

TRƯỜNG ĐẠI HỌC CỬU LONG
KHOA CÔNG NGHỆ THÔNG TIN – TRUYỀN THÔNG



BÁO CÁO HỌC PHẦN
CÁC CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI
ĐỀ TÀI

LẬP TRÌNH WEB VỚI ANGULAR

GVHD : ThS. PHAN HỒ DUY PHƯƠNG

SVTH : Nguyễn Trung Chánh Trực

Hồ Quốc Bảo

LỚP: CÔNG NGHỆ THÔNG TIN – K18



1.1 Giới thiệu đề tài

Angular là một JavaScript framework dùng để viết giao diện web (Front-end), được phát triển bởi Google. Hiện nay, **Angular** đang được sử dụng bởi rất nhiều các công ty lớn: Forbes, General Motors, Upwork... Muốn trở thành Front-end Developer giỏi, bạn không thể không biết đến sự tồn tại của **Angular**.

1.2 Mục tiêu đề tài

Angular giúp nâng cao năng suất của các lập trình viên.

Việc phát triển Web đã có bước thay đổi đáng kể trong vài năm qua. Với phiên bản ECMAScript (ES) 2015 – chúng ta quen thuộc với cái tên ES6, với những class hay arrow function. Angular 2+ ứng dụng những tính năng mới này giúp việc code với Angular trở nên rõ ràng và dễ học hơn rất nhiều.



1.3 Nội dung đề tài

Thiết kế 1 trang giỏ hàng , xác định hiện trạng và phạm vi của hệ thống

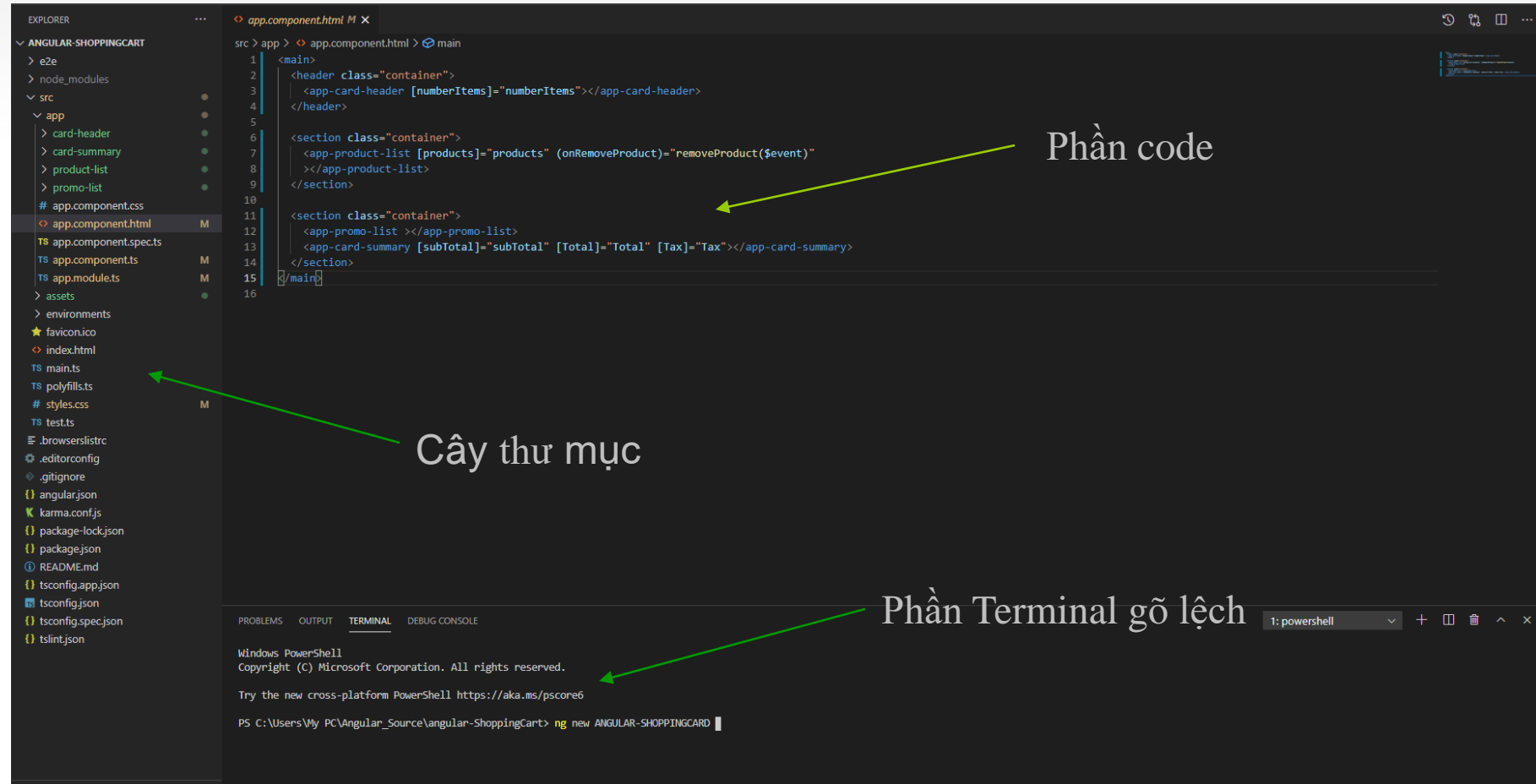
1.4 Giới hạn đề tài

Đề tài này chỉ được dùng trong giỏ hàng

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



2.1 Giới thiệu về giao diện làm việc





2.2 Các lệnh cơ bản của Angular

2.2.1 Tạo project

Cú pháp : `ng_new` 'tên project'

Ví dụ: tạo 1 project với tên ANGULAR-SHOPPINGCARD

```
ng new ANGULAR-SHOPPINGCARD
```

2.2.1 Tạo Component

Cú pháp : `ng_g_c` 'tên component'

Ví dụ: tạo 1 component với tên header

```
ng g c header
```

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



2.2 Các lệnh cơ bản của Angular

2.2.1 Tạo sever

Cú pháp : `ng_serve`

```
PS C:\Users\My PC\Angular_Source\angular-ShoppingCart> ng serve
✓ Browser application bundle generation complete.

Initial Chunk Files | Names          | Size
vendor.js           | vendor         | 2.43 MB
polyfills.js        | polyfills      | 475.03 kB
styles.css, styles.js | styles        | 351.60 kB
main.js             | main           | 28.99 kB
runtime.js          | runtime        | 6.15 kB

| Initial Total | 3.28 MB

Build at: 2021-04-22T07:37:04.666Z - Hash: 40a85847ec22a609ca4a - Time: 20037ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
```

Sao đó ta chạy project theo đường dẫn *<http://localhost:4200/>*



2.3 Thiết kế giao diện Shoppingcart

Chia nhỏ component ra làm 4 component con để dễ quản lý

```
> card-header  
> card-summary  
> product-list  
> promo-list
```

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



2.3 Thiết kế giao diện Shoppingcart

2.3.1 Thiết kế giao diện html và xử lý cơ bản card-header

```
<? card-header.component.html |
<h1>Shopping Cart</h1>

<ul class="breadcrumb">
  <li>Home</li>
  <li>Shopping Cart</li>
</ul>

<span class="count">{{numberItems}} items in the bag</span>
```

```
ts card-header.component.ts
import { Component, OnInit, Input } from '@angular/core';

@Component({
  selector: 'app-card-header',
  templateUrl: './card-header.component.html',
  styleUrls: ['./card-header.component.css'],
})
export class CardHeaderComponent implements OnInit {
  @Input() numberItems: number;

  constructor() {}

  ngOnInit(): void {}
}
```


CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



2.3 Thiết kế giao diện Shoppingcart

2.3.2 Thiết kế giao diện html và xử lý cơ bản card-summary

```
<? card-summary.component.html
<div class="summary">

  <ul>
    <li>Subtotal <span>{{subTotal}}</span></li>
    <li>Tax <span>{{Tax}}</span></li>
    <li class="total">Total <span>{{Total}}</span></li>
  </ul>
</div>

<div class="checkout">
  <button type="button">Check Out</button>
</div>
```

```
TS card-summary.component.ts
import { Component, Input, OnInit } from '@angular/core';

@Component({
  selector: 'app-card-summary',
  templateUrl: './card-summary.component.html',
  styleUrls: ['./card-summary.component.css']
})
export class CardSummaryComponent implements OnInit {
  @Input() subTotal: number;
  @Input() Total: number;
  @Input() Tax: number;

  constructor() { }

  ngOnInit(): void {
  }
}
```

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



2.3 Thiết kế giao diện Shoppingcart

2.3.3 Thiết kế giao diện html và xử lý cơ bản product-list

```
<? product-list.component.html
<ul class="products">
  <li class="row" *ngFor="let product of products">
    <div class="col left">
      <div class="thumbnail">
        <a href="#">
          <img [src]="product.thumbnail" [alt]="product.name" />
        </a>
      </div>
      <div class="detail">
        <div class="name">
          <a href="#">{{ product.name }}</a>
        </div>
        <div class="description">{{ product.description }}</div>
        <div class="price">${{ product.price }}</div>
      </div>
    </div>
    <div class="col right">
      <div class="quantity">
        <input
          #inputQuantityElement
          type="number"
          class="quantity"
          step="1"
          [value]="product.quantity"
          (input)="UpdateQuantity(inputQuantityElement)"
        />
      </div>
      <div class="remove">
        <svg...
      </div>
    </div>
  </li>
</ul>
```

```
TS product-list.component.ts
import { element } from 'protractor';
import { Component, EventEmitter, Input, OnInit, Output } from '@angular/core';

@Component({
  selector: 'app-product-list',
  templateUrl: './product-list.component.html',
  styleUrls: ['./product-list.component.css'],
})
export class ProductListComponent implements OnInit {

  @Input() products;
  // tslint:disable-next-line:no-output-on-prefix
  @Output() onRemoveProduct = new EventEmitter();
  constructor() {}

  ngOnInit(): void {}

  // tslint:disable-next-line:typedef
  removeProduct(productId: string): void {
    this.onRemoveProduct.emit(productId);
  }

  // tslint:disable-next-line:typedef
  UpdateQuantity(element) {
    console.log(element.value);
  }
}
```

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



2.3 Thiết kế giao diện Shoppingcart

2.3.4 Thiết kế giao diện html và xử lý cơ bản product-list

```
<? promo-list.component.html
<div class="promotion">
  <label for="promo-code">Have A Promo Code?</label>
  <input type="text" id="promo-code" /> <button type="button"></button>
</div>
```

```
TS promo-list.component.ts
import { Component, OnInit, Input } from '@angular/core';

@Component({
  selector: 'app-promo-list',
  templateUrl: './promo-list.component.html',
  styleUrls: ['./promo-list.component.css'],
})
export class PromoListComponent implements OnInit {
  constructor() {}
  ngOnInit(): void {}
}
```

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



2.3 Thiết kế giao diện Shoppingcart

2.3.5 Thiết kế giao diện html và xử lý cơ bản của app component

```
<> app.component.html
<main>
  <header class="container">
    <app-card-header [numberItems]="numberItems"></app-card-header>
  </header>

  <section class="container">
    <app-product-list [products]="products" (onRemoveProduct)="removeProduct($event)"
    ></app-product-list>
  </section>

  <section class="container">
    <app-promo-list ></app-promo-list>
    <app-card-summary [subTotal]="subTotal" [Total]="Total" [Tax]="Tax"></app-card-summary>
  </section>
</main>
```

```
TS app.component.ts
import { element } from 'protractor';
import { Component } from '@angular/core';
import { Product } from '../card-summary/product.model';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css'],
})
export class AppComponent {
  title = 'angular-ShoppingCart';
  > products: Product[] = [ ...
  ];

  numberItems: number = 4;
  Total: number = 3800.0;
  subTotal: number = 4000.0;
  Tax: number = 200;
  Discount: String = '';
  // tslint:disable-next-line:typedef
  > removeProduct(productId: string) { ...
  }
}
```

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



2.3 Thiết kế giao diện Shoppingcart

2.3.6 Giao diện html

Card-header




Product-list

Promo-list

Card-summary

Shopping Cart

Home > Shopping Cart 4 items in the bag

	IPHONE 12 PRO MAX , 256G Color : Đen \$950	2	×
	SAMSUNG S21 ULTRA , 256G Color : Đen \$900	1	×
	SAMSUNG S21 ULTRA , 512G Color : Đen \$1200	1	×

Have A Promo Code?

Subtotal	4000
Tax	200
Total	3800

Check Out >






2.44 Thiết kế chức năng Shoppingcart

2.4.1 Các chức năng yêu cầu

Shopping Cart

Home > Shopping Cart 4 items in the bag

	IPHONE 12 PRO MAX , 256G Color : Đen \$950	<div>2</div> <div>×</div>
	SAMSUNG S21 ULTRA , 256G Color : Đen \$900	<div>1</div> <div>×</div>
	SAMSUNG S21 ULTRA , 512G Color : Đen \$1200	<div>1</div> <div>×</div>

Have A Promo Code?

Subtotal 4000
Tax 200
Total 3800

Check Out >

Khi **xóa()** , **update()** số lượng 1 sản phẩm cần phải tính lại số lượng **item()** và tính lại **giá tiền ()**

Khi nhập đúng **mã giảm giá()** cần phải tính lại **giá tiền ()**

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



2.44 Thiết kế chức năng Shoppingcart

2.4.1.1 Các chức năng xóa

```
removeProduct(productId: string) {  
  const index = this.products.findIndex(  
    (product) => product.id === productId  
  );  
  if (index !== -1) {  
    this.products.splice(index, 1);  
  }  
  
  let Tax = 0;  
  let numberItems = 0;  
  let subTotal = 0;  
  for (const product of this.products) {  
    numberItems += product.quantity;  
    subTotal += product.price * product.quantity;  
    Tax = subTotal * 0.05;  
  }  
  this.Total = subTotal - Tax;  
  if (subTotal === 0) {  
    this.Total = subTotal;  
  }  
  this.numberItems = numberItems;  
  this.subTotal = subTotal;  
  this.Tax = Tax;  
}
```

Tìm vị trí cần xóa theo id

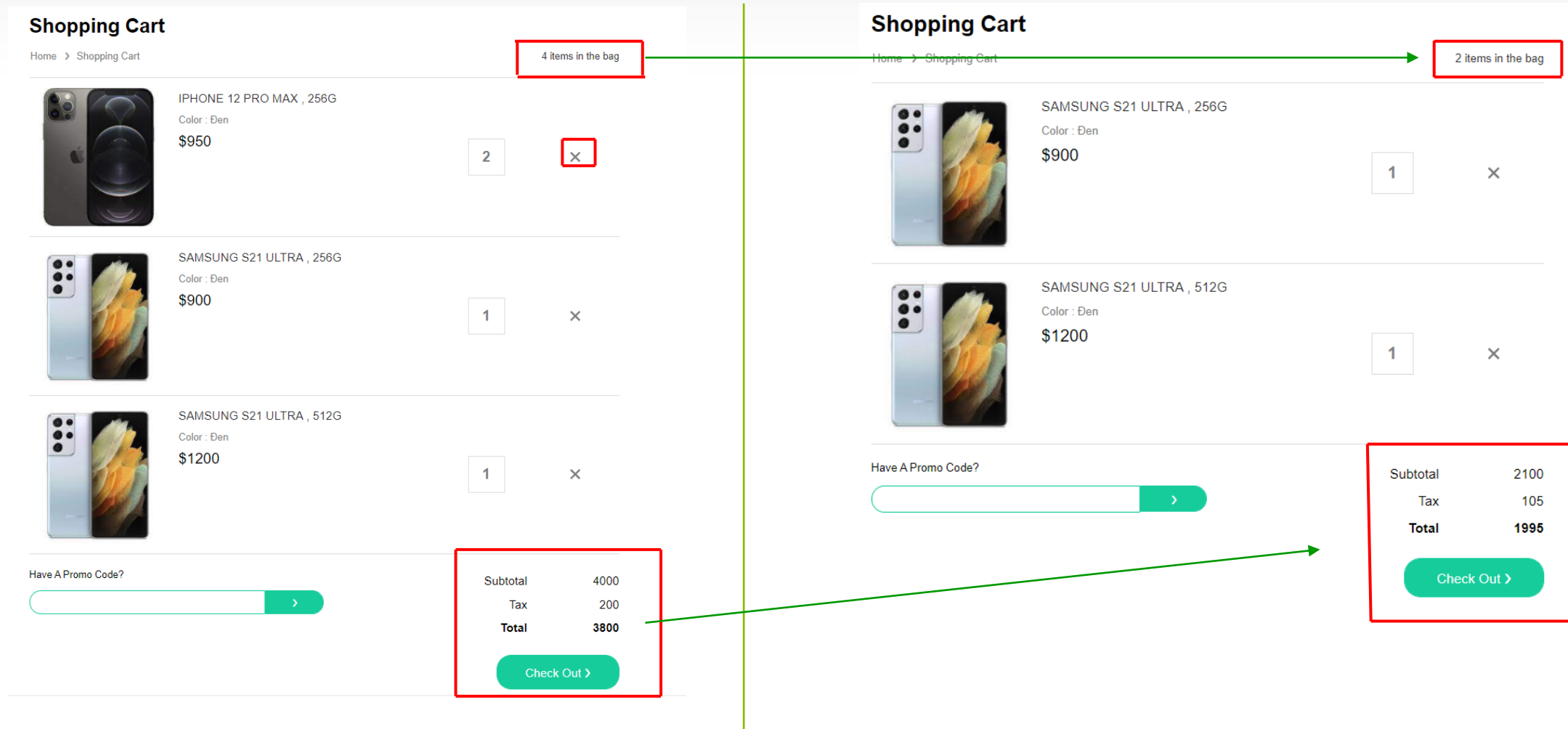
Vòng lặp tính lại tiền

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



2.44 Thiết kế chức năng Shoppingcart

2.4.1.1 Các chức năng xóa



CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



2.44 Thiết kế chức năng Shoppingcart

2.4.1.2 Các chức năng update số lượng

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG



2.44 Thiết kế chức năng Shoppingcart

2.4.1.3 Các chức năng check mã giảm giá

Chương 3. CƠ SỞ LÝ THUYẾT



3.1 Cơ sở lý thuyết

Giáo trình: CÁC CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI
(chủ đề: Tìm hiểu về Angular)

Các trang web tham khảo : youtube.com , google.com ,
ggdrive...

Chương 4. KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN



4.1 Kết quả đạt được

- Hiểu thêm về framework Angular