BSc in Software Development

Year 3

COMP07030 Software Design Project

*Shopfront with database*

*G00314054*

*Liam Joyce*

Contents

[**Introduction** 3](#_Toc449366139)

[**Installation** 3](#_Toc449366140)

[**Technologies used** 4](#_Toc449366141)

[**Problems Encountered/Solved** 5](#_Toc449366142)

[**Conclusions** 5](#_Toc449366143)

[**Recommendations** 6](#_Toc449366144)

Student Number: G00314054

Student Name: Liam Joyce

Supervisor: Kevin O Brien

GitHub Link: https://github.com/Tangler203/3rdYearProject

# **Introduction**

This project is a storefront where the user is able to search for products and place an order.

The products are sourced from a database

The store calculates the price of the order and saves it and the list of products ordered into the database.

The main aim of the project was developing a web app that is able to read and update a database on a server while also exploring ways to integrate JavaScript, a client-side language, with a server side language.

Objectives include connecting to a database and using its data in JavaScript and vice versa, place data from JavaScript in the database. I will also look to bring data from one page to another, something that can’t be done with JavaScript.

# **Installation**

Required: Wamp server, MySQL database, any browser – preferably Google Chrome

* Index.html: This can be anywhere.
* Checkout.php/order.php/results.php/layout.css: These must be put in directory called “project” in the www folder of the Wamp server
* In MySQL create a database called storedb and import “storedb.mysql” into it.
* The MySQL Server is accesed with the username “root” and password “” (No password)

# **Technologies used**

Languages used:

* JavaScript
* Html/Css
* PHP

Database/webserver Used:

* MySQL
* Wamp Server

Development Environment:

* Notepad++/ Google Chrome

The website consists of one HTML file, one css file, four PHP files and a MySQL database accessed through a Wamp server. I wrote the files using notepad++ and created the database using a MySQL console.

I used JavaScript as I’m most familiar with it. However I have only used for making simple mobile apps and wanted to try doing something new with it.

I chose PHP as the server side language as that integrates well with both html and MySQL and had not used it before. I chose MySQL as the database as it suited my needs the best and integrated well with PHP as mentioned above. The Wamp server was used as it came with MySQL database.

The code for the project was written in notepad++ and tested on Google Chrome. I used this because it was easy to set up and run without much difficulty.

# **Problems Encountered/Solved**

Originally I had planned to use NodeJS as my Web server. However while I was able to write to the database I had set up from an html page I was unable to format the data received to a satisfactory level. Furthermore I creating pages became an issue as the webserver would go new page upon gathering data. This new page would have nothing but the new data on it and I was unable to work with it. To deal with this issue I changed my server to Wamp and immediately saw an improvement as I could now get the data on webpages I had already written.

During development I was using a session object to bring an array of ids from one page to another. However I could not make the application cycle through the array automatically without throwing up errors. This was because the application searched for the array but in the array itself. I instead used a Get method which was already an array and could use a for-each loop to cycle through the values.

# **Conclusions**

*This section should discuss what you learned from the process and from the work you did.*

*This is a key reflective portion of the write up.*

From the project I drew the following conclusions:

* JavaScript is a rather obtuse language when trying to work with servers. While it can work server-side languages, being a client-side only language it can run into multiple problems such as not receiving variables easily.
* PHP is an extremely flexible language. I had issues with it at the start as I was not used to its syntax. Towards the end though I found could much more with it if had more time

# **Recommendations**

* Mobile hub: The project could be further enhanced with access from the mobile. After an order is placed a technician working for the shop could access the database and could check which orders were assigned to him. From there they could write in saying whether the order was sent or not.
* Better Search: The search in index.html could be expanded from a simple string search to include search by category or price range. The search could then be refined to sort by name or price.
* Session Objects: Bring objects between webpages with a session object. This method is more secure than a get method which I had used in the project. Issues with it though prevented me from using it.
* Wamp: If I could start the project again I would begin with Wamp from the start. I had changed to it from NodeJS rather late into the project and was not able to work on the project as much due to issues with the web server.