Table of Contents

# The ONYX Bible

## Comprehensive Guide to the Building Assessment & Capital Planning Platform

## Table of Contents

1. [Executive Summary](#executive-summary)
2. [Introduction](#introduction)
3. [Core Concepts](#core-concepts)
4. [Platform Architecture](#platform-architecture)
5. [Features Overview](#features-overview)
6. [User Roles & Permissions](#user-roles--permissions)
7. [Complete Feature Guide](#complete-feature-guide)
8. [Technical Documentation](#technical-documentation)
9. [API Reference](#api-reference)
10. [Deployment Guide](#deployment-guide)
11. [Security & Compliance](#security--compliance)
12. [Troubleshooting](#troubleshooting)
13. [Future Roadmap](#future-roadmap)
14. [Glossary](#glossary)

## Executive Summary

ONYX is a comprehensive multi-tenant SaaS platform designed for building condition assessments and lifecycle reporting for capital planning. It enables organizations to systematically evaluate their facilities, track maintenance needs, and make data-driven decisions about capital investments.

### Key Value Propositions

* **Streamlined Assessments**: Two-phase assessment workflow ensures thorough evaluation
* **Data-Driven Insights**: FCI calculations and analytics drive informed decisions
* **Multi-Tenant Architecture**: Secure, isolated environments for each organization
* **Real-Time Collaboration**: Team-based assessments with role-based access
* **Comprehensive Reporting**: Detailed reports for stakeholders and decision-makers

## Introduction

### What is ONYX?

ONYX is a modern web application that digitizes and streamlines the facility condition assessment process. It replaces paper-based inspections and disconnected spreadsheets with a unified platform that captures, analyzes, and reports on building conditions.

### Who Uses ONYX?

* **Facility Managers**: Track building conditions and maintenance needs
* **Capital Planners**: Make informed decisions about investments
* **Assessment Teams**: Conduct systematic building evaluations
* **Executives**: Access high-level insights and reports
* **Maintenance Staff**: Understand immediate repair needs

### The Problem ONYX Solves

Traditional building assessments suffer from: - Inconsistent data collection methods - Lost or incomplete paper forms - Difficulty tracking assessment history - Manual FCI calculations prone to errors - Lack of real-time visibility into facility conditions - Challenges in prioritizing capital investments

ONYX addresses these challenges with a digital-first approach that ensures consistency, accuracy, and accessibility.

## Core Concepts

### 1. Facility Condition Index (FCI)

The FCI is the foundational metric in ONYX:

FCI = Total Repair Costs / Current Replacement Value

**FCI Interpretation Ranges:** - **0.00 - 0.10**: Excellent (like new condition) - **0.10 - 0.40**: Good (minor repairs needed) - **0.40 - 0.70**: Fair (significant renovation required) - **0.70+**: Critical (consider replacement)

### 2. Uniformat II Classification

ONYX uses the industry-standard Uniformat II system to categorize building elements: - **Level 1**: Major Group Elements (A-G) - A: Substructure - B: Shell - C: Interiors - D: Services - E: Equipment & Furnishings - F: Special Construction - G: Building Sitework

### 3. Two-Phase Assessment Process

**Phase 1 - Pre-Assessment:** - Building selection and basic information - Element selection based on building type - Preliminary checklist and planning

**Phase 2 - Field Assessment:** - On-site element evaluation - Condition rating (1-5 scale) - Deficiency documentation - Photo evidence collection

### 4. Deficiency Categories

ONYX categorizes deficiencies to help prioritize repairs: 1. **Life Safety & Code Compliance**: Immediate safety concerns 2. **Critical Systems**: Essential building operations 3. **Energy Efficiency**: Sustainability improvements 4. **Asset Life Cycle**: Long-term preservation 5. **User Experience**: Comfort and functionality 6. **Equity & Accessibility**: Inclusive design needs

## Platform Architecture

### Technology Stack

**Frontend:** - React 18.3 with TypeScript - Vite for build tooling - Tailwind CSS for styling - ShadCN UI component library - React Router for navigation - React Hook Form with Zod validation - Recharts for data visualization

**Backend:** - Node.js with Express - TypeScript for type safety - PostgreSQL database - JWT authentication - RESTful API design - Cloudinary for image storage

**Infrastructure:** - Deployed on Render.com - PostgreSQL hosted on Render - Environment-based configuration - HTTPS encryption

### System Architecture Diagram

┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐  
│ │ │ │ │ │  
│ React Frontend │────▶│ Express API │────▶│ PostgreSQL DB │  
│ │ │ │ │ │  
└─────────────────┘ └─────────────────┘ └─────────────────┘  
 │ │ │  
 │ │ │  
 ▼ ▼ ▼  
┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐  
│ │ │ │ │ │  
│ Cloudinary │ │ Mailgun │ │ Render.com │  
│ (Image Storage)│ │ (Email Service)│ │ (Hosting) │  
│ │ │ │ │ │  
└─────────────────┘ └─────────────────┘ └─────────────────┘

### Database Schema

**Core Tables:** 1. **organizations**: Multi-tenant isolation 2. **users**: User accounts with roles 3. **buildings**: Facility information 4. **elements**: Uniformat II building components 5. **assessments**: Assessment records 6. **assessment\_elements**: Element evaluations 7. **assessment\_deficiencies**: Identified issues 8. **reports**: Generated reports

## Features Overview

### Dashboard

* Real-time metrics and KPIs
* Buildings at risk (high FCI)
* Recent assessment activity
* FCI distribution charts
* Trend analysis

### Building Management

* Complete CRUD operations
* Building profiles with photos
* Assessment history tracking
* Document management
* Location mapping

### Assessment System

* Two-phase workflow
* Element-by-element evaluation
* Photo documentation
* Deficiency tracking
* FCI auto-calculation
* Status management

### Reporting

* Automated report generation
* Multiple format exports (PDF, Excel)
* Executive summaries
* Detailed cost breakdowns
* Trend analysis

### Analytics

* Portfolio-wide insights
* Cost projections
* Efficiency metrics
* Predictive maintenance
* Custom date ranges

### Team Management

* User invitations
* Role assignment
* Activity tracking
* Permissions management

## User Roles & Permissions

### 1. Admin

* Full system access
* Organization management
* User management
* All building/assessment operations
* Report generation
* System configuration

### 2. Manager

* Building management
* Assessment assignment
* Report generation
* Team member invitation
* Analytics access

### 3. Assessor

* Conduct assessments
* View assigned buildings
* Upload photos
* Complete evaluations
* View own assessments

## Complete Feature Guide

### 1. Authentication & Onboarding

**Login Process:** 1. Navigate to login page 2. Enter email and password 3. System validates credentials 4. JWT tokens issued 5. Redirect to dashboard

**Current Login:** - Email: admin@onyx.com - Password: password123

**Security Features:** - JWT with refresh tokens - Automatic token renewal - Secure password hashing (bcrypt) - Session management

### 2. Dashboard Features

**Key Metrics Display:** - Total Buildings: Active facility count - Assessments YTD: Yearly completions - Average FCI: Portfolio health indicator - Estimated Repairs: Total cost projection

**Buildings at Risk:** - Displays top 5 highest FCI buildings - Color-coded status indicators - Quick navigation to details - Progress bars for visual FCI

**Assessment Activity Tabs:** - Recent: Completed assessments - Upcoming: Scheduled assessments - Assignee information - Status indicators

### 3. Building Management

**Adding a Building:** 1. Click “Add Building” button 2. Fill required information: - Name and address - Building type and use - Square footage - Year built - Replacement value 3. Upload building photo 4. Set location on map 5. Save building profile

**Building Details Page:** - Overview tab with key information - Assessment history timeline - Document repository - Photo gallery - Location map - Quick actions menu

### 4. Assessment Workflow

**Pre-Assessment Phase:** 1. Select building from dropdown 2. Choose assessment type 3. Select applicable elements 4. Complete preparation checklist 5. Save and continue

**Field Assessment Phase:** 1. Load pre-assessment data 2. For each element: - Set condition rating (1-5) - Document deficiencies - Upload photos - Add notes 3. Review summary 4. Complete assessment

**Condition Rating Scale:** - 5: Excellent - Like new - 4: Good - Minor wear - 3: Fair - Moderate issues - 2: Poor - Significant problems - 1: Critical - Immediate attention

### 5. Report Generation

**Creating Reports:** 1. Navigate to Reports section 2. Click “Generate Report” 3. Select assessment or building 4. Choose report type: - Facility Condition Report - Maintenance Plan - Capital Assessment 5. Add executive summary 6. Generate and download

**Report Contents:** - Executive summary - Building overview - FCI calculation details - Element conditions - Deficiency list with costs - Photo documentation - Recommendations

### 6. Analytics Dashboard

**Available Analytics:** - Portfolio Overview - FCI Trends - Cost Projections - Building Comparisons - Deficiency Analysis - Budget Planning

**Filtering Options:** - Date ranges - Building types - Geographic regions - FCI thresholds - Assessment status

## Technical Documentation

### Frontend Structure

src/  
├── components/ # Reusable UI components  
│ ├── ui/ # ShadCN components  
│ └── [feature]/ # Feature-specific components  
├── context/ # React context providers  
│ ├── auth-context.tsx  
│ └── org-context.tsx  
├── hooks/ # Custom React hooks  
├── layouts/ # Page layouts  
├── pages/ # Route components  
├── services/ # API integration  
├── lib/ # Utilities  
└── types/ # TypeScript definitions

### Backend Structure

backend/  
├── src/  
│ ├── config/ # Database configuration  
│ ├── controllers/ # Request handlers  
│ ├── middleware/ # Express middleware  
│ ├── routes/ # API routes  
│ ├── services/ # Business logic  
│ ├── types/ # TypeScript types  
│ └── utils/ # Helper functions  
├── migrations/ # Database migrations  
└── tests/ # Test files

### Key Design Patterns

**1. Context-Based State Management:** - AuthContext for user authentication - OrgContext for organization data - Prevents prop drilling - Centralized state updates

**2. Protected Routes:** - Route guards check authentication - Role-based access control - Automatic redirects

**3. API Service Layer:** - Centralized API configuration - Interceptors for auth tokens - Error handling - Request/response transformation

**4. Component Composition:** - ShadCN UI primitives - Consistent styling - Accessible by default - Customizable variants

## API Reference

### Authentication Endpoints

**POST /api/auth/login**

Request:  
{  
 "email": "user@example.com",  
 "password": "password123"  
}  
  
Response:  
{  
 "success": true,  
 "data": {  
 "user": { ... },  
 "tokens": {  
 "accessToken": "...",  
 "refreshToken": "..."  
 }  
 }  
}

**POST /api/auth/refresh**

Request:  
{  
 "refreshToken": "..."  
}  
  
Response:  
{  
 "success": true,  
 "data": {  
 "tokens": { ... }  
 }  
}

### Building Endpoints

**GET /api/buildings** - Fetches all buildings for organization - Supports pagination and filtering

**POST /api/buildings** - Creates new building - Requires building data and organization context

**PUT /api/buildings/:id** - Updates building information - Partial updates supported

**DELETE /api/buildings/:id** - Soft deletes building - Maintains assessment history

### Assessment Endpoints

**GET /api/assessments** - Query parameters: - building\_id - status - type - assigned\_to

**POST /api/assessments** - Creates new assessment - Links to building and user

**PUT /api/assessments/:id/elements/:elementId** - Updates element evaluation - Includes condition, notes, deficiencies

**POST /api/assessments/:id/complete** - Finalizes assessment - Calculates FCI - Updates building status

### Report Endpoints

**POST /api/reports/generate/:assessmentId** - Generates report from assessment - Returns report ID

**GET /api/reports/:id/download** - Downloads report file - Supports PDF and Excel formats

## Deployment Guide

### Prerequisites

* Node.js 18+
* PostgreSQL 14+
* Git
* Cloudinary account
* Mailgun account (optional)

### Local Development Setup

1. **Clone Repository:**

git clone [repository-url]  
cd onyx

1. **Install Dependencies:**

# Frontend  
npm install  
  
# Backend  
cd backend  
npm install

1. **Environment Configuration:**

Frontend (.env):

VITE\_API\_URL=http://localhost:5001/api  
VITE\_ENV=development

Backend (.env):

PORT=5001  
NODE\_ENV=development  
DATABASE\_URL=postgresql://user:password@localhost:5432/onyx  
JWT\_SECRET=your-secret-key  
JWT\_REFRESH\_SECRET=your-refresh-secret  
CLOUDINARY\_CLOUD\_NAME=your-cloud-name  
CLOUDINARY\_API\_KEY=your-api-key  
CLOUDINARY\_API\_SECRET=your-api-secret

1. **Database Setup:**

# Create database  
createdb onyx  
  
# Run migrations  
cd backend  
npm run migrate

1. **Start Development Servers:**

# Terminal 1 - Frontend  
npm run dev  
  
# Terminal 2 - Backend  
cd backend  
npm run dev

### Production Deployment

**Render.com Deployment:**

1. **Database Service:**
   * Create PostgreSQL instance
   * Note connection string
2. **Backend Service:**
   * Connect GitHub repository
   * Set build command: cd backend && npm install && npm run build
   * Set start command: cd backend && npm start
   * Add environment variables
3. **Frontend Service:**
   * Connect GitHub repository
   * Set build command: npm install && npm run build
   * Set publish directory: dist
   * Add environment variables
4. **Domain Configuration:**
   * Add custom domain
   * Configure SSL certificates
   * Update CORS settings

## Security & Compliance

### Security Measures

1. **Authentication:**
   * JWT tokens with expiration
   * Refresh token rotation
   * Bcrypt password hashing
   * Account lockout after failed attempts
2. **Authorization:**
   * Role-based access control (RBAC)
   * Organization-level isolation
   * API endpoint protection
   * Resource ownership validation
3. **Data Protection:**
   * HTTPS encryption in transit
   * Database encryption at rest
   * Secure file upload validation
   * SQL injection prevention
4. **Infrastructure:**
   * Regular security updates
   * Automated backups
   * DDoS protection
   * Rate limiting

### Compliance Considerations

* GDPR compliance for EU users
* SOC 2 readiness
* Data retention policies
* User consent management
* Audit logging

## Troubleshooting

### Common Issues

**1. Dashboard Shows Blank Page** - Check browser console for errors - Verify API connection - Ensure organization data loads - Check authentication status

**2. Login Fails** - Verify credentials - Check API endpoint - Confirm database connection - Review CORS settings

**3. Images Not Uploading** - Check Cloudinary configuration - Verify file size limits - Ensure proper file formats - Check network connectivity

**4. Reports Not Generating** - Verify assessment completion - Check report service status - Ensure sufficient data - Review error logs

### Debug Mode

Enable debug logging:

// Frontend  
localStorage.setItem('debug', 'true');  
  
// Backend  
DEBUG=onyx:\* npm run dev

### Support Resources

* GitHub Issues: Report bugs
* Documentation: This guide
* API Status: Check service health
* Logs: Application and error logs

## Future Roadmap

### Planned Features

**Q1 2025:** - Mobile application (iOS/Android) - Offline assessment capability - Advanced analytics dashboard - API v2 with GraphQL

**Q2 2025:** - AI-powered deficiency detection - Predictive maintenance modeling - Integration marketplace - Custom report templates

**Q3 2025:** - IoT sensor integration - Real-time monitoring - Automated workflows - Multi-language support

**Q4 2025:** - Machine learning FCI predictions - Virtual reality walkthroughs - Blockchain audit trail - Enterprise SSO

### Integration Roadmap

* **Accounting Systems**: QuickBooks, SAP
* **CMMS Platforms**: Maximo, Maintenace Connection
* **BIM Software**: Revit, AutoCAD
* **ERP Systems**: Oracle, Microsoft Dynamics

## Glossary

**Assessment**: Systematic evaluation of building condition

**Building Element**: Component of a building system (Uniformat II)

**Deficiency**: Identified problem requiring repair or replacement

**FCI (Facility Condition Index)**: Ratio of repair costs to replacement value

**Field Assessment**: On-site building evaluation phase

**Multi-tenant**: Architecture supporting multiple isolated organizations

**Pre-Assessment**: Planning phase before field evaluation

**Replacement Value**: Current cost to rebuild facility

**SaaS**: Software as a Service delivery model

**Uniformat II**: Standard classification for building elements

**Condition Rating**: 1-5 scale for element evaluation

**Capital Planning**: Long-term facility investment strategy

## Conclusion

ONYX represents a significant advancement in facility condition assessment technology. By digitizing and streamlining the assessment process, it enables organizations to make data-driven decisions about their facility investments.

This Bible serves as the definitive guide to understanding, implementing, and maximizing the value of the ONYX platform. As the platform evolves, this documentation will be updated to reflect new features and capabilities.

For the latest updates and community support, visit the ONYX GitHub repository or contact the development team.

*Last Updated: December 2024* *Version: 1.0* *© 2024 ONYX Platform - Building Intelligence for Tomorrow*