# BGHS Alumni Website - Test Cases Document

## Document Information

**Project:** BGHS Alumni Website  
**Version:** 1.0  
**Date:** December 2024  
**Prepared by:** QA Team  
**Review Status:** Draft

## Table of Contents

1. [Test Overview](#test-overview)
2. [Test Environment](#test-environment)
3. [Test Data](#test-data)
4. [Functional Test Cases](#functional-test-cases)
5. [Non-Functional Test Cases](#non-functional-test-cases)
6. [Security Test Cases](#security-test-cases)
7. [Integration Test Cases](#integration-test-cases)
8. [User Acceptance Test Cases](#user-acceptance-test-cases)
9. [Test Execution Plan](#test-execution-plan)
10. [Defect Management](#defect-management)

## Test Overview

### Purpose

This document outlines comprehensive test cases for the BGHS Alumni Website to ensure all functionality works as expected, meets user requirements, and maintains high quality standards.

### Scope

* User registration and authentication
* Profile management
* Alumni directory
* Events management
* Blog system
* Photo gallery
* Donation system
* Admin functions
* Security and performance

### Test Objectives

* Verify all functional requirements are met
* Ensure system security and data protection
* Validate user experience and interface usability
* Confirm system performance under various loads
* Test integration with external services

## Test Environment

### Test Environment Setup

* **URL:** http://localhost:3001 (Development)
* **Database:** Supabase PostgreSQL
* **Storage:** Cloudflare R2
* **Email Service:** SendGrid
* **Payment Gateway:** Stripe (Test Mode)

### Browser Compatibility

* Chrome (Latest)
* Firefox (Latest)
* Safari (Latest)
* Edge (Latest)
* Mobile browsers (iOS Safari, Chrome Mobile)

### Test Data Requirements

* Valid user accounts (various roles)
* Sample alumni data
* Test images for gallery
* Mock payment data
* Email test accounts

## Test Data

### User Accounts

| Role | Email | Password | Status |
| --- | --- | --- | --- |
| Super Admin | admin@alumnibghs.org | AdminPass123! | Active |
| Alumni Member | member@test.com | MemberPass123! | Active |
| Content Moderator | moderator@test.com | ModPass123! | Active |
| Event Manager | events@test.com | EventPass123! | Active |
| Pending User | pending@test.com | PendingPass123! | Pending |

### Sample Data

* **Alumni Records:** 50+ test alumni with various profiles
* **Events:** 10+ test events (past, current, future)
* **Blog Posts:** 15+ test blog posts
* **Gallery Photos:** 25+ test images in various categories
* **Donation Campaigns:** 5+ test campaigns

## Functional Test Cases

### TC-001: User Registration

**Test Case ID:** TC-001  
**Test Case Name:** User Registration with Valid Data  
**Priority:** High  
**Test Type:** Functional

**Preconditions:** - User is on the registration page - Registration form is accessible

**Test Steps:** 1. Navigate to /register 2. Enter valid first name: “John” 3. Enter valid last name: “Doe” 4. Enter valid email: “john.doe@test.com” 5. Enter valid password: “TestPass123!” 6. Enter batch year: “2010” 7. Click “Create Account” button

**Expected Result:** - User account is created successfully - Confirmation message is displayed - User receives email confirmation - Account status is set to “Pending Approval”

**Test Data:** - First Name: John - Last Name: Doe - Email: john.doe@test.com - Password: TestPass123! - Batch Year: 2010

### TC-002: User Registration - Invalid Email

**Test Case ID:** TC-002  
**Test Case Name:** User Registration with Invalid Email  
**Priority:** High  
**Test Type:** Functional

**Preconditions:** - User is on the registration page

**Test Steps:** 1. Navigate to /register 2. Enter valid first name: “Jane” 3. Enter valid last name: “Smith” 4. Enter invalid email: “invalid-email” 5. Enter valid password: “TestPass123!” 6. Enter batch year: “2015” 7. Click “Create Account” button

**Expected Result:** - Error message displayed: “Please enter a valid email address” - Account is not created - Form remains on registration page

### TC-003: User Login - Valid Credentials

**Test Case ID:** TC-003  
**Test Case Name:** User Login with Valid Credentials  
**Priority:** High  
**Test Type:** Functional

**Preconditions:** - User account exists and is approved - User is on the login page

**Test Steps:** 1. Navigate to /login 2. Enter valid email: “member@test.com” 3. Enter valid password: “MemberPass123!” 4. Click “Sign In” button

**Expected Result:** - User is successfully logged in - Redirected to dashboard - User session is established - Navigation shows user-specific options

### TC-004: User Login - Invalid Credentials

**Test Case ID:** TC-004  
**Test Case Name:** User Login with Invalid Credentials  
**Priority:** High  
**Test Type:** Functional

**Preconditions:** - User is on the login page

**Test Steps:** 1. Navigate to /login 2. Enter valid email: “member@test.com” 3. Enter invalid password: “WrongPassword” 4. Click “Sign In” button

**Expected Result:** - Error message displayed: “Invalid email or password” - User remains on login page - No session is established

### TC-005: Password Reset - Valid Email

**Test Case ID:** TC-005  
**Test Case Name:** Password Reset with Valid Email  
**Priority:** Medium  
**Test Type:** Functional

**Preconditions:** - User account exists - User is on the forgot password page

**Test Steps:** 1. Navigate to /forgot-password 2. Enter valid email: “member@test.com” 3. Click “Send Reset Code” button 4. Check email for OTP code 5. Enter received OTP code 6. Enter new password: “NewPass123!” 7. Confirm new password: “NewPass123!” 8. Click “Reset Password” button

**Expected Result:** - OTP is sent to email - Password is successfully reset - User can login with new password

### TC-006: Profile Management - Update Profile

**Test Case ID:** TC-006  
**Test Case Name:** Update User Profile Information  
**Priority:** Medium  
**Test Type:** Functional

**Preconditions:** - User is logged in - User is on profile page

**Test Steps:** 1. Navigate to /dashboard 2. Click “My Profile” option 3. Update phone number: “+91-9876543210” 4. Update profession: “Software Engineer” 5. Update company: “Tech Corp” 6. Click “Save Changes” button

**Expected Result:** - Profile information is updated successfully - Success message is displayed - Updated information is visible on profile page

### TC-007: Alumni Directory - View Directory

**Test Case ID:** TC-007  
**Test Case Name:** View Alumni Directory  
**Priority:** High  
**Test Type:** Functional

**Preconditions:** - User is logged in - Alumni data exists in database

**Test Steps:** 1. Navigate to /directory 2. Verify directory page loads 3. Check if alumni list is displayed 4. Verify search functionality 5. Test filtering options

**Expected Result:** - Directory page loads successfully - Alumni list is displayed with proper information - Search and filter functions work correctly - Privacy settings are respected

### TC-008: Alumni Directory - Anonymous Access

**Test Case ID:** TC-008  
**Test Case Name:** Anonymous User Directory Access  
**Priority:** High  
**Test Type:** Functional

**Preconditions:** - User is not logged in - Alumni data exists in database

**Test Steps:** 1. Navigate to /directory without login 2. Verify teaser content is displayed 3. Check if login prompt is shown 4. Attempt to view full directory

**Expected Result:** - Teaser content is displayed - Login prompt is shown - Full directory access is restricted - User is redirected to login page

### TC-009: Events - View Events List

**Test Case ID:** TC-009  
**Test Case Name:** View Events List  
**Priority:** Medium  
**Test Type:** Functional

**Preconditions:** - User is logged in - Events exist in database

**Test Steps:** 1. Navigate to /events 2. Verify events page loads 3. Check if events are displayed 4. Test event filtering (past/current/future) 5. Click on an event to view details

**Expected Result:** - Events page loads successfully - Events are displayed with proper information - Filtering works correctly - Event details are accessible

### TC-010: Events - Create Event (Authorized User)

**Test Case ID:** TC-010  
**Test Case Name:** Create New Event  
**Priority:** Medium  
**Test Type:** Functional

**Preconditions:** - User with event manager role is logged in - User is on events page

**Test Steps:** 1. Navigate to /events 2. Click “Create Event” button 3. Fill in event details: - Title: “Annual Alumni Meet 2025” - Description: “Annual gathering of BGHS alumni” - Date: “2025-06-15” - Time: “10:00 AM” - Location: “School Auditorium” 4. Click “Create Event” button

**Expected Result:** - Event is created successfully - Event appears in events list - Success message is displayed - Event is visible to all users

### TC-011: Events - Create Event (Unauthorized User)

**Test Case ID:** TC-011  
**Test Case Name:** Create Event - Unauthorized Access  
**Priority:** Medium  
**Test Type:** Functional

**Preconditions:** - Regular user (not event manager) is logged in - User is on events page

**Test Steps:** 1. Navigate to /events 2. Look for “Create Event” button 3. Attempt to access event creation

**Expected Result:** - “Create Event” button is not visible - Access is denied with appropriate message - User cannot create events

### TC-012: Blog - View Blog Posts

**Test Case ID:** TC-012  
**Test Case Name:** View Blog Posts  
**Priority:** Low  
**Test Type:** Functional

**Preconditions:** - User is logged in - Blog posts exist in database

**Test Steps:** 1. Navigate to /blog 2. Verify blog page loads 3. Check if blog posts are displayed 4. Click on a blog post to read full content 5. Test pagination if available

**Expected Result:** - Blog page loads successfully - Blog posts are displayed with proper formatting - Full content is accessible - Pagination works correctly

### TC-013: Blog - Create Blog Post (Authorized User)

**Test Case ID:** TC-013  
**Test Case Name:** Create Blog Post  
**Priority:** Low  
**Test Type:** Functional

**Preconditions:** - User with content creator role is logged in - User is on blog page

**Test Steps:** 1. Navigate to /blog 2. Click “Create Post” button 3. Fill in blog post details: - Title: “Memories of School Days” - Content: “A nostalgic look back at our school days…” - Category: “Memories” 4. Click “Publish” button

**Expected Result:** - Blog post is created successfully - Post appears in blog list - Success message is displayed - Post is visible to all users

### TC-014: Gallery - View Photo Gallery

**Test Case ID:** TC-014  
**Test Case Name:** View Photo Gallery  
**Priority:** Medium  
**Test Type:** Functional

**Preconditions:** - User is logged in - Photos exist in gallery

**Test Steps:** 1. Navigate to /gallery 2. Verify gallery page loads 3. Check if photos are displayed in masonry layout 4. Click on a photo to view enlarged version 5. Test photo navigation (next/previous) 6. Test zoom and rotation features

**Expected Result:** - Gallery page loads successfully - Photos are displayed in proper layout - Photo viewer opens with full functionality - Navigation and controls work correctly

### TC-015: Gallery - Upload Photo (Authorized User)

**Test Case ID:** TC-015  
**Test Case Name:** Upload Photo to Gallery  
**Priority:** Medium  
**Test Type:** Functional

**Preconditions:** - User with upload permissions is logged in - Test image file is available

**Test Steps:** 1. Navigate to /gallery/upload 2. Click “Choose File” button 3. Select a test image file 4. Enter photo details: - Title: “School Building” - Description: “Main school building” - Category: “School” 5. Click “Upload” button

**Expected Result:** - Photo is uploaded successfully - Thumbnail is generated - Photo appears in gallery - Success message is displayed

### TC-016: Gallery - Upload Photo (Unauthorized User)

**Test Case ID:** TC-016  
**Test Case Name:** Upload Photo - Unauthorized Access  
**Priority:** Medium  
**Test Type:** Functional

**Preconditions:** - Regular user (without upload permissions) is logged in

**Test Steps:** 1. Navigate to /gallery/upload 2. Attempt to access upload page

**Expected Result:** - Access is denied - Appropriate error message is displayed - User is redirected or shown unauthorized message

### TC-017: Donations - View Donation Campaigns

**Test Case ID:** TC-017  
**Test Case Name:** View Donation Campaigns  
**Priority:** Medium  
**Test Type:** Functional

**Preconditions:** - User is logged in - Donation campaigns exist

**Test Steps:** 1. Navigate to /donate 2. Verify donation page loads 3. Check if campaigns are displayed 4. View campaign details 5. Test campaign filtering

**Expected Result:** - Donation page loads successfully - Campaigns are displayed with proper information - Campaign details are accessible - Filtering works correctly

### TC-018: Donations - Make Donation

**Test Case ID:** TC-018  
**Test Case Name:** Make Donation  
**Priority:** High  
**Test Type:** Functional

**Preconditions:** - User is logged in - Donation campaign exists - Stripe test mode is configured

**Test Steps:** 1. Navigate to /donate 2. Select a donation campaign 3. Enter donation amount: “100” 4. Click “Donate Now” button 5. Fill in payment details (test card) 6. Complete payment process

**Expected Result:** - Payment is processed successfully - Donation is recorded - Confirmation email is sent - Donation appears in user’s history

### TC-019: Admin - User Management

**Test Case ID:** TC-019  
**Test Case Name:** Admin User Management  
**Priority:** High  
**Test Type:** Functional

**Preconditions:** - Super admin is logged in - Pending users exist

**Test Steps:** 1. Navigate to /admin/users 2. Verify user management page loads 3. View list of all users 4. Approve a pending user 5. Change user role 6. Reset user password

**Expected Result:** - User management page loads successfully - All users are listed - User approval works correctly - Role changes are applied - Password reset is successful

### TC-020: Admin - Content Moderation

**Test Case ID:** TC-020  
**Test Case Name:** Content Moderation  
**Priority:** Medium  
**Test Type:** Functional

**Preconditions:** - Content moderator is logged in - Unapproved content exists

**Test Steps:** 1. Navigate to admin dashboard 2. Access content moderation section 3. Review pending blog posts 4. Approve or reject content 5. Review gallery photos 6. Moderate comments

**Expected Result:** - Moderation interface is accessible - Content can be reviewed and approved/rejected - Changes are applied correctly - Moderation actions are logged

## Non-Functional Test Cases

### TC-021: Performance - Page Load Time

**Test Case ID:** TC-021  
**Test Case Name:** Page Load Performance  
**Priority:** High  
**Test Type:** Performance

**Preconditions:** - Application is deployed - Network conditions are normal

**Test Steps:** 1. Open browser developer tools 2. Navigate to home page 3. Measure page load time 4. Test with various page sizes 5. Test with different network speeds

**Expected Result:** - Home page loads within 3 seconds - All pages load within acceptable time limits - Performance is consistent across browsers

**Acceptance Criteria:** - Page load time < 3 seconds - Time to interactive < 5 seconds - First contentful paint < 2 seconds

### TC-022: Performance - Database Query Performance

**Test Case ID:** TC-022  
**Test Case Name:** Database Query Performance  
**Priority:** Medium  
**Test Type:** Performance

**Preconditions:** - Database has test data - Monitoring tools are available

**Test Steps:** 1. Execute various database queries 2. Monitor query execution time 3. Test with large datasets 4. Check for slow queries 5. Verify indexing effectiveness

**Expected Result:** - All queries execute within acceptable time - No slow queries detected - Database performance is optimized

**Acceptance Criteria:** - Simple queries < 100ms - Complex queries < 500ms - No queries > 1 second

### TC-023: Load Testing - Concurrent Users

**Test Case ID:** TC-023  
**Test Case Name:** Concurrent User Load Testing  
**Priority:** High  
**Test Type:** Performance

**Preconditions:** - Load testing tools are configured - Test environment is ready

**Test Steps:** 1. Simulate 50 concurrent users 2. Monitor system performance 3. Test various user actions 4. Increase load gradually 5. Identify breaking point

**Expected Result:** - System handles 50+ concurrent users - Response times remain acceptable - No system crashes or errors

**Acceptance Criteria:** - Supports 100+ concurrent users - Response time degradation < 50% - Error rate < 1%

### TC-024: Mobile Responsiveness

**Test Case ID:** TC-024  
**Test Case Name:** Mobile Device Compatibility  
**Priority:** High  
**Test Type:** Usability

**Preconditions:** - Application is accessible - Mobile devices are available

**Test Steps:** 1. Access application on mobile device 2. Test all major pages 3. Verify responsive design 4. Test touch interactions 5. Check mobile navigation

**Expected Result:** - Application is fully functional on mobile - Layout adapts to screen size - Touch interactions work correctly - Mobile navigation is intuitive

### TC-025: Cross-Browser Compatibility

**Test Case ID:** TC-025  
**Test Case Name:** Cross-Browser Testing  
**Priority:** Medium  
**Test Type:** Compatibility

**Preconditions:** - Multiple browsers are available - Application is accessible

**Test Steps:** 1. Test on Chrome (latest) 2. Test on Firefox (latest) 3. Test on Safari (latest) 4. Test on Edge (latest) 5. Compare functionality across browsers

**Expected Result:** - Application works consistently across browsers - No browser-specific issues - All features are accessible

## Security Test Cases

### TC-026: Authentication Security

**Test Case ID:** TC-026  
**Test Case Name:** Authentication Security Testing  
**Priority:** High  
**Test Type:** Security

**Preconditions:** - Application is accessible - Security testing tools are available

**Test Steps:** 1. Test SQL injection in login form 2. Test XSS in user input fields 3. Test CSRF protection 4. Test session management 5. Test password security

**Expected Result:** - No SQL injection vulnerabilities - XSS attacks are prevented - CSRF protection is active - Sessions are secure - Passwords are properly hashed

### TC-027: Authorization Testing

**Test Case ID:** TC-027  
**Test Case Name:** Role-Based Access Control  
**Priority:** High  
**Test Type:** Security

**Preconditions:** - Users with different roles exist - Application is accessible

**Test Steps:** 1. Login as regular user 2. Attempt to access admin functions 3. Test URL manipulation 4. Test API endpoint access 5. Verify role-based restrictions

**Expected Result:** - Unauthorized access is prevented - Role-based restrictions work correctly - API endpoints are protected - Error messages don’t reveal sensitive information

### TC-028: Data Protection

**Test Case ID:** TC-028  
**Test Case Name:** Data Protection and Privacy  
**Priority:** High  
**Test Type:** Security

**Preconditions:** - Application contains user data - Privacy settings are configured

**Test Steps:** 1. Test data encryption in transit 2. Test data encryption at rest 3. Verify privacy settings 4. Test data anonymization 5. Check for data leaks

**Expected Result:** - Data is encrypted in transit (HTTPS) - Sensitive data is encrypted at rest - Privacy settings are respected - No data leaks detected

### TC-029: File Upload Security

**Test Case ID:** TC-029  
**Test Case Name:** File Upload Security  
**Priority:** Medium  
**Test Type:** Security

**Preconditions:** - File upload functionality is available - Security testing tools are ready

**Test Steps:** 1. Upload malicious file types 2. Test file size limits 3. Test file content validation 4. Test path traversal attacks 5. Verify file storage security

**Expected Result:** - Malicious files are rejected - File size limits are enforced - File content is validated - Path traversal attacks are prevented - Files are stored securely

## Integration Test Cases

### TC-030: Email Service Integration

**Test Case ID:** TC-030  
**Test Case Name:** SendGrid Email Integration  
**Priority:** High  
**Test Type:** Integration

**Preconditions:** - SendGrid is configured - Test email accounts are available

**Test Steps:** 1. Trigger registration email 2. Trigger password reset email 3. Trigger donation confirmation email 4. Verify email delivery 5. Check email formatting

**Expected Result:** - All emails are sent successfully - Emails are delivered to correct recipients - Email content is properly formatted - Email templates are applied correctly

### TC-031: Payment Gateway Integration

**Test Case ID:** TC-031  
**Test Case Name:** Stripe Payment Integration  
**Priority:** High  
**Test Type:** Integration

**Preconditions:** - Stripe is configured in test mode - Test payment methods are available

**Test Steps:** 1. Process test payment 2. Test payment success flow 3. Test payment failure flow 4. Test webhook handling 5. Verify payment data storage

**Expected Result:** - Payments are processed correctly - Success and failure flows work - Webhooks are handled properly - Payment data is stored securely

### TC-032: Cloud Storage Integration

**Test Case ID:** TC-032  
**Test Case Name:** Cloudflare R2 Storage Integration  
**Priority:** Medium  
**Test Type:** Integration

**Preconditions:** - Cloudflare R2 is configured - Upload functionality is available

**Test Steps:** 1. Upload image file 2. Verify file is stored in R2 3. Test file retrieval 4. Test thumbnail generation 5. Test file deletion

**Expected Result:** - Files are uploaded to R2 successfully - Files are retrievable via public URLs - Thumbnails are generated correctly - File deletion works properly

### TC-033: Database Integration

**Test Case ID:** TC-033  
**Test Case Name:** Supabase Database Integration  
**Priority:** High  
**Test Type:** Integration

**Preconditions:** - Supabase is configured - Database schema is deployed

**Test Steps:** 1. Test CRUD operations 2. Test RLS policies 3. Test database triggers 4. Test data integrity 5. Test backup and recovery

**Expected Result:** - All CRUD operations work correctly - RLS policies are enforced - Triggers function properly - Data integrity is maintained - Backup and recovery work

## User Acceptance Test Cases

### TC-034: End-to-End User Journey

**Test Case ID:** TC-034  
**Test Case Name:** Complete User Registration to Profile Setup  
**Priority:** High  
**Test Type:** UAT

**Preconditions:** - Application is fully functional - Admin approval process is working

**Test Steps:** 1. Register new user account 2. Receive and click email confirmation 3. Wait for admin approval 4. Login with approved account 5. Complete profile setup 6. Explore alumni directory 7. View events and blog posts 8. Upload photo to gallery 9. Make a donation

**Expected Result:** - Complete user journey works smoothly - All features are accessible and functional - User experience is intuitive and satisfying - No major issues or blockers

### TC-035: Admin Workflow

**Test Case ID:** TC-035  
**Test Case Name:** Admin User Management Workflow  
**Priority:** High  
**Test Type:** UAT

**Preconditions:** - Admin account is available - Pending users exist

**Test Steps:** 1. Login as admin 2. Access user management 3. Review and approve pending users 4. Manage user roles 5. Reset user passwords 6. Moderate content 7. Manage events and campaigns

**Expected Result:** - Admin workflow is efficient and intuitive - All admin functions work correctly - Admin can effectively manage the system - No workflow bottlenecks or issues

### TC-036: Content Management Workflow

**Test Case ID:** TC-036  
**Test Case Name:** Content Creation and Management  
**Priority:** Medium  
**Test Type:** UAT

**Preconditions:** - Content creator account is available - Moderation system is working

**Test Steps:** 1. Login as content creator 2. Create blog post 3. Upload photos to gallery 4. Create event 5. Submit content for moderation 6. Review moderation feedback 7. Publish approved content

**Expected Result:** - Content creation process is smooth - Moderation workflow is effective - Content is published correctly - Content management is user-friendly

## Test Execution Plan

### Test Phases

1. **Unit Testing** (Week 1-2)
   * Individual component testing
   * API endpoint testing
   * Database function testing
2. **Integration Testing** (Week 3-4)
   * Service integration testing
   * Third-party service testing
   * Database integration testing
3. **System Testing** (Week 5-6)
   * End-to-end functionality testing
   * Performance testing
   * Security testing
4. **User Acceptance Testing** (Week 7-8)
   * Business scenario testing
   * User workflow testing
   * Final validation

### Test Environment Schedule

* **Development Environment:** Continuous testing
* **Staging Environment:** Weekly testing cycles
* **Production Environment:** Pre-deployment testing only

### Test Data Management

* **Test Data Creation:** Automated scripts
* **Data Refresh:** Daily for development, weekly for staging
* **Data Cleanup:** After each test cycle

### Test Reporting

* **Daily Reports:** Test execution status
* **Weekly Reports:** Defect summary and trends
* **Final Report:** Overall test results and recommendations

## Defect Management

### Defect Classification

* **Critical:** System crash, data loss, security breach
* **High:** Major functionality not working
* **Medium:** Minor functionality issues
* **Low:** Cosmetic issues, minor improvements

### Defect Lifecycle

1. **New:** Defect reported
2. **Assigned:** Assigned to developer
3. **In Progress:** Being fixed
4. **Fixed:** Fix implemented
5. **Verified:** Fix confirmed by tester
6. **Closed:** Defect resolved

### Defect Tracking

* **Tool:** Issue tracking system
* **Fields:** ID, Title, Description, Priority, Status, Assignee, Reporter
* **Reporting:** Daily defect reports

### Exit Criteria

* All critical and high priority defects are fixed
* 95% of test cases pass
* Performance requirements are met
* Security requirements are satisfied
* User acceptance criteria are met

**Document Version:** 1.0  
**Last Updated:** December 2024  
**Next Review:** January 2025  
**Approved By:** QA Manager  
**Distribution:** Development Team, Project Manager, Stakeholders