Project Delivery Report

Live link:-SEALTH | Smart Living. Simple Health.

SEALTH – Holistic Health Platform

Client: Sealth Team
Developer: Divyansh
Delivery Timeline: 2 Weeks (January 2026)
Deployment Status: ✅ Production Ready & Live

---

1. Executive Overview

The SEALTH Holistic Health Platform was delivered as a full-scale, production-grade health-tech solution within a strict two-week timeline. The objective of this engagement was to design, develop, secure, and deploy a scalable multi-role platform capable of supporting diverse stakeholders in the healthcare ecosystem.

The platform successfully delivers:

- A robust **role-based system** with six independent user personas
- **Enterprise-level authentication and security** mechanisms
- **Cloud-native deployment** with serverless scalability
- **Real-time activity monitoring** for administrative oversight

The system is now live, stable, secure, and ready for onboarding users at scale.

---

2. Business Objectives & Scope

2.1 Core Objectives

- Build a unified digital health platform covering end-to-end workflows
- Enable role-specific dashboards for operational clarity
- Ensure production-level security suitable for health-tech use cases
- Deploy a scalable architecture with minimal operational overhead

2.2 In-Scope Deliverables

- Frontend application with premium UI/UX
- Backend APIs with authentication & authorization
- Database schema design and cloud integration

- Logging and monitoring setup
- Production deployment and configuration

---

3. System Architecture Overview

The platform follows a **modern full-stack architecture**:

- **Client Layer:** React + TypeScript SPA
- **API Layer:** Node.js & Express running as Vercel Serverless Functions
- **Data Layer:** MongoDB Atlas (Cloud-hosted)
- **Monitoring Layer:** Google Apps Script connected to Google Sheets

This architecture ensures:

- Horizontal scalability
- Low-latency performance
- Secure data access
- Minimal DevOps complexity

---

4. Development Timeline & Execution

Week 1: Foundation, UI/UX & Core Modules

4.1 Role-Based Architecture

Six distinct user roles were designed and implemented with independent permissions and dashboards:

1. Customer
2. Doctor
3. Admin
4. Health Coach
5. Kitchen Team
6. Delivery Team

Each role has:

- Dedicated routes
- Controlled access permissions
- Customized UI workflows

4.2 UI/UX Design System

- Implemented a **premium design language** using glassmorphism

- Smooth micro-interactions powered by Framer Motion
- Consistent spacing, typography, and color system
- Focus on clarity, accessibility, and modern aesthetics

4.3 Responsive Design

- Fully responsive layouts
- Optimized for mobile, tablet, and desktop
- Cross-browser compatibility ensured

4.4 Core Functional Modules

- Doctor discovery and selection system
- Health metrics tracking and visualization
- Expert communication and messaging interface

Week 2: Backend, Security & Deployment

4.5 Authentication & Authorization

- Secure password hashing using `bcryptjs`
- JWT-based session management
- Protected API routes based on user roles
- Google OAuth integration for seamless social login

4.6 Database Design & Integration

- MongoDB Atlas cloud deployment
- Mongoose schemas designed for:
    - User profiles
    - Health records
    - Appointments and interactions
- Indexed queries for performance optimization

4.7 Activity Logging & Monitoring

- Automated logging using Google Apps Script
- Real-time tracking of:
    - User registrations
    - Login activity
- Admin-accessible Google Sheet for transparency

4.8 Production Deployment

- Full-stack deployment on Vercel
- Backend APIs deployed as serverless functions
- Environment variables securely managed
- Production build optimization completed

---

5. Security Implementation Details

5.1 Data Protection

- All passwords encrypted
- No plaintext credentials stored

5.2 Access Control

- Strict role-based authorization
- Removal of all development shortcuts and mock credentials

5.3 Application Security

- COOP and security headers configured
- Protection against XSS and unauthorized cross-origin access

5.4 Database Network Security

- MongoDB IP whitelisting
- Secure access limited to production environment

---

6. Technology Stack

Frontend

- React.js
- TypeScript
- Vite
- Tailwind CSS
- Framer Motion

Backend

- Node.js
- Express.js (Serverless)

Database

- MongoDB Atlas

Monitoring

- Google Apps Script
- Google Sheets

Hosting

- Vercel

---

7. Quality Assurance & Stability

- Manual functional testing across all roles
- Authentication and authorization validation
- Cross-device UI testing
- Production build verification

---

8. Final Deliverables

1. **Live Production Application**
   https://sealth-psi.vercel.app
2. **Clean & Maintainable Codebase**
3. **Admin Monitoring Google Sheet**
4. **Complete Documentation**
    - README
    - Deployment checklist
    - Architecture overview

---

9. Project Outcome & Status

- All planned features delivered
- No critical or high-priority issues pending
- Platform ready for user onboarding and scale