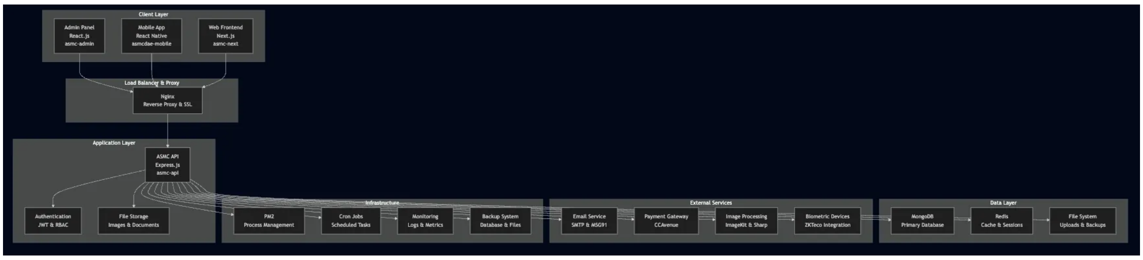
ASMC System - Complete Deployment Overview

This document provides a comprehensive overview of deploying the entire ASMC (Anushaktinagar Sports Management Committee) system, including all four main components and their infrastructure requirements.

Table of Contents

System Architecture

Complete System Overview



Client Layer
Admin Panel React.js asmc-admin
Mobile App React Native asmcdae-mobile
Web Frontend Next.js asmc-next
Load Balancer & Proxy
Nginx Reverse Proxy & SSL
Application Layer
ASMC API Express.js asmc-api
Authentication JWT & RBAC
File Storage Images & Documents
Infrastructure
PM2 Process Management
Cron Jobs Scheduled Tasks
Monitoring Logs & Metrics
Backup System Database & Files
External Services
Email Service SMTP & MSG91
Payment Gateway CCAvenue
Image Processing ImageKit & Sharp
Biometric Devices ZKTeco Integration
Data Layer
MongoDB Primary Database
Redis Cache & Sessions
File System Uploads & Backups

Component Overview

1. ASMC API (Backend)

- **Technology**: Node.js + Express.js
- **Database**: MongoDB with Mongoose
- **Authentication**: JWT with RBAC
- **Port**: 7055
- **Features**:
  - Member management
  - Payment processing
  - Booking system
  - Biometric integration
  - Report generation

2. ASMC Admin (Admin Panel)

- **Technology**: React.js
- **Build**: Static files
- **Port**: 3000 (development) / 80 (production)
- **Features**:
  - Member management interface
  - Payment processing

    - Booking management
    - Report generation
    - System administration

3. ASMCDAE Mobile (Mobile App)

- **Technology**: React Native
- **Platform**: Android & iOS
- **Features**:
    - Member login and profile
    - Booking management
    - Payment processing
    - Biometric attendance
    - Notifications

4. ASMC Next (Web Frontend)

- **Technology**: Next.js + React
- **Port**: 3000 (development) / 80 (production)
- **Features**:
    - Public website
    - Member portal
    - Booking interface
    - Payment integration

Infrastructure Requirements

Server Specifications

Minimum Requirements

- **CPU**: 2 cores, 2.0 GHz
- **RAM**: 4GB
- **Storage**: 50GB SSD
- **Network**: 1 Gbps
- **OS**: Ubuntu 20.04 LTS

Recommended Requirements

- **CPU**: 4 cores, 2.5 GHz
- **RAM**: 8GB
- **Storage**: 100GB SSD
- **Network**: 1 Gbps
- **OS**: Ubuntu 22.04 LTS

Production Requirements

- **CPU**: 8 cores, 3.0 GHz
- **RAM**: 16GB
- **Storage**: 200GB SSD
- **Network**: 10 Gbps
- **OS**: Ubuntu 22.04 LTS

Software Requirements

Core Software

- **Node.js**: 18.0.0+

- **MongoDB**: 6.0.0+
- **Nginx**: 1.18.0+
- **PM2**: Latest
- **Git**: Latest

Optional Software

- **Redis**: 6.0.0+ (for caching)
- **Docker**: Latest (for containerization)
- **Certbot**: Latest (for SSL)

Deployment Strategy

Deployment Environments

Development Environment

```
# Local development setup
- API: http://localhost:7055
- Admin: http://localhost:3000
- Next.js: http://localhost:3001
- Database: mongodb://localhost:27017/asmc_dev
```

Staging Environment

```
# Staging server setup
- API: https://staging-api.asmcdae.in
- Admin: https://staging-admin.asmcdae.in
- Next.js: https://staging.asmcdae.in
- Database: mongodb://staging-server:27017/asmc_staging
```

Production Environment

```
# Production server setup
- API: https://api.asmcdae.in
- Admin: https://admin.asmcdae.in
- Next.js: https://asmcdae.in
- Database: mongodb://prod-server:27017/asmc_prod
```

Deployment Order

1. **Infrastructure Setup**
    - Server preparation
    - Software installation
    - Security configuration
2. **Database Setup**
    - MongoDB installation
    - Database creation
    - Index creation
    - Backup configuration
3. **API Deployment**

    - Application deployment
    - Environment configuration
    - PM2 setup
    - Health checks

4. **Frontend Deployment**

    - Admin panel deployment
    - Next.js deployment
    - Static file serving
    - CDN configuration

5. **Mobile App Deployment**

    - Build generation
    - App store deployment
    - Push notification setup

6. **Integration & Testing**

    - Component integration
    - End-to-end testing
    - Performance testing
    - Security testing

Environment Configuration

Environment Variables Structure

```
asmc-system/
├── asmc-api/
│   ├── .env.development
│   ├── .env.staging
│   └── .env.production
├── asmc-admin/
│   ├── .env.development
│   ├── .env.staging
│   └── .env.production
└── asmc-next/
    ├── .env.development
    ├── .env.staging
    └── .env.production
```

Shared Configuration

Database Configuration

```
# MongoDB
MONGO_URI=mongodb://localhost:27017/asmc
MONGO_TEST_URI=mongodb://localhost:27017/asmc_test

# Redis (Optional)
REDIS_URL=redis://localhost:6379
REDIS_PASSWORD=your-redis-password
```

Authentication Configuration

```
# JWT
JWT_SECRET=your-super-secret-jwt-key-minimum-64-characters
JWT_EXPIRE=7d
JWT_ISSUER=asmc-system
JWT_AUDIENCE=asmc-clients
```

External Services

```
# Email
SMTP_HOST=smtp.gmail.com
SMTP_PORT=587
SMTP_USER=your-email@gmail.com
SMTP_PASS=your-app-password
MSG91_AUTH_KEY=your-msg91-auth-key

# Payment Gateway
CCAVENUE_MERCHANT_ID=your-merchant-id
CCAVENUE_ACCESS_CODE=your-access-code
CCAVENUE_WORKING_KEY=your-working-key

# Image Processing
IMAGEKIT_PUBLIC_KEY=your-imagekit-public-key
IMAGEKIT_PRIVATE_KEY=your-imagekit-private-key
IMAGEKIT_URL_ENDPOINT=https://ik.imagekit.io/your-imagekit-id

# Biometric
BIOMETRIC_IP=192.168.1.100
BIOMETRIC_PORT=4370
```

Security Considerations

Network Security

Firewall Configuration

```
# UFW Firewall Rules
sudo ufw allow 22/tcp    # SSH
sudo ufw allow 80/tcp    # HTTP
sudo ufw allow 443/tcp   # HTTPS
sudo ufw allow 27017/tcp # MongoDB (internal only)
sudo ufw enable
```

SSL/TLS Configuration

```
# SSL Certificate Setup
sudo certbot --nginx -d api.asmcdae.in
sudo certbot --nginx -d admin.asmcdae.in
sudo certbot --nginx -d asmcdae.in
```

Application Security

API Security

- JWT authentication
- Rate limiting
- Input validation
- CORS configuration
- Security headers

Database Security

- Authentication enabled
- Network access restricted
- Regular backups
- Encryption at rest

File Security

- Upload validation
- File type restrictions
- Size limitations
- Virus scanning

Monitoring & Maintenance

Monitoring Stack

Application Monitoring

```
# PM2 Monitoring
pm2 monit
pm2 logs

# System Monitoring
htop
iotop
nethogs
```

Database Monitoring

```
# MongoDB Monitoring
mongo --eval "db.stats()"
mongo --eval "db.serverStatus()"
```

Log Management

```
# Log Rotation
sudo nano /etc/logrotate.d/asmc-system

# Log Aggregation
# Consider ELK Stack or similar
```

Backup Strategy

Database Backups

```
# Daily automated backups
0 2 * * * /opt/asmc-api/scripts/backup-db.sh

# Weekly full backups
0 2 * * 0 /opt/asmc-api/scripts/full-backup.sh
```

File Backups

```
# Application files backup
0 3 * * * /opt/asmc-system/scripts/backup-files.sh

# Configuration backup
0 4 * * * /opt/asmc-system/scripts/backup-config.sh
```

Maintenance Schedule

Daily Tasks

- Health checks
- Log monitoring
- Performance monitoring
- Backup verification

Weekly Tasks

- Security updates
- Performance optimization
- Database maintenance
- Log rotation

Monthly Tasks

- Full system backup
- Security audit
- Performance review
- Capacity planning

Quick Deployment Commands

Complete System Deployment

```
# 1. Server Setup
sudo apt update && sudo apt upgrade -y
sudo apt install -y curl wget git nginx mongodb-org nodejs

# 2. Clone All Repositories
git clone <asmc-api-url> /opt/asmc-api
git clone <asmc-admin-url> /opt/asmc-admin
git clone <asmc-next-url> /opt/asmc-next

# 3. Deploy API
cd /opt/asmc-api
npm ci --production
npm run start:prod
```

```bash
# 4. Deploy Admin Panel
cd /opt/asmc-admin
npm ci --production
npm run build
sudo cp -r build/* /var/www/admin/

# 5. Deploy Next.js
cd /opt/asmc-next
npm ci --production
npm run build
npm run start

# 6. Configure Nginx
sudo nano /etc/nginx/sites-available/asmc-system
sudo ln -s /etc/nginx/sites-available/asmc-system /etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl restart nginx

# 7. Setup SSL
sudo certbot --nginx -d api.asmcdae.in -d admin.asmcdae.in -d asmcdae.in

# 8. Setup Monitoring
pm2 install pm2-logrotate
pm2 save
pm2 startup
```

Component-Specific Documentation

- **ASMC API**: Backend Documentation
- **ASMC Admin**: Admin Panel Documentation
- **ASMCDAE Mobile**: Mobile App Documentation
- **ASMC Next**: Frontend Documentation

Troubleshooting

Common Issues

1. Component Communication Issues

```bash
# Check API health
curl https://api.asmcdae.in/health

# Check CORS configuration
# Verify environment variables
```

2. Database Connection Issues

```bash
# Check MongoDB status
sudo systemctl status mongod

# Check connection strings
# Verify authentication
```

3. SSL Certificate Issues

```
# Check certificate status
sudo certbot certificates

# Renew certificates
sudo certbot renew
```

Support Resources

- **Documentation**: Check component-specific documentation
- **Logs**: Review application and system logs
- **Monitoring**: Use PM2 and system monitoring tools
- **Health Checks**: Implement comprehensive health checks

Support & Maintenance

For deployment support and maintenance:

1. **Documentation**: Refer to component-specific guides
2. **Monitoring**: Use built-in monitoring tools
3. **Logs**: Check application and system logs
4. **Health Checks**: Monitor system health endpoints
5. **Backups**: Verify backup integrity regularly

System Deployment Complete!

Your ASMC system is now ready for production use with all components properly configured and integrated.