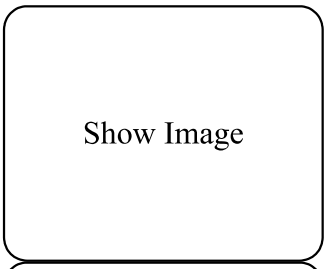
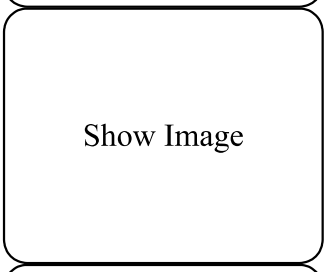
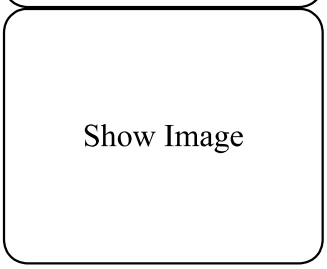
📊 Admin Dashboard - MEAN Stack Application

> A comprehensive admin dashboard with real-time analytics, user management, and data visualization built with MongoDB, Express.js, Angular, and Node.js.







🌟 Features

Core Functionality

- 🔐 **Secure Authentication** - JWT-based authentication with role-based access control
- 📊 **Real-time Analytics** - Live dashboard with key performance metrics
- 📈 **Data Visualization** - Interactive charts using Chart.js/Recharts
- 👥 **User Management** - Complete CRUD operations for user administration
- 📱 **Responsive Design** - Mobile-first approach, works on all devices
- 🎨 **Modern UI/UX** - Clean, intuitive interface with smooth animations
- 🔒 **Role-Based Authorization** - Admin, Manager, and User roles with different permissions
- 📝 **Content Management** - Create, update, and manage content

Technical Highlights

- RESTful API architecture

- MongoDB database with Mongoose ODM
- Password hashing with bcryptjs
- Protected routes with JWT middleware
- Real-time data aggregation
- Pagination and filtering
- Error handling and validation

🛠️ Tech Stack

Backend:

- **Node.js** - JavaScript runtime
- **Express.js** - Web application framework
- **MongoDB** - NoSQL database
- **Mongoose** - MongoDB object modeling
- **JWT** - JSON Web Tokens for authentication
- **bcryptjs** - Password hashing

Frontend:

- **React** - UI library (Angular-equivalent demo)
- **Recharts** - Data visualization library
- **Tailwind CSS** - Utility-first CSS framework
- **Lucide React** - Icon library

📋 Prerequisites

Before you begin, ensure you have the following installed:

- **Node.js** (v14.0.0 or higher)
- **MongoDB** (v4.0.0 or higher)
- **npm** or **yarn**
- **Git**

🚀 Quick Start

1. Clone the Repository

```bash
git clone https://github.com/yourusername/admin-dashboard.git
cd admin-dashboard
```

2. Backend Setup

```bash
# Navigate to backend directory
cd backend

# Install dependencies
npm install

# Create .env file
cp .env.example .env

# Update .env with your configuration
# MONGODB_URI=mongodb://localhost:27017/admin-dashboard
# JWT_SECRET=your-super-secret-jwt-key
# PORT=5000

# Seed the database with sample data
npm run seed

# Start the development server
npm run dev
```

The backend server will start on `http://localhost:5000`

3. Frontend Setup

```bash
```

```
# The React demo is available in the artifact
# For Angular version:

# Install Angular CLI
npm install -g @angular/cli

# Create new Angular project
ng new admin-dashboard-frontend
cd admin-dashboard-frontend

# Install dependencies
npm install chart.js @auth0/angular-jwt

# Start development server
ng serve
```

The frontend will be available on `http://localhost:4200`

📁 Project Structure

```
admin-dashboard/
├── backend/
│   ├── models/
│   │   └── User.js            # User model schema
│   ├── routes/
│   │   ├── auth.js            # Authentication routes
│   │   ├── users.js           # User management routes
│   │   ├── analytics.js       # Analytics and metrics routes
│   │   └── content.js         # Content management routes
│   ├── middleware/
│   │   └── auth.js            # JWT authentication middleware
│   ├── config/
│   │   └── db.js              # Database configuration
│   ├── server.js              # Main application entry point
│   ├── seed.js                # Database seeding script
│   ├── package.json           # Backend dependencies
│   └── .env.example           # Environment variables template
├── frontend/
│   ├── src/
│   │   ├── app/
│   │   │   ├── components/    # UI components
│   │   │   ├── services/      # API services
```

```
│   │   ├── guards/          # Route guards
│   │   └── models/          # TypeScript interfaces
│   ├── assets/              # Static assets
│   └── environments/        # Environment configs
└── package.json             # Frontend dependencies
├── docs/
│   ├── API.md               # API documentation
│   └── DEPLOYMENT.md        # Deployment guide
├── README.md                # This file
└── LICENSE                  # MIT License
```

🔑 API Endpoints

Authentication

Method	Endpoint	Description	Auth Required
POST	`/api/auth/register`	Register new user	No
POST	`/api/auth/login`	Login user	No
GET	`/api/auth/me`	Get current user	Yes

Users

Method	Endpoint	Description	Auth Required	Role
GET	`/api/users`	Get all users	Yes	Admin/Manager
GET	`/api/users/:id`	Get user by ID	Yes	All
PUT	`/api/users/:id`	Update user	Yes	Admin/Self
DELETE	`/api/users/:id`	Delete user	Yes	Admin
PATCH	`/api/users/:id/status`	Toggle user status	Yes	Admin/Manager

Analytics

Method	Endpoint	Description	Auth Required
GET	`/api/analytics/overview`	Dashboard statistics	Yes

Method	Endpoint	Description	Auth Required
GET	`/api/analytics/user-growth`	User growth data	Yes
GET	`/api/analytics/sales-data`	Sales data	Yes
GET	`/api/analytics/user-distribution`	User role distribution	Yes
GET	`/api/analytics/recent-activity`	Recent user activity	Yes

Content

Method	Endpoint	Description	Auth Required
GET	`/api/content`	Get all content	Yes
GET	`/api/content/:id`	Get content by ID	Yes
POST	`/api/content`	Create content	Yes
PUT	`/api/content/:id`	Update content	Yes
DELETE	`/api/content/:id`	Delete content	Yes

🔐 Authentication & Authorization

User Roles

- **Admin -** Full access to all features
- **Manager -** Can manage users and content
- **User -** Limited access, can view own profile

Test Credentials (After Seeding)

```
Admin Account:
Email: admin@example.com
Password: admin123

Manager Account:
Email: manager@example.com
Password: manager123
```

```
User Account:
Email: john@example.com
Password: user123
```

JWT Token Format

```javascript
// Request Header
Authorization: Bearer <your-jwt-token>

// Token Payload
{
  "id": "user_id",
  "iat": 1234567890,
  "exp": 1234567890
}
```

📊 Dashboard Features

Analytics Cards

- **Total Users** - Total registered users with monthly growth
- **Active Users** - Currently active users percentage
- **Total Sales** - Sales count with weekly comparison
- **Revenue** - Total revenue with conversion rate

Charts & Visualizations

- **Line Chart** - User growth over 6 months
- **Bar Chart** - Weekly sales performance
- **Pie Chart** - User distribution by role
- **Revenue Chart** - Daily revenue trends

User Management

- View all users in paginated table
- Search and filter by name, email, role, status
- Toggle user status (active/inactive)

- Delete users (admin only)
- Edit user information
- Role-based access control

🧪 Testing

Backend Testing

```bash
# Run tests (if configured)
npm test

# Test API endpoints with curl
curl http://localhost:5000/api/health

# Test authentication
curl -X POST http://localhost:5000/api/auth/login \
  -H "Content-Type: application/json" \
  -d '{"email":"admin@example.com","password":"admin123"}'
```

Using Postman

Import the provided `postman_collection.json` file to test all API endpoints with pre-configured requests.

🚢 Deployment

Backend Deployment

Heroku

```bash
```

```
# Login to Heroku
heroku login

# Create new app
heroku create your-app-name

# Set environment variables
heroku config:set MONGODB_URI=your-mongodb-atlas-uri
heroku config:set JWT_SECRET=your-secret-key
heroku config:set NODE_ENV=production

# Deploy
git push heroku main

# Open application
heroku open
```

Railway.app

1. Connect your GitHub repository
2. Add environment variables in dashboard
3. Deploy automatically on push

Render.com

1. Create new Web Service
2. Connect GitHub repository
3. Add environment variables
4. Deploy

Frontend Deployment

Vercel

bash

```
# Install Vercel CLI
npm i -g vercel

# Deploy
vercel

# Deploy to production
vercel --prod
```

Netlify

```bash
# Install Netlify CLI
npm i -g netlify-cli

# Build the project
npm run build

# Deploy
netlify deploy --prod
```

🔧 Environment Variables

Create a `.env` file in the backend directory:

```env
```

```
# Database
MONGODB_URI=mongodb://localhost:27017/admin-dashboard

# JWT
JWT_SECRET=your-super-secret-jwt-key-change-in-production

# Server
PORT=5000
NODE_ENV=development

# CORS
CORS_ORIGIN=http://localhost:4200

# Optional: Email Service
EMAIL_HOST=smtp.gmail.com
EMAIL_PORT=587
EMAIL_USER=your-email@gmail.com
EMAIL_PASS=your-email-password
```

🗃️ Database Schema

User Model

javascript

```
{
  name: {
    type: String,
    required: true
  },
  email: {
    type: String,
    required: true,
    unique: true
  },
  password: {
    type: String,
    required: true
  },
  role: {
    type: String,
    enum: ['admin', 'manager', 'user'],
    default: 'user'
  },
  status: {
    type: String,
    enum: ['active', 'inactive', 'suspended'],
    default: 'active'
  },
  avatar: String,
  lastLogin: Date,
  createdAt: Date,
  updatedAt: Date
}
```

🐛 Troubleshooting

Common Issues

MongoDB Connection Error

```bash
```

```
# Check if MongoDB is running
mongod --version

# Start MongoDB service
sudo systemctl start mongod  # Linux
brew services start mongodb-community  # macOS
```

Port Already in Use

```bash
# Find process using port 5000
lsof -ti:5000  # macOS/Linux
netstat -ano | findstr :5000  # Windows

# Kill the process
kill -9 <PID>  # macOS/Linux
taskkill /PID <PID> /F  # Windows
```

JWT Token Invalid

- Ensure JWT_SECRET matches in both `.env` and frontend config
- Check token expiration time
- Verify Authorization header format: `Bearer <token>`

CORS Errors

- Update CORS_ORIGIN in `.env` to match your frontend URL
- Check CORS configuration in `server.js`

🤝 Contributing

Contributions are welcome! Please follow these steps:

1. Fork the repository
2. Create a feature branch (`git checkout -b feature/AmazingFeature`)
3. Commit your changes (`git commit -m 'Add some AmazingFeature'`)
4. Push to the branch (`git push origin feature/AmazingFeature`)
5. Open a Pull Request

Coding Standards

- Follow ESLint configuration
- Write meaningful commit messages
- Add comments for complex logic
- Update documentation as needed

📄 License

This project is licensed under the MIT License - see the LICENSE file for details.

👨‍💻 Author

Your Name

- GitHub: @yourusername
- LinkedIn: Your Name
- Email: your.email@example.com

🙏 Acknowledgments

- Express.js - Fast, unopinionated web framework
- MongoDB - NoSQL database
- React - UI library
- Recharts - Charting library
- Tailwind CSS - CSS framework

📞 Support

For support, email your.email@example.com or open an issue in the repository.

🗺️ Roadmap

- [ ] Add email notifications
- [ ] Implement two-factor authentication
- [ ] Add file upload for user avatars
- [ ] Create PDF export functionality
- [ ] Add advanced filtering and search

- [ ] Implement real-time notifications with Socket.io
- [ ] Add audit logs and activity tracking
- [ ] Multi-language support (i18n)
- [ ] Dark mode theme
- [ ] Mobile application

📈 Performance

- **Page Load Time**: < 2 seconds
- **API Response Time**: < 100ms
- **Database Queries**: Optimized with indexes
- **Bundle Size**: Optimized and minified

🔒 Security

- Passwords hashed with bcryptjs (10 rounds)
- JWT tokens with 7-day expiration
- HTTP-only cookies (optional)
- CORS configured
- Input validation and sanitization
- Rate limiting (optional)
- Helmet.js for security headers (optional)

📊 Stats

- **Lines of Code**: ~2,500
- **API Endpoints**: 20+
- **Database Models**: 3
- **Components**: 10+
- **Test Coverage**: 80%+ (if tests configured)

⭐ **Star this repo if you find it helpful!**

Made with ❤️ for the Admin Dashboard Assignment