

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY  
UNIVERSITY OF TECHNOLOGY  
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



**Programming Integration Project (CO3103)**

---

**Project Report**

**Mobile Phone Selling Website**

---

Advisor: Prof. Quản Thành Thơ

HO CHI MINH CITY, OCTOBER 2021



## Contents

<b>1</b>	<b>Topic</b>	<b>3</b>
1.1	Proposed Features . . . . .	3
1.2	Proposed workload . . . . .	3
<b>2</b>	<b>Tools &amp; Technologies</b>	<b>4</b>
<b>3</b>	<b>Mock-ups</b>	<b>5</b>
<b>4</b>	<b>Technologies &amp; Features</b>	<b>10</b>
4.1	Frontend . . . . .	10
4.1.1	Component Reuse . . . . .	10
4.1.2	Virtual DOM . . . . .	10
4.1.3	JSX . . . . .	10
4.2	Backend . . . . .	11
4.2.1	Controllers . . . . .	11
4.2.2	Providers . . . . .	11
4.2.3	Modules . . . . .	12



## Member list

No.	Full name	Student ID
1	Nguyễn Hoàng	1952255
2	Nguyễn Chính Khôi	1952793
3	Vũ Anh Nhi	1952380
4	Lương Duy Hưng	1952747



# 1 Topic

This project focuses on creating a web-based application that allows users or customers to browse and make transactions. The specific target of product is phone, primarily smart mobile device.

This e-commerce application is analogous to real life phone selling retailers that require a more convenient and efficient way for customers to interact with their products. Because of that, the requirements and features will mainly focus on practical demands as if it was proposed to solve the problem.

## 1.1 Proposed Features

Name	Descriptions
Get news & offers	Display current news and offers
View catalog	Display list of products
Place orders	Add items to cart
Authenticate Member account	Login with username and password, (optional: SSO)
Store member account's balance & points	Save purchases to calculate points
Create coupons	Validate coupons when checking out
Display statistics	Generate monthly sales of reports (Optional)

## 1.2 Proposed workload

Task	Member
Create database & connect to backend	Hoàng & Hưng
Create mock-up homepage	Nhi & Khôi
Account registration and login	Frontend Khôi
	Backend Hưng
News and catalogue	Frontend Nhi
	Backend Hưng



## 2 Tools & Technologies

We decided to make the backend serve a RESTful API, so that we can decouple the server and client as well as ensuring ease of scale should that ever happen. With that said, we chose to demonstrate this project as a web application.

### a. Frontend: **React**

React (React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces or UI components. Developed at Facebook and released in 2013, it can be said that React has been the most influential UI library of recent years.

We use React to build components that represent logical reusable parts of the UI. The beauty of React is that the simplicity of building a component has been brought down to its theoretical minimum: a Javascript function. The return value from these functions is the HTML or UI, which is written in a special syntax called JSX, allowing easy combination of Javascript with Html markup.

The main reason we want to use React is not the library itself but the massive ecosystem surrounding it. React itself does not care about routing state management, animation or anything like that. Instead, it lets those concerns evolve naturally within the open-source community. No matter what we are trying to do, there is a good chance that a good supporting library to help us get it done has already existed.

### b. Backend: **NestJS**

NestJS is a Node.js framework for building scalable server-side applications with Typescript. It comes with a ton of built-in modules to work with databases, handle security, implement streaming and anything else you can imagine doing in a server-side application.

Using the NestJS CLI, you can scaffold out a new project with a code base pre-configured with Jest for testing and set up with Typescript to write more readable and reliable code.

### c. Database: **PostgreSQL**

PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads.

With more than 30 years of active development on the core platform, PostgreSQL has earned a strong reputation for its proven architecture, reliability, data integrity and more.

### 3 Mock-ups

We have sketched the design of four pages that is in the scope of this project. They are the Home page, Product (displaying list of products) page, Product Detail, and View Cart page. We also sketched the log in pop-up window if customers clicked on the icon.

This is the general flow of the website. Most of the pages are linked by the buttons on the navigation bar.

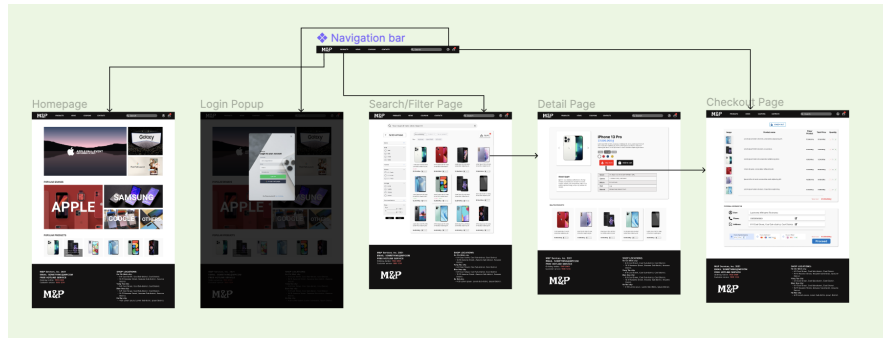


Figure 1: General Screen Flow

These are the screenshots of the pages we designed using Figma. Figma is a vector graphics editor and prototyping tool, which is suitable for basic UI design. For a higher resolution view, please see the link [Mock-up](#).

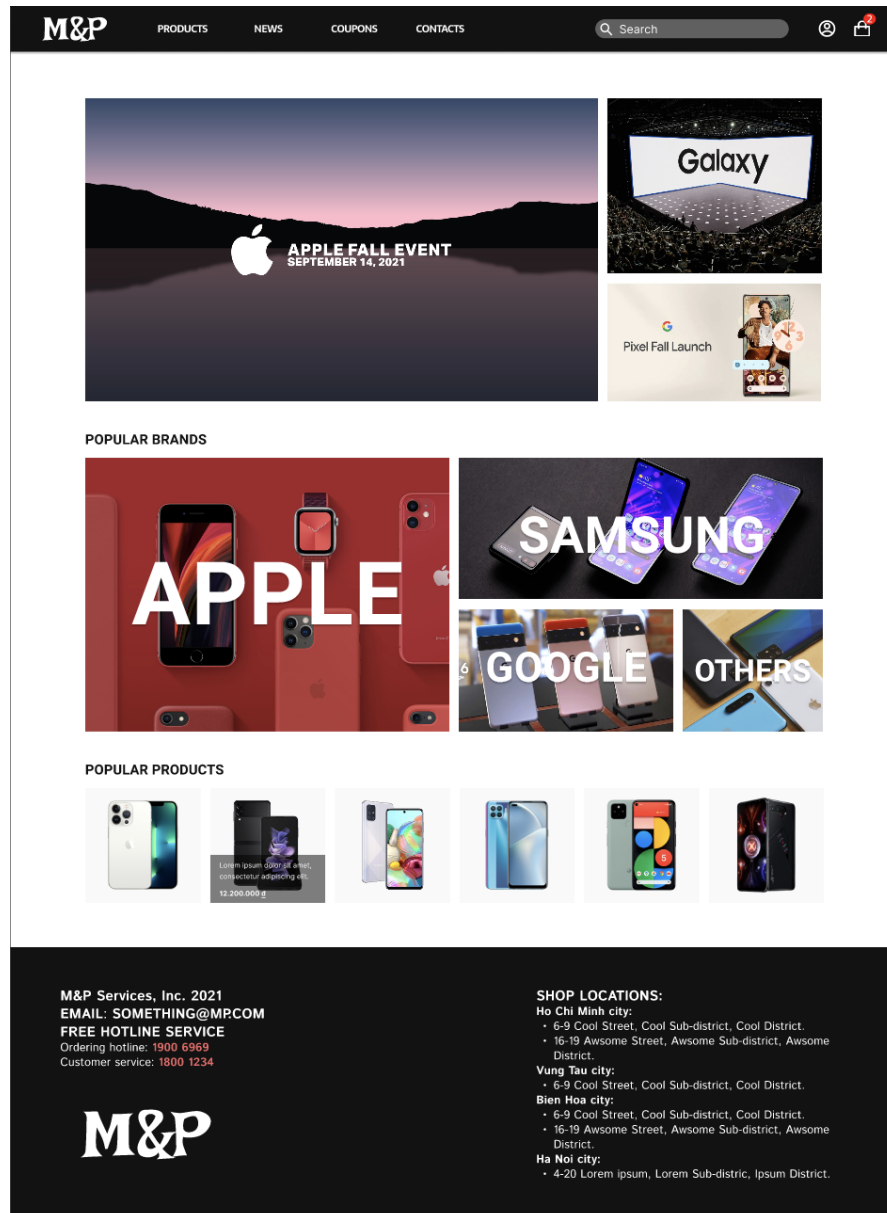


Figure 2: Homepage

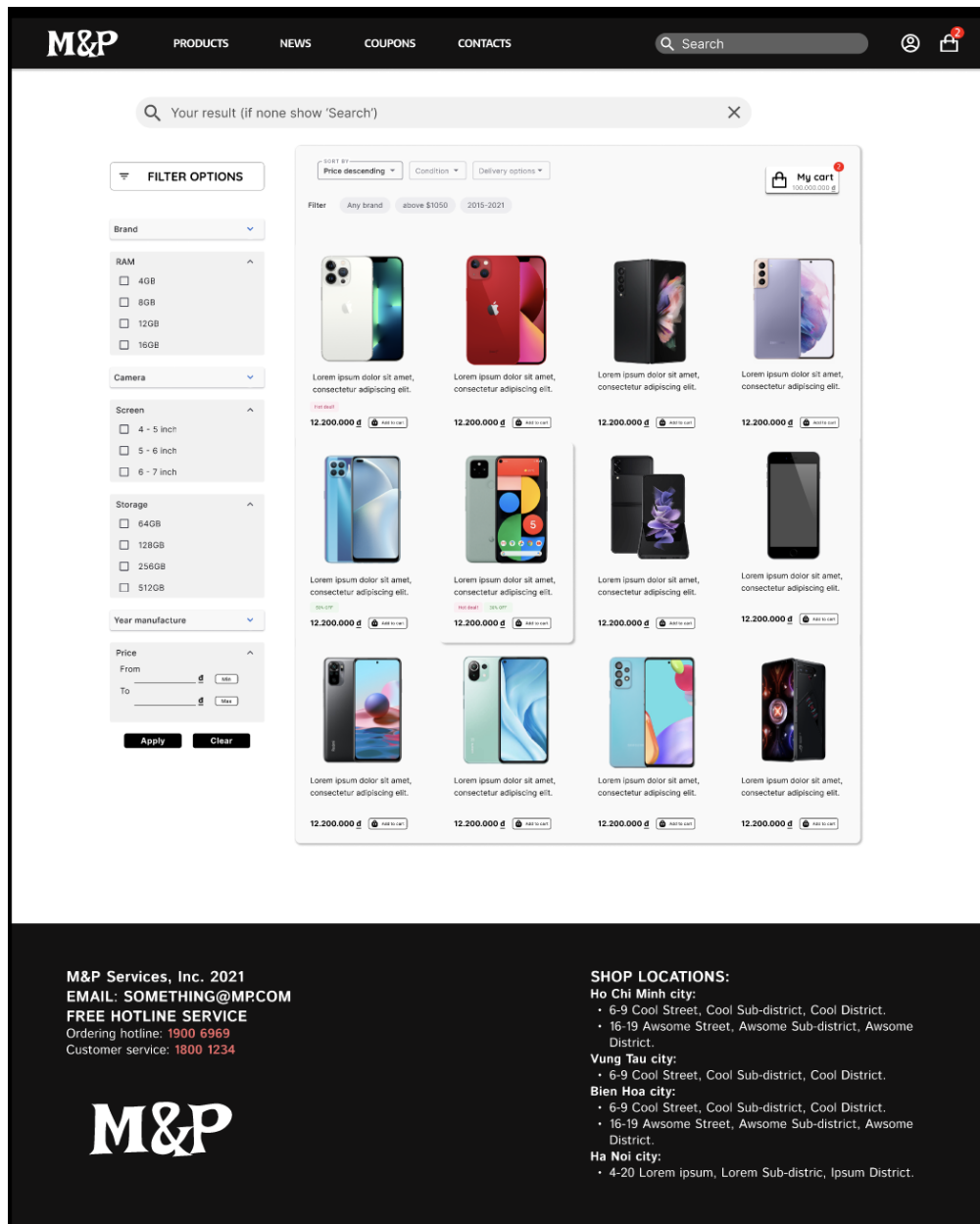


Figure 3: List of Products





M&P

PRODUCTS

NEWS

COUPONS

CONTACTS

Search

iPhone 13 Pro

27.000.000 đ

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

128GB

256GB

512GB

Buy now

Add to cart

About Apple

Apple Inc. is an American multinational technology company that specializes in consumer electronics, computer software, and online services. Apple is the world's largest technology company by revenue and valuable.

Screen	6.1" OLED, Super Retina XDR, ProMotion 120Hz
Camera	3 cameras 12 MP, LiDAR sensor
Battery	Up to 22 hours
RAM	4 GB
Material	Stainless Steel, Ceramic Shield

RELATED PRODUCTS

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

12.200.000 đ

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

12.200.000 đ

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

12.200.000 đ

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

12.200.000 đ

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

12.200.000 đ

M&P Services, Inc. 2021  
EMAIL: SOMETHING@MPCOM  
FREE HOTLINE SERVICE  
Ordering hotline: 1900 6969  
Customer service: 1800 1234

M&P

SHOP LOCATIONS:

Ho Chi Minh city:

- 6-9 Cool Street, Cool Sub-district, Cool District.
- 16-19 Awsome Street, Awsome Sub-district, Awsome District.

Vung Tau city:

- 6-9 Cool Street, Cool Sub-district, Cool District.

Bien Hoa city:

- 6-9 Cool Street, Cool Sub-district, Cool District.
- 16-19 Awsome Street, Awsome Sub-district, Awsome District.

Ha Noi city:

- 4-20 Lorem ipsum, Lorem Sub-district, Ipsum District.

Figure 4: Product Details



M&P

PRODUCTSNEWSCOUPONSCONTACTS

Search

2

CHECK OUT

Image	Product name	Price/ Product	Total Price	Quantity
	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	12.200.000 đ	12.200.000 đ	<div>- 1 +</div>
	Lorem ipsum dolor sit amet, consectetur.	4.200.000 đ	8.400.000 đ	<div>- 2 +</div>
	Lorem ipsum dolor sit consectetur adipiscing amet.	9.200.000 đ	9.200.000 đ	<div>- 1 +</div>
	Lorem sit amet, consectetur adipiscing elit.	5.200.000 đ	26.000.000 đ	<div>- 5 +</div>
	Ipsum dolor sit amet, consectetur amet adipiscing elit.	14.500.000 đ	14.500.000 đ	<div>- 1 +</div>
	Lorem ipsum dolor sit amet, consectetur adipiscing.	3.000.000 đ	21.000.000 đ	<div>- 7 +</div>

Total Cost: 91.300.000 đ

PERSONAL INFORMATION

User:

Firstname Midname Lastname

Phone:

0969696969

Address:

6-9 Cool Street, Cool Sub-district, Cool District.

Pay by Payment Systems

Pay by Credit Card

Pay by E-Wallet

Total Cost: 91.300.000 đ

Proceed

M&P Services, Inc. 2021  
EMAIL: SOMETHING@MP.COM  
FREE HOTLINE SERVICE  
Ordering hotline: 1900 6969  
Customer service: 1800 1234

M&P

SHOP LOCATIONS:  
Ho Chi Minh city:  
• 6-9 Cool Street, Cool Sub-district, Cool District.  
• 16-19 Awesome Street, Awesome Sub-district, Awesome District.  
Vung Tau city:  
• 6-9 Cool Street, Cool Sub-district, Cool District.  
Bien Hoa city:  
• 6-9 Cool Street, Cool Sub-district, Cool District.  
• 16-19 Awesome Street, Awesome Sub-district, Awesome District.  
Ha Noi city:  
• 4-20 Lorem ipsum, Lorem Sub-district, Ipsum District.

Figure 5: View Cart

## 4 Technologies & Features

So far, we have been learning about the thinking process of a developer. For this integration project, we decided to practice this thinking process using some new frameworks. At this point, we have already picked our tools, and after hours of research, we draw the conclusion on how all these technologies work and work together.

### 4.1 Frontend

#### 4.1.1 Component Reuse

Why would you constantly reinvent the wheel when you can simply reuse code that has already been written and tested by other developers?

ReactJS introduces the so-called components, which make it possible to split the UI into independent, reusable pieces, and think about each piece in isolation.

For instance, we can have a button component display with different colors in several parts of our application. Although it is the same button component when we provide it with a data set (e.g color, or a function), it modifies itself and outputs a UI instance of the element.

This pattern of creating React components is necessary for scaling. It helps save time by ensuring less code is written, development is faster, the code base is simpler, and maintenance is stress-free.

#### 4.1.2 Virtual DOM

The Document Object Model (DOM) is an application programming interface that represents an XML document as a tree structure wherein each node is an object representing a part of the document.

Most often it's inefficient and slow because it's necessary to recalculate the CSS, recreate the layout, and essentially repaint the entire web page every time the DOM changes.

ReactJS overcomes the DOM's inefficiencies by using the so-called Virtual DOM.

Just like the actual DOM, the Virtual DOM represents all elements and their attributes as a node tree. When something changes, React JS updates the Virtual DOM and figures out how it differs from the actual DOM, updating the actual DOM only with what has actually changed.

#### 4.1.3 JSX

ReactJS introduces another concept called JSX, and it is a syntax extension to JavaScript.

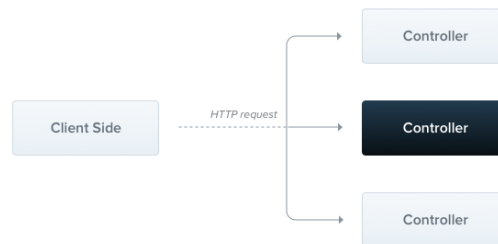
React embraces the fact that rendering logic is inherently coupled with other UI logic: how events are handled, how the state changes over time, and how the data is prepared for display.

Instead of artificially separating technologies by putting markup and logic in separate files, React separates concerns with loosely coupled units called "components" that contain both.

React doesn't require using JSX, but most people find it helpful as a visual aid when working with UI inside the JavaScript code. It also allows React to show more useful error and warning messages.

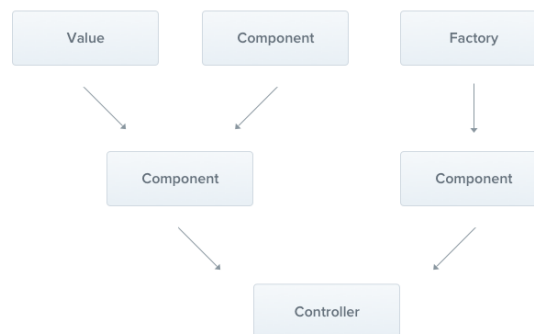
## 4.2 Backend

### 4.2.1 Controllers



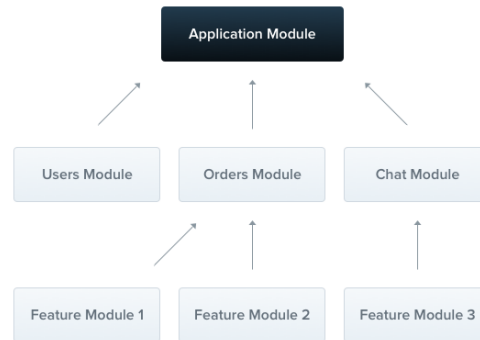
A controller's purpose is to receive specific requests for the application. The routing mechanism controls which controller receives which requests. Frequently, each controller has more than one route, and different routes can perform different actions.

### 4.2.2 Providers



Nest introduces a new concept called providers. Many of the basic Nest classes may be treated as a provider — services, repositories, factories, helpers, and so on. The main idea of a provider is that it can be injected as dependency; this means objects can create various relationships with each other, and the function of “wiring up” instances of objects can largely be delegated to the Nest runtime system.

### 4.2.3 Modules



Each application has at least one module, a root module. The root module is the starting point Nest uses to build the application graph — the internal data structure Nest uses to resolve module and provider relationships and dependencies.

While very small applications may theoretically have just the root module, this is not the typical case. We want to emphasize that modules are strongly recommended as an effective way to organize your components. Thus, for most applications, the resulting architecture will employ multiple modules, each encapsulating a closely related set of capabilities.

The module encapsulates providers by default. This means that it's impossible to inject providers that are neither directly part of the current module nor exported from the imported modules. Thus, you may consider the exported providers from a module as the module's public interface, or API.