8.2.1 Requirements 15](#_Toc96009524)

[8.2.2 Design 15](#_Toc96009525)

[8.2.3 Implementation 15](#_Toc96009526)

[8.2.4 Testing 15](#_Toc96009527)

[8.3 SCRUM Methodology 15](#_Toc96009528)

[8.4 Project Management Tools 16](#_Toc96009529)

[8.4.1 Github Project 16](#_Toc96009530)

[8.4.2 GitHub 16](#_Toc96009531)

[9 Reflection 17](#_Toc96009532)

[9.1 Your views on the project 17](#_Toc96009533)

[9.2 How could the project could be developed further? 17](#_Toc96009534)

[9.3 Assessment of your learning. 17](#_Toc96009535)

[9.4 Completing a large software development project 17](#_Toc96009536)

[9.5 Technical skills 17](#_Toc96009537)

[9.6 Further competencies and skills 17](#_Toc96009538)

[10 References 18](#_Toc96009539)

# Introduction

Overall aim

Application area

Technologies

PHP, MySQL, Bootstrap, CSS, Vanilla

Tools

IDE, phpMyAdmin, Miro

Project management

GitHub

Business Concept

Requirements

Design

Implementation

Testing

Reflection

# Business Concept

## Business Idea

The business idea is an online shop that allows users to sell their guitars or start bids for them. This will give an alternative and sustainable option for buying guitars instead of relying on buying guitars new from larger companies like Epiphone or fender.

## Business model

The business model is loosely based off how Depop works. Depop allows users to sell their clothes online and they take a commission off every transaction. The website would work in the same way.

## Market Research

The market for musical instruments was worth USD 14.20 billion in 2022, and it is anticipated to increase by 7.4% between 2023 and 2030 (Musical Instrument Market Size, Share & Trends Analysis Report by Type, by Distribution Channel) retrieved from https://www.grandviewresearch.com/industry-analysis/musical-instruments-market. There is a gap in the market for users to find and sell sustainably sourced guitars.

The most obvious demographic for guitars is musicians but there is also a market for music shops that don’t have their own online store. These two demographics make up most of the users for this platform. It also goes without saying not all customers are musicians or shop owners, some may just be people buying an instrument as a gift.

## Marketing/Advertising

Market research would be conducted to see how to connect with musicians online. Another strategy that was considered is getting a music influencer or Youtuber to endorse the product.

## Suppliers

There is no need for suppliers for this business as the users are their own suppliers as they are the ones selling and buying their own guitars.

## Competitors

The major competitors in this space are the traditional instrument distributors like fender or Epiphone. They have a major market share but most of them don’t have an option to purchase second-hand guitars, their goal is to sell their new products.

## Employees

As mentioned in the supplier’s section the customers would handle supplying their own guitars.

## Environmental Impact

While guitars are not the worst contributor to climate change, they are far from perfect. Some of the most endangered tree species are used for creating guitars for example Honduran rose wood or Pau Brazil. This store hopes to reduce the environmental impact.

From manufacturing new guitars.

# Requirements

## Introduction

This portion of the report is dedicated to the research that was conducted for the website.

This was used a precursor to the design phase. Use case diagrams where made and competitor analysis was done. In addition to these two interviews were conducted with two musicians.

## Requirements gathering

For the requirement gathering competitor analysis on fender and gear for music.

I also did two interviews with two musicians. They were the ones who recommended at fender and gear for music for the competitor analysis.

### Similar applications

The first competitor that was looked at was a more traditional guitar shop. Fender is the world’s largest seller of guitars. The main advantage of buying from Fender is you can trust the quality of their instruments are superb and you can guarantee what they advertise is accurate to the product. A major negative feature of fender is the products are sustainably sourced and there is no option to sell or buy second hand instruments.

A group of guitars

Description automatically generated with medium confidence

Figure 1: fender shop page

Their website design is very sleek and minimalistic. It is very easy to navigate and understand every detail about the product you are buying see figure 1.

A picture containing diagram

Description automatically generated

Figure 2: single product page fender

You can see again in figure 2 fenders minimalistic design. Figure two also highlights two of the issues with Fender. You can also see that the finger board is made from ebony wood which is an endangered wood. Despite the environmental impact and the lack of a second-hand shop feature the fender website is very well-designed website that is intuitive and aesthetic.

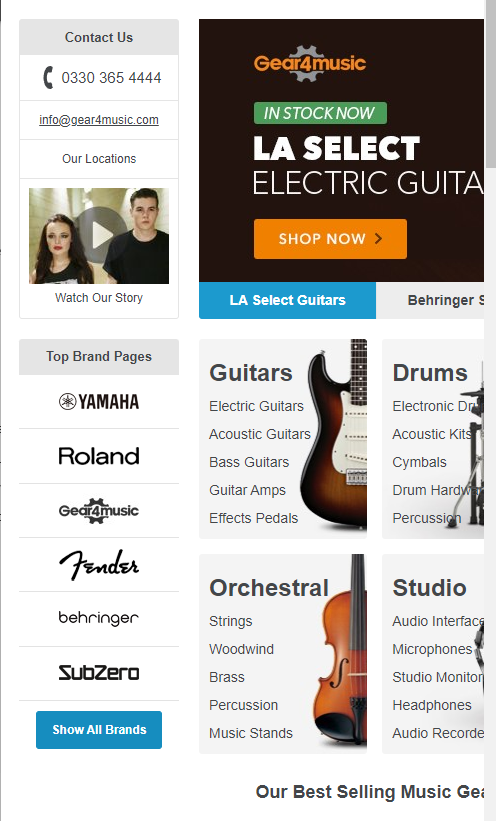


Figure 3 nonresponsive web design

The second competitor that was analysed was gear for music. Gear for music is one of the biggest online vendors of instruments in Europe. They have a huge variety of stock, and their website is very easy to use. Despite the websites size and practical functionality, the website is not designed very well and feels very clustered at parts see figure 4. The major drawback with gear for music is that their website is not responsive. When scaled down to a mobile size the components were overlapping and made it hard to navigate see figure 3.

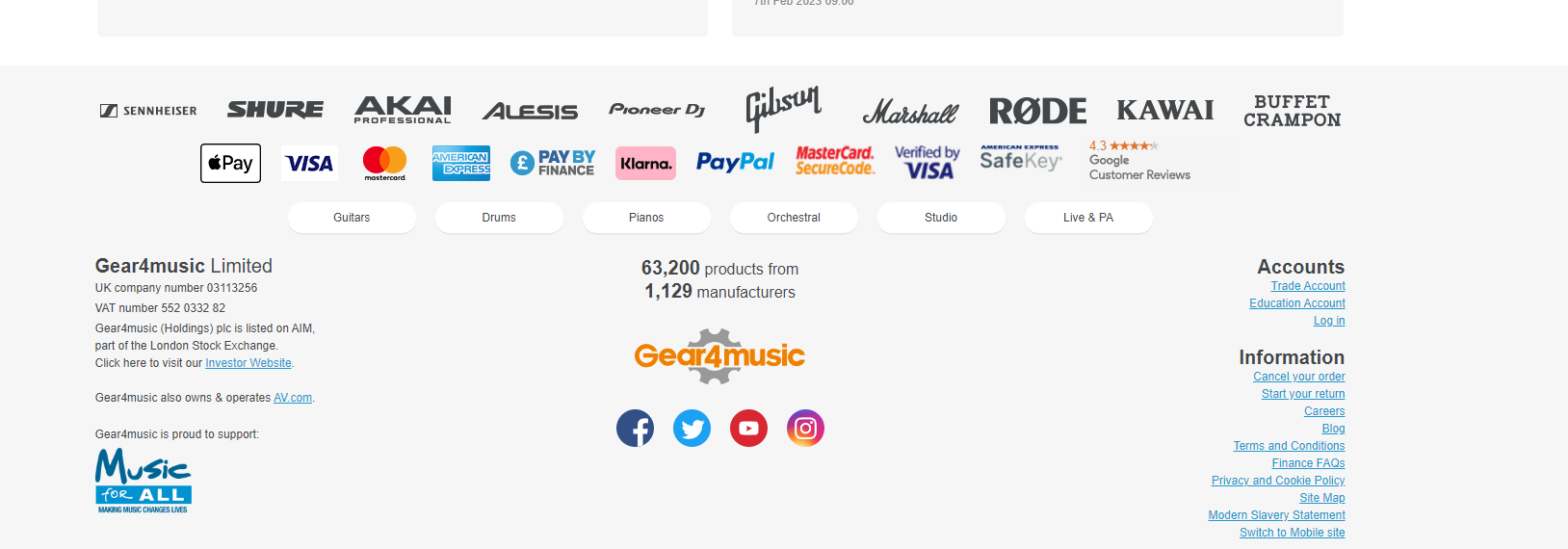


Figure 4 over cluttered footer

The best part of this design was in the product page. While the aesthetic was not perfect it was very functional and easy to make a purchase and see the specifications of each product see figure 5.

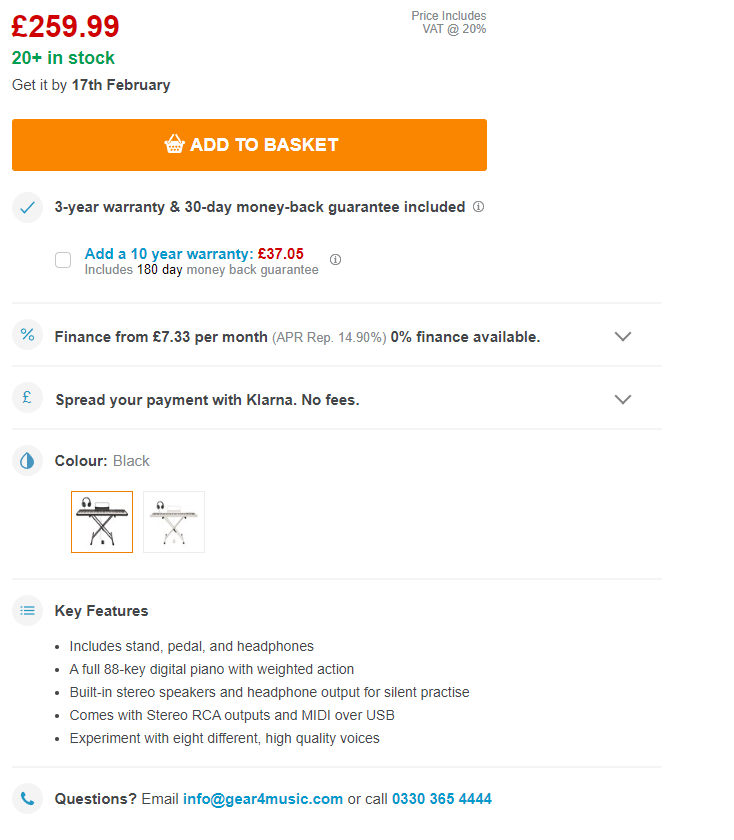


Figure 5 gear for music product purchase

The main problem with the aesthetic is the colour palette. The red of the price doesn’t pop enough and the green that symbolises the amount left in stock clashes with the other colours on the page.

### Interviews

Graphical user interface, text, application, email

Description automatically generated

Figure 6: interview questions

Interviews with two musicians where conducted. They were asked five questions for the interviews see figure three.

Question 1:

For the first questions the first user said they buy their instruments in their local music shop when the other users cited the fender website or gear for music for all their instruments.

Question 2:

The first user said they prefer buying in shop because they can test the instrument before buying it. The second user likes the convenience of shopping online and there generally very quick at delivering whatever she ordered.

Question 3:

Both users had the same answer that they wouldn’t mind bidding, but they would still like an option to buy the product outright. The first user reiterated that knowing the quality or sound of the instrument would help them make a purchase.

Question 4:

The first user said they want it to be easy to use and give many options for different types of instruments. The second user likes when the website gives detailed explanations of the specifications of each instrument.

Question 5:

Both users had the same answer again. They both disliked that you are never certain as to what you are buying, and quality may vary from different online vendors.

## Requirements modelling

### Functional requirements

1. Users can buy guitars.
2. Set up a shop and sell guitars.
3. Make bids or buy outright.
4. Admin uses can do crud on all post’s comments.
5. Comment on posts.
6. Wishlist / favourite guitars.

### Non-functional requirements

The three primary non-functional issues are Usability, Performance and Security.

I want my website to be as user friendly as possible I plan to do user tester to try remedy any issues in this section. I will eager load all my data and optimize my code to improve performance. Security is a very important issue when it comes to ecommerce websites. I would use a trusted API for handling purchases like snip cart.

### Use Case Diagrams

Diagram, engineering drawing

Description automatically generated

Figure 7 use case diagram

See figure seven for the use case diagram. This was used to map how users will interact with the website. Three roles where defined NormalUser, ShopUser and Admin.

## Feasibility

Laravel 8.1

Livewire 2.6

Tailwind 3.2.7

Php 8.1

JavaScript ES13

PhpMyAdmin 4.9

# Web application Design

## Layout

For my layout I will be using tailwind and livewire to have my design be responsive and have some of my components be reactive for example the like button and the bids on each post. I aim to have a minimalistic design that is heavily inspired by the depop and fender websites.

See figma file [here](https://www.figma.com/file/J2qKXjUh1eIxAHhf3QvflH/SoftwareProject?node-id=0%3A1&t=1xfspNLJbvhvOlpR-1).

## Interaction

The main feature of my website will be the ability to make posts (putting on instruments for sale) and the ability to make bids on them. For the user to make a post they would have to fill out a form which would gather information about the guitar they are selling. Examples of data would be the make, the condition, the colour. I want the bidding to be reactive. The product will list what the top bid is currently, and the user can raise that by putting in their own bid. This will update the database / current highest bid without a page refresh using livewire.

## Colour schemes

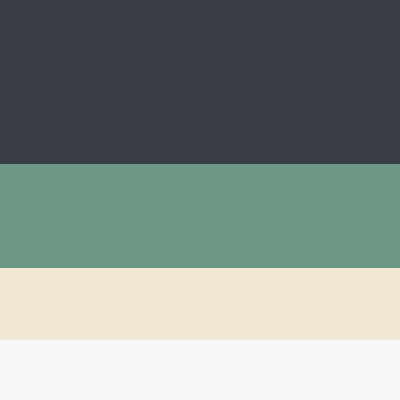


Figure 8 main colour palette

Before the design process was started several possible colour pallets where chosen. Example of this is seen in figure 8.



Figure 9 possible colour palette

The main palette (figure 8) was chosen as it is minimalistic, and the green primary colour is representative of the website’s sustainable values.

## Font choices

Text, logo

Description automatically generated with medium confidence

Figure 10 main font pairing

The font Karla was chosen for the titles and Spectral was chosen for the regular body text. See figure 10.

These were chosen as they are like fonts chosen on the fender website while still being unique to this website.

Graphical user interface, text, application

Description automatically generated

Figure 11 font palette

After the font styles where selected a font palette was created (figure 11). The design of the webpage is very minimalistic, and this is reinforced through the simple font palette.

## Wireframes



Figure 12 nav bar

Every page will have this nav bar implemented. This is essential to navigating the website. The capo text at the top left will take the user to the home page. Under that are different types of guitars. Clicking on these will take you to the search page (see figure 12) showing results with the selected guitar type.

Diagram

Description automatically generated

Figure 13 homepage

Figure 13 will display the top trending shops and products. At the bottom of the page there is an article section where it explains the process of selling and shopping on this website.

Calendar

Description automatically generated

Figure 14 search page

Figure 14 is the search page. When a user makes a search the result of their query will be shown here. Above all the products there are options to filter the search by type, condition or price.

Diagram

Description automatically generated

Figure 15 profile page

Figure 15 displays the current users posts and their liked posts.

Graphical user interface

Description automatically generated with low confidence

Figure 16 product page

Figure 16 is the individual product page. From here users can make bids and purchase guitars. They can also favourite the guitar from this page.

Diagram

Description automatically generated

Figure 17 user flow diagram

See figure 17 for a basic user flow diagram. This was used to understand the flow of the website and keep track of what features are to be developed for the website. Red signifies a page where yellow signifies an action. The log in / sign up is marked in blue as to not be mixed up with the other elements in the diagram.

# Database Design

## Description

This businesses database tracks many important details about the products and the users.

The database will store data about each guitar and its condition. For each product the name, type, price, bid-expiration date and the condition will be recorded. It will also track the bids made by each user and show who has won the bid when the bidding time expires.

Table for the three user roles (user, admin and shop) will also be created. Users favourited posts will be saved as well.

## Business Reporting Requirements

1. admin user need to do all crud operations on every product

2. shop users will be able to do crud operations on products they are selling

3. regular users won’t be able to do any crud operations but will be able to make bids on guitars.

4. regular users and shop users can favourite any post.

5. users will be able to search for products and filter their results.

## Textual Representation of Dataset

User (id, user\_name, real\_name, role\_id, email, password)

Guitar (name, description, make, type\_id, price, condition\_id, bid\_expiration, user\_id)

Type (id, name)

Condition (id, state)

Role (id, role\_name)

User\_Bid (user\_id, guitar\_id, bid\_amount)

User\_Like (user\_id, guitar\_id)

## Business Rules

One user has one role.

Many users can favourite one guitar.

Many users can bid on one guitar.

One condition can have many guitars.

One type can have many guitars.

## Entity Relationship Diagram

An entity relationship diagram (E.R.D) was designed. These are used to visually map the relationships of all the tables in the businesses database. See figure 18.

Diagram

Description automatically generated

Figure 18 E.R.D diagram

## Tables

The E.R.D see figure 18 is expanded upon and the tables are added in with all their elements in figure 19.

**Chart, box and whisker chart

Description automatically generated**

Figure 19 database tables

# System Design/ Architecture Overview

* 1. Introduction

The website uses Laravel as its main back end. Laravel is written in php and uses the Model, View, Controller (MVC) Design Pattern. For the front-end blade templates will be used for static components and live wire for reactive components. The components will be styled with tailwind CSS. Factories and seeders will be used during development to populate the database, Laravel has the functionality for this built in. Laravel breeze will be used to handle the user authentication.

* 1. Model View Controller

The model view controller has three components. The Model handles all the database logic. The view is the UI that is rendered to the screen. The controller is the brain of the app that controls how the model and view work together. Each table in the data base is represented by a migration and each table has a corresponding model. The model is used when adding data to the database in the seeders. It is also used to create the relationships between each table. Each page is a blade file and is stored in the views file. These are served to the user, and they see it in the browser when they go to the website. The controller handles all the logic for the website. Creating, updating and deleting is handled here. The bidding logic will have its own controller to handle the bids.

* 1. User Authentication

Laravel breeze is used for the user authentication. Breeze is an extension made by the creators of Laravel that implements the authentication features of Laravel in an easy-to-use package. The Authentication source code will be edited to add user roles and, in the sign-up form.

* 1. Routing

Each user and view will have its own routes. So, there will be the admin index route and the user index route. There will also be a special route to handle assigning user roles on signing up. This route will then redirect them to the correct route depending on their role.

* 1. Templating

Blade templates were used for the static components, but the reactive components will be made using live wire. Each component will be styled with Tailwind CSS.

# Testing

* 1. Introduction

The testing phase was an important step in the development of this application. Two types of testing were conducted functional and user testing. Functional testing showed bugs and logical errors in the code and user testing showed how users interact with the UI.

* 1. Functional Testing

The black box method was used for testing. The tester interacts with the website testing the features from the front end.

* + 1. Login/Registration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| A0 | Create new user with user role | User data | Home page with use role | Pass  Home page with use role | Make role select a drop down |
| A1 | Create new user with shop role | User data | Home page with shop role | Pass  Home page with shop role | Make role select a drop down |
| A2 | Login | User data | Home page | Pass  Home page | - |

* + 1. Navigation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| B0 | Tried to join admin, user and shop routes as no role user | Role URL | 402 error | Fail  No role user can access these | Fix authentication code |
| B1 | Tried to join different role URL than current users’ role | Role URL | 402 error | Pass  402 Error | - |

* + 1. Calculation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| C0 | Make bid | A number | Current bid changes | Pass  Current bid changes | - |
| C1 | Add guitar to favourite | Press like button | Guitar is added to like guitar table | Pass  Guitar is added to like guitar table | - |
| C2 | Buy guitar | Press buy button | Guitar shows sold component | Fail  Foreign key constraint fails error | Remove guitar from likes table before bought |
| C3 | Remove guitar from favourites | Press dislike button | Remove guitar from guitar like table | Pass  Remove guitar from guitar like table | - |

* + 1. CRUD

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| D0 | Create guitar | Press create button | Return to index with success message | Pass  Return to index with success message | - |
| D1 | Update guitar | Press update guitar | Return to product with success message | Pass  Return to product with success message | - |
| D2 | Delete guitar | Press delete button | Return to index with success message | Fail  Foreign key constraint fails error | Remove guitar from likes table before bought |

### Discussion of Functional Testing Results

The functional testing revealed bugs in the code and helped show logical errors in the code.

* 1. User Testing

User tests were conducted in person. Users would be assigned a task of creating a shop account and creating a new guitar. After they tried at the task, they took part in a post testing questionnaire. This showed user pain points and issues with the usability of the website. This was conducted with three users.

* 1. Conclusion

The functional tests showed issues with the code that the developer couldn’t find. The results of these tests can be seen in section 7.2 of the report and all bugs were fixed before submission. User testing showed errors in the design flow and usability of the website. The users then took part in a post-test questionnaire. The main issue that arose from this was the users couldn’t find the create button on the website. The rest of the results can be seen in the figma file. The design issues are still in the final website, but they would be changed in the next iteration of the website.

# Project Management

## Introduction

The project was managed mostly by using GitHub. Commits where made weekly to the project. The project was split into four phases. The requirements gathering, the design, the implementation and finally the testing phase.

## Project Phases

### Requirements

The requirement’s gathering section was the first. In this stage research was conducted on competitors. This helped inform the business plan. Creating a business plan was a challenge as the website relies on a critical mass of users to make money. The website also uses a bidding system to help stand out. The business also has sustainable goals. There is a gap in the market for this as most guitar vendors don’t have an option for sustainably sourced guitars. Interviews were conducted with two musicians to learn what they want from an online music shop.

### Design

After the research and business plan was complete the design phase began. The design was made in iterations. Firstly, use case diagrams were made to map the flow of ui and show what features are available to each user role. Colour and font pallets where made and then wire frames where designed. After this the websites ui was being designed. Later user testing would show some issues with this design. These would be fixed in the next iteration of the website.

### Implementation

This phase was about the development of website. The website was made using Laravel. One of the hardest features was allowing users to interact with the website with no role. The problem with this was that many of the components on the website used authentication code and when their user has no role this causes the website to crash. The website has a very basic search feature that takes a word and makes a where like request to the database and returns all records that fit the result. This works well but is limiting. A feature that could not be added was the ability to filter by genres. This feature would be added in another iteration.

### Testing

Two types of testing were conducted functional and user testing. The functional testing helped find bugs and allowed us to find bugs and logical errors in the code. The user testing phase showed some issues with the design. Every user had an issue with finding the create button. In future iterations I would address this issue in an updated version of the ui.

## Project Management Tools

### GitHub

GitHub is an online code repository. This was used to track commits and save all progress on the cloud. Commits would be made weekly. GitHub works by using an app called Git. Git tracks whenever a change is made in the client’s repository. Git then stages these changes and pushes them to the online repository. This allows changes to be made from many computers as the project is stored online.

# Reflection

## Your views on the project

I think the project all round was a success but there were a few issues. The main issue was in the design phase. I wish I spent longer designing and iterating on the design. These issues are seen in the final product.

## How could the project be developed further?

One of the main issues with the website is that it only sells guitars in future iterations I would like to expand on its products like guitar peripherals like amps or straps. I also want to see other instruments like drums be sold as well. A feature I didn’t get a chance to add to the final product was the ability to filter searches. This would be added in future iterations if given the chance.

## Assessment of your learning.

I think I worked well throughout the project making regular commits. Despite this I often would make too many changes on one day then wait to long to work on the project again. This made development difficult as I would start working on something then take a big break making it hard to restart what I was working on. In future I would make smaller changes spread over a longer period. I also feel I could have done more iterations to refine my design. This project developed my skills as a developer. I used tailwind and live wire in this project. I had a small bit of experience with these technologies but using them in this made me learn more about them.

## Completing a large software development project

I had never made a project this big before and I think it was an important and successful experience. Despite the success there was things I would do better in the future. As mentioned before in 9.3 I would have spread my work out better doing smaller changes more consistently. I also learned about the development life cycle from research to implementation. I learned about how important the stages before the implementation where. The database design took a long time to get it perfect before it was time to start developing. In the long run this saved time in the development phase.

## Technical skills

As mentioned in 9.3 I had to develop my skills primarily in two technologies I had some minor experience with. These two frameworks are tailwind and livewire. Tailwind is a css framework that gives many built in utility classes. I had some experience from my advanced web development project using tailwind, but this project had very little styling in it. This new project gave me a chance to develop some more complicated components and so develop my tailwind skills further. The second technology I learned more about was livewire. Livewire is a front-end framework that creates reactive components. Reactive components give websites more complex features that run on the client without refreshing the page. It does through ajax requests. This was used for the like button and bid modal.

# References

Adam Gallagher. (n.d.). *GitHub - AdamGallagher27/Advanced\_Web\_Dev\_CA* https://github.com/AdamGallagher27/Advanced\_Web\_Dev\_CA/tree/15\_like\_movies

*Coolors - The super fast color palettes generator!* (n.d.).

Coolors.co. https://coolors.co/

Depop. (n.d.). *Depop - buy, sell, discover unique fashion*.

https://www.depop.com/

*Etsy Ireland - Shop for handmade, vintage, custom, and unique gifts for everyone*. (n.d.). Etsy. https://www.etsy.com/ie

*Fender Guitars | Electric, Acoustic & Bass Guitars, Amps, Pro Audio*. (n.d.). https://www.fender.com/en-IE/start

Gear4music. (n.d.). *Gear4music Ireland | Musical Instruments, Equipment & Software*. www.gear4music.ie. https://www.gear4music.ie

*Google Fonts*. (n.d.). Google Fonts.

https://fonts.google.com/

*JavaScript | MDN*. (2023, April 5).

https://developer.mozilla.org/en-US/docs/Web/JavaScript

*Laravel - The PHP Framework For Web Artisans*. (n.d.).

https://laravel.com/docs/10.x

*Musical Instrument Market Size, Share & Trends Analysis Report By Type (Stringed, Percussion, Wind, Keyboard), By Distribution Channel (Offline, Online), By Region, And Segment Forecasts, 2023 - 2030*. (n.d.). https://www.grandviewresearch.com/industry-analysis/musical-instruments-market

*PHP: Documentation*. (n.d.).

https://www.php.net/docs.php

*Quickstart | Livewire*. (n.d.). Livewire.

https://laravel-livewire.com/docs/2.x/quickstart

*Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.* (n.d.). Tailwind CSS. https://tailwindcss.com/

Unsplash. (n.d.). *Beautiful Free Images & Pictures | Unsplash*. Unsplash. https://unsplash.com/