**A black and grey logo

Description automatically generated**

QuickFind APP

Design & Requirements Document

**Document Information**

|  |  |
| --- | --- |
| Document version | 1.2 |
| Release date | 18.09.2024 |
| Authors | Shane Fuller  Paschal Cooney |

Table of Contents

[1 Company Overview 3](#_Toc177636826)

[1.1 Executive Summary 3](#_Toc177636827)

[1.2 Mission Statement 3](#_Toc177636828)

[1.3 Vision Statement 4](#_Toc177636829)

[2 Application Overview 4](#_Toc177636830)

[2.1 Solution Overview 4](#_Toc177636831)

[3 APP Design Guidelines 5](#_Toc177636832)

[3.1 Brand Identity and Vision 5](#_Toc177636833)

[3.2 App Design Principles 5](#_Toc177636834)

[3.3 Typography 6](#_Toc177636835)

[3.4 Colour Palette 6](#_Toc177636836)

[3.5 UI Elements and Components 6](#_Toc177636837)

[3.6 Imagery and Product Displays 7](#_Toc177636838)

[3.7 Functional Features and Interaction Design 7](#_Toc177636839)

[3.8 Animations and Micro interactions 8](#_Toc177636840)

[3.9 Accessibility Guidelines 8](#_Toc177636841)

[3.10 Tone of Voice 8](#_Toc177636842)

[3.11 Error Handling and Notifications 9](#_Toc177636843)

[3.11.1 Template to Requirements Mapping 10](#_Toc177636844)

[3.12 How it Works 12](#_Toc177636845)

[3.13 QuickFind Site Map 14](#_Toc177636846)

[4 Application Functional Requirements 16](#_Toc177636847)

[4.1 FUNCTIONAL - Step 0 – Identity Management 16](#_Toc177636848)

[4.1.1 Register Page 16](#_Toc177636849)

[4.1.2 Sign-in Page (or Guest Access) 17](#_Toc177636850)

[4.1.3 Password Reset Request Page 17](#_Toc177636851)

[4.1.4 Profile Management Page 18](#_Toc177636852)

[4.1.5 App Logout 19](#_Toc177636853)

[4.2 FUNCTIONAL - Step 1 - Product Capture & Detection 20](#_Toc177636854)

[4.2.1 Search Page 20](#_Toc177636855)

[4.2.2 Analyse Page 20](#_Toc177636856)

[4.2.3 Detect & Select Page 21](#_Toc177636857)

[4.3 FUNCTIONAL - Step 2 - Product Search & Discovery 24](#_Toc177636858)

[4.3.1 Recommendations Page 24](#_Toc177636859)

[4.4 APPLICATION - Step 4 - Product Selection & Exploration 26](#_Toc177636860)

[4.4.1 Products Page 26](#_Toc177636861)

[4.5 APPLICATION - Step 5 - Product Purchase 30](#_Toc177636862)

[4.5.1 Cart Page 30](#_Toc177636863)

[5 Profile & Footer Icon Navigation 32](#_Toc177636864)

[5.1 Profile Icons 32](#_Toc177636865)

[5.2 Footer Icons 32](#_Toc177636866)

[6 Email Templates 34](#_Toc177636867)

[7 Screen Design Samples 35](#_Toc177636868)

# Company Overview

## Executive Summary

QuickFind is set to transform the online fashion shopping experience by addressing the inefficiencies and challenges that both consumers and e-commerce stores face in the current digital landscape. Despite significant advancements in internet usage over the past two decades, the process of searching for and discovering fashion products online has remained stagnant. Major platforms like Amazon and Google offer limited shopping related AI-based capabilities, often prioritising their own ecosystems over consumer needs.

The global e-commerce market, which exceeded $6.1 trillion in 2023, continues to grow rapidly, with fashion accounting for 25% of this market. However, the shopping experience has become increasingly overwhelming, with consumers expressing frustration over the time-consuming nature of finding the right products. Additionally, e-commerce stores, especially small and medium-sized ones, are grappling with rising customer acquisition costs and low conversion rates, as traditional marketing channels become less effective.

QuickFind aims to address these issues by leveraging state-of-the-art AI technology to create a mobile-first, user-centric platform that simplifies product discovery. Users can search for fashion items using images or natural language, and QuickFind’s AI engine will deliver precise recommendations from a vast database of 139 million products across 369,000 stores from Shopify English speaking stores alone (EU, UK, Australia, Canada and US). This approach not only enhances the consumer shopping experience but also drives higher conversion rates and lowers customer acquisition costs for e-commerce stores.

QuickFind’s innovative solution is poised to redefine the online shopping landscape, offering a seamless, efficient, and personalised experience that benefits both consumers and retailers. With a focus on transparency, simplicity, and continuous improvement, QuickFind is positioned to become a leader in the digital shopping space, driving sustainable growth and setting a new standard for e-commerce.

## Mission Statement

To revolutionise online fashion shopping by harnessing the power of advanced AI technology to deliver a personalised, seamless and consumer focused experience.

## Vision Statement

Become the world's leading AI-driven fashion discovery platform that empowers consumers and drives sustainable growth for e-commerce businesses globally.

# Application Overview

## Solution Overview

QuickFind is a cutting-edge platform designed to revolutionise the online fashion shopping experience by leveraging advanced Artificial Intelligence (AI) and a mobile-first approach. It simplifies the product discovery process and enhances user experience through the following key features:

* **AI-Driven Search**
  + Users can search for fashion items using images or natural language. QuickFind’s AI engine scans a comprehensive database of 139 million products across 369,000 stores to deliver precise recommendations, including exact matches and similar items.
* **Mobile-First Experience**
  + QuickFind is optimized for mobile devices, allowing users to easily search and shop using their smartphones. The app supports various input methods, including photo uploads, screenshots, and descriptions in natural language.
* **Comprehensive Product Database**
  + Access to a vast array of fashion products from multiple stores enables users to compare prices, check for promotions, and find items that best fit their needs.
* **Streamlined Checkout Process**
  + After selecting preferred items, users are redirected to the store’s checkout page with all their details pre-filled, making the purchase process faster and more convenient.
* **AI Copilot Facility**
  + Users can text-based natural language product search, explore product details, shipping costs and ask questions about the item or the store, enhancing the overall shopping experience.

# APP Design Guidelines

We will be using a Multi-Purpose PWA Mobile App Template for the APP development, as the template offers robust support for various QuickFind App features with pre-built UI elements and layouts, speeding up development while maintaining a modern, responsive design.

You can navigate the template and view the pages, features etc here:

[Odis: Multi Purpose PWA Mobile App Template (jaybabani.com)](https://jaybabani.com/odis-html/?storefront=envato-elements)

* Pre-built Pages like Registration, Login, Profile, and Password Reset cover user authentication and management.
* Ecommerce templates like Products, Shopping Cart, and Checkout directly support product browsing, shopping, and purchasing.
* UI Components like Forms, Buttons, Grids, Animations, and Pre-loaders are useful for implementing interactions on pages such as Analyse Image, Detect & Select, and Search.
* Pre-built Chat & Messaging modules can be adapted for the Product Page Chatbot.
* Error Pages are leveraged for error handling scenarios like Out of Stock.

## Brand Identity and Vision

**App Vision:**

QuickFind revolutionizes the online shopping experience by providing users with personalised, AI-driven product discovery and recommendations. The app focuses on simplicity, efficiency, and accessibility, offering a seamless shopping journey from search to checkout.

**Core Values**:

* **Innovation**: Leveraging AI to provide personalised and consumer-needs focused product recommendations.
* **User-Centric**: Focused on providing an easy-to-use and intuitive shopping experience.
* **Transparency**: Offering unbiased product recommendations based on user preferences rather than ad revenue.
* **Empowerment**: Supporting small and medium-sized stores by connecting users to their products.

## App Design Principles

**Simplicity and Efficiency**:

* The user interface (UI) must be intuitive, with minimal friction at every stage of interaction.
* Key features like product search, recommendations, shopping cart, and checkout should be easily accessible with a clear flow from one step to the next.

**Mobile-First Approach**:

* The app must be optimised for mobile devices, with responsive design principles that ensure usability across screen sizes.
* Touch interactions, gestures (e.g., swipe), and mobile-friendly navigation should be the foundation of the user experience (UX).

**AI Integration**:

* AI-driven features, such as personalised product recommendations and chatbot assistance, should feel seamless and natural.

## Typography

* **Font Family**: Oswald and Roboto(in line with react-based PWA mobile app template provided)
* **Weights**: In line with react-based PWA mobile app template provided.
* **Header Text**: In line with react-based PWA mobile app template provided.
* **Body Text**: In line with react-based PWA mobile app template provided.
* **Button Text**: In line with react-based PWA mobile app template provided.

## Colour Palette

In line with react-based PWA mobile app template provided.

* #26A69A (Dark Teal Green)
* #DFF2F0 (Light Teal Green)
* #495E67 (Grey-Level 1)
* #455A64 (Grey-Level 2)
* #E7E7E7 (Grey-Level 3)
* #F5F5F5 (Grey-Level 4)
* #FFFFFF (White)
* #000000 (Black)

## UI Elements and Components

In line with react-based PWA mobile app template provided.

**Interface**:

* Buttons
* Shadow
* Dropdowns
* Progress Bars
* Badges
* Text Input
* Iconic Text Inputs
* Text Input Validation
* Multiple Select
* Select Options
* Avatar Chips
* Checkboxes
* Chips
* Pagination
* Collection
* Icons
* Switches

**Features**:

* Carousel
* Tabs
* Footer
* Pre-loader
* Swipable
* Footer Menu
* Message Popups

## Imagery and Product Displays

**Product Images**:

* The app must support multiple images per product with carousel navigation.

**Banner Images**:

* **Marketing Banner**: Ensure banners are adaptable to screen sizes and support responsive layouts. Use bold, legible text and clear calls to action within banners, promoting the QuickFind our native or PWA app.

## Functional Features and Interaction Design

**Product Search & Discovery**:

* **Product Listings**: Ensure a clear layout for product listings, showing images, prices, and key information (e.g., free shipping).

**Add to Cart and Buy Now:**

* **Add to Cart Button**: Should be clearly visible below the product details and instantly show confirmation after successful action.
* **Buy Now Button**: Positioned next to "Add to Cart" but triggers an immediate checkout process.

**Chatbot Interaction**:

* The chatbot input field allows users to input custom queries or select predefined prompts. The responses should be displayed cleanly.

## Animations and Micro interactions

**Loading Indicators**:

* Use subtle animations (e.g., pre-loader or progress bar) to indicate loading states when retrieving product details or during checkout processes.

**Confirmation Animations**:

* Upon successful actions like "Add to Cart", provide visual feedback such as a cart icon update.

**Transitions**:

* Ensure smooth transitions between pages and states, particularly when moving from the product page to the cart or checkout.

## Accessibility Guidelines

**Colour Contrast**:

* Maintain high contrast between text and background colours, ensuring legibility for all users. Primary and secondary actions should stand out clearly.

**Font Size and Readability**:

* Use scalable font sizes for readability on different screen sizes. Ensure line height and spacing are optimized for legibility.

**Alt Text**:

* Provide alt text for all product images to enhance accessibility for users with screen readers.

**Touch Targets**:

* Ensure all interactive elements (e.g., buttons, icons) have a minimum touch target size for easy use on mobile devices.

## Tone of Voice

**Messaging**:

* **Friendly**: Provide helpful, concise instructions and messages.
* **Clear and Informative**: Error messages, product descriptions, and confirmation messages should be clear, direct, and actionable.

**Error Handling**:

* Use clear and polite language in error messages, e.g., “Oops! Something went wrong. Please try again.”

## Error Handling and Notifications

**Form Validation**:

* Provide real-time feedback on form inputs, highlighting errors in a contrasting colour and offering helpful guidance.

**Notifications**:

* Use non-intrusive notifications to inform users of successful actions (e.g., adding items to the cart).

### Template to Requirements Mapping

**As a guide only**, the following is a table showing how the Multi-Purpose PWA Mobile App Template features and pages could potentially be mapped to the QuickFind App pages and their respective requirements:

| **QuickFind App Page** | **Template Features/Pages** | **Template Elements to Use** |
| --- | --- | --- |
| Register Page | Registration, Welcome Page | - Registration form (pre-built)  - Input fields for Name, Email, Password  - Social Login integration  - Validation and error handling  - Loading spinner |
| Sign-in Page | Login, Welcome Page | - Login form (pre-built)  - Input fields for Email/Password  - Remember Me checkbox  - Guest access option  - Social Login options  - Forgot Password link |
| Password Reset Request Page | Forgot Password | - Password reset form (pre-built)  - Email input field  - Password reset link functionality  - Validation for expired links  - Error messages for invalid inputs |
| Profile Page | Profile, Edit Profile, User Profile Card | - Profile management module (pre-built)  - Editable fields for Name, Email, Address  - Password change functionality  - Logout button |
| Search Page | Search, Dashboard, Ecommerce Home | - Search bar  - Image search icons (can be customized with image buttons)  - Filters (Price, Free Shipping)  - Materialize framework components for responsive design |
| Analyse Image Page | Pre-loader, File Upload | - File/image upload functionality  - Loading spinner/Pre-loader animations  - Text input for natural language description |
| Detect & Select Page | Cards, Grid Layout | - Image display (part of the Ecommerce home or gallery)  - Bounding boxes (custom overlays)  - Interactive selection with visual feedback using Materialize animations |
| Product Search Page | Products (Grid View), Products (Single Column) | - Progress bar animation  - Product listing with product images (grid or list view) |
| Recommendations Page | Ecommerce Home Page, QuickFind Marketing Banner | - Marketing banner (pre-built banner display)  - Product recommendations in grid layout  - Product details (image, price, discount, free shipping) |
| Product Page (Main) | Product Details, Ecommerce Product Page | - Product details display with image carousel  - Price, comparison price, discount info  - Size, colour selection dropdowns  - Store details  - Tabs for features, returns |
| Cart Page | Shopping Cart, Invoice | - Cart layout with sections per store  - Product details (price, colour, size, quantity)  - Shipping options  - "Proceed to Checkout" button |

## How it Works

The following is a high-level step-by-step explanation of the QuickFind solution from a user’s perspective:

* **Step 0 - Identify Management**
  + Enables user registration, sign-in, and guest access, with secure password reset and profile management options. Users can update personal details, manage addresses, and change passwords. It also includes security features like email verification, session management, and logout, ensuring a seamless and protected user experience across the app.
  + Related Pages
    - Registration Page
    - Sign-in / Guest Page
    - Reset Password Page
    - Profile Page
* **Step 1. Product Capture**
  + The user begins the process using one of several methods: uploading a photo or image file, sharing a screenshot they have taken, sharing images from their gallery, direct messaging an image, or describing the item in natural language. The image or natural language input is then analysed to detect the items of clothing present which the user can select from to begin the search process.
  + Related Pages
    - Analyse Page
    - Detect/Select Page
* **Step 2. Product Search & Discovery**
  + QuickFind’s advanced AI engine processes the user's item selection, scans a vast database of 139 million products across 369,000 stores, and then generates a curated list of direct matches and similar items.
  + Related Pages
    - Search Page
* **Step 3 - Product Review & Filtering**
  + The user reviews the recommended items, with the ability to filter based on price, free shipping, discount levels, etc.
  + Related Pages
    - Recommendations Page
* **Step 4 - Product Selection & Exploration**
  + After comparing and refining, the user selects their preferred product. QuickFind allows further exploration of the product’s details or the ability to ask questions about the item or the store in natural language.
  + Related Pages
    - Product Page
* **Step 5 - Product Purchase**
  + Once the user decides to make a purchase, they can either add the product to their shopping cart to check out later or be automatically redirected to the store’s checkout page. QuickFind pre-fills their name, email address, street address, and phone number details, ensuring a quick and seamless purchase process
  + Related Pages
    - Cart Page

## QuickFind Site Map

1. **Register Page**
   * Registration Details (Name, Email, Password)
   * Sign-in (link)
   * Social Login
2. **Sign-in Page**
   * Email/Password
   * Remember me
   * Sign-in as Guest
   * Forgot Password (link)
   * Register (link)
   * Sign-in (link)
   * Social Login
3. **Password Reset Request Page**
   * Email
   * Reset Password Link
   * Sign-in (link)
   * Register (link)
4. **Profile Page**
   * View Profile
   * Manage Profile (Update Name, Email, Address, DOB, Password, etc.)
   * Logout Button
5. **Search Page**
   * Image Search Icons
   * Chatbot Search Bar
   * Free Shipping Filter
6. **Analyse Image Page**
   * Product Image
   * Spinner Animation Icon
7. **Detect & Select Page**
   * Product Image
   * Detected Item Overlays
   * Progress Bar Animation
8. **Recommendations Page**
   * QuickFind Marketing Banner
   * Price Filter
   * Recommended Products
     + Product Name
     + Product Price
     + Product Images
     + Comparison Price
     + Percentage Discount (calculated by Product Price vs Comparison Price)
     + Free Shipping Information
     + Direct Match / Similar Item Tag
   * Footer with Icons (Search, Recommendations & Cart)
9. **Product Page (Main)**
   * Product Name
   * Product Price
   * Product Images
   * Product Description
   * Comparison Price
   * Percentage Discount (calculated by Product Price vs Comparison Price)
   * Free Shipping Information
   * Product Sizes
   * Product Colours
   * Tabs (Features, Shipping, Returns, Store)
   * Store Logo and Details (Refund, Shipping, etc.)
   * Store Rating ***(Scope inclusion to be confirmed)***
   * Chatbot Enquiry Bar
   * Buttons (Add to Cart, Buy Now)
   * Footer with Icons (Search, Recommendations & Cart)
10. **Product Page (Chatbot Section/Interaction)**
    * Prompt Suggestions
    * Chat Interactions
11. **Product Page (Error Popup)**
    * ‘Out of Stock’ Information Text
12. **Cart Page**
    * Sections (Per Store)
    * Merchant Name
    * List of Products Added to Cart (Per Store)
    * Total Cost (Per Store)
    * Product Price
    * Product Colour
    * Product Size
    * Product Quantity
    * Proceed to Checkout Button/Link (Per Store)
    * Footer with Icons (Search, Recommendations & Cart)

# Application Functional Requirements

## FUNCTIONAL - Step 0 – Identity Management

### Register Page

#### Functional User Story

As a new user, I want to create an account as quickly and easily as possible.

#### Functional Requirements

* + **Registration (Sign-Up Options)**:
    - Users should be able to sign up using an email address and a password.
    - ***Alternatively, provide the option for social sign-up (Scope inclusion to be confirmed, but would include the following integration, depending upon what is available in the template)***
      * Google Login
      * Facebook Login
      * TikToc Login
      * Apple Login
      * Instagram Logon
  + **Email address and password option**
    - **Form Fields**:
      * First Name (Mandatory)
      * Last Name (Mandatory)
      * Email Address (Mandatory, must be unique)
      * Password (Mandatory, must meet security criteria: min. 8 characters, mix of letters, numbers, and symbols, and should provide real-time feedback advising if minimum security criteria has been met)
      * Confirm Password (must match the password)
      * Date of Birth (Optional)
      * Address (Optional)
        + Delivery (Shipping) addresses.
    - **Verification**:
      * Once the user fills in the required details and submits the form, they will receive a verification email.
      * The verification email must contain a confirmation link that expires after 24 hours.
      * The user cannot log in without confirming their email address.
    - **Validation & Error Handling**:
      * The form should validate email formats, password strength, and matching passwords in real-time.
      * Error messages should be user-friendly and highlight any missing or invalid inputs.
    - **User Interface Considerations**:
      * The register button should be disabled until all mandatory fields are filled.
      * Ensure a loading spinner or progress indicator is shown during form submission.

### Sign-in Page (or Guest Access)

#### Functional User Story

* I want to sign in with my account details or continue as a guest, so I can search, browse and make purchases without needing to sign up immediately.

#### Functional Requirements

* + **Sign-In Options**:
    - Users must have the option to sign in using their registered email and password.
    - Allow users to sign in using social media accounts - if integrated.
    - Provide a "Forgot Password" link for easy recovery.
    - Include a "Remember Me" checkbox, so users can stay signed in for future sessions.
  + **Guest Access**:
    - Users should be allowed to continue using the app as a guest without signing up.
    - Guest access should allow browsing and adding items to the cart
    - Once a user decides to sign up or sign in, any cart or session data should be seamlessly transferred to their newly created account.
  + **Validation & Error Handling**:
    - On the sign-in page, provide clear error messages for incorrect email/password combinations.
    - Include validation for email format when submitting the form.
  + **Security Considerations**:
    - After a certain number of failed login attempts (e.g., 10), the account should be temporarily locked, requiring a password reset to unlock.

### Password Reset Request Page

#### Functional User Story

* As a user, I want to be able to reset my password if I forget it, so I can regain access to my account.

#### Functional Requirements

* + **Password Reset Request**:
    - The user should have a "Forgot Password?" link on the sign-in page.
    - Upon clicking the link, they should be redirected to a **password reset request page** where they can enter their registered email.
  + **Email Delivery**:
    - The system will send an email containing a unique password reset link.
    - This link should have a time limit (e.g., expire in 24 hours) for security purposes.
  + **Reset Page**:
    - The user clicks on the link in the email, which redirects them to a password reset page.
    - The reset page should include fields for entering the new password and confirming it.
      * The new password must adhere to security policies: min. 8 characters, mix of letters, numbers, and symbols, and should provide real-time feedback advising if minimum security criteria has been met
  + **Validation & Error Handling**:
    - Show clear error messages for expired links.
    - Password confirmation should ensure that the "New Password" and "Confirm Password" fields match.
  + **Post-Reset**:
    - After successfully resetting their password, the user should be notified that the password has been updated and should be directed to the sign-in page.
    - Send a confirmation email about the password change to the user’s registered email.

### Profile Management Page

#### Functional User Story

* As a user, I want to manage my profile settings so that I can keep my information up to date.

#### Functional Requirements

* + **Navigation**
    - Navigation to the profile page is via a profile icon on the header page of all key pages:
      * *See Profile & Footer Icon Navigation Section for details*
  + **Profile Management**:
    - Users should be able to update their personal details:
      * Name (First and Last)
    - Changes should be saved and applied immediately via a Save button
  + **Address Management**:
    - Users should be able to add, edit, or delete multiple delivery (shipping) addresses.
    - Allow users to set a default delivery address for future purchases.
    - Changes should be saved and applied immediately via a Save button
  + **Password Change**:
    - Users should be able to change their password by entering the current password and setting a new one.
    - Password should only be updated once the user confirms the new password by entering it twice.
    - Show error messages if the current password is incorrect, is the same as the current password or the new password does not meet security standards.
    - Changes should be saved and applied immediately via a Save button

### App Logout

#### Functional User Story

* As a user, I want to log out of my account securely to protect my profile and information from unauthorised access.

#### Functional Requirements

* + **Logout Functionality**:
    - Users must be able to log out from the app from their profile screen
    - Logging out should clear the user's session, and they should be redirected to the sign-in page.
  + **Security Considerations**:
    - Ensure that all session data (like cart items, profile details, etc.) is cleared when logging out to prevent unauthorised access.

## FUNCTIONAL - Step 1 - Product Capture & Detection

### Search Page

**Image Search**

#### Functional User Story

* As a user, I want to upload or take a photo of a fashion item, **so that** I can receive personalized product recommendations based on the image I provide.

#### Functional Requirements

* Users must be able to upload an image from their device or take a photo using the device’s camera.
  + E.g. A camera icon for taking a photo and a gallery icon for uploading an image from the device

**Text Search**

#### Functional User Story

* As a user, I want to describe a fashion item in a text search, **so that** I can find matching products based on my description.

#### Functional Requirements

* Users can type a natural language description of the fashion item they are looking for (e.g., “blue denim jacket”).
* Users can apply one or more filters, such as:
  + **Free Shipping**: Users can toggle an option to only show products eligible for free shipping.

### Analyse Page

#### Functional User Story

* I want the app to:
  + Analyse the image I’ve provided and detect the clothing items and present them to me for selection

#### Functional Requirements

* + **Image/Description Input Handling:**
    - The system must accept image input from:
      * Uploaded photo or image file.
      * Shared screenshot.
      * Image from the device's gallery.
      * Direct messaging an image.
  + **Image Processing and Analysis:**
    - Upon receiving input, the system should initiate an analysis of the image or text to detect clothing items such as tops, jeans, and shoes.
    - A background process must analyse the image using AI-based object detection techniques to identify clothing items.
  + **Loading Message and Indicator**:
    - A loading spinner must be shown until the analysis is complete.
    - The page must show text "Scanning image" along with the loading indicator.
    - While the system is analysing the image, the following message must be displayed prominently on the page:

“One moment while we analyse the image to identify the clothing items.”

* + **Specific Error Handling**:
    - If the image cannot be analysed (due to low quality or missing data), an error message should be displayed:  
      “We couldn’t analyse the image. Please try again with a different image.”
  + **Transition to Detect & Select Page**:
    - Once the analysis is complete, the system should automatically navigate to the "Detect & Select" page.
    - The analysed image must be passed to the next page with clothing items identified and boxed.

### Detect & Select Page

#### Functional User Story

* As a user, I want to view the clothing items detected from my image and select the item I am interested in, so that I can find similar products or recommendations based on that specific item

#### Functional Requirements

* + **Image and Detected Items Display**
    - **Display Analysed Image**:
      * The system must display the analysed image in its entirety, ensuring the image is clearly visible and scaled properly for the user's screen size.
    - **Highlight Detected Clothing Items**:
      * The system must draw bounding boxes or overlays around each detected item of clothing (e.g., top, jeans, shoes).
      * Each bounding box must have a corresponding clothing label (e.g., “Top,” “Shoes,” “Jeans”).
      * The bounding boxes and labels should be visually distinct but unobtrusive.
  + **Selection of Clothing Items**
    - **Instructions for User**:
      * The system must display instructions such as *“Select the item of clothing you wish to search for”* in a prominent location on the page.
    - **Item Selection**:
      * The user must be able to tap on any of the highlighted items (bounding boxes) to select it.
      * Upon selecting an item, the bounding box and label for that item must change visually to confirm selection (e.g., a change in colour, or an animation).
      * Only one item can be selected at a time.
  + **Feedback and Loading Indicators**
    - **Display Feedback on Selection**:
      * Once an item is selected, the system must immediately show feedback such as a highlighted box, a confirmation message (*"Top selected"*), or a visual cue.
    - **Loading Indicator and Message**:
      * After the item is selected, the system must display a progress indicator with a message like *“Preparing search results…”*.
      * The progress indicator must remain visible until the search results are ready, and the system navigates to the next page(Recommendations Page)
  + **Navigation to Recommendations Page**
    - **Trigger Search Process**:
      * Once the user selects a clothing item, the system must automatically trigger a search based on the selected item.
    - **Pass Data to Recommendations Page**:
      * The system must pass relevant data about the selected item (e.g., item type, detected area in the image) to the "Recommendations Page."
  + **Seamless Transition**:
    - The system must transition smoothly to the "Recommendations Page" once the search is initiated. A subtle animation (e.g., fade or slide) can be used to enhance the user experience.
  + **Mobile Optimization**
    - **Mobile Responsiveness**:
      * The layout and design must be optimized for mobile devices, ensuring that bounding boxes, text, and buttons are appropriately sized and easy to interact with on smaller screens.
    - **Touch-Friendly Interface**:
      * The system must ensure that all interactive elements (e.g., bounding boxes, buttons) are touch-friendly with adequate spacing to prevent mis-taps on mobile devices.

## FUNCTIONAL - Step 2 - Product Search & Discovery

### Recommendations Page

#### Functional User Story

* As the QuickFind business, I want to see a static marketing banner at the top which advises the use that they can download our native or PWA app so that they can share images with us from anywhere (i.e. screenshot, from gallery, etc)

#### Functional Requirements & Design Considerations

* **Recommendations Page Header** 
  + **Marketing Banner Display**
    - **Functionality:**
      * The header should display a **marketing banner** retrieved from the **Shopping Database**.
    - **Data Source:**
      * **Shopping Database** should supply the banner image URL, associated text
    - **Design & Behavior:**
      * The banner must adapt to the screen size and support responsive layouts.
      * The banner can support **multiple images** with the option for **slideshow-style scrolling**.
      * If clicked, the user should be redirected to the appropriate page to download the Native or PDA APP
    - **Requirements:**
      * **Inputs:**
        + Marketing Banner URL
        + Marketing Text (optional)
        + Redirection URL (click behaviour)
      * **Outputs:**
        + Banner displays at the top of the recommendations page with clickable behavior.

#### Functional User Story

* As a user, I want to see a list of recommended products based on my search, with details like images, descriptions, prices, and free shipping status, so I can easily browse and click on items to view more information on the product page.

#### Functional Requirements & Design Considerations

* **Recommendations Page – Body**
  + **Product Listing Display**
* **Functionality:**
  + Display a list of **recommended products** based on the user’s input (image-based or text-based search).
  + Provide the ability to sort by price
  + For each product, show:
    - **Product Title**
    - **Product Price**
    - **Product Comparison Price**
    - **Product Percentage Discount**
    - **Product Image**
    - **Product Price**
    - **Free Shipping Availability**
    - **Direct Match / Similar Item**
    - **Link to the product page.**
* **Data Source:**
  + **Products Database**
    - **Product Title**: Product title text.
    - **Product Image**: URL for the product's image
    - **Product Comparison Price:** Original price of the product
    - **Product Price**: Price of the product
    - **Free Shipping Availability**: Text indicating whether the product is eligible for free shipping.
    - **Product Page URL**: Link to redirect the user to the product's page on the e-commerce site.
    - **Direct Match / Similar Item:** Text indicating whether the product is a Direct Match or Similar Item.
  + **Calculated**
    - **Product Percentage Discount:** Value calculated based on Product Price vs Product Comparison Price
* **Design & Behaviour:**
  + Products are displayed in a **grid view**
  + Each product block must include the items listed above
  + Clicking the product’s image or description will redirect the user to the **Product Page**.

**Footer Icons**

#### Functional User Story

* As a user, I want quick access to key features via the footer icons
  + ***See Profile & Footer Icon Navigation Section for details***

## APPLICATION - Step 4 - Product Selection & Exploration

### Products Page

#### Functional User Story

* As a user, I want to see detailed information about the product I’ve selected, including available sizes, colours, images, pricing, and key features, so I can make an informed decision before purchasing.

#### Functional Requirements & Design Considerations

* **Products Page – Body**
  + **Product Selected Details Display**

**SIZES**

* **Requirements**: Display available sizes for the selected product.
  + **Source**: Sizes are retrieved from the **Products Database**.
  + **Input**: Product ID used to fetch available sizes (e.g., S, M, L, XL).
  + **Display**: Sizes are shown as selectable buttons or dropdowns for user selection.

**COLOURS**

* **Requirements**: Display available colour options for the selected product.
  + **Source**: Colour options are retrieved from the **Products Database**.
  + **Input**: Product ID used to fetch available colours (e.g., Red, Blue, Green).
  + **Display**: Colours are shown as selectable swatches or buttons.

**PRODUCT IMAGES**

* **Functionality**: Display available images of the selected product.
  + **Source**: Product image(s) are retrieved from the **Products Database**.
  + **Input**: Product ID used to fetch image URLs.
  + **Display**: Where >1 image, then carousel allowing user to scroll through product images.

**PRODUCT TITLE**

* **Functionality**: Display the product title.
  + **Source**: Product title is retrieved from the **Products Database**.
  + **Input**: Product ID used to fetch the title.
  + **Display**: Title should appear prominently.

**PRODUCT PRICE**

* **Functionality**: Display the price of the selected product.
  + **Source**: Product price is retrieved from the **Products Database**.
  + **Input**: Product ID used to fetch the price.
  + **Display**: Price is shown in the appropriate currency (e.g., $99.99).

**PRODUCT COMPARISON PRICE**

* **Functionality:** Display the comparison price of the product with a strikethrough (e.g. ~~$129.99~~).
  + **Source:** Comparison price is retrieved from the **Products Database.**
  + **Input:** Product ID used to fetch comparison price.
  + **Display:** comparison price is displayed next to Product Price.

**PRODUCT DISCOUNT**

* **Functionality**: If applicable, display the discount or promotional offer for the product.
  + **Source**: Discount data is retrieved from the **Products Database**.
  + **Input**: Product ID used to fetch any applicable discounts.
  + **Display**: Show discount (e.g., "20% off" or "$10 off") under the product price.

**PRODUCT DESCRIPTION**

* **Functionality**: Provide a detailed description of the product (e.g., materials, fit, etc.).
  + **Source**: Product description is retrieved from the **Products Database**.
  + **Input**: Product ID used to fetch the description.
  + **Display**: The description is shown in the tabbed section below the product images and pricing sections.

**PRODUCT FEATURES**

* **Functionality**: List the main features of the product (e.g., waterproof, breathable).
  + **Source**: Product features are retrieved from the **Products Database**.
  + **Input**: Product ID used to fetch feature information.
  + **Display**: Shown in a bulleted list shown in the tabbed section below the product images and pricing sections.

**SHIPPING INFO**

* **Functionality:** List details of the shipping policy of the store
  + **Source:** Shipping details are retrieved from the **Products Database.**
  + **Input:** Product ID used to fetch shipping details.
  + **Display:** Shown in a bulleted list shown in the tabbed section below the product images and pricing sections.

**RETURNS INFO**

* **Functionality:** List details of the returns policy of the store
  + **Source:** Returns details are retrieved from the **Products Database.**
  + **Input:** Product ID used to fetch returns details.
  + **Display:** Shown in a bulleted list shown in the tabbed section below the product images and pricing sections.

**STORE INFO**

* **Functionality:** Provide a detailed description of the store
  + **Source:** Store details are retrieved from the **Products Database.**
  + **Input:** Product ID used to fetch store details.
  + **Display:** The description is shown in the tabbed section below the product images and pricing sections.

**FREE SHIPPING AVAILABILITY**

* **Functionality:** Advise if the product has free shipping or not (Yes/No)
  + **Source:** Free Shipping details are retrieved from the **Products Database.**
  + **Input:** Product ID used to fetch free shipping details.
  + **Display:** The description ”FREE SHIPPING” is shown below the product image.

**STORE RATING**

* **Functionality:** Provide a rating for the store
  + **Source:** Store ratings are retrieved from the **Products Database.**
  + **Input:** Product ID used to fetch the store rating.
  + **Display:** The rating is shown in the tabbed section below the product images and pricing sections.
  + **Add to shopping cart and buy now options**
* Covered under the ‘Order Management’ section
  + **Specific Error Handling**
* If any product details fail to load (e.g., due to API failure or missing data), display fallback error messages (e.g., “Product details unavailable”).
* Implement caching to ensure fast loading times for repeat visits.
* Handle cases where the store logo or product images are missing by displaying placeholders.

**Footer Icons**

#### Functional User Story

* As a user, I want quick access to key features via the footer icons
  + ***See Profile & Footer Icon Navigation Section for details***

1. **Product Page (Chatbot Section/Interaction)**

#### Functional User Story

* As a user, I want to ask questions or select common prompts about the product or store directly from the product page via a chatbot, so I can quickly get the information I need, such as product specifics or store policies, without navigating away from the page.

#### Functional Requirements & Design Considerations

* + **Product / Store Enquiry (Chatbot)**
* **Chatbot enquiry**
* **Functionality**: Allow the user to enter a custom query or select a pre-defined prompt about the selected product or store.
  + **Source**: QuickFind App (User Input Field).
  + **Input**:
    - * Text-based user input (custom question).
      * Pre-defined prompts (selectable options).
      * Product context (Product ID, Store ID).
  + **Display**:
    - * Chatbot input field on the product page.
      * List of pre-defined prompts for common questions (e.g., “What are the product features?”, “What is the return policy?”).

## APPLICATION - Step 5 - Product Purchase

### Cart Page

**Add to Cart Button**

#### Functional User Story

As a user, I want to add a product to my shopping cart so I can continue shopping and later review my items before making a purchase.

#### Functional Requirements

* **Add Product to Cart**: When the user clicks the "Add to Cart" button, the selected product (with chosen size, colour, and quantity) is added to the shopping cart and a permalink is generated
  + As part of this process, we need to:
  1. Check if the store is out of stock and if yes, display error message
  2. Ensure the button should be disabled if the user hasn't selected the required options (e.g., size, colour).
* **Cart Updates**: The shopping cart should update the item count in the cart icon displayed in the footer, reflecting the new total of items in the cart.
* **Persistent Cart**: The cart should persist across user sessions, allowing the user to come back later and continue shopping without losing their selections.
* **Feedback**: Once clicked, a loading spinner or animation should appear to show the process of adding the product to the cart. After confirmation, a toast notification should appear confirming the successful addition.

**Buy Now Button**

#### Functional User Story

As a user, I want to quickly purchase a product by adding it to my cart and immediately proceeding to checkout to save time.

#### Functional Requirements

* **Buy Now**: Like the "Add to Cart" function, clicking "Buy Now" adds the selected product (with chosen size, colour, and quantity) to the shopping cart and a permalink generated
  + As part of this process, we need to:

1. Check if the store is out of stock and if yes, display error message
2. Ensure the button should be disabled if the user hasn't selected the required options (e.g., size, colour).

* **Redirect to Cart:** After adding the product, the user is immediately redirected to the shopping cart page where they can review the items.
* **Proceed to Checkout**: The user can review the cart contents, update quantities if needed, and click the "Proceed to Checkout" button to continue the purchase process.

**Shopping Cart to Checkout (via Shopify)**

#### Functional User Story

As a user, I want to quickly proceed from the shopping cart to the store's checkout page with my personal details pre-filled, making the purchase process fast and easy.

#### Functional Requirements

* + **Cart Overview**: Users can review their selected items in the shopping cart, with options to adjust quantity, remove items, or proceed to checkout.
  + **Checkout Redirection**: Clicking "Proceed to Checkout" redirects the user to the Shopify store's checkout page.
  + **Pre-Filled Details**: User details (such as name, email address, shipping address and phone number) are pre-filled at the Shopify checkout, pulled from the user’s profile in QuickFind.
  + **Persistent Data**: Cart contents and user details are saved so the user can pick up where they left off even if they return to the app later.
  + **Cart Page Layout**: The cart page should:
    - List all products with clear details on price, size, colour, price and quantity
    - Show the total cost value for all products
    - Provide options to edit items or remove them from the cart.
  + **Proceed to Checkout Button**: Place a "Proceed to Checkout" button prominently at the bottom of the cart page.
  + **Continue Shopping Button**: Place a "Continue Shopping" button prominently at the bottom of the cart page.
  + **Feedback**: After clicking "Proceed to Checkout", show a animated indicator while redirecting to Shopify.

**Footer Icons**

#### Functional User Story

* As a user, I want quick access to key features via the footer icons
  + ***See Profile & Footer Icon Navigation Section for details***

# Profile & Footer Icon Navigation

## Profile Icons

* **Functionality:**
  + Navigates the user to their **Profile** section, where they can manage profile settings.
* **Design:**
  + Profile icon should be easily distinguishable and intuitive for users to find and manage their account. (Right Aligned)

Profile icon accessible on the following pages:

* Search Page
* Recommendations Page
* Product Page
* Cart Page

## Footer Icons

The footer will contain icons to the following pages:

* Search
* Recommendations
* Cart
* **Search Icon:**
  + **Functionality:**
    - Acts as a navigation button that brings the user back to the **Search** page.
    - This icon allows the user to initiate a new search by taking a picture, uploading an image, or using the natural language description etc
  + **Design**:
    - Displayed as a prominent, recognizable icon in the footer for quick access (Left aligned)
    - When on the search page, the icon should use a different coloured version of the icon
* **Recommendations Icon**:
  + **Functionality**:
    - Navigates the user to their **Recommendations** page, where they can view the recommendations from the last search results.
  + **Design**:
    - Profile icon should be easily distinguishable and intuitive for users to find their previous recommendations (Centre aligned)
    - When on the recommendations page, the icon should use a different coloured version of the icon
* **Cart Icon:**
  + **Functionality**:
    - Links to the user’s **shopping cart** where items are saved for checkout.
    - Users can access their cart to review items, modify quantities, and proceed to the purchase.
  + **Design**:
    - Display a cart icon with an indicator showing the number of items currently in the cart (if applicable). (Right aligned)
    - When on the recommendations page, the icon should use a different coloured version of the icon

# Email Templates

Email templates will be required for the following user cases:

* **Registration**
  + Once the user fills in the required details and submits the form, they will receive a verification email.
  + The verification email must contain a confirmation link that expires after 24 hours.
* **Password reset request**
  + The system will send an email containing a unique password reset link.
* **Password reset request**
  + Send a confirmation email about the password change to the user’s registered email.
* **Cart**
  + Periodic email to the user to advise that they have products in their Cart (pro-active marketing)

# Screen Design Samples

* **Link to Figma**

<https://www.figma.com/proto/piseTIAR7FlN0IGQcX4tlh/QuickFind?page-id=64%3A6075&node-id=89-510&node-type=frame&viewport=479%2C-877%2C0.15&t=92i4PHTCAuu1IIEx-1&scaling=scale-down&content-scaling=fixed&starting-point-node-id=89%3A510&share=1&show-proto-sidebar=1>