

Bug Reporting System Implementation

Overview

Complete bug reporting system for beta testing with user-friendly submission and admin management capabilities.

Implementation Date

January 2, 2026

Features Implemented

1. Database Schema

- **Model:** BugReport
- **Enums:** BugSeverity (LOW, MEDIUM, HIGH), BugStatus (NEW, IN_PROGRESS, RESOLVED, CLOSED)
- **Fields:**
 - User information (userId, userEmail, userName, userType)
 - Bug details (title, description, stepsToReproduce, severity)
 - Technical info (pageUrl, browserInfo, deviceInfo, screenshotUrl)
 - Admin fields (status, assignedTo, adminNotes, resolvedAt)
 - Timestamps (createdAt, updatedAt)

2. Database Migration

- **File:** prisma/migrations/20260102212003_add_bug_report_system/migration.sql
- **Safety:** Uses idempotent patterns with IF NOT EXISTS guards
- **Changes:**
 - Created BugSeverity enum
 - Created BugStatus enum
 - Created BugReport table with proper indexes

3. API Endpoints

POST /api/bug-reports

- **Purpose:** Submit new bug report
- **Access:** All users (logged in or anonymous)
- **Features:**
 - Auto-captures browser and device info
 - Auto-fills user info if logged in
 - Email notification to admin (async)
 - Validates input with Zod schemas

GET /api/bug-reports

- **Purpose:** List all bug reports with filtering

- **Access:** Admin only
- **Features:**
- Filter by status and severity
- Pagination support
- Ordered by creation date (newest first)

GET /api/bug-reports/[id]

- **Purpose:** Get single bug report details
- **Access:** Admin only

PATCH /api/bug-reports/[id]

- **Purpose:** Update bug report status and admin fields
- **Access:** Admin only
- **Features:**
- Update status, assignedTo, adminNotes
- Auto-set resolvedAt when status is RESOLVED or CLOSED

4. Frontend Components

BugReportButton

- **Location:** `src/components/bug-report/BugReportButton.tsx`
- **Features:**
- Fixed position floating button (bottom-right)
- Bug icon with “Report Bug” label
- Hover effects and animations
- Opens BugReportModal on click

BugReportModal

- **Location:** `src/components/bug-report/BugReportModal.tsx`
- **Features:**
- Comprehensive bug report form
- Fields: title, description, steps to reproduce, severity, screenshot URL
- Guest user fields: email, name (if not logged in)
- Auto-capture: pageTitle, browserInfo, deviceInfo
- Form validation
- Success/error states
- Professional design with Headless UI

5. Admin Dashboard

- **Location:** `src/app/admin/bug-reports/page.tsx`
- **Features:**
- Table view of all bug reports
- Filter by status and severity
- Color-coded severity and status badges
- Click to view full details in modal
- Update bug status directly from detail view
- Shows: title, user, severity, status, date, actions

6. Integration

- Added BugReportButton to root layout (`src/app/layout.tsx`)
- Added “Bug Reports” quick action to admin dashboard (`src/app/admin/page.tsx`)
- Available on all pages for all user types

Files Created/Modified

Created Files

1. `prisma/migrations/20260102212003_add_bug_report_system/migration.sql`
2. `src/app/api/bug-reports/route.ts`
3. `src/app/api/bug-reports/[id]/route.ts`
4. `src/components/bug-report/BugReportButton.tsx`
5. `src/components/bug-report/BugReportModal.tsx`
6. `src/app/admin/bug-reports/page.tsx`

Modified Files

1. `prisma/schema.prisma` - Added BugReport model and enums
2. `src/app/layout.tsx` - Added BugReportButton
3. `src/app/admin/page.tsx` - Added Bug Reports quick action

Technical Details

Authentication

- Bug submission: Available to all (logged in users auto-filled, guests provide email/name)
- Admin endpoints: Require ADMIN role

Data Capture

- Automatically captures:
- Current page URL
- Browser user agent
- Screen resolution
- User info (if logged in)

Email Notifications

- Placeholder for email notification to admin (`profyt7@gmail.com`)
- Ready for implementation with nodemailer or email service

UI/UX

- Professional design matching CareLinkAI branding
- Mobile-responsive
- Accessible (keyboard navigation, screen readers)
- Loading states and error handling

Deployment Steps

1. Database Migration

```
# On production (Render):
npx prisma migrate deploy
```

2. Prisma Client Generation

```
# Generate updated Prisma client
npx prisma generate
```

3. Environment Variables

No new environment variables required for basic functionality.

Optional (for email notifications):

- SMTP_HOST
- SMTP_PORT
- SMTP_USER
- SMTP_PASSWORD
- BUG_REPORT_EMAIL (defaults to profyt7@gmail.com)

4. Deploy

```
# Commit changes
git add .
git commit -m "feat: Add comprehensive bug reporting system for beta testing"
git push origin main

# Render will auto-deploy
```

Testing Checklist

User Testing

- [] Floating button visible on all pages
- [] Modal opens with bug report form
- [] Can submit bug report as logged-in user
- [] Can submit bug report as guest
- [] Form validation works
- [] Success message displays
- [] Email is sent to admin

Admin Testing

- [] Admin can access /admin/bug-reports
- [] Bug reports list displays correctly
- [] Can filter by status and severity
- [] Can view bug report details
- [] Can update bug status

- [] Status changes persist

Mobile Testing

- [] Floating button works on mobile
- [] Modal is responsive
- [] Form is usable on mobile
- [] Admin dashboard works on mobile

Future Enhancements

Email Notifications

- Implement nodemailer or email service
- Send detailed email to admin with bug details
- Include link to admin dashboard

File Uploads

- Add screenshot upload functionality
- Store in cloud storage (S3, Cloudinary)

Advanced Features

- Assign bugs to specific team members
- Add comments/discussion thread
- Priority scoring
- Duplicate detection
- Bug analytics dashboard

Success Criteria

- Floating button on all pages for all user types
- Bug report form works for logged-in and guest users
- Data saved to database with proper structure
- Admin dashboard functional with filters
- Can update bug status from admin interface
- Mobile-responsive design
- Professional UI matching CareLinkAI branding
- Ready for beta testing

Support

For issues or questions:

- Email: profyt7@gmail.com
- Admin Dashboard: <https://getcarelinkai.com/admin/bug-reports>

Status: Implementation Complete

Next Steps: Deploy to production and monitor for bug reports during beta testing