

# VolunteerHub Project Documentation

## Project Overview

VolunteerHub is a web-based platform designed to connect volunteers with meaningful opportunities while providing organizers tools to create and manage volunteer events. The application promotes community engagement through an intuitive interface and secure functionalities.

## User Requirements

### For Volunteers

1. **User Registration & Authentication:**
  - a. Ability to create an account securely with role-based access.
  - b. Login functionality using email and password.
  - c. Option to reset password.
2. **Event Discovery:**
  - a. Browse and search volunteer opportunities by category, location, and date.
  - b. Filter events based on interests.
3. **Event Registration:**
  - a. Sign up for events with real-time availability tracking.
  - b. Ability to unregister from events.
4. **Profile Management:**
  - a. Customize profiles with interests, phone number, and location.
  - b. View and update volunteer hours and registered events.
5. **Messaging System:**
  - a. Communicate directly with event organizers.
  - b. Receive notifications for messages and event updates.

### For Organizers

1. **Event Creation & Management:**
  - a. Create detailed volunteer events with requirements and capacity limits.
  - b. Edit and update existing events.
  - c. Delete events as needed.
2. **Volunteer Management:**

- a. View registered volunteers and their contact information.
  - b. Manage volunteer registrations.
3. **Dashboard Analytics:**
  - a. Track event performance and volunteer engagement statistics.
  - b. Monitor volunteer hours logged for events.
4. **Profile Management:**
  - a. Manage organization profiles and event categories.
  - b. Update contact information.

## Technology Stack

- **Frontend:** HTML5, CSS3, JavaScript (ES6+)
- **Backend:** PHP 7.4+, MySQL
- **Development Environment:** XAMPP

## Installation Guide

### Prerequisites

- Install **XAMPP** from [Apache Friends](#).
- Start Apache and MySQL services.

### Setup Steps

1. **Clone the repository:**

```
git clone <repository-url>
```

2. **Place the project in the htdocs directory** of your XAMPP installation.
3. **Database Setup:**
  - a. Open phpMyAdmin (<http://localhost/phpmyadmin>).
  - b. Import the database schema from `database/volunteer_hub.sql`.
4. **Configuration:**
  - a. Update database credentials in `api/config.php`.
5. **Access the Application:**
  - a. Open [http://localhost/volunteerHub\\_pj](http://localhost/volunteerHub_pj) in your web browser.

# API Documentation

## Base URL

[http://localhost/volunteerHub\\_pj/api/](http://localhost/volunteerHub_pj/api/)

## Authentication

### Register User

POST /api/auth.php

Content-Type: application/json

```
{
  "action": "register",
  "name": "John Doe",
  "email": "john@example.com",
  "password": "securepassword",
  "role": "volunteer",
  "phone": "555-0123",
  "location": "Los Angeles, CA"
}
```

### Login User

POST /api/auth.php

Content-Type: application/json

```
{
  "action": "login",
  "email": "john@example.com",
  "password": "securepassword"
}
```

### *Token Verification*

GET /api/auth.php?verify\_token&token=jwt\_token\_here

## Events Management

### Create Event

POST /api/events.php

Authorization: Bearer <jwt\_token>

Content-Type: application/json

```
{
  "title": "Beach Cleanup Drive",
  "category": "Environment",
  "date": "2025-03-15",
  "time": "09:00",
  "location": "Santa Monica Beach",
  "description": "Join us for a community beach cleanup...",
  "maxVolunteers": 50,
  "organizerId": 1
}
```

### Get All Events

GET /api/events.php

### Register for Event

POST /api/events.php?action=register

Authorization: Bearer <jwt\_token>

Content-Type: application/json

```
{
  "eventId": 1,
  "volunteerId": 2
}
```

## Example Usage (JavaScript)

// Example: User Login

```
const loginResponse = await fetch('/api/auth.php', {
  method: 'POST',
  headers: {
```

```
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    action: 'login',
    email: 'user@example.com',
    password: 'password'
  })
});
const loginData = await loginResponse.json();
const token = loginData.access_token;

// Example: Create Event
const eventResponse = await fetch('/api/events.php', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    'Authorization': `Bearer ${token}`
  },
  body: JSON.stringify({
    title: 'New Volunteer Event',
    category: 'Community',
    date: '2025-04-10',
    time: '10:00',
    location: 'City Park',
    description: 'Join us for a day of community service!',
    maxVolunteers: 30,
    organizerId: 1
  })
});
const eventData = await eventResponse.json();
```

## Conclusion

VolunteerHub is designed to empower both volunteers and organizers, fostering a vibrant community through effective volunteer management. For further details, please refer to the project's GitHub repository or contact the development team for support.