


# Arthomed Healthcare Backend

---

Comprehensive Documentation - README









Generated on 7/25/2025

**Arthomed Healthcare Backend!**  
**[[Node.js](https://nodejs.org/)](https://nodejs.org/)!  
[[Express](https://expressjs.com/)]  
(https://expressjs.com/)!  
[[MongoDB](https://www.mongodb.com/)]  
(https://www.mongodb.com/)!  
[[License](#)](LICENSE)A  
comprehensive healthcare  
management backend system  
built with Node.js, Express, and  
MongoDB, designed specifically  
for React Native applications.**

**Features role-based authentication, OTP verification, appointment management, and file upload capabilities.##** 

**Overview**Arthomed is a production-ready healthcare backend that provides:- Mobile OTP Authentication with Twilio SMS integration- Role-based Access Control (Admin, Doctor, Receptionist, Patient)- Appointment Booking System with real-time slot management- File Upload System for medical reports and prescriptions- Comprehensive API Documentation with

**Swagger/OpenAPI 3.0- Secure  
JWT Authentication with refresh  
token support## 📋 Table of  
Contents- [Features](#)- [Technology  
Stack](#)- [Architecture](#)- [Installation](#)-  
[Environment Setup](#)- [API  
Documentation](#)- [Database  
Schema](#)- [Authentication Flow](#)-  
[Role-based Access](#)- [Appointment  
System](#)- [File Upload System](#)-  
[Security Features](#)- [Testing](#)-  
[Deployment](#)- [Contributing](#)-  
[License](#)## ✨ **Features### Core  
Functionality- 🗝️ OTP-based  
Authentication - Secure mobile  
number verification- 👤 Multi-role  
User Management - Admin,****

**Doctor, Receptionist, Patient roles-**  **Appointment Booking - Real-time slot management and booking-**  **File Upload - Medical reports and prescription image uploads-**  **Dashboard APIs - Role-specific dashboards and analytics-**  **SMS Notifications - Appointment confirmations and reminders**### **Technical Features-**  **RESTful API Design - Clean, standardized endpoints-**  **API Documentation - Interactive Swagger documentation-**  **Security First - JWT tokens, rate limiting, data validation-**  **Mobile-First - Optimized for React**

**Native integration-  Production Ready - Error handling, logging, monitoring-  Database Optimization - Indexed queries and aggregations## **

**Technology Stack### Backend Framework- Node.js (18+) - JavaScript runtime- Express.js (4.19) - Web application framework- MongoDB Atlas - Cloud database service- Mongoose (8.x) - MongoDB object modeling### Authentication & Security- JWT - JSON Web Tokens for authentication- bcryptjs - Password hashing- express-rate-limit - Rate limiting middleware-**

**helmet - Security headers- cors -  
Cross-origin resource sharing###  
External Services- Twilio - SMS  
OTP delivery- Multer - File upload  
handling- Swagger/OpenAPI 3.0 -  
API documentation###  
Development Tools- nodemon -  
Development server- express-  
validator - Input validation-  
dotenv - Environment variable  
management## 🏗️  
Architecture### Project  
Structure**


```
arthomed-backend/├── src/|   └──
```

**## 🚀 Installation###**

**Prerequisites- Node.js 18+**

**installed- MongoDB Atlas account-  
Twilio account (for SMS)- Git###  
Quick Start**

```
bash# Clone the repositorygit c
```

**##  Environment Setup**Create a  
**.env** file in the root directory:

```
env# Server ConfigurationNODE_E
```

**##  API Documentation###**

**Interactive Documentation**Once  
the server is running, visit:-

**Swagger UI:**

<http://localhost:3001/api-docs> -

**Health Check:**

<http://localhost:3001/health> ###

## **Main API Endpoints####**

### **Authentication**

```
POST /api/auth/send-otp
```

### **#### User Management**

```
GET /api/users/doctors
```

### **#### Appointments**

```
GET /api/appointments/slots
```

### **#### File Management**

```
GET /api/files/:type/:filename
```

### **## Database Schema#### User Collection**

```
{ _id: ObjectId,  mobileNumber
```



### ### Appointment Collection

```
{ _id: ObjectId, patient: Obj
```

**## 🔑 Authentication Flow** The system uses a secure OTP-based authentication flow:

- 1. Send OTP:** Patient enters mobile number
- 2. Verify OTP:** System validates the 6-digit code
- 3. Auto Login/Register:** Existing users login, new users auto-register
- 4. JWT Tokens:** Secure access and refresh tokens issued
- 5. Role Assignment:** Users get appropriate roles and permissions


**## 👤 Role-based Access**

### ### Permission Matrix |

Endpoint	Admin	Doctor	Receptionist	Patient
GET /users/doctors	✓	✓	✓	✓
POST /admin/create-user	✓	✗	✗	✗
GET /appointments/pending	✓	✗	✗	✗
POST /appointments/confirm	✓	✗	✗	✗
POST /appointments/book	✓	✗	✓	✓
GET /appointments/my	✗	✓	✗	✓
###	###	###	###	###

**Appointment System### Key Features- Real-time Slot Management: Dynamic slot**

**generation based on doctor  
schedules- Conflict Prevention:  
Prevents double booking with  
database constraints- Status  
Tracking: Complete appointment  
lifecycle management- File  
Attachments: Support for medical  
reports and prescriptions-  
Payment Integration: Ready for  
payment gateway integration###  
Appointment States- Pending:  
Awaiting receptionist  
confirmation- Confirmed:  
Approved and scheduled- In-  
Progress: Currently ongoing-  
Completed: Successfully finished-  
Cancelled: Cancelled by**

**patient/admin- Rejected:**  
**Rejected by receptionist- No-**  
**Show: Patient didn't attend##**   
**File Upload System###**  
**Supported Features- Multiple File**  
**Types: Images (JPG, PNG) and**  
**PDFs- Size Validation: Maximum**  
**10MB per file- Secure Storage:**  
**Organized file structure with**  
**unique naming- Access Control:**  
**Role-based file access**  
**permissions### File Organization**

**uploads/|— appointments/| |—**

**##**  **Security Features###**  
**Multi-layer Security- JWT**  
**Authentication: Secure token-**

**based access- OTP Verification:  
SMS-based mobile verification-  
Rate Limiting: Prevents abuse and  
DoS attacks- Input Validation:  
Comprehensive data sanitization-  
CORS Protection: Controlled  
cross-origin requests- File Upload  
Security: Type and size  
validation## 🧪 Testing###  
Health Check**

```
bashcurl http://localhost:3001/
```

**### API Testing**

```
bash# Send OTPcurl -X POST http
```

**## 🚀 Deployment### Quick  
Deployment Commands**

```
bash# Production build  
npm run s
```

### **### Deployment Platforms-**

**Heroku: Easy deployment with git**

**integration- AWS/DigitalOcean:**

**VPS deployment with full control-**

**Docker: Containerized**

**deployment- Vercel/Netlify:**

**Serverless deployment options###**



### **Performance & Monitoring###**

**Database Optimization- Indexed**

**queries for fast lookups-**

**Aggregation pipelines for complex**

**reports- Connection pooling for**

**scalability### Monitoring Ready-**

**Comprehensive error logging-**

**Request/response logging-**

**Performance metrics collection-**

**Health check endpoints## 🤝**

**ContributingWe welcome  
contributions! Please see our  
[Contributing Guidelines](#) for  
details.### Development Setup**

```
bashgit clone cd arthomed-backen
```

**## 📝 LicenseThis project is  
licensed under the MIT License -  
see the [LICENSE](#) file for  
details.## 📞 Support-  
Documentation: Complete API  
docs at [/api-docs](#) - Issues: GitHub  
Issues for bug reports- Email:  
support@arthomed.com---🏥 Built  
with ❤️ for healthcare  
professionals and patients**

---

*Arthomed Backend - Empowering healthcare through technology*