**Computer Science Project**

**Deliverable 1 (Started -> 14/08/2025):**

**Investigating (14/08/2025):**

**Article 1: *Volunteer Management For Nonprofits: Strategies, Challenges, and Best Practices***

* Article basically outlines why effective volunteer management is crucial, and shares the practical strategies for building sustainable programs

**Importance of Community Engagement:**

* Volunteers are needed, as they expand the nonprofits impact, boosts community service, and enhances credibility, all while keeping the costs of operation lower
* Effective program motivates volunteers, improves outcomes, and may help reduce staff turnover
* If volunteers’ skills & interests aligned with needs of organisation, it increases their productivity and fosters more volunteer engagement, also benefiting the volunteer themselves by allowing them to gain real-world experience

**Common Challenges:**

* Nonprofit organisations often face staff, budget, and time constraints associated with effective volunteer management
  + *Solution:* Investing in a volunteer management system can help to optimise these available resources
* Nonprofits can often get unorganised, and volunteers are not too sure what they are signing up for
  + *Solution:* Explain the organisation’s mission, and explain how volunteers can contribute to achieving that mission
* Establish roles, responsibilities, expectations for volunteers, and set measurable goals
* Non-profit organisations often have trouble attracting volunteers, and often lack positions that a diverse range of people can volunteer in
  + *Solution:* Create engaging volunteer opportunities that match the needs of potential volunteers
* Use a variety of channels (E.g. Social media, Local partnerships & events) to attract volunteers

**Reference:**

* NGOFeed. (2025). Volunteer Management for Nonprofits: Strategies, Challenges, and Best Practices. [online] Available at: <https://ngofeed.com/blog/volunteer-management-for-nonprofit/>[Accessed 14 Aug. 2025].

**Article 2: *Overcoming Challenges in Volunteer Management***

* Article basically provides practical advice on how to tackle the common volunteer management challenges effectively

**Importance of Community Engagement:**

* Strengthens social bonds, where the community engagement fosters trust, cooperation, and relationships between volunteers, organisations, and the people
* Boosts volunteer retention, where volunteers feel like they’re a part of the community, and see the real-world impacts of their effects, encouraging them to stay long-term
* Communities that are engaged with organisations more likely to donate, advocate, and recruit others to join

**Common Challenges:**

* Scheduling conflicts, where everyone has different availabilities, hence making it “a nightmare trying to coordinate”
  + *Solution:* Using a scheduling tool, from simple spreadsheets to more advanced software, find something that works for your team
* Encourage colleagues to be transparent about time commitments, so the volunteers know what’s expected
* Dealing with burnout, where volunteers can get easily overwhelmed, especially if they are taking on too much or if they are simply new to the whole system
  + *Solution:* Set clear role boundaries, and encourage breaks and self-care
* Respect volunteers’ limits and avoid overloading them
* Communication gaps, where volunteers are often don’t communicate as much as they need to, which leads to an unorganised structure, and a broken flow
  + *Solution:* Be open to using multiple channels (E.g. Email, Text, Social Media)
* Send regular updates to volunteers to stay informed and connected

**Key Takeaways:**

* Volunteers boost nonprofits impact, community service, and credibility, while also lowering operational costs
* It also brings communities closer together, and strengthens social bonds
* Some issues related to volunteer management include volunteer burnout, communication gaps, scheduling conflicts, organisation, attracting volunteers, and money

**Deconstruction (14/08/2025):**

* Will have the four main tables: Organisations, Volunteers, Events, Skills
  + But these all have many-to-many relationships (Except organisation -> event, which is one-to-one), and hence they need junction tables to resolve the many-to-many relationships
* Will be listing these in relational notation (I didn’t know what this meant in the test 🙁)
* Need a user table to login to the account (Mentioned in-class today);

**User**(user\_id, email, phone\_number, password\_hash, phone\_number, role, created\_at)

**Volunteer**(volunteer\_id, *user\_id*, first\_name, last\_name, availability)

**Organisation**(organisation\_id, *user\_id*, name, description, address, website\_url)

**Event**(event\_id, *organisation\_id*, title, description, event\_date, location, max\_volunteers)

**Skills**(skill\_id, name, description)

**Relationships:**

* Organisation and Event
  + Is a one-to-many relationship (One organisation can host many events, but every event belongs to exactly one organisation, no collaborated organisations or anything)
  + No junction table or anything needed
* Event and Volunteer
  + Is a many-to-many relationship (Volunteers can register for multiple events, and events can have multiple volunteers)
  + Requires junction table: Volunteer\_Event
  + Can also have the signup date

**Volunteer\_Event**(*volunteer\_id*, *event\_id*, signup\_date) (Composite primary key)

* Volunteer and Skill
  + Is a many-to-many relationship (Volunteers can have many skills, each skill can belong to multiple volunteers)
  + Requires junction table: Volunteer\_Skill
  + Should also have the proficiency of the volunteer in the skill

**Volunteer\_Skill**(*volunteer\_id*, *skill\_id*, proficiency\_level) (Composite primary key)

* Event and Skill:
  + Is the skills required by volunteers for every event
  + Is a many-to-many relationship (An event can require multiple skills, a skill can be required by multiple events)
  + Requires junction table: Event\_Skill

**Event\_Skill**(*event\_id*, *skill\_id*) (Composite primary key)

**Final List of Tables:**

**User**(user\_id, email, password\_hash, phone\_number, role, created\_at)

**Volunteer**(volunteer\_id, *user\_id*, first\_name, last\_name, availability, location)

**Organisation**(organisation\_id, *user\_id*, name, description, address, website\_url)

**Event**(event\_id, *organisation\_id*, title, description, event\_date, location, max\_volunteers)

**Skills**(skill\_id, name, description)

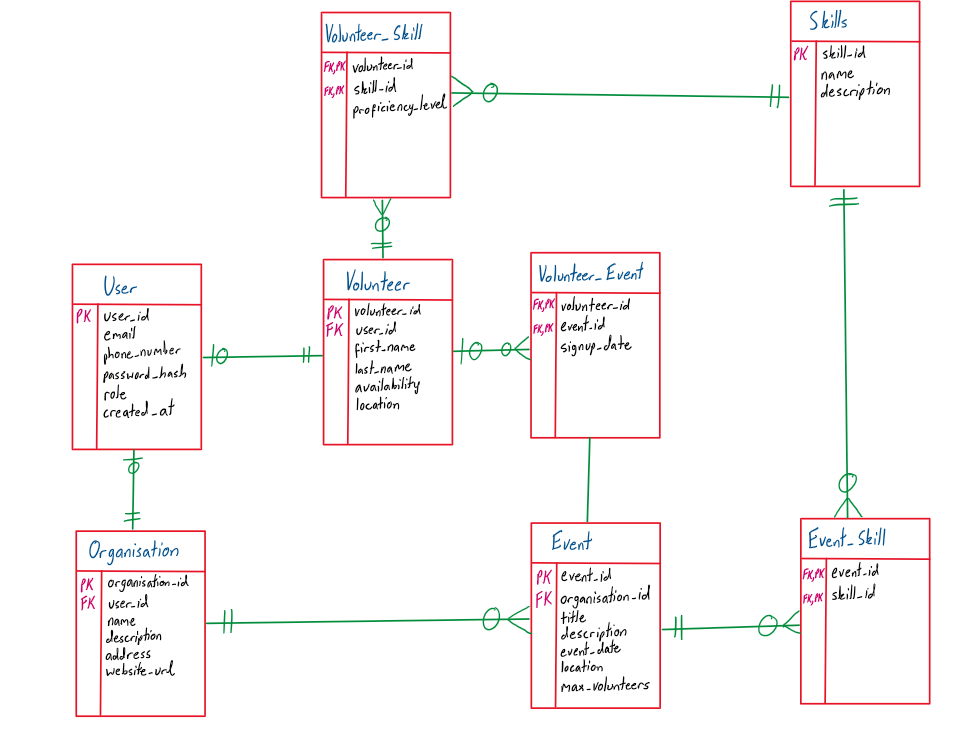
**Volunteer\_Event**(*volunteer\_id*, *event\_id*, signup\_date) (Composite primary key)

**Volunteer\_Skill**(*volunteer\_id*, *skill\_id*, proficiency\_level) (Composite primary key)

**Event\_Skill**(*event\_id*, *skill\_id*) (Composite primary key)

**Entity Relationship Diagram (18/08/2025, updated 22/08/2025):**

* One thing to make clear is that the relationship between the Event and Skill entities references the skills required by the volunteers for that specific event



**Normalisation (19/08/2025, updated 22/08/2025):**

**Unnormalised Form (0NF):**

* Is the single table, with everything in it, but only in relational notation, calling it ‘Community\_Connect’

**Issues:**

* Non-Atomic values, where the skills\_required lists all the skills the volunteer needs (Hence is 1NF)
* Organisation details are repeated for every event, and volunteer information repeated for every signup, so data is redundant
* Update Anomaly Example: Updating an organisation’s phone number requires updating it in all of the records
* Insert Anomaly Example: Can’t add a new volunteer until they sign up for an event
* Delete Anomaly Example: Deleting an event of an organisation, if it is their only event, wipes out the entire organisation
* Table also has partial and transitive dependencies, but talked about more in 2NF and 3NF

**Community\_Connect**(user\_id, email, password\_hash, role, created\_at,

volunteer\_id, volunteer\_first\_name, volunteer\_last\_name, volunteer\_email, volunteer\_phone\_number, volunteer\_availability, volunteer\_required\_skills, volunteer\_location, signup\_date

organisation\_id, organisation\_name, organisation\_description, organisation\_contact\_email, organisation\_phone\_number, organisation\_address, organisation\_website\_url,

event\_id, event\_title, event\_description, event\_date, event\_location, max\_volunteers)

**First-Normal Form (1NF):**

* All values are now atomic, with no lists or nested data, where repeating groups are eliminated
* No actual change in the relational notation of the tables, but would see change with actual data, where for example, every skill required would be on separate row

**Issues:**

* Primary key is no longer unique, so is no longer a “primary” key
* The non-key attributes do not depend on the whole primary key, so need to split them up into their separate tables, and hence create new tables for the fields that are partially dependent
* Data is redundant, where organisation information, volunteer information, event information repeated for each required skill of the volunteer

**Second-Normal Form (2NF):**

* Tables are now split up, and all partial dependencies are removed

**Issues:**

* A lot of transitive dependencies, mainly with the email and phone\_number, where, for example, the volunteer\_email, and the organisation\_email should depend on the user\_id, as they are the email of the user
* Need to also resolve the many-to-many relationships using junction tables

**User**(user\_id, password\_hash, role, created\_at)

**Volunteer**(volunteer\_id, *user\_id*, volunteer\_first\_name, volunteer\_last\_name, volunteer\_email, volunteer\_phone\_number, volunteer\_availability, volunteer\_location)

**Organisation**(organisation\_id, *user\_id*, organisation\_name, organisation\_description, organisation\_email, organisation\_phone\_number, organisation\_address, organisation\_website\_url)

**Event**(event\_id, *organisation\_id*, event\_title, event\_description, event\_date, event\_location, max\_volunteers)

**Skills**(skill\_id, skill\_name, skill\_description)

**Third-Normal Form (3NF):**

* No more transitive dependencies in the data
  + Moved email to the user table, and hence avoided repeating it in Organisation and Volunteer entities
* Also moved phone\_number to the user table, and hence avoided repeating it in the Organisation and Volunteer entities

**Issues:**

* Need to also resolve the many-to-many relationships using junction tables, but is a part of database design and not normalising the database
  + Will be later resolved in the database design phase

**User**(user\_id, email, password\_hash, phone\_number, role, created\_at)

**Volunteer**(volunteer\_id, *user\_id*, volunteer\_first\_name, volunteer\_last\_name, volunteer\_availability, volunteer\_location)

**Organisation**(organisation\_id, *user\_id*, organisation\_name, organisation\_description, organisation\_address, organisation\_website\_url)

**Event**(event\_id, *organisation\_id*, event\_title, event\_description, event\_date, event\_location, max\_volunteers)

**Skills**(skill\_id, skill\_name, skill\_description)

**Data Dictionary (22/08/2025):**

| **User** | | | | |
| --- | --- | --- | --- | --- |
| **ElementName** | **DataType** | **Size** | **Description** | **Constraints** |
| user\_id | Integer |  | Unique identifier for each user | Primary Key  Autoincrement, Unique, Not Null |
| email | Varchar | 255 | User’s email address | Unique, Not Null |
| phone\_number | Varchar | 20 | User’s phone number | Optional |
| password\_hash | Varchar | 255 | Encrypted/Hashed password | Not Null |
| role | Varchar | 255 | Role of user (Either volunteer, or organisation) | Not Null |
| created\_at | Datetime |  | When the account was created | Default: CURRENT\_TIMESTAMP |

| **Organisation** | | | | |
| --- | --- | --- | --- | --- |
| **ElementName** | **DataType** | **Size** | **Description** | **Constraints** |
| organisation\_id | Integer |  | Unique identifier for each organisation | Primary Key,  Autoincrement, Unique, Not Null |
| user\_id | Integer |  | Identifier for user account of organisation | Foreign Key -> User(User\_id),  Not Null, Unique |
| name | Varchar | 255 | Name of organisation |  |
| description | Text |  | Description of organisation | Optional |
| address | Varchar | 255 | Address of organisation | Optional |
| website\_url | Varchar | 255 | Website URL | Optional |

| **Volunteer** | | | | |
| --- | --- | --- | --- | --- |
| **ElementName** | **DataType** | **Size** | **Description** | **Constraints** |
| volunteer\_id | Integer |  | Unique identifier for each volunteer | Primary Key  Autoincrement, Unique, Not Null |
| user\_id | Integer |  | Identifier for user account of volunteer | Foreign Key -> User(user\_id),  Not Null, Unique |
| first\_name | Varchar | 20 | First name of volunteer | Not Null |
| last\_name | Varchar | 20 | Last name of volunteer | Not Null |
| availability | Varchar | 255 | Availability of volunteer (E.g. Weekdays, and their hours) | Optional |
| location | Varchar | 255 | Location of volunteer | Optional |

| **Event** | | | | |
| --- | --- | --- | --- | --- |
| **ElementName** | **DataType** | **Size** | **Description** | **Constraints** |
| event\_id | Integer |  | Unique identifier for each event | Primary Key,  Autoincrement, Unique, Not Null |
| organisation\_id | Integer |  | Identifier for organisation hosting event | Foreign Key -> Organisation(organisation\_id),  Unique, Not Null |
| title | Varchar | 255 | Title of the event | Not Null |
| description | Text |  | Description of the event | Optional |
| event\_date | Datetime |  | Scheduled date/time of event | Not Null |
| location | Varchar | 255 | Event location | Not Null |
| max\_volunteers | Integer |  | Maximum number of volunteers | Check if greater than 1  Optional |

| **Skills** | | | | |
| --- | --- | --- | --- | --- |
| **ElementName** | **DataType** | **Size** | **Description** | **Constraints** |
| skill\_id | Integer |  | Unique identifier for each skill | Primary Key,  Unique, Not Null |
| name | Varchar | 255 | Name of each skill | Not Null |
| description | Text |  | Description of each skill | Optional |

| **Volunteer\_Skill** | | | | |
| --- | --- | --- | --- | --- |
| **ElementName** | **DataType** | **Size** | **Description** | **Constraints** |
| volunteer\_id | Integer |  | Unique identifier for each volunteer | Primary Key, Foreign Key -> Volunteer(volunteer\_id)  Not Null |
| skill\_id | Integer |  | Identifier for the relevant skill of that volunteer | Primary Key, Foreign Key -> Skills(skill\_id)  Not Null |
| proficiency\_level | Varchar | 255 | Proficiency rating of the volunteer for that skill | Optional |

| **Volunteer\_Event** | | | | |
| --- | --- | --- | --- | --- |
| **ElementName** | **DataType** | **Size** | **Description** | **Constraints** |
| volunteer\_id | Integer |  | Unique identifier for each volunteer | Primary Key, Foreign Key -> Volunteer(volunteer\_id),  Not Null |
| event\_id | Integer |  | Identifier for the event the volunteer is a part of | Primary Key, Foreign Key -> Event(event\_id),  Not Null |
| signup\_date | Datetime |  | The date when volunteer signed up | Default: CURRENT\_TIMESTAMP |

| **Event\_Skill** | | | | |
| --- | --- | --- | --- | --- |
| **ElementName** | **DataType** | **Size** | **Description** | **Constraints** |
| event\_id | Integer |  | Unique identifier for the event | Primary Key, Foreign Key -> Event(event\_id),  Not Null |
| skill\_id | Integer |  | Identifier for the skills required for that event | Primary Key, Foreign Key -> Skills(skill\_id),  Not Null |

**Ethical & Legal Research (22/08/2025):**

* Main law that regulates the handling of personal information in Australia is the Privacy Act, where it applies to most government agencies and other large private organisations, enforced by the Office of the Australian Information Commissioner (OAIC)
* Its purpose is to protect privacy, regulate how personal information is collected, used, stored, disclosed
* The Australian Privacy Principles are a set of 13 principles that guide how organisations must handle personal information

**Main APPs In Syllabus (Taken From Notes):**

**APP5:**

* Notification of Collection of Personal Information

* Organisation must take reasonable steps to notify individual about what personal information they might be collecting and why

**APP8:**

* Cross-Border Disclosure of Personal Information

* Organisation must take reasonable steps to ensure that any overseas recipient of personal information complies with APPs (Crucial for data on international servers)

**APP10:**

* Quality of personal information

* Organisation must take reasonable steps to ensure personal information collected is accurate, up-to-date, complete

**APP11:**

* Security of Personal Information

* Organisation must take reasonable steps to protect personal information from misuse, interference, unauthorised access, modification, disclosure
  + Must also have safe destruction of data it no longer uses

**APP12:**

* Access to Personal Information

* Organisation must give individual access for personal information upon request
  + Gives individuals right to see what an organisation holds about them

* Ensures transparency, allows individuals to check for accuracy and request corrections

**Other Ethical/Security Concerns Related to Storing Personal Data:**

**Ethical Issues**

**Collecting Data About Individuals:**

* **Consent:** Do people know their data is being collected?

* **Purpose:** Is data collected for specific, legitimate purpose, or just because you can

* **Minimisation:** Is the data collected absolutely necessary

* **Anonymisation:** How to ensure data collected can't be traced back to individual

**Privacy Concerns:**

* Right of individuals to control personal data, ensuring data is collected and used responsibly

* **Public vs. Personal Data:** Line is often blurry, is a public social media post truly "public"?

* **Surveillance:** Use of data to monitor individuals' behaviour without explicit knowledge/consent raises major privacy concerns

* **Data Breaches:** How to protect sensitive data from being stolen/leaked

**Appropriate Use of Data:**

* Data collected for one purpose shouldn't be used for another without consent

* **Fairness and Bias:** Is data being used to discriminate against certain groups?

* **Impact:** What are real-world consequences of data-driven decisions?

**Reliability of Data Sources:**

* **Accuracy:** Is data correct and free from errors?

* **Bias:** Is data influenced by the source? Can it be verified by other sources?

* **Verifiability:** Can the data be confirmed as authentic

**Acknowledgement of Data Sources:**

* **Giving Credit:** Must acknowledge source of data

* **Transparency:** Shows transparency and allows others to verify your information

* **Legal and Licensing:** Many datasets gave specific licences for use, respecting licences is ethical and legal decision

**Use of Data Mining:**

* Data mining is process of discovering patterns & insights from large datasets

* **Profiling:** Created a detailed profile of an individual's habits, interests, behaviour

* **Predictive Analysis:** Using data to predict future behaviour (E.g. Credit scores, Insurance premiums)

**Data Security**

**Keeping Personal Data Private:**

* Data security is the technical and procedural protection of data, you can have secure data that still violates privacy if it's used inappropriately

* Methods for ensuring privacy (E.g. Encryption, Anonymisation, Access Controls)

**Backups of Organisational Data:**

* Backups are copies of data made to restore it in the event of data loss

* **Disaster Recovery:** Protection against hardware failure, cyberattacks (E.g. Ransomware), Natural disasters

* **Accidental Deletion:** Allows for restoration of inadvertently deleted files or records

* **Business Continuity:** Ensures that organisations can quickly recover and continue its operations after an incident

* **Methods:** On-Site (Local storage), and Off-Site (Cloud/Physical copies) backups are both crucial

**Restricting Access to Data:**

* Access Controls = Security techniques that regulates who or what can view/use resources in a computing environment

* **Authentication:** Verifying the identity of the user (E.g. Username and Password)

* **Authorisation:** Granting a user specific permissions based on their verified identity

* **Principle of Least Privilege:** Key security concept where users are granted minimum level of access necessary to perform their jobs

**Ownership and Control of Data:**

* For personal data, the individual has a fundamental right to ownership, who decides how the data is used, shared, stored?

* Companies collect & control vast amounts of personal data, challenge is ensuring they use data in a way that respects individual's ownership

**Project Management Plan Outlining Timelines (22/08/2025):**

* Will be following the plan set by our teacher

**Deliverable 1 - Due 22/08/2025:**

* A project management plan outlining timelines.
* A research summary with referenced articles.
* A complete ERD in Crow's Foot notation.
* Full documentation of the normalisation process to 3NF.
* A complete data dictionary.

**Deliverable 2 - Due 29/08/2025:**

* A .sql file containing all CREATE TABLE and INSERT statements.
* The SQLite database file (.db).
* The Python Flask project files.
* Screen captures of the basic web application performing the CRUD operations.

**Deliverable 3 - Due 5/09/2025:**

* The final, fully functional Python Flask application and SQLite database.
* A comprehensive final report in digital format.

**Deliverable 2 (Started -> 26/08/2025):**

**Screen Captures (29/08/2025, Updated 30/08/2025):**

**Create – Add Form For New Volunteer:**

**A screenshot of a login form

AI-generated content may be incorrect.**

**Read – Page That Displays List of Organisations From Database:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Update – Form To Update Volunteer’s Contact Number:**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**Delete – Button To Remove Specific Event:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**ACID Properties (30/08/2025):**

* When a volunteer signs up for an event, a new record is inserted into the ‘volunteer\_event’ junction table, which records which volunteer attends which event, and the signup date
  + Feature has not yet been added to the database

**Atomicity (“A transaction is atomic, it’s an indivisible unit”):**

* The signup is one single transaction
* Either signup is fully recorded (All database changes happen), or nothing happens at all
* E.g. If database crashes in middle of inserting signup, the row isn’t just partially added

**Consistency (“The database must always be in a valid state”):**

* Database rules (E.g. Foreign key constraints, ‘Unique’ constraints, etc.) are enforced
* Only the valid signups are allowed, for example, a volunteer and event must exist for the transaction to occur
* E.g. You can’t sign up to an event that doesn’t even exist

**Isolation (“Multiple transactions simultaneously don’t affect each other’s execution):**

* Multiple volunteers can sign up without interfering each other
* Each transaction is isolated, so concurrent signups don’t cause incorrect data

**Durability (“Once transaction is saved, it stays saved”):**

* Once volunteer’s signup is successfully recorded, it remains in database even if there is a crash afterwards
* E.g. Volunteer’s attendance record won’t disappear after transaction is completed