

**LA GRANDEE INTERNATIONAL COLLEGE**

**Simalchaur - 8, Pokhara, Nepal**

A Project Report on

“Electromart”

**Submitted to**

Bachelor of Computer Application (BCA) Program

In partial fulfillment of the requirement for the degree of BCA affiliate to

Pokhara University

**Submitted by:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Program, Semester** | **P.U. Registration Number** |
| Ganesh Tiwari | BCA, 8th Sem | 2021-1-53-0055 |
| Kapil Tiwari | BCA, 8th Sem | 2021-1-53-0059 |
| Prabhat Raj Poudel | BCA, 8th Sem | 2021-1-53-0066 |

Date: 2025-07-03

# Acknowledgement

We have presented this report focusing on the topic **“Electromart”**. This report has been prepared for partial fulfilment of the requirement for degree of BCA and to have practical experience.

We are heartily thankful to the faculty of BCA, **LA Grandee International College** and our supervisor **Mr. Sunil Sapkota** for his role to motivate and lead for this report. We obliged towards his constant guidance, supervision and feedbacks which enabled us to prepare a well-executed report.

Further, we express our gratitude to LA Grandee family, classmates, seniors and teachers who have directly and indirectly supported us during our report.

# Student's Declarations

**"Electromart"**

We hereby declare that we are the only authors of this work and that no sources other than the mentioned here we have been used in this. We assure you that the work we present here is unique to ourselves and resemblances to another similar project are purely coincidental.

……………………….

**Ganesh Tiwari (2021-1-53-0055)**

……………………….

**Kapil Tiwari (2021-01-53-0059)**

………………………..

**Prabhat Raj Poudel (2021-1-53-0066)**

Program: BCA 8th Semester

Date: 2025-07-03

# Supervisor's Declaration

I hereby declare that the project entitled **“Electromart”** has been carried out under my direct supervision by **Ganesh Tiwari, Kapil Tiwari and Prabhat Raj Poudel** during their sixth semester for the partial fulfilment of the requirements for the degree of BCA program under Pokhara University.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mr. Sunil Sapkota**

(Project Supervisor)

Date: 2025-07-03

****

…………………………..

**Er. Kiran K.C**

Principal

# Letter of Approval

We certify that we have examined this report entitled “**Electromart**” and are satisfied with the project defense. It is satisfactory in the scope and qualify as project in partial fulfillment of the requirements for the degree of BCA under Pokhara University.

Date: 2025-07-03

…………………………..

**Mr. Kundan Chaudhary**

Department Coordinator

………………………………….

**Er. Amit Parajuli**

External Examiner

…………………………..

**Mr. Sunil Sapkota**

Supervisor

# Abstract

Electromart is a dynamic eCommerce application developed to streamline the buying and selling of electronic gadgets. The platform provides an intuitive and responsive user interface, supporting key features such as product categorization, user roles (admin and customer), inventory management, secure payment integration, and order tracking. Electromart offers a wide selection of electronic devices, including smartphones, laptops, tablets, and accessories, aiming to fulfill the diverse needs of modern consumers.

The system enhances the shopping experience through real-time stock updates, advanced product search and filtering, customer reviews, and personalized product recommendations. These features are designed to increase customer satisfaction and operational efficiency for businesses. The development process emphasized creating a user-centric service while also applying full-stack development principles, user interface design strategies, and best practices in eCommerce. This hands-on approach provided valuable experience in building practical, real-world applications.

Beyond its current functionality, Electromart is built with scalability, performance, and security in mind. Its modular codebase allows for future expansion, such as adding promotional tools, loyalty rewards, multi-vendor support, and mobile application integration. Cross-browser compatibility and responsive design were also prioritized to ensure accessibility across all devices. By simulating real-world shopping scenarios, we validated the platform’s ability to handle business-critical operations. Overall, the project strengthened our technical, collaborative, and problem-solving skills, making Electromart a key milestone in our growth as software developers.

Table of Contents

[Acknowledgement ii](#_Toc202434448)

[Student's Declarations iii](#_Toc202434449)

[Supervisor's Declaration iv](#_Toc202434450)

[Letter of Approval v](file:///D:\8th%20sem\Project\electromart\documents\final_defence.docx#_Toc202434451)

[Abstract vi](#_Toc202434452)

[List of Tables viii](#_Toc202434453)

[List of Figures ix](#_Toc202434454)

[Abbreviation x](#_Toc202434455)

[1. Introduction 1](#_Toc202434456)

[2. Problem Statement 3](#_Toc202434457)

[3. Objective 4](#_Toc202434458)

[4. Background Study 5](#_Toc202434459)

[5. Requirement Document 7](#_Toc202434460)

[6. Methodology 12](#_Toc202434461)

[6.1. Iterative Model 12](#_Toc202434462)

[6.2. Work Assigned 15](#_Toc202434463)

[7. System Design 17](#_Toc202434464)

[7.1. ER Diagram 17](#_Toc202434465)

[7.2. Dataflow Diagram: 18](#_Toc202434466)

[7.3. UML Diagram 21](#_Toc202434467)

[7.4. Use Case Diagram 22](#_Toc202434468)

[8. Development Process 23](#_Toc202434469)

[9. Testing 25](#_Toc202434470)

[10. Project Result 27](#_Toc202434471)

[11. Future Enhancements 29](#_Toc202434472)

[12. Conclusion 30](#_Toc202434473)

[13. Reference 31](#_Toc202434474)

[14. Anex 33](#_Toc202434475)

List of Tables

[Table 5.1 - Requirement Matrix 11](#_Toc202385804)

[Table 6.1 - Work Assigned 16](#_Toc202385805)

[Table 9.1 - Test Case 26](#_Toc202385806)

List of Figures

[Figure 6.1 - Iterative Model 12](#_Toc202385797)

[Figure 7.1 - ER Diagram 17](#_Toc202385798)

[Figure 7.2 - DFD Level 0 18](#_Toc202385799)

[Figure 7.3 - DFD Level 1 19](#_Toc202385800)

[Figure 7.4 - DFD Level 2 20](#_Toc202385801)

[Figure 7.5 - UML Diagram 21](#_Toc202385802)

[Figure 7.6 – Use Case Diagram 22](#_Toc202385803)

Abbreviation

|  |  |
| --- | --- |
| **ACRONYM** | **FULL FORM** |
| API | Application Product Interface |
| APP | Application |
| AUTH | Authentication |
| BAAS | Backend as a Service |
| BCA | Bachelor of Computer Application |
| BNPL | Buy Now Pay Later |
| CD | Continuous Delivery |
| CI | Continuous Integration |
| CRUD | Create, Read, Update, Delete |
| CSRF | Cross-Site Request Forgery |
| CSS | Cascading Style Sheets |
| DB | Database |
| DFD | Data Flow Diagram |
| EMI | Equated Monthly Installment |
| ERD | Entity Relationship Diagram |
| FR | Functional Requirement |
| GIT | Global Information Tracker |
| HTTPS | Hyper Text Transfer Protocol Secure |
| ID | Identification |
| IOS | iPhone Operating System |
| IT | Information Technology |
| ITTI | Industrial Technology Transformation Index |
| JS | Java Script |
| LGIC | LaGrande International College |
| LTD | Limited |
| MS | Microsoft |
| NFR | Non-Functional Requirement |
| NPM | Node Package Manager |
| OS | Operating System |
| PU | Pokhara University |
| PVT | Private |
| SDK | Software Development Kit |
| SN | Serial Number |
| SQL | Structured Query Language |
| TS | TypeScript |
| UAT | User Acceptance Testing |
| UI | User Interface |
| UML | Unified Modeling language |
| UX | User Experience |
| XSS | Cross-Site Scripting |

# Introduction

The Gadget App E-Commerce project (Electromart) is a comprehensive solution for businesses selling electronic gadgets, designed to automate and streamline the online shopping experience. It provides a user-friendly interface with functionalities like managing product inventory, processing orders, handling secure payments, and generating sales and activity reports. The goal is to boost efficiency and profitability for businesses while ensuring a convenient and seamless customer experience. By eliminating manual processes, businesses can focus on growth and enhanced customer service.

The proposed application will offer a wide range of gadgets, including smartphones, tablets, laptops, smartwatches, and accessories, to meet diverse user needs. Advanced features such as real-time stock availability, user-friendly navigation, flexible order and return systems, and secure payment gateways will ensure a smooth shopping experience. This project aims to enhance business efficiency and profitability by automating and streamlining the e-commerce process. With this system in place, businesses can expand their digital sales and serve their customers more effectively.

Moreover, the app is built with scalability and performance in mind, using modern technologies to support high traffic and large product catalogs without compromising speed or reliability. It also integrates analytics tools to help businesses track user behavior, optimize marketing strategies, and make data-driven decisions. With mobile responsiveness and cross-platform compatibility, users can enjoy a consistent shopping experience across all devices. Overall, this project not only simplifies the e-commerce process but also empowers businesses with the tools they need to stay competitive in a rapidly evolving digital marketplace.

In today’s fast-paced digital world, eCommerce platforms must provide a seamless, engaging, and secure shopping experience to attract and retain customers. The Gadget App E-Commerce project addresses these demands by leveraging cutting-edge technologies such as TypeScript, Next.js, and React Native, which ensure robust performance, scalability, and cross-platform accessibility. This combination of technologies enables rapid development and deployment while maintaining high standards of code quality and user experience. By focusing on both business needs and customer satisfaction, this project aims to bridge the gap between technological innovation and practical commerce solutions.

# Problem Statement

* **Lack of an efficient and user-friendly platform** makes it difficult for gadget sellers to effectively manage their product listings, inventory, and overall e-commerce operations, leading to reduced productivity and customer dissatisfaction.
* **Inadequate systems for handling product and customer data** result in poor organization, limited accessibility, and ineffective data presentation, hindering a smooth and seamless shopping and management experience.

# Objective

The major goals of this project are as follows:

* To provide a user-friendly and efficient system that enables sellers to manage gadget products and e-commerce operations effectively and efficiently.
* To collect, store, organize, retrieve, and present product information and customer data for a seamless shopping and management experience.

# Background Study

Electromart is a cross-platform mobile application built using React Native, designed to provide seamless interaction for both buyers and sellers on Android and iOS devices. Its backend is powered by Node.js and Next.js, developed in TypeScript—offering a scalable, high-performance, and maintainable architecture. The system efficiently handles product management, order processing, and customer data storage in real time, enabling users to manage their business operations from anywhere.

To better understand the current landscape of gadget retail operations in our country, we studied existing local platforms such as ITTI Pvt. Ltd. and Nagmani International:

* ITTI Pvt. Ltd. offers a mobile application and web interface that showcases a wide range of electronic gadgets such as laptops, smartphones, and accessories. Their platform provides product listings with technical specifications, promotional offers, and customer support. However, the mobile experience still lacks full seller-side management tools like inventory tracking, real-time updates, and advanced analytics.
* Nagmani International primarily operates through a website, offering renowned brands like ASUS, MSI, and Acer. While the site supports bulk product listings and handles direct customer inquiries, it lacks a dedicated mobile application and offers limited features for real-time inventory and order management, especially for small vendors who want to use the platform for distribution.

In addition to these platforms, we visited a local gadget store to observe how small retailers manage their sales and inventory in real time. The store used a mix of an offline POS system and basic online listings. We noticed several challenges:

* The use of manual and disconnected tools made it difficult to sync inventory and order records efficiently.
* Staff struggled with frequent errors and confusion while tracking stock and customer orders.
* The absence of centralized systems meant that data entry was repetitive and time-consuming.

From these insights, we identified a clear opportunity to develop a **comprehensive and customizable solution** for gadget retailers. Our application is intended to address these gaps by providing:

* A mobile-first system where sellers can easily **add, edit, or remove products**, manage inventory, and process orders.
* A structured database for **real-time storage and retrieval** of customer and product data.
* Advanced features like **search filters by products and categories**, and **reporting tools** to track sales and performance.
* An intuitive user interface designed to reduce complexity for store staff and improve the shopping experience for customers.

Ultimately, the Gadget App is designed to empower small and mid-sized electronics businesses by improving operational efficiency, reducing human error, and offering a more modern, digital approach to retail management in the growing e-commerce landscape of our country.

# Requirement Document

This requirement documentation outlines the functional, non-functional requirements and the tools and technologies required for the **Electromart**.

**Functional Requirements**

* **User Registration and Authentication**  
  Enable users to sign up and log in securely using email/password or third-party providers (Google, Apple, etc.) using Supabase Auth.
* **User Profiles**  
  Each user (customer/admin) will have a profile to manage personal data, order history, preferences, and saved items.
* **Product Listings and Categories**  
  Admins can add, update, and manage gadgets with images, descriptions, specs, prices, and availability. Products will be categorized (e.g., Mobile, Laptop, Accessories).
* **Search and Filter**  
  Users can search for products by name, category, brand, or price range and apply filters for sorting results (by popularity, price, ratings).
* **Product Details Page**  
  Users can view detailed information, reviews, and related products for each item.
* **Cart Management**  
  Users can add/remove products in the cart, update quantities, and proceed to checkout.
* **Seamless Payment Integration**

asd

* **Wishlist**  
  Users can add products to a Wishlist for later viewing or purchasing.
* **Order Placement and Management**  
  Users can place orders, choose delivery addresses, and payment methods. Admins can manage orders (status updates, cancellations).
* **Ratings and Reviews**  
  Users can rate products and leave reviews; admins can moderate the reviews.
* **Dashboard (Admin Panel)**  
  Admin interface for managing products, orders, user accounts, and analytics.

**Non-Functional Requirements**

* **Cross-Platform Compatibility**  
  The application must work seamlessly on iOS, Android (mobile), and desktop (web via Next.js).
* **Scalability & Performance**  
  Ensure high performance and scalability using serverless architecture with Supabase and optimized rendering with Next.js.
* **Clean Code Architecture**  
  Use modular, reusable TypeScript components with proper folder structures and state management (e.g.: Zustand).
* **Responsive and Intuitive UI/UX**  
  Provide a user-friendly experience using responsive designs, accessible components, and consistent branding.
* **Security**  
  Implement secure authentication, protect customer data, and prevent common threats.
* **Future Growth**  
  Architecture should support adding new features like subscription plans, promotions, or affiliate systems without major restructuring.

**Tools and Technologies**

**Development Tools**

* **IDE:** Visual Studio Code
* **Version Control:** Git
* **Repository Hosting:** GitHub
* **Database & Backend as a Service (BaaS):** Supabase (PostgreSQL, Auth, Realtime)

**Frontend**

* **Mobile App:** React Native Expo & Typescript
* **Web App:** Next.js with TypeScript
* **UI Libraries:** CSS, shadcn UI & Recharts for fast and consistent UI components

**Package Managers**

* **npm:** For dependency and package management
* **npx:** For package execution

**Libraries**

* **zod:** For typescript validation & schema declaration

**Other Tools**

* **Analytics:** Supabase Analytics
* **Payment Integration:** Stripe

**Requirement Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement  ID | Requirement Description | Type | Priority | Status |
| FR-01 | Users can securely register and log in using credentials. | Functional | High | Completed |
| FR-02 | Users can browse products by category and search/filter items | Functional | High | Completed |
| FR-03 | Users can view product details, specs, images, and reviews | Functional | High | Completed |
| FR-04 | Users can add/remove items in the cart and update quantity | Functional | |  | | --- | | High | | |  | | --- | | Completed | |
| |  | | --- | | FR-05 | | Users can view and update their personal information. | |  | | --- | | Functional | | Medium | |  | | --- | | Completed | |
| FR-06 | Users can place orders with delivery and payment information | Functional | High | Completed |
| FR-07 | Users can write and view product reviews | Functional | Medium | Completed |
| FR-08 | Admins can manage product listings, inventory, and orders | Functional | High | Completed |
| NFR-01 | Supports real-time sync of categories and products with automatic updates. | Non-Functional | High | Completed |
| NFR-02 | UI should be intuitive and user-friendly using reusable TypeScript components | Non-Functional | High | Completed |
| NFR-03 | Ensures consistent functionality across major browsers | Non-Functional | High | Completed |
| NFR-04 | User data and payment information must be securely stored (Supabase + HTTPS) | Non-Functional | High | Completed |
| NFR-05 | Offers intuitive navigation for better user experience. | Non-Functional | Medium | Completed |

Table . - Requirement Matrix

# Methodology

The Iterative Model is a software development approach that involves repeating a sequence of steps—such as planning, designing, implementing, and testing—until the desired level of quality is achieved. This model is used for the development of this Android-based e-commerce application, as it ensures that each iteration is thoroughly tested, reviewed, and improved. By incorporating feedback in each cycle, the development team can continuously enhance the app’s performance, usability, and functionality, leading to a more reliable and user-friendly final product.

## Iterative Model



Figure . - Iterative Model

For the development of Electromart, we followed the Iterative Model, completing the project through multiple development cycles. In each iteration, we planned, developed, tested, and improved based on the outcome of the previous phase.

Here’s what we actually did during each phase:

* + 1. **Requirement Gathering:**

We began the project by identifying and listing all the essential features required for Electromart. These included products browsing by categories, shopping cart functionality, order placement and tracking, user login and registration, and integration of a secure payment system. After finalizing the feature set, we selected our technology stack: TypeScript was used for both the mobile application (developed using React Native) and the admin panel, ensuring consistency across the frontend components. As our database and authentication solution, we chose Supabase, which provided a reliable PostgreSQL backend along with built-in support for user management, real-time data handling, and secure access control.

* + 1. **Analysis:**

In the analysis phase, we carefully studied the listed features and broke them down into detailed components. We defined two main user roles: admin and user each with specific access levels and functionality. During this stage, we identified key technical needs such as implementing secure user authentication and enabling real-time stock updates. To support these requirements, we explored Supabase’s built-in features like authentication, real-time database syncing, and PostgreSQL querying capabilities, which aligned well with our project needs.

* + 1. **Design:**

During the design phase, we focused on both the visual and structural aspects of the app. We used Figma to design the complete UI for the mobile application. On the backend side, we designed the necessary PostgreSQL database tables within Supabase, covering entities such as users, products, orders, and carts. In parallel, we planned and documented the API routes using Next.js with TypeScript, outlining how data would flow between the frontend and backend. We also organized the codebase with a well-structured folder hierarchy, promoting clean separation of logic and easier maintainability.

* + 1. **Coding:**

In the coding phase, we began development in multiple iterations, breaking the work into manageable modules. The mobile application was developed using React Native with TypeScript, where we created screens for user login, signup, product listing, product details, cart, and order history. Simultaneously, we developed backend APIs using Next.js and TypeScript to handle user authentication, product management, and order processing. We successfully integrated Supabase for database operations and user authentication which allowed for efficient handling of data and business logic.

* + 1. **Testing:** After each development iteration, we carried out manual testing on various Android devices to ensure functionality and user experience. We verified major processes like login, cart operations, product loading, and overall app responsiveness. Particular attention was given to testing API integration and checking Supabase-powered features such as real-time database updates and secure login. Any UI bugs, API errors, or performance issues were promptly identified and resolved before moving on to the next cycle, which helped maintain a stable and usable product throughout the development process.
    2. **Implementation:**

Once all major features were completed and thoroughly tested, we proceeded with the implementation phase. We used Supabase to manage the PostgreSQL database and handle user authentication seamlessly. The mobile application was built successfully, and The APK was generated using Expo with the Metro bundler and shared among team members for internal testing and demonstration. This phase allowed us to evaluate the app’s overall performance, ensure all modules worked together effectively, and simulate real-world usage within a controlled environment.

* + 1. **Review:**

In the final phase, after deployment, we gathered feedback from initial users who tested the app. Based on the input, we worked on improving key features such as product search, screen transitions, and handling of edge cases during login or checkout. We released minor updates that included UI enhancements and additional filtering options. Throughout the process, we maintained our codebase using GitHub, where we followed basic version control practices like branch management and regular commits to ensure the project remained organized and easy to collaborate on.

## Work Assigned

We are a group of 3 students were confined to the development of the final year project and here is the work division chart among us:

|  |  |  |  |
| --- | --- | --- | --- |
| S.N. | Name | Work Assigned | Remarks |
| 1. | Ganesh Tiwari | Week 1: Project Setup, Home Screen UI, Login/Signup Screen Week 2: UI for Authentication Week 3: Product Listing Page, Filter UI Week 4: Cart UI Week 5: Checkout Page UI Week 6: Admin Dashboard UI & UI Enhancements Week 7: UI Bug Fixes & Responsiveness Testing | Week 1: Satisfied  Week 2: Satisfied  Week 3: Refinement  Week 4: Satisfied  Week 5: Refinement  Week 6: Satisfied  Week 7: Satisfied |
| 2. | Kapil Tiwari | Week 1: Database Schema (Users, Products, Orders) Week 2: Supabase Auth Setup, Role Config Week 3: Product CRUD APIs (Create, Read, Update, Delete) Week 4: Cart & Wishlist API (Relations & Joins) Week 5: Order API & Mock Payment Flow Week 6: Admin Product Management APIs Week 7: Error Handling & Backend Testing | Week 1: Satisfied  Week 2: Satisfied  Week 3: Refinement  Week 4: Refinement  Week 5: Refinement  Week 6: Satisfied  Week 7: Satisfied |
| 3. | Prabhat Raj Poudel | Week 1: Requirement Specification, Tech Stack Documentation Week 2: Auth Flow Diagram & API Contracts Week 3: UI/User Flow Documentation Week 4: Sequence Diagrams, Test Case Drafts Week 5: Deployment Plan, Flow Charts Week 6: Admin Guide, Bug Reporting Week 7: Final Report, README, Presentation Slides | Week 1: Satisfied  Week 2: Satisfied  Week 3: Refinement  Week 4: Satisfied  Week 5: Refinement  Week 6: Satisfied  Week 7: Satisfied |

Table . - Work Assigned

# System Design

System design is the process of planning, structuring, and organizing the components of an application to meet specific business or user needs. It involves outlining how different modules of the system interact, the flow of information, and how users engage with the application. In **Electromart**, an e-commerce app focused on gadgets, commonly used tools in system design include **Data Flow Diagrams (DFDs)**, **Flowcharts**, and **ER Diagrams**.

## ER Diagram

An ER diagram shows the relationships between various entities in a database. Each entity represents a real-world object like a user, product, or order, and contains attributes. The ER diagram helps in designing the database structure efficiently and maintaining data consistency and integrity throughout the app.

Figure . - ER Diagram

**Admin Adds Product**

Admin users can upload gadgets with details like Product Name, Description, Price, Quantity, Brand, Specifications, and Images.

**User Browses & Orders**

Users can browse the product catalog, view gadget details, and add items to their cart. Once confirmed, they place orders by providing delivery information and making payments.

**User Wishlist**

Users can add their favorite gadgets to the Wishlist for future consideration.

## Dataflow Diagram:

A **Data Flow Diagram (DFD)** visually represents how data moves through the Electromart system. It outlines user inputs (such as placing orders or signing up), processes (like order processing and payment verification), data stores (like user info or product lists), and external systems (like payment gateways or shipping APIs). This helps understand the overall functionality and interaction between components.

**DFD Level 0**

This Level 0 DFD shows a high-level overview of the **Electromart** system, capturing interactions between the **Customer**, **Admin**, and the **System**. The central process involves browsing products, placing orders, managing payments, and maintaining product listings. It provides a simplified snapshot of how the user and admin interact with the e-commerce platform.



Figure . - DFD Level 0

**DFD Level 1**

Level 1 DFD expands on the core system to reveal more detail on sub-processes. It breaks the Electromart system into the following sub-processes:

* **Browse & Search Products:** Customers explore available gadgets.
* **Add to Cart / Wishlist:** Customers store items temporarily or for later.
* **Place Order:** Finalizing order with delivery info and payment.
* **Manage Products:** Admin adds/updates/deletes products.
* **Manage Orders:** Admin tracks and updates order status.
* **Process Payment:** Payments are processed and verified.

This level shows how data flows between customers, system processes, and data stores like the product catalog, order history, and user profiles.



Figure . - DFD Level 1

**DFD Level 2 (From the perspective of Customer)**

Level 2 DFD explores deeper into how the **Customer** interacts with the system. It focuses on the user’s flow from browsing to purchase:

* **Search Gadget → View Gadget Details → Add to Cart / Wishlist**
* **Proceed to Checkout → Enter Address → Make Payment**
* **Track Orders → View Order History**

Each process connects with related data stores like the product list, Wishlist data, and payment details. This level of detail is helpful in understanding the customer experience journey in the Electromart app.



Figure . - DFD Level 2

## UML Diagram

Figure . - UML Diagram

## Use Case Diagram



Figure . - Use Case Diagram

# Development Process

The development of the Electromart mobile e-commerce application followed the Iterative development model, allowing for incremental progress through multiple cycles of planning, development, testing, and refinement. This approach provided flexibility to adapt to changing requirements and incorporate feedback throughout the project lifecycle, ultimately resulting in a polished and user-centric product.

**Iterative Planning and Execution**

The development process was divided into two-week iterations (sprints). Each sprint focused on a clearly defined set of features or improvements, with goals aligned to user needs and project priorities. At the start of each sprint, the team conducted planning sessions to define tasks, estimate effort, and assign responsibilities. At the end of each cycle, retrospectives were held to reflect on progress, address bottlenecks, and continuously enhance team performance.

**Frontend and Backend Development**

Development was conducted in parallel across the frontend and backend:

* The frontend was built using Flutter to ensure a responsive, cross-platform UI/UX.
* The backend was developed with scalable architecture, utilizing cloud services and secure data handling via tools like Supabase.
* APIs were integrated for payment processing, product search/filtering, and user authentication.
* Key Features Implemented
* Major features developed during the iterative process included:
* Product listing with search and filter options
* User registration, login, and profile management
* Cart management and secure checkout
* Admin dashboard for product and order management
* Review and rating system for user feedback
* Continuous Testing and Integration

Each sprint cycle included unit testing, integration testing, and user acceptance testing (UAT). This helped ensure the app met both functional and non-functional requirements.

We adopted Continuous Integration (CI) practices using GitHub Actions, which enabled frequent code commits, automated builds, and real-time issue detection—ensuring a stable, deployable app throughout the development cycle.

**Team Collaboration and Communication**

Daily stand-up meetings and collaborative tools (e.g., Slack, Trello, GitHub) supported transparent communication and rapid issue resolution. Cross-functional collaboration between developers, designers, and testers played a key role in maintaining development velocity and product quality.

# Testing

Regardless of the development methodology used, the goal of testing is to ensure that the developed application functions properly. It is an essential step in maintaining the quality and reliability of software. To ensure the quality of the product, testing has been integrated into each step of the development process.

To detect as many bugs or errors as possible, test cases are carefully designed to target areas where problems are most likely to occur. This approach helps in making the application more robust. Accordingly, a set of test cases has been created and executed to verify that the system meets its functional requirements.

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE  ID | TEST CASE | EXPECTED RESULT | TEST  RESULTS |
| TC-01 | Provide valid user login credentials. | User successfully logged in. | PASS |
| TC-02 | Enter invalid user login credentials. | Display error message. | PASS |
| TC-03 | Register a new user account. | User successfully registered and redirected to dashboard. | PASS |
| TC-04 | Browse available gadgets using filters (category product name). | Gadgets filtered and displayed correctly. | PASS |
| TC-05 | Add gadget to shopping cart. | Item added to cart successfully. | PASS |
| TC-06 | Proceed to checkout with items in cart. | Redirected to checkout page with correct item details. | PASS |
| TC-07 | Make payment using available payment options. | Payment processed successfully. | PASS |
| TC-08 | View order history and details. | Order history displayed correctly. | PASS |
| TC-09 | Search for a gadget using the search bar. | Search results displayed based on entered keywords. | PASS |
| TC-10 | Access site from different devices (mobile, tablet, desktop). | Responsive layout loads correctly on all devices. | PASS |

Table . - Test Case

# Project Result

After completing the initial development and testing phases, our e-commerce application **“Gadget App”** has successfully met its core objectives. The app now functions as a robust platform for browsing, purchasing, and managing electronic gadgets. While additional advanced features are planned for future releases, the current version offers a complete and reliable shopping experience for both users and business owners.

* 1. **What are the functions achieved by our project?**

**Gadget App** delivers a comprehensive set of features tailored to the needs of customers and e-commerce businesses in the electronics sector:

* **Product Catalog and Search**

Users can explore a wide range of gadgets including smartphones, laptops, tablets, smartwatches, and accessories. The product catalog supports smart search, sorting, and filtering by brand, price, and specifications.

* **Cart and Checkout System**

The app allows users to add items to the cart and proceed with a smooth checkout process. It supports discount codes, multiple delivery options, and preview of the final invoice before payment.

* **Secure Payment Gateway Integration**

Multiple payment options including credit/debit cards, digital wallets, and bank transfers are supported through secure and encrypted gateways, ensuring safe and fast transactions.

* **User Profile and Order Management**

Users can manage their profiles, view past orders, download invoices, and track shipment status in real-time, enhancing transparency and customer control.

* **Mobile Responsiveness and Cross-Platform Support**

The app offers a seamless user experience across devices including smartphones, tablets, and desktops, thanks to responsive design and cross-platform development using React Native and Next.js.

These features combine to create a powerful and user-friendly e-commerce solution that supports both customers and administrators efficiently.

* 1. **What problems have been solved by the project?**

**Gadget App** addresses several key pain points commonly experienced in traditional and digital gadget shopping:

* **Lack of a Unified Gadget Shopping Platform**

Previously, users had to visit multiple sites to compare and purchase gadgets. Gadget App consolidates product discovery, comparison, and checkout into one streamlined platform.

* **Manual Inventory and Order Tracking**

Businesses often struggled with inventory management and manual order handling. This app automates inventory updates, order processing, and status tracking—reducing errors and delays.

* **Unsecure Payment Methods**

Many platforms lacked secure and diverse payment options. Gadget App ensures safe transactions with industry-standard encryption and multiple verified payment gateways.

By solving these real problems, **Electromart** helps customers shop for electronics more efficiently and securely while enabling businesses to manage operations with greater accuracy and speed. It encourages a more streamlined, convenient, and trusted online shopping experience, empowering users to make informed decisions and businesses to thrive in the competitive digital marketplace.

# Future Enhancements

Future enhancements in **Electromart** refer to potential improvements and added functionalities that can significantly enhance user experience, performance, and business scalability. As customer needs evolve and technology advances, these enhancements aim to ensure that Electromart remains competitive, user-friendly, and feature-rich. Below are five key enhancements planned for future development:

* **Expansion of Product Categories**

Expand its catalog to include more electronic categories such as home automation devices, gaming accessories, personal care electronics, and refurbished gadgets.

* **AI-Driven Smart Recommendations**

Integrate AI algorithms to offer personalized product suggestions based on users’ browsing behavior, purchase history, and reviews.

* **Real-Time Order Tracking**

Implement real-time GPS tracking for order deliveries, allowing customers to monitor their package location and estimated delivery time which enhances customer trust and transparency in the shipping process.

* **In-App Customer Support Chat (with Video/Voice Options)**

Enable real-time chat support with options for voice or video communication to help users resolve technical queries or product-related concerns.

* **Multiple Payment and Financing Options**

Offer a range of secure payment modes, including digital wallets, EMI (installment) plans, and Buy Now Pay Later (BNPL) services.

# Conclusion

The development of the **Electromart** eCommerce gadget application using TypeScript, Next.js, and React Native has been a comprehensive and insightful journey that successfully addressed the challenges of building a modern, scalable, and user-friendly platform. This project demonstrates the effective integration of cutting-edge technologies to deliver a seamless shopping experience across both web and mobile platforms.

TypeScript played a critical role in enhancing the codebase by providing static typing, which helped reduce runtime errors and improved maintainability. Its compatibility with JavaScript and modern frameworks allowed us to write clean, robust, and scalable code. Next.js contributed significantly by enabling server-side rendering and static site generation, which improved the app’s performance, SEO capabilities, and overall user experience. The framework’s routing and API handling features simplified backend integration and enhanced development efficiency.

React Native empowered the creation of a cross-platform mobile application, enabling us to deliver a consistent and performant experience on both Android and iOS devices without the need for maintaining separate codebases. This choice greatly optimized development time and resource allocation, while still providing native-like app capabilities.

Throughout the project, emphasis was placed on key eCommerce functionalities such as product categorization, real-time inventory management, secure payment processing, and order tracking, all designed to meet the expectations of modern consumers. Features like user authentication, product reviews, and personalized recommendations were also integrated to enhance user engagement and trust.

In conclusion, the Electromart project not only fulfills its primary goal of providing an efficient, secure, and user-friendly eCommerce platform for electronic gadgets but also represents a valuable learning experience. It stands as a testament to the power of modern web and mobile technologies in transforming digital commerce, preparing the development team to contribute effectively in the fast-evolving tech industry.

# Reference

Rouse, M. (2017). *Mobile application development (mobile app development)*. TechTarget. <https://www.techtarget.com/searchsoftwarequality/definition/mobile-application-development>

ITTI Pvt. Ltd. (n.d.). *ITTI Computers - Your Trusted IT Training Institute in Nepal*. Retrieved May 7, 2025, from <https://www.itti.com.np>

Nagmani International. (n.d.). *Official distributor of ASUS and other tech brands in Nepal*. Retrieved May 7, 2025, from <https://www.nagmani.com.np>

Alahmari, M., & Khalil, I. (2021). Building scalable eCommerce web applications with Next.js and React. *International Journal of Computer Science and Network Security*, 21(4), 45–52. https://doi.org/10.22937/IJCSNS.2021.21.4.6

Banks, A. (2020). *React Native in Action* (1st ed.). Manning Publications.

Bierman, G., Abadi, M., & Torgersen, M. (2014). Understanding TypeScript. *Proceedings of the ACM on Programming Languages*, 1(ICFP), 1–20. https://doi.org/10.1145/2628136

Chen, J., & Lin, Y. (2022). Leveraging Next.js for server-side rendering in eCommerce platforms. *Journal of Web Engineering*, 21(2), 115–132. https://doi.org/10.1145/3456789

Gravina, D., & Mastroeni, L. (2021). Building cross-platform mobile apps with React Native and TypeScript. *Software: Practice and Experience*, 51(12), 2665–2681. https://doi.org/10.1002/spe.2961

Hossain, M., & Islam, M. (2020). A study on UI/UX design principles for eCommerce mobile applications. *International Journal of Computer Applications*, 175(9), 15–22. https://doi.org/10.5120/ijca2020919677

Kumar, R., & Singh, A. (2021). Secure payment integration strategies in modern eCommerce apps. *International Journal of Advanced Research in Computer Science*, 12(5), 65–73. https://doi.org/10.26483/ijarcs.v12i5.7043

Li, Q., & Wang, Y. (2021). Real-time inventory management in online retail systems. *Journal of Retailing and Consumer Services*, 58, 102300. https://doi.org/10.1016/j.jretconser.2020.102300

Rajput, N., & Patil, S. (2022). Implementing product recommendation engines in eCommerce platforms using React. *International Journal of Engineering Research & Technology*, 11(3), 144–150. https://doi.org/10.17577/IJERTV11IS030218

Singh, P., & Sharma, R. (2020). Mobile-first approach for eCommerce application development using React Native. *International Journal of Computer Science Trends and Technology*, 8(1), 50–58.

# Annex

Once the system is created, there are certain criteria it'll need to fulfill in order to be called a successful system. So, this app is built with features such as user account management, product browsing, cart and checkout functionality, and admin panel for inventory and order management. Additionally, it includes support for vendor accounts, product listings, and customer reviews. When we will complete our project, we will provide the following things:

A functional and user-friendly Android application that allows users to access the app’s features and services on their Android devices. Seamless integration with Android-specific features and APIs, such as push notifications, location services (for delivery), and secure payment gateways. Provides features for browsing and searching electronic gadgets, adding items to cart, placing orders, tracking order status, managing user profiles, and leaving reviews and ratings for products and vendors.







