Custom Guitars

**Team:**

David Byrne X00203114

Adrian Donnelly X00194620

**Supervisor:**

Ben Walsh

Student Compliance with Academic Integrity

All students are expected to complete their courses in compliance with university regulations and standards. No student shall engage in any activity that involves attempting to receive a grade by means other than honest effort, for example:

1. No student shall complete, in part or in total, any examination or assignment for another person.
2. No student shall knowingly allow any examination or assignment to be completed, in part or in total, for himself or herself by another person.
3. No student shall plagiarize or copy the work of another and submit it as his or her own work.
4. No student shall employ aids excluded by the instructor in undertaking course work.
5. No student shall knowingly procure, provide, or accept any materials that contain questions or answers to any examination or assignment to be given at a subsequent time.
6. No student shall procure or accept assignments from any other student from current or prior classes of this course.
7. No student shall provide their assignments, in part or in total, to any other student in current or future classes of this course.
8. No student shall submit substantially the same material in more than one course without prior authorization.
9. No student shall alter graded assignments or examinations and then resubmit them for regrading.
10. All programming code and documentation submitted for evaluation or existing inside the student's computer accounts must be the students original work or material specifically authorized by the instructor.
11. Collaborating with other students to develop, complete or correct course work is limited to activities explicitly authorized by the instructor.
12. For all group assignments, each member of the group is responsible for the academic integrity of the entire submission.

**N.B to be filled out by each member of the team.**

By including my name in the form below, I declare that I understand and will abide by the University Regulations and Policies covering Academic Integrity. I accept that each member of the group is responsible for the academic integrity of the entire submission. I will retain a copy of this agreement for future reference.

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name (1):** | **David Byrne** | **Student No.:** | **X00203114** |
| **Student Name (2):** | **Adrian Donnelly** | **Student No.:** | X00194620 |
|  |  |  |  |
|  |  |  |  |
| **Module title:** | **YR2 Project** | | |
| **Programme Title:** | Bachelor of Science (Hons) in Computing with Software Development (2022 – present. | | |
| **Date: DD/MM/YYYY** | **22/04/2024** |

**Table of contents:**

**1. Cover Page**

**2. Declaration of academic Integrity**

**3. Outline of each members Participation**

**4.Introduction to the project**

**5.Project Plan and Requirements**

**6.Report for each iteration**

**7. Concluding paragraph**

**Project Breakdown**

**David Byrne:**

For my part in the project,

**Shopping cart functionality** allowing a user to add/remove/delete the cart. Authenticate if a user is logged in allowing them into the cart page. I also done the html for the shopping cart page and the thank you page displaying the customer’s order.

**Payment System**, I used Stripes payment system that would allow a user to make payments in the shop displaying a customer’s total in euros.

**Voucher System** that’s added in the cart page displaying a summer code at the top once an item has been added into the cart calculates the price discount and updates the price if another item is added into the cart. Then the voucher is tied to the account and is good to a set date and time in the admin side.

**Email Functionality** so that a user can be emailed order details once an order has been placed sending them a breakdown of the order what they ordered and how much it cost.

**Order History,** customer can click into their account into a page I made called order history allowing them to see all past orders the name of the item, the order id, and the price of the item they paid for.

**Search/Advanced Search ability**, search bar up the top of website allowing a user to search for items in the database it checks the items name if its upper case or lower case. Advanced search for the brand of an item min/max prices, Aswell as a paginator to only display a set number of products per page and if it goes over the limit a button saying next page will appear.

**Wishlist**, on each product page beside the compare product button there is a add to Wishlist button allowing a user to add an item to their Wishlist once they are logged in. Items in the Wishlist update their stock availability with the data base displaying if an item is in stock or out of stock.

**Adrian Donnelly:**

**UI Desing and implementation:** I designed the UI in Figma and implemented it within the website using bootstrap Html CSS and JavaScript. The primary design goal was simplicity and a good user experience, and I believe that we have achieved that.

**Featured products:** On the home page I implemented a featured products section. This displays featured products within a selected category. To change a product to feature an admin ticks a box on our admin panel that selects a product as a featured product.

**Order report printing:** Developed functionality in the admin panel to allow admins to select and print PDF reports of orders.

**Custom Admin panel:** Custom Admin panel with our name and logo in the header and change colours to suit the website better. I used Django-admin-tools to create graphs depicting the total number of users on a given day and the total number of orders. These graphs can be changed to use days/weeks/months, a bar graph/line graph and change the to/from dates.

**Compare Products:** Users can select up to 3 products from the product catalogue to compare.

**Image Viewer:** Users can click on the images within the product page to view them in more detail.

**Gmail API: Connected emails to Gmails API to have them sent out to a user's email address.**

**Product review system:** I designed a product review system which allows users to comment on products. A user can only comment on a product once. Implemented a rating system that allows customers to rate products on a scale,1 to 5 stars.

**Accounts:** I implemented the following for the accounts portion of the website:

Accounts overview page: this displays the users information.

Edit Account: This allows the user to edit their details.

Sign in/Sign out, Change Password,

**2FA:** A 32bit seceret code is generated on signup, this along with our email and the users email get fed into a QR code generator. This QR code can be scanned by the user with their authentication app, and it will generate a new 2FA code every 30 seconds. This code is the same as the one being sent to the user's email.

**Products:** I designed a product system that allows an admin to create a new product with images, descriptions and more that will get displayed when the user clicks into a certain product page. On this product page is buttons to add to cart, Wishlist and compare product. The review section, the products images and the description.

**Show Room:** Developed a show room in Three JS that allows the user to choose between multiple different guitars and view them in 3D. Some of the main fewatures of the show room are directional lights that cast reflections on the guitars, shadow casting. antialiasing for smoother reflections. Orbit controls allowing the user to move the guitar, rotate the camera and zoom in and out. The background is made using a High Dynamic Range Image, this is a 360 image that gets wrapped around a sphere giving the illusion of depth.

**Unit testing:** I Implemented unit testing and URL for orders, cart, accounts and for products and categories. these tests validate functionality, ensuring smooth transactions, secure authentication, and an accurate catalogue to display.

**Introduction to Custom Guitars**

For our second-year project we brainstormed up ideas that we could do for the project and we both have a passion for guitars so we thought a guitar website would be a great idea allowing a user to sign up search through our guitar brands and amp stock and purchase. We had learned from last semesters project when we done a gaming website where you could order games what we done wrong and where we fell down at and what improvements we would bring in this site. Adrian had an idea of a customizable guitar in a 3d model a user could change to make their own guitar but due to time constraints we broke the idea down into a show room showing three guitar models in a 3d environment allowing a user to see it spin it around and get an idea of what guitar they wanted.

Iteration 1

When we started work on the project, we decided we would go with an agile method approach so at the start of each week we would get on a call or meet in college and say what we got down over the week what issues we had and what we intend on doing for this week’s push like the scrum meetings we would have with our project coordinator. We used a site called Figma to design our project and how we intended it to look with a guitar icon as the logo for our store and a dark background with the text for products categories and guitars to a have a bright colour to stand out and catch a user’s attention.

Which the first iteration of the site we wanted to try and get a big push on having major parts of the site done like the cart functions allowing a user to add remove delete their basket and checkout using stripe payment and being able to apply a voucher to their basket. With the voucher system we tested it by generating a test voucher code and loaded a dummy product into our categories and assigned a value of 250 to the product and then gave a voucher the code test with twenty percent discount to see if it works.

We had the bones of the site done where the categories, products pages worked we then added in the login, logout, edit and sign up. Using Djangos built in forms for it we loaded the models up with the details we wanted like name dob address email. Which would allow the user to sign up if they haven’t already made an account, then if they want to change anything they must verify the current account details to then update it.

With the website prototype we implemented the basic search function to the site as we haven’t full flushed out our products and categories for the advanced search for the moment, we tested it with 2 dummy products in our products data base to make sure if a user searches for an item it will come up and allows the user to click on the product image bringing it to its product page which then loads the reviews for the product down the bottom of the page.

# Test

# Issues

When we were working on the shopping cart, we came across an issue from the data base where we had forgot to make migration command and migrate it to then load the database that would then load the data and would link into the products showing what the user had picked and allowing them to add or remove an item.

With the products page we noticed the categories wasn’t loading the image when you clicked categories but when u went into the products page for a set item the image would load, we looked at the html for the categories page, and it was just a simple line of code to point to the static media folder was missing.

When we were working on the review system for the products, we wanted the user to be logged in and to not have left a review before to be able to leave a review. users who weren’t logged in were able to leave a review after a quick change in the way the review system interacts with the account model it was fixed.

Another issue with the review system was when a user left a review the button to submit a review would stay on the account, where we wanted it to disappear after a user submitted a review just showing what their comment was and star rating.

Iteration 2.

**Work Breakdown**

**David:** Shopping cart, order history, Stripe payment checkout, Search Functionality, Cart Functionality add/remove/full remove, Vouchers, User auth for shopping cart, Email Customer order details html/CSS split between team.

**Adrian:** UI Desing and implementation, Custom Logo, Featured products, Order report printing, Product review system, Account – Profile one to one mapping. 2FA Emails and Authenticator,2FA QR code generation, Accounts overview page, Edit Account, sign in/Sign out, Change Password, Accounts models, Categories, Products, Configurator-Models, Lighting and Textures in ThreeJS, URL unit testing.

**Progress**

During this iteration, the team made significant progress on various aspects of the project. Each team member was assigned specific tasks that would suit their skills and expertise the best.

The UI design and implementation was a key focus of this Iteration, centring around creating an intuitive and visually appealing user interface. Custom logo creation was another key task completed during this phase, ensuring our brand identity is well-represented throughout the platform.

Account management functionalities received attention during this iteration, primarily user sign in/Sign out and the implementation of Two-Factor Authentication via email and authenticator apps enhances the security of user accounts.

Shopping cart Functionality was a big goal for this iteration. We allocated significant time to ensure users can seamlessly add, remove, and manage items within their carts. This included implementing features for smooth navigation, updating quantities, and enabling users to easily review and modify their selections before proceeding to checkout.

In this iteration, some of our focus was on implementing the core functionality of the 3D guitar configurator, this includes loading guitar models into the configurator and enabling users to view them in a dynamic 3D environment.

**Testing**

The shopping cart functionality was tested by introducing temporary products into the database and simulating user interactions. Upon adding products to the cart, the shopping cart icon located at the top right of the site updated to reflect the items added by the user. Following this, tests were conducted to ensure the operation of adding, removing, and deleting items from the cart, as well as testing the checkout process, which integrated the Stripe payment system. After a successful transaction, users were directed to a thank-you page displaying their order details alongside an order number and an email was sent to their email address. There were issues throughout development such as the products button to add remove and delete when clicked would do nothing but when All of these issues were fixed the system worked seamlessly.

When an order was placed, the customer could go to their profile and see their order history and a bug occurred where it wouldn't pull the customers data correctly from the database. The solution to this was to make sure the user was logged in to place an order so the order history's views and model could pull the info from the users account and load correctly and the admin can go into the orders and see all orders made on the backend.

After a customer makes an order, we thought it be good to implement an email function which takes in the customer profile which stores the customers details and would trigger after an order has been placed. Initially it was tested using Django's inbuilt email service to display it to the console to make sure it's working. After researching Django's email functions, we discovered a temporary folder could be set up, where the emails will be sent to, showing order details.

Due to OTP/2FA being new to everyone on the team this was difficult to implement. Originally it was meant only to be sent through emails, but we discovered that Authentication apps such as the one used in TUD (Microsoft Authenticator) were able to be used. To implement this a full redesign of the login system was required, this produced many bugs and errors, such as the bug where OTP codes were generated differently between emails and the Authenticator apps, we discovered that in order to generate the same codes across platforms, a secret key needed to be generated on signup, which will be fed into a QRcode generator along with the users email and the user can scan this and have a code generated every 30 seconds on these apps that is able to be used as a 2FA code.

**Iteration 3 Report**

**Progress:**

We completed a considerable amount this Iteration. Many features were polished out and made up to a standard we were happy with. Each task was distributed to the team member who had more knowledge in that area, because of this we worked well together and achieved a lot.

**Unit Tests**

Implemented unit testing to test the main aspects of our code such as the product and category models in shop, The order and order item models in order, The Cart and CartItem models in cart and the custom user model in Accounts. These help to ensure that our code is up to standard and works correctly.

Unit tests in the Shop app validate Category and Product creation in the database by creating a product and related category and testing to see if they exist within the database.

Unit tests in the Accounts app validate user account creation by creating a test user and validating that it exists in the database.

**Show room**

Our original Idea for this was a customisable guitar configurator, unfortunately due to one of our teammates dropping out of the project the was not a reasonable goal as we thought this would impact on our ability to complete the integral features of this project. We decided to repurpose the code that had been written to work as a show room. This was developed using Three JS and allows the user to choose between multiple different guitars and view them in 3D. Some of the main features of the show room are directional lights that cast reflections on the guitars, shadow casting. antialiasing for smoother reflections. Orbit controls allowing the user to move the guitar, rotate the camera and zoom in and out. The background is made using a High Dynamic Range Image, this is a 360 image that gets wrapped around a sphere giving the illusion of depth.

**Conclusion**

After working through the 3 iterations on the project we talked about what we think went good for us. Next time we would choose our team better as we had to drop a teammate and it added to our workload where we had to drop the 3d configurator for a 3d showcase which was good but not the full scope of what we wanted to do and was more of a secondary thing to do as we wanted to focus on getting the main functionality of the website done. We looked into adding WhatsApp and text implementation and had it working to the extent that if it was the number we used to in Django's hardcoded variable it would text the order details to you, but to implement a complete WhatsApp feature we would of had to of made a new business account which would wipe your personal account and you would have to pay for the business one. After getting main functionality done with our limited time, we would've liked to of got the stripe payment screen to load up a user's details to auto populate it like it would on the browser, and for a user to be emailed a voucher upon sign up. We are happy though with the final iteration of the site, we got major functionality done and added in extra things like QR code authenticator that lets a user sign in scan the authenticor and link it to their phone so when they login again the app will display the code for them.