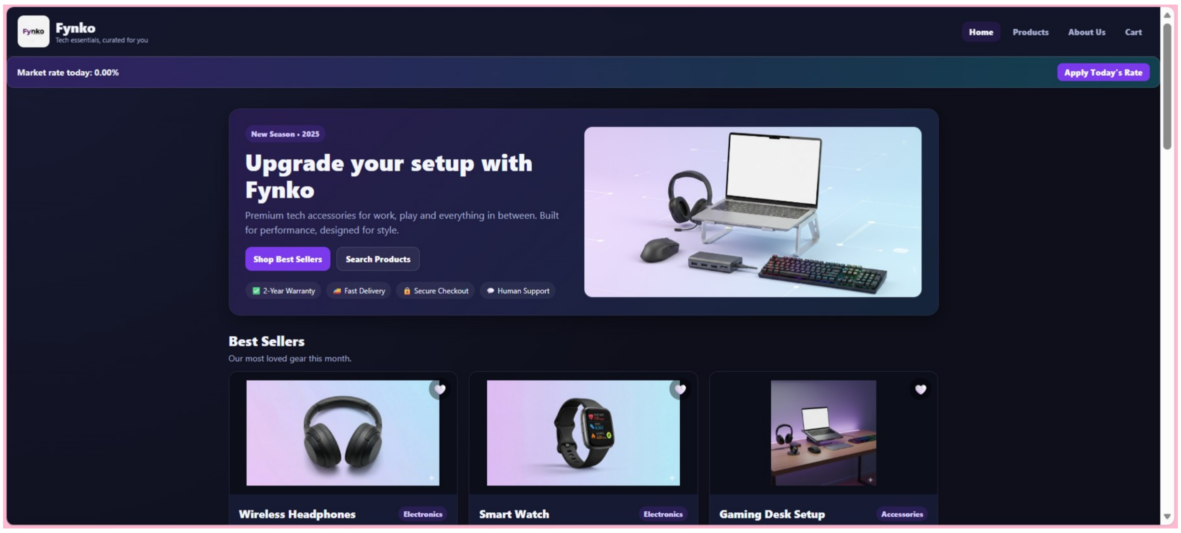
CCT College Dublin Continuous Assessment

Programme Title:	*H.Dip. in Computing*		
Delivery Mode:	FT		
Cohort Details:	Feb 2025 cohort		
Module Title(s):	Web Development		
Assignment Type:	*Individual*	**Weighting(s):**	60%
Assignment Title:	Full Stack Website		
Lecturer(s):	Shree Krishna Acharya		
Issue Date:	6 Nov. 2025 @ 23:59		
Submission Deadline Date:	7 Dec. 2025 @ 23:59		
Late Submission Penalty:	Late submissions will be accepted up to 5 calendar days after the deadline. All late submissions are subject to a penalty of 10% of the mark awarded. Submissions received more than 5 calendar days after the deadline above will not be accepted and a mark of 0% will be awarded.		
Method of Submission:	**This assignment is submitted via Moodle.**		
Instructions for Submission :	**Web Development:** Submit your website files in ONE FOLDER (in .zip format) to the Web Development Moodle page. You will also create a short demo video. DO NOT UPLOAD THIS TO MOODLE. Instead, save the video to your GDrive space and upload a link to the video. MAKE SURE that your video is accessible for viewing by others!		
Feedback Method:	**Results posted in Moodle gradebook**		
Feedback Date:			

Table of Contents:



1. Introduction

This report presents the development of Fynko, a web-based e-commerce application developed for the Continuous Assessment 2 (CA2) of the Web Development module at CCT College Dublin. The main objective of the project was related to the design and implementation of a digital platform capable of reproducing the basic functionalities of a modern commercial system in terms of product browsing, cart management, dynamic price updating, and delivering a responsive and intuitive user experience. To accomplish such purposes, a full-stack architecture has been adopted, integrating both front-end and back-end technologies with a relational database system.

The whole implementation of the application was carried out from scratch, with HTML5 for semantic structuring, CSS3 for responsive design, JavaScript enabling dynamic interactions, and a Node.js-Express-based server that exposes REST APIs for communication across layers. Data storage and management are handled by the SQLite3 database, responsible for storing and updating information about all products, cart operations, and the pricing mechanism.

One of the most relevant features related to the project is the Market Rate Engine, which simulates fluctuations in real-world markets by applying changes in prices within the range of –3% to +3%, updating the value of products directly into the database for consistency of successive operations. This gives a dynamic behavior similar to that adopted in modern commercial platforms and enhances the practical and applied nature of the development process.

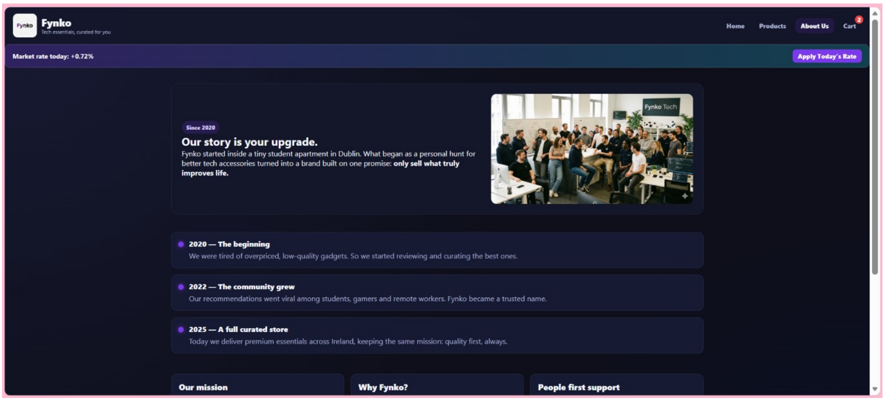
The methodology adopted, the architectural decisions, server configuration, data model, dynamic logic implementation, and the design strategies are presented throughout this report. This document shows how Fynko fully meets the learning outcomes of the module, showcasing the construction of a functional, scalable, and technically sound solution.

2.Justification of the Chosen Methodology

A pragmatic iterative full-stack methodology was used to develop Fynko. The system was developed incrementally in layers, starting with the front-end layout and page-to-page navigation, then the product catalog and client-side interactions. Later versions included the REST API and server-side routing, database persistence using SQLite3, and finally, the dynamic pricing engine. Constructing the application in this fashion, in clearly demarcated iterations, mitigates development risk since each layer is a known-good state before new functionality is added. This follows best practices in industry for modern web application engineering, where the system's complexity grows incrementally and each subsystem is tested independently before integrating with other subsystems.

Additionally, the separation of concerns has been made across the presentation layer, which is HTML, CSS, and JavaScript; the application layer with Node.js/Express REST API; and the data layer, SQLite3. Such represents a three-tier architecture with structure that will enhance scalability, maintainability, and extensibility since changes in one layer would not compromise the functionality of the others.

This methodology is particularly suitable for an e-commerce platform where the systems continuously evolve to meet users' needs and market dynamics. Iterative development supports ongoing user experience refinement, ensures that API endpoints remain aligned with UI requirements, and allows for the easy addition of new features-like category filters or market-rate pricing-without requiring substantial core-logic refactoring. As such, the approach is robust, future-proof, and able to accommodate enhancements in the future.



Market rate today: +0.72%

Since 2020

Our story is your upgrade.

Fynko started inside a tiny student apartment in Dublin. What began as a personal hunt for better tech accessories turned into a brand built on one promise: **only sell what truly improves life.**

2020 — The beginning
We were tired of overpriced, low-quality gadgets. So we started reviewing and curating the best ones.

2022 — The community grew
Our recommendations went viral among students, gamers and remote workers. Fynko became a trusted name.

2025 — A full curated store
Today we deliver premium essentials across Ireland, keeping the same mission: quality first, always.

Our mission

Why Fynko?

People first support

3.System Architecture Overview

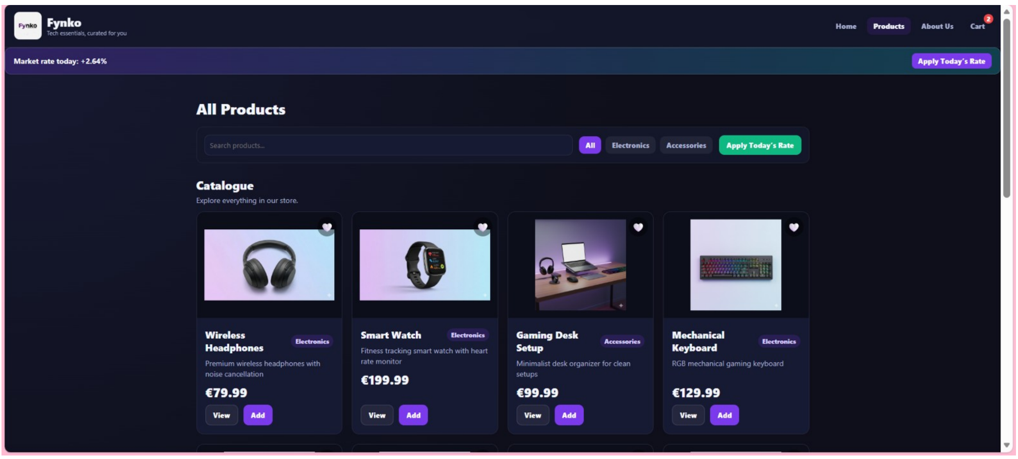
Fynko is built using a client-server model aimed at enhancing clarity, scalability and effective interaction, between the system’s parts. The client side comprises HTML, CSS and JavaScript files delivered as resources. Its responsibilities include displaying the user interface executing product searches implementing filters managing the wishlist through LocalStorage and processing all user-driven actions.

The server tier is built with Node.js utilizing the Express framework, which provides a collection of API endpoints managing all application logic. Via these endpoints the server supplies product data, handles cart functionalities. Carries out dynamic price adjustments managed by the Market Rate Engine. Consequently it guarantees that business logic stays centralized and can be reused across client platforms.

The data layer depends on an SQLite3 database that holds persistent data regarding product information, current pricing and cart contents. The backend application carries out all create-read-update- CRUD) tasks, on this database to ensure data integrity and provide regulated access.

Data moves in a direction from the database to the API and subsequently to the client interface. User interactions, like adding products to the cart or using the market rate start on the client side and are sent to the server via API requests. These interactions cause changes, in the database guaranteeing that all state modifications are logged and consistently displayed throughout the application.

This layered architecture reinforces logical separation and maintainability, along with best practices concerning modern full-stack web development, supporting future scalability and the expansion of features.



4.Server System Configuration

The Node.js runtime, along with the Express.js framework, is used for the server-side architecture of Fynko, which offers a modular, scalable, and high-performance environment while managing application logic. The server exposes RESTful API endpoints, processes all requests coming from the client side, performs database operations, and serves the static assets of the front-end. This configuration provides a clear separation of concerns and, hence, a reliable way of communicating between the client interface and the database layer.

The configuration of the Express server comprises several key parts: initialization of the application with core middlewares-that is, JSON body parsing and serving of static files-which enables structured request handling and efficient delivery of UI files. Utilization of dedicated route modules to organize API responsibilities by separating functionalities into domains like product management, shopping cart operations, and dynamic market-rate updates ensures code maintainability and has been devised with best practices for scalable back-end design in mind.

Finally, there's error-handling middleware that's similarly robust in constructing uniform HTTP status codes and standardized response messages. This will add predictability to the application behaviors, make debugging easier, and generally increase the robustness of the application. Several NPM scripts can also be created for more efficient workflow automation at both development and production, enabling tasks such as starting the server, installing dependencies, or environment setup.

The RESTful endpoints implemented in the server are structured in a resource-oriented fashion. Examples include:

GET /api/products returns the entire product catalogue

GET /api/products/:id returns details of a specific product

POST /api/cart/add - adds an item to the persistent shopping cart

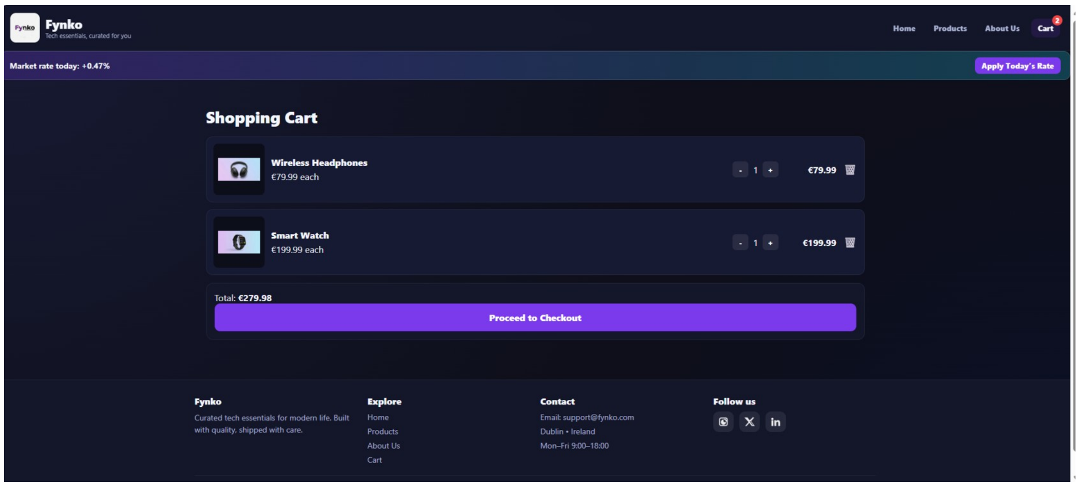
PATCH /api/cart/update - updates product quantities within the cart.

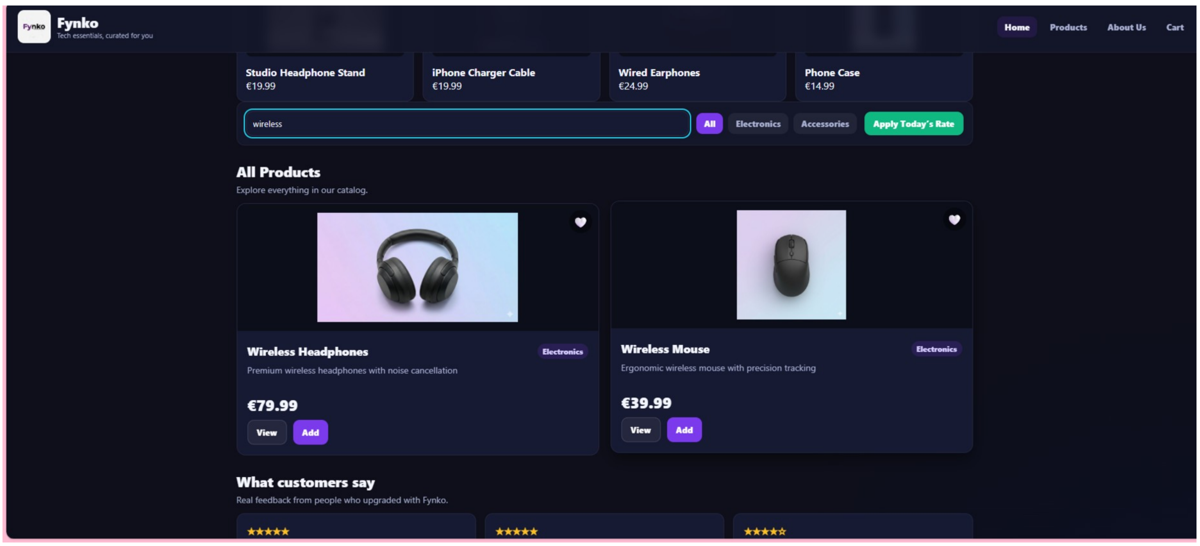
DELETE /api/cart/remove/:id – removes item

POST /api/market/apply rate - applies today's market fluctuation to all products.

These endpoints provide the backbone for the communication layer, taking care that all front-end interactions would be processed, validated, and persisted through structured API calls.

Overall, the server configuration reflects industry norms with regard to full-stack web development, through providing a stable, modular, and extensible platform for supporting the current and future features of the Fynko application.





5.Database Connection & Dynamic Content

SQLite3 is used as the relational database management system in the Fynko application because it is lightweight, highly reliable, and appropriate for local development and simplified deployment. SQLite works with only one file-based database to efficiently perform reads and writes without the overhead of a full database server. The Express back-end connects to this SQLite file in a persistent fashion, performing SQL queries to perform all data interactions: selects, inserts, updates, and deletes.

The database schema revolves around two central tables, which provide the crucial functionality for the application.

products (id, name, description, category, imageUrl, price, stock, rating, …)

cart (id, productId, quantity, subtotal, createdAt, …)

Taken together, these tables manage product metadata, pricing, and availability, along with the state of the user's cart. This can and will be modified to suit specific implementation requirements; however, the relational nature of these tables ensures data integrity and query performance.

The Market Rate Engine is a server-side implementation of a key dynamic feature of the system, focusing on simulating price variation realistically driven by markets. This mechanism will create a daily fluctuation ranging from –3% to +3% and recalculate the prices of the products accordingly. If the user activates the "Apply Today's Rate" function, the server will loop through the product list, adding the calculated adjustment to each price before writing back the new value into the SQLite database. Following this process, the refreshed product catalogue is fetched by the front-end to ensure that all displayed prices remain valid at any moment in time, synchronized, and persistent across user sessions and devices.

This unified flow of dynamic computation on the server, persistence of data, and synchronized client updates is an effective integration of database management and real-time content generation. It also reflects well on contemporary e-commerce pricing strategies, thereby reinforcing both technical and practical relevance for the solution.

6.Front-End Design, Responsiveness & Accessibility

Fynko's front-end design philosophy involved designing for mobile first, so that the UI blends fluidly into mobile, tablet, and desktop breakpoints. This responsive layout is dependent on CSS Grid and Flexbox for structuring, allowing flexibility in changing screen dimensions with ease. For the visual, much emphasis was made on hierarchical organization, consistent spacing, and proper typography to meet the professional and modern aesthetic of any e-commerce site.

At the core, interactive features are powered by vanilla JavaScript that manages live product search filtering, category selection, adjustment of cart quantities, persistence of a wishlist through LocalStorage, and dynamic price updates against market-rate adjustments. This set of behaviors enables a smooth and dynamic user experience, minimizing the need for full page reloads and thereby enhancing overall usability.

Accessibility strongly influenced the frontend implementation to ensure that it catered to modern web standards for access and inclusive design. Key accessibility measures include:

Using Semantic HTML5 elements to clearly define document structure and support assistive technologies.

Applying ARIA labels to interactive elements for which standard HTML controls are not used.

All images will include alternative text to make content more accessible to screen-reader users.

Adoption of high-contrast color schemes would improve the readability of the website for users with visual impairments.

Ensure that there is full keyboard navigability for access by users who have limited motor abilities.

Together, these design strategies make sure that Fynko's front-end is visually appealing and responsive but also inclusive, user-centered, and aligned with industry best practices for modern web development.

7.Testing & Validation

A proper testing strategy was implemented to verify that the Fynko application would meet both functional requirements and standards related to user experience. Such a validation process involved several layers of the system: the behavior of APIs, the responsiveness of interfaces, and input handling. The following categories summarize the testing procedures conducted:

• API Testing:

RESTful endpoints have been rigorously validated through browser-based fetch calls and tools like Postman. Each of these requests has been tested for proper HTTP status codes, correct payload structures, and proper handling of errors. These tests ensured the server responded reliably to valid and invalid operations-exhibiting consistent behavior across all interactions.

• UI Testing:

Manual testing was conducted for the main user flows, including product search functionality, category filters, and cart interactions. Extra attention was given to edge cases, such as adding duplicate items to the cart, removing the last item, and verifying that the wishlist persists via LocalStorage, in order to confirm the robustness and stability of the user interface.

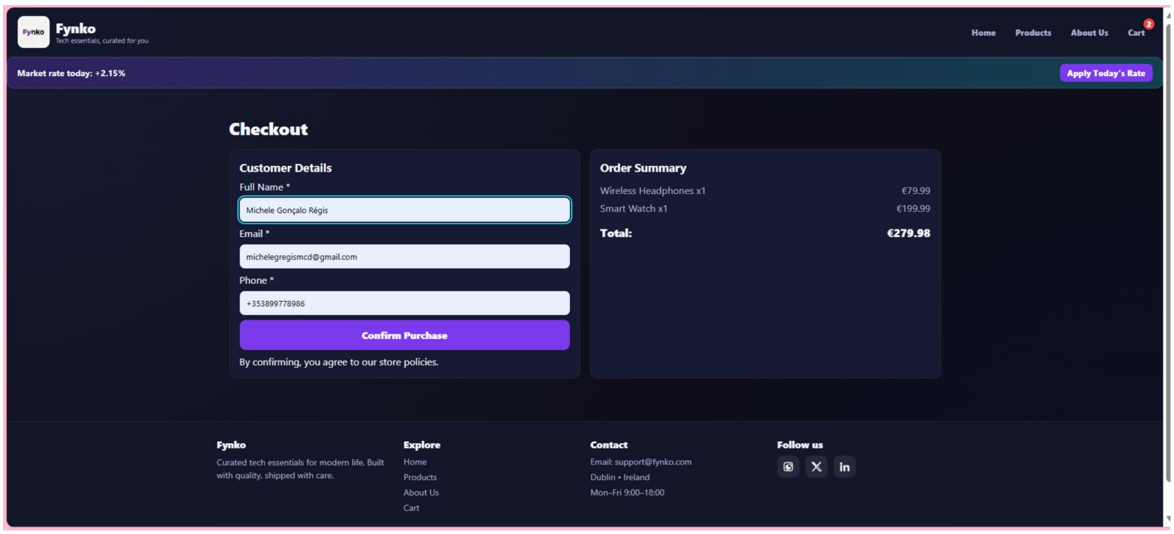
• Responsiveness Testing:

Testing of the layout adaptability was performed by using Chrome DevTools over a range of screen resolutions. The breakpoints considered were for mobile, tablet, and desktop; thus, the elements would scale properly and be usable across devices.

• Input Validation:

Server-side validation was implemented and tested to ensure that invalid product IDs, malformed quantities, or other unexpected inputs would be rejected safely. Clear, secure error messages were returned to the client, preventing unintended behaviours while reinforcing the reliability of the system.

With this multi-layer approach, the application demonstrated consistent performance and a high level of usability, confirming that its components function cohesively within the intended full-stack architecture.



Fynko
Tech essentials, curated for you

Home Products About Us Cart 2

Market rate today: +2.15%
Apply Today's Rate

Checkout

Customer Details
Full Name *
Michele Gonçalo Régis
Email *
michelegregismcd@gmail.com
Phone *
+353899778986
Confirm Purchase
By confirming, you agree to our store policies.

Order Summary

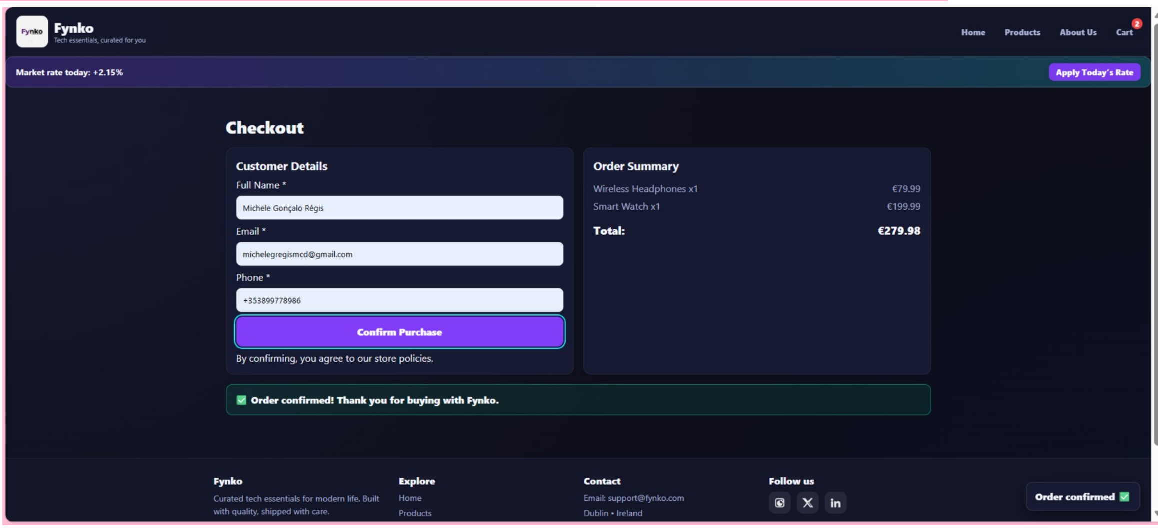
Wireless Headphones x1	€79.99
Smart Watch x1	€199.99
Total:	**€279.98**

Fynko
Curated tech essentials for modern life. Built with quality, shipped with care.

Explore
Home
Products
About Us
Cart

Contact
Email: support@fynko.com
Dublin • Ireland
Mon–Fri 9:00–18:00

Follow us



Fynko
Tech essentials, curated for you

Home Products About Us Cart 2

Market rate today: +2.15%
Apply Today's Rate

Checkout

Customer Details
Full Name *
Michele Gonçalo Régis
Email *
michelegregismcd@gmail.com
Phone *
+353899778986
Confirm Purchase
By confirming, you agree to our store policies.

Order Summary

Wireless Headphones x1	€79.99
Smart Watch x1	€199.99
Total:	**€279.98**

✅ Order confirmed! Thank you for buying with Fynko.

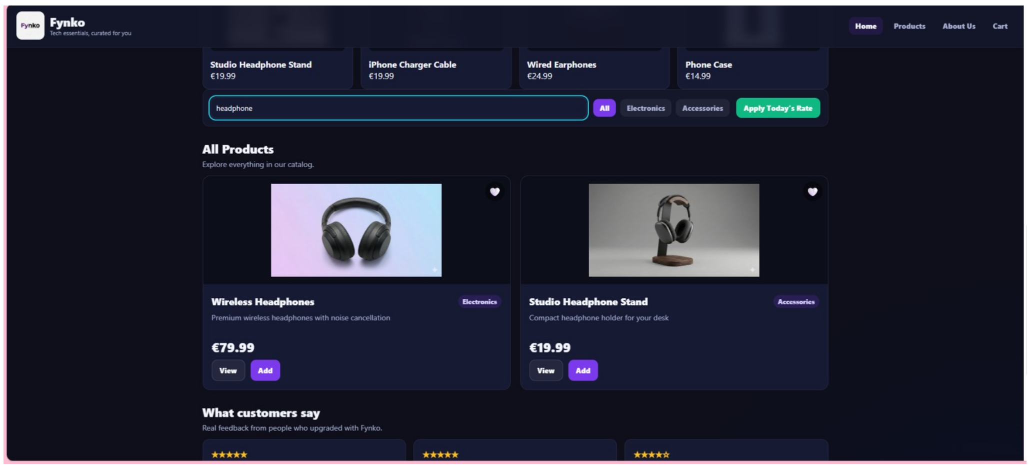
Fynko
Curated tech essentials for modern life. Built with quality, shipped with care.

Explore
Home
Products

Contact
Email: support@fynko.com
Dublin • Ireland

Follow us

Order confirmed ✅



Fynko
Tech essentials, curated for you

Home Products About Us Cart

Studio Headphone Stand
€19.99
iPhone Charger Cable
€19.99
Wired Earphones
€24.99
Phone Case
€14.99

headphone
All Electronics Accessories Apply Today's Rate

All Products
Explore everything in our catalog.

Wireless Headphones
Electronics
Premium wireless headphones with noise cancellation
€79.99
View Add

Studio Headphone Stand
Accessories
Compact headphone holder for your desk
€19.99
View Add

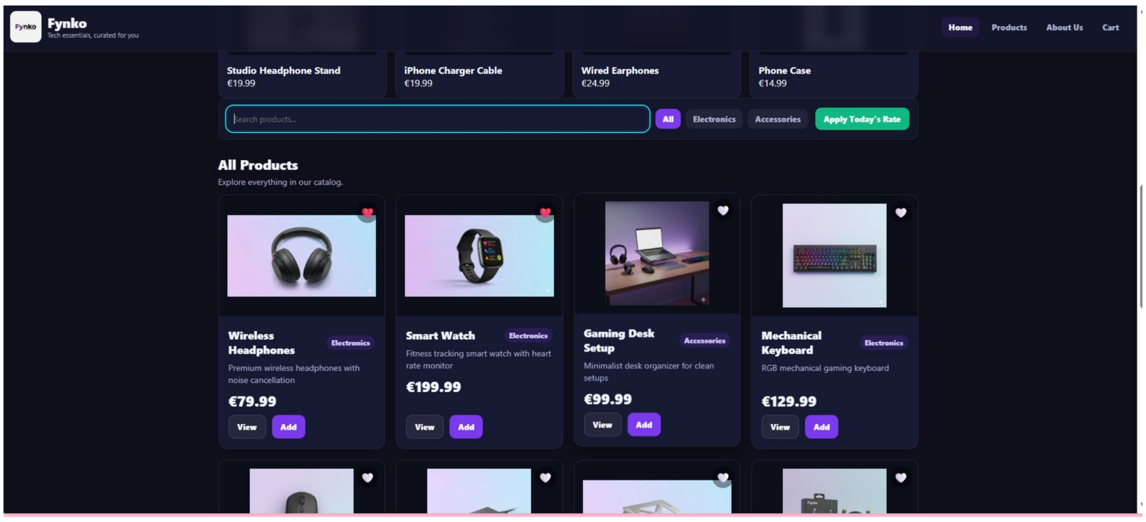
What customers say
Real feedback from people who upgraded with Fynko.
★★★★★ ★★★★★ ★★★★★

8.Conclusion

Fynko application meets the requirements of the CA2 Web Development assessment through the delivery of a fully functional, responsive, and database-driven e-commerce platform with an integrated dynamic market-rate pricing engine. The system formalizes a separation of concerns into three distinct logical layers: the presentation, application, and data tiers, thus ensuring a well-maintainable system with realistic operational behavior consistent with modern full-stack development best practices.

The iterative development methodology followed for the entire project helped in the structured enhancement of features, continuous verification of components, and effective integration of the front-end and back-end functionalities. Consequently, Fynko exhibits solid performance over product browsing, cart management, dynamic pricing, and user-interface interactions that reflect cohesive implementation in alignment with real-world e-commerce workflows.

Going forward, the platform provides great scope for further improvement. Future development may include the process of user authentication, role-based access control, integration of secure payment gateways, enhancement of administrative functionalities like inventory dashboards, and analytics related to user behavior and sale trends. These extensions would even better position the system to be at a production level in an e-commerce solution.



9. References

Mozilla Developer Network (2024) ‘HTML: HyperText Markup Language’. Available at: https://developer.mozilla.org/en-US/docs/Web/HTML (Accessed: 24 November 2025).

Mozilla Developer Network (2024) ‘CSS: Cascading Style Sheets’. Available at: https://developer.mozilla.org/en-US/docs/Web/CSS (Accessed: 24 November 2025).

Mozilla Developer Network (2024) ‘JavaScript Guide’. Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide (Accessed: 24 November 2025).

Express.js (2024) ‘Express – Node.js web application framework’. Available at: https://expressjs.com/ (Accessed: 24 November 2025).

Node.js (2024) ‘Node.js Documentation’. Available at: https://nodejs.org/en/docs (Accessed: 24 November 2025).

SQLite (2024) ‘SQLite Documentation’. Available at: https://www.sqlite.org/docs.html (Accessed: 24 November 2025)

GitHub (2025) - https://github.com/MicheleRegis/CA2Ecommerce.app.git