

Blazar

**Tariq Horan**

**N00202437**

Software Project

Develop an application based off a case study using Laravel

Year 2 2021-22

DL836 BSc (Hons) in Creative Computing

Table of Contents

[1 Introduction 6](#_Toc100676421)

[2 Business Concept 7](#_Toc100676422)

[2.1 Business Idea 7](#_Toc100676423)

[2.2 Business model 7](#_Toc100676424)

[2.3 Market Research 7](#_Toc100676425)

[2.4 Marketing/Advertising 7](#_Toc100676426)

[2.5 Suppliers 8](#_Toc100676427)

[2.6 Competitors 8](#_Toc100676428)

[2.7 Employees 8](#_Toc100676429)

[2.8 Environmental Impact 8](#_Toc100676430)

[3 Requirements 8](#_Toc100676431)

[3.1 Introduction 8](#_Toc100676432)

[3.2 Requirements gathering 8](#_Toc100676433)

[3.2.1 Similar applications 8](#_Toc100676434)

[3.2.2 Interviews 13](#_Toc100676435)

[3.3 Requirements modelling 13](#_Toc100676436)

[3.3.1 Functional requirements 13](#_Toc100676437)

[3.3.2 Non-functional requirements 13](#_Toc100676438)

[3.3.3 Use Case Diagrams 14](#_Toc100676439)

[14](#_Toc100676440)

[3.4 Feasibility 14](#_Toc100676441)

[4 Web application Design 15](#_Toc100676442)

[4.1 Layout 15](#_Toc100676443)

[4.2 Interaction 15](#_Toc100676444)

[4.3 Colour schemes 15](#_Toc100676445)

[4.4 Font choices 15](#_Toc100676446)

[4.5 Wireframes 16](#_Toc100676447)

[5 Database Design 20](#_Toc100676448)

[5.1 Description 20](#_Toc100676449)

[5.2 Business Reporting Requirements 20](#_Toc100676450)

[5.3 Textual Representation of Dataset 20](#_Toc100676451)

[5.4 Business Rules 21](#_Toc100676452)

[5.5 Entity Relationship Diagram 21](#_Toc100676453)

[5.6 Tables 22](#_Toc100676454)

[5.7 Database Dictionary 23](#_Toc100676455)

[6 System Design/ Architecture Overview 24](#_Toc100676456)

[6.1 Introduction 24](#_Toc100676457)

[6.1.1 What is Laravel 24](#_Toc100676458)

[6.1.2 What Version 24](#_Toc100676459)

[6.1.3 Dependencies 24](#_Toc100676460)

[6.1.4 Benefits 24](#_Toc100676461)

[6.2 Model View Controller 24](#_Toc100676462)

[6.3 User Authentication 24](#_Toc100676464)

[6.4 Routing 24](#_Toc100676465)

[6.5 Templating 24](#_Toc100676466)

[7 Testing 25](#_Toc100676467)

[7.1 Functional Testing 25](#_Toc100676468)

[7.1.1 Login/Registration 25](#_Toc100676469)

[7.1.2 Navigation 26](#_Toc100676470)

[7.1.3 Calculation 26](#_Toc100676471)

[7.1.4 CRUD 26](#_Toc100676472)

[7.1.5 Discussion of Functional Testing Results 27](#_Toc100676473)

[7.2 User Testing 27](#_Toc100676474)

[7.3 Conclusion 27](#_Toc100676475)

[8 Project Management 28](#_Toc100676476)

[8.1 Introduction 28](#_Toc100676477)

[8.2 Project Phases 28](#_Toc100676478)

[8.2.1 Requirements 28](#_Toc100676479)

[8.2.2 Design 28](#_Toc100676480)

[8.2.3 Implementation 29](#_Toc100676481)

[8.2.4 Testing 29](#_Toc100676482)

[8.3 Project Management Tools 29](#_Toc100676483)

[8.3.1 GitHub Project 29](#_Toc100676484)

[8.3.2 GitHub 30](#_Toc100676485)

[9 Reflection 31](#_Toc100676486)

[9.1 Your views on the project 31](#_Toc100676487)

[9.2 How could the project be developed further? 31](#_Toc100676488)

[9.3 Assessment of your learning. 31](#_Toc100676489)

[9.4 Completing a large software development project 31](#_Toc100676490)

[10 References 32](#_Toc100676491)

# Introduction

## Overall aim

The aim of this software project is to modify and improve fashion trading sites to be more accessible so the site can reach an older demographic.

## Application area

Fashion.

## Technologies

PHP, MySQL, Bootstrap, CSS

## Tools

IDE, PHPMyAdmin, Miro, Figma, GitHub

## Project management

GitHub

## Business Concept

Trading shoes and clothes that are rare or in high demand.

## Requirements

The application must allow a user to browse products and if they choose, they can buy or sell products.

## Design

This project will take inspiration from preexisting websites and will have its own design identity.

## Implementation

The project will gradually progress starting from the planning and sketches of the website then later the functionality will be completed.

## Testing

The application will be constantly tested while being built and by getting users to complete a task on the site and record results.

## Reflection

The finished project will be reviewed on how much functionality is actually working and it ticks all brief requirements.

# Business Concept

## 2.1 Business Idea

The business idea is to develop an online marketplace where you can buy and sell clothes and shoes. Some clothes and shoes that are released are hard to get due to resellers and regional restrictions. This shop allows consumers and sellers from all around the world to trade the products.

## 2.2 Business model

The business will make money by taking a percentage of the sale price on a sale by a seller for example 15%

## 2.3 Market Research

People of all ages but mainly genZ, people will buy and sell products everyday as there is a constant demand for clothes and shoes. They use marketplace type sites to sell unwanted or very valuable products. The market is constantly expanding with the demand for high end and limited goods.

## 2.4 Marketing/Advertising

The business will be advertised on social media as the market is predominately young that’s where they can be reached easiest.

## 2.5 Suppliers

The users off the site provide the products as they are selling them through the application.

## 2.6 Competitors

Sites like stockx and goat are the sites that effectively execute this service on their respective applications.

## 2.7 Employees

The project will require a quality control team, an advertising team, a customer service team, and an application admin.

## 2.8 Environmental Impact

The application is all online no physical shop the main environmental impacts the business has is shipping and packaging products.

# Requirements

## Introduction

It is important that I have requirements for my application so I can figure out what I need the site to do. Not only for a user but for the admin too so I can keep on track on what needs to be working.

The user will be able to login when they use the application (they don’t have to), first they will need to register an account which will be stored in the database. The user will have to login using an email and password, they will also need to enter their shoe size and general clothing size.

There will be an admin for the application so for when they login they will have control over certain aspects of the application. They will have CRUD functionality so that it is easy to manipulate the contents of a certain page by altering the database.

The user will be able to browse products, filter and sort products and also bid on the products they desire. There is no cart, as far as purchase goes there will only be a successful bid pop up as a visual confirmation.

## Requirements gathering

This section is important because it is good to identify the exact requirements I need at the start of the project, I am basically analyzing similar applications to use their requirements as a template.

### 3.2.1 Similar applications

o Screenshots

Graphical user interface, website

Description automatically generated

Figure 1 The homepage of stockx

Graphical user interface

Description automatically generated with low confidence

Figure 2 The browse page for sneakers with some sort and filter options

Graphical user interface

Description automatically generated

Figure 3 The product view page with ability to bid

Graphical user interface

Description automatically generated with medium confidence

Figure 4 The bidding and selling page for products

A picture containing text

Description automatically generated

Figure 5 GOAT homepage

Graphical user interface, chart

Description automatically generated

Figure 6 The browse shoes page for GOAT

Graphical user interface

Description automatically generated

Figure 7 The product view page with ability to bid

Graphical user interface

Description automatically generated

Figure 8 The product view page with ability to bid

### o Descriptions

The screen shots above are the basic flow for the process of buying a pair of shoes on both stockx and GOAT, it starts off on the browse page and it quickly allows you to buy or bid for a pair of shoes. I found that stockx was slightly easier to use and clearer.

### o Advantages

Both sites have a quick way to find products as they both have search options.

Pictures of the products with clear descriptions.

Coloured buttons to differentiate between different services.

### o Disadvantages

Prices are for specific sizes which is misleading.

Inexperienced users can find the searching process hard on GOAT as there is no clear browse link.

On GOAT the buying process uses one click too many and can be frustrating

### 3.2.2 Interviews

In the interviews I conducted I asked the participant to carry out a task on each competitor site. I asked the participants to find a particular shoe and get as far as they could in the purchasing process without actually buying it.

The general takeaways from the interviews were that the site layout is clear but there is some jargon that the testers couldn’t quite understand. The other important note in the interviews was that they didn’t understand the pricing on the products as they are different for each size and also the processing and delivery fee varies without any explanation, the GOAT site is slightly different in that there are multiple buying options which they also found confusing because there is no explanation.

## Requirements modelling

This section is used to map out the different requirements we need, I need to have the ideas for the functional and non-functional requirements planned out to meet the requirements. The functional requirements are important, but they don’t stop the application from working. The non-functional ones must work so users can use the application

### 3.3.1 Functional requirements

Users will be able to browse products

Users can register/login

Users can bid or sell

Admin has Create, Read, Update and Delete (CRUD) functionality over application

### 3.3.2 Non-functional requirements

### o Usability

The application should work in all browser sizes and still keep its design identity

The application should have the ability to go back to a page easily or back to the homepage

### o Performance

There will be visual signs that inform users on how their request is coming along e.g., a spinner when they bid on a product

### o Security

When a user registers or logs in they must follow a password convention that requires certain characters

If the bid is successful, they should be redirected to the correct page

### 3.3.3 Use Case Diagrams

## Diagram Description automatically generated

## 3.4 Feasibility

For the project I will be using Laravel, MySQL, GitHub and Bootstrap. I will be using Laravel to build the application, this is where I will be making web pages and controllers to control the flow of the application. Laravel has Bootstrap built into it so I will be using it to build the webpages using HTML and CSS. Laravel will be communicating to the MySQL database where I will be retrieving data to populate my application. This will all be hosted on GitHub where I can access and change files remotely.

# Web application Design

## Layout

The application will be fully responsive as I will be using Bootstrap which makes it easy to manipulate elements on the page. The layout is simple I will use a lot of cards for the products and a lot of forms for things like logging in and buying products.

## Interaction

The user will use the buttons located at the top of the page to navigate through the application there will be clear indications on what page the user is on as it will be in the URL, and I plan to implement a breadcrumb feature on each page.

## Colour schemes

The website will have a dark theme with the background on most pages being black and the text being white. I will use a light blue to accent the page just to break it up a bit. For alert pop ups or messages, I will use red to make it stand out and most of the background on titles will have either a grey or white shade.

# 343A40 Colour for Navbar

# 91929F Text colour

# E43A36 Colour for sale button

# 3490DC Colour for buy button

# 38C172 Colour for bid button

## Font choices

For the titles and headings, I will be using Garamond and for the text on the main body I will use Montserrat. I chose Garamond because it stands out as the thickness of the text is small and I chose Montserrat because it has a lot of styles which will help when italicizing or highlighting text.

## Wireframes

Graphical user interface, website

Description automatically generated

Figure 1This is the homepage

The user can login, sign up or browse the website from this page

Graphical user interface, text

Description automatically generated with medium confidence

Figure 2 Page 2

This is essentially another homepage the user can see featured shoes and new shoes

Diagram

Description automatically generated

Figure 3 This is the page the user can browse products on

The user gets here by clicking the browse shoes or browse clothes button they can also sort or filter through the products

Graphical user interface, diagram

Description automatically generated

Figure 4 Product view page

The user gets here by clicking a product on the browse page it shoes a description of the product and the buying options.

# Database Design

This is where I determine what data needs to be stored and the way the data is stored. This then allows me to add content to the database.

## Description

My company will require two products tables one for clothes and one for shoes, the tables will store the product name, price, description, popularity, and the image. The company will also keep track of the number of products sold. There will also be a user table where customers have to login to purchase or sell an item and the admin will have to login to have CRUD functionality.

## Business Reporting Requirements

Admins need to be able to create, read, update, and delete festivals, stages, shows, performers, and genres.

Users will need to be able to find all products and sort through them and also add products to the database.

Users need to find all products using a list of genres.

Users need to find the products using a product name.

Users need to be able to bid on products.

## Textual Representation of Dataset

**SHOES** (id, Name, price, description, popularity, image\_id, genre\_id),

**CLOTHES** (id, Name, price, description, popularity, image\_id, genre\_id),

PRODUCTS\_SOLD (id, product\_id),

**GENRE** (id, trainer, boot, lifestyle, classics, grails)

**IMAGE** (id, filename)

**USERS (id, first\_name, last\_name, email, password)**

**CUSTOMERS (id, users\_id)**

## Business Rules

 A **Product** has many customers.

 A **customer can buy many products**

 A **Product** belongs to one **Genre**

 A **Product can have one image**.

## Entity Relationship Diagram

Diagram

Description automatically generated

## Tables

*A diagram of a house

Description automatically generated with low confidence*



## Database Dictionary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table** | **Attribute** | **Datatype** | **Range** | **Required** | **PK/FK** | **FK Ref Table** |
| Shoes/clothes | Id | Int |  | Yes | Pk |  |
|  | Name | Varchar |  | Yes |  |  |
|  | Price | Float |  | Yes |  |  |
|  | Description | Text |  | Yes |  |  |
|  | Popularity | Varchar |  | Yes |  |  |
|  | Image\_id | Int |  | Yes | Fk | Image |
|  | Genre\_id | int |  | Yes | fk | genre |
| Genre | Id | Int |  | Yes | Pk |  |
|  | Trainer | Varvhar |  | Yes |  |  |
|  | Boot | Varchar |  | Yes |  |  |
|  | Lifestyle | Varchar |  | Yes |  |  |
|  | classics | varchar |  | Yes |  |  |
| Customers | Id | Int |  | Yes | Pk |  |
|  | User\_id | int |  | yes | Fk | User |
|  |  |  |  |  |  |  |
| User | Id | Int |  |  |  |  |
|  | First\_name | Varchar |  |  |  |  |
|  | Last\_name | Varchar |  |  |  |  |
|  | Email | string |  |  |  |  |
|  | password | String |  |  |  |  |
| Image | Id | Int |  | Yes | pk |  |
|  | filename | varchar |  | yes |  |  |

# System Design/ Architecture Overview

* 1. Introduction

This outlines the interfaces that I will use. In this section of the report I also explain the processes that happen behind the scenes in the interfaces.

### What is Laravel

Laravel is a web application framework that aids the development of projects by allowing programmers to access databases, make routes to different webpages etc.

### What Version

The version of Laravel that I am using is Laravel 8.

### Dependencies

Laravel uses composer to download packages that allows Laravel to run, Laravel also uses MySQL to access databases that serve the webpage

### Benefits

Laravel is quick and simple, although there is a lot of folders and files to get lost in once you get used to the application you find it is immensely powerful. Having everything in the on place allows users to make great applications.

* 1. Model View Controller

To start off the process the browser sends a request to the Controller. The controller then communicates with the model and exchanges data when that’s done, the View will send its final presentation to the Controller and the Controller will then display that data from the user to interact with.

* 1. User Authentication

Laravel has a built-in authentication feature that uses the attempt method, it allows user to enter an email and password that must meet a character requirement. This can also be done manually.

* 1. Routing

The route application requests to a certain controller and the route accept a Uniform Asset Identifier. These URL’s are both human are easily accessible by humans and don’t have to directly point to any files in the application

* 1. Templating

The templating engine used in Laravel is Blade. Blade is based off PHP and has a remarkably similar syntax it also lets you use plain PHP while some templates don’t.

A diagram of a flowchart

Description automatically generated with low confidence

# Testing

* 1. Functional Testing

The project is constantly tested. Every time I add a new section or add functionality the site, I test it before I write any more code. Every error is fixed, and the site is functioning as intended

* + 1. Login/Registration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | User logs in with a valid email and password | [example@blazar.ie](mailto:example@blazar.ie)  \*\*\*\*\*\*\*\*\*\* | You are now logged in | You are now logged in | Pass |
| 2 | User enters an unknown email and/or password | [unknown@blazar.ie](mailto:unknown@blazar.ie)  \*\*\*\*\*\*\*\*\*\* | These credentials do not match our records. | These credentials do not match our records. | Pass |
| 3 | User enters an email without the @ sign with a valid password | Example.ie  \*\*\*\*\*\*\*\*\*\* | Please include a @ the email address example.ie is missing a @ | Please include a @ the email address example.ie is missing a @ | Pass |

* + 1. Navigation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | Click home button | http://127.0.0.1:8000/ | Blazar homepage | Blazar  homepage | Pass |
| 2 | Click browse button | http://127.0.0.1:8000/user/shoes | Blazar products page | Blazar products page | Pass |
| 3 | Click view product | <http://127.0.0.1:8000/user/shoes/> {id} | Blazar product view page | Blazar product view page | Pass |

* + 1. Calculation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | User buying product, shipping calculated |  | Shipping added to product price | Shipping added to product price |  |
| 2 | User selling product, selling fees calculated |  | Fees subtracted from product selling | Fees subtracted from product selling |  |
|  |  |  |  |  |  |

* + 1. CRUD

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | Admin edits product in the database | http://127.0.0.1:8000/admin/shoes/1/edit | Redirected to http://127.0.0.1:8000/admin/shoes | Redirected to http://127.0.0.1:8000/admin/shoes | Pass |
| 2 | Admin deleted product from the database | http://127.0.0.1:8000/admin/shoes | Product is deleted | Product is deleted | Pass |
| 3 | Admin adds product to the database | http://127.0.0.1:8000/admin/shoes/create | Product is added to database | Product is added to the database | Pass |

### Discussion of Functional Testing Results

The results were positive anything that failed was a quick fix because they were minor syntax errors.

* 1. User Testing

User testing is having a user carry out tasks on your project. The tasks are recorded and the results are analysed to make the project better.

### User Testing Findings

I tested my application with my classmates and also with friends and family. There wasn’t much variation in the feedback. The main criticism was that wasn’t a filter and sort option on the product browse pages. This was something I was looking to implement into my application but didn’t get a chance to. Some of the users also suggested that I change the colour of the buttons they felt as though they “the sell button should stand out and be almost like a warning”.

* 1. Conclusion

The overall results were satisfactory the admin has CRUD functionality over the application as intended. The user can do everything that was expected in the requirements for example they can buy, bid, and sell products. The user and admin can also login and users can also sign up to the site.

# Project Management

## Introduction

In this project I talk about how I manage my project the issues that I faced during different sections and also the tools I used that helped me complete the project.

## Project Phases

In this section, describe each of the following project phases. Explain any issues which arose for each of the phases.

### Requirements

I didn’t run into any major problems in this section. It was quite time consuming trying to figure out how the functionality of the site would work, planning all of this while also doing interviews was time consuming but it wasn’t a major problem.

### Design

This part of the project was interesting to me because I was able to analyze similar websites to the one, I am attempting to create and refine it to the way I think looks best. The only issue was I couldn’t really put a lot of detail into my wireframes so when creating my webpages, they won’t be as complex as I might have wanted, maybe if I have time at the end, I will enhance the design.

I had no problems with the system design, but I am currently still working on my database design and getting that perfect before I move on, so it doesn’t cost any problems in the future.

### Implementation

Implementing the work, I have done so far has gone well, I had to refresh my memory on some of the folder paths and commands needed in Laravel but once I got them it was easy to navigate and complete the work I needed to complete. I have also been working consistently over the course of the project only missing classes due to covid, this helped me greatly and allowed me to stay on top of the project.

### Testing

I am constantly testing my site for bugs; I don’t push to GitHub without all known problems being fixed. Every time I add new functionality to the application or change an element, I always use it as if I am a user to make sure it works and fits in with the aim of my application.

## Project Management Tools

These are different tools that I used to keep track of my project and display my progression from beginning to end.

### GitHub Project

This is where I log my progress manually, I explain what I need to do and what I am working on. This allows me to always keep track of what I am working on and how my progress is coming along in terms of completing the project. I have kept relatively on top of my project so far, so my GitHub project is quite tidy and well managed at this moment in time.

A screenshot of a computer

Description automatically generated with medium confidence

### GitHub

GitHub is where I am putting my code, so it is kept safe in one place. Every time I change a section of the application, I push my changes to GitHub so then the project is updated locally and on GitHub. If something were to happen to my local file or if someone else wanted to alter my project, they can pull it down locally.

So far, my GitHub repository has been working perfectly. I had set it up properly at the start of the project, so I haven’t run into any problems now further into the project, I find it quick and easy to push my code and continue working after.

# Reflection

## Your views on the project

I feel as though the project went well, I achieved what I set out to do. It was interesting to research and get a deeper understanding of the sites I was trying to replicate, and it was also interesting to have control over the whole project and trying to manage the whole application.

## How could the project be developed further?

I would have liked to implement a sort and filter feature on the product pages, it would also be interesting to have a better post method when uploading an image to the database. I believe these features would greatly improve the application

## Assessment of your learning.

I have developed my understanding in PHP, I achieved this by writing loops and programming in the language blade which is very similar to PHP. I also had the chance to get more familiar with Bootstrap which helped me create the different sections of the website quickly. During this project I became more comfortable with GitHub by committing regularly.

## Completing a large software development project

Developing this project helped me get a better idea of how to keep up to date with the project progress which is very important in relation to the workplace. The format of this projects submissions (GitHub, Report Sections, Kanban) it helped me be more organized and let me stay on top of the project making it less stressful.

# References

*Goat: Sneakers, apparel, accessories*. (n.d.). Retrieved April 12, 2022, from https://www.goat.com/

Mark Otto, J. T. (n.d.). *Bootstrap*. Bootstrap · The most popular HTML, CSS, and JS library in the world. Retrieved April 12, 2022, from [https://getBootstrap.com](https://getbootstrap.com)

*Sneakers, streetwear, trading cards, handbags, watches*. StockX. (n.d.). Retrieved April 12, 2022, from https://stockx.com/

*Where developers learn, share, & build careers*. Stack Overflow. (n.d.). Retrieved April 12, 2022, from https://stackoverflow.com/