**A red shield with white circles on a black background

Description automatically generated**

**Price Comparison Website.**

Student Id: M00964494

Student Name: Morolari Boluwatife Emmanuel

Student Email: BM913@live.mdx.ac.uk

Table of Contents

[Introduction 3](#_Toc153342406)

[Website Description 3](#_Toc153342407)

[Screen shots 4](#_Toc153342408)

[Database Design Diagram 6](#_Toc153342409)

[Testing Documentation 6](#_Toc153342410)

# **Introduction**

In today's digital era, the abundance of laptop options across retailers overwhelms consumers, hindering informed decisions. To simplify this process, I've developed a concise laptop price comparison website. Scraping data from Amazon, Ebuyer, Laptops Direct, Box, and StockMustGo. I focused on Apple, HP, and Lenovo laptops.

The website's core features are:

**Homepage:** Featuring highlighted laptops and current deals, along with a user-friendly search function.

**Product Listings:** A clear display of searched products.

**Price Comparison:** Real-time comparison of the same laptop across the five retailers.

# **Project Description**

Technologies Used

**Spring:** Employed for dependency injection and managing dependencies within the project.

**Maven:** Facilitated dependency management for the project.

**Selenium:** Utilized for web scraping data from Amazon, Ebuyer, LaptopsDirect, Box, and StockMustGo.

**Hibernate:** Employed to store scraped laptop data efficiently into a MySQL database.

**Node.js:** Leveraged with RESTful APIs for retrieving laptop data stored in the MySQL database.

**Ajax:** Integrated the backend with the frontend website for seamless data communication.

**Vue.js:** Implemented most of the frontend functionalities, providing an interactive user interface.

The scraping process involved extracting data from five different website utilizing threads to retrieve data asynchronously. By doing this I was able to accumulate over 500 laptops. The collected data was structured and stored efficiently in a MySQL database using Hibernate. The database architecture employed normalization and was structured to store information on laptop details, Laptop variations and Comparison details. Node.js was employed to create a backend server that communicates with the MySQL database through RESTful APIs. This helped to retrieve data seamlessly for the frontend, where I implemented Vue.js.

# **Screen shots**

Some screen shots of the website key functionality.

**Home Page –** This is the first Page that is shown when user access the webpage. User can search for products or pick from the brand that want to check as provided below the search

A search box on a computer

Description automatically generated

**Search Result –** Below is the screen shot of the product listing page based on the user search. Here user can pick a product they would like to compare the price of from the different websites.

A screenshot of a computer

Description automatically generated

**Comparison page:** Below is a screenshot of the comparison page when MacBook air 2023 was entered.

A computer with a blue design on it

Description automatically generated

A screenshot of a computer

Description automatically generated

# **Database Design Diagram**

Below is the final database design. There is a structured organization of the tables to effectively manage and store information pertaining to laptops, their variations and comparison as well. The laptop variations table correlates with the laptop table through a foreign key relationship and then the comparison table links to the laptop variations.

A screenshot of a computer

Description automatically generated

# 

# **Testing Documentation**

Here are the pictures showcasing the results of the Junit tests, the outcomes of the Mocha and Chai tests and the results obtained from the Postman API assessments.

**JUNIT TEST**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**MOCHA AND CHAI**

A screenshot of a computer program

Description automatically generated

**POSTMAN API**

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated