News Website Project Context - Complete Reference

Project Overview

Stack: MERN (MySQL, Express.js, React, Node.js)

Database: MySQL with Sequelize ORM

Authentication: JWT + bcrypt

File Handling: Multer (images, videos, documents)

Database Schema & Models

Core Tables & Relationships

Users Table

```sql
- id (PK)
- name (STRING, required)
- email (STRING, unique, required)
- password (STRING, hashed, required)
- role (ENUM: 'user', 'reporter', 'admin', default: 'user')
- profilePicture (STRING)
- bio (TEXT)
- reporterApplicationStatus (ENUM: 'none', 'pending', 'approved', 'rejected')
- reporterApplicationDate (DATE)
- reporterApplicationFeedback (TEXT)
- articlesPublished (INTEGER, default: 0)
- articlesRejected (INTEGER, default: 0)
- lastArticleDate (DATE)
- createdAt, updatedAt (timestamps)
```

News Table

```sql
- id (PK)
- title (STRING(100), required)
- content (TEXT, required)
- summary (STRING(200), required)
- categoryId (FK → Categories.id, required)
- authorId (FK → Users.id, required)
- coverImage (STRING, required)
- featuredVideo (STRING, optional)
- hasVideo (BOOLEAN, default: false)
- videoThumbnail (STRING, optional)
- media (JSON array, default: [])
- additionalCategories (JSON array, default: [])
- isPublished (BOOLEAN, default: false)
- publishedAt (DATE)
- views (INTEGER, default: 0)
- slug (STRING, unique, required)
- tags (JSON array, default: [])
- createdAt, updatedAt (timestamps)
```

Categories Table

```sql
- id (PK)
- name (STRING, required)
- slug (STRING, unique)
- description (TEXT)
- createdAt, updatedAt
```

Comments Table

```sql
- id (PK)
- content (TEXT, required)
- newsId (FK → News.id)
- userId (FK → Users.id)
- createdAt, updatedAt
```

Pages Table (CMS)

```sql
- id (PK)
- title (STRING, required)
- content (TEXT, required)
- slug (STRING, unique)
- isPublished (BOOLEAN)
- createdById (FK → Users.id)
- lastUpdatedById (FK → Users.id)
- createdAt, updatedAt
```

Persons Table

```sql
- id (PK)
- name (STRING(100), required)
- slug (STRING, unique, required)
- description (TEXT)
- image (STRING)
- profession (STRING(100))
- createdAt, updatedAt
```

NewsPersons Table (Junction)

```sql
- newsId (FK → News.id)
- personId (FK → Persons.id)
- Primary Key: (newsId, personId)
```

Settings Table

```sql
- id (PK)
- key (STRING, unique)
- value (TEXT)
- type (STRING)
- createdAt, updatedAt
```

Database Relationships

- User → News (1:many, authorId)

- Category → News (1:many, categoryId)
- News → Comments (1:many, newsId)
- User → Comments (1:many, userId)
- User → Pages (1:many, createdById/lastUpdatedById)
- News ↔ Persons (many:many through NewsPersons)

Database Indexes

- Users: email (unique)
- News: slug (unique), categoryId, authorId, isPublished
- News: Full-text search on (title, content, summary)
- Persons: slug (unique)

API Architecture

Authentication Routes (/api/auth)

```
POST /register - Register new user
POST /login - User login
POST /apply-reporter - Apply for reporter role (protected)
GET /profile - Get user profile (protected)
PUT /profile - Update user profile (protected)
```

News Routes (/api/news)

```
Public:
GET / - Get all published news
GET /videos - Get video news only
GET /:id - Get single news by ID

Reporter (protected + reporter role):
GET /reporter/mynews - Get reporter's news
GET /reporter/stats - Get reporter statistics
POST / - Create news (with file upload)
PUT /:id - Update news (with file upload)
DELETE /:id - Delete news

Admin (protected + admin role):
GET /admin - Get all news (published + unpublished)
PUT /:id/publish - Toggle publish status
POST /import - Import single news
POST /import/parse - Bulk import from file
```

Other Routes

```
/api/users - User management
/api/categories - Category management
/api/pages - CMS page management
/api/people - Person/personality management
/api/settings - Site settings
/api/upload - File upload utilities
/api/admin - Admin-specific operations
```

File Upload Configuration

Upload Directories

```
uploads/
├── images/ - Cover images, general images
├── videos/ - Featured videos, video content
├── thumbnails/ - Video thumbnails
└── temp/ - Temporary file storage
```

Multer Configuration

- **Images**: JPG, JPEG, PNG, GIF (10MB limit)

- **Videos**: MP4, WEBM, MOV, AVI, WMV (50MB limit)
- **Documents**: 10MB limit for imports
- **Multiple fields**: coverImage, video, videoThumbnail

Authentication & Authorization

Middleware Stack

```javascript
protect - Validates JWT token
reporter - Requires 'reporter' or 'admin' role
admin - Requires 'admin' role only
```

User Roles & Permissions

- **user**: Basic access, can comment, view content
- **reporter**: Can create/edit/delete own news, view stats
- **admin**: Full access, can publish/unpublish, manage all content

Reporter Application Workflow

1. User applies via `/api/auth/apply-reporter`
2. Status: 'pending' → Admin reviews
3. Admin approves/rejects → Status: 'approved'/'rejected'
4. Approved users get 'reporter' role

Frontend Structure (React Router)

Public Routes

- `/` - HomePage
- `/login` - LoginPage
- `/register` - RegisterPage
- `/news/:id` - NewsDetailPage
- `/videos` - VideoNewsPage
- `/people` - PeopleListPage
- `/people/:slug` - PersonNewsPage

Protected Routes

- `/profile` - ProfilePage

Admin Routes (/admin)

- `/admin` - AdminDashboard
- `/admin/news` - NewsManagement
- `/admin/news/create` - NewsEditor
- `/admin/news/edit/:id` - NewsEditor
- `/admin/reporter-applications` - ReporterApplications
- `/admin/reporters` - ReporterManagement
- `/admin/categories` - CategoryManagement
- `/admin/pages` - PageManagement
- `/admin/users` - UserManagement
- `/admin/people` - PeopleManagement
- `/admin/people/create` - PersonEditor
- `/admin/people/edit/:id` - PersonEditor
- `/admin/settings` - SiteSettings

Reporter Routes (/reporter)

- `/reporter` - ReporterDashboard
- `/reporter/create` - CreateNews
- `/reporter/edit/:id` - EditNews
- `/reporter/stats` - ReporterStats

Key Features & Functionality

News Management

- Rich content with title, summary, full content
- Cover images and featured videos
- Category assignment + additional categories
- Tag system (JSON array)
- Person/personality tagging (many-to-many)

- Draft/publish workflow
- View tracking
- SEO-friendly slugs

Media Handling

- Image uploads for covers and content
- Video uploads with thumbnail generation
- Multiple file type support
- Organized storage structure

User Management

- Role-based access control
- Reporter application system
- Performance tracking (articles published/rejected)
- Profile management with bio and images

Search & Discovery

- Full-text search on news content
- Category-based filtering
- Person-based news filtering
- Video-specific content section

Technical Implementation Notes

Database Connection

- Sequelize ORM with MySQL
- Connection pooling (max: 5 connections)
- Auto-sync with `alter: true` for schema updates

Security Features

- bcrypt password hashing (salt rounds: 10)
- JWT token authentication
- Role-based route protection
- File type validation for uploads

Performance Optimizations

- Database indexes on frequently queried fields
- Full-text search indexes
- Connection pooling
- Static file serving for uploads

Environment Variables Required

```
DB_NAME=breaking_news
DB_USER=root
DB_PASSWORD=root
DB_HOST=localhost
PORT=5000
JWT_SECRET=your_jwt_secret
```

Development Patterns

Error Handling

- Try-catch blocks in controllers
- Centralized error middleware
- Validation at model level (Sequelize validators)

File Upload Pattern

- Multer middleware for route-specific uploads
- Dynamic destination based on file type
- Unique filename generation with timestamps

Authentication Pattern

- JWT tokens for stateless authentication
- Middleware chaining for role-based access
- Password hashing hooks in User model

This document serves as the complete reference for the news website project. Use this context when asking for specific help or when continuing development across multiple chat sessions.