AI Diagnostics System - User Interface Documentation

Overview

The AI Diagnostics System provides a comprehensive web interface for monitoring, diagnosing, and maintaining the Sports Bar TV Controller application. It features an intelligent chatbot, real-time health monitoring, and interactive dashboards.

Access

Navigate to: **http://192.168.1.25:3000/diagnostics**

Or click **“System Diagnostics”** in the navigation menu.

Features

1. AI Diagnostics Chatbot

An intelligent conversational assistant that can:

- **Answer health questions**: “Is everything running OK?”
- **Explain issues**: “What problems have been detected?”
- **Show recent fixes**: “What fixes were applied recently?”
- **Provide recommendations**: “How can I improve system performance?”
- **Explain monitoring**: “Tell me about the diagnostics system”
- **Show patterns**: “What patterns has the system learned?”

How to use:
1. Navigate to the “AI Assistant” tab
2. Type your question in the input box
3. Press Enter or click Send
4. The AI will respond with relevant information from the system

Suggested questions:
- Is everything running OK?
- What issues have been detected recently?
- Show me the latest fixes applied
- What patterns has the system learned?
- How can I improve system performance?
- Explain the monitoring system

2. System Health Dashboard

Overview Tab

System Status Cards:
- **Overall Health**: Green (healthy), Yellow (warning), or Red (critical)
- **Active Issues**: Count of current problems

- **Recent Fixes**: Fixes applied in last 24 hours
- **Uptime**: System availability percentage (7 days)

Manual Controls:
- **Light Check**: Run quick health check now (5-minute check)
- **Deep Diagnostics**: Run comprehensive analysis (Sunday check)
- **Self-Heal**: Trigger automatic issue resolution

Component Status:
Real-time status of all monitored components:
- PM2 processes
- API endpoints
- Database
- System resources (CPU, memory, disk)
- External integrations

Charts & Metrics

Health Metrics Chart (24h):
- CPU usage trends
- Memory usage trends
- Disk usage trends
- Average and maximum values

Issue Frequency Chart:
- Bar chart showing issues by type
- Color-coded by severity
- 7-day historical data

Fix Success Rate:
- Pie chart showing successful vs failed fixes
- Success percentage
- Average fix duration
- Total fixes applied

Uptime Gauge:
- Visual gauge showing system uptime
- 7-day uptime percentage
- Downtime calculation
- Status indicator (Excellent/Good/Needs Attention)

3. Details Tab

Recent Health Checks:
- Last 10 health checks performed
- Component checked
- Status (healthy/warning/critical)
- Timestamp
- Response time

Active Issues:
- Current open issues
- Severity level
- Description

- Component affected
- Auto-fix status

Recent Fixes Applied:
- Last 10 fixes
- Action taken
- Success/failure status
- Timestamp
- Duration

Learning Patterns:
- Identified patterns
- Frequency (hourly/daily/weekly)
- Number of occurrences
- AI recommendations

Understanding the System

Health Status Indicators

Healthy (Green):
- All systems operational
- No critical issues
- Resources within normal range

Warning (Yellow):
- Minor issues detected
- Resources approaching limits
- Non-critical problems

Critical (Red):
- Major issues detected
- System functionality affected
- Immediate attention required

Issue Severity Levels

Low (Blue):
- Minor issues
- No immediate impact
- Can be addressed during maintenance

Medium (Yellow):
- Noticeable issues
- Should be addressed soon
- May affect some functionality

High (Orange):
- Significant problems
- Affecting functionality
- Requires prompt attention

Critical (Red):
- System-breaking issues

- Immediate action required
- May cause downtime

Monitoring Schedule

Light Checks (Every 5 minutes):
- PM2 process status
- API health
- Database connectivity
- System resources
- Quick response time checks

Deep Diagnostics (Sunday 5:00 AM):
- Full dependency audit
- Security vulnerability scan
- Performance analysis (7-day trends)
- Log file analysis
- Database integrity check
- External integration testing
- Configuration validation
- Optimization recommendations

Self-Healing (Automatic):
- Triggered when issues detected
- Attempts automatic fixes
- Logs all actions
- Reports success/failure

Self-Healing Actions

The system can automatically perform:

1. **restart_pm2**: Restart Node.js processes
2. **clear_disk**: Clean temporary files and logs
3. **reinstall_deps**: Reinstall npm dependencies
4. **repair_db**: Run database integrity checks
5. **optimize_db**: Vacuum and optimize database
6. **clear_cache**: Clear application caches

API Endpoints

The diagnostics system exposes these endpoints:

- `GET /api/diagnostics/status` - Get current system status
- `POST /api/diagnostics/light-check` - Run light health check
- `POST /api/diagnostics/deep` - Run deep diagnostics
- `POST /api/diagnostics/self-heal` - Trigger self-healing
- `GET /api/diagnostics/metrics?hours=24` - Get metrics
- `GET /api/diagnostics/issue-stats` - Get issue statistics
- `GET /api/diagnostics/fix-stats` - Get fix statistics
- `POST /api/chat/diagnostics` - AI chatbot endpoint

Database Schema

The system uses these Prisma models:

- **SystemHealthCheck**: Individual health check results
- **Issue**: Detected problems and their status
- **Fix**: Applied fixes and their outcomes
- **SystemMetric**: Historical metrics for trends
- **LearningPattern**: Identified patterns and predictions
- **DiagnosticRun**: Summary of diagnostic executions

Troubleshooting

Dashboard not loading

1. Check if the server is running: `pm2 status`
2. Check database connectivity
3. Review browser console for errors

Chatbot not responding

1. Verify OpenAI API key is configured
2. Check `/api/chat/diagnostics` endpoint
3. Review server logs for errors

Charts showing no data

1. Run a light check to generate data
2. Wait for scheduled checks to run
3. Check database for records

Manual checks failing

1. Review error message in UI
2. Check server logs
3. Verify database connectivity
4. Ensure PM2 is running

Best Practices

1. **Regular Monitoring**: Check dashboard daily
2. **Review Patterns**: Look at learning patterns weekly
3. **Manual Checks**: Run deep diagnostics before major events
4. **Issue Resolution**: Address high/critical issues promptly
5. **Uptime Goals**: Maintain >99% uptime
6. **Resource Management**: Keep CPU/memory/disk below 80%

Integration with AI Hub

The diagnostics system integrates with the AI Hub:

1. Navigate to AI Hub → AI Assistant tab

2. The codebase index includes diagnostics code
3. Ask AI Assistant about diagnostics implementation
4. Use enhanced chat for troubleshooting help

Support

For issues or questions:
1. Use the AI chatbot for immediate help
2. Review the diagnostics-system.md documentation
3. Check server logs: `/var/log/pm2/`
4. Review GitHub issues

Future Enhancements

Planned features:
- Email/SMS alerts for critical issues
- Predictive maintenance recommendations
- Performance optimization suggestions
- Custom alert thresholds
- Historical trend analysis
- Export reports (PDF/CSV)
- Integration with external monitoring tools