International Journal of Computer Engineering in Research Trends
Multidisciplinary, Open Access, Peer-Reviewed and fully refereed
Research Paper Volume-8, Issue-11, 2021 Regular Edition E-ISSN: 2349-7084

A Web-Mart Design Using Angularjs

[1]B.Shreyank , [2]T. Varun , [3] Peddi.P.M. Laasya Rao

[1, 2, 3] IV Year,CSE Dept,CVR College of Engineering, Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), Rangareddy (D), Telangana, India

Email ID: [1]shreyankbss@gmail.com , [2]varunsekhar208@gmail.com , [3]pranava.mihi@gmail.com

**Corresponding Author: shreyankbss@gmail.com*

Available online at: http://www.ijcert.org

Received: 08/11/2021, Revised: 12/11/2021, Accepted: 16/11/2021, Published: 18/11/2021

Abstract: - Most consumers are looking online for information that will help them make smarter purchasing decisions. In fact, according to the eCommerce Foundation, 88% of consumers will research product information before they make a purchase online or in the store. This buying behaviour trend emphasizes the importance of a website for today's businesses. This paper describes the implementation of a web-mart design using AngularJS. The cart uses PayPal and Google e-Wallet payment services. Adding other providers is fairly easy. This designed product supports online payment infrastructure like integrating all the credit or debit card payment and also supports the e wallet and other UPI because offering more payment options should increase sales.

Keywords: eCommerce, web-mart, AngularJS, MVC frame work.

---

1. Introduction

A shopping cart is basically a list that contains products selected by the user while he shops. When the user is finished shopping, he will usually examine the list to double-check that the items, quantities, and prices are correct. If he finds any errors, he should be able to edit the list. Once he is ready, he should be able to check out. The checkout process involves an exchange of information that results in a sale.

Sounds simple, right? And it actually is. The only challenge is performing the checkout, because that involves personal information and money. Fortunately, there are services that handle this type of transaction and you can leverage them. Some of the most popular are PayPal and Google Wallet.

This paper describes the implementation of a web cart using JavaScript. The cart uses PayPal and Google Wallet payment services. Adding other providers is fairly easy. If you have your own payment infrastructure for example, you can extend the shopping cart to use that in addition to the PayPal and Google Wallet options. Offering more payment options should increase sales. The article includes a sample application called "Angular Store" that demonstrates how to use the shopping cart in AngularJS applications.

AngularJS, built by Google, is an open-source web application framework that is designed to make both front end development and testing easier for web developers. The main goal of AngularJS is to elongate web applications with MVC (Model–view–controller) capability. It is a client-side JavaScript MVC/MVVM framework that is fully extensible, runs with no library dependencies, and also works great with other libraries. Even, you can also modify or replace its every feature to fit your specific needs.

AngularJS was developed in 2009 by Adam Abrons and Miško Heverym, who were Google engineers at that time. As mentioned on official website, it is a "structural framework for dynamic web apps" which is best fitted for creating one-page web applications that only require JavaScript, CSS, and HTML on the client-side.

2. Related Work

By Google, is an open-source web application framework that is designed to make both front end development and testing easier for web developers. The main goal of AngularJS is to elongate web applications that only require JavaScript, CSS, and HTML on the client-side with MVC (Model–view–controller) capability. It is a client-side JavaScript MVC/MVVM framework that is fully extensible, runs with no library dependencies, and also works great with other libraries. Even, you can also modify or replace its every feature to fit your specific needs.

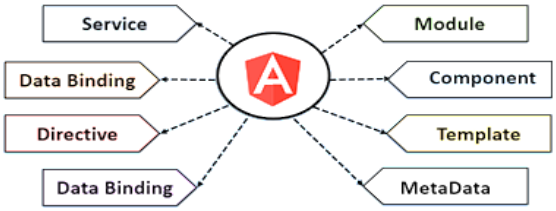

Figure 1. Angular architecture

There are basically eight main things involved in angular architecture:

- Module
- Component
- Template
- Metadata
- Service
- Data Binding
- Directive
- Dependency Injection

2.1 Module

It is basically a block or piece of code that is responsible to perform a single task, it is an independent task. Angular applications can have any number of the module in it. We can also export these modules in the form of a class. The angular application should have at least one module. A module class must be decorated with the following annotations i.e. @NgModule it takes a metadata object.

It has many properties which describe below.

- *Declarations*: It is used to declare the view class that belongs to the current module. There are three types of classes provided by my angular i.e. component, pipes, and directive. In this, we define our custom component.
- *Imports:* This is used to import other modules to use in the existing module. That can be anything.
- *Providers:* These are used to create service and they can be associable to any part of the application.
- *Bootstrap:* The root module has to set this property which is going to host all other modules.

2.2 Component

The component is a class that contains the core of business logic for the application.

2.3 Template

template is the HTML that we want to display. It is a simple HTML page that will show the data to the user.

2.4 Metadata

In angular, we can define metadata by using a decorator. For example, if we make any component in angular but how will angular identity it as a component? We need to tell it by using the annotation @Component.

2.5 Data Binding

Data binding is the main block of angular architecture. It allows us to have communication between a component and a template which is very much necessary to render our business logic to the user in the form of data. There are four types of data binding provided by angular 2.

Interpolation: It is used to display a component property to a template or view. For this, we use double curly braces. We can display anything using this i.e. str

Property Binding: Property binding allows us to directly access the element property of HTML. We can directly assign our variable to an HTML element. number, array, etc.

Event Binding: This binding uses the event to bind value. We have various events available for instance click.

Two-Way Binding: This is basically a combination of property binding and event binding. In which we can simultaneously call our method and bind value to the element property.

2.6 Service

It is an injectable class which is used to share data among various class or application. It is also responsible to make the server call and get data to display. Any class annotated with @Injectable and making server calls can be considered as service.

2.7 Directive

They provide a special behavior to the DOM element. They are extended HTML attributes.

Decorator Directive: There are many built-in directives like ngModel, ngFor, ngIF, etc. many build-in directives provided by angular.

Component Directive: It is extended of @Directive decorates with template-oriented features.

Template Directive: It is also known as Structural directive. It converts HTML into a reusable template

2.8 Metadata

In angular, we can define metadata by using a decorator. For example, if we make any component in angular but how will angular identity it as a component? We need to tell it by using the annotation @Component.

3. System Study

3.1 Existing System

By understanding and searching the web sites related to web-mart, we have seen there are many web-mart such as Amazon, Flipkart etc. In the proposed system customers need not go to the shops for purchasing the products. He/she can order the product he/she wishes to buy using this system. The shop owner can be the admin of the system. The shop owner can appoint officials particularly to handle this, who will help the owner in managing the customers and product orders. The system also endorses a home delivery system for delivering the purchased products.

3.2 Limitations of Existing System: -

- Frauds in online shopping
- Delay in the delivery
- We can't touch the product
- We cannot bargain
- Returning the product.

3.3 Proposed System:

This project aims to develop an online shopping for customers with the goal so that it is very easy to shop your loved things from our web-mart sites available on the web. With the help of this you can carry out an online shopping from your home. Here is no compelling reason to go to the crowed stores or shopping centers during pandemic year. You simply require a PC or a laptop or a mobile and one important payment sending option to shop online. Upon successful login the customers can purchase a wide range of things such as mobiles, books, apparel, jewelry, infant care, gifts, tools, etc. can be dispatched using online shopping system and of course you will get your requested ordered items at your doorstep.

3.4 Objectives

The objectives of the Project are

- Promoting a service or product online
- Selling a service or product
- Providing product support or customer service.

3.5 System Architecture

The system architecture of the proposed work represents number of components that are using as part of our project and the flow of request processing i.e. what components in procession the request and in which order. An architecture description is a formal description and representation of the system, organized in a way that supports reasoning about the structure of the system.

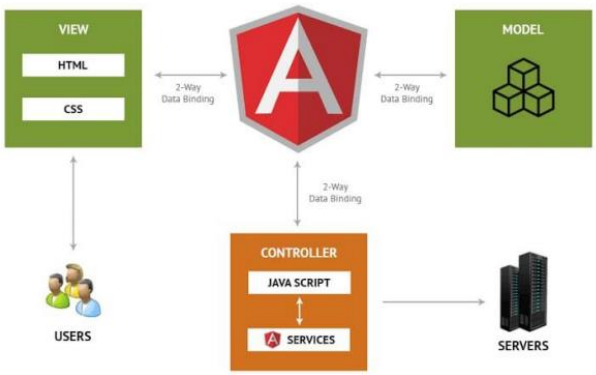

Figure 2. System Architecture

The MVC pattern first took hold in the server-side end of web development, through toolkits like Ruby on Rails and the ASP.NET MVC Framework. In recent years, the MVC pattern has been seen as a way to manage the growing richness and complexity of client-side web development as well, and it is in this environment that Angular has emerged. The three main building blocks are the model, the controller, and the view. In the following figure 3., you can see the traditional exposition of the MVC pattern as it applies to server-side development.

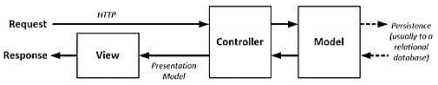

Figure 3. MVC pattern at Server Side development

You can see that the expectation is that the model is obtained from a database and that the goal of the application

is to service HTTP requests from the browser. This is the basis for round-trip web apps Of course, Angular exists in the browser, which leads to a twist on the MVC theme, as illustrated in the following figure 4..

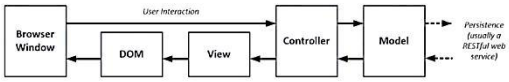

Figure 4. HTTP requests from the browser

The client-side implementation of the MVC pattern gets its data from server-side components, usually via a RESTful web service. The goal of the controller and the view is to operate on the data in the model to perform DOM manipulation so as to create and manage HTML elements that the user can interact with. Those interactions are fed back to the controller, closing the loop to form an interactive application.

Angular uses slightly different terminology for its building blocks, which means that the MVC model implemented using Angular looks more like in the figure 5.

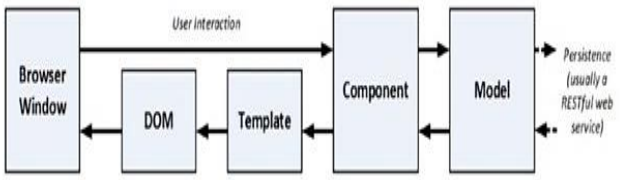

Figure 5. Basic mapping of Angular building blocks to the MVC pattern

The figure 5. shows the basic mapping of Angular building blocks to the MVC pattern. To support the MVC pattern, Angular provides a broad set of additional features, which I describe throughout the website.

Understanding Models

***Models*—**the M in MVC—contain the data that users work with. There are two broad types of model: view models, which represent just data passed from the component to the template, and domain models, which contain the data in a business domain, along with the operations, transformations, and rules for creating, storing, and manipulating that data, collectively referred to as the model logic.

Many Angular models will effectively push the logic to the server-side and invoke it via a RESTful web service because there is little support for data persistence within the browser, and it is simply easier to get the data you require over Ajax.

- Contain the domain data
- Contain the logic for creating, managing, and modifying the domain data (even if that means executing remote logic via web services)
- Provide a clean API that exposes the model data and operations on it.

Understanding Controllers/Components

Controllers, which are known as components in Angular, are the connective tissue in an Angular web app, acting as conduits between the data model and views. Components add business domain logic required to present some aspects of the model and perform operations on it. A component that follows the MVC pattern should

- Contain the logic required to set up the initial state of the template
- Contain the logic/behaviors required by the template to present data from the model
- Contain the logic/behaviors required to update the model based on user interaction

Understanding View Data

The domain model isn't the only data in an Angular application. Components can create view data (also known as view model data or view models) to simplify templates and their interactions with the component.

Understanding Views/Templates

Views, which are known as templates in Angular, are defined using HTML elements that are enhanced by data bindings. It is the data bindings that make Angular so flexible, and they transform HTML elements into the foundation for dynamic web applications. I explain the different types of data bindings that Angular provides in detail in Part 2. Templates should

- Contain the logic and markup required to present data to the user.

4. Result and Analysis

The goal is to provide a user friendly platform where the customers can easily shop their loved things from our web-mart sites available on the web. With the help of this they can carry out an online shopping from their home. Upon successful login the customers can purchase a wide range of things such as mobiles, books, apparel, jewelry, infant care, gifts, tools, etc. can be dispatched using online shopping system and of course you will get your requested ordered items at your doorstep.

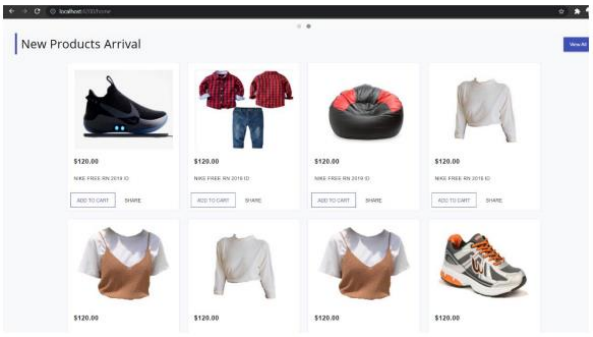

Figure 6. New Product arrival page

Figure 6. Shows the updated product in the server, which shows at client side as new product arrival page , where customers can select and purchase from the new product arrival page .

PROFILE INFO PAGE TESTING:

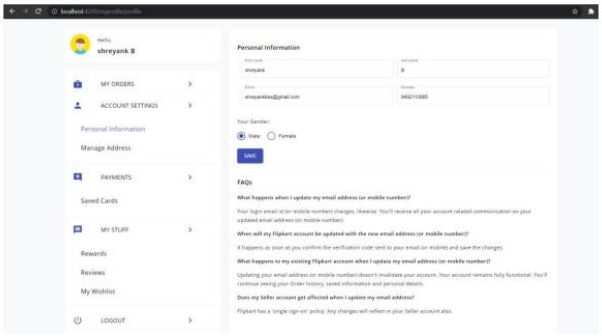

Figure 7. Profile page

This page shows the user's personal information like first name , last name , mobile no, Gender etc..

SAVE CREDIT/DEBIT CARD PAGE TESTING:

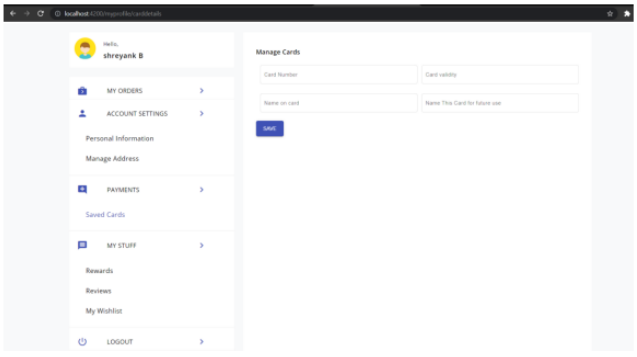

Figure 8. Credit /Debit Card Saving page

Figure 8 shows the details if credit/Debit card saving page for to manage the account sections which stores the credit/debit card details for easy payment for future reference.

PRODUCT PAGE TESTING:

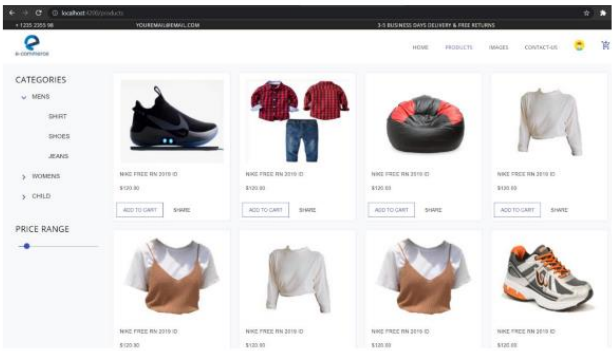

Figure 9.Product page Testing

This page shows the product filtering form the main menu where user can search and filters as per the requirements

SHOPPING CART PAGE TESTING:

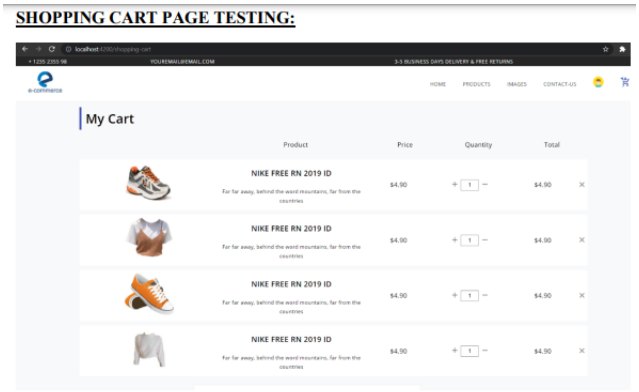

Figure 10. Shopping cart page

Figure 10 shows the shopping cart page for select the items to purchase where user can select the product with quantity depends on selected quantity , amount will be paid .

5. Conclusion

Technology has made significant progress over the years to provide consumers a better online shopping experience and will continue to do so for years to come. With the rapid growth of products and brands, people have speculated that online shopping will overtake in-store shopping. However, the availability of online shopping has produced a more educated consumer that can shop around with relative ease without having to spend a large amount of time. In exchange, online shopping has opened up doors to many small retailers that would never be in business if they

had to incur the high cost of owning a brick and mortar store. At the end, it has been a win-win situation for both consumer and sellers

Future Scope: Our designed online shopping system provides a 24×7 service that is customers can surf the website, place orders anytime they wish to. Also, the delivery system works 24×7 hours a week. Some of the features that can be modified and added to this system in the future involve its implementation by local shopkeepers, where shops will be providing an online ***interface*** to customers for shopping and placing orders. Then some delivery persons can perform their work. This will be adding on benefit for the customers as it will save their time, plus it adds on for the shopkeepers also, as people will continue to shop from local shops rather than preferring to supermarkets every time.

References

Website References: - https://angularjs.org/
https://www.tutorialspoint.com/angularjs/index.htm
https://www.guru99.com/angularjs-tutorial.html
https://www.youtube.com/playlist?list=PL7BTNHMTJY6jAH4-5X6vyD-qPqcMu2Q0u
https://www.w3schools.com/angular/

Textbooks References: - "Learn AngularJS in 1 Day" written by Krishna Rungta. "Angular: Up and Running" written by Shyam Seshadri. "AngularJS, JavaScript, and jQuery All in One" written by Brad Dayley and Brendan Dayley

