

MERN List Distribution System

Comprehensive Installation and Configuration Guide

Table of Contents

Prerequisites

Project

Structure

Backend Setup

Frontend Setup

Database Configuration

Running the Application

Testing the Application

Troubleshooting

Prerequisites

Required Softwares:

- Node.js (version 14)
- MongoDB (Local installation or MongoDB Atlas account)
- Git (for version control)
- Web Browser (Chrome)

Verify

Installation:

node --version

npm --version

mongod

-version

Project Structure

```
mern-list-distribution/  
├── backend/  
│   ├── models/  
│   ├── routes/  
│   └── middleware/
```



Backend Setup

1. Create backend directory and initialize npm.
2. Install dependencies (express, mongoose, bcryptjs, JWT, multer, csv-parser, xlsx, dotenv).
3. Create file structure: models, routes, middleware, server.js, .env.
4. Configure .env with PORT=5001, MongoDB URI, JWT secret.
5. Add scripts to package.json.
6. Create models (User, ListDistribution), middleware (auth), and routes (auth.js, agents.js, lists.js).
7. Setup server.js with Express, MongoDB connection, CORS, and routes.

Frontend Setup

1. Create React app.
2. Install dependencies: axios, react-router-dom.
3. Setup file structure: components, contexts.
4. Create AuthContext.js for authentication state.
5. Build components (Login, Dashboard, Agents, ListUpload, ListDistribution, MyLists, Navbar).
6. Configure App.js with router and protected routes.
7. Apply CSS for styling.

Database Configuration

Option 1: Local MongoDB

```
brew tap mongodb/brew
```

```
brew install mongodb-community
```

```
brew services start mongodb-
```

```
community mongosh
```

Option 2: MongoDB Atlas (Cloud) --- (Using)

- Create cluster on <https://www.mongodb.com/cloud/atlas>
- Update backend/.env with connection string.

Running the Application

Backend:(commands to start backend server)

```
cd backend
```

```
npm run dev
```

Frontend:(commands to start front-end server)

```
cd frontend
```

```
npm start
```

Access URL:

Frontend: <http://localhost:3000>

Backend: <http://localhost:5001>

Testing the Application

1. Create admin user with curl POST request.
2. Test login at <http://localhost:3000>.
3. Use dashboard, create agents, upload CSV files.
4. Verify distribution.

Sample CSV:

FirstName,
Phone,Notes

John Doe,1234567890,Important

client Jane Smith,0987654321,Follow

up required

Troubleshooting

Common Issues:

- Backend not starting ? process on port 5001.
- MongoDB connection failed ? Restart MongoDB service.
- CORS errors ? Verify config and ports.
- Frontend build errors ? Reinstall node_modules

Debugging:

- CORS ORIGIN ERROR, verify env variables, Added admin user using curl

Optimization:

- Use database indexing, caching.
- Build frontend for production.
- Secure JWT, env vars, and CORS settings.

Login Credentials

For admin-

Email - admin@system.com

Password - admin123

For Agent -

Email - admin@example.com

Password - admin123