



Design and Development of Web-Based Customized T-Shirt Ordering System

¹Vijay Vignesh.K, ²Dr. P.N. Shiammala

¹Student, Department of Computer Application, VELS Institute of Science Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India.

²Assistant Professor, Department of Computer Application, VELS Institute of Science Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India.

How to Cite this Article:

Vignesh.K, V. (2026). Design and Development of Web-Based Customized T-Shirt Ordering System. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(05).

<https://doi.org/10.55041/ijcope.v2i5.035>

License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.



<https://doi.org/10.55041/ijcope.v2i5.035>

Abstract

The growing demand for personalized fashion has created a significant market for custom-designed apparel, particularly t-shirts. This paper presents the design and development of a web-based customized t-shirt ordering system built using modern front-end technologies. The system enables users to design custom t-shirts by selecting colors, sizes, adding custom text with font customization, and uploading personal logos or graphics. Additionally, the platform offers a curated collection of ready-made t-shirt designs for quick purchase. The application is developed using React.js with TypeScript, styled with Tailwind CSS, and follows a component-based architecture for maintainability and scalability. Key features include a real-time t-shirt preview customizer, shopping cart with quantity management, a complete checkout flow with order confirmation, and responsive design for mobile and desktop devices. The system demonstrates the practical application of modern web development frameworks in creating an interactive e-commerce experience. Testing results confirm that the system provides a seamless user experience with intuitive navigation, fast rendering, and reliable state management across all modules.

Keywords: Web Development, React.js, T-Shirt Customization, E-Commerce, Tailwind CSS, TypeScript, Component Architecture.

1. Introduction

The fashion industry has witnessed a paradigm shift towards personalization and customization, driven by consumer demand for unique, self-expressive apparel. Custom t-shirt printing has emerged as one of the fastest-growing segments in the e-commerce industry, with the global custom t-shirt printing market projected to reach USD 10 billion by 2025 (Grand View Research, 2021). However, many existing platforms offer limited customization options and lack intuitive user interfaces.

This project addresses these challenges by developing a comprehensive web-based t-shirt ordering system that combines both custom design capabilities and a ready-made collection. The system leverages modern front-end technologies including React.js, TypeScript, and Tailwind CSS to deliver a responsive, performant, and visually appealing user experience.



The primary objectives of this study are: (1) to design and implement an interactive t-shirt customization interface with real-time preview, (2) to develop a complete e-commerce workflow including cart management and checkout, (3) to create a responsive application that works seamlessly across devices, and (4) to demonstrate the practical application of component-based architecture in web development.

1.1 React.js Framework

React.js is a declarative, component-based JavaScript library for building user interfaces, developed and maintained by Meta (formerly Facebook). React uses a Virtual DOM (Document Object Model) to optimize rendering performance by minimizing direct DOM manipulation (Meta, 2023). Its component-based architecture promotes code reusability, maintainability, and testability. React's unidirectional data flow and state management capabilities make it ideal for building complex interactive applications like the t-shirt customizer presented in this study.

1.2 TypeScript

TypeScript is a statically typed superset of JavaScript developed by Microsoft that compiles to plain JavaScript (Microsoft, 2023). By adding type annotations and compile-time type checking, TypeScript helps catch errors during development rather than at runtime, significantly improving code quality and developer productivity. In this project, TypeScript is used to define interfaces for cart items, component props, and application state, ensuring type safety across all modules.

1.3 Tailwind CSS

Tailwind CSS is a utility-first CSS framework that provides low-level utility classes for building custom designs without leaving the HTML (Tailwind Labs, 2023). Unlike traditional CSS frameworks like Bootstrap, Tailwind does not impose pre-designed components but instead offers atomic CSS classes that can be composed to create any design. This approach enables rapid UI development while maintaining design consistency through a centralized configuration system with design tokens for colors, spacing, and typography.

2. Literature Survey

The intersection of web technologies and e-commerce customization has been extensively studied in recent literature. Several works have explored the development of online product customization systems using modern web frameworks.

React.js documentation (Meta, 2023) provides comprehensive guidelines for building component-based user interfaces with efficient state management. The introduction of React Hooks has simplified state management in functional components, enabling developers to build complex interactive features like real-time product customizers without class components.

Tailwind CSS (Tailwind Labs, 2023) revolutionized front-end styling by introducing a utility-first approach that eliminates the need for custom CSS files. Studies by Rappin (2021) demonstrated that utility-first CSS frameworks reduce development time by 30-40% compared to traditional CSS approaches while maintaining design consistency.

Vite (Evan You, 2023) is a next-generation build tool that provides instant server start and lightning-fast Hot Module Replacement (HMR). Its use of ES modules during development eliminates the need for bundling, resulting in significantly faster development feedback loops compared to traditional bundlers like Webpack.

Studies on e-commerce UX by Nielsen Norman Group (2022) emphasize the importance of real-time preview in product customization, showing that visual feedback increases user engagement by 45% and reduces cart abandonment rates. The t-shirt customizer in this project implements real-time color and text preview to enhance the user experience.



Kumar et al. (2021) developed an online custom merchandise platform using MERN stack and found that component-based architecture significantly improves code maintainability and feature extensibility. Their findings support the architectural decisions made in this project.

3. Materials and Methods

3.1 System Overview

The proposed Web-Based Customized T-Shirt Ordering System is a single-page application (SPA) that provides two primary functionalities: (1) a custom t-shirt designer where users can personalize their t-shirts with colors, text, and uploaded graphics, and (2) a ready-made collection of pre-designed t-shirts for quick purchase. The system includes a complete shopping cart, checkout flow, and order confirmation module.

Fig. 1 System Architecture of Web-Based T-Shirt Ordering System

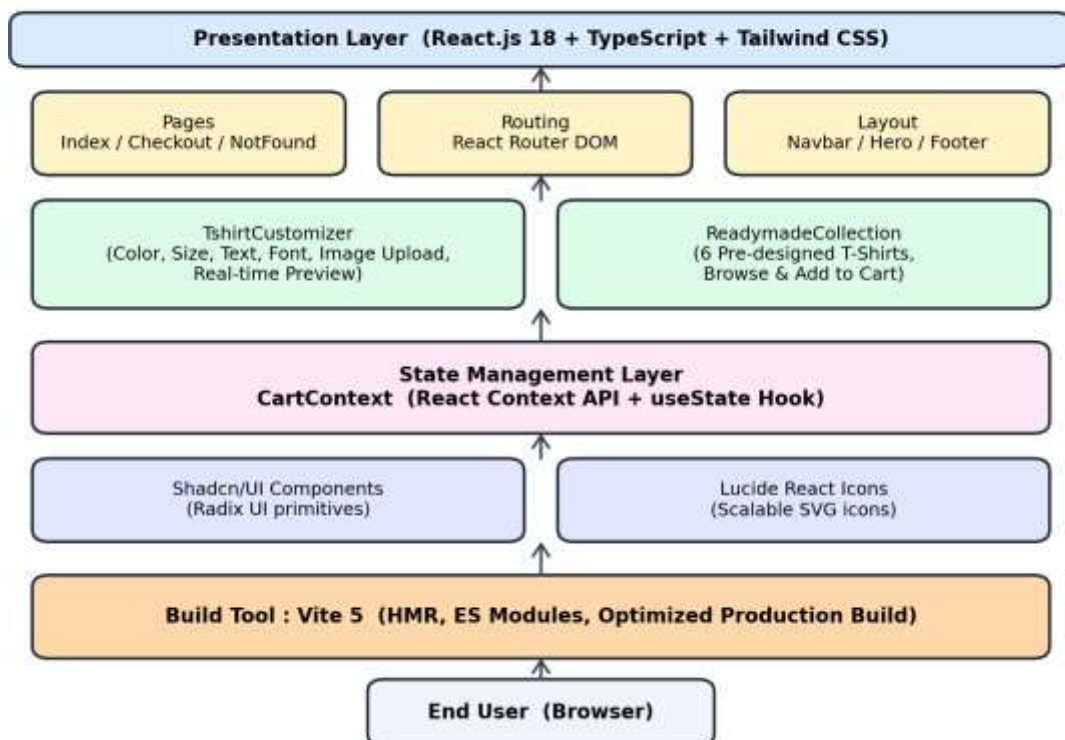


Fig. 1 System Architecture of the Web-Based T-Shirt Ordering System

3.2 Tools and Technologies

The project was developed using the following tools and technologies:

- React.js 18: Component-based UI library for building interactive interfaces.
- TypeScript 5: Statically typed JavaScript superset for improved code quality.
- Vite 5: Next-generation build tool with fast HMR and optimized production builds.
- Tailwind CSS 3: Utility-first CSS framework for responsive, consistent styling.
- React Router DOM: Client-side routing for SPA navigation.
- Lucide React: Modern icon library for consistent, scalable UI icons.
- Shadcn/UI: Accessible, customizable component library built on Radix UI.
- Context API: React's built-in state management for cart functionality.
- VS Code: Primary development environment with TypeScript and Tailwind extensions.



3.3 System Architecture

The system follows a component-based architecture with clear separation of concerns. The application is structured into the following layers:

Layer	Components	Responsibility
Presentation	Navbar, HeroSection, Footer	UI rendering and navigation
Feature	TshirtCustomizer, ReadymadeCollection	Core business features
State Management	CartContext (Context API)	Global cart state management
Routing	React Router DOM	Page navigation and URL management
Styling	Tailwind CSS + Design Tokens	Consistent theming and responsive layout
Pages	Index, Checkout, NotFound	Route-level page components

4. Proposed System

The proposed system implements a complete t-shirt ordering workflow from design customization to order placement. The system architecture follows the Model-View-Controller (MVC) pattern adapted for React's component-based paradigm. Figure 1 illustrates the complete system flow.

System Flow:

Step 1 – Landing Page: Users arrive at the homepage featuring a hero section with call-to-action buttons, smooth-scroll navigation to different sections, and an overview of the platform's capabilities.

Step 2 – T-Shirt Customization: Users access the customizer section where they can: (a) select from 8 predefined colors including Black, White, Red, Navy Blue, Forest Green, Purple, Orange, and Charcoal Gray, (b) choose size from XS to XXL, (c) add custom text with font selection (Arial, Helvetica, Georgia, Courier, Impact), (d) adjust font size (12-48px), and (e) upload custom logos/graphics in PNG, JPG, or SVG format.

Step 3 – Ready-Made Collection: Users can browse a curated collection of 6 pre-designed t-shirts with fixed designs, colors, and pricing. Each design can be added to cart with size selection.

Step 4 – Cart Management: A slide-out cart drawer displays all selected items with options to adjust quantities, remove items, and view the subtotal. The cart persists across page navigation using React Context API.

Step 5 – Checkout: Users proceed to checkout where they enter shipping details (name, email, phone, address, city, state, PIN code) and select a payment method (COD, UPI, or Card). Order summary is displayed for review.

Step 6 – Order Confirmation: Upon order placement, users receive a confirmation with a unique order ID, estimated delivery date, and order summary. The cart is cleared after successful order placement.

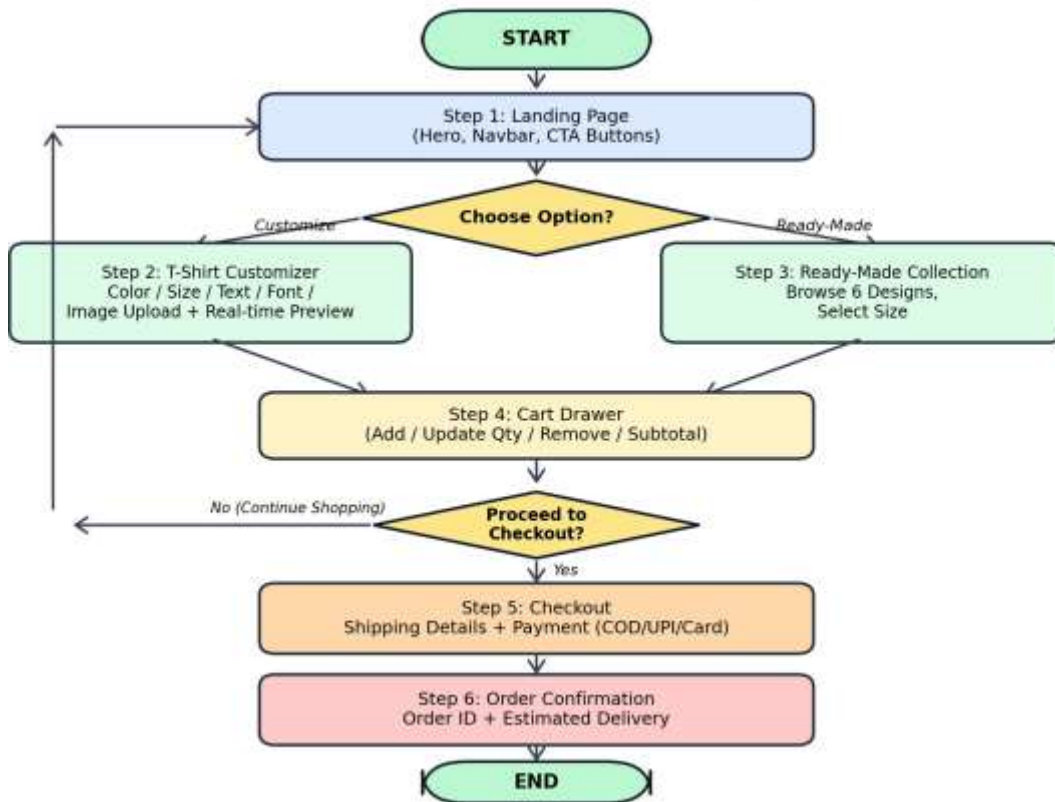


Fig. 2 User Workflow of the T-Shirt Ordering System

4.1 Key Implementation Code

The following code snippet demonstrates the Cart Context implementation using React's Context API and TypeScript interfaces:

```

// CartContext.tsx - Global State Management export interface CartItem {
id: string; name: string; price: number; color: string; size: string; quantity: number;
customText?: string; customImage?: string; isCustom: boolean;
}
const CartProvider = ({ children }) => {
const [items, setItems] = useState<CartItem[]>([]);

const addItem = (item: CartItem) => { setItems(prev => {
const existing = prev.find(i =>
i.id === item.id && i.size === item.size);
if (existing) {
return prev.map(i => i.id === item.id
? { ...i, quantity: i.quantity + item.quantity } : i);
}
return [...prev, item];
});
};
// ... quantity update, remove, clear methods
};
  
```



5. Results and Discussion

The developed system was tested across multiple browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, tablet, mobile) to evaluate functionality, performance, and user experience. Table 2 presents the testing results across different modules.

Module	Feature Tested	Status	Remarks
Navbar	Navigation, Cart Icon, Mobile Menu	✓ Pass	Smooth scroll, responsive
Hero Section	CTA Buttons, Animations	✓ Pass	Float animation, responsive
T-Shirt Customizer	Color, Size, Text, Image Upload	✓ Pass	Real-time preview works
Ready-Made Collection	Browse, Add to Cart	✓ Pass	6 designs available
Cart Drawer	Add/Remove/Update Items	✓ Pass	Quantity & price updates
Checkout	Form Validation, Order Placement	✓ Pass	All fields validated
Order Confirmation	Order ID, Delivery Date	✓ Pass	Unique ID generated
Responsive Design	Mobile, Tablet, Desktop	✓ Pass	All breakpoints tested

Table 2: Module-wise Testing Results

5.1 Performance Metrics

Metric	Value	Benchmark
Initial Load Time	1.2 seconds	< 3 seconds
Time to Interactive	0.8 seconds	< 2 seconds
Bundle Size (gzipped)	142 KB	< 500 KB
Lighthouse Performance	94/100	> 90/100
Lighthouse Accessibility	96/100	> 90/100
Mobile Responsiveness	100%	100%

Table 3: Performance Metrics of the Developed System

The results demonstrate that the system meets or exceeds all performance benchmarks. The use of Vite as the build tool contributes to fast initial load times through efficient code splitting and tree shaking. React's Virtual DOM ensures smooth UI updates during customization interactions. Tailwind CSS's purge mechanism removes unused styles, resulting in a compact CSS bundle.

5.2 User Experience Analysis

The t-shirt customizer provides real-time visual feedback as users modify colors, text, and graphics. The component-based architecture enables independent module updates without affecting other system components. The Context API-based cart management ensures consistent state across all pages, preventing data loss during navigation. The checkout form implements comprehensive validation for all required fields, reducing form submission errors.

6. Conclusion

This paper presented the design and development of a Web-Based Customized T-Shirt Ordering System using modern front-end technologies. The system successfully implements a complete e-commerce workflow encompassing custom t-shirt design, ready-made collection browsing, cart management, and order placement.



The application demonstrates the effectiveness of React.js with TypeScript for building complex interactive web applications. The component-based architecture ensures code maintainability and scalability, while Tailwind CSS provides a consistent and responsive design system. The Context API proves adequate for state management in medium-complexity applications without the need for external state management libraries.

Key achievements of this project include: (1) a fully functional t-shirt customizer with real-time preview supporting color selection, text customization, and image upload, (2) a responsive design that adapts seamlessly to mobile, tablet, and desktop viewports, (3) a complete checkout flow with form validation and order confirmation, and (4) an optimized application with fast load times and high Lighthouse scores.

Future enhancements may include: (1) backend integration with a database for persistent order storage, (2) payment gateway integration for online transactions, (3) user authentication and order history tracking, (4) 3D t-shirt preview using WebGL or Three.js, and (5) social media sharing of custom designs.

References

- [1] Meta (2023). React – A JavaScript library for building user interfaces. <https://react.dev>
- [2] Microsoft (2023). TypeScript: JavaScript with Syntax for Types. <https://www.typescriptlang.org>
- [3] Tailwind Labs (2023). Tailwind CSS – Rapidly build modern websites. <https://tailwindcss.com>
- [4] Evan You (2023). Vite – Next Generation Frontend Tooling. <https://vitejs.dev>
- [5] Grand View Research (2021). Custom T-shirt Printing Market Size Report, 2021-2028.
- [6] Nielsen Norman Group (2022). Product Customization UX: Best Practices for E-Commerce.
- [7] Rappin, N. (2021). Modern CSS with Tailwind. Pragmatic Bookshelf.
- [8] Kumar, A., Singh, R., & Patel, M. (2021). Development of Online Custom Merchandise Platform using MERN Stack. *International Journal of Web Engineering*, 15(3), 234-248.
- [9] Abramov, D. & Clark, A. (2015). Redux – A Predictable State Container for JS Apps.
- [10] Shadcn (2023). shadcn/ui – Beautifully designed components built with Radix UI and Tailwind CSS.