# Shopez E-commerce Project Documentation

## Introduction

Project Title: Shopez E-commerce

Team I’D:  PNT2025TMID09943

Team Members: **Varad Patil** – Frontend Developer (React.js)

**Rajveer Jadhav** – Backend Developer (Node.js / Express.js)

**Sejal Akare** – UI/UX Designer

**Kartik Kamble** – QA & Testing Engineer

## Project Overview

Purpose: Shopez E-commerce is an online shopping platform aimed at providing a seamless and user-friendly experience for customers to browse, search, and purchase products online. The goal is to create a responsive and secure application with efficient navigation and smooth transactions.

Features: User authentication, product catalog, advanced search and filter, cart and checkout system, order history, and responsive design.

## Architecture

Component Structure: The application follows a modular React component structure with reusable UI components such as Header, Footer, ProductCard, and CategoryList. Pages like Home, ProductDetails, Cart, and Checkout interact with these components.

State Management: The Context API is used for managing global states such as user authentication, cart items, and order history.

Routing: react-router-dom is used to handle navigation between different pages.

## Setup Instructions

Prerequisites: Node.js, npm

Installation:  
1. Clone the repository.  
2. Navigate to the project directory.  
3. Run npm install to install dependencies.  
4. Set up environment variables.  
5. Run npm start to start the server.

## Folder Structure

Client: Contains src/components for UI components, src/pages for different views, src/assets for images and styles, and src/context for global state management.

Utilities: Includes helper functions for API calls and custom hooks for state handling.

## Running the Application

Frontend: Navigate to the client directory and run npm start.

## Component Documentation

Key Components: ProductCard (displays product details), Header (navigation and branding), Footer (site links), Cart (selected items and checkout process).

Reusable Components: Buttons, Modals, Form inputs.

## State Management

Global State: Managed via Context API for cart, authentication, and order tracking.

Local State: Managed using useState within individual components for UI-specific changes.

## User Interface

[Insert screenshots of Home page, Product Details, Cart, and Checkout]

## Styling

CSS Frameworks/Libraries: Styled-components and CSS modules are used for styling.

Theming: Custom color palette with light and dark mode support.

## Testing

Testing Strategy: Unit testing for components using Jest, integration testing for pages, and manual end-to-end testing.

Code Coverage: Jest coverage reports to ensure all critical features are tested.

## Screenshots or Demo

## Known Issues

Some product images may load slowly on low bandwidth. Payment gateway testing is limited to sandbox mode.

## Future Enhancements

Add AI-based product recommendations, enhance mobile UI, and integrate multi-language support.