Development of E-Commerce Website using ASP.net Core and Model-View-Controller Architecture

Raghavi R[1], Padmaja K.V[2]

[1]Student, Dept. of Electronics and Instrumentation Engineering, R.V. College of Engineering, Karnataka, India
[2]Professor and Associate Dean, Dept. of Electronics and Instrumentation Engineering, R.V. College of Engineering, Karnataka, India

---***---

Abstract - *The world of Information Technology is very vast and it's advancing everyday with the applications it has to offer. With this growth, today there are numerous software applications available for various requirements. Many such software applications aim to make our lives easier by improving factors like speed, quality, accuracy involved in the process. Especially, in the fast running world of business, these time saving applications are important. Here, development of one such e-commerce web application using ASP.NET Core MVC (Model-View-Controller) is discussed. Microsoft's ASP.NET with its every update brings to developers a chance to deploy highly scalable and high-performance web applications. Along with ASP.NET Core to implement the project, tools HTML, CSS, Bootstrap for frontend and Microsoft SQL server 2019 for database are discussed. The scope of the project is to provide a common platform to purchase products for customers while providing better marketing for the products developed by an organization.*

Key Words: **E-commerce website, web development, ASP.NET Core, MVC (Model-View-Controller) Architecture, Microsoft SQL Server.**

1.INTRODUCTION

The world of Information Technology is growing rapidly bringing along new applications with it. Software products come as a useful administration in the world of business to many bringing with them more advantages than traditional services they tend to replace. They help to improve factors like speed, quality and accuracy involved in a process. This project is about developing one such advantageous web application in the domain of e-commerce, making product recruitment easy for customers or clients to a company. As suggested by D. S. Bhat et al., (2016) [1], important factors for an e-commerce site like good user interface are implemented. After considering different tools and framework to build e-commerce websites, Microsoft's ASP.NET was chosen. It gives a chance to deploy highly secure applications as stated by M. Medenica and D. Dihovicni (2017) in [2] and scalable and high performance web applications along with MVC(model-view-controller) architecture.

This project, Software Hub is a web-based platform which the customers can visit on the internet using the given URL and browse through the various software products developed by an organization. With authorized login credentials they can purchase the required products according to their need. Based on the customer's selection of the products an invoice will be generated in the billing section. The total cost of the purchase will be displayed, the customer can then pay using the integrated payment gateway method with verified credentials for purchasing the products.

The aim of this project is to develop a web-based platform that provides a common place for a customer to pur-chase the software products and services according to their requirement. The aim is also to provide better marketing for the products developed by an organization and bring ease of product purchase for customers by optimizing the manual work involved and time consumed in the process of purchasing products.

1.1. Overview of the Project

This web application is built on HTML, CSS, and JavaScript for the front-end, and ASP.net Core MVC framework, it uses Microsoft SQL Server for storing and handling the data. There are two login options in the project, that is the user login and the admin login. The user will register in the portal by giving the required and proper details, once the credentials are stored successfully in the database the user can login to the application by providing valid credentials and browse through the project site. Another login is the admin login which only administrators can access and the credentials are predefined and stored in the database. Using this, admin can add new products, edit existing ones and delete any if required as per company's produce. The project is divided into four modules through which the complete software procurement process can be done, those modules are: The Home module, the Product module, the Cart module and the Admin module.

Home Module

The Home module displays the detailed list of all the products available by reading from the database including any changes that are added by the admin, the user loads an https URL (Uniform Resource Locator) and lands on to home page and from there the user can either see the product list and click on any product to go to the product module or the user can either navigate to different modules or search any products in the same page.

Product Module

Redirected from the home page, the product module displays the complete details of any selected product to the user. These details are fetched from the database and can be updated by the admin anytime if required. The user can check all the features and price of the product and add the product to the cart if it matches the requirement, else the user can navigate back to the home page and check some other products instead.

Cart Module

The Cart module shows the list of products added to the cart by the user. Users can add any number of products to cart and can see the individual product details and its price. Users can also remove the products if it is not required and add some other products. Finally, the user can make the payment for all the products available in the cart and get a detailed generated bill to his email.

Admin Module

In the Admin module, the admin can add, update and remove any product available for marketing on the portal. Once the admin adds any product, it will be stored in the database and the same details will be displayed in the home and product module to the user. Hence whenever the admin is adding any product or modifying any product then it should be done with the utmost carefulness.

2. METHODOLOGY

The flow of the project begins from an individual giving the URL of the software site on the internet, signing himself up as a customer to ending the purchase with a payment gateway. Other modules are interfaced in between. Like seen in figure 1, the four main modules are: Home, Product, Cart and Admin modules.

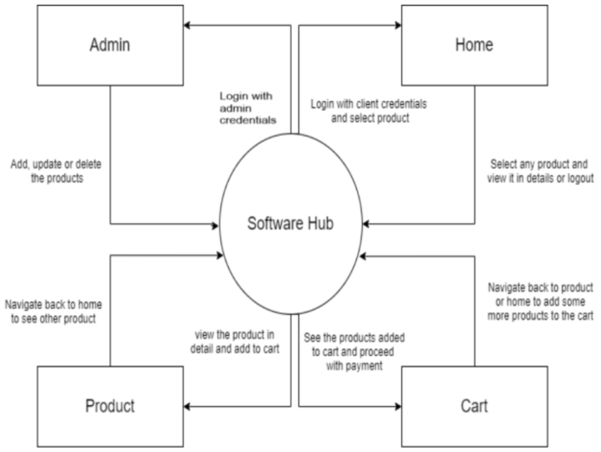

Admin
Home
Login with admin credentials
Login with client credentials and select product
Add, update or delete the products
Select any product and view it in details or logout
Software Hub
Navigate back to home to see other product
Navigate back to product or home to add some more products to the cart
view the product in detail and add to cart
See the products added to cart and proceed with payment
Product
Cart

Fig -1: Context Diagram

2.1. Design concepts

The idea behind design of this project is to implement effective interface of front-end and back-end development. MVC Architecture was used to design the data model and render a view to the same in this project.

2.1.1. Front-end and Back-end Development

Front-end development, is client-side web development, it involves the process of creating HTML, CSS, and JavaScript for a website so that a user can see and interact with the application directly [3]. The idea of this project site's design is to implement a good User Interface (UI), that is, make sure when users visit the site, they view the material in an easy-to-read and relevant format, hence front-end development was essential.

The development of server-side logic that powers websites and apps from behind the scenes is referred to as back-end development [4]. In this project, it was critical to facilitate API integration and the logic to build the database and interface it with the application. C#, a language rich in object oriented approach was used for building this application.

2.1.2. Database (DB) Management

A database contains organised data managed by a Database Management System (DBMS). Different tables like User Table, Product Table, Cart Table are a part of this project's database. These individual tables contain all information concerned to them, in the form of rows and columns. Data from the database is accessed in the project using back-end logic. Microsoft SQL Server was the database management system used in this project, SQL Server Management Studio (SSMS) was used to write queries to the database.

2.1.3. Model-View-Controller

As Krasner et al., (1998) described in [5], MVC is a design paradigm in which the user interface (view), data (model), and application functionality (controller) are all separated (Model View Controller). This pattern can help you separate your worries of handling data and rendering a view to it. Requests are passed to a Controller, which works with the Model using the MVC pattern for websites to perform actions and acquire data. The Controller decides which View to show, and the Model receives it. The View creates the final page using the Model's data. This project implements this architecture to achieve rapid development of the application.

2.2. Methodology

Number of different algorithms are implemented at all stages of the project to ensure correct and effective functionality. Starting from the addition of a user to the database of the server to the completion of payment by user, database operations play a very important role. The application is carried out using ASP.NET Core MVC Architecture with help of different Nuget packages to aid different functionalities.

Upon entering the specific URL to the page on the internet, the customer lands on the Home page first. Then the flow of the project can take two ways such that the customer can login and purchase products or login just before the billing process. The project flow is described in the below section in different steps.

2.2.1. Project Flow

1. Sign-Up Page

- Registration option is provided for new users who are required to give their credentials and sign up.
- All user credentials are verified and stored in the database.

2. Home Module

- Users can search for the software products of their needs using search functionality.
- From the home page, users can navigate to Sign up, Login, Product pages.
- Algorithm:

```
PROCEDURE HOME
Display all the products from the product table
DO FOR each product
    print (product list with view details feature)
ENDDO
```

3. Product Module

- Detailed description of every product is provided for user's convenience.
- Users can add required products with the add to cart option on the product page.
- Algorithm:

```
PROCEDURE PRODUCT
DETAILS
Display the details of the selected product
IF #productId is valid THEN
    print (product details)
ELSE
    print (error message)
ENDIF
```

4. Payment

- Based on the user's purchase of software products, that is, products they've added on the cart, a bill is generated.
- Paypal payment gateway is integrated to facilitate the payment process, through which users can pay for the purchase.

5. Admin Module

- A login option is provided for admins, using which they can update products available in the market.
- Different operations like deleting products, adding new products, editing products can be done by an admin. This helps to keep the users up to date with products in the market.
- Algorithm:

```
PROCEDURE ADMIN
HOME
Add new product to product table and display it with edit and delete feature
DO FOR each new product
    IF data is valid THEN
        Add product to product table
        print (product with edit and delete feature)
    ELSE
        print (error message)
ENDDO
```

2.2.2. Database Operations

Accessing data from databases is an essential part of building web applications, Y. Bai (2020a) in [6] speaks about accessing data from databases. Entity Framework Core is a cross platform version of EF (Entity Framework), it is lightweight, extensible, and open-source. It provides Object-relational mapping (O/RM) feature, which allows developers to use .net objects while working with databases, that is, it maps database table entries to an object. Most of the data-access code that is generally required is avoided. EF Core supports a wide range of database engines.

In this project, Entity Framework is used to map database with objects to make Create-Read-Update-Delete (CRUD) operations on them. It's used in the sign up page to create a new user entry in user table in the database. In home page, it's used to read and display all products from the database. It's used in admin pages, to edit existing product entries or add new products or delete products from product table in database.

3. IMPLEMENTATION

This platform is built on Asp.net 3.0 Core. Microsoft's Asp.net framework delivers a programming model for the development of console or web applications. A web application can be developed from front-end to back-end in many different ways and today, with the improvement in technology, there are alot of tools which are available to build efficient and reliable websites.

3.1. Software Requirements

Hosting

- This project is hosted using IIS Server

Database

- This project uses Microsoft SQL server 2019, for Relational Database Management System (RDBMS)

Front-end languages and framework

The following are used in front-end development of this website:

- HTML
- CSS
- Bootstrap
- JavaScript

Back-end languages and framework

The following are used in back-end development of this website:

- C#
- Model-View-Controller (MVC)

Integrated Development Environment (IDE)

The following code editors or IDEs were used in building the program for the project:

- Visual Studio 2019
- SQL Server Management Studio (SSMS)

3.2. Development Platform

The platform used was Microsoft's Asp.net Core framework which is an open source subset of .net framework. A widely used web application framework, .net, provides a programming model for development of console or web applications. It allows developers to build websites using programming languages like C# or VB.net which has an object oriented approach. On the server side, web applications built using the .NET framework or its subsets are required to run on Microsoft's Internet Information Services (IIS).

In this project, the language opted to develop was C# due to it's rich OOP (Object Oriented Programming) approach. Internet Information Services (IIS) is used to serve HTML pages. The advantages of Asp.net Core which made it a preferable tool for this project are discussed below:

- It is Open source.
- Easy to build cross platform applications on Windows, Linux, Mac.
- Web API (Application Programming Interface), MVC and Asp.Net Web pages are all supported.

3.3. Design Pattern

The design pattern used in this project was Model-View-Controller (MVC). MVC is a design paradigm in which the user interface, data, and application functionality are all separated as view, model and controller respectively. Requests are passed to a Controller, which works with the Model using the MVC pattern for websites to perform actions and acquire data. The Controller decides which View to show, and the Model receives it. The View creates the final page using the Model's data. This project implements this architecture to achieve rapid development of the application. The figure 2 represents MVC Architecture.

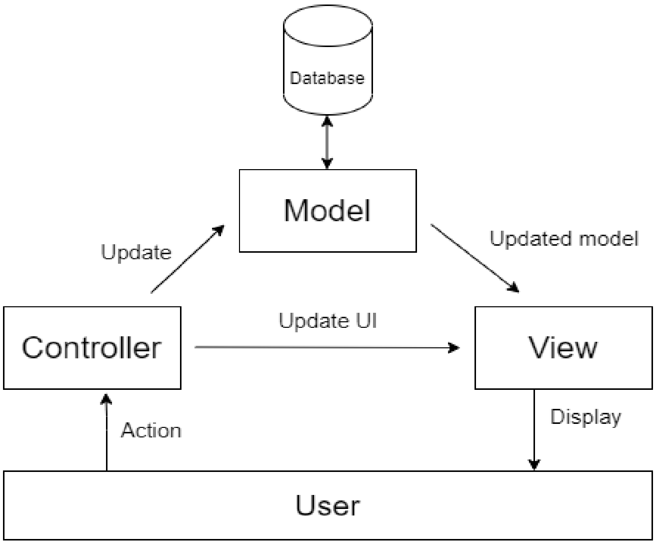
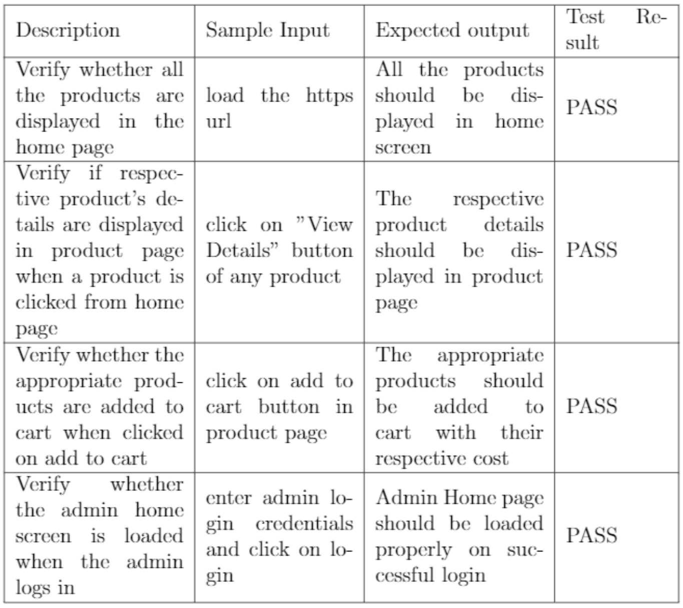


Fig -2: MVC Architecture

4. RESULTS AND DISCUSSIONS

Two kinds of testing were essential to this project: Unit Testing and Integration Testing. A single part of the application's logic is tested in Unit Testing. A unit test doesn't test how the code works with dependencies or infrastructure, a unit test doesn't test the framework on which the program is written. This is where integration testing helps in verifying the application. Integration testing was used to verify the functionalities provided by external dependencies and to verify connection between different pages in a site. The results after integration testing are tabulated and presented below in figure 3.



Description	Sample Input	Expected output	Test Result
Verify whether all the products are displayed in the home page	load the https url	All the products should be displayed in home screen	PASS
Verify if respective product's details are displayed in product page when a product is clicked from home page	click on ”View Details” button of any product	The respective product details should be displayed in product page	PASS
Verify whether the appropriate products are added to cart when clicked on add to cart	click on add to cart button in product page	The appropriate products should be added to cart with their respective cost	PASS
Verify whether the admin home screen is loaded when the admin logs in	enter admin login credentials and click on login	Admin Home page should be loaded properly on successful login	PASS

Fig -3: Test Results

5. CONCLUSIONS

The process of product procurement is present across many businesses. But, it was noted that the process of software procurement is not an easy task. There are a lot of manual tasks involved currently and they are very time consuming. Hence, the concept of Software Hub was brought into picture where the clients can procure any software according to their requirements from the web application with ease. All the detailed description about the product is given and clients can just go through that and order any product according to the requirement. This eases the process of software procurement for the client and saves a lot of time. It also improves marketing for products on this marketplace.

Throughout this project the complete software procurement process is made online and very minimal manual user intervention is required. After the user orders the required product and completes the payment process, additional enhancements like installing it in the user's environment by the assigned support staff can be made. This concludes the process of e-software procurement or online software procurement.

6. FUTURE SCOPE

The Future Enhancement to the project would be adding technical assistance online for the users. Any problems or issues faced by the user with the software products which have been purchased and deployed in their environment can be reported by them online and within a few minutes one of the technical assistant/support staff will be contacting the respective user regarding that issue.Currently, the project is created for a single locale, and not exposed worldwide. In the future, a measure can be taken to make this project available worldwide and even the transactions can be handled for different countries and for different currencies.

REFERENCES

[1] D. S. Bhat et al., (2016) "A review paper on e-commerce"

[2] M. Medenica and D. Dihovicni (2017) "Security point of view of asp.net application," in 2017 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), pp. 403–406.

[3] Wikipedia. [online] https://en.wikipedia.org/wiki/Front-end_web_development

[4] Course Report. [online] https://www.coursereport.com/

[5] Krasner et al., (1998) A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk80 System. Journal of Object-oriented Programming - JOOP.1

[6] Y. Bai (2020a) "Accessing data in asp.net," in SQL Server Database Programming with Visual Basic.NET: Concepts, Designs and Implementations. pp. 429–504.

[7] S. E. Ullah et al., (2016) "Developing an e-commerce website," in 2016 International Conference on Microelectronics, Computing and Communications (MicroCom), pp. 1–4.

[8] M. del Pilar Salas-Z´arate et al., (2015) "Analyzing best practices on web development frameworks: The lift approach," Science of Computer Programming, vol. 102, pp. 1–19.

[9] M. Jailia et al., (2016) "Behavior of mvc (model view controller) based web application developed in php and. net framework," in 2016 International Conference on ICT in Business Industry & Government (ICTBIG), IEEE, pp. 1–5.

[10] Y. Bai (2020b) "Asp.net web services," in SQL Server Database Programming with Visual Basic.NET: Concepts, Designs and Implementations. pp. 505–636.