

Ecommerce Web Application

By: **RAJSHRI TOTLA**

Table of Contents

Table of Figures	2
Aim	3
Technologies Used	3
Application Info.....	3
Components.....	3
Services	4
Pipes	4
Main View Pages.....	4
Other library/technologies implemented	5
Steps to run project	5
Project GUI layout.....	6

Table of Figures

Figure 1 ProductListComponent	6
Figure 2 ProductDetailComponent	6
Figure 3 CartComponent.....	6
Figure 4 CheckoutComponent	7
Figure 5 searchproduct PIPE	8
Figure 6 InvalidPathComponent	8
Figure 7 Mobile View	9

E-Commerce Application

Aim

To make a functional web application front end satisfying minimum requirements of an e-commerce mobile first/responsive web application

Technologies Used

- AngularJS 7
- HTML/CSS
- Local Storage
- Bootstrap
- USPS Address API
- Footer Template
- Flex Box
- Mobile Responsive
- TypeScript
- AngularJS (Routing, Services)

Application Info

Components

1. *header*

Have Header's view, model and functioning of the web application , it remains same through-out the web application. Used Bootstrap NavBar and flex to implement.

2. *footer*

Have Footer's view, model and functioning of the web application.

3. *product-list*

It contains the view and functioning of retrieving all products. Integrated with ProductService.

4. *product-detail*

It contains the dedicated view and functioning for specific product. Also the view change for mobile application, also adding few more features like(back button) to mobile view.

5. *cart*

Whenever a user will choose "Add to Cart", product will be added to the cart and will be shown this component's view and functioning. Items in the cart are editable. If some item quantity becomes 0 it is automatically removed from the cart.

6. *checkout*

This component view/path is lead after user chooses to "checkout" in cart component. This shows user the final bill and take billing information like address, payment method, etc from user. USPS API is used in this component to verify the user's address.

7. *invalid-path*

If anytime application receives an unspecified route request, this component will be loaded.

Services

Services contain the classes and function to share the functioning among the components.

1. *ProductService*

It contains product list or data, product class and all the functions that can be performed on the products like getting the products.

2. *CartService*

It contains all the functioning related to the products being added to the cart. It manages local storage and provides function to handle the data related to cart products.

Pipes

- *Search Product*

This pipe/filter will return all the products which have some same characters being searched. This filter is being used in "Search" element in header of application where user can query for products. It will give a dropdown menu to user which will take user to the specific product page.

e.g. Search: "pe" will return products contain "pe" like pen, pencil, etc.

Main View Pages

- *Product Page*

Page showing all products

- *Product Detail Page*

Page showing specific product

- *Cart*

Cart where all the products are added and allow user to edit products

- *Checkout*

Show final cart (uneditable) and take billing information like address and payment methods from user

- **Invalid Page**

Any unknown route/path in application will lead to this page

Other library/technologies implemented

- angular-alert-module

Documentation: <https://www.npmjs.com/package/angular-alert-module>

- Routings

- { path:'', redirectTo:'/products', pathMatch:'full' },
- { path:'products', component:ProductListComponent },
- { path:'product/:pid', component:ProductDetailComponent },
- { path:'cart', component:CartComponent, },
- { path:'cart/checkout', component:CheckoutComponent },
- { path:'cart/:pid', component:CartComponent, },
- { path:'checkout', component:CheckoutComponent },
- { path: '**', component: InvalidPathComponent }

Steps to run project

- i. Open project terminal
- ii. [Install node modules Run]
 - npm install
 - npm install typescript@3.1.1 **OR** npm install typescript
- iii. [Run project]
 - ng serve --open
{could specify port like ng serve --port 8080}
- iv. README Document in the project could be referenced for commands if application created from scratch
- v. Images could be added to assets folder
- vi. To run USPS address API generate USER KEY from following link.
<https://www.usps.com/business/web-tools-apis/welcome.htm>
 User key will be placed in function finalPay() defined in checkout component ts file.

Project GUI layout

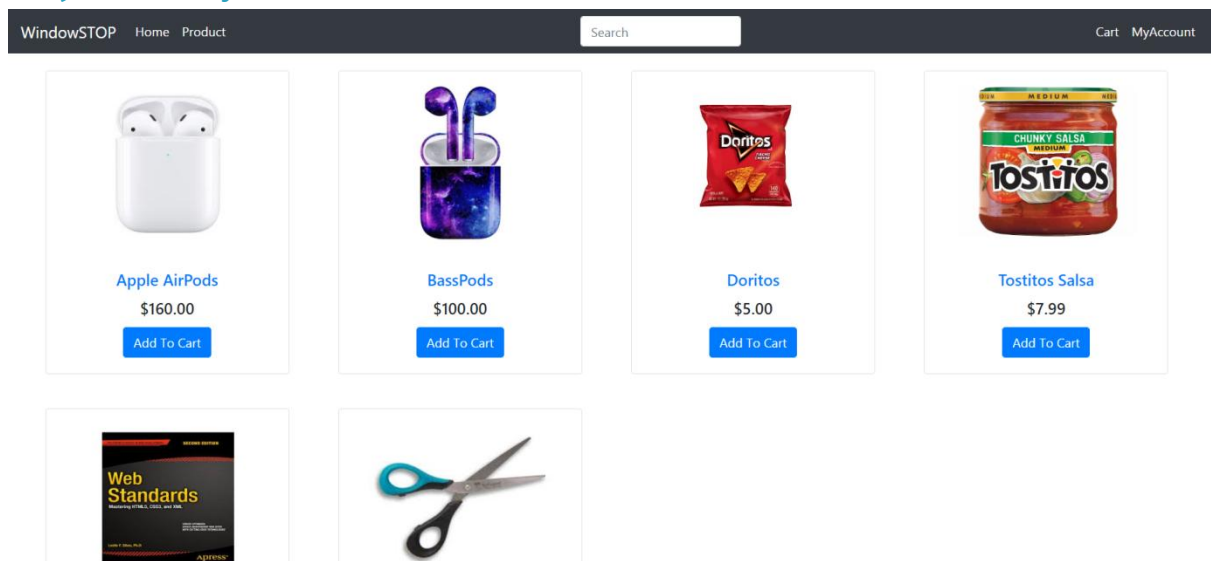


Figure 1 ProductListComponent

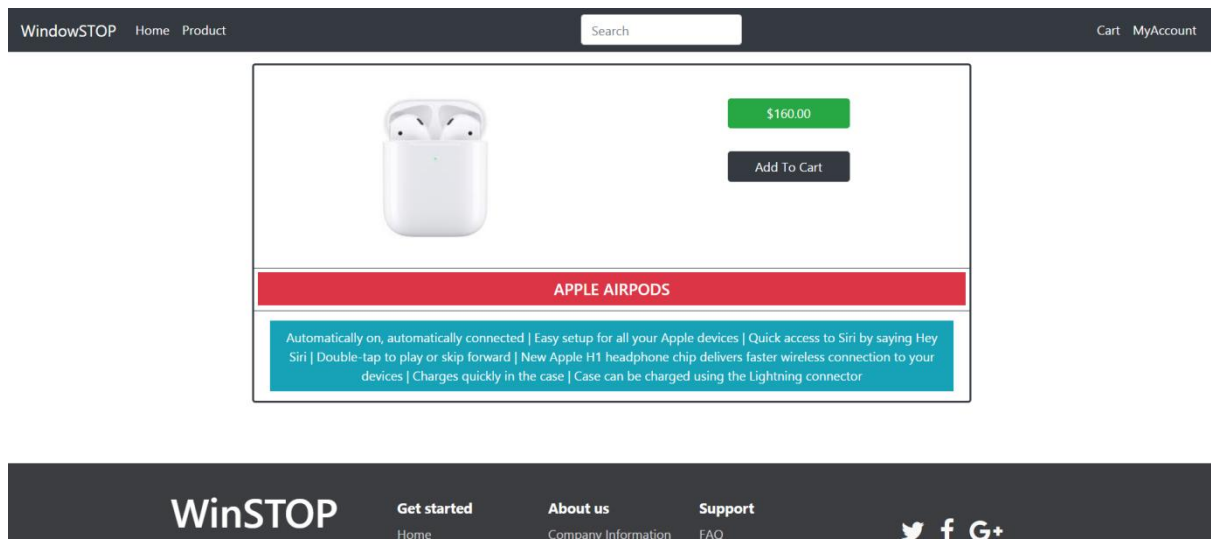


Figure 2 ProductDetailComponent

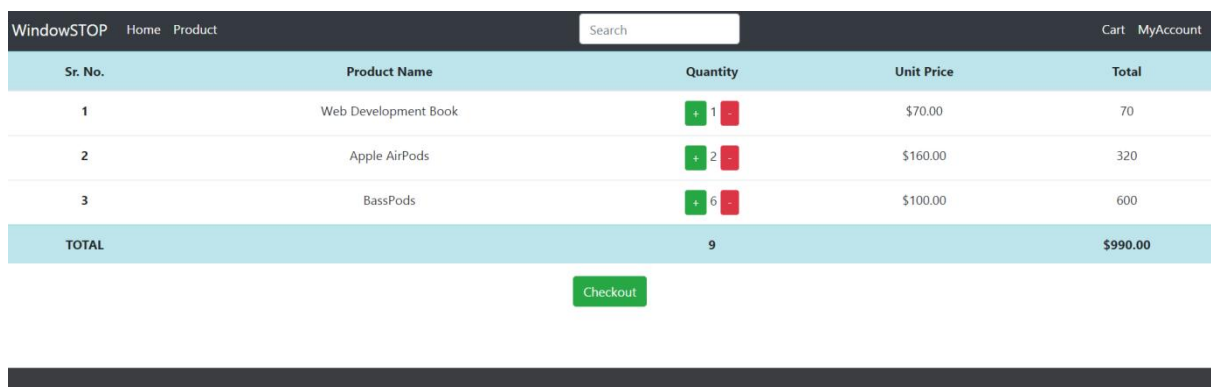


Figure 3 CartComponent

Sr. No.	Product Name	Quantity	Unit Price	Total
1	Web Development Book	1	\$70.00	70
2	Apple AirPods	2	\$160.00	320
3	BassPods	6	\$100.00	600
TOTAL		9		\$990.00

Billing Address

First Name

Last Name

Username

Email(Optional)

Address

Address 2(Optional)

City

State

Zip

☐ Shipping address is same as my billing address

☐ Save this information for next time

Payment

☒ Credit Card

☐ Debit Card

☐ Cash On Delivery

Name on card

Credit card number

Full name as displayed on the card

Expiration

CVV

Submit

Figure 4 CheckoutComponent

tos

Doritos

Tostitos Salsa



Figure 5 searchproduct PIPE

WindowSTOP

Home

Product

Search

Cart

MyAccount

Yikkessssssssss! Wrong link. :(

Figure 6 InvalidPathComponent

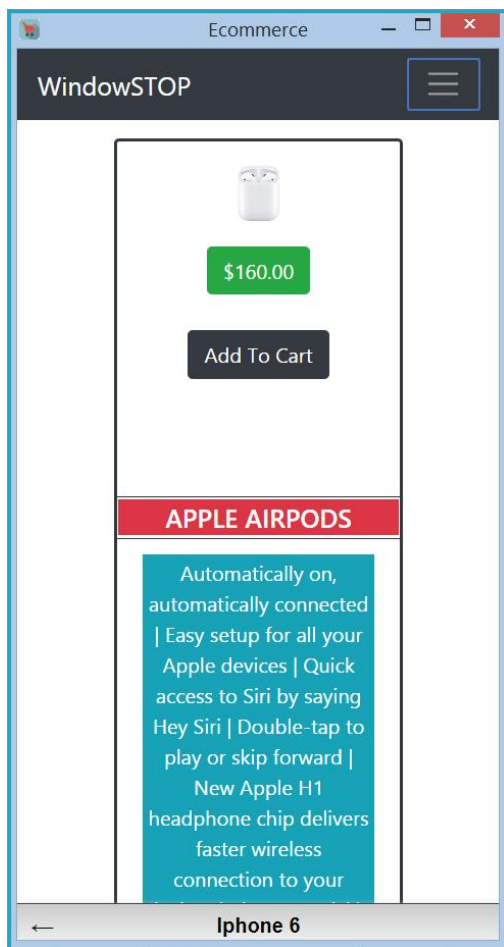


Figure 7 Mobile View