

RETAIL INNOVATIONS LTD

Task 1A(i): Research

Existing Solutions • Technology Review • Findings

DPDD Occupational Specialism — Set Task

Document Version 1.0 | January 2025

1. Introduction

Before designing a digital solution for Retail Innovations Ltd, it is important to research existing platforms and technologies to understand what works well in the retail industry, what features customers expect, and what technology choices would be most suitable for this project. This research will directly inform the proposal and justify the decisions made throughout the design and development process.

The research focuses on three areas that align with the client's key features: product search and filtering, retail analytics dashboards, and customer loyalty programmes.

2. Existing Retail Platforms

2.1 Amazon — Product Search and Filtering

Amazon is one of the world's largest e-commerce platforms and is widely considered the industry standard for product search. Its search system allows customers to type a product name, keyword, or product code and see results instantly. The results can then be filtered by category, price range, customer rating, brand, availability, and many other options.

Amazon's search also includes features such as auto-suggest (showing possible results as the user types), spelling correction, and sorting options (relevance, price low to high, customer reviews, newest arrivals). Product results are displayed with thumbnail images, prices, star ratings, and availability information, making it easy for customers to compare options without clicking into each product.

Strengths: The instant search with auto-suggest reduces typing. The wide range of filters allows users to narrow down results quickly. Thumbnail images and prices on the results page help customers make fast decisions.

Weaknesses: The interface can feel cluttered due to the number of filters and sponsored results. The platform is very complex and not suitable to replicate on a small scale.

Relevance to Retail Innovations: The core principle of real-time search with category filtering is directly applicable. The Retail Innovations platform should offer a search bar that filters as the user types, combined with a category dropdown. Thumbnails and pricing in the results table are also features worth adopting.

2.2 Shopify Admin Dashboard — Retail Analytics

Shopify is a popular e-commerce platform that provides retailers with an analytics dashboard as part of its admin panel. The dashboard displays key performance indicators (KPIs) such as total sales, total orders, online store sessions, returning customer rate, and average order value. These are shown as large numbers with trend comparisons to the previous period.

Shopify also includes visualisations such as line charts for sales over time, bar charts for top-selling products, and breakdowns by traffic source, location, and device. The dashboard is designed for non-technical retail business owners, so the information is presented in a simple, visual format without requiring any data analysis knowledge.

Strengths: The KPI cards are clear and easy to understand at a glance. The use of charts makes trends visible without needing to read raw numbers. The dashboard is designed for people without technical backgrounds.

Weaknesses: The free version has limited analytics. More detailed reports require a paid plan. The dashboard cannot be customised by the retailer.

Relevance to Retail Innovations: The KPI card layout is an effective pattern for the Retail Innovations dashboard. Showing totals for products, customers, orders, and revenue as large stat cards, combined with simple charts for category distribution and customer tiers, follows the

same approach. The brief specifically says the dashboard should present data in a 'simple, easy-to-understand format', which matches this approach.

2.3 Boots Advantage Card — Loyalty Programmes

The Boots Advantage Card is one of the UK's most well-known loyalty programmes. Customers earn 4 points for every £1 spent, and points can be redeemed in-store or online. The system uses a tiered approach where customers unlock additional benefits (such as exclusive offers and early access to sales) based on their spending level.

Boots displays the customer's points balance prominently in the app and on the website. Customers can see available rewards, browse what they can redeem their points for, and track their earning history. The programme also sends personalised offers based on purchase history.

Strengths: The tier system encourages continued spending. Displaying the points balance prominently keeps customers engaged. The simple earn-and-redeem model is easy to understand.

Weaknesses: The redemption value is relatively low (4 points = 1p). The programme requires significant backend infrastructure. Smaller retailers may struggle to offer meaningful rewards.

Relevance to Retail Innovations: A tiered loyalty system (Bronze, Silver, Gold, Platinum) with visible point tracking is appropriate for the Retail Innovations platform. The brief asks for 'predefined reward structures', which can be modelled on the Boots approach of different reward types at different point thresholds.

2.4 Tesco Clubcard — Loyalty Programmes (Comparison)

Tesco Clubcard is another major UK loyalty programme. It uses a simpler structure: customers earn 1 point per £1 spent and receive vouchers based on accumulated points. Tesco also offers 'Clubcard Prices' — exclusive lower prices available only to loyalty members, which encourages sign-ups.

Strengths: Very simple model that customers understand easily. Clubcard Prices provide immediate, visible benefit without needing to accumulate many points.

Weaknesses: Less tiered than Boots — all members get the same level of benefits. Less personalisation in rewards.

Relevance to Retail Innovations: Confirms that loyalty programmes work best when the benefit is clear and immediate. The Retail Innovations platform should make reward values visible (e.g., '10% Off for 500 points') so customers understand exactly what they earn.

3. Technology Research

3.1 Frontend Technology Options

Technology	Description	Pros	Cons	Suitability
HTML5 + CSS3 + Vanilla JS	Standard web technologies with no framework dependencies	No build tools needed. Runs in all browsers. Easy to deploy. Demonstrates core skills.	Larger apps can become harder to maintain. No component system.	HIGH — Ideal for this project scale. No complexity overhead.
React	JavaScript library for building user interfaces using components	Reusable components. Large ecosystem. Virtual DOM for performance.	Requires Node.js, npm, and build tools. Learning curve. Overkill for small projects.	MEDIUM — Powerful but adds unnecessary complexity for this project.
Vue.js	Progressive JavaScript framework for building UIs	Gentle learning curve. Single-file components. Good documentation.	Still requires build tools for full features. Smaller job market than React.	MEDIUM — Good option but still adds framework dependency.
Bootstrap	CSS framework providing pre-built components and grid system	Fast prototyping. Consistent design. Mobile-first grid.	Generic look. Large file size. Hard to customise without overriding styles.	LOW — Limits design flexibility. Dark themes require extensive overrides.

Decision: HTML5, CSS3, and Vanilla JavaScript were selected. This combination requires no build process, works by opening the HTML file directly, and demonstrates understanding of core web technologies rather than framework-specific abstractions. The project scale does not justify the complexity of a framework.

3.2 Backend and Database Options

Technology	Description	Pros	Cons	Suitability
Supabase (PostgreSQL)	Open-source Firebase alternative with hosted PostgreSQL, auth, and REST API	Free tier. Built-in auth (bcrypt, JWT). Row Level Security. Auto-generated JS client. No server code needed.	Requires internet. Config step for URL/key. Limited free tier rows.	HIGH — Provides auth, DB, and security without a custom server.
Firebase (Google)	Google's Backend-as-a-	Real-time sync. Good	NoSQL is less structured than	MEDIUM — Good but NoSQL is less

	Service with NoSQL (Firestore)	documentation. Generous free tier.	SQL. No Row Level Security equivalent. Google lock-in.	suitable for relational retail data.
Custom Express + MySQL	Node.js server with MySQL database	Full control. Familiar SQL. No vendor lock-in.	Requires hosting. Must build auth, API, and security from scratch. Much more development time.	LOW — Too complex for this project. Requires server deployment.
JSON/localStorage	Client-side data storage using browser APIs	No backend needed. Instant. Simple.	No persistence between devices. No security. No multi-user. Not suitable for real applications.	VERY LOW — Not appropriate for a data-driven retail platform.

Decision: Supabase was selected because it provides authentication, a relational PostgreSQL database, Row Level Security, and a JavaScript client library — all without needing to write any server-side code. This significantly reduces development time and allows the project to focus on frontend functionality. The relational structure of PostgreSQL is well-suited to retail data (products, customers, orders, rewards) where entities have clear relationships.

3.3 Design Approach Research

Dark Theme: Research into modern web design trends shows that dark themes are increasingly popular for dashboard-style applications. Platforms like Spotify, Discord, and GitHub all offer dark mode. Dark themes reduce eye strain during extended use, make accent colours stand out more clearly, and create a modern, premium aesthetic. For a retail analytics dashboard that may be used by store managers for extended periods, a dark theme is appropriate.

Responsive Design: According to Statista (2024), over 60% of global web traffic comes from mobile devices. The platform must work on desktop, tablet, and mobile. Using CSS Grid and Flexbox with media queries (breakpoints at 900px and 600px) provides responsive layouts without requiring a CSS framework.

Single Page Application (SPA): An SPA architecture loads a single HTML file and uses JavaScript to switch between views. This eliminates page reloads, provides instant navigation, and reduces server requests. For a dashboard-style application with multiple panels (products, customers, orders, loyalty), tab-based navigation within an SPA is an effective pattern.

4. Key Findings and Summary

The research above leads to the following conclusions that will inform the proposal:

1. Product search should be real-time and client-side, filtering as the user types. A category dropdown alongside a search bar provides the two most useful filtering mechanisms, as demonstrated by Amazon's approach at a smaller scale.
2. The analytics dashboard should follow the KPI card pattern used by Shopify — large numbers for key metrics, supported by simple visual charts. The brief requires data on sales, customer behaviour, and inventory, which maps to: revenue (from orders), tier distribution (customer behaviour), and product counts by category (inventory overview).
3. The loyalty programme should use a tiered system (like Boots) with clearly visible reward values (like Tesco). Predefined reward structures with configurable point thresholds give retailers flexibility while keeping the system simple for customers.
4. Vanilla HTML/CSS/JS is the most appropriate frontend stack for this project, as it avoids unnecessary complexity and demonstrates core web skills.
5. Supabase is the best backend choice, providing authentication, database, and security features without a custom server. Its Row Level Security feature is particularly valuable for separating admin and customer data access.
6. A dark-themed SPA with responsive breakpoints is the most suitable design approach for a retail analytics platform, based on current industry trends and the target user base.