Mainfold

Case Studies

OUR CLIENTS

TranPort

Continental

EASTCOAST BEDDING

AYCS

Microsoft

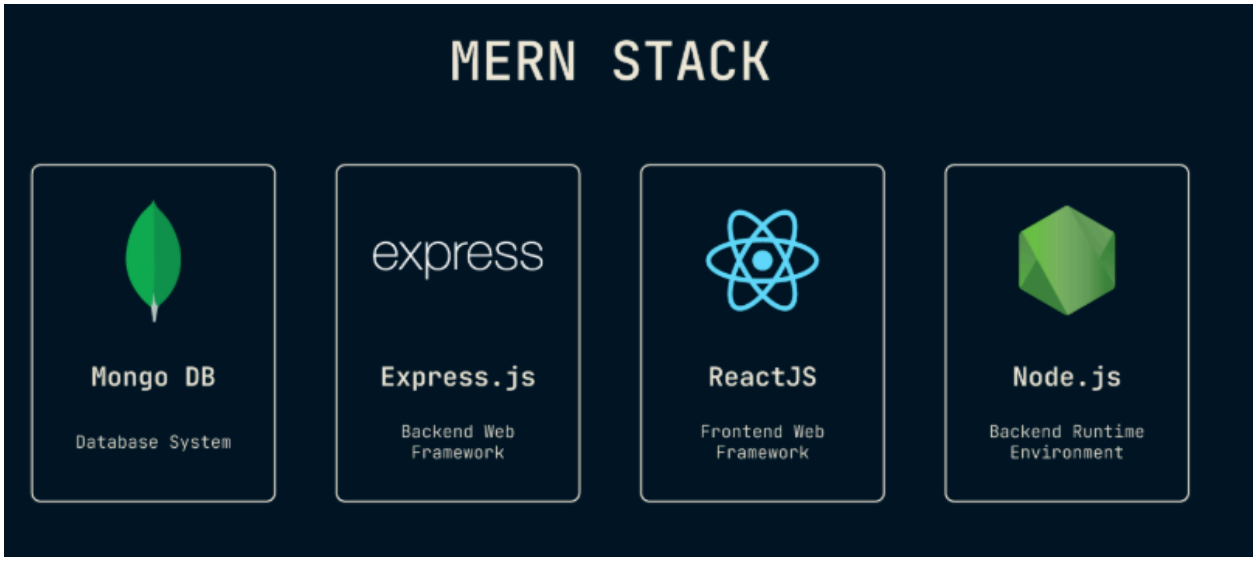
MDIPrep

🛏️ MERN Stack Case Study: Internal Operations App for Mattress Company

Developed by: Mainfold

Client: Leading Regional Mattress Manufacturer

Stack: MongoDB, Express.js, React.js, Node.js



🧩 Problem Statement

The client, a rapidly growing mattress manufacturing company, struggled with inefficient internal workflows across departments. Key pain points included:

- Manual tracking of raw materials and production status
- Disjointed communication between warehouse, production, and sales teams
- Lack of real-time inventory insights
- Delay in order processing and fulfillment
- No centralized dashboard for performance metrics

🎯 Objective

To build a **centralized web application** to streamline **production, inventory, sales**, and **order tracking**, accessible by various internal departments, ensuring real-time visibility and operational efficiency.

🔧 Tech Stack (MERN)

Layer	Technology	Purpose
Frontend	React.js + TailwindCSS	Dynamic UI with responsive dashboards
Backend	Node.js + Express.js	REST APIs to handle business logic
Database	MongoDB Atlas	Schema-flexible storage for inventory, orders, etc.
Authentication	JWT + bcrypt.js	Role-based access for Admin, Production Staff, Sales, etc.

📲 Core Features

1. Role-Based Login Dashboard

- Admin, Warehouse Staff, Production Managers, and Sales Reps
- Dynamic dashboard widgets based on role permissions

2. Inventory Management

- Track raw materials (foam, fabric, springs, etc.)
- Automated alerts for low stock
- Real-time quantity updates after usage or restock

3. Production Pipeline Tracker

- Status tracking for mattress batches (e.g., Cutting → Assembly → Packaging)
- Assign tasks and mark stages as complete
- Timeline view for managers

4. Order Management

- Internal order creation from sales team
- Track status from production to delivery
- Invoice generation & PDF export

5. Analytics Dashboard

- Production output (daily/weekly/monthly)
- Inventory turnover ratio
- Most ordered mattress models
- Average production time per unit

🔐 Security & Access Control

- JWT-based token authentication
- Password encryption with bcrypt
- API route protection for sensitive endpoints

🚀 Impact & Results

Metric	Before App	After Implementation
Production Delay per Batch	2-3 Days	< 1 Day
Inventory Error Rate	~15%	< 2%
Cross-team Communication Time	4–5 Hours	Real-time updates
Manual Reporting	Weekly	Auto-generated daily
Admin Satisfaction (Survey)	—	9.3/10

🛠️ Future Enhancements (Planned)

- Mobile app for factory floor staff (React Native)
- Integration with logistics partners for delivery tracking
- Barcode scanning for inventory input/output

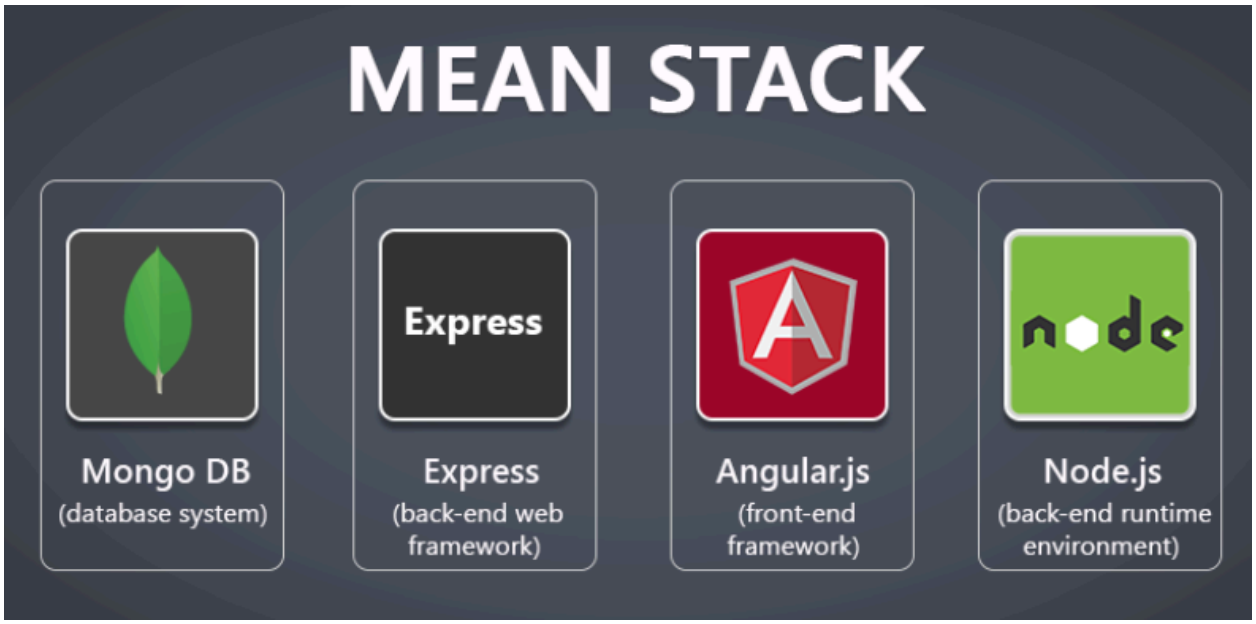
📢 Client Testimonial

> *“Mainfold transformed our internal operations with a system that fits like a glove. The transition to digital was seamless, and we now have full control and clarity over our daily operations.”*
> — Operations Head, Mattress Co.

🛡️ MEAN Stack Case Study: Insurance Management System

Developed by: Mainfold

Client: Mid-Sized Insurance Company (India-based)Stack: MongoDB, Express.js, Angular, Node.js



🧩 Problem Statement

The insurance company was managing policies, clients, and claims through fragmented tools and manual processes. Key challenges included:

- No centralized digital platform for managing policies, claims, or agents
- Delays in claim approval processes
- Difficulty in tracking policy renewals
- Poor customer data visibility
- Non-compliance with audit trail requirements

🎯 Objective

To build a secure, scalable **Insurance Management System (IMS)** to digitize and automate the full lifecycle of policy issuance, claim tracking, agent onboarding, and customer relationship management (CRM).

🔧 Tech Stack (MEAN)

Layer	Technology	Purpose
Frontend	Angular 15	Robust, modular UI with form-heavy components
Backend	Node.js + Express.js	API for business logic and data operations
Database	MongoDB Atlas	Flexible document storage for policies, claims, agents
Authentication	JWT + Role-based Guarding	Secure user access control across modules

---

📲 Core Features

1. Agent & Customer Management

- Register and onboard new agents with KYC validation
- Customer record creation and linking with active/inactive policies
- Agent hierarchy management (regional → zonal → individual)

2. Policy Lifecycle Automation

- Add/edit/remove policy types (health, auto, life, term)
- Assign policies to customers with custom terms
- Auto-reminder engine for renewals via email/SMS integration

3. Claims Processing Workflow

- Claim request submission (with file upload)
- Dynamic status stages: Initiated → Under Review → Approved/Rejected
- Manager override & audit logs

4. Dashboard & Reports

- Real-time metrics: Active Policies, Pending Claims, Agent Performance
- Downloadable reports for compliance teams (PDF, Excel)
- Custom filters by region, time period, claim type

5. Notification System

- Push notifications to agents/customers via in-app and SMS
- Renewal reminders, status updates, escalation alerts

🔐 Security & Compliance

- JWT-based secure login
- Field-level validation using Angular Reactive Forms
- Role-based route protection (admin, manager, agent, customer)
- Audit logs for all actions to support IRDAI compliance

🚀 Results & Impact

KPI	Before IMS	After Implementation
Claim Processing Time	7–10 Days	1–3 Days
Policy Lapse Rate	25%	< 8%

Agent Onboarding Time	1 Week	1 Day
Manual Paper Records	~100%	0%
Regulatory Compliance Score	~60%	95%+

💡 Special Integrations

- **Aadhaar + PAN API Integration** for KYC
- **SMS Gateway** for customer alerts
- **MongoDB Change Streams** for real-time UI updates in Angular
- **Role-Based Routing in Angular** to protect sensitive dashboards

---

🛠️ Planned Enhancements

- Mobile app for field agents using Ionic + Angular
- OCR for reading policy documents during claim submission
- Integration with insurance rating engines (third-party APIs)

📢 Client Feedback

> *"Mainfold delivered a fully digitized insurance operations suite that modernized how we serve customers and manage our network of agents. The MEAN stack was a game-changer."*
> — CTO, Insurance Company

👟 .NET Stack Case Study: Internal Operations App for Shoe Brand

Developed by: Mainfold

Client: Premium Footwear Brand (India-based)

Stack: ASP.NET Core, C#, SQL Server, Entity Framework, Angular

🧩 Problem Statement

The client, a well-established shoe manufacturing and retail brand, was experiencing challenges with siloed operations and lack of real-time visibility across departments:

- No unified system for production, stock, and distribution
- Manual coordination between warehouse, production, and store teams
- Frequent stockouts and overproduction due to poor demand forecasting
- Inconsistent SKU management and tracking
- No centralized internal app for process monitoring

🎯 Objective

To develop a secure, scalable **internal web application** for managing **inventory, production workflows, SKU tracking, distribution, and performance reporting** across manufacturing and retail units.

🔧 Tech Stack

Layer	Technology	Purpose
Frontend	Angular + Bootstrap	Modern UI with dynamic form components
Backend	ASP.NET Core (C#)	Business logic and API services
ORM	Entity Framework Core	Seamless data modeling & access
Database	Microsoft SQL Server	Structured storage of inventory, production logs
Authentication	ASP.NET Identity + JWT	Secure login, role-based access

---

📲 Core Features

1. User & Role Management

- Role-based access for Admin, Production Supervisor, Inventory Manager, and Store Manager
- Central login with ASP.NET Identity and two-factor authentication

2. SKU & Inventory Management

- Manage SKUs with variants (size, color, material)
- Real-time stock monitoring at multiple warehouses and retail outlets
- Low-stock alerts and reorder suggestions

3. Production Planning & Tracking

- Define production batches and track stages (cutting → stitching → finishing → QC)
- Assign tasks to departments and monitor timelines
- Barcode-based movement tracking

4. Distribution & Store Replenishment

- Auto-suggestion for restocking based on POS data from stores
- Track goods in transit and delivery confirmations
- Route-wise distribution mapping

5. Reporting & Dashboards

- Inventory turnover, production efficiency, and stock movement reports
- Custom filters by product category, location, or time frame
- Export to Excel, PDF formats

🔐 Security & Performance

- ASP.NET Core middleware for error handling and API security
- HTTPS enforced for all endpoints
- Role-based route guards in Angular
- SQL indexing and stored procedures for high-volume operations

---

🚀 Impact & Results

Metric	Before App	After Implementation
Inventory Accuracy	~70%	98%+
Order Fulfillment Time	4–5 Days	< 2 Days
Production Delay Rate	High (~25%)	< 5%
Admin Effort in Data Collection	Manual, 6–7 hrs/week	Automated, < 30 mins/week
User Satisfaction (Survey)	—	9.5/10

💡 Special Features Implemented

- **Barcode Scanner Integration** for inventory and production tracking
- **Task Notification System** with email and in-app alerts
- **Multi-location inventory support** (warehouses + retail stores)
- **Audit Trail Logging** for every major update and user action

---

🛠️ Future Enhancements (Planned)

- Mobile app for floor managers (Xamarin or MAUI)
- AI-based demand prediction module
- Integration with POS and ERP systems

📢 Client Testimonial

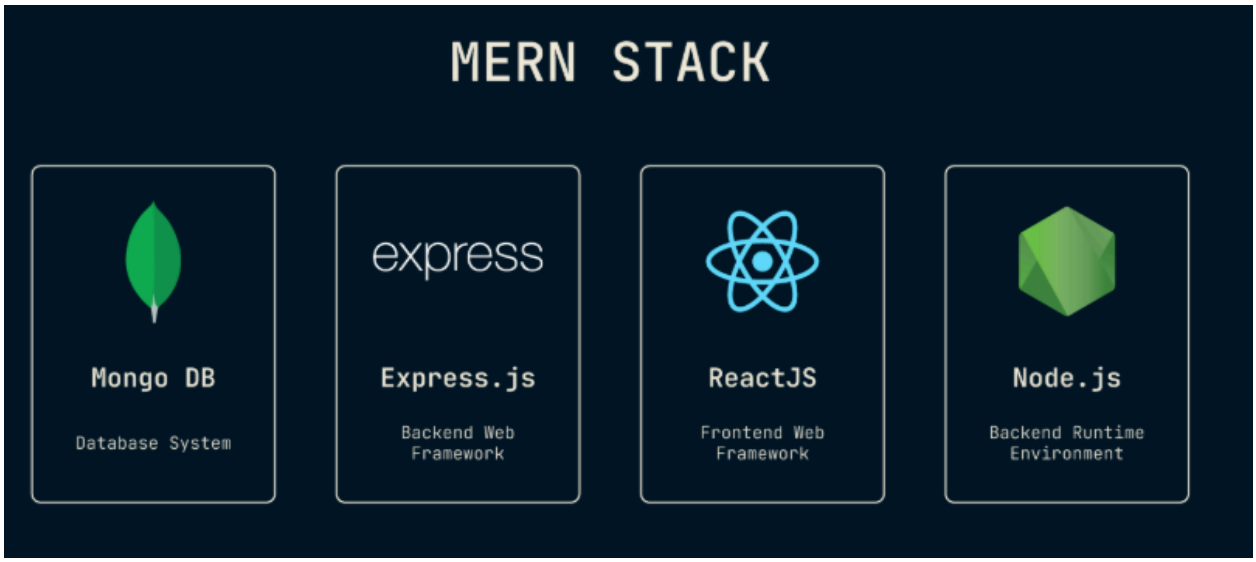
> *“The .NET-based internal system built by Mainfold gave us unprecedented control over production, inventory, and restocking. It's a powerful backbone for our multi-location operations.”*
> — Operations Head, Shoe Brand India

🎉 MERN Stack Case Study: Internal Event Operations Management App

Developed by: Mainfold

Client: National Events & Experiences Company (India)

Stack: MongoDB, Express.js, React.js, Node.js



🧩 Problem Statement

The client, a leading events management company handling corporate and social events nationwide, was struggling with fragmented communication, manual planning, and a lack of centralized project oversight. Specific challenges included:

- Event planning and vendor coordination were tracked in spreadsheets
- No centralized visibility on staff availability and task assignments
- Difficulty managing multiple events simultaneously
- Budget tracking and client deliverables were disorganized
- Communication gaps between internal teams (creative, logistics, production)

🎯 Objective

To develop a **centralized web-based platform** that enables seamless **event lifecycle management**, covering planning, task allocation, vendor handling, budgeting, and team collaboration — all from one internal dashboard.

🔧 Tech Stack

Layer	Technology	Purpose
Frontend	React.js + TailwindCSS	Clean, responsive UI with modular views
Backend	Node.js + Express.js	REST API for business logic and process automation
Database	MongoDB Atlas	NoSQL storage for flexible event records, task boards, users
Auth	JWT + bcrypt.js	Secure login and session handling
Extras	Socket.IO	Real-time task updates and communication alerts

---

📲 Core Features

1. Event Dashboard & Lifecycle Management

- Create and manage events with timeline view
- Track stages: Planning → Execution → Closure
- Assign responsible teams to each milestone

2. Team & Task Collaboration

- Assign tasks by event and department (creative, logistics, production)
- Real-time Kanban board with drag-and-drop updates
- Task deadlines, dependencies, and priority levels

3. Vendor & Supplier Management

- Vendor database with categories (sound, lighting, catering, venue, etc.)
- Contract uploads, payment tracking, and contact logs
- Rating system for post-event feedback

4. Client Brief & Deliverable Tracker

- Upload client briefs and attach to event records
- Track approvals, deliverables, and revisions
- Real-time notification for updates/approvals

5. Budgeting & Cost Management

- Budget planner with category breakdown
- Track actual vs estimated expenses
- Flag over-budget segments with alerts

6. Notifications & Calendar Sync

- Real-time alerts for new tasks or deadline updates
- Sync events and tasks to Google Calendar
- Daily summary email with pending actions

🔐 Security & Access Control

- Role-based access: Admin, Event Manager, Department Head, Team Member
- Encrypted passwords using bcrypt
- Protected API routes using JWT tokens

🚀 Results & Impact

Metric	Before App	After Implementation
Event Coordination Time	20+ Hours/Event	< 8 Hours/Event

Missed Deadlines	~30% Events	< 5%
Vendor Payment Errors	Frequent	Rare (<1%)
Team Communication Efficiency	Low	High (real-time)
Client Satisfaction (CSAT)	Not Tracked	92% (avg.)

💡 Unique Highlights

- **Modular architecture** to support hundreds of concurrent events
- **Socket.IO integration** for live collaboration across departments
- **Drag-and-drop task management** using `react-beautiful-dnd`
- **PDF generation** for event briefs, schedules, and vendor contracts

---

🛠️ Future Roadmap

- Mobile app for on-site staff coordination (React Native)
- AI assistant for budget forecasting & vendor suggestions
- Offline sync for remote/on-ground teams

📢 Client Testimonial

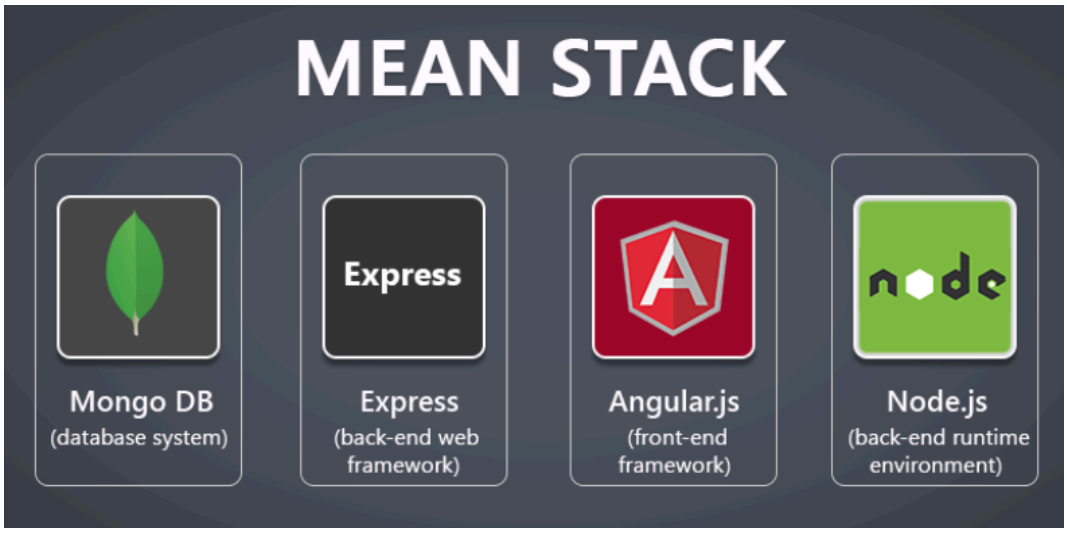
> *"Mainfold's event management system has transformed our internal workflows. What once took hours now happens in real-time. It's like having a digital control room for every event."*
> — Director of Operations, Events Co.

🚚 MEAN Stack Case Study: Transport Bidding Management App for ECTrans

Developed by: Mainfold

Client: ECTrans – Logistics & Freight Management Company

Stack: MongoDB, Express.js, Angular, Node.js



🧩 Problem Statement

ECTrans, a fast-scaling logistics provider, faced major inefficiencies in how it managed transport bids from third-party vendors for shipping routes. The challenges included:

- Manual bidding process via phone, emails, and spreadsheets
- No centralized platform to compare vendor bids per route or vehicle type
- Delays in vendor selection and route confirmations
- No historical data to analyze bidding trends or costs
- Lack of role-based access for internal teams and supervisors

---

🎯 Objective

To build a **robust internal web app** to manage **transport bidding** from registered vendors for different delivery routes, vehicle types, and cargo requirements — with automated comparison, approval workflows, and performance analytics.

🔧 Tech Stack (MEAN)

Layer	Technology	Purpose
Frontend	Angular 15 + Material UI	Fast, scalable, and responsive UI
Backend	Node.js + Express.js	RESTful APIs for bid handling and data operations
Database	MongoDB Atlas	Flexible document structure for bids, routes, vendors
Authentication	JWT + Role Guards	Secure access control for internal teams

---

📲 Core Features

1. Route & Load Management

- Add/manage delivery routes with origin, destination, distance
- Attach load details: weight, cargo type, packaging requirements
- Pre-set bidding deadline and route availability

2. Vendor Bidding Portal (Internal Use)

- Internal entry of bids from vetted vendors (via phone, portal, etc.)
- Each bid includes rate per km, vehicle type, driver info, terms
- Automatic flagging of lowest bid & historical bid comparison

3. Bid Approval Workflow

- Approval matrix: Supervisor → Manager → Operations Head
- Role-based UI showing only relevant routes and approvals
- Notification system for pending bid reviews and approvals

4. Vendor Database

- Track vendor performance: on-time delivery %, average rate, issues
- Upload documents (licenses, permits, contracts)
- Blacklist or prioritize vendors based on criteria

5. Analytics & Reporting

- Average bid rates per route
- Monthly savings through competitive bidding
- Vendor performance comparison charts
- Export data in Excel/PDF

🔐 Security & Access Management

- JWT-based user authentication
- Angular role guards for dashboard access (Manager, Bidder, Admin)
- MongoDB field-level validation for sensitive data (rates, personal info)

🚀 Results & Business Impact

KPI	Before App	After Implementation
Bid Processing Time per Route	4–5 hours	< 45 minutes
Cost Fluctuation Control	Unpredictable	15–20% savings avg.
Missed Bidding Windows	Frequent	< 1%
Data Accuracy & Vendor History	Manual/Scattered	100% centralized

Staff Satisfaction (Internal Poll)	—	9.4/10

💡 Key Highlights

- Built for scalability to support 1000+ monthly route bids
- Dynamic filters for city, vehicle type, or delivery urgency
- Auto-calculated per-km rate comparison table
- Angular reactive forms with built-in validation for accuracy

---

🛠️ Future Enhancements

- Vendor-side bidding portal (separate login)
- Real-time bid alerts via SMS/email
- GPS integration for delivery tracking
- AI-based vendor recommendations based on past route performance

---

📢 Client Testimonial

> *“Mainfold’s MEAN-stack bidding platform revolutionized how we manage our transport partners. We've gained transparency, speed, and cost efficiency — all in one app.”*
> — Logistics Director, ECTrans

Mainfold

Contact:
www.mainfold.in
info@mainfold.in

Ph: +91 8921364775