

# AVGC Tender Management Platform

A comprehensive web-based tender management system designed for the Animation, Visual Effects, Gaming, and Comics (AVGC) industry. This platform streamlines the entire tender lifecycle from discovery to award, with integrated EMD/DD management, security deposits, and advanced analytics.

## Features

### Core Features

- **Tender Management:** Complete tender lifecycle management with multi-stage workflows
- **Bid Management:** Comprehensive bid submission and evaluation system
- **EMD/DD Management:** Automated calculation and tracking of Earnest Money Deposits
- **Security Management:** Bank guarantee and security deposit management
- **Document Management:** Secure document upload, version control, and digital signatures
- **Real-time Notifications:** WebSocket-based real-time updates and notifications
- **Advanced Analytics:** Comprehensive dashboards and reporting capabilities
- **Multi-tenant Support:** Firm-based access control and data isolation

### Security Features

- JWT-based authentication with refresh tokens
- Role-based access control (RBAC)
- API rate limiting and throttling
- Data encryption at rest and in transit
- Audit logging and compliance tracking
- Two-factor authentication (optional)

## Technology Stack

### Frontend

- **Framework:** React 18 with TypeScript
- **State Management:** Redux Toolkit & React Query
- **UI Library:** Material-UI (MUI) v5
- **Routing:** React Router v6
- **Forms:** React Hook Form with Yup validation

- **Charts:** Recharts & Chart.js
- **Real-time:** Socket.io Client
- **Build Tool:** Vite

## Backend

- **Framework:** NestJS with TypeScript
- **Database:** PostgreSQL (primary), MongoDB (documents), Redis (cache)
- **ORM:** TypeORM
- **Authentication:** Passport.js with JWT
- **Message Queue:** Bull (Redis-based)
- **Search:** Elasticsearch
- **File Storage:** AWS S3
- **API Documentation:** Swagger/OpenAPI
- **WebSockets:** Socket.io

## DevOps

- **Containerization:** Docker & Docker Compose
- **Orchestration:** Kubernetes
- **CI/CD:** GitHub Actions
- **Monitoring:** Prometheus & Grafana
- **Logging:** ELK Stack (Elasticsearch, Logstash, Kibana)



## Prerequisites

- Node.js  $\geq 18.0.0$
- npm  $\geq 9.0.0$
- PostgreSQL  $\geq 14$
- Redis  $\geq 7.0$
- MongoDB  $\geq 6.0$
- Elasticsearch  $\geq 8.0$  (optional)
- Docker & Docker Compose (for containerized deployment)



## Quick Start

## 1. Clone the repository

```
bash

git clone https://github.com/avgc/tender-management-platform.git
cd tender-management-platform
```

## 2. Install dependencies

```
bash

# Install backend dependencies
cd backend
npm install

# Install frontend dependencies
cd ../frontend
npm install
```

## 3. Environment Setup

```
bash

# Copy environment files
cp backend/.env.example backend/.env
cp frontend/.env.example frontend/.env

# Update the .env files with your configuration
```

## 4. Database Setup

```
bash

cd backend

# Run database migrations
npm run migration:run

# Seed initial data (optional)
npm run seed
```

## 5. Start Development Servers

```
bash
```

```
# Start backend (from backend directory)
```

```
npm run start:dev
```

```
# Start frontend (from frontend directory)
```

```
npm run dev
```

## 6. Access the Application

- Frontend: <http://localhost:5173>
- Backend API: <http://localhost:3000/api/v1>
- API Documentation: <http://localhost:3000/api-docs>



## Docker Deployment

### Using Docker Compose

```
bash
```

```
# Build and start all services
```

```
docker-compose up -d
```

```
# View logs
```

```
docker-compose logs -f
```

```
# Stop all services
```

```
docker-compose down
```

## Production Deployment

```
bash
```

```
# Build production images
```

```
docker build -t avgc-frontend:latest ./frontend
```

```
docker build -t avgc-backend:latest ./backend
```

```
# Deploy to Kubernetes
```

```
kubectl apply -f k8s/
```



## Project Structure

```
avgc-tender-platform/
├── frontend/           # React frontend application
│   ├── src/
│   │   ├── components/ # Reusable components
│   │   ├── modules/    # Feature modules
│   │   ├── services/   # API services
│   │   ├── hooks/      # Custom React hooks
│   │   ├── store/      # Redux store
│   │   ├── utils/      # Utility functions
│   │   └── types/       # TypeScript types
│   └── public/         # Static assets
├── backend/           # NestJS backend application
│   ├── src/
│   │   ├── modules/    # Feature modules
│   │   ├── common/     # Common utilities
│   │   ├── config/     # Configuration
│   │   ├── database/   # Database entities & migrations
│   │   └── main.ts      # Application entry point
│   └── test/           # Test files
├── docker-compose.yml  # Docker Compose configuration
├── k8s/                # Kubernetes manifests
└── docs/               # Documentation
```

## Testing

### Backend Tests

```
bash

cd backend

# Unit tests
npm run test

# E2E tests
npm run test:e2e

# Test coverage
npm run test:cov
```

### Frontend Tests

```
bash
```

```
cd frontend
```

```
# Run tests
```

```
npm run test
```

```
# Run tests with UI
```

```
npm run test:ui
```

```
# Test coverage
```

```
npm run test:coverage
```

## API Documentation

The API documentation is automatically generated using Swagger and is available at:

- Development: <http://localhost:3000/api-docs>
- Production: <https://api.avgtender.com/api-docs>

## Default Credentials

For development environment:

- **Admin User:** [admin@avgtender.com](mailto:admin@avgtender.com) / Admin@123
- **Manager User:** [manager@avgtender.com](mailto:manager@avgtender.com) / Manager@123
- **Analyst User:** [analyst@avgtender.com](mailto:analyst@avgtender.com) / Analyst@123

**Note:** Change these credentials immediately in production!

## Contributing

1. Fork the repository
2. Create your 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

## License

This project is licensed under the MIT License - see the [LICENSE](#) file for details.

## Support

For support and queries:

- Email: [support@avgctender.com](mailto:support@avgctender.com)
- Documentation: <https://docs.avgctender.com>
- Issues: <https://github.com/avgc/tender-management-platform/issues>



## Acknowledgments

- AVGC Task Force for requirements and guidance
- All contributors who have helped shape this platform
- Open source community for the amazing tools and libraries

---

**Note:** This is a reference implementation. Customize it according to your specific requirements and security policies before deploying to production.