# **Legacy Concierge WordPress Project**

Legacy Concierge is a professional WordPress website built with the Bridge theme by Qode Interactive. This project uses a Dockerized WordPress installation with MariaDB, optimized for performance and security.

## **🚨 Critical Security Issues Requiring Immediate Action**

### **❗ Misplaced WordPress Core Files (CRITICAL)**

**IMMEDIATE ACTION REQUIRED**: The following files are in incorrect locations and pose severe security risks:

1. wp-content/about.php - Should be in wp-admin/
2. wp-content/includes/class-wp-site-health.php - Should be in wp-admin/includes/

**Actions to Take**:

1. **DELETE** these files immediately from wp-content
2. Run full security scan (Wordfence/Sucuri)
3. Verify WordPress core integrity from Dashboard → Updates
4. Re-install current WordPress version to restore proper files

### **❗ Child Theme Implementation (CRITICAL)**

**Status**: Not implemented - All customizations at risk **Impact**: Theme updates will permanently delete all customizations **Priority**: Must implement before any theme updates

### **❗ Performance Optimization Issues**

**Dual Page Builders**: Both Elementor and Visual Composer active **Impact**: Significant performance degradation, poor Core Web Vitals **Action**: Plan migration from VC to Elementor, remove VC when complete

## **🚀 Quick Start**

### **Prerequisites**

1. Docker and Docker Compose installed
2. Git for version control
3. Basic knowledge of WordPress administration

### **Initial Setup**

1. **Clone and Navigate to Project**

cd /path/to/legacyConcierge

1. **Environment Configuration**

# Copy the environment template

cp .env.example .env

# Edit .env with your secure credentials

nano .env

1. **Start the Development Environment**

# Start all services

docker-compose up -d

# Check status

docker-compose ps

1. **Access the Site**
2. **Frontend**: [http://localhost:8000](http://localhost:8000/)
3. **WordPress Admin**:<http://localhost:8000/wp-admin>
4. **Database**: localhost:3306 (from host machine)

## **🏗️ Project Structure**

legacyConcierge/

├── wp-content/ # WordPress content directory

│ ├── themes/

│ │ ├── bridge/ # Active theme (Qode Interactive Bridge)

│ │ ├── twentytwentyfive/ # Default themes

│ │ └── ...

│ ├── plugins/ # Installed plugins

│ │ ├── elementor/ # Page builder

│ │ ├── woocommerce/ # E-commerce functionality

│ │ ├── jetpack/ # WordPress.com features

│ │ ├── yoast-seo/ # SEO optimization

│ │ └── ...

│ └── uploads/ # Media files

├── bin/ # Utility scripts

│ ├── backup.sh # Automated backup script

│ ├── security-check.sh # Security monitoring

│ └── docker-healthcheck.sh # Container health monitoring

├── mysql/ # Database configuration

│ └── my.cnf # MariaDB custom configuration

├── docker-compose.yml # Container orchestration

├── wp-config.php # WordPress configuration

├── .htaccess # Apache security & rewrite rules

├── .env.example # Environment template

└── docs/ # Project documentation

## **🎨 Theme & Design**

### **Active Theme: Bridge by Qode Interactive**

1. **Version**: 30.8.8.3
2. **Type**: Creative Multipurpose WordPress Theme
3. **Features**:
4. Visual Composer integration
5. Responsive design
6. Multiple layout options
7. WooCommerce compatibility
8. Custom post types for portfolios

### **Theme Customization**

1. **Theme files**: /wp-content/themes/bridge/
2. **Custom CSS**: Located in /wp-content/themes/bridge/css/
3. **Child theme**: ⚠️ **CRITICAL - NOT IMPLEMENTED** (Must be implemented immediately)
4. **Risk**: All theme customizations will be lost on theme updates
5. **Action Required**: Create child theme and migrate all customizations

## **🔌 Key Plugins**

### **Page Builder & Design**

1. **Elementor Pro** - Advanced page building (Primary)
2. **Visual Composer (WPBakery)** - ⚠️ **PERFORMANCE CONCERN**
3. **Issue**: Running dual page builders significantly impacts performance
4. **Recommendation**: Migrate all VC content to Elementor, then remove VC
5. **Qode Addons for Elementor** - Extended widgets
6. **Revolution Slider** - Advanced slider functionality

### **E-Commerce**

1. **WooCommerce Extensions**:
2. Qode Quick View
3. Qode Wishlist
4. Various payment integrations

### **SEO & Marketing**

1. **Yoast SEO** - Search engine optimization
2. **Google Site Kit** - Analytics integration
3. **Jetpack** - WordPress.com features
4. **Instagram Feed Pro** - Social media integration

### **Performance & Security**

1. **LiteSpeed Cache** - ⚠️ **HOSTING DEPENDENCY**
2. **Note**: Maximum benefit only on LiteSpeed servers
3. **Action Required**: Verify hosting environment and optimize accordingly
4. **EWWW Image Optimizer** - Image compression (Ensure WebP generation enabled)
5. **Easy Hide Login** - Security enhancement
6. **All-in-One WP Migration** - Backup & migration (Evaluate necessity)

### **Development Tools**

1. **Classic Editor** - Legacy editor support
2. **Classic Widgets** - Widget management

## **🔧 Development Environment**

### **Docker Configuration**

The project uses Docker Compose with two main services:

#### **WordPress Container**

1. **Image**: wordpress:6.4-php8.2-apache
2. **Port**: 8000 (mapped to 80 inside container)
3. **PHP Version**: 8.2
4. **Web Server**: Apache

#### **Database Container**

1. **Image**: mariadb:10.11-jammy
2. **Database**: MariaDB 10.11
3. **Port**: 3306
4. **Custom Configuration**: mysql/my.cnf

### **Environment Variables**

Required environment variables (set in .env):

# Database Configuration

MYSQL\_DATABASE=wordpress

MYSQL\_USER=wordpress\_user

MYSQL\_PASSWORD=your\_secure\_password

MYSQL\_ROOT\_PASSWORD=your\_root\_password

# WordPress Configuration

WORDPRESS\_DEBUG=false

WORDPRESS\_DEBUG\_LOG=true

WORDPRESS\_DEBUG\_DISPLAY=false

### **WordPress Configuration Highlights**

1. **Security**: File editing disabled, automatic updates enabled
2. **Performance**: Memory limits increased (512MB/1024MB)
3. **Debugging**: Enabled for logs, disabled for display
4. **Database**: Optimized with custom MariaDB configuration

## **🔒 Security Features**

### **Implemented Security Measures**

1. **Environment Variable Protection**
2. Database credentials stored in .env
3. Sensitive data excluded from version control
4. **WordPress Hardening**
5. File editing disabled (DISALLOW\_FILE\_EDIT)
6. File modifications restricted (DISALLOW\_FILE\_MODS)
7. Automatic security updates enabled
8. **Apache Security Headers** (.htaccess)
9. XSS Protection
10. Content Type Options
11. Frame Options (DENY)
12. Referrer Policy
13. Permissions Policy
14. **Database Security**
15. Custom user with limited privileges
16. SQL strict mode enabled
17. Connection limits configured
18. **File Protection**
19. wp-config.php access denied
20. .htaccess access restricted
21. Sensitive files hidden

### **Security Monitoring**

1. **Automated Script**: bin/security-check.sh
2. **File Permission Checks**: Ensures proper file/directory permissions
3. **Suspicious Code Detection**: Scans for malicious code patterns
4. **Core Integrity Verification**: Checks WordPress core file integrity
5. **Plugin/Theme Security**: Validates installed components
6. **Failed Login Monitoring**: Tracks authentication attempts
7. **Regular Checks**: File permissions, suspicious code, core integrity
8. **Security Report**: Documented in SECURITY\_AUDIT\_REPORT.md

## **📊 Performance Optimization**

### **Database Optimization**

1. **InnoDB Buffer Pool**: 256MB
2. **Connection Limits**: 50 max connections
3. **Query Logging**: Slow queries logged (>2 seconds)

### **WordPress Optimization**

1. **Memory Limits**: 512MB standard, 1024MB maximum
2. **Post Revisions**: Limited to 5 per post
3. **Autosave Interval**: 5 minutes
4. **HTTP Timeout**: 60 seconds

### **Caching**

1. **LiteSpeed Cache Plugin**: Active for performance
2. **A2 Optimized**: Additional performance features

## **🔄 Backup & Maintenance**

### **Automated Backup Script**

The bin/backup.sh script provides:

1. **Database Backup**: MySQL dump with compression
2. **File Backup**: WordPress content and configuration
3. **Retention Policy**: 30-day automatic cleanup
4. **Schedule**: Run via cron for regular backups

### **Manual Backup Process**

# Run backup script

./bin/backup.sh

# Manual database backup

docker exec legacy-concierge\_db mysqldump -u $MYSQL\_USER -p$MYSQL\_PASSWORD $MYSQL\_DATABASE > backup.sql

# Manual file backup

tar -czf wordpress-backup.tar.gz wp-content/

### **Update Procedures**

1. **WordPress Core**: Automatic minor updates enabled
2. **Plugins**: Update through admin dashboard or WP-CLI
3. **Theme**: Update through Envato Market plugin
4. **Database**: Automatic optimization via plugins

## **🛠️ Development Workflow**

### **Local Development**

1. **Start Environment**

docker-compose up -d

1. **Development URLs**
2. Site: [http://localhost:8000](http://localhost:8000/)
3. Admin:<http://localhost:8000/wp-admin>
4. **Database Access**

# Connect to database container

docker exec -it legacy-concierge\_db mysql -u $MYSQL\_USER -p

1. **File Changes**
2. Direct file editing in wp-content/
3. Theme customization in wp-content/themes/bridge/
4. Plugin development in wp-content/plugins/

### **Content Management**

#### **For Content Authors**

1. **Accessing Admin**
2. URL:<http://localhost:8000/wp-admin>
3. Use provided WordPress admin credentials
4. **Content Creation**
5. **Posts**: Blog articles and news
6. **Pages**: Static content pages
7. **Portfolios**: Project showcases (Bridge theme feature)
8. **Products**: WooCommerce items
9. **Page Building**
10. **Elementor**: Primary page builder for new content
11. **Visual Composer**: Legacy content (avoid for new pages)
12. **WordPress Editor**: Simple content and blog posts
13. **Media Management**
14. **Upload Location**: wp-content/uploads/
15. **Optimization**: Automatic via EWWW plugin
16. **Organization**: Use folders and tags

#### **For Developers**

1. **Theme Development**
2. **Base Theme**: Bridge (Qode Interactive)
3. **Customization**: Direct theme files or child theme
4. **Testing**: Always test changes in staging
5. **Plugin Development**
6. **Custom Plugins**: Place in wp-content/plugins/
7. **Hooks**: Use WordPress action/filter system
8. **Dependencies**: Document in plugin headers
9. **Database Changes**
10. **Migrations**: Use WordPress update functions
11. **Backups**: Always backup before changes
12. **Testing**: Test on copy of production data

## **🚨 Troubleshooting**

### **Common Issues**

#### **Site Not Loading**

# Check container status

docker-compose ps

# Check logs

docker-compose logs wordpress

docker-compose logs db

# Restart services

docker-compose restart

#### **Database Connection Issues**

# Verify environment variables

cat .env

# Test database connection

docker exec legacy-concierge\_db mysql -u $MYSQL\_USER -p$MYSQL\_PASSWORD -e "SHOW DATABASES;"

#### **Performance Issues**

# Check resource usage

docker stats

# Clear cache

# Via LiteSpeed Cache plugin admin

# Or manually clear wp-content/cache/

#### **Plugin Conflicts**

1. Deactivate all plugins
2. Activate one by one to identify conflict
3. Check error logs: wp-content/debug.log

### **Log Locations**

1. **WordPress Errors**: wp-content/debug.log
2. **Apache Errors**: Container logs via docker-compose logs wordpress
3. **Database Errors**: Container logs via docker-compose logs db
4. **Security Logs**: Generated by bin/security-check.sh

## **📋 Maintenance Checklist**

### **🚨 IMMEDIATE Priority Tasks**

1. **CRITICAL**: Remove misplaced core files from wp-content directory
2. **CRITICAL**: Implement child theme for Bridge theme
3. **HIGH**: Plan migration from Visual Composer to Elementor
4. **HIGH**: Verify LiteSpeed Cache compatibility with hosting environment
5. **MEDIUM**: Audit and remove redundant plugins

### **Weekly Tasks**

1. Run security check script
2. Check for plugin updates
3. Review performance metrics
4. Backup verification
5. Monitor Core Web Vitals performance

### **Monthly Tasks**

1. WordPress core updates
2. Theme updates (via Envato Market)
3. Database optimization and cleanup
4. Security audit review
5. Clean old backups
6. Review plugin necessity and performance impact

### **Quarterly Tasks**

1. Full security penetration test
2. Performance optimization review
3. Backup strategy evaluation
4. Plugin audit (remove unused)
5. Update documentation
6. Child theme maintenance review

## **🔗 Important Links**

1. **WordPress Codex**:<https://codex.wordpress.org/>
2. **Bridge Theme Documentation**:<https://bridgelanding.qodeinteractive.com/>
3. **Elementor Documentation**:<https://docs.elementor.com/>
4. **WooCommerce Documentation**:<https://docs.woocommerce.com/>

## **⚠️ Important Notes**

### **Security Considerations**

1. Never commit .env file to version control
2. Regularly update all components
3. Monitor security logs
4. Use strong passwords
5. Enable two-factor authentication

### **Performance Considerations**

1. Monitor database size and optimize regularly
2. Use caching plugins effectively
3. Optimize images before upload
4. Monitor server resources

### **Backup Considerations**

1. Test backup restoration regularly
2. Store backups offsite
3. Document restoration procedures
4. Maintain multiple backup versions

## **📞 Support**

For technical issues or questions:

1. **Check Logs**: Review error logs first
2. **Documentation**: Consult this README and linked resources
3. **Community**: WordPress.org support forums
4. **Professional**: Consider WordPress maintenance services

## **📋 Related Documentation**

1. [Improvement Checklist](https://www.docstomarkdown.pro/convert-markdown-to-google-docs-online/IMPROVEMENT_CHECKLIST.md) - Detailed improvement recommendations
2. [Security Audit Report](https://www.docstomarkdown.pro/convert-markdown-to-google-docs-online/SECURITY_AUDIT_REPORT.md) - Security analysis and fixes
3. [Maintenance History](https://www.docstomarkdown.pro/convert-markdown-to-google-docs-online/MAINTENANCE_HISTORY.md) - Complete maintenance activity log

**Last Updated**: August 18, 2025 **WordPress Version**: 6.4 (⚠️ Version discrepancy requires verification) **PHP Version**: 8.2 **Database**: MariaDB 10.11