

CalmNest Backend API

A comprehensive Node.js backend for the CalmNest mental health platform, providing REST APIs for user management, content delivery, and therapy session tracking.

Features

Authentication & User Management

- **Secure Registration/Login** with JWT tokens
- **Password encryption** using bcryptjs
- **User profiles** with preferences and progress tracking
- **Role-based access control** (user, therapist, admin)
- **Rate limiting** to prevent abuse

Content Management

- **Audio Therapy**: Music, podcasts, meditation, nature sounds, affirmations
- **Reading Therapy**: Quotes, articles, stories, poems, affirmations
- **Yoga Therapy**: Video tutorials, instructions, difficulty levels
- **File uploads** for audio, images, and videos
- **Content categorization** and search functionality

Session Tracking

- **User sessions** with mood tracking (before/after)
- **Progress analytics** for individual users
- **Admin dashboard** with platform-wide analytics
- **Therapy completion tracking**

Security & Performance

- **Helmet.js** for security headers
- **CORS** configuration
- **Request compression**
- **File upload validation**
- **Database indexing** for performance

Prerequisites

- **Node.js** (v16 or higher)

- **MongoDB** (v4.4 or higher)
- **npm** or **yarn**

Installation

1. Clone and Install Dependencies

```
bash

# Clone the repository
git clone <repository-url>
cd calmnest-backend

# Install dependencies
npm install
```

2. Environment Setup

```
bash

# Copy environment template
cp .env.example .env

# Edit the .env file with your configuration
nano .env
```

3. Required Environment Variables

```
env

# Server Configuration
PORT=5000
NODE_ENV=development

# Database
MONGODB_URI=mongodb://localhost:27017/calmnest

# JWT Configuration
JWT_SECRET=your-super-secret-jwt-key-here-make-it-long-and-random

# Frontend URL for CORS
FRONTEND_URL=http://localhost:3000
```

4. Create Upload Directories

```
bash
```

```
mkdir -p uploads/audio uploads/images uploads/videos
```

5. Start MongoDB

```
bash

# Start MongoDB service
sudo systemctl start mongod

# Or if using Docker
docker run -d -p 27017:27017 --name mongodb mongo:latest
```

6. Seed the Database

```
bash

# Run the database seeder
npm run seed
```

7. Start the Server

```
bash

# Development mode with auto-restart
npm run dev

# Production mode
npm start
```

API Endpoints

Authentication

Method	Endpoint	Description	Auth Required
POST	<code>/api/auth/register</code>	Register new user	No
POST	<code>/api/auth/login</code>	Login user	No

User Management

Method	Endpoint	Description	Auth Required
GET	<code>/api/user/profile</code>	Get user profile	Yes
PUT	<code>/api/user/profile</code>	Update user profile	Yes

Audio Content

Method	Endpoint	Description	Auth Required
GET	/api/audio	Get audio content list	No
POST	/api/audio	Upload audio content	Yes

Query Parameters for GET /api/audio:

- `category`: music, podcast, meditation, nature-sounds, affirmations
- `page`: Page number (default: 1)
- `limit`: Items per page (default: 20)
- `search`: Search term

Reading Content

Method	Endpoint	Description	Auth Required
GET	/api/reading	Get reading content list	No
POST	/api/reading	Create reading content	Yes

Query Parameters for GET /api/reading:

- `category`: quotes, articles, stories, poems, affirmations
- `page`: Page number (default: 1)
- `limit`: Items per page (default: 20)
- `search`: Search term

Yoga Content

Method	Endpoint	Description	Auth Required
GET	/api/yoga	Get yoga content list	No
POST	/api/yoga	Upload yoga content	Yes

Query Parameters for GET /api/yoga:

- `difficulty`: beginner, intermediate, advanced
- `category`: stretching, meditation, breathing, full-routine
- `page`: Page number (default: 1)
- `limit`: Items per page (default: 20)

Session Tracking

Method	Endpoint	Description	Auth Required
POST	/api/sessions	Record therapy session	Yes
GET	/api/sessions/history	Get user session history	Yes

Admin Analytics

Method	Endpoint	Description	Auth Required
GET	/api/admin/analytics	Get platform analytics	Yes (Admin)



Request/Response Examples

User Registration

```
bash
curl -X POST http://localhost:5000/api/auth/register \
-H "Content-Type: application/json" \
-d '{
  "username": "johndoe",
  "email": "john@example.com",
  "password": "securepassword123",
  "firstName": "John",
  "lastName": "Doe",
  "age": 25
}'
```

Response:

```
json
{
  "message": "User registered successfully",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "user": {
    "id": "64f5a1b2c3d4e5f6g7h8i9j0",
    "username": "johndoe",
    "email": "john@example.com",
    "firstName": "John",
    "lastName": "Doe",
    "role": "user"
  }
}
```

Audio Content Upload

bash

```
curl -X POST http://localhost:5000/api/audio \
-H "Authorization: Bearer YOUR_JWT_TOKEN" \
-F "audioFile=@path/to/audio.mp3" \
-F "title=Relaxing Music" \
-F "description=Beautiful instrumental music for relaxation" \
-F "category=music" \
-F "duration=1800" \
-F "artist=CalmNest Team"
```

Session Recording

bash

```
curl -X POST http://localhost:5000/api/sessions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer YOUR_JWT_TOKEN" \
-d '{
  "therapyType": "audio",
  "contentId": "64f5a1b2c3d4e5f6g7h8i9j0",
  "duration": 30,
  "moodBefore": 4,
  "moodAfter": 8,
  "notes": "Felt much more relaxed after the session"
}'
```



Database Schema

User Model

javascript

```
{
  username: String (unique),
  email: String (unique),
  password: String (hashed),
  firstName: String,
  lastName: String,
  age: Number,
  preferences: {
    favoriteTherapies: [String],
    musicGenres: [String],
    bookCategories: [String]
  },
  progress: {
    sessionsCompleted: Number,
    totalTimeSpent: Number,
    lastActive: Date
  },
  role: String (user/therapist/admin)
}
```

Audio Content Model

javascript

```
{
  title: String,
  description: String,
  category: String,
  audioUrl: String,
  duration: Number,
  artist: String,
  tags: [String],
  uploadedBy: ObjectId,
  playCount: Number,
  ratings: [{ userId: ObjectId, rating: Number }]
}
```

Development

Running Tests

```
bash

npm test
```

Code Structure

```
calmnest-backend/  
├── server.js      # Main server file  
├── models/        # Database models  
├── routes/        # API routes  
├── middleware/    # Custom middleware  
├── utils/         # Utility functions  
├── scripts/       # Database scripts  
├── uploads/       # File uploads  
├── tests/         # Test files  
└── docs/          # Documentation
```

Adding New Content Types

1. **Create a new schema** in the models directory
2. **Add routes** for CRUD operations
3. **Update the seeder** with sample data
4. **Add tests** for the new endpoints



Deployment

Using PM2 (Recommended)

```
bash  
  
# Install PM2 globally  
npm install -g pm2  
  
# Start the application  
pm2 start server.js --name "calmnest-backend"  
  
# Save PM2 configuration  
pm2 save  
pm2 startup
```

Using Docker

```
dockerfile
```



```
FROM node:18-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install --production
COPY . .
EXPOSE 5000
CMD ["npm", "start"]
```

```
bash
```

```
# Build and run
```

```
docker build -t calmnest-backend .
```

```
docker run -p 5000:5000 calmnest-backend
```

Security Considerations

- **JWT tokens** expire after 7 days
- **Passwords** are hashed with bcrypt (12 rounds)
- **Rate limiting** prevents abuse
- **File uploads** are validated and size-limited
- **CORS** is configured for specific origins
- **Helmet.js** adds security headers

Monitoring & Analytics

The backend includes built-in analytics for:

- **User engagement** metrics
- **Content popularity** tracking
- **Session completion** rates
- **Therapy effectiveness** (mood improvements)

Access analytics via `/api/admin/analytics` (admin role required).

Contributing

1. Fork the repository
2. Create a feature branch
3. Make your changes
4. Add tests
5. Submit a pull request

License

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

Support

For support, email calmnest665@gmail.com or create an issue in the repository.

Happy Coding!

Building technology for mental wellness and stress-free living.