

---

**Project Name: Continuity Care**

**Version: 1.0**

**Date: 2025-11-19**

**Author: Shweta**

---

## **1. Purpose**

ContinuityCare is a web platform designed to provide comprehensive information, support services, and assistance for **old-age individuals and people with disabilities**.

The website aims to improve accessibility, ease of service discovery, and communication between caregivers, families, and service providers.

The primary goals:

- Provide clear information on services offered.
- Help users (families, elderly, disabled individuals) connect with support providers.
- Offer appointment/contact request functionality.
- Provide credibility and trust through blogs, testimonials, and organization philosophy.
- Ensure accessibility-friendly UI and responsive design.

---

## **2. Scope**

### **In-Scope**

- Fully responsive website (mobile, tablet, desktop)
- Pages:
  - Home
  - About Us
  - Services (Old Age Care, Disability Support, In-Home Support, NDIS related-if any)
  - Programs & Community Support
  - Contact Page (with form)

- Appointment / Enquiry Form
  - Gallery / Testimonials
  - Blog / News
- Admin Panel (Laravel-based)
  - Manage Services
  - Manage Content (About, Home sections)
  - Manage Enquiries
  - Manage Team Members (optional)
  - Manage Blog posts (if enabled)
- Tracking: Google Analytics & basic SEO setup
- Security, form validation, and spam protection

#### Out of Scope (Initial Release)

- Payment gateway
  - Full caregiving booking system
  - Mobile app
  - Multi-role system (caregiver/patient dashboard)
  - Live chat feature
- 

### 3. Stakeholders

- **End Users:** Elderly individuals, people with disability, families & caregivers.
  - **Admin:** Organization staff managing website content.
  - **Super Admin:** Website owner with full access rights.
  - **Developers & Designers:** Backend (Laravel), frontend, content team.
- 

### 4. Functional Requirements

#### FR-001 – Authentication

- Admin login using email + password
- Password reset via email
- JWT/Session-based authentication in Laravel

#### FR-002 – Content Management

- Admin can manage all content:
  - Home page banners
  - About Us content

- Services details
- Team members / caregivers
- Testimonials
- Gallery
- Blogs/Articles

#### **FR-003 – Services Module**

- User can view detailed service pages
- Admin can:
  - Add service
  - Edit service
  - Upload images
  - Manage service categories

#### **FR-004 – Contact & Enquiry Forms**

- Contact form with:
  - Name
  - Email
  - Phone
  - Service Type
  - Message
- All submissions stored in DB + email sent to admin.

#### **FR-005 – Appointment Request**

- Form with preferred date, time, and service
- Data stored in DB and notified to admin

#### **FR-006 – Accessibility Features**

- Large fonts, easy color contrast
- Alt-text for all images
- Clean minimal UI for elderly usability

#### **FR-007 – Blog/News Module (Optional)**

- Admin can add/modify/delete posts
- User can view posts on frontend

#### **FR-008 – SEO & Performance**

- Meta titles, descriptions for each page

- Optimized images, cache, HTTP compression

#### FR-009 – Security

- CSRF protection
- Validation on server & client
- Secure admin panel
- CAPTCHA on forms (Google reCAPTCHA v2 or v3)

#### FR-010 – Gallery Module

- Admin uploads photos/videos
  - User sees gallery grid
- 

## 5. Non-Functional Requirements

### Performance

- Pages load under 3 seconds
- Optimized images (WebP)

### Security

- Sanitized inputs
- HTTPS everywhere
- Laravel CSRF protection
- Admin routes protected by middleware

### Scalability

- Database organized for easy future expansion
- Ability to add more modules later (caregiver portal, bookings...)

### Usability

- Responsive UI
- Elderly-friendly spacing and fonts
- Simple navigation

### Availability

- 99% uptime expected (shared/managed hosting)

## Compatibility

- Works on Chrome, Safari, Firefox, Edge
  - Fully mobile responsive (Bootstrap 5)
- 

## 6. Suggested Tech Stack

- **Backend:** PHP 8.x, Laravel 12
  - **Frontend:** HTML5, CSS3, Bootstrap 5, JavaScript
  - **Database:** MySQL 5.7 / 8
  - **Email:** PHPMailer / SMTP (Hostinger)
  - **Storage:** Local storage / Cloud if needed
  - **Analytics:** Google Analytics 4
  - **Hosting:** Hostinger / cPanel shared hosting
- 

## 7. System Architecture

### Components:

1. **Frontend Layer**
2. **Laravel Backend API**
3. **Database (MySQL)**
4. **Admin Panel**
5. **Email System (SMTP)**
6. **Cloud / Storage for images**

A simple MVC architecture:

User → Routes → Controllers → Services → Models → Database

---

## 8 . Form Handling & Security

- Server-side validation
  - CSRF protection
  - Google reCAPTCHA for spam protection
  - Email notifications on every enquiry
-

## 9. Testing Strategy

- Browser testing (Chrome, Safari)
- Mobile testing
- Form validation tests
- Admin module functionality tests
- SEO validation

---

## 10. Deployment

- Host on Hostinger as you prefer
- Git-based versioning
- Database import from phpMyAdmin