

**Mini Project Report (KCS-554)**  
**on**  
**SHOPPER: An E-Commerce**  
**Submitted in partial fulfillment for award of**  
**BACHELOR OF TECHNOLOGY**  
**Degree**  
**In**  
**COMPUTER SCIENCE & ENGINEERING**



**2023-24**

**Under the Guidance of:**  
**Mr. Vineet Shrivastava**  
**Assistant Professor**

**Submitted By:**  
**Vivek Kr. Maurya (249)**  
**Yash Kr. Singh (253)**  
**Vivek Singh (251)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**RAJ KUMAR GOEL INSTITUTE OF TECHNOLOGY**  
**DELHI-MEERUT ROAD, GHAZIABAD**



**Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

# **SYNOPSIS**

Shopper is a dynamic E-Commerce mini project that utilizes React.js for the frontend and Express.js, Node.js, and MongoDB for the backend. This project focuses on fundamental functionalities, such as user authentication, a shopping cart system, and distinct product categories for men, women, and kids.

## **Key Features:**

### **1. User Authentication:**

Implement secure user registration and login.

Store user passwords securely in the MongoDB database.

### **2. Shopping Cart with React:**

Develop a dynamic shopping cart in React.js for real-time updates.

Enable users to easily add, remove, or modify items within the cart.

### **3. Product Categories:**

Organize products into different categories (men, women, kids) for efficient navigation.

Implement category-based filtering for an enhanced user experience.

## **Technical Implementation:**

### **1. React.js Frontend:**

Create a single page application and visually appealing user interface using React.js.

Utilize state management libraries like react-router-Dom for efficient data flow.

### **2. Express.js and Node.js Backend:**

Set up an Express.js server to handle HTTP requests.

Implement server-side logic for user authentication, cart management, and category filtering.

### **3. MongoDB Database:**

- Design and create a MongoDB database to store user information.
- Establish a connection between the Express.js server and MongoDB for data retrieval and manipulation.

#### **4. RESTful API:**

Develop a RESTful API to facilitate communication between the frontend and backend.

### **Testing and Deployment:**

#### **1. Testing:**

Conduct thorough testing of both the frontend and backend components.

#### **2. Deployment:**

Deploy the frontend on a static file server or services like Netlify or Vercel.

### **Learning Outcomes:**

Shopper provides students with a comprehensive understanding of building a modern E-Commerce application using React.js for the frontend and Express.js, Node.js, and MongoDB for the backend. This mini project emphasizes practical skills in frontend development, backend logic, and database management while incorporating industry-standard technologies.

# **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>SYNOPSIS</b>	<b>ii - iii</b>
	<b>LIST OF FIGURES</b>	<b>vii</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	PROJECT OVERVIEW	1
1.1.1	E-COMMERCE AND THE SHOPPER WEBSITE	1
1.1.2	FEATURES OF THE SHOPPER WEBSITE	1
1.1.3	USER AUTHENTICATION AND SECURITY	2
1.1.4	LIMITATIONS OF THE CURRENT SYSTEM	2
1.1.5	MOTIVATION BEHIND DEVELOPING SHOPPER	2
1.2	MOTIVATION AND OBJECTIVES	2
1.2.1	ADDRESSING DIVERSE CLOTHING NEEDS	2
1.2.2	OBJECTIVES OF THE "SHOPPER" PROJECT	3
<b>2.</b>	<b>HARDWARE AND SOFTWARE REQUIREMENTS</b>	<b>4</b>
2.1	HARDWARE REQUIREMENTS	4
2.1.1	SERVER INFRASTRUCTURE	4
2.1.2	CLIENT DEVICES	4
2.2	SOFTWARE REQUIREMENTS	4
2.2.1	OPERATING SYSTEM	4
2.2.2	BACKEND TECHNOLOGIES	5
2.2.3	FRONTEND TECHNOLOGIES	5
2.2.4	ADDITIONAL TOOLS AND LIBRARIES	5
2.2.5	DEVELOPMENT TOOLS	5
2.2.6	DEPLOYMENT AND HOSTING	5
<b>3.</b>	<b>DATA FLOW DIAGRAM</b>	<b>6</b>
3.1	FRONTEND	7

3.2 BACKEND	7
3.3 AUTHENTICATION AND AUTHORIZATION	7
3.4 USER DATA COLLECTION	7
3.5 DATABASE	8
3.6 EXPRESS	8
3.6 REACT	8
<b>4. PROJECT MODULES</b>	<b>9</b>
4.1 MAIN.JSX	9
4.2 APP.JSX	10
4.3 PRODUCT.JSX	10
4.4 CART.JSX	11
4.5 SIGNUP.JSX	11
4.6 LOGIN.JSX	12
<b>5. PROJECT SNAPSHOTS</b>	<b>13</b>
5.1 HOMEPAGE	13
5.2 PRODUCT CATALOG	13
5.2.1 MENS CATEGORY	13
5.2.2 FEMALE CATEGORY	14
5.2.3 KIDS CATEGORY	14
5.3 USER AUTHENTICATION SCREENS	15
5.3.1 SIGNUP PAGE	15
5.3.2 LOGIN PAGE	15
5.4 SHOPPING CART	16
5.4.1 ADD TO CART PAGE	16
5.4.2 CHECKOUT PAGE OF TOTAL CART AMOUNT	16
<b>6. LIMITATIONS</b>	<b>17</b>
6.1 TECHNICAL LIMITATIONS	17
6.1.1 SCALABILITY CHALLENGES	17
6.1.2 PERFORMANCE CONCERNS	17
6.1.3 INTEGRATION COMPLEXITY	17

6.2 USABILITY LIMITATIONS	17
6.2.1 USER EXPERIENCE CHALLENGES	17
6.2.2 DEVICE COMPATIBILITY	17
6.2.3 LEARNING CURVE FOR USERS	17
6.3 SECURITY LIMITATIONS	18
6.3.1 VULNERABILITIES AND THREATS	18
6.3.2 DATA ENCRYPTION CONCERNS	18
6.3.3 USER AUTHENTICATION RISKS	18
 <b>7. FUTURE SCOPE</b>	 <b>19</b>
 <b>8. CONCLUSION</b>	 <b>20</b>
 <b>9. REFERENCES</b>	 <b>21</b>

## **LIST OF FIGURES**

<b>CHAPTER NO.</b>	<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
3	Figure 3.1	Data Flow Diagram	6
4	Figure 4.1	Main.jsx	9
4	Figure 4.2	App.jsx	10
4	Figure 4.3	Product.jsx	10
4	Figure 4.4	Cart.jsx	11
4	Figure 4.5	SignUp.jsx	11
4	Figure 4.6	Login.jsx	12
5	Figure 5.1	Homepage	13
5	Figure 5.2	Product Catalog	13
5	Figure 5.2.1	Mens Category	13
5	Figure 5.2.2	Female Category	14
5	Figure 5.2.3	Kids Category	14
5	Figure 5.3	User Authentication Screen	15
5	Figure 5.3.1	SignUp Page	15
5	Figure 5.3.2	Login Page	15
5	Figure 5.4	Shopping Cart	16
5	Figure 5.4.1	Add to Cart Page	16
5	Figure 5.4.2	Checkout Page of Total Cart Amount	16