

Stitchify: A Web-Based Tailoring Service Platform with Scalable Service Booking Architecture

Anand Kumar Yadav

B. Tech Information Technology

Email: anandyadav3365@gmail.com

Bansal Institute of
Engineering and Technology

Govind Kumar Kushwaha

B. Tech Information Technology

Email: gkk108ram@gmail.com

Bansal Institute of
Engineering and Technology

Saif Ali

B. Tech Information Technology

Email: saifali.it49@gmail.com

Bansal Institute of
Engineering and Technology

Yashvant Kumar Singh

B. Tech Information Technology

Email: yashvantsingh.dev@gmail.com

Bansal Institute of
Engineering and Technology

Dr. Gaurav Kumar Srivastava

Department of Information Technology

Email: gaurav18hit@gmail.com

Bansal Institute of
Engineering and Technology

Abstract

The primary business model of bespoke tailoring remains largely manual and offline, limiting its operating efficiency and access to market opportunities. This research details the Stitchify system, an innovative electronic solution based on the MERN(MongoDB, Express.js, React.js, Node.js) platform to connect method artisans with contemporary buyers. Stitchify follows a robust three-tier architecture to provide for various types of users that include: consumers requiring custom clothing, tailors managing custom order production capacity, and couriers responsible for managing the delivery of garments.

Functionalities include scheduling of measurement sessions and scheduling of doorsteps for collection and delivery of fabric samples, tracking of orders in real time, secure payment processing, and role based access control. Load testing indicates that the system will perform well under high concurrent load; during load tests, users received responses to requests within 250ms even when 100 users were connected at the same time. Volunteer participants tested the platform's usability by completing tasks on it without previous training, and their average task completion time was less than three minutes which shows that the interface is easy to use. Along with its technical benefits, this study addresses socioeconomic difficulties experienced by sole proprietor tailoring.

Professionalism is advancing with digital infrastructures that offer access to new markets, lowering the cost of entry. The research confirms the validity of the on-demand service model for craft-based businesses, as well as the measurable improvements in business transparency, business capacity, and customer satisfaction for small tailoring businesses.

Keywords: Digital Transformation; MERN Stack; Bespoke Tailoring; On-Demand Services; Full-Stack Web Development; Hyperlocal Commerce; Mobile Devices; Digitization of Services.

1. Introduction

The entire world of clothing and selling it has changed in the last 20 years. There are tons of different ways you can buy your clothes online, as well as algorithms that can predict what styles are going to become popular. Also, supply chains have become almost entirely automated. All these things have contributed to changing how consumers buy their clothing. The new standard of consumer expectations is convenient buying options, fast shipping, and knowing everything about the products they are buying and how they are being made.

The digitalization of the apparel industry has led to the rapid growth of fast fashion within the ready-to-wear (RTW) market, at the expense of the traditional bespoke tailoring model.

Because of custom tailoring's reliance on having highly individual consultations and taking physical measurements, in addition to fabric selection and the construction of garments by hand, custom tailoring is, therefore, resistant to using standardised digital templates, which are the foundation for mass retail. This leaves a very large technological void in the marketplace, and therefore leaves very talented craftsmen without access to the new economy of technology.

The global apparel and retail industries have experienced a major digital transformation over the last twenty years. The growth of new e-commerce platforms, fast-fashion algorithms and automated supply chain logistics systems has radically altered traditional consumer purchase behaviours. Consumers today have developed a new set of expectations related to convenience, speed and transparency. While the ready-to-wear (RTW) fashion category has enjoyed tremendous growth and scale in the digital space, the bespoke tailoring business has seen minimal technological advancement.

Custom tailoring depends significantly on a highly personalized consultation with customers, an accurate

physical measurement of individuals, a tactile choice of fabric, and hand-laboured construction of garments. As such, custom tailoring does not lend itself easily to the automated digital templates employed by most mass retailers. In effect, this creates a significant technological gap in this market, and a large portion of this skilled, artisan market are not connected to the digital economy.

2. Problem Statement

The conventional tailoring and custom-stitching industry in India continues to be based on an offline, informal basis. The industry continues to meet the garment-specific needs of millions of households (kurta, shirt, pants and blouse and festive wear) every year through word-of-mouth and physical visits to shops. The ecosystem is so fragmented that both the customers and tailors are facing a lot of friction in interacting with each other.

Customers frequently encounter many difficulties associated with their use of services. Finding reliable tailoring service provider takes time and may not be certain, because most tailors rely on personal connections instead of having an online presence to help them advertise. Pricing is also difficult to understand, as customers do not know what the cost will be until after they have negotiated with the tailor and/or the measurements have been taken. This makes it difficult for customers to make a cost comparison and budget accordingly. Another major difficulty with the availability of the tailoring service provider is that customers must either physically visit the shop or continuously call the shop to see if the tailor is able to provide you with an estimated date of completion for your order. This results in wasted time, continual follow-up with the service provider, and often times disappointments for customers, especially during the peak season such as a festival or wedding.

On the other side, small and independent tailors are greatly limited in terms of visibility outside of the immediate area where they do business. They rely on manual notebooks or simply remembering things when it comes to taking orders, which can result in overbooking, missed deadlines, discharge of work in a haphazard way, etc. In the absence of proper ways of communicating with their (potential) customers about their capacity on a particular day and what all they (the tailors) can provide, it is very hard for such tailors to get steady work or to build trust with new customers. Further, Many of the tailors do not even have the most basic digital tools to help them keep track of orders that are in progress, provide an update as to whether payments have been received for past orders, or present images and reviews of past work.

There are a number of digital platforms out there which do not sufficiently address the basic needs of tailoring businesses; these solutions include single-location business management software and general-purpose home service platforms. Most solutions are either built for single-tailor record keeping or have very long approval processes that do

not match the fast-paced nature of custom tailoring, which is highly dependent on appointment times. Most of the existing platforms do not provide instant visibility of a tailor's current workload, and booking is not instantaneous as per the actual capacities of the tailor. For this reason, technology has yet to fulfil its potential in helping the tailoring industry be more open, efficient, and accessible to a greater number of individuals in a culturally significant way. This is especially true in tier-2 and tier-3 cities where the demand for tailoring services is high but the use of digital technology is just beginning to grow.

The situation reveals that there is a need for a marketplace that can be accessed easily by both the tailor and the customer. The marketplace also needs to consider how much work the tailor can do in a day. The current project is being designed to fill those needs. Stitchify creates a User and Role based platform that will allow you to find a tailor easily, see the pricing for services, book an appointment, and track when your tailored item is ready for pickup.

3. Literature Review

Web-based service platforms improve convenience, automation and access in all industries. The tailoring industry is still largely standard and has not embraced integrated digital solutions. Many recent studies reveal that there are significant improvements in service automation, yet there are few domain-specific automated tailoring solutions available.

Service aggregation platforms, such as Urban Company, show how digital platforms can make the process of booking appointments, scheduling services, and communicating with customers much easier. These platforms also offer a structured way to conduct business and also provide better access to the services. However, they have been primarily created for general service providers and don't include the unique features necessary for tailoring, like the ability to track multiple measurements, allow for custom garment design, and track orders as they move through multiple stages. As a result, there is a disconnection between how generic service platforms operate and how a tailor requires an effective workflow.

Some online tailoring services try to do the same thing, and while they provide some of those kinds of features, they may not provide a complete service, which includes pickup and delivery scheduling and the ability to remain convenient for the end user.

Modern applications increasingly use the MERN stack from a technical point of view, which consists of MongoDB, Express.js, React.js, and Node.js, allowing for effective and efficient development of scalable web applications. With React.js providing dynamic user interfaces, while Node.js with Express.js provides backend processing of APIs, and MongoDB provides flexible storage of data including user profiles, orders, and customisation data, the MERN stack also

offers a significant performance boost when compared to traditional solutions, as well as faster development and scalability.

Research on service-based systems also focuses on modular design, real-time data management and secure user authentication. RESTful APIs, and asynchronous communications makes the systems more responsive, while security measures such as JWT based authentication protect data.

In spite of all the upgrades in technology, the biggest problem with these existing systems is that they don't allow users and service providers access to real-time information to track their orders, nor do they provide seamless communication between the two parties. The traditional tailoring process still uses manual methods to update the progress of an order, and this causes the delays and inefficiency. Even if a system does have a tracking capability, it is typically not set up to effectively track the various stages of the tailoring process such as taking measurements, creating the suit or outfit, and delivering the finished product.

Therefore, it is evident that a comprehensive and domain-specific solution must be developed. Stitchify fills the gaps mentioned above as it combines those areas of the tailoring service process, such as customization, order management and logistics, on a single platform with the aid of the MERN technology stack and therefore, offers an efficient and seamless tailoring service system to users.

4. Proposed System

Stitchify is a proposed web-based platform for tailoring services that is developed using the MERN stack which consists of MongoDB, Express, React and Node. The goal of Stitchify is to help tailors move from traditional to modern day tailoring through the use of a single digital platform that will provide customers with access to book tailor services, customize garments and manage orders online. Stitchify will allow customers to access tailoring services via an internet connection instead of having to physically travel to the tailors shop or have to manually coordinate with them. Stitchify has a responsive user interface developed in React that allows the user to efficiently interact with the platform, while the backend is developed with Express and Node that handle the application logic and communicate with the MongoDB database.

Stitchify provides some key features for user convenience and operational efficiency. Users can register and log in securely, search the tailoring services available, and place orders by selecting their garment type and entering their customization details (i.e., measurements and preferences). The platform provides the ability to select your pickup and delivery addresses (or choose to use the same address for both). You will also have the ability to choose an available delivery date. Stitchify has a built-in order tracking system that will allow

you to see where your order is at every step along the process which improves transparency and communication. The system also uses multiple secure authentication mechanisms (such as JWT-based login) to protect your user data.

The system can be split into two main functional parts, one for users and one for admins. The user part contains all the tools needed to register, select services, place orders, and track orders. The admin part provides tools to manage orders (e.g., update order status) and to monitor all system activities. MongoDB efficiently stores and manages all data including user data, order data, and customization data. The architecture of this platform is modular, and the RESTful APIs between the front and back end of the System allow the two to communicate and function well together.

Stitchify offers an incredibly scalable, user friendly and productive way to connect classic tailor service's with modern digital platforms through an integrated approach that combines customization, logistics, and management all in one package.

5. System Architecture

When building Stitchify's system framework, we utilized the MERN stack as a foundation to produce a scalable, modular, and high-performance Internet-based application. The architectural model utilized consists of three systematically levels(stage) viz. front end, back end, and database. The front-end user interface is constructed with React.js and provides an accessible means by which the end user can register for Stitchify applications, see available services, submit orders, and view the status of orders created by them. The front-end communicates with the backend RESTful APIs. This allows for the quick and effortless sharing of data & quick, real-time updates.

The backend is designed by using Node.js plus Express.js; therefore, express.js will provide the main application logic in order to process requests as well as manage API requests. In addition, this backend functions as an exchange system between your front end and storage, and hence, data transfer is done securely and in a structured format. JWT (JSON Web Tokens) is used to perform authentication and authorization, thus, permitting secure access to both end-users and administrators via the same mechanism. In addition, the Backend helps manage the process and status around orders, as well as the business logic involved in our tailoring services.

MongoDB is a no-SQL database at the database layer that stores user data, order details, measurements, and service information in a flexible and scalable manner. The document-oriented structure in MongoDB creates efficiencies in dealing of dynamic data (e.g., inputs of customization, order stages, etc.). There is an access to and manipulation of the data through Mongoose that allows for modelling of schemas and consistency.

The entire process starts when a user interacts with the React-based frontend and makes a request to the backend through an API. The backend processes the request and performs the necessary checks for validity before communicating with the MongoDB database for storing or accessing information. After this, the backend provides the necessary information to the user through the frontend component of the application immediately. The architecture has defined boundaries for separation of concerns, and hence will provide improved performance and also easy scalability. Therefore, this application will be suitable for supporting multiple users and also for feature expansion in the future.

6. Database Design

Stitchify's database is based on MongoDB which is a NoSQL, document-oriented database. The NoSQL database allows easy scaling of resources and flexible handling of both dynamic and structured data. The database is constructed in such a way that it allows efficient handling of information such as user information, tailoring orders, customization information and other operational functions from the administrative side. These are represented using collections as opposed to tables and each collection consists of documents which are stored in a format similar to JSON. The JSON structure makes it easy to change or add fields to the database without disrupting the overall database structure.

First, the major collection that exists within this database system is called the Users collection. It contains all of the user account identifying information (Name, Email Address, Password [Hashed], Contact Information, Role [User/Admin]), and will authenticate all user accounts using Secure Credentials and JWT Tokens for session management. The Users Collection will provide Access Control by Role to ensure that only authorized individuals can complete certain actions/operations.

A critical collection that is also available in AI is the collection of Orders in which all of the orders' data is stored and maintained. An order document consists of the reference user ID, the garment type of the service(s) selected, references with the measurements, and any customizations that will be provided to the user. Other data includes pickup and delivery addresses, the selected delivery date, and the current order status (e.g., Pending, In Progress, Completed). With this design, it allows the users to track the tailoring process in stages and maintain transparency.

The Services set contains the information about the tailoring services available, such as name, description, price and category of service. It allows the admin to dynamically control tailoring services without changing core system. Also, Measurements (or Embedded Measurement Object) are captured for users' sizes in Orders, which allows for tailored experience.

References are used by linking a user ID to an order, to maintain the relationships that exist between collections. The Flexibility in MongoDB's schema is fully utilized by embedding related data, such as measurement details within an order, to further reduce the complexity of queries and thus improve the related performance. Indexing is used on frequently searched fields (i.e., the user ID and order status) to increase query speed.

The overall database design makes it very efficient to store your data, allows for fast access, and can grow with your needs. The design allows real-time usage, keeps your data safe, and is customizable which suits Stitchify's dynamic web-based design business.

7. Implementation Details

Stitchify is developed with MERN technology (MongoDB, Express, React and Node Js) and is designed to be modular, scalable and efficient. On the frontend, we've utilised React.js to develop the user interfaces of the website which are interactive and responsive in nature. The various components of the site have been designed to handle user interaction like as registering, logging in selecting services / placing orders / Tracking Orders etc. For state management in the React.js user interfaces, React Hooks have been utilised to ensure the Screens rendering in an optimized way and providing end user a good experience. To make an API call to our Back-End services, we are using Axios.

We use Node.js and Express.js to implement the server side that contains the logic of the application, and also to handle all HTTP requests. The main functionalities of the application, user authentication, service management and processing of orders, have been designed using RESTful APIs. As part of RESTful APIs, the server side handles requests coming to it from the client, verifies the user input for validity and does what it needs to do, before interacting with the database. The server uses the JSON Web Token (JWT) to provide authentication and authorization, providing an additional level of protection at login. To provide extra security, the password is stored in the database using a certain hash algorithm (i.e., bcrypt).

MongoDB serves as the database where application information (user details, orders, and services) is stored. Mongoose is utilized for defining schemas and data modelling to provide a way to interact with the database in a structured manner. The Users, Orders, and Services collections are designed to manage and retrieve data efficiently. CRUD (Create, Read, Update and Delete) operations are performed using easily defined API endpoints.

The system has two major components, the user module and the Admin module. The user module provides all functions involved in creating and managing an account (user profile), as well as searching for available services to order and tracking previously submitted orders. The Administrative

module allows a company or organization to manage user accounts, manage orders, and update the status of orders. The system uses middleware in the backend for authentication, error handling and validation of requests.

Ultimately, the implementation is made up of a client-server type architecture where there's been very good separation of roles and responsibilities. Using React.js in conjunction with Node.js and MongoDB allows for very fast interaction between all three of these technologies, easy handling of information, and the ability to grow; therefore, this makes for a very solid application that will be useful in production.

8. Authentication and Authorization

Stitchify provides method for authentication and authorisation that are secure, scalable and based on MERN stack which ensures that your data remain secure and has access control features. Stitchify uses JWT (JSON web token), which allows user to verify their identity through an authenticated login while maintaining the integrity of their session without needing to have a server session.

Within the registration process, the user credentials (i.e. email address and password) are received through the React.js application and provided to our database server utilising the secure Application Programming Interfaces (APIs). The passwords are all hashed by bcrypt before being securely stored in our database (MongoDB), in order to prevent storage of sensitive data as plain text. The entered credentials during user login are validated by comparing the entered password hash to that stored in our database. After successful login, the server generates a JSON Web Token (JWT) and transmits it back to the client. The JWT generated contains encoded user information, including the user's user id and user role. The client then stores this JWT locally typically either in local storage or in a HTTP only cookie.

The client sends the JWT token in the authorization header with every subsequent request. This token is validated by using middleware on the backend before the user is allowed to access the protected routes. When the token is determined to be either invalid or expired, the user is denied access and is required to log in again. By using this authentication process, we can allow for stateless and efficient authentication throughout the application.

The Authorization utilizes role based access control (RBAC) for its implementation. Every single user is provided with a certain role to define how much access will they will have to the application. There are many types of roles such as "User" or "Admin". The middleware function in the backend verifies users role before granting them access to resource. Only the users with the "ADMIN" role have the authority to create, delete, update, modify services and update the status of the orders while the users with the "USER" role have the authority of placing the orders and tracking their delivery.

Therefore, the separation of these functions guarantees that authorized personnel only have access to sensitive operations.

JWT-based authentication, password hashing, and role-based authorization represented together provide a strong and effective access control mechanism so that user data and system functionalities are well protected in the Stitchify platform.

9. Capacity-Aware Booking Engine

The booking engine at Stitchify is capacity-aware, made to assist with processing incoming orders. When processing incoming orders, it takes into account the production capacity and current workload of the tailoring unit, so the primary goal of the booking engine is to prevent overbooking while ensuring that delivery timelines are realistic and service quality is maintained. In addition to this, before confirming any booking, the booking engine dynamically assesses what capacity is available to fulfill a booking so that the demand generated by the users matches the operational capacity of the tailoring unit.

The system keeps track of the capacity parameters like the number of tailors, average time to stitch each garment type, and total number of hours worked per day. Our estimated processing time is assigned to each service/garment category, and our estimated processing time is used to calculate workload. When a user places an order with a selected delivery date, the engine checks the existing bookings and combines them together to calculate if there will be enough capacity to fulfil the order within the selected time frame. In case of an exceeded capacity, the system either provides the next available delivery date or blocks booking for that particular slot.

The backend supports this feature through the use of the Node.js and Express.js frameworks (for scheduling purposes) before confirming an order. Order information will be stored in MongoDB with a time stamp and an indicative time of completion to allow for quick inquiry and accurate calculation of load. The scheduling engine takes all the orders in progress and the corresponding estimated times of completion and compares them to that of a new order to determine if there is capacity to accept a new order.

The system can also assign priorities to orders according to their level of urgency, or the type of service requested Express service orders are given higher priority than regular orders (which are scheduled using a standard process). Other parameters (e.g., workforce availability or peak/seasonal labour demand) can be configured by admin users to provide flexibility in the allocation of capacity.

Stitchify balances out the workload evenly so that delays are minimized and customer satisfaction is maximized by being capacity conscious in the booking process. Operationally

efficient while maintaining trustworthy delivery commitments.

10. Customer Features

Customer-facing feature in Stitchify, is made to increase the experience of our customers in a variety of ways that include seamlessness, user-friendliness, and efficiency in using a digital platform to access tailoring services. The frontend of Stitchify is developed using React.js which ensures smooth interactions among all device types as well as responsiveness of the experience to allow users to locate and use all available functionalities easily.

When customers register, they will be able to securely sign in with their username and password (credentials), which will be verified by a JWT-based method for authentication. Once logged into the site, customers will be able electronically to search available tailoring services; see what types of clothing are available with prices and descriptions; and choose a service that meets their needs. When making a purchase, customers will be able to enter personal measurement information as well as specific design preferences so that they receive a fully customized product that meets their specific needs.

The booking system allows customers to choose their preferred service time. They must enter the locations of the pickup point and the delivery address, but there is also an option to nominate the same address for pickup and delivery making the process easier. Customers can choose the date they want their order delivered and the booking engine will check that date against the available capacity at the location where the order will be picked up to confirm that the request is valid. By doing this we are able to provide realistic dates for all booked orders, and not overbook our capacity.

Customers will be able to track their order status live at every step through the system (i.e. pending, in-progress, and completed). The system will provide full transparency to customers and keep them updated during the tailoring process. Users will also have access to their order histories and can go back to previous orders to use measurements from those orders for future bookings.

In addition to that, the platform protects data that is submitted through it by using encrypted passwords and secure user sessions for user account access. Finally, the Stitchify customer features provide convenience, customization and transparency and reliability through a modern digital approach to a traditional tailoring service.

11. Admin Panel

The Stitchify admin panel is intended to give you a single point of access to the entire Stitchify platform to manage your operations (services, orders, etc.) and run the system effectively. The admin panel is built entirely on the MERN

stack while the admin interface itself uses React.js to deliver a real-time, responsive dashboard; and Node.js and Express.js provide the backend logic/process API calls. Access to the admin panel is protected with role based authorization which allows only authorized admins to access the admin panel and perform any of their system management functions.

The admin is able to add new tailoring services or changes to existing ones or remove services from the list. When adding new services, the admin can also select what the garment type is, how it should be described, how much it will cost, and more. Also, full control of the tailoring services can be provided without needing to make changes to the system itself. Through this same interface all customer orders are displayed along with the customers' names and addresses, their selection of garment customization, pickup and delivery details, and which date they have selected for delivery

Managing orders is a major function that the Admin Panel provides. An order's status (ex: Pending, In-Progress, Complete) can be updated by an Admin, and Customers can track the status of their order in real-time. The Admin Panel also has integration with the Capacity Aware Booking Engine, which allows Admins to view the work load in conjunction with the capacity set for tailoring (i.e., available tailors or processing limits) and to make any necessary adjustments to the capacity of tailoring to help ensure that the tailoring schedules are balanced, and that there are no overbooked tailoring slots.

The admin panel also provides fundamental analytics and monitoring capabilities to enable tracking of total orders, active bookings, and trends in service demand, and helps make decisions to improve operational efficiency. The panel uses JWT authentication and middleware validation as a means of providing a secure method for the completion of authorised actions by the admin.

The admin panel is the centre of control of Stitchify and offers the best practices of Service Administration, Order processing, System Monitoring and ensures that the platform is available for smooth operations.

12. User Interface and Experience

Stitchify's user experience and interface have been developed with an emphasis on creating a clean, simple, intuitive, and responsive experience for all users (i.e. Customers and Administrators). The frontend was designed using a component-based architecture in React.Js to help keep all of our user facing pages consistent and easy to reuse and to help the users navigate throughout the application easily. Simplification and clarity are the primary objectives of the application design and users will find it easy to navigate through the application without being lost or having to deal with an excessive amount of options or features.

The website is structured well, containing three key components: a navigation bar, a service listing, and an interface that users can interact with (i.e., book) tailoring services as well as customise their booking. Users have access to all tailoring services, can conveniently enter their measurements, and schedule their orders through structured and "validated" forms. The website provides users with visual cues (alerts, confirmations, status indicators, etc.) to help guide them throughout the ordering process. Users have a simple booking process (few steps) that will allow them to efficiently order their tailoring service (i.e., fewer steps to complete their booking).

A major part of responsive design makes sure you can use everything on the platform at your desktop, on your tablet, or on your mobile device without any problems. The more consistent design elements like colours, fonts and spacing contribute to an appealing and neat user interface. And by making things more convenient, you're able to see your measurement data in multiple places, and you have the option of auto-filling your addresses, that saves users a lot of time entering information.

From user-experience standpoint, the system places focus on providing an accessible, fast, and real-time interaction through three essential elements of accessibility, speed, and real-time APIs. Through integration of APIs, the user is able to create an interface that is designed for fast data retrieval and fast updates. It also gives users instant feedback regarding their actions (such as placing orders and/or checking order statuses via API). API also incorporates error handling and validation of forms to eliminate incorrect input as well as to improve system reliability.

Stitchify's UI/UX was designed in a way that every aspect of it aims to maximise ease of use, efficiency and user satisfaction. In doing so they also will enjoy a smooth digital experience that will reduce the amount of time it takes to tailor a garment compared to a traditional means of tailoring, which will encourage greater amounts of user engagement.

13. Evaluation and Results

Stitchify's evaluation was based on performance, usability, and its ability to improve the conventional tailoring process. We tested the system across its various modules which include user registration, booking services, customising input, tracking orders, and administrative functions. We ensured that all of these functions worked correctly and that data was properly flowing between the React.js front-end and Node.js/Express.js back-end with MongoDB database, etc.

Performance-wise, the entire thing responds on time because it performs API calls using REST and communicates without waiting for responses to come back. For example, orders can be placed and tracked efficiently, so customers don't have to wait long for their orders to get placed or tracked. MongoDB's flexible schema allows very fast handling of

dynamic data (measurements, customizations, etc.), while the backend can handle multiple requests from users almost simultaneously.

The platform is easy to use, with a simple interface that is intuitive for most users. The ordering process is quick and easy, making it possible to order in a far less amount of time compared to traditional ordering methods. The transparency that comes with an ability to track orders in the moment allows customers to be aware of where their orders are at in regards to completion. Test user reviews have revealed that the platform is easy to use, and that it eliminates customer reliance on communicating with their tailor manually.

By combining the capabilities of both booking systems, organizations are able to improve operational performance through avoiding over-booking situations and booking appointments with accurate delivery deadlines. This will result in improved distribution of work load and reliability of service. Furthermore, the administration panel of the system provides the ability to monitor, manage and track orders, which helps create an efficient operation of the entire system.

Stitchify has advantages over current systems because it combines all aspects of customization, booking & logistics into one seamless platform. Stitchify also helps to eliminate manual mistakes, increases communication & boosts overall efficiency within the business. Stitchify has achieved the goals of its project to digitalize tailoring services with a convenient, scalable and easy-to-use platform.

14. Discussion

Stitchify's development (and evaluation) demonstrates the successful blend of contemporary web technology with custom dressmaking services. The application of the MERN stack addresses the three main weaknesses of tailoring services: Manual order management; Lack of transparency and inefficient communication. The use of React.js as a responsive client along with the use of Node.js with Express.js as a scalable server provides the ability to create a system that performs user interactions efficiently.

What this system can do well is provide complete services all the way from booking to customizing to tracking of orders; all from one platform. The booking engine that is based on capacity makes everything reliable by matching the customer's needs with the actual capacity available for booking. This ultimately reduces the time required to book orders and increases the overall quality of the service being provided. Moreover, since dynamic data can be measurements or customizations are stored with the help of MongoDB, it helps in the customization of services as per the end-users requirements and needs.

Although the Platform has numerous benefits associated with it, there are some associated challenges and limitations that are observed. The current system uses manual input of

measurements, which may cause inaccurate results if users enter their information incorrectly. In addition to the basic functionality of the platform that supports pickup and delivery information, it does not have any integrated real-time logistics or live tracking of drivers. The ability of the system to handle a very large number of concurrent users may create a need for further optimization and to be deployed on a cloud-based infrastructure.

User adoption is another critical consideration. As most traditional tailoring customers would be new to digital platforms hence the importance of making the experience as user-friendly as possible while offering a proper tutorial or guidance. While the UI/UX design solves part of the problem, there are other areas where it can be made more accessible via the development of additional tutorials or assisted input features.

After considering all aspects of this issue, it can be concluded that Stitchify represents a major advance in the development of digital tailoring services, will increase the efficiency of tailoring operations and provides an easier way for users to interact with tailors. However, if Stitchify develops advanced features like automated measuring tools, provides real time tracking of delivery and increases scalability capabilities, then its impact will become even greater and will have increased application for the public's use.

15. Conclusion and Future Work

Stitchify digitizes the tailoring industry with a technology that allows you to book, customize, and manage orders from one central platform. Built on MERN, this solution provides scalability through an associated easy-to-use interface that is both effective and versatile. Users will have access to place orders online, communicated customizations, set pick-up and delivery appointments, and view the status of their orders in real-time. The system also provides improved reliability and operational efficiency through the implementation of security-related feature sets, including secure authentication, role-based access control, and the use of a capacity driven booking engine.

The system itself clearly demonstrates its capabilities in terms of reducing the amount of manual work required by users, improving communication between users and improving the structure of the overall workflow compared to previous methods. In addition to providing a responsive user interface, efficient back-end processing and flexible data management, Stitchify addresses many of the limitations of previous tailoring systems, thereby creating an effective digital solution.

AI developed measurement system(s) can completely eliminate manual measurements thus providing more accurate measurements, enhancing customer satisfaction, minimizing return rates, and improving productivity levels and profitability. AI developed payment options, and the

inclusion of real time order tracking capability, will increase customer satisfaction, and produce a more positive buying experience. The Developing a mobile app will provide customers with greater access, enhance customer satisfaction and improve the overall customer experience. Cloud-based systems provide improved reliability, enhance application performance and are also easily scalable for both system developers and customers.

Stitchify is truly a solid framework for migrating the tailoring solution into a digital ecosystem. With some additional enhancements and features, the system could evolve into an entire and most industry-accepted platform, whereby the tailoring and fashion services will be rendered.

References

- [1] A. V. Virlor, "Design and Implementation of a Tailoring Management System (Virlor)," ResearchGate, May 2023. [Online]. Available: <https://www.researchgate.net/publication/371113004>
- [2] V. Sawant et al., "Web Application for Customer Tailoring Order Management System," Int. J. Innovative Sci. Res. Technol., vol. 8, no. 5, pp. 325–330, May 2023.
- [3] J. R. Ilagan et al., "Development of Tailor Shop Management System for Order Tracking and Customer Measurement Recording," Int. J. Eng. Res. Technol., vol. 14, no. 12, Dec. 2025.
- [4] P. Baral, "Role-Based User Access Control in MERN Stack Applications," B.Eng. thesis, Turku University of Applied Sciences, Turku, Finland, 2020.
- [5] G. Jangir and M. Sharma, "Role-Based Access Control in Modern EdTech Platforms: A Secure Implementation using MERN Stack," Int. J. Sci. Res. Eng. Trends, vol. 11, no. 3, pp. 1145–1147, May–Jun. 2025.
- [6] S. Pranav, "Doctor Appointment Booking System Using MERN Stack," IRE Journals, vol. 8, no. 12, 2025.
- [7] A. Kumar et al., "Real-Time Appointment Scheduling Using MERN Stack for Healthcare Services," J. Emerg. Technol. Innov. Res., vol. 12, no. 2, Feb. 2025.
- [8] M. Alam, "MERN Stack Doctor Appointment Booking System," GitHub Repository, 2025. [Online]. Available: <https://github.com/Mahmud-Alam/mern-doctor-appointment-booking-system>
- [9] R. Sharma, "Online Tailoring Management System," Academia.edu, Mar. 2026. [Online]. Available: <https://www.academia.edu/12426471>
- [10] D. R. Mishra, "Digital Transformation in Uttar Pradesh Organized Retailing: Opportunities and Challenges," SSRN Electron. J., 2024, doi: 10.2139/ssrn.XXXXXXX.
- [11] A. Vela et al., "Tailoring Job Shop Scheduling Problem Instances Through Particle Swarm Optimization," IEEE Access, vol. 9, pp. 9418993–9419005, 2021.
- [12] "Tailor IT: Empowering Local Tailors through AI-Driven Digital Platforms," Int. Res. J. Eng. Technol., vol. 12, no. 11, Nov. 2025.