Project Title: TailorNest

Team Members: Achuth R Kumar, Alphin K Sajan

Problem Statement: Small tailoring businesses struggle to keep up with customer orders, maintain accurate records of custom measurements, and meet deadlines for delivery. Currently, orders are handled manually, resulting in errors and miscommunication. A system is required that allows customers to place orders online, manage their measurements digitally, and track order progress, while helping tailors streamline their workflow and meet delivery commitments.

1. Understanding the Problem Statement

a. Explanation of the Problem Statement

Small tailoring businesses often struggle with managing orders, tracking customer measurements, and meeting deadlines due to reliance on manual processes. These challenges create operational inefficiencies, frequent errors, and poor customer experiences, as handwritten measurement records can easily be lost or misinterpreted, leading to inaccurate fits and dissatisfied customers. The lack of a structured system makes it difficult for tailors to prioritize tasks, resulting in delayed deliveries and impacting customer satisfaction and retention.

Implementing a digital solution like TailorPro can help these businesses streamline operations, reduce human error, and improve profitability by digitizing measurement records, order tracking, and task prioritization. This not only enhances operational efficiency but also boosts customer satisfaction by providing accurate fits, timely deliveries, and real-time order tracking. TailorPro's intended users include tailors, customers, and support staff, all of whom will benefit from improved communication, organized records, and an overall more efficient workflow.

b. Key Requirements identified

For an MVP, the solution must address core functionalities essential to streamlining operations and enhancing customer satisfaction. Key requirements include an **order management module** where customers can place orders online, along with a **digital measurement management**

system allowing customers to input and update their measurements securely. An **order tracking feature** is also crucial, enabling customers and tailors to monitor order progress, ensuring transparency and accountability. The system should support **task prioritization and deadline reminders** for tailors to help them manage orders effectively and meet delivery timelines.

Additionally, the solution must offer a **centralized communication interface** to facilitate clear, direct interaction between customers and tailors, along with basic **record-keeping and reporting capabilities** for orders and measurements to track business metrics. To meet MVP and timeline constraints, the solution will focus on an intuitive, user-friendly interface for ease of use, scalable enough to allow future feature additions but streamlined enough to support a quick initial release.

2. Solution Overview

a. Solution Summary

The proposed solution is a comprehensive digital platform tailored to meet the specific needs of small tailoring businesses and their customers. This solution enables customers to place orders online, track their orders in real-time, and manage custom measurements digitally, eliminating manual errors and miscommunications. For tailors, the platform provides tools to streamline order management, view customer measurements, track order progress, and generate monthly revenue reports, all in one place. With added features like real-time notifications, customer reviews, and a shop discovery map, the solution enhances visibility and operational efficiency. This app empowers tailors to deliver high-quality, timely service while expanding their customer base and optimizing workflow, effectively bridging the gap between traditional practices and modern customer expectations.

b. Objective

The primary objective of this solution is to modernize and streamline operations for small tailoring businesses, enabling them to manage orders, customer measurements, and delivery timelines with greater accuracy and efficiency. By digitizing order placements and communication, the solution aims to reduce manual errors, improve record-keeping, and enhance customer satisfaction through reliable service. Expected benefits include:

i. **Improved Efficiency**: Tailors can handle more orders with fewer errors and better time management, allowing for a smoother workflow and increased productivity.

ii. **Enhanced Customer Experience**: Customers gain the convenience of online order placement, measurement management, and real-time order tracking, fostering trust and repeat business.

iii. **Business Growth**: Tailors can increase their visibility through features like customer reviews and a shop discovery map, reaching new customers and gaining insights into their sales performance.

3. Features and Functionalities

a. Core Features

i. **User Account Management**: Both tailors and customers can create and manage accounts with tailored access to functionalities, ensuring secure and personalized experiences. Tailors have access to order management tools, while customers can browse tailor profiles and place orders seamlessly.

ii. **Tailor Profile Display**: Customers can easily view detailed tailor profiles, including services, experience, ratings, and past work. This helps customers make informed decisions when selecting a tailor that meets their requirements.

iii. **Order Placement and Management**: Customers can place orders online without visiting the shop. The order placement system allows them to specify garment types, upload measurement details, and choose preferred materials or styles, simplifying the entire process.

iv. **Search and Filter Functionality**: A search feature enables customers to browse a variety of garments or styles they want to get stitched. Once a selection is made, they can filter and select from available tailors, streamlining the process of finding the right match.

v. **Order Tracking and Updates**: Customers receive real-time updates on their order status, from order confirmation to completion. Tailors can manage and track each order's progress, set deadlines, and prioritize tasks, ensuring orders are processed on time.

b. Additional Features

i. **Location-Based Shop Search**: Customers can search for tailors based on their geographical location, making it convenient to find nearby tailors or tailoring shops. This feature leverages location services to improve accessibility and convenience.

ii. **Tailor Review and Rating System**: Customers can rate and review tailors after completing an order. This builds a feedback system, allowing new customers to make better-informed decisions based on other users' experiences and ratings.

iii. **Tailor Reports and Analytics**: Tailors have access to a reporting dashboard where they can view performance metrics, including total orders, revenue generated, and order status summaries. This feature provides valuable insights into their business performance, helping them identify growth areas.

iv. **Dress-Based Tailor Search**: Customers can search for tailors based on specific garment types, such as suits, dresses, or traditional attire. This search filter helps customers find specialists for their specific tailoring needs, streamlining the order process and ensuring high-quality results.

c. User Flows

i. **Admin User Flow**:

- **Dashboard Access**: Once logged in, the admin is directed to an admin dashboard, displaying an overview of the platform's activity, including the total number of registered customers and tailors, total orders placed, and order details.
- **User Management**: The admin has the authority to view, monitor, and remove users (both customers and tailors) if necessary, ensuring platform standards and user quality are maintained.
- **Order Monitoring**: The admin can also view order details, allowing them to oversee the flow of activity and maintain oversight of overall service quality.

ii. **Customer User Flow**:

- **Home Page and Dress Selection**: After login, the customer lands on a home page featuring a selection of dresses to choose from. By selecting a specific dress, the customer can proceed to browse tailor shops that specialize in stitching the chosen garment.
- **Tailor Selection and Order Placement**: The customer can view tailor shop profiles, reviews, and ratings to select a tailor. Once a tailor is

chosen, the customer can place an order directly from the selected shop's page.

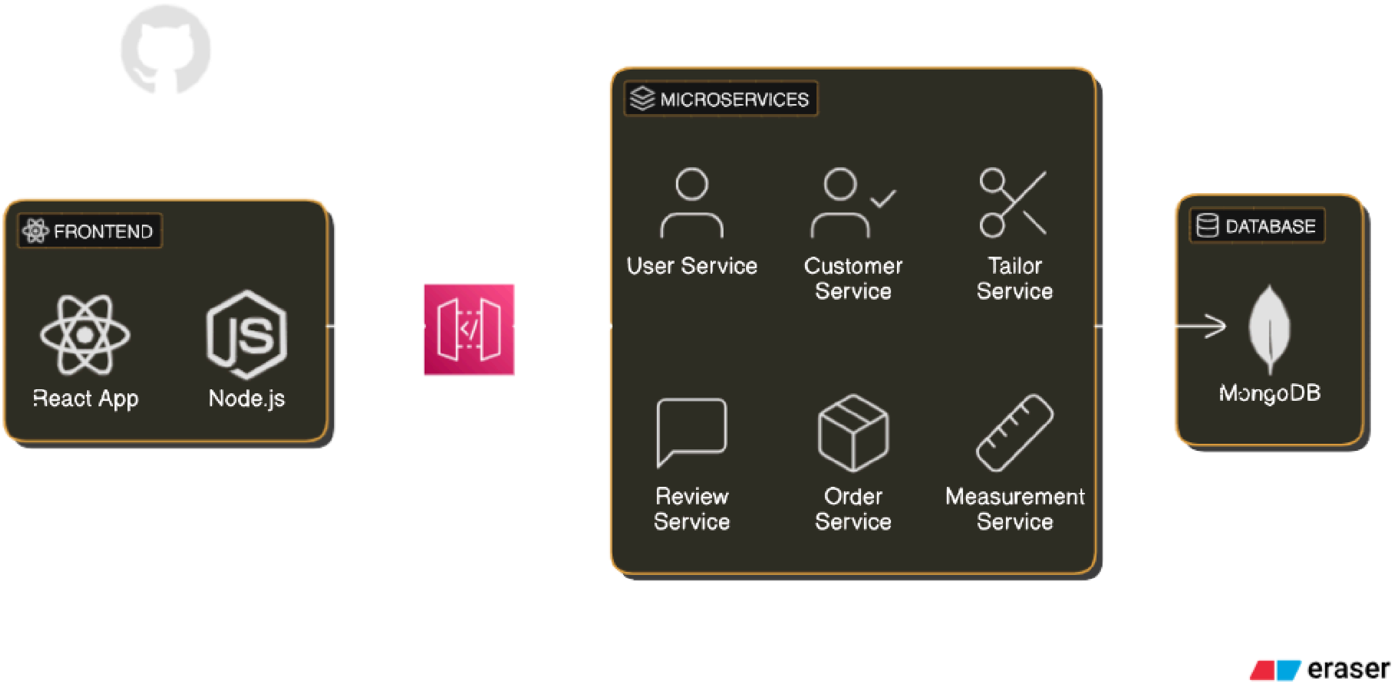
- **Customer Dashboard**: Customers have a dashboard where they can view their active and completed orders. For orders still pending, customers have the option to cancel if necessary. After an order is completed, the customer can leave a review for the tailor to share their experience with others.

iii. **Tailor User Flow**:

- **Order Management**: Upon login, the tailor is taken to a dashboard displaying current orders. Tailors can update the status of each order, moving it from "Pending" to "Accepted," "In Progress," "Completed," or "Canceled" as appropriate, ensuring customers are kept informed of their order's status.
- **Report Access**: Tailors also have access to a reports section, where they can view analytics on their order history, revenue, and performance, providing insights to help them grow their business and improve service quality.

4. **Architecture Diagram**

a. **System Architecture**



a. **Key Components**

- **Frontend (React with Chakra UI)**
  Purpose: Provides the user interface for customers, tailors, and admins to interact with the system.
  Technology: Built with React, styled using Chakra UI for responsive and accessible components.
  Functionality: Allows users to place and manage orders, view order status, chat with tailors, and search for tailoring services on a map.
- **Backend (Node.js and Express.js)**
  Purpose: Serves as the server-side logic, handling business processes, routing, and API requests.
  Technology: Developed with Node.js, using Express.js as the web framework.
  Functionality: Manages API requests from the frontend, interacts with the database, and provides endpoints for authentication, order management, and notifications.

- **Database (MongoDB)**
  Purpose: Stores and manages persistent data for the application.
  Technology: MongoDB, a NoSQL database optimized for scalable, flexible data storage.
  Data: Stores customer and tailor profiles, orders, measurements, reviews, and chat data.
- **Firebase (Authentication and Authorization**
  Purpose: Manages user authentication and secure access control.
  Functionality: Uses Firebase Auth to allow secure sign-in, sign-up, and role-based access for customers, tailors, and admins.

- **OpenStreetMap (OSM) API**
  Purpose: Provides map data for displaying nearby tailor shops and locations.
  Functionality: Enables location-based services within the app, allowing customers to find nearby tailors and view shop locations on a map.

- **GitHub (Version Control)**
  Purpose: Manages code versions and supports collaborative development.
  Functionality: Provides a central repository for the codebase, allowing team members to push updates, track changes, and handle deployments.
- **React Leaflet**
  Purpose: Integrates map functionality into the frontend using OpenStreetMap data.
  Technology: Built on Leaflet.js for adding interactive map elements.
  Functionality: Displays maps, markers, and pop-ups, allowing users to explore

locations and see nearby tailoring services.

1. Technical Stack

a. **Frontend**:

- **React**: The core frontend framework used to build a responsive and interactive user interface. React helps manage complex state and component-based structure, allowing for a modular, dynamic, and fast UI.
- **Axios**: A promise-based HTTP client used for API requests between the frontend and backend. Axios simplifies API calls, handles errors efficiently, and allows for secure communication with the backend.

b. **Backend**:

- **Node.js**: The server-side runtime environment that provides fast, scalable network applications using JavaScript. Node.js is optimal for handling concurrent requests and non-blocking operations.
- **Express.js**: A minimal and flexible Node.js web application framework that helps set up the backend server, handle routing, and manage API endpoints for interactions between the client and database.

c. **Database**:

- **MongoDB**: A NoSQL database chosen for its flexibility in handling unstructured data. MongoDB stores data in JSON-like documents, which is well-suited for managing dynamic and complex order data, customer profiles, and tailor information.

d. **Other Technologies and Tools**:

- **Firebase**: Utilized for authentication and authorization, providing secure sign-in methods for both customers and tailors. Firebase simplifies user management and supports different authentication mechanisms.
- **OpenStreetMap (OSM)**: An open-source mapping API used for locating tailoring businesses on a map, helping customers find nearby tailors and navigate to the chosen shop.

2. Prerequisites and Requirements

a. **Technical Requirements**:

- **Development Environment**: Visual Studio Code (VSCode) as the primary IDE for frontend and backend development.
- **Server Setup**: Node.js environment with Express.js for backend server setup, handling API requests and server logic.

- **Database**: MongoDB as the main database for storing customer, order, and tailor data, either via MongoDB Atlas (cloud) or a local instance.
- **Cloud Services**: Firebase for authentication and authorization services, managing user sessions and security.
- **Hosting & Version Control**: GitHub repository for code versioning, with GitHub Actions or other CI/CD tools for deployment.

b. **Data Requirements**:

- **Sample Data**: Test datasets for customers, orders, and tailors, including sample measurements and profiles for initial development and testing.
- **APIs**: Access to OpenStreetMap (OSM) through the React-Leaflet library for displaying and managing location data on the map.
- **Firebase Authentication**: Configuration for Firebase services to handle secure user login and role-based permissions.

c. **Access Permissions**:

- **GitHub Repositories**: Access to the main GitHub repository, including branch permissions for collaborative development.
- **CI/CD Pipelines**: Permissions for configuring GitHub Actions or other CI/CD tools to automate testing, deployment, and environment setup.
- **Cloud Services**: Access permissions to Firebase, MongoDB Atlas, and OSM APIs to manage project resources and data flow across environments.

d. **Other Dependencies**:

- **Libraries & Frameworks**: React for frontend development, Chakra UI for styling, Axios for API requests, and Mongoose for MongoDB interactions.
- **Map Integration**: React-Leaflet with Leaflet library to render maps and markers for tailor locations within the customer interface.
- **Plugin Versions**: Specific versions of key tools, including Node.js, MongoDB, and Firebase SDK, to ensure consistent environment setup.

3. Conclusion

In conclusion, this project provides a comprehensive digital platform to support small tailoring businesses by modernizing their order management and customer interaction processes. With features for online ordering, measurement tracking, real-time status updates, and a customer-tailor communication channel, the solution effectively addresses the key challenges faced by traditional tailoring services. By streamlining workflows and improving communication, this system not only enhances the customer experience but also enables tailors to manage orders more efficiently and grow their business. Ultimately, this platform bridges the gap between customers and tailors, bringing convenience, accuracy, and growth potential to the tailoring industry.