

Ca2

web dev play framework SECOND REPORT

Howie Lynch x00125128

Phil O’Donovan x00132046

Shane McCann x00128429

# Brief Overview of Report 1

In our previous report we explained the designing and creation of our website. In brief we started off by using a basic written out template drawing to what we would like our website to look like. We browsed other similar websites that which sold products similar to ours to find the good and the bad that these websites incorporated. We then drew more templates for how many different pages we would have in our website, we ended with four with multiple product pages. We then found a suitable bootstrap template and edified it to our specification and needs. Once that was completed we chose a color schema and inserted all of our pictures. By adding a main. Scala pages and all routes linking to our images we then had a functionally website inside the Play framework.

# Development of Server Side Functionality

In making our website to server side functional we had a few obstacles to overcome. The CSS of our website when introducing it to Play Framework we found we had multiply errors. Majority of these were that the images would not work when put into the play framework. These were because the images had no routes.Assets.versioned, this meant that we had to change all our image links versions to routes.assets.versions and then the link beside for example (“img/guitars/ Gibson.jpg”). Also we had problems with linking our links to other pages in the play framework. These problems stemmed from the navigation bar having dead links. This was fixed by adding all our pages inside the home controllers page inside the home controller file. Thus having all our links like HomeController.guitars() or similar.

Inserting our products into the database was a bit of a problem. We found that by doing the labs we were able to understand and complete the add delete and update methods to our database. We found the Login to be a major problem. We spent a lot time figuring out the admin.controller.products() we realized after a while that we did not have functionality at the start of the admin controller class. In the end we did not add this function into our website. To be able to get to the admin you enter the url localhost9000//login. Although the login and password work they are not secure as anybody user or administrator can edit it at will. Also we got the error unique constraint on primary when trying to insert our sql into our database. We knew this was that there is a constraint on the primary key but we could not get around it. Our solution was instead of updating the database with our sql we added our products inside the h2-browser by using the add product button.

Another problem we had was syntactical errors. This caused us all to have problems at some stage from a comma missing to a semi-colon or whatever the syntax error was. This made us learn a lot about the play framework and also that to double check your code before trying to run the website. A very useful piece of play framework is that it can tell where your error is resulting from, although sometimes this still is of little help.

Additional Functionality of Website

After finishing our website to a standard we feel is acceptable, there was a few additional functions we would off liked to add to our project. We tried to add a shopping cart to our website so that the user could click on an image/product and if they would have wanted to purchase it, it would be added to the shopping cart. Then while in the shopping cart it would update are database and delete from the stock we had in our database. This proved more problematic than we thought and we were unable to produce the code to replicate this in our project.

Overall Review Off Project

We as a group feel this project has taught us a lot about a lot of things. Coding in how we read and write code, how certain methods work. Using the play Framework was a huge learning curve, although we had used it in the semester in the labs, working on our own was the biggest learning curve. It taught us a lot of things about modern websites are build and are made dynamically. Another huge thing this project taught us was teamwork, we feel together as a team this projected was completed on time and nearly all to full specifications that were asked in the brief. A lot of times one of us came to a block of code that gave us errors but we worked together as a team and were able to overcome the obstacles we faced.

Work Load

Again as in the first part of our website we shared out all the work evenly. Each member of our group contributed to the overall development and completion of our web play framework project. We all worked together staying back after college in labs and the library. This made it easier to communicate our problems and to also help each other with individual problems. Each of us contributed to every aspect of the project and would like to be graded equally.

Page Indexes

Localhost:9000/

Website index (Home page)

Localhost:9000/guitars

Guitars page

Localhost:9000/drums

Drums page

Localhost:9000/keyboards

Keyboards pages

Localhost:9000/amps

Amps page

Localhost:9000/mics

Mics page

Localhost:9000/lighting

Lighting page

Localhost:9000/events

Events page

Localhost:9000/contact

Contact and info page

Localhost:9000/products

List of available products for customer to view

Localhost:9000/addProduct

When within Admin privileges, webpage with empty contents shows to add a Product

Localhost:9000/addProductSubmit

After a product is added the page is then put into the databse using a POST page

Localhost:9000/admin/products

Shows which products are within the database and can be edited or removed if you have the privileges

Localhost:9000/admin/addproduct

Allows the admin to add a product

Localhost:9000/admin/addproductsubmit

Creates entry to database after adding a product

Localhost:9000/admin/delProduct/:id

Deletes product with a certain ID

Localhost:9000/updateProduct/:id

Updates a product with a certain ID

Localhost:9000/login

Displays a login page

Localhost:9000/loginSubmit

Accepts login details and process’ credentials

Localhost:9000/logout

Logs the user out