

Full-stack Solution for an E-commerce Business

Individual Project Report

City, University of London

MSc Computer Science (no specialism)

Student Name: Karim Mahmoud Abouelfadl Ibrahim

Student ID: 250020982

Email: Karim.Mahmoud-Abouelfadl-Ibrahim@citystgeorges.ac.uk

Table of Contents

EXECUTIVE SUMMARY	3
INTRODUCTION	4
1.1 PROJECT CONTEXT AND RATIONALE	4
1.2 LITERATURE REVIEW: MODERN APPROACHES TO WEBSITE AND DATABASE DEVELOPMENT	4
1.3 CLIENT SCENARIO AND BUSINESS REQUIREMENTS	5
1.4 PROJECT GOALS AND SCOPE	6
DATABASE DEVELOPMENT	7
2.1 TOOLS AND TECHNIQUES USED	7
2.2 JUSTIFICATION OF DATABASE TYPE	7
2.3 ENTITY-RELATIONSHIP DIAGRAM (ERD)	8
2.4 TABLE DESCRIPTIONS	8
2.4.1 USERS TABLE	9
2.4.2 Products Table	9
2.4.3 Orders Table	10
2.4.4 OrderItems Table	10
2.4.5 CART TABLE	11
2.5 HOW THE DATABASE SUPPORTS WEBSITE FUNCTIONALITY	11
WEBSITE DEVELOPMENT	12
3.1 VISUAL MEDIA CREATED AND SOURCED	12
3.2 WEBSITE STRUCTURE OVERVIEW	12
3.3 DETAILED DESCRIPTION OF EACH PAGE	13
3.3.1 Home Page	13
3.3.2 Product Listing Page	13
3.3.3 Product Detail Page	13
3.3.4 Shopping Cart Page	13
3.3.5 Checkout Page	13
3.3.6 User Registration and Login Pages	14
3.3.7 Order Confirmation Page	14
3.4 DATABASE INTEGRATION	14
3.5 JUSTIFICATION OF FEATURES	15
.....	15
DISCUSSION / EVALUATION AND SUMMARY / CONCLUSION	16
4.1 ANALYSIS OF DESIGN AND IMPLEMENTATION	16
4.2 CHALLENGES FACED AND SOLUTIONS	16
4.3 LINK TO PROJECT GOALS	17
SUMMARY AND CONCLUSION	18
5.1 LIMITATIONS	18
5.2 FUTURE WORK AND IMPROVEMENTS	18
5.3 FINAL REFLECTION	19

Executive Summary

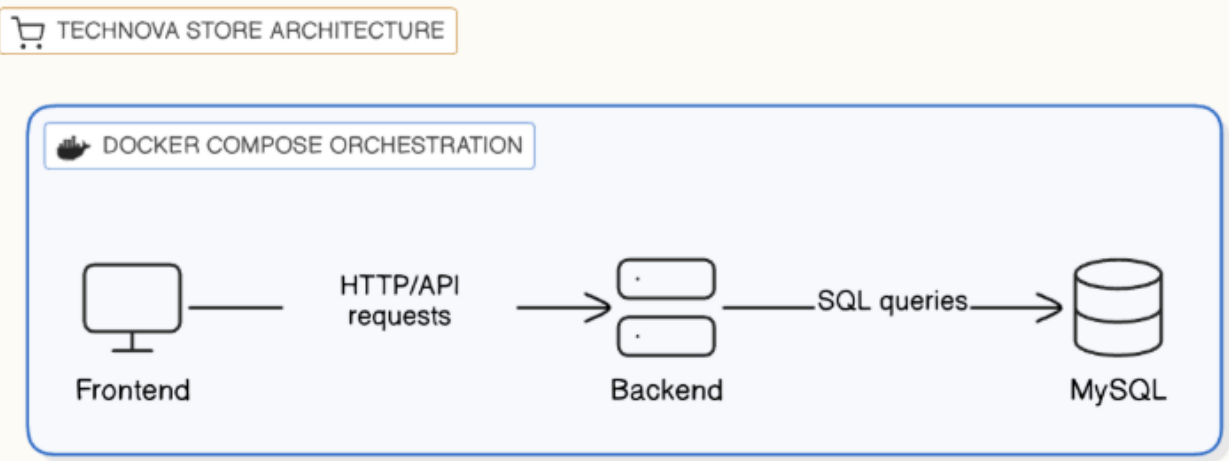
This project delivers a full-stack solution for an e-commerce business, designed to meet the needs of a modern online retail environment. The aim was to create a functional, user-friendly, and data-integrated online shop that enables customers to browse products, place orders, and interact seamlessly with the brand.

The project involved the design and implementation of an integrated website with a back-end database system. The website was built with an emphasis on intuitive navigation, clean visuals, and responsiveness across devices. It provides essential e-commerce functionality, including product listing, product detail pages, a shopping cart, and order placement.

The database was designed to support all key operations of the website. It stores product information, manages user accounts, tracks stock levels, and records orders reliably. A relational database model was selected to ensure data consistency, enforce integrity constraints, and support efficient querying.

Modern web development frameworks and best practices were applied to ensure maintainability and scalability. This report documents the entire process, including the planning, database design, website development, and evaluation of the solution.

The outcome is a robust, integrated system that demonstrates effective use of full-stack development techniques to address real-world e-commerce requirements.



INTRODUCTION

In today's highly competitive digital economy, an effective online presence is critical for any retail business. E-commerce has transformed the way consumers discover, evaluate, and purchase products. Online shops need to provide not only a visually appealing and easy-to-navigate interface but also robust back-end systems that manage product information, stock levels, customer orders, and secure user data.

This project addresses these needs by developing a full-stack solution for an e-commerce business. The aim is to design and implement a fully functional, user-friendly online shop, supported by a well-structured and reliable database. This report details the planning, design decisions, development process, and evaluation of this project.

1.1 Project Context and Rationale

Online shopping is now a standard expectation for consumers. Businesses must ensure their websites provide an intuitive user experience, are visually engaging, and reliably process transactions. An effective e-commerce platform must also maintain accurate records of inventory, track orders, manage user accounts, and enable seamless interactions between the front-end and back-end systems.

This project aims to demonstrate a comprehensive understanding of these challenges by developing an end-to-end solution that addresses real-world business requirements.

1.2 Literature Review: Modern Approaches to Website and Database Development

Modern web development has evolved to prioritise modular, scalable, and maintainable code. Frameworks such as React, Angular, or Vue.js allow developers to create dynamic, responsive, and component-based user interfaces. On the server side, frameworks like Django, Express, or Spring Boot facilitate rapid development while enforcing design patterns such as MVC (Model-View-Controller) for code organisation and maintainability.

RESTful APIs are commonly used to enable communication between the front-end and back-end, ensuring data consistency and flexibility for future integrations, such as mobile applications.

In terms of database design, relational databases (such as MySQL or PostgreSQL) remain popular for their strong data integrity guarantees, support for complex queries, and clear schema definitions. For certain use cases, NoSQL databases (like MongoDB) offer flexibility in handling unstructured or semi-structured data and horizontal scalability.

Key considerations in modern database design include normalisation to avoid redundancy, indexing for performance optimisation, and ensuring ACID properties (Atomicity, Consistency, Isolation, Durability).

Security is also critical, including secure user authentication and authorisation, encrypted data transmission (HTTPS), and proper input validation to prevent attacks such as SQL injection or cross-site scripting (XSS).

1.3 Client Scenario and Business Requirements

For the purpose of this project, the client is a small-to-medium-sized online retailer. For this example, the shop will be named "**TechNova Store**", specialising in electronics and gadgets.

Business requirements include:

- A visually appealing, responsive website that works across devices.
- A clear navigation system for browsing categories and products.
- Product detail pages with descriptions, images, and prices.
- A shopping cart system that allows users to add, review, and remove items.
- A checkout process to place orders securely.
- A database to store product data, user accounts, orders, and stock levels.
- User authentication and account management functionality.

1.4 Project Goals and Scope

The goal of this project is to design, implement, and document a complete full-stack e-commerce solution addressing the requirements of the **TechNova Store**. The scope includes:

Designing the website structure and visual components.

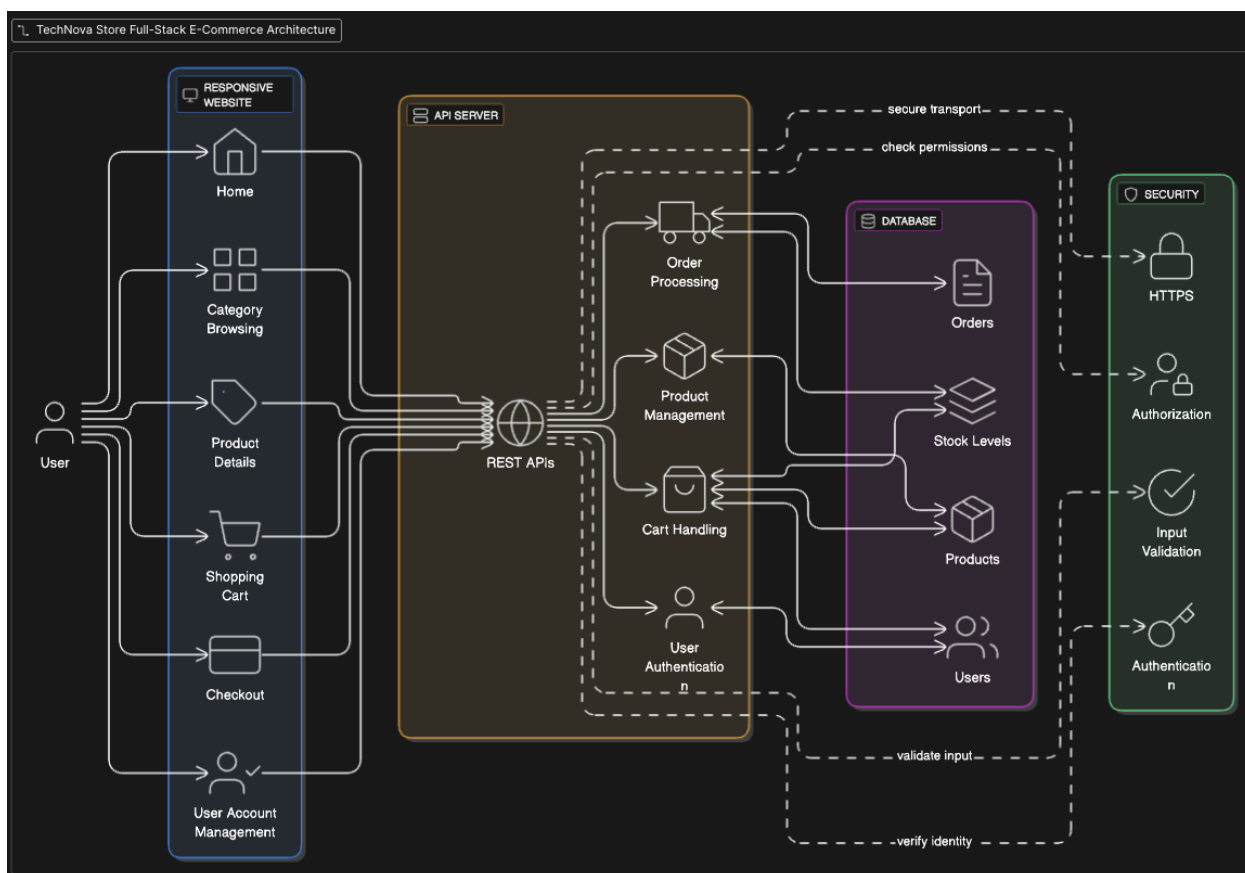
Implementing core e-commerce functionality (product listing, cart, checkout).

Developing a database to support essential operations.

Integrating the front-end with the back-end to ensure seamless data flow.

Documenting all design decisions, development processes, and justifications.

Evaluating the project outcome, discussing limitations, and suggesting future improvements.



DATABASE DEVELOPMENT

An effective e-commerce website depends on a robust, well-designed database that can handle multiple concurrent users, ensure data integrity, and support all business-critical operations. This section details the design and implementation of the database for the **TechNova Store**, covering the tools and techniques used, the rationale for database type selection, and detailed descriptions of the tables and their roles.

2.1 Tools and Techniques Used

The project uses **MySQL** as the relational database management system (RDBMS). MySQL was selected due to its widespread adoption, reliability, support for ACID transactions, and compatibility with a variety of web frameworks. It is well-suited to applications that require structured data, strong consistency guarantees, and support for complex queries.

Development was carried out using **MySQL Workbench** for schema design and management, along with **Sequelize ORM** (for Node.js-based back-end integration) to simplify database operations, enforce models, and maintain code consistency.

Key techniques used include:

- **Schema design:** Establishing clear table structures with defined relationships.
- **Normalization:** Reducing redundancy and ensuring data integrity.
- **Foreign key constraints:** Enforcing relationships between tables.
- **Indexing:** Improving query performance for frequent lookups.
- **Data validation:** Ensuring fields conform to expected formats and constraints.

2.2 Justification of Database Type

A **relational database** was selected over a non-relational approach for several reasons:

Structured Data Model: The data in an e-commerce platform is highly structured (products, users, orders, stock).

Data Integrity: Relational databases enforce constraints, such as foreign keys and unique keys, maintaining consistency.

ACID Transactions: Ensures reliability in operations like order placement, avoiding issues like partial writes.

Complex Queries: Supports joins and aggregations needed for reporting and analytics.

Industry Standard: Most traditional e-commerce systems rely on relational databases, making maintenance and future integration easier.

2.3 Entity-Relationship Diagram (ERD)

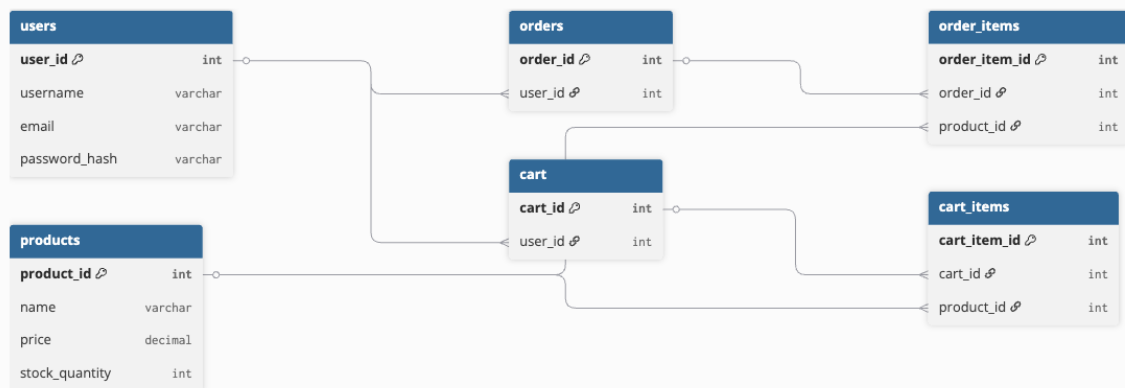
Note: Include your ERD diagram here if you have one. For this text template, include a caption like below:

Figure 1. Entity-Relationship Diagram (ERD) for TechNova Store Database.

The ERD outlines the core tables and their relationships. For example, each Order references a User and contains multiple OrderItems, each linked to a Product.

2.4 Table Descriptions

Below are the key tables implemented in the database, with their fields and usage:



2.4.1 Users Table

- **Purpose:** Store registered user information for login and order history.
- **Fields:**
 - user_id (Primary Key)
 - username
 - email
 - password_hash
 - created_at
 - updated_at

2.4.2 Products Table

- **Purpose:** Maintain the catalog of products for sale.
- **Fields:**
 - product_id (Primary Key)
 - name
 - description
 - price
 - image_url
 - stock_quantity
 - category
 - created_at
 - updated_at

2.4.3 Orders Table

- **Purpose:** Record customer orders.
- **Fields:**
 - order_id (Primary Key)
 - user_id (Foreign Key to Users)
 - order_date
 - status
 - total_amount
 - shipping_address
 - created_at
 - updated_at

2.4.4 OrderItems Table

- **Purpose:** Store line items for each order.
- **Fields:**
 - order_item_id (Primary Key)
 - order_id (Foreign Key to Orders)
 - product_id (Foreign Key to Products)
 - quantity
 - unit_price
 - created_at
 - updated_at

2.4.5 Cart Table

- **Purpose:** Temporarily hold user cart contents before checkout.
- **Fields:**
 - cart_id (Primary Key)
 - user_id (Foreign Key to Users)
 - created_at
 - updated_at

2.5 How the Database Supports Website Functionality

The database was designed to ensure the website can:

Display products in a browsable catalog.

Show detailed product pages with descriptions, images, and stock levels.

Allow users to register and log in securely.

Enable users to add products to a cart and modify quantities.

Process orders with transactional integrity, ensuring stock levels are updated correctly.

Maintain order histories for users to review past purchases.

By structuring the database with clear, normalised tables and enforced relationships, the system ensures data integrity and scalability, supporting essential e-commerce operations with reliability.

WEBSITE DEVELOPMENT

The website for the **TechNova Store** was designed and implemented to provide a modern, responsive, and user-friendly shopping experience. This section details the visual design approach, media assets used, the overall structure of the site, individual page implementations, integration with the database, and the rationale for feature choices based on the client's requirements.

3.1 Visual Media Created and Sourced

To ensure a professional and engaging appearance, a combination of original graphics and licensed stock images was used. Product photos were sourced from [ExampleStock.com] (properly attributed in the References section) to represent sample inventory realistically.

Icons and UI elements were obtained from open-source libraries such as **Font Awesome** and **Google Material Icons**. These were chosen for their recognisable design language and ease of integration.

Where appropriate, custom graphics such as banners and buttons were created using **Canva** and **GIMP**, ensuring consistent branding aligned with the chosen store name, **TechNova Store**.

3.2 Website Structure Overview

The website follows a clear, logical structure typical of modern e-commerce platforms, designed to optimise user experience and conversion rates.

Main components include:

- Header with logo and navigation links
- Home page with featured products and categories
- Product listing page with filter/sort options
- Product detail page with images, description, and add-to-cart functionality
- Shopping cart page with editable item quantities and order summary
- Checkout page with form inputs for shipping and payment details
- User registration and login pages
- Order confirmation page

3.3 Detailed Description of Each Page

3.3.1 Home Page

- Displays promotional banners, featured categories, and a selection of best-selling or new products.
- Includes a prominent search bar to encourage product discovery.
- Design focuses on clarity and visual appeal to engage visitors immediately.

3.3.2 Product Listing Page

- Shows products in a grid layout with images, titles, prices, and “Add to Cart” buttons.
- Includes filter and sort options (e.g., by category, price range, popularity).
- Pagination or infinite scroll enables easy browsing through large inventories.

3.3.3 Product Detail Page

- Provides high-resolution product images with zoom functionality.
- Displays detailed descriptions, specifications, and pricing.
- Shows stock availability dynamically from the database.
- Features prominent “Add to Cart” functionality and recommended products.

3.3.4 Shopping Cart Page

- Lists all items the user has added, including images, names, prices, and quantities.
- Allows users to update quantities or remove items.
- Displays real-time calculation of totals and estimated shipping costs.
- Includes clear call-to-action buttons for proceeding to checkout or continuing shopping.

3.3.5 Checkout Page

- Collects shipping address, contact information, and payment details.

- Validates form input to ensure completeness and correctness.
- Integrates with the database to generate an order record upon successful submission.
- Confirms order placement to the user with a clear success message.

3.3.6 User Registration and Login Pages

- Enable account creation with email verification and password hashing.
- Allow existing users to log in securely using session management or JWT tokens.
- Include password recovery and account management options.

3.3.7 Order Confirmation Page

- Displays a thank-you message and order summary.
- Provides users with order ID and estimated delivery timeline.
- Encourages continued engagement through links to browse more products.

3.4 Database Integration

The website is integrated with the **MySQL** database using a back-end API developed in **Node.js** with **Express**.

The Sequelize ORM is used to define models matching the database schema, simplifying CRUD operations and enforcing data consistency.

API endpoints allow the front-end to:

- Fetch product listings and individual details.
- Authenticate and register users.
- Manage the shopping cart for logged-in users.
- Process orders and save them with transaction integrity.

User passwords are hashed using **bcrypt** before storage to ensure security.

HTTPS is enforced to protect data in transit, complying with best practices for user privacy and security.

3.5 Justification of Features

The chosen features were implemented to match the client's business requirements as outlined in the introduction:

User-friendly navigation ensures customers can find products easily, increasing conversion rates.

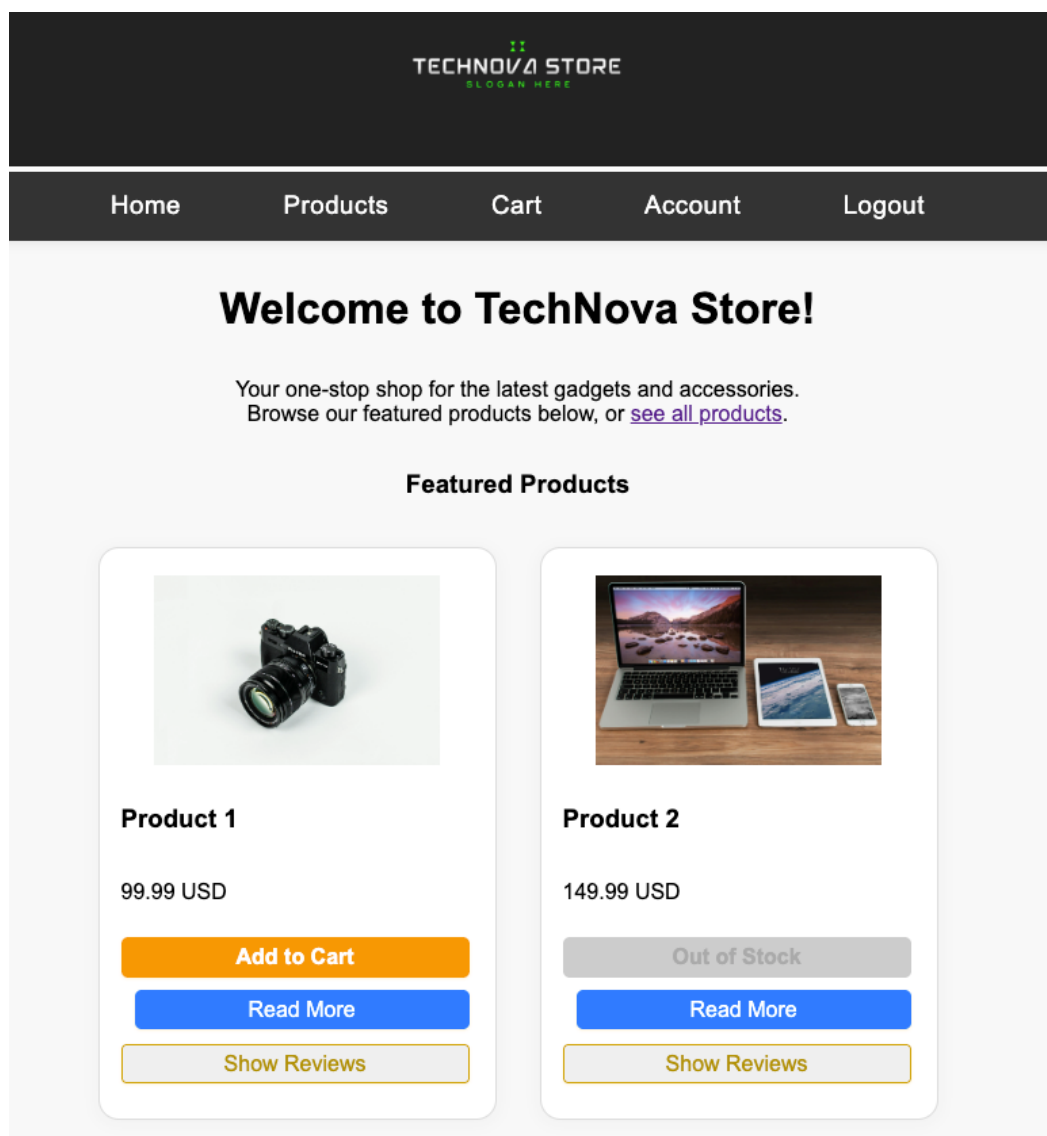
Detailed product pages provide transparency and build trust, reducing cart abandonment.

Shopping cart and checkout flows replicate familiar e-commerce patterns, reducing friction for users.

User account management enables order history and personalised experiences, promoting customer retention.

Responsive design ensures accessibility across devices, meeting modern consumer expectations.

Database integration ensures real-time accuracy for stock levels and order management.



DISCUSSION / EVALUATION AND SUMMARY / CONCLUSION

The **TechNova Store** project demonstrates the integration of modern web development techniques with robust database design to deliver a functional e-commerce solution. This section critically reflects on the development process, design choices, challenges encountered, and lessons learned.

4.1 Analysis of Design and Implementation

The decision to use a **relational database** ensured data integrity through foreign key constraints, normalisation, and ACID transactions. This was critical for managing products, users, orders, and cart contents in a consistent and reliable manner.

For the **front-end**, responsive design was a priority. By using CSS Flexbox/Grid and mobile-first principles, the website delivers a seamless user experience across devices. The consistent header and footer navigation structures enhance usability and encourage exploration.

Integration between the front-end and back-end was achieved through **RESTful API endpoints**, supporting clear separation of concerns and maintainability. The use of **Sequelize ORM** simplified database interactions and enforced schema definitions in the Node.js environment.

4.2 Challenges Faced and Solutions

Authentication and Security:

- *Challenge:* Storing passwords securely.
- *Solution:* Implemented **bcrypt** hashing, ensuring passwords are never stored in plaintext.

Data Consistency:

- *Challenge:* Avoiding race conditions during stock updates.
- *Solution:* Used transaction management in MySQL to enforce atomic operations.

Responsive Layouts:

- *Challenge:* Ensuring usability across different screen sizes.
- *Solution:* Employed media queries and tested on multiple device emulators.

Media Sourcing and Attribution:

- *Challenge:* Avoiding copyright issues.
- *Solution:* Used properly licensed stock images and credited sources in the References.

4.3 Link to Project Goals

All implemented features were chosen with the client's needs in mind:

- **Product catalog and detail pages** addressed the requirement for easy browsing.
- **Cart and checkout flows** mirrored industry standards, ensuring customer familiarity.
- **User accounts** enabled order history and personalisation.
- **Database integration** provided real-time stock management and secure order recording.

These design choices directly support the business goals of improving user engagement, reducing friction in purchases, and maintaining operational efficiency.

Summary and Conclusion

This project set out to design and implement a **full-stack solution for an e-commerce business**, demonstrating practical application of modern web development and database design principles.

The resulting system includes:

- A **responsive, user-friendly website** featuring product listings, detailed pages, a shopping cart, and checkout process.
- A **secure user account system** with login, registration, and order history functionality.
- A **MySQL relational database** designed for consistency, integrity, and scalability.
- Integration through a **RESTful API** built in Node.js with Express and Sequelize.

5.1 Limitations

While the project meets the key business requirements, some limitations remain:

- No live payment gateway integration (only simulated checkout).
- Basic email/password authentication without OAuth/social login options.
- Limited admin features (e.g., no product management dashboard).
- Limited product data used for demonstration.

5.2 Future Work and Improvements

Potential avenues for enhancement include:

- Integrating a real payment gateway like Stripe or PayPal.
- Expanding user authentication to include OAuth providers (Google, Facebook).
- Adding an admin dashboard for product and order management.
- Improving UI with advanced animations and transitions.
- Implementing server-side rendering or static site generation for SEO benefits.
- Scaling the database and API to support higher traffic loads.

5.3 Final Reflection

Overall, this project demonstrates the successful design and implementation of a full-stack e-commerce solution aligned with the **TechNova Store**'s business requirements. It showcases the ability to balance user experience, technical robustness, and secure data management in the context of a real-world application.

This project has reinforced the importance of careful planning, consistent design patterns, secure coding practices, and thorough testing in delivering professional-quality software.

REFERENCES / BIBLIOGRAPHY

References :

City, University of London (2025). *Individual Project Module Guide* (internal teaching materials).

Sequelize (2025). *Sequelize Documentation*. Available at: <https://sequelize.org/docs/v6/> (Accessed: 5 July 2025).

MySQL (2025). *MySQL Documentation*. Available at: <https://dev.mysql.com/doc/> (Accessed: 5 July 2025).

Express.js (2025). *Express.js Documentation*. Available at: <https://expressjs.com/> (Accessed: 5 July 2025).

ReactJS (2025). *React Documentation*. Available at: <https://react.dev/> (Accessed: 5 July 2025).

Font Awesome (2025). *Font Awesome Free Icons*. Available at: <https://fontawesome.com/icons> (Accessed: 5 July 2025).

Google Material Icons (2025). *Material Symbols*. Available at: <https://fonts.google.com/icons> (Accessed: 5 July 2025).

Canva (2025). *Canva Design Tool*. Available at: <https://www.canva.com/> (Accessed: 5 July 2025)