

Digital Tailoring Management System with Order Tracking

Kankal Prakash Nandkishor, Shinde Shubham Babarao, Chidre Gaurav Dagdu

Student, Information Technology

Vishweshwarayya Institute of Engineering And Technology, Almala, India

Abstract: *This paper presents the design and implementation of Kstitch: a Smart Tailoring Management System with Real-Time Order Tracking, developed to digitize and modernize traditional tailoring operations. Conventional tailoring systems rely on manual record-keeping, which leads to issues such as data loss, inefficient order tracking, and poor customer communication.*

The proposed system is built using the MERN stack (MongoDB, Express.js, React.js, Node.js) and introduces a dual-role platform for both customers and tailors. It enables digital order management, multi-item garment handling with flexible measurement inputs, and portfolio-based tailor discovery using geo-location features. A key contribution of this system is its offline-first architecture, where order data is temporarily stored in local storage when the backend server is unavailable and automatically synchronized once connectivity is restored. This ensures uninterrupted workflow, particularly in environments with unstable network conditions or server downtime.

The system also integrates features such as JWT-based authentication for secure access, Cloudinary-based image handling for portfolio management, and automated WhatsApp communication for order updates and invoice sharing. The implementation demonstrates improved efficiency in order handling, enhanced customer experience through real-time tracking, and reduced dependency on manual processes.

This solution provides a scalable and practical approach for small and medium-scale tailoring businesses to transition into a digital ecosystem while maintaining operational flexibility..

Keywords: Tailoring Management System, MERN Stack, Order Tracking System, JWT Authentication, MongoDB, React.js, Cloudinary Integration, Geo-Location Search.

I. INTRODUCTION

The Tailoring Industry, Particularly In Small And Medium-Scale Businesses, Continues To Rely On Traditional Methods Such As Manual Record-Keeping Using Physical Registers, Commonly Known As “Tailor Books.” While This Approach Has Been Followed For Decades, It Introduces Several Inefficiencies Including Difficulty In Managing Customer Data, Lack Of Proper Order Tracking, Risk Of Data Loss, And Limited Communication Between Tailors And Customers. Additionally, Customers Often Face Challenges In Discovering Skilled Tailors Based On Specific Requirements Such As Garment Type Or Specialization, Leading To A Time-Consuming And Inconvenient Experience.

With The Advancement Of Digital Technologies, There Is A Growing Need To Transform These Manual Processes Into Efficient And Scalable Digital Solutions. Existing Systems In The Market Primarily Focus On Either E-Commerce Clothing Platforms Or Generic Service Listing Platforms, But They Fail To Address The Unique Workflow Of Tailoring Businesses, Such As Handling Customized Measurements, Managing Multiple Garments Within A Single Order, And Maintaining Long-Term Customer Records.

To Address These Challenges, This Paper Proposes **Kstitch**, A Smart Tailoring Management System Developed Using The Mern Stack (Mongodb, Express.Js, React.Js, And Node.Js). The System Provides A Dual-Role Platform For Both Customers And Tailors, Enabling Digital Order Management, Multi-Item Garment Handling, And Real-Time Order



Tracking. It Also Includes Portfolio-Based Tailor Discovery Using Geo-Location Features, Allowing Customers To Find Suitable Tailors Based On Their Needs And Location.

Furthermore, The System Incorporates Secure Authentication Using Json Web Tokens (Jwt), Flexible Database Design For Dynamic Measurement Storage, And Integration With Communication Platforms Such As Whatsapp For Order Updates And Invoice Sharing. By Digitizing Traditional Tailoring Workflows, The Proposed System Improves Operational Efficiency, Enhances Customer Experience, And Reduces Dependency On Manual Processes.

II. LITERATURE REVIEW

THE Digital Transformation Of Service-Based Industries Has Led To The Development Of Various Platforms For Online Booking, Service Discovery, And Customer Management. SEVERAL Existing Systems Attempt To Address Problems Related To Service Accessibility And Management, But They Fall Short When Applied To The Specific Needs Of The Tailoring Industry.

PLATFORMS Such As Online Marketplaces And Service Listing Applications (E.G., Real Estate And Local Service Platforms) Provide Features Like User Registration, Search Filters, And Basic Profile Management. THESE Systems Allow Users To Discover Service Providers Based On Categories And Location. HOWEVER, They Are Primarily Designed For Standardized Services And Do Not Support Highly Personalized Workflows Such As Custom Garment Stitching, Where Each Order May Involve Unique Measurements And Multiple Items.

SOME Tailoring Management Solutions And Small-Scale Applications Focus On Digitizing Customer Records And Order Details. WHILE These Systems Improve Data Storage Compared To Manual Registers, They Often Lack Advanced Features Such As Multi-Item Order Handling, Structured Order Tracking, And Integrated Communication Mechanisms. ADDITIONALLY, Many Of These Systems Do Not Provide A Unified Platform That Connects Both Customers And Tailors, Limiting Their Usability In Real-World Scenarios.

Existing E-Commerce Platforms Offer Order Tracking And Payment Systems, But They Are Designed For Ready-Made Products Rather Than Customized Services. THEY Do Not Support Flexible Measurement Inputs Or Personalized Tailoring Workflows. SIMILARLY, Generic CUSTOMER RELATIONSHIP MANAGEMENT (CRM) Systems Provide Tools For Managing Customer Data But Are Not Optimized For Domain-Specific Requirements Such As Garment Measurement Variations And Tailoring-Specific Order Processes.

Based On The Analysis Of Existing Systems, It Is Evident That There Is A Gap In Providing A Comprehensive Digital Solution Tailored Specifically For The Tailoring Industry. THE Proposed System, KSTITCH, Addresses These Limitations By Integrating Features Such As Dynamic Measurement Handling, Multi-Item Order Management, Real-Time Order Tracking, And Portfolio-Based Service Discovery Within A Single Platform. THIS Makes It More Suitable For Real-World Tailoring Operations Compared To Existing Generalized Solutions.

III. METHODOLOGY

The proposed system, **Kstitch**, is designed using a modular and scalable approach based on the **MERN stack (MongoDB, Express.js, React.js, and Node.js)**. The system follows a **client-server architecture**, where the frontend and backend operate independently and communicate through RESTful APIs.

System Architecture

The system is divided into three main layers:

Frontend (Client Layer):

Developed using React.js, responsible for user interaction and dynamic UI rendering. It provides separate interfaces for customers and tailors, including dashboards, order forms, and tracking views.

Backend (Application Layer):

Built using Node.js and Express.js, it handles business logic, authentication, API routing, and communication with the database.



Database (Data Layer):

MongoDB is used as a NoSQL database to store user data, orders, measurements, and portfolio details in a flexible JSON-like format.

Functional Modules

The system is divided into the following core modules:

User Management Module:

Handles registration and login for both customers and tailors using JWT-based authentication to ensure secure access.

Order Management Module:

Allows tailors to create and manage orders. Each order can contain multiple garments, with individual measurement details stored dynamically.

Measurement Handling Module:

Uses flexible data structures (Map in MongoDB) to store varying measurement parameters for different garment types.

Order Tracking Module:

Provides real-time updates on order status such as order created, stitching in progress, and completed.

Portfolio & Discovery Module:

Enables tailors to upload images of their work and allows customers to search for tailors based on specialization and location.

Communication Module:

Generates WhatsApp message links for sending order updates and invoices directly to customers.

Data Flow Process

- The user interacts with the frontend interface (React.js).
- API requests are sent to the backend using Axios.
- The backend validates requests through middleware (JWT authentication).
- Business logic is executed in controllers.
- Data is stored or retrieved from MongoDB using Mongoose models.
- The server sends a response back to the client.
- The frontend updates the UI dynamically.

Security Mechanism

Passwords are encrypted using **bcrypt hashing** before storage.

Authentication is handled using **JSON Web Tokens (JWT)**.

Protected routes ensure that only authorized users can access specific functionalities.

Image Handling Strategy

Small images such as profile pictures are stored as Base64 strings.

Portfolio images are uploaded using Multer and stored on Cloudinary, with URLs saved in the database.

IV. IMPLEMENTATION

The implementation of **Kstitch** focuses on developing a fully functional web-based application using the MERN stack. The system is designed to handle real-world tailoring workflows, including customer management, order processing, measurement handling, and communication.



Frontend Implementation

The frontend of the system is developed using **React.js**, following a component-based architecture. The user interface is divided into reusable components such as forms, cards, dashboards, and navigation elements.

- **Pages:** Login, Registration, Dashboard, Order Management, and Profile
- **State Management:** React Hooks such as `useState` and `useEffect` are used to manage dynamic data and API calls
- **API Integration:** Axios is used to send HTTP requests to the backend
- **Routing:** React Router is used to navigate between pages without reloading

The frontend provides separate dashboards for tailors and customers, ensuring role-based interaction and smooth user experience.

Backend Implementation

The backend is developed using **Node.js and Express.js**, which handle all server-side operations and business logic.

- **Routing:** API endpoints are defined for users, orders, posts, and authentication
- **Controllers:** Handle logic such as order creation, status updates, and payment calculations
- **Middleware:** Used for authentication (JWT verification) and request validation
- **Error Handling:** Implemented using try-catch blocks and HTTP status codes

The backend ensures secure and efficient communication between the frontend and the database.

Database Implementation

The system uses **MongoDB** with **Mongoose** for data modeling.

- **Collections:** Users, Orders, Tailors, Posts
- **Dynamic Measurements:** Implemented using Mongoose Map data type to allow flexible key-value pairs
- **Relationships:** Orders are linked with users and tailors using unique identifiers

This structure allows efficient storage and retrieval of complex and variable tailoring data.

Order Management System

The order system is one of the core components of the application.

Each order can contain **multiple garments**

Each garment has **individual measurements and pricing**

Order status is updated in stages such as:

- Order Created
- In Progress
- Completed
- Delivered

The system also tracks payment details including total amount, advance payment, and remaining balance.

Authentication and Security

User authentication is implemented using **JWT (JSON Web Tokens)**.

On login, a token is generated and stored on the client side

The token is sent with every protected API request

Passwords are encrypted using **bcrypt hashing**

This ensures secure access control and protects user data.



Image Upload and Storage

The system uses a hybrid approach for handling images:

Base64 Storage: Used for small images such as profile pictures

Cloudinary Integration: Used for storing portfolio images

Multer Middleware: Handles file uploads before sending them to Cloudinary

This approach ensures efficient storage and fast image access.

Communication Integration

The system integrates **WhatsApp messaging** for customer communication.

Automated message links are generated using predefined templates

Tailors can send order updates and invoices directly to customers

Reduces manual communication effort

User Interface and Experience

Clean and responsive UI design

Dashboard-based workflow for tailors

Easy navigation between modules

Visual order tracking system inspired by e-commerce platforms

V. RESULTS AND DISCUSSION

The implementation of **Kstitch** was tested across multiple scenarios to evaluate its functionality, usability, and performance in real-world tailoring operations. The system was successfully able to handle core operations such as user authentication, order creation, multi-item garment management, measurement storage, and real-time order tracking.

Functional Results

The system was tested with different user roles, including both customers and tailors.

Tailors were able to:

- Create and manage orders efficiently
- Store and retrieve customer measurements dynamically
- Update order status at different stages
- Generate WhatsApp messages for customer communication
- Customers were able to:
 - Search for tailors based on specialization and location
 - View tailor portfolios
 - Track order progress through different stages

These functionalities confirm that the system meets its primary objective of digitizing traditional tailoring workflows.

System Performance

The application demonstrated smooth performance under normal usage conditions:

Fast UI rendering due to React.js component-based architecture

Efficient API response handling using Node.js and Express.js

Quick data retrieval from MongoDB due to flexible schema design

The system performs well for small to medium-scale usage, making it suitable for local tailoring businesses.

Usability and User Experience

The user interface was designed to be simple and intuitive:

Copyright to IJARSCT

www.ijarsct.co.in



DOI: 10.48175/IJARSCT-31805



- Dashboard-based navigation allows tailors to manage orders easily
- Clear order status indicators improve tracking visibility
- Minimal learning curve for users unfamiliar with digital systems

The integration of WhatsApp messaging further enhances user convenience by simplifying communication between tailors and customers.

VI. DISCUSSION

The results indicate that the proposed system effectively addresses the limitations of traditional manual record-keeping. By digitizing order management and customer data, the system reduces the chances of data loss and improves operational efficiency.

The use of MongoDB enables flexible handling of varying measurement data, which is a critical requirement in tailoring systems. Additionally, the modular architecture ensures that the system can be extended with new features such as online payments or mobile application support.

However, the system is currently optimized for small-scale deployment and may require further enhancements such as scalability optimization and advanced analytics for handling large-scale operations.

VII. CONCLUSION

The development of **Kstitch** demonstrates an effective approach to digitizing traditional tailoring operations through a modern web-based system. The project successfully addresses key challenges associated with manual record-keeping, such as inefficient order management, lack of tracking, and difficulty in maintaining customer data.

By implementing a MERN stack-based architecture, the system provides a scalable and efficient platform for both tailors and customers. Features such as multi-item order management, dynamic measurement handling, real-time order tracking, and integrated communication through WhatsApp significantly improve the overall workflow and user experience.

The system proves to be a practical solution for small and medium-scale tailoring businesses, enabling them to transition from manual processes to a structured digital environment. It enhances operational efficiency, reduces errors, and simplifies customer interaction.

In conclusion, the project not only fulfills its intended objectives but also establishes a strong foundation for future enhancements such as mobile application development, online payment integration, and advanced analytics. This makes Kstitch a scalable and impactful solution in the domain of digital tailoring management systems.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my project guide for their valuable guidance, continuous support, and encouragement throughout the development of this project. Their insights and suggestions have been instrumental in shaping this work.

I am also thankful to the Head of the Department and all the faculty members of the Computer Engineering Department for providing the necessary resources and a supportive environment to complete this project successfully.

I extend my appreciation to my teammates for their cooperation, collaboration, and contribution during the project development process.

Finally, I would like to thank my family and friends for their constant support and motivation, which helped me complete this project.

REFERENCES

- [1]. MongoDB Documentation, “*MongoDB Manual*”. Available: <https://www.mongodb.com/docs/>
- [2]. Express.js Documentation, “*Fast, unopinionated, minimalist web framework for Node.js*”. Available: <https://expressjs.com/>



- [3]. React.js Documentation, “*React – A JavaScript library for building user interfaces*”. Available: <https://reactjs.org/>
- [4]. Node.js Documentation, “*Node.js Official Documentation*”. Available: <https://nodejs.org/>
- [5]. Mongoose Documentation, “*MongoDB Object Modeling for Node.js*”. Available: <https://mongoosejs.com/>
- [6]. Cloudinary Documentation, “*Cloudinary Image and Video Management*”. Available: <https://cloudinary.com/documentation>
- [7]. JSON Web Token (JWT), “*JWT Introduction*”. Available: <https://jwt.io/introduction>
- [8]. Axios Documentation, “*Promise-based HTTP client for the browser and Node.js*”. Available: <https://axios-http.com/>
- [9]. Bcrypt Documentation, “*A library to hash passwords*”. Available: <https://www.npmjs.com/package/bcrypt>
- [10]. WhatsApp Business API, “*Click to Chat Feature*”. Available: <https://faq.whatsapp.com/>

