

# Safha Volunteer Management Platform

## Purpose

This is a role based volunteer management platform/web app with one login and three authenticated portals: Volunteer, Organization, Admin. Users sign in once, and their access is determined by role. Admins can additionally switch views to see what volunteers and organizations see. Members of organizations can also switch views to see what a volunteer sees. Organization accounts should have a main organization email that can assign co-owners (who have access to all organization account capabilities) and Org Members (who can access backend information, but can't make large changes. Especially to the volunteer lists and opportunities)

## SUMMARY

### Core Roles

- Volunteer (default for all users)
- Organization (view-switching)
- Admin (full access + view-switching)

### Volunteer Portal (core user experience)

- Browse volunteer opportunities
- Filters: category, date, location, org + keyword search
- Register in one click (supports shifts + waitlists)
- Check-in / check-out methods: QR, numeric code, geofence, manual
- Automatic hours calculation
- Dashboard showing hours, impact value, upcoming events, awards
- Profile: skills, interests, availability, emergency contact, certifications
- Post-service reflections
- Training module with quizzes (GUIDE framework)
- Awards, badges, downloadable certificates
- Volunteer teams (create/join via 6-char code, team hours tracking)

### Organization Portal

- Create/manage volunteer opportunities
  - Draft / publish / cancel
  - Public or private visibility
  - Recurring + multi-shift support
  - Waitlists + templates
- Manage volunteers tagged to their organizations and uploaded by organization
- View registrants, profiles, history, hours
- Manual check-in/out and hour edits
- Tag volunteers to org
- Communication tools

- Bulk email volunteers
- Custom email templates (welcome, reminder, thank you)
- Settings
  - Org profile, logo, users, waivers
  - Email templates
- Enable in-kind donation tracking
- Request reports from admins

### **Admin Portal**

- Full platform access
- Manage all users and orgs
- Assign roles (volunteer / org / admin)
- View/edit all opportunities, check-ins, hours
- View and moderate reflections and testimonials
- Awards and certificate management
- Training module creation and tracking
- Platform-wide analytics and reports
- Mass email exports (BCC style)
- Incident reports, corporate programs
- Demo account management (3 reusable demo accounts)

### **Key Data Concepts**

- Users + roles
- Volunteer profiles
  - Training modules + completion
- Organizations
  - Waivers + minor consent
- Opportunities (with shifts, visibility, recurrence)
- Registrations + check-ins (hours tracked)
- Reflections
- Awards and milestones
- Teams
- Incident reports
- In-kind donations
- Demo accounts

### **Automation**

- Welcome email on signup
- Birthday emails
- Opportunity reminders (24h before)
- Thank-you email after check-out (different email depending on pre or post-reflection)
- Admin daily morning + evening reports
- Automatic milestone awards (10, 25, 50, 100, 250, 500 hrs)
- GUIDE certification when training is complete

## **Key Rules**

- Volunteers only see their own data
- Orgs only see their own data + connected volunteers
- Admins see everything
- Private opportunities only visible to invited volunteers
- Impact value = hours × 31.80
- Hours auto-calculated but editable by org/admin

## **Tech Direction (flexible)**

- Modern React web app
- Role-based routing and navigation
- Backend (serverless?) with scheduled jobs
- Email service integration
- Secure auth + RBAC

# **DETAILED SPEC SHEET PULLED DIRECTLY FROM VIBE CODER**

## **1. Portals and Access Model**

### **1.1 Roles**

Roles are:

- volunteer (default for all users)
- organization
- admin

Role logic:

- Every authenticated user is treated as a volunteer unless explicitly assigned another role via UserRole.
- organization and admin roles unlock additional portals and navigation.
- Admins can switch view mode between admin, organization, and volunteer perspectives.

### **1.2 Core access rules**

- Volunteers can only view and manage their own data.
- Organizations can only view their own organization record, their opportunities, and volunteers connected to them (registrants and or tagged volunteers).
- Admins have platform wide access to all entities and management views.
- Private opportunities are only visible to invited volunteers.

## **2. Pages and Navigation**

## **2.1 Volunteer Portal pages**

### Dashboard and navigation

- VolunteerDashboard: main dashboard with stats and summaries
- VolunteerProfile: profile management
- Contact: contact Safha form

### Opportunities

- BrowseOpportunities: browse, search, filter opportunities
- MyOpportunities: view registrations (upcoming, past, cancelled)
- AIRecommendations: AI powered opportunity matching based on profile

### Engagement

- MyReflections: submit and view post service reflections
- MyAwards: view awards, badges, certificates
- GUIDETraining: training modules with quizzes

### Teams

- VolunteerTeams: create and join teams

## **2.2 Organization Portal pages**

### Core management

- OrganizationDashboard: overview and stats
- ManageOpportunities: create and edit opportunities, templates, publishing controls
- OrganizationVolunteers: volunteer database view for the org

### Tools and settings

- OrganizationSettings: org profile, waiver settings, email template settings, users
- OrganizationReports: request reports from admins
- OrganizationAITools: AI email generation and opportunity optimization tools

## **2.3 Admin Portal pages**

### User management

- AdminDashboard: overview
- AdminVolunteers: all volunteers
- AdminUsers: user management
- AdminUserRoles: role assignment

### Organization management

- AdminOrganizations: organization oversight
- AdminOpportunities: all opportunities
- AdminCheckIns: check in management

### Content and communication

- AdminReflections: view reflections and testimonial controls
- AdminMassEmail: email list export and mass communication tooling
- AdminAwards: award management
- TestEmails: email testing tool

### Tools and reports

- AdminReports: analytics and reporting
- AdminAITools: AI features

- AdminVolunteerTools: volunteer tooling
- AdminDemoManager: demo account management

#### Special programs

- AdminCorporatePrograms: corporate programs
- AdminIncidentReports: incidents

## 2.4 Public and special pages

#### Public access

- Index: landing and login page
- PublicPortal: public opportunity browsing
- Home: home page

#### Demo and setup

- DemoSetup: demo account setup flow
- DemoOrganization: demo org page
- InviteDemoAdmin: invite demo admin

#### Documentation pages

- PlatformSpec: feature overview
- ComprehensiveSpec: this spec display page in the prototype

## 3. Feature Requirements by Portal

### 3.1 Volunteer Portal features

#### Opportunity discovery

- Browse all published opportunities.
- Filters: category, date, location, organization.
- Keyword search.
- AI powered matching based on volunteer profile.
- Public portal supports browsing without authentication.
- Supports shift based opportunities.
- Supports waitlist joining when full.

#### Registration and participation

- One click registration.
- My opportunities view splits into upcoming, past, cancelled.
- Check in and check out methods supported:
  - numeric code
  - QR code
  - geofence
  - manual
- Hours served should be automatically calculated.
- Volunteer cancellation flow includes a reason.
- For multi shift opportunities, volunteer can choose a shift.

#### Profile and preferences

- Manage skills and interests.
- Bio and profile photo.
- Availability schedule with available and unavailable blocks.

- Emergency contact info.
- Certification uploads (examples: background check, CPR).
- Location and demographic info.
- Organization tagging preferences.

#### Engagement and recognition

- Submit post service reflections.
- View all submitted reflections and impact stories.
- View awards and badges.
- Milestones: 10, 25, 50, 100 plus hours (and higher listed later).
- Downloadable certificates.
- Total hours and opportunity count.
- GUIDE training modules with quizzes.
- Training completion certification.

#### Team features

- Create volunteer teams.
- Join via 6 character join code.
- Team leaderboard and rankings.
- Team hours tracking.
- Team types: family, friends, corporate, youth group, school, community, other.

#### Dashboard and analytics

- Total hours served.
- Completed opportunities count.
- Impact value calculation at 31.80 per hour.
- Upcoming registrations module.
- Recent reflections module.
- Award showcase module.

### **3.2 Organization Portal features**

#### Opportunity management

- Create opportunities, single or recurring.
- Status lifecycle: draft, published, completed, cancelled.
- Visibility: public, sign in required, private.
- Multi shift support.
- Skills requirements.
- Choose check in method: QR, numeric code, geofence, manual.
- Spots available, plus waitlist management when enabled.
- Save opportunities as templates.
- Create new opportunities from templates.
- Private invitations to tagged volunteers.

#### Volunteer management

- View all registered volunteers for the org.
- Access volunteer profiles and skills.
- View volunteer history and hours.
- Tag volunteers to the organization.
- Search volunteers by name or email.

- Bulk email selected volunteers.
- View volunteer check in history.
- Manual check in and check out.
- Edit hours served.

#### Communication tools

- AI email generator for outreach.
- Email templates: welcome, reminder, thank you, update.
- Bulk email registrants.
- Custom email template editor.
- Email preview before sending.
- Recipient list export.

#### AI tools

- AI email generation.
- Opportunity time optimizer.
- Volunteer availability analysis.
- Suggested optimal time slots.
- Support custom instructions.

#### Settings and configuration

- Organization profile editing.
- Logo upload.
- Contact information management.
- Organization type selection.
- Co owner management.
- Authorized user management.
- Custom waiver text.
- Email template customization.
- In kind donations enable toggle.

#### Reports and analytics

- Organizations can request custom reports from admins.
- Volunteer hours summaries.
- Attendance rates.
- Registration statistics.

### **3.3 Admin Portal features**

#### User management

- View all volunteers, including users without profiles.
- View all organizations.
- Assign user roles (admin and organization).
- Edit user information.
- Add admin notes to users.
- View user activity and history.
- Merge duplicate profiles.
- Certification verification workflows.

#### Content management

- View all opportunities across all orgs.

- Edit any opportunity.
- Cancel opportunities.
- View all check ins.
- Edit hours served.
- Flag check ins for review.
- Verify check ins.
- View all reflections.
- Testimonial management.

#### Recognition system

- Create and assign awards.
- Milestone badges auto or manual.
- Certificate generation.
- Custom award types.
- Award history.

#### Training management

- Create training modules.
- Upload training videos.
- Create quiz questions.
- Set passing scores.
- Track completion rates.
- Activate deactivate modules.

#### Communication

- Mass email list export for BCC use.
- Email by recipient type (all volunteers, by org, by region).
- Select individual volunteers or orgs.
- Email composer for drafts.
- Email testing tools.
- Monitor automated emails.

#### Reports and analytics

- Platform wide stats.
- Engagement metrics.
- Org performance.
- Total hours served.
- Impact value calculations.
- Registration trends.
- Custom report generation.
- Daily beginning of day and end of day reports.

#### Demo management

- Create demo accounts 1, 2, 3.
- Demo setup wizard.
- Populate example data.
- Track demo status.
- Track current user using demo.
- Demo reset functionality.



#### Special programs

- Corporate program management including verification workflows.
- Incident report viewing and status updates.
- In kind donation tracking.

#### AI and automation

- Platform wide AI tools.
- Email automation monitoring.
- Automated report scheduling.
- Birthday automation.

---

## 4. Data Model and Entity Schemas

Note: The prototype references Base44, where User is a built in auth entity. If you are rebuilding this outside Base44, you still need equivalent tables and access control.

### 4.1 Core user entities

#### User (built in)

- id
- email
- full\_name
- role
- created\_date
- profile\_image\_url

Security note: admins can manage all users. Regular users only see themselves.

#### UserRole

Purpose: assigns volunteer, organization, admin to a user email.

#### Fields:

- user\_email: string required
- role: enum volunteer, organization, admin required
- notes: string optional

Rule: default role is volunteer unless a UserRole record exists.

#### VolunteerProfile

Purpose: extended volunteer details and preferences.

#### Fields:

- user\_email: string required
- interests: array of string
- skills: array of string
- bio: string
- date\_of\_birth: date
- gender: enum male, female, non-binary, prefer\_not\_to\_say, other
- location: string
- availability\_type: enum available, unavailable
- availability\_schedule: array of objects
- emergency\_contact\_name: string
- emergency\_contact\_phone: string

- certifications: array of objects
- available\_to\_all\_orgs: boolean
- admin\_notes: string

## Organization

### Fields:

- user\_email: string required (primary account)
- owner\_emails: array of string (co owners)
- organization\_name: string required
- logo\_url: string
- description: string
- address: string
- contact\_person: string
- contact\_phone: string
- website: string
- organization\_type: enum islamic\_center\_masjid, school, business, nonprofit, community\_org, other
- email\_template\_reminder\_subject: string
- email\_template\_reminder\_body: string
- email\_template\_thankyou\_subject: string
- email\_template\_thankyou\_body: string
- authorized\_users: array of string
- custom\_waiver\_text: string
- requires\_waiver: boolean
- in\_kind\_donations\_enabled: boolean
- admin\_notes: string

## 4.2 Opportunity management entities

### Opportunity

#### Fields:

- organization\_id: string required
- title: string required
- description: string
- category: enum education, events, youth, admin, maintenance, outreach, fundraising, other
- date: date required
- start\_time: string required
- end\_time: string required
- location: string
- spots\_available: number
- skills\_needed: array of string
- status: enum draft, published, completed, cancelled
- visibility: enum public, sign\_in\_required, private required
- is\_recurring: boolean
- recurrence\_pattern: enum daily, weekly, biweekly, monthly, custom
- check\_in\_method: enum qr\_code, numeric\_code, geofence, manual

- check\_in\_code: string
- has\_shifts: boolean
- shifts: array of objects
- allow\_waitlist: boolean

#### OpportunityTemplate

Purpose: reusable templates for recurring opportunities.

Schema: same as Opportunity plus template\_name field.

#### Registration

Fields:

- volunteer\_email: string required
- opportunity\_id: string required
- shift\_id: string optional
- status: enum registered, waitlist, checked\_in, checked\_out, no\_show, cancelled
- waitlist\_position: number optional
- notes: string optional

#### CheckIn

Fields:

- registration\_id: string required
- volunteer\_email: string required
- opportunity\_id: string required
- check\_in\_time: datetime required
- check\_out\_time: datetime optional
- hours\_served: number calculated
- notes: string
- check\_in\_method: enum manual, qr\_code, code, self
- verified\_by\_admin: boolean
- flagged\_for\_review: boolean
- flag\_reason: string

#### OpportunityInvitation

Purpose: track invitations sent for private opportunities.

Fields: not fully specified, but must support invitation tracking by volunteer and opportunity.

#### OrganizationTag

Purpose: association between a volunteer and an organization, including how the link was created.

Fields:

- volunteer\_email: string required
- organization\_id: string required
- tagged\_by: enum volunteer, admin, organization
- status: enum active, inactive

### 4.3 Engagement and recognition entities

#### Reflection

Fields:

- volunteer\_email: string required
- opportunity\_id: string required

- check\_in\_id: string optional
- reflection\_text: string required
- what\_learned: string
- impact\_felt: string
- would\_recommend: boolean
- allow\_testimonial: boolean

Award

Fields:

- volunteer\_email: string required
- award\_type: enum certificate, badge, recognition, milestone
- title: string required
- description: string
- issued\_by: string required
- issued\_by\_name: string
- hours\_milestone: number
- certificate\_url: string
- icon: string

TrainingModule

Fields:

- title: string required
- description: string
- video\_url: string required
- order: number required
- quiz\_questions: array of objects
- passing\_score: number default 80
- is\_active: boolean

ModuleCompletion

Fields:

- volunteer\_email: string required
- module\_id: string required
- quiz\_score: number
- passed: boolean
- attempts: number
- completed\_date: datetime

## 4.4 Teams and corporate entities

VolunteerTeam

Fields:

- team\_name: string required
- description: string
- created\_by: string required
- join\_code: string required, unique 6 characters
- team\_image\_url: string
- is\_private: boolean
- team\_type: enum family, friends, corporate, youth\_group, school, community, other

- status: enum active, inactive

TeamMember

Fields:

- team\_id: string required
- volunteer\_email: string required
- role: enum creator, member
- joined\_date: datetime
- status: enum active, left

CorporateProgram

Fields:

- volunteer\_email: string required
- company\_name: string required
- employee\_id: string
- program\_type: enum matching, volunteer\_time\_off, team\_building, skilled\_volunteering, other
- matching\_eligible: boolean
- matching\_ratio: string
- vto\_hours: number
- verification\_status: enum pending, verified, rejected

## 4.5 Compliance and safety entities

Waiver

Fields:

- volunteer\_email: string required
- organization\_id: string optional
- waiver\_type: enum general, organization\_specific, opportunity\_specific
- waiver\_text: string required
- signature\_data: string required, base64
- signed\_date: datetime required
- parent\_signature: boolean
- parent\_email: string

MinorConsent

Fields:

- minor\_email: string required
- minor\_name: string required
- minor\_date\_of\_birth: date required
- parent\_name: string required
- parent\_email: string required
- consent\_text: string required
- signature\_data: string required
- medical\_conditions: string
- expiration\_date: date

Incident

Fields:

- opportunity\_id: string required

- reported\_by: string required
- incident\_type: enum injury, property\_damage, safety\_violation, harassment, theft, other
- severity: enum minor, moderate, severe, critical
- incident\_date: datetime required
- description: string required
- people\_involved: array of string
- witnesses: array of string
- medical\_attention\_needed: boolean
- authorities\_contacted: boolean
- status: enum reported, under\_review, resolved, closed

## 4.6 Additional systems

InKindDonation

Fields:

- organization\_id: string required
- donor\_name: string required
- donation\_type: enum goods, services, equipment, food, supplies, professional\_services, other
- description: string required
- quantity: number
- estimated\_value: number
- donation\_date: date required
- receipt\_sent: boolean

ReportRequest

Purpose: track report generation requests from organizations to admins.

Fields: not fully specified, but must support request submission and admin tracking.

DemoAccount

Fields:

- demo\_number: enum 1, 2, 3
- demo\_email: string required
- password: string required
- status: enum available, in\_use
- current\_user\_name: string
- setup\_completed: boolean
- using\_example\_data: boolean

---

## 5. Backend Functions and APIs

The prototype specifies serverless functions with Deno Deploy. If you are rebuilding elsewhere, replicate endpoints and schedules.

### 5.1 Email automation functions

- autoSendWelcomeEmail
  - Trigger: new user registration
  - Sends: welcome message and getting started guide
- sendWelcomeEmail

- Manual trigger endpoint (GET or POST)
- checkBirthdays
  - Daily job sends birthday emails to volunteers
- sendOpportunityReminders
  - Sends reminder emails for opportunities happening tomorrow
- sendThankYouEmails
  - Sends thank you email after volunteer check out
  - Condition: only if no reflection has been submitted yet
- sendAdminBODReport
  - Beginning of day admin email report
- sendAdminEODReport
  - End of day admin email report
- testEmailDelivery
  - Test endpoint for email system verification

Monitoring rule:

- All automated emails send a copy to [asser@safha.org](mailto:asser@safha.org) for monitoring.

## 5.2 System functions

- autoAssignVolunteerRole
  - Ensures new users get volunteer role by default.
- getPublicOpportunities
  - Public API endpoint for the public opportunity portal.
  - No authentication required.

## 5.3 Function architecture expectations

- Functions are wrapped with `Deno.serve`.
- Initialize SDK client from request.
- Use service role for admin level operations.
- Scheduled functions run via cron jobs configured outside the code.
- Email monitoring copies go to [asser@safha.org](mailto:asser@safha.org).

---

# 6. Automated System Rules

## 6.1 Email automation schedule and triggers

1. Welcome email
  - Trigger: immediately on new registration
2. Birthday emails
  - Trigger: daily at 9 AM
3. Opportunity reminders
  - Trigger: daily at 6 PM for opportunities happening in the next 24 hours
4. Thank you emails
  - Trigger: after volunteer checks out, within 24 hours
  - Condition: only if volunteer has not submitted a reflection
5. Admin morning report
  - Trigger: daily at 7 AM
  - Recipient: [info@safha.org](mailto:info@safha.org)

- Copy: asser@safha.org
- Contents: today's opportunities, birthdays, expected volunteers, platform stats
- 6. Admin evening report
  - Trigger: daily at 11 PM
  - Recipient: info@safha.org
  - Copy: asser@safha.org
  - Contents: active volunteers, hours served, impact value, new signups, opportunities added or edited

## 6.2 Other automation rules

- Auto role assignment: all new users default to volunteer role.
- Hours calculation: automatically computed at check out.
- Milestone awards: automatic at 10, 25, 50, 100, 250, 500 hour milestones.
- GUIDE certification: automatically grant a GUIDE certified badge after all training modules are completed.
- Registration status updates:
  - checked\_in and checked\_out update based on check in actions.
- Waitlist promotion:
  - automatic notification when a spot opens, if implemented.

---

## 7. Technical Architecture Targets

This section describes the prototype stack. If you are rebuilding the app, treat it as guidance for equivalent implementation.

### 7.1 Frontend technology expectations

- React 18
- React Router DOM for routing
- Tailwind CSS for styling
- TypeScript for type safety

UI and interaction libraries

- shadcn ui components
- Radix UI primitives
- Lucide React icons
- Framer Motion animations

Data and forms

- TanStack Query for server state
- React Hook Form for forms
- date-fns for date handling
- lodash for utilities

Additional libraries

- react-markdown
- react-quill rich text editor
- recharts for charts
- sonner for toast notifications

### 7.2 Backend infrastructure expectations



- Auth and user management
- Database for entities above
- File storage for public and private uploads
- Real time subscriptions if desired
- Role based permissions enforcement

Serverless functions

- Scheduled jobs
- Environment variable support

### **7.3 External integrations**

Email service

- Resend integration
- From name customizable
- HTML email support
- Requires secret RESEND\_API\_KEY

AI service

- LLM invocation integration for text generation and structured JSON output
- Supports custom instructions
- Can attach file URLs for context
- Internet search available in the prototype integration

File management

- Public uploads
- Private uploads
- Signed URL generation for temporary access
- Extraction parsing for uploaded files (CSV, PDF, images)

Image generation

- Text prompt based generation
- Returns URL

---

## **8. Security and Access Control Requirements**

Authentication

- Email based authentication
- JWT style session handling
- Password reset flow
- Email verification

Role based access control

- Default volunteer role for all users
- Organization role allows managing their org assets
- Admin role has full visibility
- Admin role can switch view mode into volunteer or org experience

Data isolation

- Volunteer sees own records only.
- Org sees own org records and connected volunteers only.
- Admin sees everything.

- Private opportunities visible only to invited volunteers.

#### Special user management

- User entity is admin manageable.
- Invitation system supports inviting new users.
- Demo accounts have special handling: setup wizard, isolation, reset.

---

## 9. Component and App Structure Expectations

#### Layout system

- One main authenticated layout wrapper used across portals.
- Responsive navigation: sidebar on desktop, hamburger on mobile.
- Navigation changes based on role.
- Admin and org users support view mode switching.
- Demo mode banner when applicable.

#### Reusable components referenced in prototype

- OpportunityForm: opportunity create and edit form
- VolunteerListModal: volunteer list display
- DemoAccountPopup: demo notifications
- PWAInstallPrompt: optional PWA prompt
- NotificationManager: toast management
- MobileBrowserPrompt: compatibility prompt
- UI components library folder

#### Utility functions

- createPageUrl(pageName) used to generate URLs
- Custom hooks for data fetching
- Date and time formatting utilities

---

## 10. Key Calculations and Constants

#### Impact value

- Impact dollars shown as  $\text{total\_hours\_served} * 31.80$

#### Hours served

- Computed on check out using check out time minus check in time
- Admin and organization users can edit hours served manually