**🏟️ SportAI Suite Enterprise Edition**

Enterprise-grade AI-powered sports facility management platform designed for modern sports complexes, recreation centers, and athletic facilities.

**🎯 Features**

**Core Capabilities**

* 🤖 **AI-Powered Optimization** - Smart scheduling, demand forecasting, and revenue maximization
* 📊 **Comprehensive Analytics** - Real-time dashboards and detailed reporting
* 💰 **Revenue Management** - Dynamic pricing, sponsorship matching, and financial tracking
* 👥 **Member Management** - CRM, loyalty programs, and engagement tracking
* 🏟️ **Facility Management** - Multi-facility support, resource allocation, and maintenance scheduling
* 📅 **Event Management** - Tournament scheduling, league coordination, and program management
* 🔐 **Enterprise Security** - PBKDF2 password hashing, session management, and audit logging
* 🎨 **White-Label Support** - Custom branding for each facility
* 📱 **Mobile Responsive** - Works seamlessly on all devices

**Subscription Tiers**

| **Feature** | **Starter** | **Professional** | **Enterprise** |
| --- | --- | --- | --- |
| Price | $99/month | $299/month | Custom |
| Users | 5 | 25 | Unlimited |
| Facilities | 1 | 3 | Unlimited |
| AI Modules | ❌ | ✅ | ✅ |
| API Access | ❌ | ✅ | ✅ |
| White Label | ❌ | ✅ | ✅ |
| Support | Email | Priority | Dedicated |

**🚀 Quick Start**

**Prerequisites**

* Python 3.8 or higher
* pip package manager
* 2GB RAM minimum
* 10GB storage space

**Installation**

1. **Clone the repository**

git clone https://github.com/yourusername/sportai-suite.git

cd sportai-suite

1. **Create virtual environment**

python -m venv venv

source venv/bin/activate # On Windows: venv\Scripts\activate

1. **Install dependencies**

pip install -r requirements.txt

1. **Set up environment variables**

cp .env.example .env

# Edit .env with your configuration

1. **Initialize the database**

python scripts/initialize\_db.py

1. **Run the application**

streamlit run sportai\_main\_app\_file.py

1. **Access the application**

* Open browser to http://localhost:8501
* Check database/[facility\_id]\_setup.txt for initial admin credentials
* Login and change password immediately

**📁 Project Structure**

sportai-suite/

├── sportai\_main\_app\_file.py # Main application file

├── requirements.txt # Python dependencies

├── .env.example # Environment variables template

├── .gitignore # Git ignore file

├── LICENSE # MIT License

├── README.md # This file

├── docker-compose.yml # Docker configuration

├── Dockerfile # Docker container setup

│

├── ai\_modules/ # AI-powered modules

│ ├── \_\_init\_\_.py

│ ├── demand\_forecasting.py

│ ├── scheduling\_optimizer.py

│ ├── sponsorship\_matcher.py

│ ├── membership\_churn.py

│ └── marketing\_optimizer.py

│

├── modules/ # Feature modules

│ ├── facility\_management/

│ ├── membership\_management/

│ ├── event\_management/

│ ├── financial\_management/

│ └── reporting/

│

├── database/ # Database files (gitignored)

│ └── .gitkeep

│

├── logs/ # Application logs (gitignored)

│ └── .gitkeep

│

├── audit\_logs/ # Security audit logs (gitignored)

│ └── .gitkeep

│

├── configurations/ # Facility configurations (gitignored)

│ └── .gitkeep

│

├── scripts/ # Utility scripts

│ ├── initialize\_db.py

│ ├── backup.py

│ ├── migrate.py

│ └── health\_check.py

│

├── tests/ # Test suite

│ ├── \_\_init\_\_.py

│ ├── test\_security.py

│ ├── test\_modules.py

│ └── test\_api.py

│

└── docs/ # Documentation

├── API.md

├── DEPLOYMENT.md

├── SECURITY.md

└── USER\_GUIDE.md

**🔧 Configuration**

**Environment Variables**

Create a .env file with the following variables:

# Application Settings

APP\_NAME=SportAI Suite

APP\_ENV=production

SECRET\_KEY=your-secret-key-here

# Database

DATABASE\_URL=sqlite:///database/sportai.db

# Security

SESSION\_TIMEOUT=3600

MAX\_LOGIN\_ATTEMPTS=5

PASSWORD\_MIN\_LENGTH=8

# Email (Optional)

SMTP\_HOST=smtp.gmail.com

SMTP\_PORT=587

SMTP\_USER=your-email@gmail.com

SMTP\_PASSWORD=your-password

# License Server (Optional)

LICENSE\_SERVER\_URL=https://license.sportai.com

LICENSE\_KEY=your-license-key

# Features

ENABLE\_AI\_MODULES=true

ENABLE\_MULTI\_FACILITY=false

ENABLE\_API\_ACCESS=true

**First-Time Setup**

1. **Change default admin password**
2. **Configure facility information** (Settings → Configuration)
3. **Set up branding** (colors, logo, facility name)
4. **Add users** with appropriate roles
5. **Configure integrations** (payment, email, SMS)
6. **Set up backup schedule**

**🐳 Docker Deployment**

**Using Docker Compose**

# Build and run

docker-compose up -d

# View logs

docker-compose logs -f

# Stop

docker-compose down

**Manual Docker Build**

# Build image

docker build -t sportai-suite .

# Run container

docker run -d -p 8501:8501 \

-v $(pwd)/database:/app/database \

-v $(pwd)/logs:/app/logs \

--name sportai \

sportai-suite

**🔐 Security Features**

* **Password Security**: PBKDF2-HMAC SHA256 with salt and pepper
* **Session Management**: Secure tokens with configurable timeout
* **Account Protection**: Lockout after failed attempts
* **Audit Logging**: Complete activity tracking
* **Data Encryption**: At-rest and in-transit encryption
* **Role-Based Access**: Granular permission system
* **API Security**: Unique keys per user with rate limiting

**📊 API Documentation**

**Authentication**

import requests

# Login and get token

response = requests.post('http://localhost:8501/api/auth/login', json={

'email': 'user@example.com',

'password': 'secure\_password'

})

token = response.json()['token']

# Use token for API calls

headers = {'Authorization': f'Bearer {token}'}

**Example Endpoints**

# Get facility metrics

GET /api/facilities/metrics

# Create event

POST /api/events

{

"name": "Basketball Tournament",

"date": "2025-02-15",

"facility\_id": "court-1"

}

# Get member list

GET /api/members?limit=50&offset=0

Full API documentation available at /api/docs when running.

**🧪 Testing**

# Run all tests

pytest

# Run with coverage

pytest --cov=.

# Run specific test file

pytest tests/test\_security.py

# Run with verbose output

pytest -v

**📈 Performance Optimization**

* **Lazy Loading**: Modules loaded on-demand
* **Caching**: Redis support for session and data caching
* **Database Indexing**: Optimized queries
* **Async Operations**: Background task processing
* **CDN Support**: Static asset delivery

**🤝 Contributing**

We welcome contributions! Please see our [Contributing Guidelines](https://claude.ai/chat/CONTRIBUTING.md).

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](https://claude.ai/chat/LICENSE) file for details.

**🆘 Support**

* **Documentation**: [docs.sportai.com](https://docs.sportai.com/)
* **Email**: support@sportai.com
* **Issues**: [GitHub Issues](https://github.com/yourusername/sportai-suite/issues)
* **Discord**: [Join our community](https://discord.gg/sportai)

**🏆 Testimonials**

"SportAI Suite transformed our facility management. We've seen a 35% increase in utilization and 25% revenue growth in just 6 months." - *John Smith, Recreation Director*

"The AI scheduling alone saved us 20 hours per week. It's been a game-changer." - *Sarah Johnson, Facility Manager*

**🗺️ Roadmap**

* [x] Core facility management
* [x] AI-powered scheduling
* [x] Multi-tenant support
* [x] White-label capabilities
* [ ] Mobile app (Q2 2025)
* [ ] Advanced analytics dashboard (Q2 2025)
* [ ] IoT sensor integration (Q3 2025)
* [ ] Blockchain ticketing (Q4 2025)

**📊 Stats**

**Built with ❤️ by the SportAI Team**

*Empowering sports facilities with intelligent management solutions*