Volunteer Management Project Documentation

1. Abstract

The Volunteer Management Project is designed to facilitate seamless coordination between event organizers,
coordinators, and volunteers. It focuses on automating the management of events, tasks, and feedback,
ensuring efficient allocation of resources and improved communication. The project leverages a robust
database system to store and manage data securely, alongside user-friendly interfaces for interaction.

2. Introduction

Volunteer Management Systems play a crucial role in organizing large-scale events efficiently.
This project aims to create a comprehensive solution for managing volunteers, events, and related tasks.
The system ensures that event organizers can streamline workflows while volunteers experience a hassle-free
engagement process.

3. List of Entities

1. Volunteer

2. Event

3. Task

4. Coordinator

5. Registration

6. Feedback

7. Donation

8. Skill

4. List of Attributes of Each Entity

- Volunteer: VolunteerID (PK), Name, Email, Phone Number, Address, Skills, Availability, Status.

- Event: EventID (PK), Name, Description, Date, Time, Location, OrganizerID, Required Volunteers, Status.

- Task: TaskID (PK), EventID, Name, Description, Assigned Volunteers, Deadline, Priority Level, Status.

- Coordinator: CoordinatorID (PK), Name, Email, Phone Number, Role.

- Registration: RegistrationID (PK), VolunteerID (FK), EventID (FK), Registration Date, Status.

- Feedback: FeedbackID (PK), VolunteerID (FK), EventID (FK), Comments, Rating.

- Donation: DonationID (PK), EventID (FK), Amount, Donor Details.

- Skill: SkillID (PK), Name, Description.

5. Relationship Between Entities

- Volunteer and Event: Many-to-Many (through Registration table).

- Volunteer and Task: Many-to-Many (through Schedule table).

- Event and Task: One-to-Many.

- Event and Coordinator: Many-to-One.

- Volunteer and Feedback: One-to-Many.

- Event and Feedback: One-to-Many.

- Volunteer and Skill: Many-to-Many (through VolunteerSkill table).

- Event and Donation: One-to-Many.

6. E-R Diagram

Note: The E-R Diagram visually represents the relationships between the entities.

(Insert E-R Diagram here if creating an actual diagram.)

7. Database with Constraints

- Primary Keys: Ensure unique identification of records.

- Foreign Keys: Maintain referential integrity.

- Unique Constraints: Ensure fields like Email and Phone Number are unique.

- ENUM Constraints: Enforce predefined values (e.g., Status).

- CHECK Constraints: Validate data input (e.g., Non-negative values for Donations).

- ON DELETE CASCADE: Automatically delete related records on deletion.

8. Work Distribution Chart

Task/Module: **Member Responsible:** **Role/Responsibilities:** **Timeline:**

Database Design Member 1 (e.g., Alice) Define entities, create ER diagram, and set constraints. Week 1

Backend Development Member 2 (e.g., Bob) Develop APIs, implement user/event management. Week 2-3

Frontend Development Member 3 (e.g., Carol) Design UI for volunteers and coordinators. Week 2-3

Project Documentation Member 4 (e.g., Dave) Create database schema and project summary. Week 3

Integration and Testing All Members Integrate modules and conduct testing. Week 4

Presentation and Deployment All Members Final presentation and application deployment. Week 5