

FPT – APTECH COMPUTER EDUCATION

RHEEM COOLING

Center Name	FPT – Aptech – HCM Address: 21 Bis Hau Giang, Tân Sơn Nhất Ward, Ho Chi Minh City		
Instructor	Ms. Le Mong Thuy		
Class	T3.2502.E0		
Group	2		
Group members	Name	Student ID	
	Pham Gia Bao	Student1628435	
	Nguyen Thanh Bao	Student1634684	
	Vo Cong Quoc Quang	Student1634688	
Duration	Start date	23 rd July 2025	
	End date	23 rd August 2025	

This is to certify that

Mr.:

Pham Gia Bao

Nguyen Thanh Bao

Vo Cong Quoc Quang

has successfully designed & developed

eProject: Rheem Cooling WebSite

Submitted by:

Pham Gia Bao

Date of issue:

Authorized Signature:

TABLE OF CONTENTS

FPT – APTECH COMPUTER EDUCATION	1
ACKNOWLEDGMENT.....	3
SYNOSIS	3
PROBLEM DEFINITION	4
CUSTOMER'S REQUIREMENTS SPECIFICATIONS (CRS)	5
Business/Project Objective:.....	5
Functional Requirements Specifications:.....	5
Hardware/Software Requirements:	6
Project Deliverables:	7
SCOPE OF THE WORK (IN BRIEF).....	8
LIMITATIONS OF THE PROJECT	9
ARCHITECTURE AND DESIGN OF THE SYSTEM	11
DIAGRAM OF THE WEBSITE.....	12
TASK SHEET REVIEW 1	13
SITE MAP	14
MOCK OF THE WEB SITE	15
1. Master Layout.....	15
2. Homepage.....	16
3. Product.....	17
4. Product Details.....	18
5. Contact Us.....	19
6. About us	20
7. Customer Support.....	21
TASK SHEET REVIEW 2	22
WEBSITE DESCRIPTION	23
RESPONSIVE	42
JAVASCRIPT DESCRIPTION.....	45
TASK SHEET REVIEW 3	66

ACKNOWLEDGMENT

The team extends its gratitude to everyone who supported the successful completion of this eProject report. Special thanks are given to the instructor, **Ms. Le Mong Thuy**, for her meticulous, enthusiastic, and strict guidance since the beginning of their studies at FPT Aptech. The team members are also thanked for their hard work and earnest contributions throughout the month of collaboration. Finally, appreciation is extended to classmates and family for sharing, creating the best environment for project focus, and motivating the team to achieve its goals.

SYNOSIS

The eProject focuses on developing a Single-Page-Application (SPA) and a responsive Website for Rheem Cooling. Rheem Cooling is identified as a specialist Air Conditioner retail store, which was founded in 1998. The company brings together multi-brand air conditioning products under one roof.

Rheem Cooling currently operates with products from over 20 brands distributed across more than 40 stores. The company's promise is to deliver a "Brighter Every Day" experience, achieved through providing a world-class ambiance for shopping both in-store and online. They aim to create a seamless omnichannel shopping experience, allowing customers to enjoy the benefits of both online and offline worlds. Additionally, Rheem Cooling emphasizes customer support, with experts available at their stores to assist customers in finding the right Air Conditioner to meet their specific needs.

The primary objective of this web development project is to establish an online presence for Rheem Cooling that accurately reflects their business model and enhances the customer experience. The developed website is required to function effectively across all leading browsers, including Chrome, IE, and Firefox.

PROBLEM DEFINITION

The project addresses the need for Rheem Cooling, a specialized Air Conditioner retail store, to establish and enhance its digital presence.

Context of Rheem Cooling:

- Rheem Cooling was founded in 1998.
- It operates as a retail store that brings together multi-brand air conditioning products under one roof.
- The company boasts a significant scale, representing over 20 brands and operating in more than 40 stores across various locations.
- Rheem Cooling's core promise is to deliver a "Brighter Every Day" experience to its customers.
- They strive to offer a world-class ambiance for shopping, both in-store and online.
- A key business objective is to create a seamless omnichannel shopping experience, enabling customers to leverage the benefits of both online and offline purchasing.
- To ensure customer satisfaction and informed decision-making, Rheem Cooling provides experts at their physical stores who assist customers in selecting the appropriate Air Conditioner for their specific needs.

The Problem/Challenge to be Addressed by the Project:

Despite its established physical presence and customer-centric approach, the implied problem is the absence or inadequacy of a modern, efficient, and user-friendly online platform that fully supports Rheem Cooling's multi-brand, omnichannel strategy.

To overcome this, the project is tasked with creating a Single-Page-Application (SPA) and a responsive Website for Rheem Cooling. This digital platform must align with the company's business model and enhance the customer experience by providing comprehensive product information and seamless interaction online.

Key Technical Requirement for the Solution:

The developed website must be fully functional and perform well across all leading web browsers, specifically including Chrome, Internet Explorer, and Firefox.

CUSTOMER'S REQUIREMENTS SPECIFICATIONS (CRS)

The development of the Rheem Cooling website is driven by specific requirements from the client to establish a robust and user-friendly online presence.

Client: Rheem Cooling

Business/Project Objective:

The primary objective of this project is to create a Single-Page-Application (SPA) and a responsive Website for Rheem Cooling. This digital platform is intended to align with Rheem Cooling's business model as a specialist Air Conditioner retail store, which was founded in 1998 and brings together multi-brand air conditioning products under one roof. The website should enhance the customer experience and reflect Rheem Cooling's promise of a "Brighter Every Day" through a world-class online shopping ambiance, fostering a seamless omnichannel shopping experience. The developed website must function effectively across all leading browsers, specifically including Chrome, Internet Explorer, and Firefox.

Functional Requirements Specifications:

The portal will be designed as a Single-Page-Application and a responsive Website with a set of pages and menus to represent various activities.

- Visual Design: Pages, menus, and other visual elements must be designed in a visually appealing manner with attractive fonts, colors, and animations. The layout must also be responsive.
- Top of the Page: Should feature a suitable logo and a banner with images of air conditioners.
- Navigation Menu: Must contain links to specific AC types such as Split AC, Window AC, Cassette AC, etc..

- Product Sections: Various sections like "Top Selling ACs", "Window AC", "Split AC", "Cassette AC" should be provided with listings of products.
- Product Detail Pop-up: Clicking on a product should open a pop-up window displaying images of the product along with its description and price.
- Product Filtering: A filter functionality must be provided with options based on "AC Type", "Tonnage", "Color", "Coverage Area", "Brand", and "Price".
- Image Gallery: A gallery should be included for viewing different product images.
- Product Rating: Viewers must be allowed to enter ratings for products.
- Standard Page Links: Site map, Gallery, About Us, FAQ section link, and Contact Us link must be added.
- About Us and Contact Us Details: These sections should display the Email ID, address, and contact number of Rheem Cooling.
- Navigation Smoothness: Navigation throughout the website must be made smoother.
- Design Uniformity: Uniformity in the design must be maintained across the entire website.
- Continuous Scrolling Ticker: A continuous scrolling ticker displaying the current date, time, and location (using HTML5 geolocation features) should be at the bottom of the page.
- Visitor Count: A visitor count should be displayed at the top right corner of the page next to the logo image.
- Menu Interaction: Menu options should change color on hover and also after clicking.
- Visual Effects: Fade in and fade out options can be used for the menus.

Hardware/Software Requirements:

- Operating System: The portal should operate on Windows.
- Browsers: The website must be compatible with and function well on Edge, Chrome, Mozilla Firefox, and Safari.
- Frontend Technologies: The primary technologies to be used for the frontend

development include HTML5, CSS, Bootstrap, JavaScript, jQuery, React/AngularJS, Figma, and XML.

- Data Storage: Data will be stored using JSON files or TXT files.
- Hardware for Development:
 - Intel Core i3/i5 Processor or higher.
 - 8 GB RAM or above.
 - Color SVGA.
 - 500 GB Hard Disk space.
 - Mouse, Keyboard.

Project Deliverables:

The project team is required to design and build the project, and then submit it along with a complete project report. The report must include:

- Problem Definition.
- Design specifications.
- Diagrams such as flowcharts for various activities and Data Flow Diagrams (DFDs).
- Source Code.
- Test Data Used in the Project.
- Project Installation Instructions (if any). Documentation is considered very important and must be complete and comprehensive. The consolidated project will be submitted as a zip file with a ReadMe.doc listing any assumptions made. A video clip demonstrating the working of the Website must also be submitted. Optionally, a live hosted URL can be provided for the site. The team is encouraged to apply their creativity and logic to improve the portal beyond the given specifications.

SCOPE OF THE WORK (IN BRIEF)

The project involves the design and development of a Single-Page-Application (SPA) and a responsive Website for Rheem Cooling. After a long and detailed discussion, our group has decided to create a Single Page Application (SPA) and responsive website for Rheem Cooling, containing the following webpages and functionalities:

1. Homepage:

Includes a search bar for finding products, displays promotional banners, best-selling air conditioners, and introduces the Rheem Cooling brand. The header will show the company logo and a real-time visitor counter on the top right corner.

2. Product:

Displays all available air conditioner models based on categories such as Split AC, Window AC, Cassette AC, etc. Users can filter products by AC type, cooling capacity (tonnage), color, coverage area, brand, and price. Clicking on a product opens a pop-up with detailed information, images, and price.

3. About Us:

Displays company profile, customer reviews, and relevant background information about Rheem Cooling.

4. Contact Us

- a. Displays company **contact information**: phone number, email, and operating hours.
- b. Embeds a **map** showing all **store branches/locations** with Google Maps integration.
- c. Shows **social media links**: Facebook, Instagram, YouTube, and LinkedIn.
- d. Provides a **contact form** for users to send inquiries, suggestions, or service requests.
- e. Fields: Name, Email, Subject, Message, and optional file upload.
- f. Includes a success/failure alert when the message is submitted.

5. Customer Support

- a. FAQ : Contains answers to common customer questions related to products, services, payments, and installation.
- b. Site Map Page: Visual and textual overview of the website's structure and main sections to improve accessibility and SEO.

LIMITATIONS OF THE PROJECT

Based on the provided information for the Rheem Cooling website project, here are the areas that were explicitly not within the project's scope or not fully implemented:

- No Robust or Complex Backend Database System: The Rheem Cooling website project was specified to use JSON files or TXT files as the data store. This implies that a more robust and complex backend database system (such as SQL or NoSQL databases) was not within the scope of the project. Consequently, functionalities requiring complex data management, advanced queries, or large-scale data persistence might not be supported.
- Lack of Comprehensive Server-Side Logic: The technical requirements focused heavily on frontend technologies like HTML5, CSS, Bootstrap, JavaScript, jQuery, React/AngularJS. However, no server-side programming languages (e.g., Node.js, Python, PHP, Java) were explicitly mentioned in the sources. This suggests that the project did not include the development of complex functionalities requiring extensive server-side data processing, more robust user authentication, or server-side session management, focusing instead on a client-side heavy application with file-based data storage.
- No Full E-commerce Functionality (Shopping Cart, Payment Gateway, Order Management): While the Rheem Cooling website is designed to display air conditioning products, prices, and detailed product information, the functional requirements do not include core features of a full e-commerce platform, such as a shopping cart, payment gateway integration, or an order management system for direct online sales.
- No Content Management System (CMS): The sources do not describe any Content Management System or similar interface that would allow Rheem Cooling staff to independently update product information, news articles, or promotional content without requiring direct modification of the website's underlying code or the JSON/TXT data files.
- No Complex User Account Management or Login/Registration Features: Although the website allows viewers to enter product ratings, there are no explicit requirements for a user registration or login system for personal accounts, user profiles, purchase history, or other advanced personalization options.
- No Real-time Inventory Management Integration: Given that the project uses JSON or

TXT files for product data storage, there is no requirement for integration with a real-time inventory management system, which could be an important factor for a multi-brand retail store like Rheem Cooling.

ARCHITECTURE AND DESIGN OF THE SYSTEM

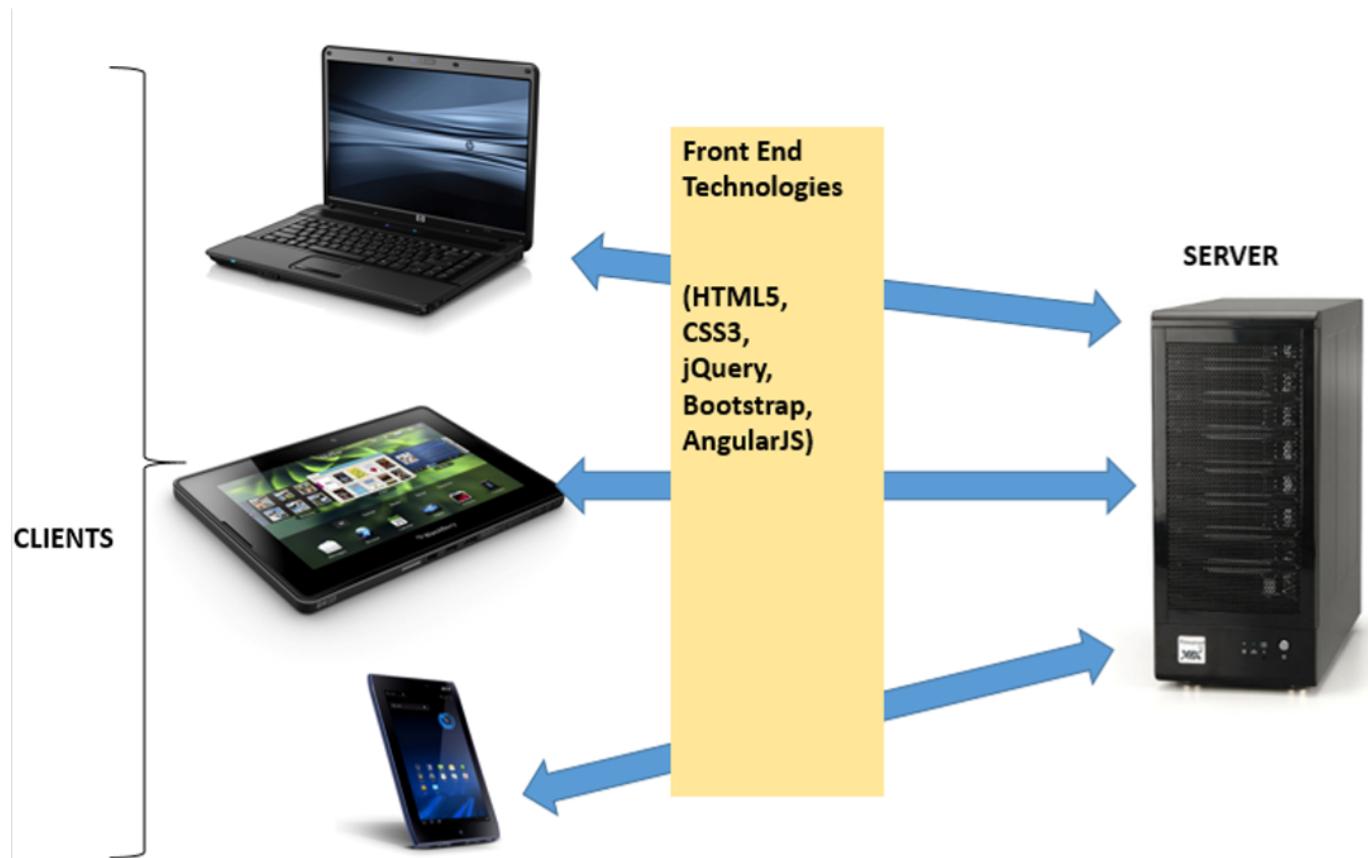
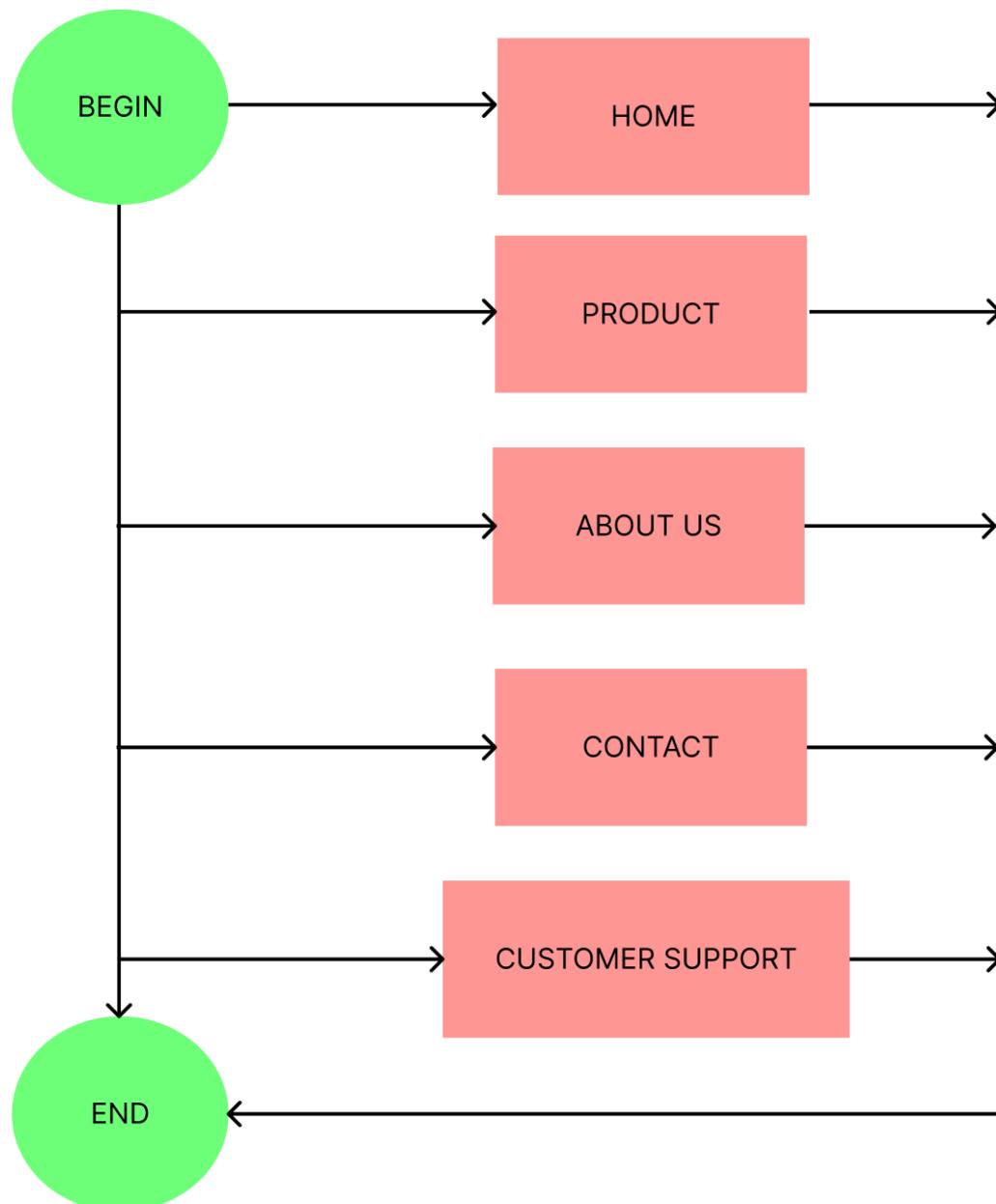


DIAGRAM OF THE WEBSITE

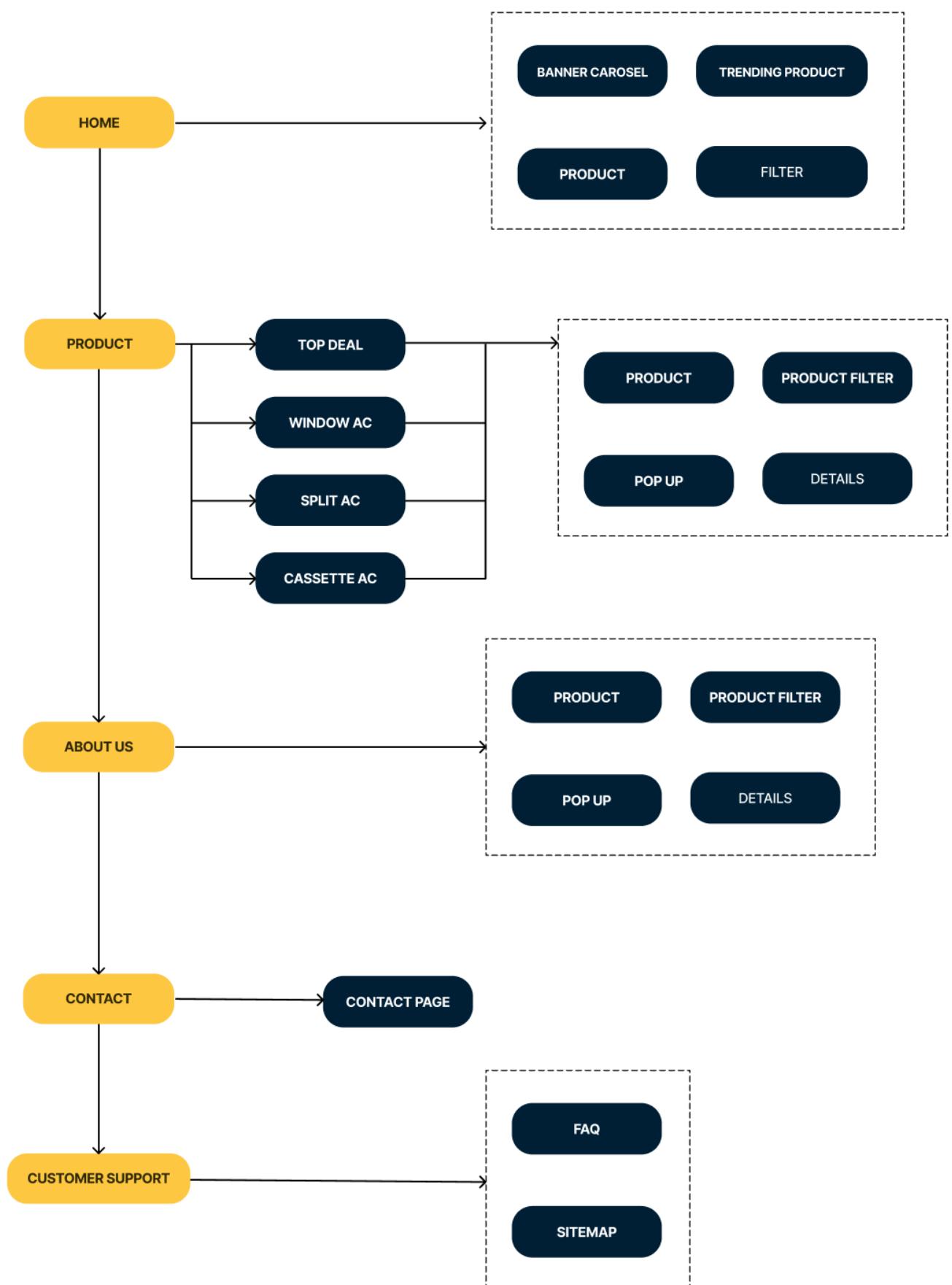


TASK SHEET REVIEW 1

Project Ref. No.: eP/Advertisement Portal Management System/01		Project Title: RHEEM COOLING	Activity Plan Prepared By: Pham Gia Bao	Date of Preparation of Activity Plan:			
Sr. No.	Task			Actual Start Date	Actual Days	Team Mate Names	Status
1	Problem Statement			23-July-2025	1	Quang	Completed
2	Analysis of Rheem Cooling requirements about the website			23-July-2025	1	Quang	Completed
3	The scope of the work (in brief)			23-July-2025	1	Quang	Completed
4	Architecture and design of the system			23-July-2025	1	Team	Completed
5	Diagram of the website			23-July-2025	1	Team	Completed

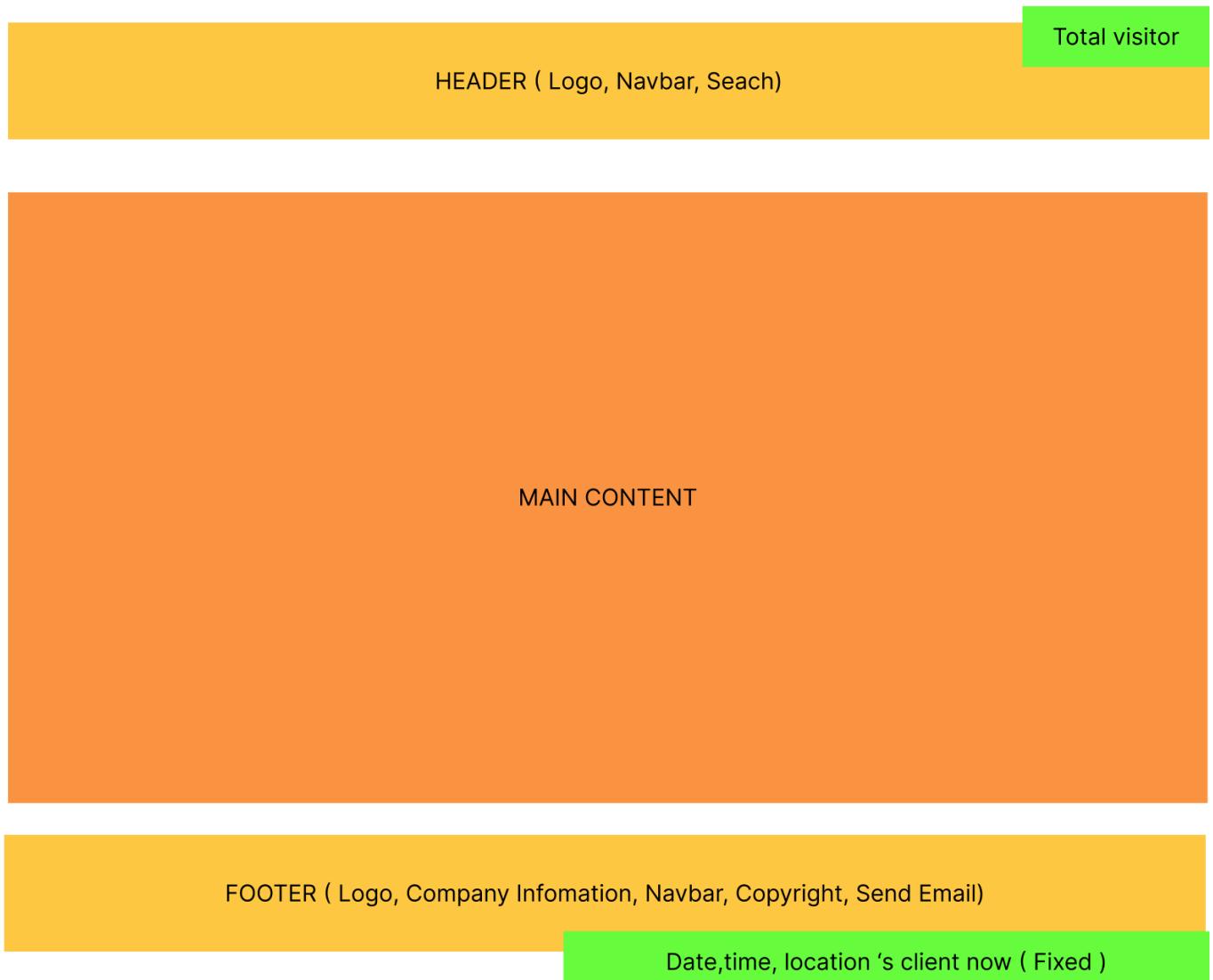
Date:	
Signature of Instructor: Ms. Le Mong Thuy	Signature of Team Leader: Pham Gia Bao

SITE MAP

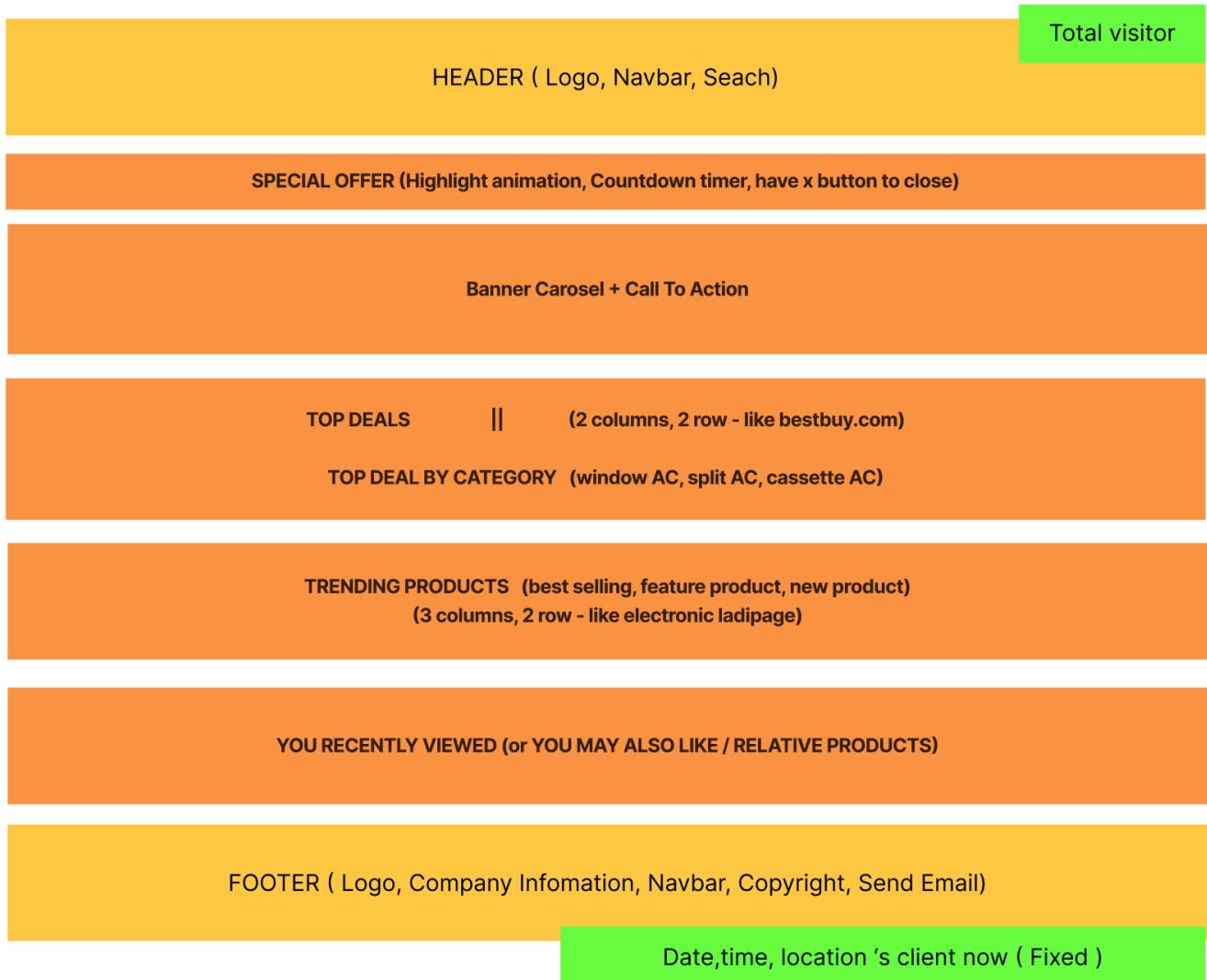


MOCK OF THE WEB SITE

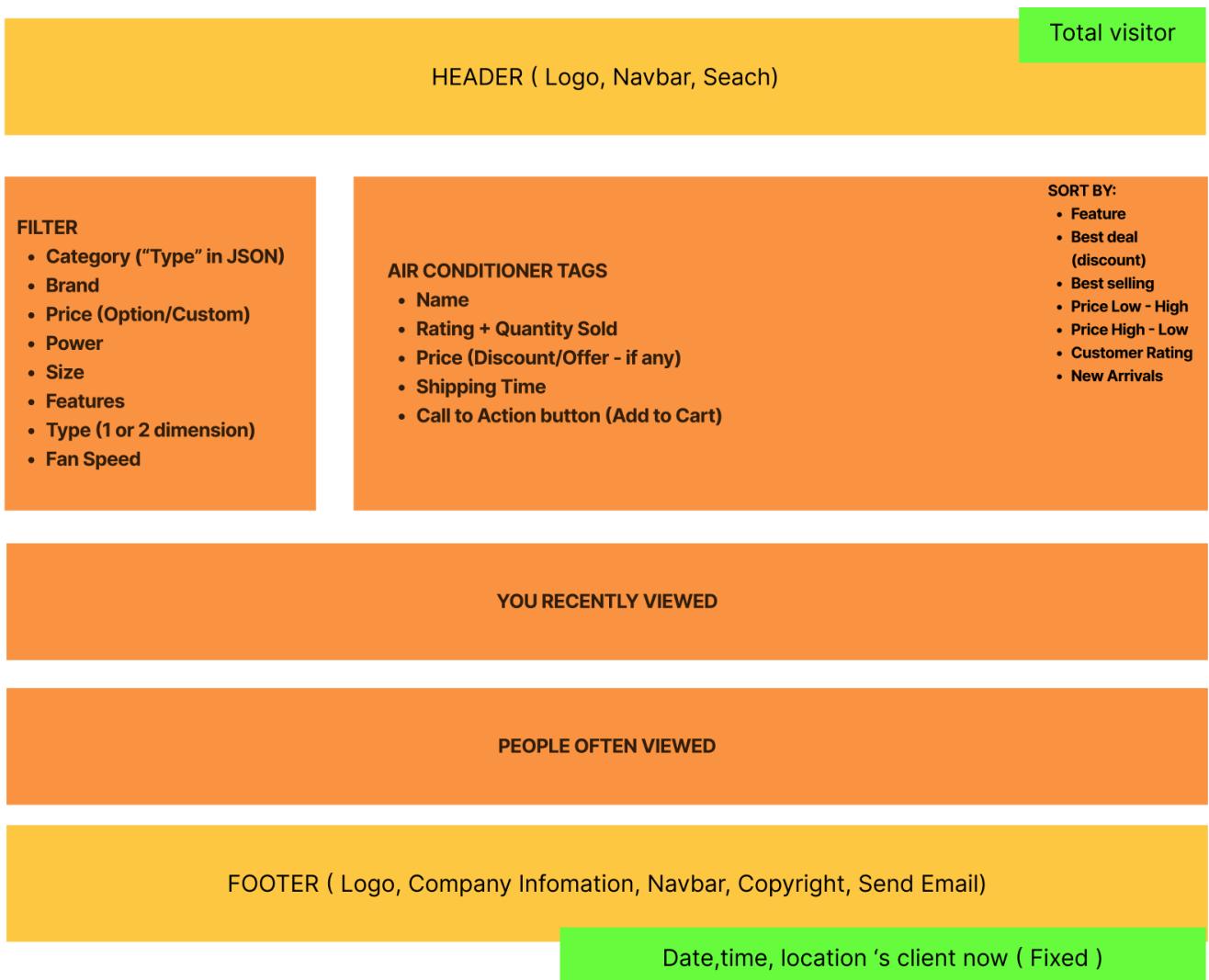
1. Master Layout



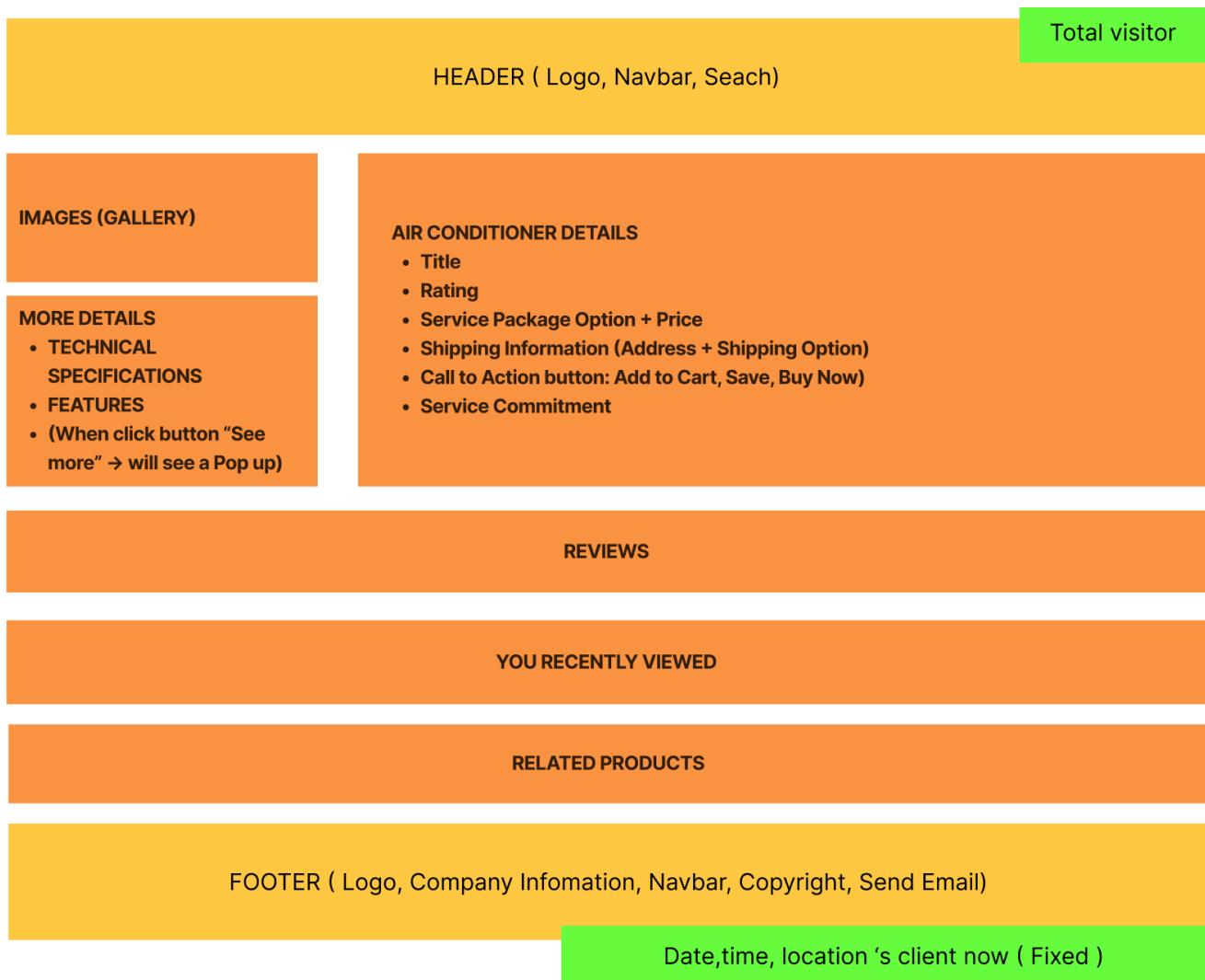
2. Homepage



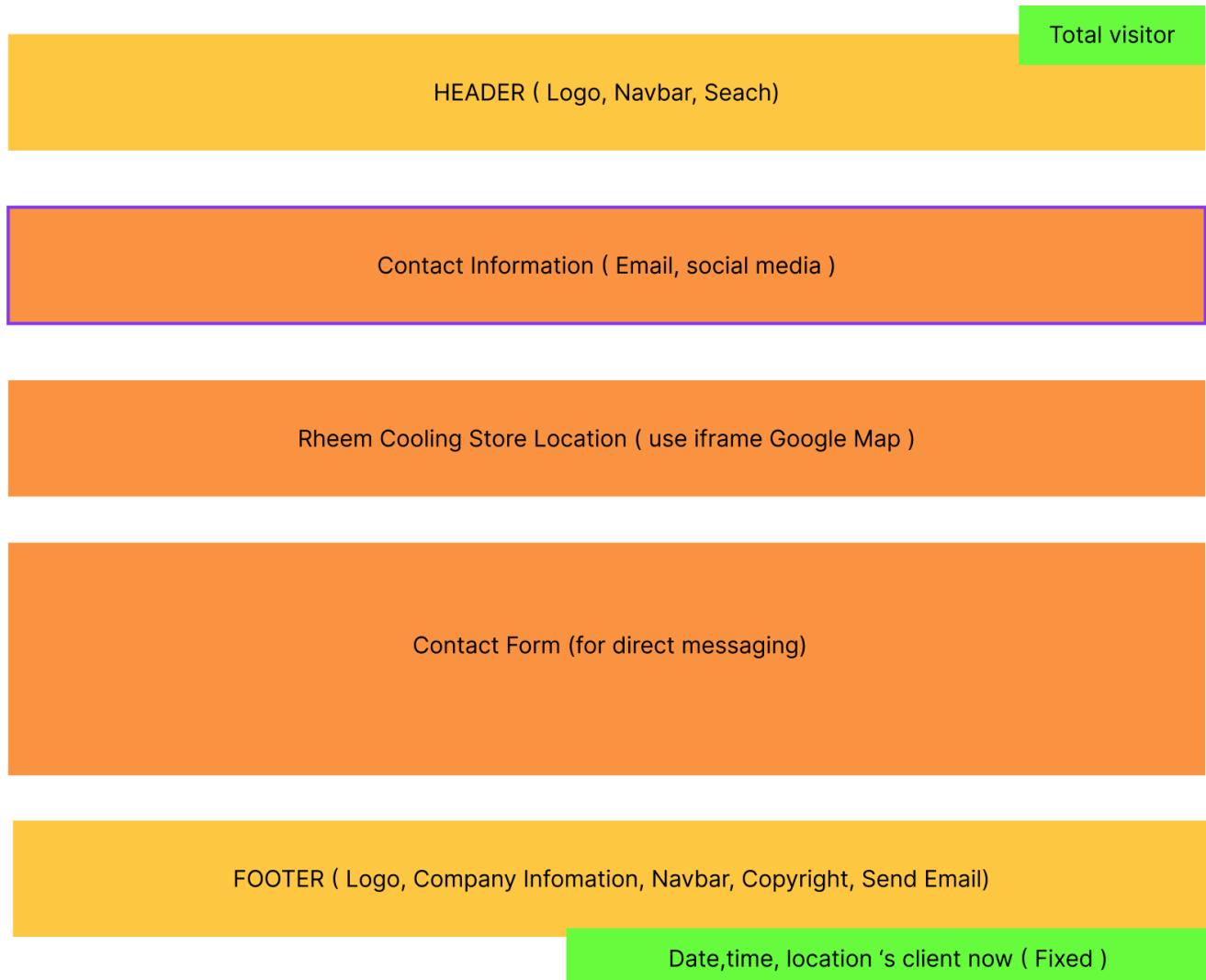
3. Product



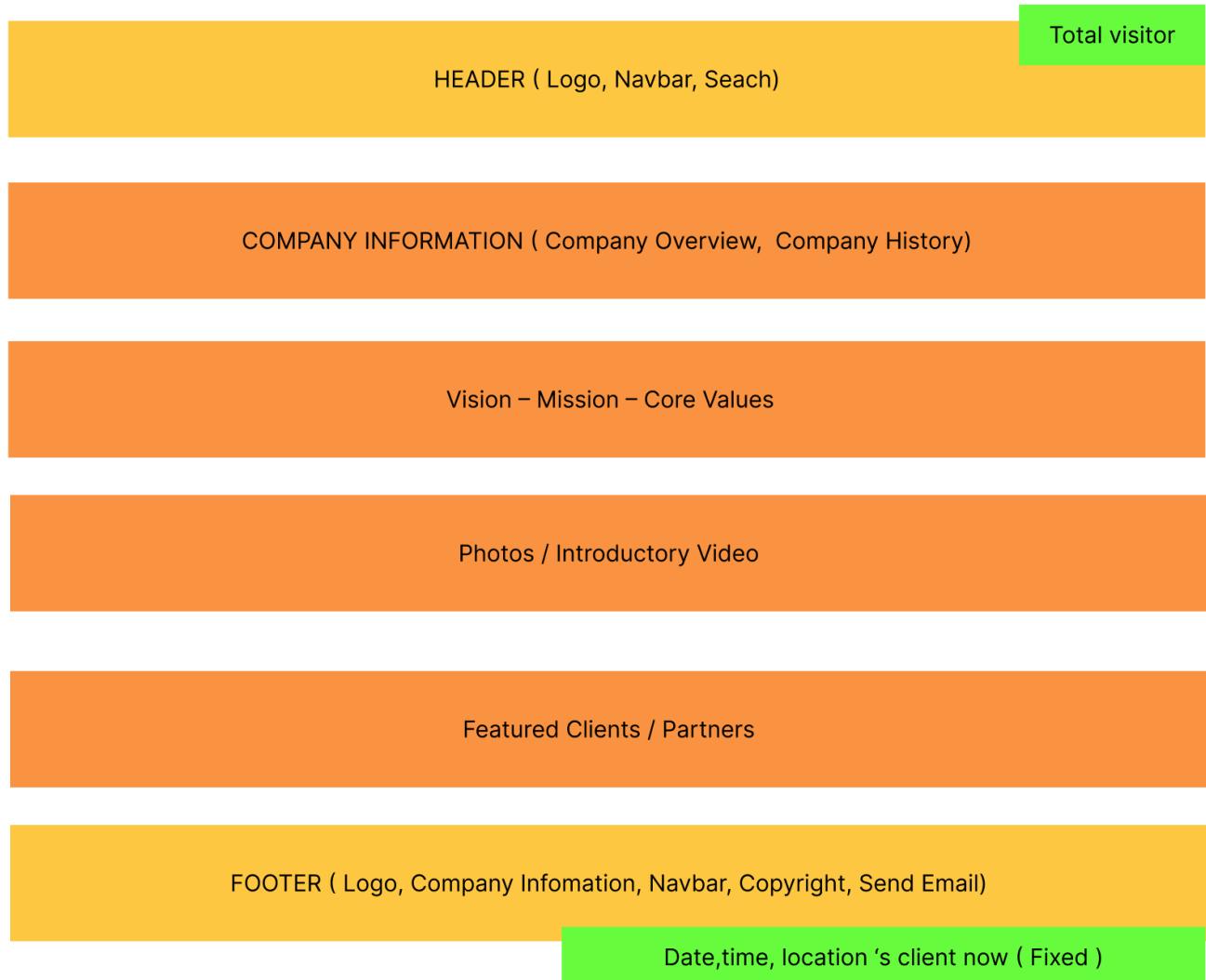
4. Product Details



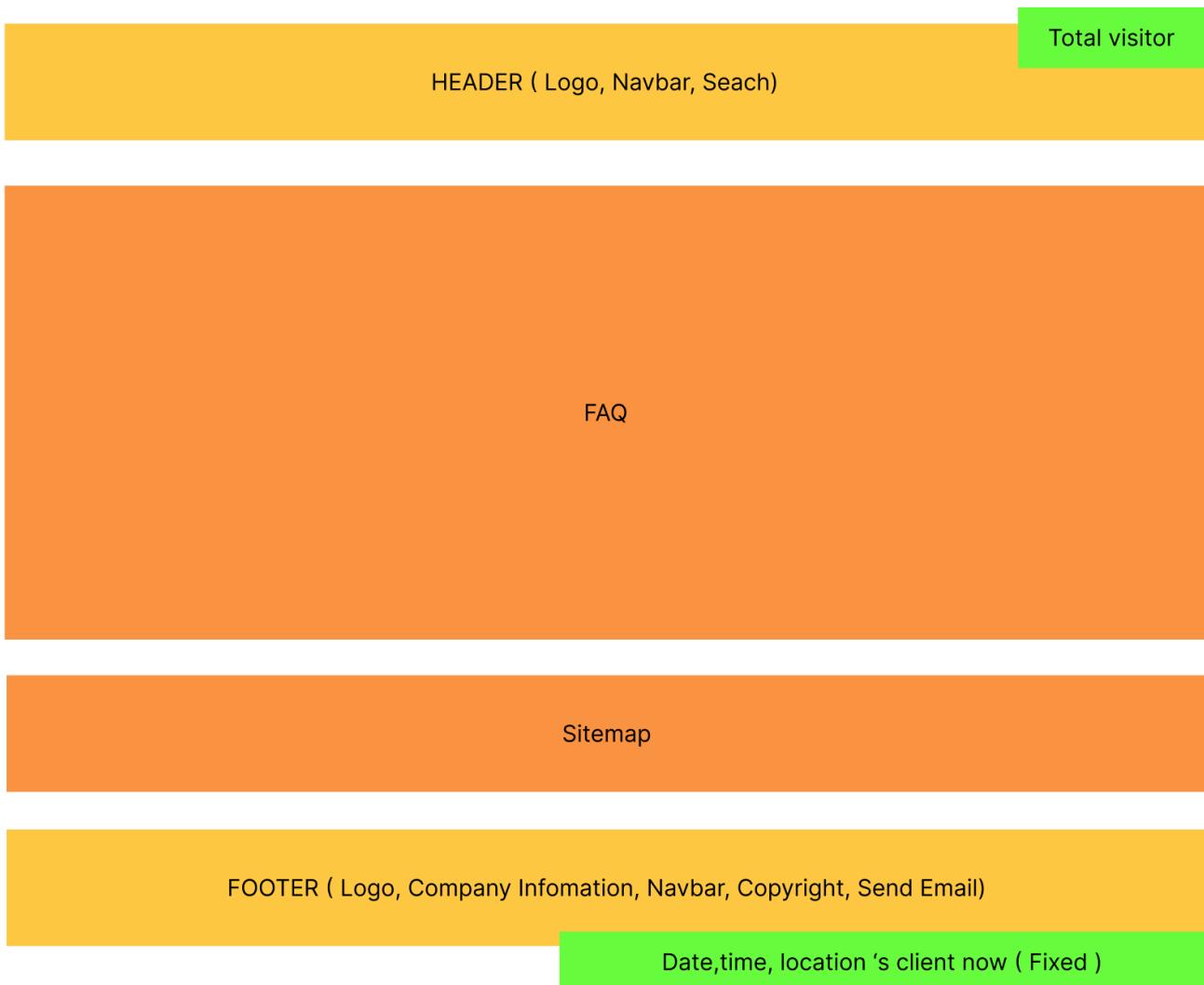
5. Contact Us



6. About us



7. Customer Support



TASK SHEET REVIEW 2

Project Ref. No.: eP/Advertisement Portal Management System/01		Project Title: RHEEM COOLING	Activity Plan Prepared By: Pham Gia Bao	Date of Preparation of Activity Plan:			
Sr. No.	Task			Actual Start Date	Actual Days	Team Mate Names	Status
1	Site Map			25-July-2025	1	Team	Completed
2	Mock of the website			25-July-2025	2	Bao & Bao	Completed

Date: 25 July 2025

Signature of Instructor:

Ms. Le Mong Thuy

Signature of Team Leader:

Pham Gia Bao

WEBSITE DESCRIPTION

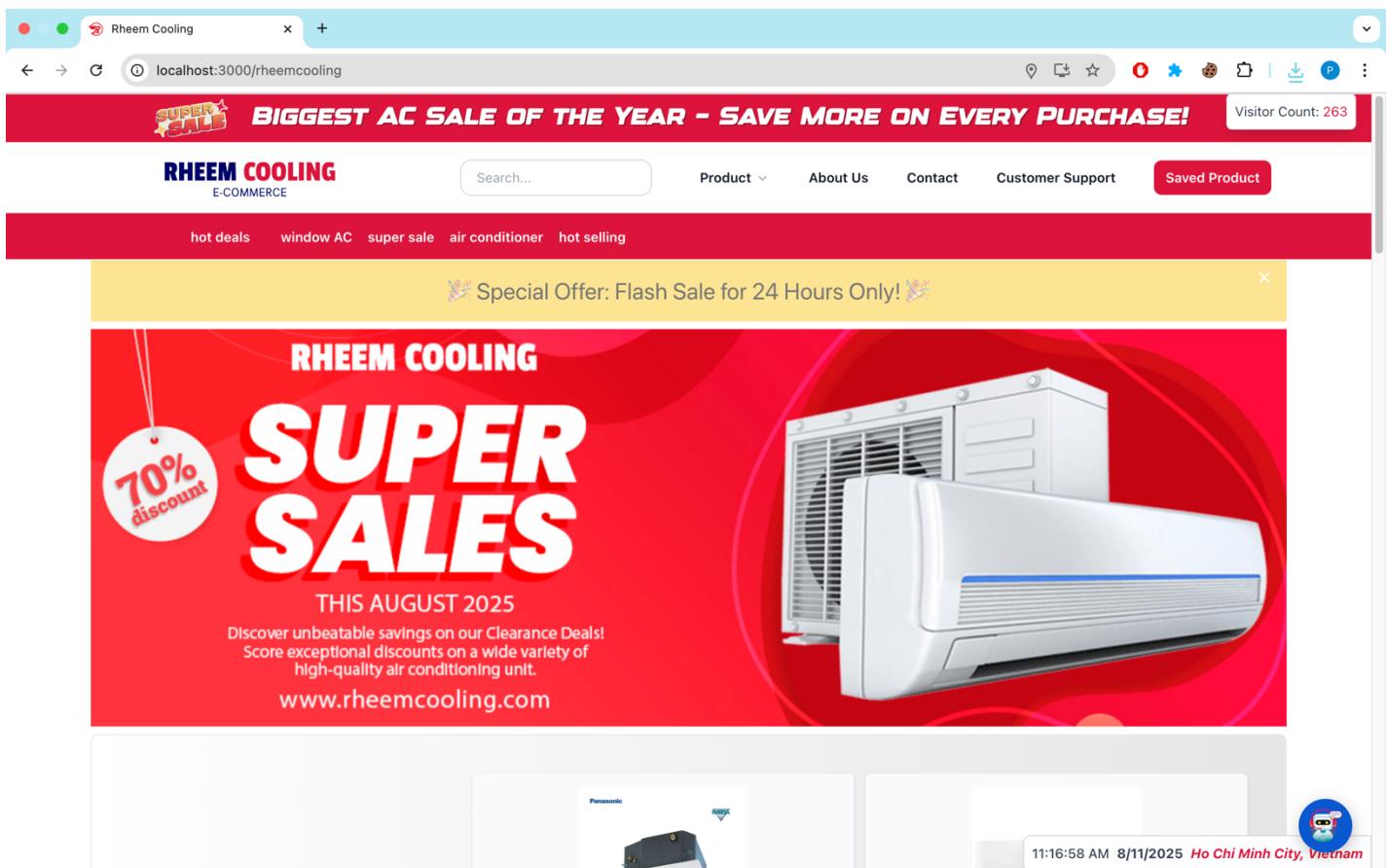
1. Homepage

a. Description

Home page displays following information:

- Banner product promotions
- Top Deals
- Trending product
- Product You Recently View
- Product List

b. Screenshot



Rheem Cooling localhost:3000/rheemcooling Visitor Count: 263

TOP DEALS

Don't miss it

[Shop now](#)



Panasonic - 200 Sq. Ft. 5,000 BTU Portable Air Conditioner

[Shop now](#)



Carrier - 200 Sq. Ft. 12,000 BTU Floor Standing Air Conditioner

[Shop now](#)



Carrier - 100 Sq. Ft. 12,000 BTU Floor Standing Air...

[Shop now](#)



Daikin - 150 Sq. Ft. 9,000 BTU Split Air Conditioner

[Shop now](#)



Toshiba - 150 Sq. Ft. 12,000 BTU Window Air Conditioner

[Shop now](#)



Carrier - 100 Sq. Ft. 5,000 BTU Window Air Conditioner

[Shop now](#)

[Shop deals by category](#)



Window AC



Split AC



Cassette AC

11:17:14 AM 8/11/2025 Ho Chi Minh City, Vietnam

Rheem Cooling localhost:3000/rheemcooling Visitor Count: 263

You Recently Viewed

<
>



Samsung - 100 Sq. Ft. 12,000 BTU...

\$ 380



Samsung - 200 Sq. Ft. 9,000 BTU Split...

\$ 455



Samsung - 150 Sq. Ft. 12,000 BTU Spli...

\$ 313



Samsung - 100 Sq. Ft. 5,000 BTU Floo...

\$ 413

[Clear H](#)

Trending Products



UP TO 50% OFF
PRICE STARTING \$80⁹⁹
TRENDING PRODUCTS
PROMO:BLACKFRIDAY50
THIS WEEKEND ONLY



New - 20%
Carrier - 100 Sq. Ft. 5,000 BTU Window Air Conditioner
\$ 208 \$ 257
★ (4.9) • Sold 816

[Save Product](#)



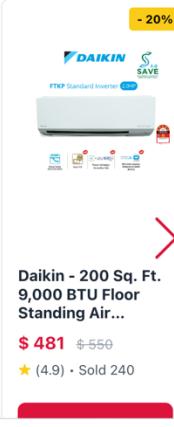
- 20%
Mitsubishi - 200 Sq. Ft. 5,000 BTU Window Air...
\$ 223 \$ 247
★ (4.9) • Sold 713

[Save Product](#)



- 20%
LG - 150 Sq. Ft. 5,000 BTU Window Air Conditioner
\$ 251 \$ 348
★ (4.9) • Sold 891

[Save Product](#)



- 20%
DAIKIN S SAVE
FTK Standard Inverter
Daikin - 200 Sq. Ft. 9,000 BTU Floor Standing Air...
\$ 481 \$ 550
★ (4.9) • Sold 240

[Save Product](#)

11:17:30 AM 8/11/2025 Ho Chi Minh City, Vietnam

Rheem Cooling localhost:3000/rheemcooling

Product Category

	Top Selling AC	Window AC	Split AC	Cassette	Visitor Count: 263						
	Gree - 200 Sq. Ft. 9,000 BTU Split Air Conditioner \$ 296 \$-320 ★ (4.9) • Sold 997		Panasonic - 100 Sq. Ft. 5,000 BTU Split Air... \$ 373 \$-440 ★ (4.9) • Sold 973		LG - 200 Sq. Ft. 12,000 BTU Window Air... \$ 422 \$-449 ★ (4.9) • Sold 942		Toshiba - 150 Sq. Ft. 12,000 BTU Window Air... \$ 189 \$-235 ★ (4.9) • Sold 928		Samsung - 200 Sq. Ft. 9,000 BTU Split Air... \$ 455 \$-494 ★ (4.9) • Sold 925		Mitsubishi - 100 Sq. Ft. 12,000 BTU Split Air... \$ 283 \$-354 ★ (4.9) • Sold 925
	Daikin - 200 Sq. Ft. 9,000 BTU Portable Air... \$ 308 \$-337 ★ (4.9) • Sold 917		Daikin - 150 Sq. Ft. 5,000 BTU Portable Air... \$ 293 \$-365 ★ (4.9) • Sold 904		Samsung - 200 Sq. Ft. 5,000 BTU Window Air... \$ 403 \$-455 ★ (4.9) • Sold 895		Trane - 200 Sq. Ft. 5,000 BTU Portable Air... \$ 191 \$-249 ★ (4.9) • Sold 893		LG - 150 Sq. Ft. 5,000 BTU Window Air... \$ 251 \$-348 ★ (4.9) • Sold 891		Samsung - 100 Sq. Ft. 12,000 BTU Window Air... \$ 380 \$-436 ★ (4.9) • Sold 886

[View more products >>](#)

11:17:41 AM 8/11/2025 Ho Chi Minh City, Vietnam

Rheem Cooling localhost:3000/rheemcooling

	Top Selling AC	Window AC	Split AC	Cassette	Visitor Count: 263						
	Daikin - 200 Sq. Ft. 9,000 BTU Portable Air... \$ 308 \$-337 ★ (4.9) • Sold 917		Daikin - 150 Sq. Ft. 5,000 BTU Portable Air... \$ 293 \$-365 ★ (4.9) • Sold 904		Samsung - 200 Sq. Ft. 5,000 BTU Window Air... \$ 403 \$-455 ★ (4.9) • Sold 895		Trane - 200 Sq. Ft. 5,000 BTU Portable Air... \$ 191 \$-249 ★ (4.9) • Sold 893		LG - 150 Sq. Ft. 5,000 BTU Window Air... \$ 251 \$-348 ★ (4.9) • Sold 891		Samsung - 100 Sq. Ft. 12,000 BTU Window Air... \$ 380 \$-436 ★ (4.9) • Sold 886

[View more products >>](#)

RHEEM COOLING
E-COMMERCE

We provide genuine air conditioning products with competitive prices and dedicated customer service.

About Us

- [Company Information](#)
- [Vision - Mission - Core Values](#)
- [Partners](#)

Hotline

- Call to buy: [1900 232 461](#)
- Complaint: [1800 1063](#)
- Warranty: [1900 232 465](#)

Product

- [Selling AC](#)
- [Window AC](#)
- [Split AC](#)
- [Cassette AC](#)

Customer Support

- [FAQs](#)
- [Sitemap](#)

Connect with Us

Email: support@rheemcooling.com

Subscribe to our newsletter

Get the latest updates and exclusive offers.

[Sign Up](#)

© 2025 Rheem Cooling Company. All rights reserved.

11:17:50 AM 8/11/2025 Ho Chi Minh City, Vietnam

FPT-APTECH – T3.2502.E0 – GROUP 2

25

2. Product

a. Description

This product page displays following information:

- All product that be sold in Rheem Cooling (Window AC, Split AC, Cassette AC)
- Menu filter to easily find items based on categories
- Product can sort by top deal, top selling, price, customer rating, new item.
- When you click name product, it will redirect to product detail page
- And the last, product page have product recently view

b. Screenshot

The screenshot shows a web browser displaying the 'RHEEM COOLING E-COMMERCE' website. The URL in the address bar is 'localhost:3000/rheemcooling/topsellingAC'. The page title is 'AIR CONDITIONER (100)'. On the left, there are filters for Category (Window, Split, Cassette), Brand (Samsung, Daikin, Lg, Panasonic, Mitsubishi, Hitachi, Carrier, Trane, Toshiba, Gree), and Price (Min, Max, Set). The main content area shows four Samsung air conditioner products with their details and prices:

Product Type	Description	Price	Rating
Window AC	Samsung - 100 Sq. Ft. 12,000 BTU Window Air Conditioner	\$ 380	★ (4.9) • Sold 886
Split AC	Samsung - 200 Sq. Ft. 9,000 BTU Split Air Conditioner	\$ 455	★ (4.9) • Sold 925
Split AC	Samsung - 150 Sq. Ft. 12,000 BTU Split Air Conditioner	\$ 313	★ (4.9) • Sold 717
Floor Standing	Samsung - 100 Sq. Ft. 5,000 BTU Floor Standing Air...	\$ 413	★ (4.9) • Sold 650

The page also includes a 'Sort By' dropdown set to 'Customer Rating', a timestamp '11:35:28 AM 8/11/2025', and a location 'Ho Chi Minh City, Vietnam'.

Rheem Cooling localhost:3000/rheemcooling/topsellingAC

Power

- 1HP
- 1.5HP

New - 20% **New** - 20% **New** - 20% **New** - 20% Visitor Count: 263

Samsung - 200 Sq. Ft. 12,000 BTU Portable Air Conditioner \$ 314 \$369 ★ (4.9) • Sold 836	Samsung - 100 Sq. Ft. 5,000 BTU Split Air Conditioner \$ 190 \$280 ★ (4.9) • Sold 838	Samsung - 150 Sq. Ft. 5,000 BTU Window Air Conditioner \$ 359 \$428 ★ (4.9) • Sold 604	Samsung - 100 Sq. Ft. 9,000 BTU Window Air Conditioner \$ 387 \$458 ★ (4.9) • Sold 219
Save Product	Save Product	Save Product	Save Product

[View more 92 Air Conditioner ↓](#)

Recently Viewed



Rheem Cooling localhost:3000/rheemcooling/topsellingAC

Visitor Count: 263

Recently Viewed

-6% Daikin - 200 Sq. Ft. 9,000 BTU Portable Air Conditioner \$308 \$337	-19% Gree - 200 Sq. Ft. 9,000 BTU Split Air Conditioner \$296 \$328	-9% Panasonic - 100 Sq. Ft. 5,000 BTU Window Air Conditioner \$373 \$440	-15% Gree - 150 Sq. Ft. 9,000 BTU Window Air Conditioner \$398 \$448	-25% Trane - 200 Sq. Ft. 5,000 BTU Split Air Conditioner \$269 \$366
Save Product	Save Product	Save Product	Save Product	Save Product

RHEEM COOLING
E-COMMERCE

We provide genuine air conditioning products with competitive prices and

About Us

- Company Information
- Vision - Mission - Core Values
- Partners

Product

- Selling AC
- Window AC
- Split AC
- Cassette AC

Customer Support

- FAQs
- Sitemap

Connect with Us

- [Facebook](#)
- [Instagram](#)
- [Twitter](#)
- [YouTube](#)

Email: support@rheemcooling.com

11:36:13 AM 8/11/2025 Ho Chi Minh City, Vietnam

3. Product Details

a. Description

Product Details Page displays following information:

- The **Product Detail** page provides comprehensive information about a selected item, including high-quality images, specifications, pricing, and customer reviews.
- The **Image Gallery** component allows users to browse multiple product images for a closer look at details (5 images per product)
- The **Description** section displays the product's key features, specifications, and overall summary, along with the average review score, and have a **Save Product button** to save product into Saved Product list
- The **Product Info** section highlights technical details, available variations, and purchase options, it has button **See more information details** to open pop-up with full information about this product
- The **Review** section contains customer feedback, ratings, and comments to help buyers make informed decisions.
- The **Recent View** component shows products the user has recently viewed for easy navigation back to previously explored items.
- The **Consider** section suggests related products and displays reviews for those recommendations, helping customers discover similar options.

b. Screenshot

The screenshot shows a web browser displaying the Rheem Cooling e-commerce website. The URL in the address bar is `localhost:3000/rheemcooling/product/daikin_ac10`. The page features a red header with the text "BIGGEST AC SALE OF THE YEAR - SAVE MORE ON EVERY PURCHASE!" and "Visitor Count: 263". Below the header, there's a navigation bar with links for "Product", "About Us", "Contact", "Customer Support", and "Saved Product". A search bar is also present. The main content area displays a product detail page for a "Daikin - 200 Sq. Ft. 9,000 BTU Portable Air Conditioner". The product image shows a white Daikin portable AC unit. To the left of the main image are five smaller thumbnail images of different AC models. To the right, there are buttons for "Fast \$15", "Normal \$12", and "Saving \$9". A "Save Product" button is located at the bottom of this section. Below this, a "Standard Geek Squad Protection" section is shown with a timestamp "11:51:15 AM 8/11/2025 Ho Chi Minh City, Vietnam" and a small Geek Squad logo. At the very bottom, there's a "Product Information" section with a table containing columns for "Name", "Description", and "Price".

Rheem Cooling localhost:3000/rheemcooling/product/daikin_ac10 Visitor Count: 263

Product Information

Name	Daikin - 200 Sq. Ft. 9,000 BTU Portable Air Conditioner
Brand	daikin
Type	Split
Power	1.0 HP (9000 BTU)

[View more product details →](#)

Standard Geek Squad Protection

No Plan
2-Year \$24.99 About \$1.04/mo
4-Year \$49.99 About \$1.04/mo

Review / Rating

★ 4.0
10 reviews
✓ 100% would recommend to a friend

5 ★ 4
4 ★ 2
3 ★ 4
2 ★ 0
1 ★ 0

Jordan Maldonado
2025-06-24
Could Be Better
★★★★☆
Cools my room within minutes, and runs very quietly. Love it!
39

Christine Hernandez
2025-06-25
Amazing Cooling Power
★★★★★
Sometimes makes a bit of noise but nothing too bad

11:51:24 AM 8/11/2025 Ho Chi Minh City, Vietnam

Rheem Cooling localhost:3000/rheemcooling/product/daikin_ac10 Visitor Count: 263

Product Information

Name	Daikin - 200 Sq. Ft. 9,000 BTU Portable Air Conditioner
Brand	daikin
Type	Split
Power	1.0 HP (9000 BTU)

[View more product details →](#)

Standard Geek Squad Protection

2-Year \$24.99 About \$1.04/mo
4-Year \$49.99 About \$1.04/mo

Product Information

Product Information

Name	Daikin - 200 Sq. Ft. 9,000 BTU Portable Air Conditioner
Brand	daikin
Type	Split
Power	1.0 HP (9000 BTU)
Size	800 x 250 x 200 mm

Review / Rating

★ 4.0
10 reviews
✓ 100% would recommend to a friend

5 ★ 4
4 ★ 2
3 ★ 4
2 ★ 0
1 ★ 0

Amazing Cooling Power
★★★★★
Sometimes makes a bit of noise but nothing too bad

11:51:39 AM 8/11/2025 Ho Chi Minh City, Vietnam

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Rheem Cooling

localhost:3000/rheemcooling/product/daikin_ac10

Product Information

Name	Daikin - 200 S
Brand	daikin
Type	Split
Power	1.0 HP (9000 BTU)

[View more product details →](#)

Review / Rating

4.0     

10 reviews

✓ 100% would recommend to a friend

Standard Geek Squad Protection

Product Information

Name	Daikin - 200 Sq. Ft. 9,000 BTU Portable Air Conditioner
Brand	daikin
Type	Split
Power	1.0 HP (9000 BTU)
Size	800 x 250 x 200 mm
Weight	9.5 kg
Color	Gray
Warranty	1 year on-site warranty
Made in	Malaysia
Install Fee	\$23
In Stock	30
Sold	917
Description	Durable build with multi-stage air filtration and stable performance.

2-Year \$24.99
about \$1.04/mo

4-Year \$49.99
about \$1.04/mo

Amazing Cooling Power

Sometimes makes a bit of noise but nothing too bad.

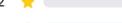
11:51:47 AM 8/11/2025 Ho Chi Minh City, Vietnam

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Rheem Cooling

localhost:3000/rheemcooling/product/daikin_ac10

Review / Rating

4.0     

10 reviews

✓ 100% would recommend to a friend

Visitor Count: 264

Jordan Maldonado 2025-06-24
Could Be Better

Cools my room within minutes, and runs very quietly. Love it!
 

Christine Hernandez 2025-06-25
Amazing Cooling Power

Sometimes makes a bit of noise but nothing too bad.
 

Peter Pearson 2025-06-22
Energy Saving and Reliable

Installation was smooth, and performance is better than expected.
 

Anonymous

Hi! How are you doing?  11:52:21 AM 8/11/2025 Ho Chi Minh City, Vietnam

Recently Viewed



-6%

Daikin - 200 Sq. Ft.
9,000 BTU Portable Air Conditioner

\$308 \$337

[Save Product](#)



-19%

Gree - 200 Sq. Ft. 9,000 BTU Split Air Conditioner

\$296 \$328

[Save Product](#)



Panasonic
1.5HP

-9%

Panasonic - 100 Sq. Ft.
5,000 BTU Split Air Conditioner

\$373 \$440

[Save Product](#)



-15%

Gree - 150 Sq. Ft. 9,000 BTU Window Air Conditioner

\$398 \$448

[Save Product](#)



TRANE

-25%

NOTE: THE PICTURE IS FOR REFERENCE PURPOSES ONLY.
SOME MODELS ARE SHOWN.

Trane - 200 Sq. Ft. 5,000 BTU Split Air Conditioner

\$269 \$366

[Save Product](#)

• • • • •

More to consider

11:52:40 AM 8/11/2025 Ho Chi Minh City, Vietnam

More to consider

New



-15%

Daikin - 150 Sq. Ft.
5,000 BTU Portable Air Conditioner

★ 4.2 | Sold 904

\$293 \$365

[Save Product](#)



-21%

Daikin - 100 Sq. Ft.
9,000 BTU Portable Air Conditioner

★ 4.1 | Sold 883

\$210 \$306

[Save Product](#)



-22%

Daikin - 200 Sq. Ft.
12,000 BTU Split Air Conditioner

★ 3.7 | Sold 442

\$346 \$373

[Save Product](#)



DAIKIN

-21%

Daikin - 150 Sq. Ft.
9,000 BTU Split Air Conditioner

★ 3.9 | Sold 530

\$167 \$247

[Save Product](#)



COANDA

New

-14%

Daikin - 100 Sq. Ft.
9,000 BTU Portable Air Conditioner

★ 4.0 | Sold 777

\$224 \$299

[Save Product](#)

• • • • •

RHEEM COOLING
E-COMMERCE

We provide genuine air conditioning

About Us

- [Company Information](#)
- [Vision - Mission - Core Values](#)
- [Partners](#)

Product

- [Selling AC](#)
- [Window AC](#)
- [Split AC](#)

Customer Support

- [FAQs](#)
- [Sitemap](#)

Connect with Us

Email: support@rheemcooling.com

4. About Us

a. Description

The **About Us** page introduces the company's background, values, and strategic direction through engaging content and smooth animations:

- The **Company Info** section presents the organization's history, achievements, and core business areas.
- The **Mission** section highlights the company's vision, mission statement, and commitment to delivering value to customers and stakeholders.
- The **Partners** section showcases leading air conditioner brands available in our store, reinforcing product diversity and trusted quality.
- Subtle **scroll animations** powered by AOS enhance the user experience, making the page more dynamic and visually appealing.
- Background patterns and clean layouts maintain a professional yet modern design.

b. Screenshot

The screenshot shows a web browser window for 'Rheem Cooling' at 'localhost:3000/rheemcooling/aboutus'. The header features a red banner with 'SUPER SALE' and 'BIGGEST AC SALE OF THE YEAR - SAVE MORE ON EVERY PURCHASE!' followed by a visitor count of '264'. Below the banner is a navigation bar with 'RHEEM COOLING E-COMMERCE', a search bar, and links for 'Product', 'About Us', 'Contact', 'Customer Support', and a 'Saved Product' button. A secondary navigation bar below includes 'hot deals', 'window AC', 'super sale', 'air conditioner', and 'hot selling'. The main content area has a grid pattern background and contains a section titled 'Company Information'. It includes a bio about the company's history and global reach, and a bio about their engineering and service team. To the right of the text are four images: a smiling man in a white shirt standing next to a red car, a red service van parked outdoors, and a view of an industrial or construction site. A timestamp '12:01:53 PM 8/11/2025 Ho Chi Minh City, Vietnam' is at the bottom right, along with a small robot icon.

Rheem Cooling localhost:3000/rheemcooling/aboutus Visitor Count: 264

Our Mission

At **Rheem Cooling**, our mission is to make sustainable comfort accessible for everyone.

We strive to:

- Develop environmentally friendly and energy-saving cooling technologies.
- Deliver outstanding customer service from purchase to long-term maintenance.
- Build long-term relationships with our clients through trust and reliability.

We're not just cooling your spaces—we're improving your lifestyle.

30%
Energy Reduction by 2030

50 Countries
will have our branch by 2028

98%
Customer Satisfaction



12:02:09 PM 8/11/2025 Ho Chi Minh City, Vietnam

Rheem Cooling localhost:3000/rheemcooling/aboutus Visitor Count: 264

Partners

SAMSUNG DAIKIN LG PANASONIC MITSUBISHI ELECTRIC HITACHI

RHEEM COOLING
E-COMMERCE

We provide genuine air conditioning products with competitive prices and dedicated customer service.

About Us

- Company Information
- Vision - Mission - Core Values
- Partners

Product

- Selling AC
- Window AC
- Split AC
- Cassette AC

Customer Support

- FAQs
- Sitemap

Connect with Us

[f](#) [i](#) [t](#) [y](#)

Email: support@rheemcooling.com

Subscribe to our newsletter

Get the latest updates and exclusive offers.

Enter your email Sign Up

© 2025 Rheem Cooling Company. All rights reserved.

12:02:25 PM 8/11/2025 Ho Chi Minh City, Vietnam

5. Contact

a. Description

The **Contact** page provides multiple ways for customers to connect with the company, combining location details, social channels, and a direct inquiry form.

- The **Social Media** section lists the company's official social platforms, enabling customers to stay updated on news, promotions, and events.
- The **Branches** section displays all store locations with addresses, maps, and contact details, making it easy for customers to find a nearby branch.
- The **Contact Form** allows visitors to send inquiries, feedback, or requests directly to the company, ensuring quick communication.
- Smooth **scroll animations** powered by AOS create an engaging and interactive browsing experience.
- The layout is designed with a responsive grid, ensuring the page looks clean and accessible on all devices.

b. Screenshot

The screenshot shows a web browser displaying the 'Rheem Cooling' website at localhost:3000/rheemcooling/contact. The page features a red header with a 'SUPER SALE' banner and a 'BIGGEST AC SALE OF THE YEAR - SAVE MORE ON EVERY PURCHASE!' message. Below the banner, the 'RHEEM COOLING E-COMMERCE' logo is visible, along with a search bar and navigation links for Product, About Us, Contact, Customer Support, and Saved Product. The main content area is divided into two sections: 'Connect with us' on the left and 'Contact Us' on the right. The 'Connect with us' section includes a brief description, social media links for Facebook, Image gallery, News, and Video review, and a 'Our Branches' section with a dropdown menu showing 'Australia – 2-10 Captain Cook Dr, Caringbah NSW 2229'. The 'Contact Us' section has a form for entering First name, Last name, Email, Phone number, Title, and Message, with a timestamp of '12:07:26 PM 8/11/2025 Ho Chi Minh City, Vietnam' and a small map icon.

Connect with us

Follow us for the latest promotions, product tips, and customer support. We're always here to help!

- [f \(Support + Sell\)](#)
- [g \(Image gallery + feedback\)](#)
- [t \(News\)](#)
- [v \(Video review + tutorial\)](#)

Our Branches

Trusted service, wherever you are

Australia – 2-10 Captain Cook Dr, Caringbah NSW 2229

Country: Australia
Address: 2-10 Captain Cook Dr, Caringbah NSW 2229
Email: john45@cruz.org
Phone: (592)796-9746



Contact Us

We'd love to hear from you!

First name *
First name is required!

Last name *
Last name is required!

Email *
Email is required!

Phone number *
Phone number is required!

Title *
Title is required!

Message *
Message is required!

I agree to the [privacy policy](#).
You must agree to the privacy policy.

Send us

Visitor Count: 264

12:07:52 PM 8/11/2025 Ho Chi Minh City, Vietnam

Our Branches

Trusted service, wherever you are

Australia – 2-10 Captain Cook Dr, Caringbah NSW 2229

Country: Australia
Address: 2-10 Captain Cook Dr, Caringbah NSW 2229
Email: john45@cruz.org
Phone: (592)796-9746



Contact Us

Phone number *

Visitor Count: 264

Title *

Message *

I agree to the [privacy policy](#).

Send us

Customer Support

12:08:22 PM 8/11/2025 Ho Chi Minh City, Vietnam

Rheem Cooling +

localhost:3000/rheemcooling/contact

Our Branches

Trusted service, wherever you are

- ✓ Australia – 2-10 Captain Cook Dr, Caringbah NSW 2229
- Germany – Wittestraße 30, 13509 Berlin
- Singapore – 1 Kallang Sector, Singapore 349276
- United Kingdom – 3 St James's Square, London SW1Y 4JU
- Gambia – Trans-Gambia Hwy, Soma, Gambia
- Algeria – 436G+CP6, Tebesbest, Algeria
- Mexico – 98440 Zacatecas, Mexico
- USA – 1600 Amphitheatre Parkway, Mountain View, CA
- USA – 1 Infinite Loop, Cupertino, CA
- USA – 350 5th Ave, New York, NY 10118
- USA – 4059 Mt Lee Dr, Hollywood, CA 90068
- USA – 2211 North First Street, San Jose, CA 95131

Phone number *
US 123-456-7890

Title *

Message *

I agree to the [privacy policy](#)

Send us

About Us Product Customer Support

12:08:29 PM 8/11/2025 Ho Chi Minh City, Vietnam

6. Customer Support

a. Description

The **Customer Support** page helps customers quickly find answers to common questions through an organized FAQ section.

- The FAQ content is divided into clear **categories**: *Product & Features*, *Service & Warranty*, and *Usage & Buying Tips*, allowing users to filter questions easily.
- Each FAQ item can be expanded or collapsed using an **accordion interface**, ensuring a clean and focused browsing experience.
- Customers who cannot find their answer are prompted to **contact the support team** via a direct link to the Contact page.
- Smooth animations and category tabs make navigation intuitive and visually engaging.
- The layout is responsive, ensuring accessibility on desktop, tablet, and mobile devices.

b. Screenshot

Frequently Asked Questions

Have a different question and can't find the answer you're looking for? Reach out to our support team [by sending us an email](#) and we'll get back to you as soon as we can.

[Product & Features](#) [Service & Warranty](#) [Usage & Buying Tips](#)

Are Rheem air conditioners energy-efficient?
Yes, Rheem units feature inverter technology to maximize energy savings.

Do the air conditioners have air purification features?

Do Rheem air conditioners support smart control?

What is the noise level of Rheem AC units?

12:14:22 PM 8/11/2025 Ho Chi Minh City, Vietnam

The screenshot shows a web browser window for 'Rheem Cooling' at 'localhost:3000/rheemcooling/customersupport'. The page features a red header with a 'SUPER SALE' banner and the text 'BIGGEST AC SALE OF THE YEAR - SAVE MORE ON EVERY PURCHASE!'. Below the header is a navigation bar with links for 'RHEEM COOLING E-COMMERCE', 'Search...', 'Product', 'About Us', 'Contact', 'Customer Support', and 'Saved Product'. A red menu bar below the navigation bar includes links for 'hot deals', 'window AC', 'super sale', 'air conditioner', and 'hot selling'. The main content area is titled 'Frequently Asked Questions' and contains a message: 'Have a different question and can't find the answer you're looking for? Reach out to our support team [by sending us an email](#) and we'll get back to you as soon as we can.' Below this message are four expandable FAQ sections: 'How long is the warranty period?', 'Do you offer installation services?', 'How often should I service my AC?', and 'Can I return or exchange a product?'. A timestamp '12:14:31 PM 8/11/2025 Ho Chi Minh City, Vietnam' is visible in the bottom right corner.

This screenshot is identical to the one above, showing the same website layout and content for the 'Frequently Asked Questions' section. It includes the same header, navigation bar, menu bar, and FAQ list. The timestamp '12:14:31 PM 8/11/2025 Ho Chi Minh City, Vietnam' is also present in the bottom right corner.

Rheem Cooling

localhost:3000/rheemcooling/customersupport

- Is it okay to set the AC temperature below 20°C?
- Does turning the AC on and off frequently waste energy?
- How often should I clean or service my air conditioner?
- Can I install the indoor unit near a window?
- How can I use the remote to save power?
- Do air conditioners help eliminate odors?
- How do I check the model of my AC unit?

RHEEM COOLING
E-COMMERCE

We provide genuine air conditioning products with competitive prices and

About Us	Product	Customer Support
Company Information	Selling AC	FAQs
Vision - Mission - Core Values	Window AC	Sitemap
Partners	Split AC	
	Cassette AC	

Connect with Us

f i t y

Email: support@rheemcooling.com

12:14:52 PM 8/11/2025 Ho Chi Minh City, Vietnam

7. Saved Product

a. Description

The **Saved Product** section displays a list of products that the user has recently viewed or saved for quick reference.

- Products are loaded from **localStorage**, ensuring users' saved items persist across sessions.
- Each product card shows the product image, name, average rating based on reviews, total sales, and current price alongside the original price if discounted.
- Users can **remove** any saved product individually with a confirmation prompt to avoid accidental deletions.
- Product names link directly to their detailed product pages for easy navigation.
- The layout is responsive, presenting multiple items in a grid that adjusts from mobile to desktop views.
- When no products are saved, a clear message informs the user with a visually distinct notice.

b. Screenshot

Product Name	Rating	Sold Count	Original Price	Current Price
Carrier - 200 Sq. Ft. 12,000 BTU Window Air Conditioner	4.4	487	\$291	\$191
Carrier - 100 Sq. Ft. 12,000 BTU Floor Standing Air Conditioner	3.7	412	\$348	\$303
Carrier - 200 Sq. Ft. 12,000 BTU Floor Standing Air Conditioner	3.4	665	\$468	\$447
Carrier - 100 Sq. Ft. 12,000 BTU Floor Standing Air Conditioner	4.0	437	\$291	\$248

Rheem Cooling localhost:3000/rheemcooling/savedproduct

localhost:3000 says
Are you sure to delete ?
Cancel OK

Visitor Count: 264

Contact Customer Support Saved Product

Saved Product



Carrier - 200 Sq. Ft. 12,000 BTU Window Air Conditioner
★ 4.4 | Sold 487
\$191 \$281



Carrier - 100 Sq. Ft. 12,000 BTU Floor Standing Air Conditioner
★ 3.7 | Sold 412
\$303 \$348



Carrier - 200 Sq. Ft. 12,000 BTU Floor Standing Air Conditioner
★ 3.4 | Sold 665
\$447 \$468



Carrier - 100 Sq. Ft. 12,000 BTU Floor Standing Air Conditioner
★ 4.0 | Sold 437
\$248 \$291

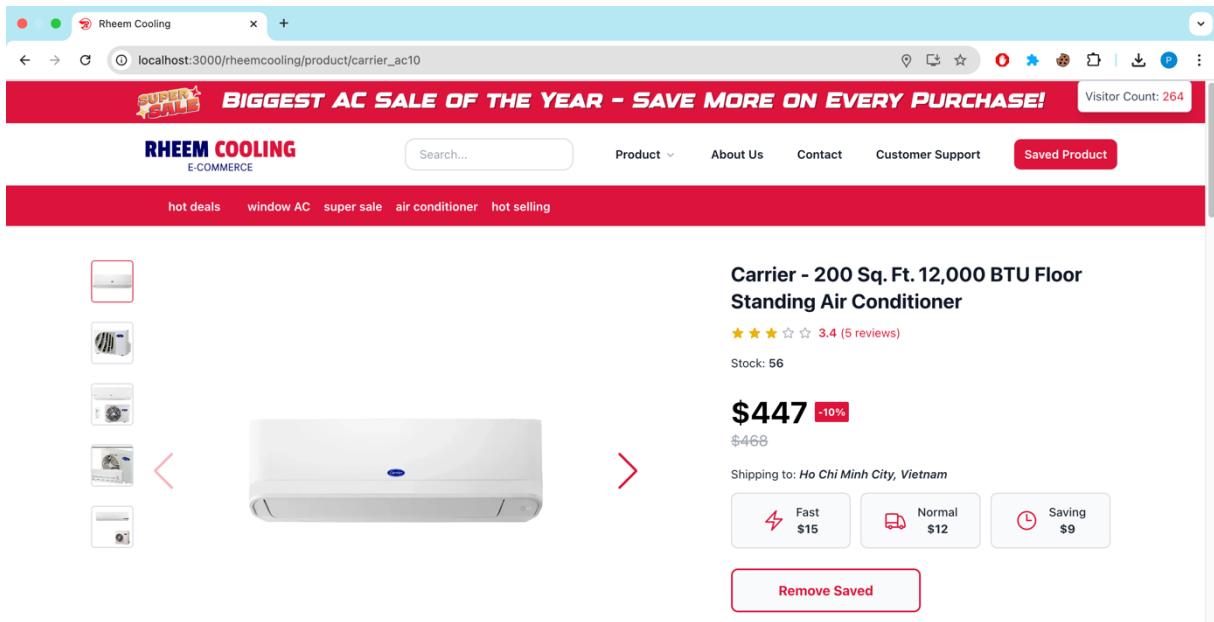
About Us Product Customer Support 12:22:17 PM 8/11/2025 Ho Chi Minh City, Vietnam

RESPONSIVE

The application is designed with **responsive layouts** to ensure optimal user experience across a wide range of devices, including desktops, tablets, and smartphones.

- Components use **CSS Grid** and **Flexbox** with utility-first frameworks like **Tailwind CSS** to create flexible and adaptive layouts that automatically adjust based on screen size.
- Media queries and responsive classes handle changes such as column spans, padding, margins, font sizes, and element visibility, providing a seamless and readable interface on smaller or larger screens.
- Navigation elements, such as menus and buttons, transform into mobile-friendly versions (e.g., hamburger menus, collapsible accordions) for easy interaction on touch devices.
- Images and galleries are styled with object-fit and dynamic sizing to maintain aspect ratios and prevent overflow or distortion on different screen widths.
- Interactive elements and animations are optimized to maintain performance and usability without overwhelming smaller or less powerful devices.
- Overall, the design philosophy ensures that no matter what device the user accesses the website from, the content remains accessible, visually appealing, and easy to navigate.

Desktop:



Tablet:

RHEEM COOLING
E-COMMERCE

**Carrier - 200 Sq. Ft.
12,000 BTU Floor
Standing Air
Conditioner**

3.4 (5 reviews)

Stock: 56

\$447 -10%
\$468

Shipping to: Ho Chi Minh City,
Vietnam

Fast \$15 **Normal \$12** **Saving \$9**

Remove Saved

Items are covered under [Rheem Cooling Promise](#)

Product Information

Name	Carrier - 200 Sq. Ft. 12,000 BTU Floor Standing Air Conditioner
Brand	carrier
Type	Window
Power	1.5 HP (12000 BTU)

[View more product details →](#)

Standard Geek Squad Protection

No Plan	2-Year \$24.99 About \$1.04/mo
4-Year \$49.99 About \$1.04/mo	

12:31:50 PM 8/11/2025 Ho Chi Minh City, Vietnam



Smart Phone:

The image shows a smartphone displaying a mobile version of the Rheem Cooling e-commerce website. At the top, there is a red banner with the text "BIGGEST AC SALE OF THE YEAR - SAVE MORE ON EVERY PURCHASE!" Below the banner, the "RHEEM COOLING" logo is displayed, followed by "E-COMMERCE". On the right side of the header is a menu icon consisting of three horizontal lines. The main content area features a large image of a white Carrier floor-standing air conditioner unit. Below the image, the product title is shown in bold black text: "Carrier - 200 Sq. Ft. 12,000 BTU Floor Standing Air Conditioner". Underneath the title is a rating section showing 5 yellow stars and the text "3.4 (5 reviews)". The price is prominently displayed as "\$447" in large black text, with a red "-10%" discount badge next to it. Below the price, the original price "\$468" is shown in gray. The text "Stock: 56" indicates availability. At the bottom of the screen, there are three shipping options: "Fast \$15" (indicated by a lightning bolt icon), "Normal \$12" (indicated by a truck icon), and a third option partially visible on the right. The text "Shipping to: Ho Chi Minh City, Vietnam" is also present.

JAVASCRIPT DESCRIPTION

1. Slide, Carousel Product Card

a. Description

The project leverages **Swiper.js**, a modern and flexible slider library, to build responsive and interactive product carousels. Swiper.js provides key features that enhance the user experience when browsing products, including:

- **Responsive design** with configurable breakpoints, allowing the carousel to show varying numbers of slides depending on screen size (e.g., 1 slide on mobile, 3 on tablets, 5 on desktop).
- **Customizable navigation controls**, such as previous/next buttons that can be styled and positioned freely, giving users intuitive manual control over sliding.
- **Clickable pagination indicators** that display the current slide position and enable quick navigation across slides.
- **Smooth sliding animations** optimized for performance with hardware acceleration and fluid transitions.
- **Modular architecture**, allowing the selective import of features like navigation and pagination to keep bundle size minimal and improve loading speed.
- **Seamless React integration**, enabling dynamic rendering of slides based on data and smooth handling of user interactions within React's component lifecycle.

By using Swiper.js, the product carousels across the application deliver an engaging, visually appealing, and user-friendly browsing experience on all device types.

b. Code Javascript (using code in Consider component)

```

1 // Consider.jsx
2
3 import { Swiper, SwiperSlide } from 'swiper/react';
4 import { Navigation, Pagination } from 'swiper/modules';
5 import { StarIcon as StarSolid } from "@heroicons/react/20/solid";
6 import 'swiper/css';
7 import 'swiper/css/navigation';
8 import 'swiper/css/pagination';
9 import { Link } from 'react-router-dom';
10 import SaveButton from '../../components/SaveButton'; //
11
12 function Consider({ data, review }) {
13   return (
14     <div className="partners px-4 md:px-0">
15       <h2 className="text-2xl font-bold text-gray-800 mb-3">More to consider</h2>
16
17       <div className="relative w-full px-6 py-10">
18
19         {/* Custom Navigation Buttons */}
20         <div
21           className="hidden md:block absolute z-10 top-1/2 left-0 -translate-y-1/2 cursor-pointer bg-white border
22             border-gray-300 rounded-full p-2 shadow hover:bg-red-100 transition"
23             id="custom-prev-consider"
24           >
25             <svg xmlns="http://www.w3.org/2000/svg" className="h-6 w-6 text-red-600" fill="none" viewBox="0 0 24 24"
26               stroke="currentColor">
27               <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M15 19l-7-7 7-7" />
28             </svg>
29           </div>
30
31         <div
32           className="hidden md:block absolute z-10 top-1/2 right-0 -translate-y-1/2 cursor-pointer bg-white border
33             border-gray-300 rounded-full p-2 shadow hover:bg-red-100 transition"
34             id="custom-next-consider"
35           >
36             <svg xmlns="http://www.w3.org/2000/svg" className="h-6 w-6 text-red-600" fill="none" viewBox="0 0 24 24"
37               stroke="currentColor">
38               <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M9 51l 7-7 7-7" />
39             </svg>
40           </div>
41
42         <Swiper
43           modules={[Navigation, Pagination]}
44           spaceBetween={20}
45           navigation={{
46             prevEl: '#custom-prev-consider',
47             nextEl: '#custom-next-consider',
48           }}
49           pagination={{
50             clickable: true,
51             el: '.custom-swiper-pagination-consider',
52           }}
53           breakpoints={{
54             0: {
55               slidesPerView: 1,
56             },
57             768: {
58               slidesPerView: 3,
59             },
60             1024: {
61               slidesPerView: 5,
62             },
63           }
64           className="mySwiper"
65         >
66           {data.map((product) => {
67             const productReviews = review.filter(r => r.product_id === product.ac_id);
68             const totalReviews = productReviews.length;
69             const avgRating = totalReviews
70               ? (productReviews.reduce((sum, r) => sum + r.rating, 0) / totalReviews).toFixed(1)
71               : 0;
72
73             return (
74               <SwiperSlide key={product.ac_id} className="p-6 border border-gray-200 rounded-md hover:shadow-sm">
75                 <div className="flex justify-between">
76                   <p className={`px-2 font-bold rounded ${product.isNew ? 'bg-[#DC143C] text-white text-sm' :
77                     'invisible text-sm'}`}>
78                     New
79                   </p>
80                   {product.sale ? (
81                     <p className="bg-yellow-400 text-black text-sm px-2 font-bold rounded">
82                       -{product.sale}%
83                     </p>
84                   ) : null}
85                 </div>
86               </SwiperSlide>
87             )
88           )}
89         </Swiper>
90       </div>
91     </div>
92   )
93 }

```

```

76 |           <p className="bg-yellow-400 text-black text-sm px-2 font-bold rounded">
77 |             -{product.sale}%
78 |           </p>
79 |         ) : (
80 |           <span />
81 |         )
82 |       </div>
83 |
84 |       <div className="w-full h-[180px] md:h-[150px] flex justify-center items-center">
85 |         <img
86 |           src={`https://storage.googleapis.com/rheemcooling/${product.brand}/${product.ac_id}/${product.ac_id}_img1.webp`}
87 |           alt={product.name}
88 |           className="max-h-full object-contain"
89 |         />
90 |       </div>
91 |
92 |
93 |       <Link
94 |         to={`/product/${product.ac_id}`}
95 |         className="font-bold text-sm line-clamp-3 mt-5 hover:underline min-h-[3.6em] leading-[1.2em]"
96 |       >
97 |         {product.name}
98 |       </Link>
99 |
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
    </SwiperSlide>
  );
}
</Swiper>
<div className="custom-swiper-pagination-consider flex justify-center mt-6"></div>
</div>
</div>
);
};

export default Consider;

```

2. Geolocation

a. Description

- The **YourLocation** component utilizes the browser's **Geolocation API** to obtain the user's current geographic coordinates (latitude and longitude) after permission is granted.
- It then performs a **reverse geocoding** request using the OpenStreetMap Nominatim API to translate coordinates into a human-readable location, displaying the city (or town/village) and country.
- The component updates and displays the **current local time and date**, refreshing every second to keep the clock accurate in real-time.
- It handles possible errors and fallback states gracefully, such as denied permissions, unsupported browsers, or API failures, showing appropriate messages like "Permission denied" or "Location unavailable."
- The location and time display is styled as a fixed bottom-right widget, visible only on medium and larger screens, providing users with contextual temporal and geographic information while browsing.
- This enhances user experience by personalizing content and offering useful real-time info without interfering with main page content.

b. Code (using code in YourLocation.jsx)

```
import { useEffect, useState } from "react";

function YourLocation() {
  const [time, setTime] = useState(new Date());
  const [location, setLocation] = useState("Loading...");

  useEffect(() => {
    // Cập nhật giờ mỗi giây
    const timer = setInterval(() => {
      setTime(new Date());
    }, 1000);

    // Lấy vị trí người dùng
    if (navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(
        async (position) => {
          const { latitude, longitude } = position.coords;

          try {
            const res = await
fetch(`https://nominatim.openstreetmap.org/reverse?format=jsonv2&lat=${latitude}&lon=${longitude}`);
            const data = await res.json();
            const city = data.address.city || data.address.town || data.address.village ||
"";
            const country = data.address.country || "";
            setLocation(`${city}, ${country}`);
          } catch (error) {
            setLocation("Location unavailable");
          }
        }
      );
    }
  });
}

export default YourLocation;
```

```

        }
    },
    () => setLocation("Permission denied")
);
} else {
    setLocation("Geolocation not supported");
}

return () => clearInterval(timer);
}, []);
}

return (
    <div className="hidden md:block fixed bottom-0 right-0 bg-white border border-gray-300 rounded-l-lg shadow-lg text-sm p-2 text-gray-700 z-50 md:flex gap-2 items-center">
        <div className="font-medium">
            {time.toLocaleTimeString([], { hour: '2-digit', minute: '2-digit', second: '2-digit'})}
        </div>
        <div className="font-bold">
            {time.toLocaleDateString()}
        </div>
        <div className="italic font-bold text-[#DC143C]">{location}</div>
    </div>
);
}

export default YourLocation;

```

3. Store Data in LocalStorage

a. Description

- The application utilizes the browser's **localStorage** to persist user-specific data such as recently viewed products or saved/favorite items.
- Data stored in localStorage ensures that user preferences and interactions are retained across page reloads and browser sessions without requiring user authentication.
- Components read from localStorage during initialization to populate state, enabling dynamic rendering of personalized content.
- Updates to the user's saved or viewed items are synchronized back to localStorage to maintain consistency between the UI and stored data.
- This client-side storage approach enhances user experience by providing quick access to previously interacted products and enabling seamless navigation without server dependency.
- Proper error handling is implemented to handle cases where localStorage is unavailable or contains corrupted data.

b. Code (using code in SaveButton.jsx and SavedProduct.jsx)

```
//components/SaveButton.jsx
import { useEffect, useState } from "react";
import { createPortal } from "react-dom";
import { HeartIcon } from "@heroicons/react/20/solid";
import "../css/style.css";

function SaveButton({ product, width, isHeart = false }) {
  const [isSaved, setIsSaved] = useState(false);
  const [toast, setToast] = useState({ show: false, message: "" });

  useEffect(() => {
    const saved = JSON.parse(localStorage.getItem("saved_product")) || [];
    const existed = saved.find(s => s.ac_id === product.ac_id);
    setIsSaved(!existed);
  }, [product]);

  const showToast = (message) => {
    setToast({ show: true, message });
    setTimeout(() => {
      setToast({ show: false, message: "" });
    }, 3000);
  };

  const handleSave = () => {
    const saved = JSON.parse(localStorage.getItem("saved_product")) || [];
    const existed = saved.find(s => s.ac_id === product.ac_id);

    if (existed) {
      const updated = saved.filter(s => s.ac_id !== product.ac_id);
      localStorage.setItem("saved_product", JSON.stringify(updated));
      setIsSaved(false);
    } else {
      const updated = [...saved, product];
      localStorage.setItem("saved_product", JSON.stringify(updated));
      setIsSaved(true);
    }
  };
}

export default SaveButton;
```

```

        showToast("Removed from Saved successfully");
    } else {
        const updated = [...saved, product];
        localStorage.setItem("saved_product", JSON.stringify(updated));
        setIsSaved(true);
        showToast("Saved successfully");
    }
};

return (
    <>
    {isSaved ? (
        <button
            className={`${width} bg-gray-50 hover:bg-[#DC143C] hover:text-white transition font-bold px-8 rounded-lg text-[#DC143C] border-2 border-[#DC143C] flex items-center justify-center mb-4 hover:cursor-pointer whitespace nowrap`}
            onClick={handleSave}
        >
            Remove Saved
        </button>
    ) : (
        <button
            className={`${width} bg-[#DC143C] hover:bg-red-700 transition font-bold px-8 rounded-lg text-white flex items-center border-2 border-[#DC143C] justify-center mb-4 hover:cursor-pointer whitespace nowrap`}
            onClick={handleSave}
        >
            {isHeart ? <HeartIcon className="mx-1 w-5 h-5 text-white" /> : ""}
            Save Product
        </button>
    )}
);

{toast.show &&
    createPortal(
        <div className="fixed bottom-5 left-1/2 bg-black text-white px-6 py-4 text-sm font-bold rounded-lg z-50 popup-slide-up">
            {toast.message}
        </div>,
        document.body
    )
}

</>
);
}

export default SaveButton;

```

```
// components/SavedProduct.jsx
import { useEffect, useState } from "react";
import { Link } from "react-router-dom";
import { X } from "lucide-react";
import { StarIcon as StarSolid } from "@heroicons/react/20/solid";

function SavedProduct({ review = [] }) {
  const [saved, setSaved] = useState([]);
  const handleDelete = ((id) => {
    if (window.confirm("Are you sure to delete ?")) {
      const updated = saved.filter(p => p.ac_id !== id);
      localStorage.setItem("saved_product", JSON.stringify(updated));
      setSaved(updated);
    }
  });
  useEffect(() => {
    const stored = JSON.parse(localStorage.getItem("saved_product")) || [];
    setSaved(stored);
  }, []);

  if (saved.length === 0) return (
    <div className="my-10">
      <h2 className="text-2xl font-bold ">Saved Product</h2>
      <div className="noproduct h-100 flex justify-center items-center">
        <h2 className="font-bold text-4xl text-[#DC143C]">No Product Here !</h2>
      </div>
    </div>
  );
  return (
    <div className="px-4 md:px-0 my-10">
      <h2 className="text-2xl font-bold mb-4">Saved Product</h2>
      <div className="w-full grid grid-cols-8 gap-5">
        {saved.map((product) => {
          const productReviews = review.filter(r => r.product_id === product.ac_id);
          const totalReviews = productReviews.length;
          const avgRating = totalReviews
            ? (productReviews.reduce((sum, r) => sum + r.rating, 0) /
              totalReviews).toFixed(1)
            : 0;
          return (
            <div key={product.ac_id} className="col-span-8 md:col-span-4 lg:col-span-2 p-6 border border-gray-200 rounded-md hover:shadow-sm">
              <div className="flex justify-end">
                <button onClick={() => handleDelete(product.ac_id)}><X /></button>
              </div>
              <img
                src={`https://storage.googleapis.com/rheemcooling/${product.brand}/${product.ac_id}/${product.ac_id}_img1.webp`}
                alt={product.name}
                className="w-full h-50 my-5 md:mb-0 object-contain"
              />
            </div>
          );
        })
      </div>
    </div>
  );
}

export default SavedProduct;
```

```

        <Link
            to={`/product/${product.ac_id}`}
            className="font-bold text-sm line-clamp-3 mt-5 hover:underline min-h-[3.6em] leading-[1.2em]"
        >
            {product.name}
        </Link>

        <div className="flex items-center gap-1 mt-1 mb-3">
            <StarSolid className="w-4 h-4 text-yellow-500" />
            <span className="text-sm text-gray-500">{avgRating}</span>
            <span className="text-sm text-gray-500">| Sold {product.sale_quantity}</span>
        </div>
        <div className="flex gap-2 items-center mt-5">
            <p className="font-bold text-[#DC143C]">${product.price}</p>
            <p className="text-sm text-gray-500 line-through">${product.old_price}</p>
        </div>

        </div>
    );
);
</div>
</div>
);
}

export default SavedProduct;

```

4. Create Product Card

a. Description

- The **ProductCard** component is a reusable UI element designed to display product information succinctly and attractively.
- It includes conditional labels such as "New" and **discount percentage badges** to highlight promotional offers or new arrivals.
- The product image is prominently displayed with customizable wrapper and image styling to fit various layout contexts.
- The product name is a clickable link directing users to the detailed product page, supporting smooth navigation within the app.
- Pricing information shows the current price prominently, alongside the original price with a strikethrough if discounted, effectively communicating savings.
- Customer ratings and sales volume are displayed to build trust and encourage purchases.
- The component uses utility-first CSS classes for responsive typography and spacing, ensuring good appearance across devices.
- It is designed for flexibility and reuse, with props allowing custom styling and dynamic content injection.

b. Code (using code in ProductCard.jsx)

```
import React from "react";
import { Link } from "react-router-dom";

const ProductCard = ({  
  product,  
  imageSrc,  
  wrapperClassName = "",  
  imageClassName = "",  
}) => {  
  return (  
    <div className="">  
      {/* Label */}  
      {product.isNew && (  
        <span className="absolute top-2 left-2 bg-[#DC143C] text-white text-[10px] sm:text-xs font-bold px-1.5 sm:px-2 py-0.5 sm:py-1 rounded">  
          New  
        </span>  
      )}  
      <span className="absolute top-2 right-2 bg-yellow-400 text-[10px] sm:text-xs font-bold px-1.5 sm:px-2 py-0.5 sm:py-1 rounded">  
        - 20%  
      </span>  
      {/* {product.discount > 0 && (  
        <span className="absolute top-2 right-2 bg-yellow-400 text-xs font-bold px-2 py-1 rounded">  
          -{product.discount}%  
        </span>  
      )} */}  
    </div>  
  );  
}
```

```

/* Product Image */


<Link to={`/product/${product.ac_id}`} className="text-gray-800 font-bold line-clamp-3 mt-5
  hover:underline min-h-[3.6em] leading-[1.2rem] ">
    {product.name}
  </Link>
  <div className="flex items-baseline gap-2 sm:gap-3 mt-1 sm:mt-2 mb-1">
    <p className="text-[#DC143C] font-bold text-sm sm:text-lg md:text-lg">
      $ {product.price}
    </p>
    <p className="text-gray-400 line-through text-xs sm:text-sm">
      $ {product.old_price}
    </p>
  </div>
  <p className="text-yellow-400 text-xs sm:text-sm mb-1">
    ★
    <span className="text-gray-600">
      {" "}
      (4.9) • Sold {product.sale_quantity}
    </span>
  </p>
  </div>
</div >
);
};

export default ProductCard;


```

5. Scroll To Top

a. Description

- The **ScrollToTop** component ensures that the browser window scrolls to the top whenever the route (URL path) changes.
- It uses React Router's `useLocation` hook to monitor changes in the current pathname.
- On every pathname update, a `useEffect` hook triggers `window.scrollTo(0, 0)`, resetting the scroll position to the top-left corner of the page.
- This behavior improves user experience by preventing users from landing mid-page when navigating between different views or pages.
- The component itself does not render any visible UI (returns null) and solely manages scroll behavior as a side effect.
- It can be included at the root of the app or inside layout components to apply globally.

b. Code (using code in `ScrollToTop.jsx`)

```
import { useEffect } from "react";
import { useLocation } from "react-router-dom";

export default function ScrollToTop() {
  const { pathname } = useLocation();

  useEffect(() => {
    window.scrollTo(0, 0);
  }, [pathname]);

  return null;
}
```

6. Filter

a. Description

- The FilterProduct component provides a dynamic and interactive interface for refining the displayed product list based on multiple criteria.
- It supports filtering by **Category**, **Brand**, **Price Range**, and **Power**, offering users precise control over their search results.
- Filter options are automatically generated from the database (db) to ensure accuracy and adaptability when the product dataset changes.
- Each filter section uses checkboxes, allowing users to select multiple criteria simultaneously. The filtering logic dynamically updates the filterOption state without requiring page reloads.
- The useProductFilter custom hook centralizes filtering logic, maintaining a filterOption object and filteredProducts array, while also exposing helper functions like setFilterOption and clearAllFilters.
- Selected filters are preserved in the UI, ensuring a clear visual indication of active filter criteria.
- The component exposes filtering actions (setFilterOption and clearAllFilters) to parent components via the onExposeActions callback, enabling centralized control of filter behavior.
- Utility-first CSS classes are applied for consistent spacing, typography, and responsive layout across devices.

b. Code (using code in FilterProduct.jsx)

```
import React, { useState, useEffect } from "react";
import useProductFilter from "../customhooks/useProductFilter";

export default function FilterProduct({ db, onFilter, onExposeActions }) {
  const categorieList = Array.from(new Set(db.map((p) => p.type)));
  const brandList = Array.from(new Set(db.map((p) => p.brand)));
  const priceOptions = [
    { label: "Under $75", min: 0, max: 75 },
    { label: "$75 - $99.99", min: 75, max: 99.99 },
    { label: "$150 - $199.99", min: 150, max: 199.99 },
    { label: "$200 - $249.99", min: 200, max: 249.99 },
    { label: "$250 - $499.99", min: 250, max: 499.99 },
    { label: "$500 - $749.99", min: 500, max: 749.99 },
    { label: "Over $749.99", min: 749.99, max: Infinity },
  ];
  const powerList = Array.from(new Set(db.map((p) => p.power))).sort((a, b) => {
    const numA = parseFloat(a);
    const numB = parseFloat(b);
    return numA - numB;
  });
}
```

```

const {
  filterOption, //object
  filteredProducts, //array
  setFilterOption,
  clearAllFilters,
} = useProductFilter(db, onFilter);

useEffect(() => {
  if (onExposeActions) {
    onExposeActions({ setFilterOption, clearAllFilters });
  }
}, [onExposeActions, setFilterOption, clearAllFilters]);

return (
  <div className="p-4 bg-white border rounded-lg space-y-6 h-auto md:h-full max-h-none md:max-h-screen overflow-x-auto md:overflow-y-auto">
    {/* Category */}
    <div>
      <h3 className="text-[#DC143C] font-semibold mb-2">Category</h3>
      {categorieList.map((c) => (
        <label
          key={c}
          className="flex items-center gap-1 flex-wrap text-sm sm:text-base"
        >
          <input
            type="checkbox"
            className="mr-2"
            checked={filterOption.category.includes(c)}
            onChange={(e) =>
              setFilterOption((prev) => {
                const current = prev.category;
                return {
                  ...prev,
                  category: e.target.checked
                    ? [...current, c] // thêm category `c` vào mảng prev
                    : current.filter((cur) => cur !== c), // giữ lại những phần tử khác với c
                };
              });
            }
          >
          {c}
        </label>
      ))}
    </div>

    {/* Brand */}
    <div>
      <h3 className="text-[#DC143C] font-semibold mb-2">Brand</h3>
      {brandList.map((b) => (
        <label
          key={b}
          className="flex items-center gap-1 flex-wrap text-sm sm:text-base"
        >

```

(đã bị bỏ chọn)

```

<input
  type="checkbox"
  className="mr-2"
  checked={filterOption.brand.includes(b)}
  onChange={(e) =>
    setFilterOption((prev) => {
      const current = prev.brand;
      return {
        ...prev,
        brand: e.target.checked
          ? [...current, b]
          : current.filter((cur) => cur !== b),
      };
    })
  }
/>
{b.charAt(0).toUpperCase() + b.slice(1)}
</label>
))}

</div>

/* Price Range */
<div>
<h3 className="text-[#dc143c] font-semibold mb-2">Price</h3>
{priceOptions.map((p) => (
  <div key={p.label}>
    <label className="flex gap-2 items-center flex-wrap mb-1 text-sm sm:text-base">
      <input
        type="checkbox"
        name="price"
        checked={filterOption.priceRange.some(
          (x) => x.label === p.label
        )}
        onChange={(e) =>
          setFilterOption((prev) => {
            const current = prev.priceRange;
            return {
              ...prev,
              priceRange: e.target.checked
                ? [...current, p]
                : current.filter((x) => x.label !== p.label),
            };
          })
        }
      />
      {p.label}
    </label>
  </div>
))
}
</div>

/* Power */
<div>
<h3 className="text-[#dc143c] font-semibold mb-2">Power</h3>

```

```
{powerList.map((p) => (
  <label
    key={p}
    className="flex items-center gap-1 flex-wrap text-sm sm:text-base"
  >
    <input
      type="checkbox"
      className="mr-2"
      checked={filterOption.power.includes(p)}
      onChange={(e) =>
        setFilterOption((prev) => {
          const current = prev.power;
          return {
            ...prev,
            power: e.target.checked
              ? [...current, p]
              : current.filter((po) => po !== p),
          };
        })
      }
    />
    {p.split("(")[0].trim()}
  </label>
))})
</div>
</div>
);
}
```

7. Sort

a. Description

- The Sort component provides an intuitive interface for ordering products according to different criteria, improving browsing efficiency and user experience.
- Users can select from multiple sorting options, including **Top Selling, Top Deal, Price Low - High, Price High - Low, Customer Rating, and New Arrivals**.
- Sorting is implemented using the `useMemo` hook to optimize performance, ensuring that the sorted product list is only recalculated when the source data (db) or the selected sorting option changes.
- Each sorting criterion is defined with specific comparison logic, enabling accurate ordering based on numerical or boolean product attributes.
- The sorted product list is passed to the parent component via the `onSort` callback. To prevent unnecessary re-renders, a `useRef` is used to track and compare the serialized sorted output before sending updates.
- The sorting interface is implemented as a `<select>` dropdown for clarity and accessibility, with responsive styling applied through utility-first CSS classes.
- This component ensures seamless integration with other filtering and pagination features, supporting an enhanced and cohesive product browsing experience.

b. Code (using code in Sort.jsx)

```
import React, { useState, useEffect, useMemo, useRef } from "react";

export default function Sort({ db, onSort }) {
  const [sortOption, setSortOption] = useState("Top Selling");

  const sortedProducts = useMemo(() => {
    return [...db].sort((a, b) => {
      switch (sortOption) {
        case "Top Selling":
          return b.sale_quantity - a.sale_quantity;
        case "Top Deal":
          return b.discount - a.discount;
        case "Price Low - High":
          return a.price - b.price;
        case "Price High - Low":
          return b.price - a.price;
        case "Customer Rating":
          return b.rating - a.rating;
        case "New Arrivals":
          return b.isNew - a.isNew;
        default:
          return 0;
      }
    });
  }, [sortOption, db]);

  useEffect(() => {
    onSort(sortedProducts);
  }, [sortedProducts, onSort]);
}
```

```

        }
    });
}, [db, sortOption]);

const lastSentRef = useRef(null);
useEffect(() => {
    if (!onSort) return;
    const serialized = JSON.stringify(sortedProducts);
    if (lastSentRef.current !== serialized) {
        onSort(sortedProducts);
        lastSentRef.current = serialized;
    }
}, [sortedProducts, onSort]);

// useEffect(() => {
//   onSort(sortedProducts);
// }, [sortedProducts]);

return (
    <>
        <div>SORT BY: </div>
        <select
            value={sortOption}
            onChange={(e) => {
                setSortOption(e.target.value);
                // setCurrentPage(1); // reset về page 1 khi sort
            }}
            className="border rounded px-3 sm:px-4 py-1 sm:py-2 text-sm sm:text-base"
        >
            <option>Top Selling</option>
            <option>Top Deal</option>
            <option>Price Low - High</option>
            <option>Price High - Low</option>
            <option>Customer Rating</option>
            <option>New Arrivals</option>
        </select>
    </>
);
}
}

```

8. Search

a. Description

- Retrieves search query from URL and normalizes it for case- and accent-insensitive matching.
- Filters the product list based on the search keywords.
- Supports advanced filtering via FilterProduct component and sorting via Sort component.
- Manages filtered and sorted product states separately to ensure smooth updates.
- Displays filtered and sorted products with a limit of 10 items.
- Shows a fallback message and recommended products if no results are found.
- Includes a RecentView section to show recently viewed products.

b. Code (using code in SearchProduct.jsx & SearchPage.jsx)

```
//components/SearchProduct.jsx
import { useState } from "react";
import { useNavigate } from "react-router-dom";

function SearchProduct() {
  const [keyword, setKeyword] = useState("");
  const navigate = useNavigate();

  const handleKeyDown = (e) => {
    if (e.key === "Enter" && keyword.trim()) {
      navigate(`/search?query=${encodeURIComponent(keyword.trim())}`);
    }
  };

  return (
    <div className="searchproduct">
      <div className="hidden lg:block w-[200px] ms-5">
        <input
          type="text"
          placeholder="Search..."
          className="w-full px-4 py-2 text-sm border border-gray-300 rounded-xl
focus:outline-none focus:ring-2 focus:ring-blue-500"
          value={keyword}
          onChange={(e) => setKeyword(e.target.value)}
          onKeyDown={handleKeyDown}
          required
        />
      </div>
    </div>
  );
}

export default SearchProduct;
```

```
//components/SearchPage.jsx
import React, { useState, useEffect, useMemo } from "react";
import { useLocation } from "react-router-dom";
import FilterProduct from "../../src/components/product/FilterProduct";
import FinalList from "../../src/components/product/FinalList";
import db from "../data/product.json";
import RecentView from "../components/RecentView";
import Sort from "../components/product/Sort";

function useQuery() {
  return new URLSearchParams(useLocation().search);
}

function removeVietnameseTones(str) {
  return str
    .normalize("NFD")
    .replace(/[\u0300-\u036f]/g, "")
    .replace(/\d/g, "d")
    .replace(/\D/g, "D")
    .toLowerCase();
}

function SearchPage() {
  const query = useQuery();
  const keysearch = query.get("query") || "";

  const initialFiltered = useMemo(() => {
    if (!keysearch) return db;

    const keywords = removeVietnameseTones(keysearch).split(" ").filter(Boolean);

    return db.filter((p) => {
      const name = removeVietnameseTones(p.name);
      return keywords.some((kw) => name.includes(kw));
    });
  }, [keysearch]);

  const [filteredProducts, setFilteredProducts] = useState(initialFiltered);

  const [sortedProducts, setSortedProducts] = useState(initialFiltered);

  useEffect(() => {
    setFilteredProducts(initialFiltered);
    setSortedProducts(initialFiltered);
  }, [initialFiltered]);

  function handleFilter(filtered) {
    setFilteredProducts(filtered);
    setSortedProducts(filtered);
  }

  function handleSort(sorted) {

```

```

        setSortedProducts(sorted);
    }

    return (
        <div className="container mx-auto">
            <div className="grid grid-cols-10 gap-4">
                <section className="col-span-3 lg:col-span-2 my-5">
                    <FilterProduct db={initialFiltered} onFilter={handleFilter} />
                </section>

                <section className="col-span-7 lg:col-span-8 my-5 p-6 bg-white border rounded-lg">
                    <div>
                        <h2 className="text-xl font-semibold mb-4">
                            Search results for "{keysearch}"
                        </h2>
                        <hr className="mb-4" />
                    </div>

                    {/* Sort By */}
                    <div className="flex justify-end items-center gap-4 my-6">
                        <Sort db={filteredProducts} onSort={handleSort} />
                    </div>

                    {/* List */}
                    {sortedProducts.length > 0 ? (
                        <FinalList db={sortedProducts} visibleCount={10} />
                    ) : (
                        <div>
                            <p>No products found.</p>
                            <h2 className="text-xl font-semibold my-4">
                                You may also be interested in
                            </h2>
                            <FinalList db={db} visibleCount={10} />
                        </div>
                    )}
                </section>
            </div>
            <div>
                <RecentView />
            </div>
        </div>
    );
}

export default SearchPage;

```

TASK SHEET REVIEW 3

Project Ref. No.:		Project Title:	Activity Plan Prepared By:	Date of Preparation of Activity Plan:			
Sr. No.	Task			Actual Start Date	Actual Days	Team Mate Names	Status
1	Home Page	RHEEM COOLING	Pham Gia Bao	10-Aug-25	1	Quang	Completed
2	Product			10-Aug-25	1	Quang	Completed
3	Product Detail			10-Aug-25	1	Quang	Completed
4	About Us			10-Aug-25	1	Quang	Completed
5	Contact Us			10-Aug-25	1	Gia Bao	Completed
6	Customer Service			10-Aug-25	1	Gia Bao	Completed
7	Saved Product			10-Aug-25	1	Gia Bao	Completed
8	Java Script			12-Aug-25	2	G.Bao & T.Bao	Completed
9	Responsive			12-Aug-25	1	Gia Bao	Completed