

# CSC 12 ATAR Database Project - Community Connect

## Github Link:

[https://github.com/IMRAN-Anisha/CMSCI-DB\\_PRO](https://github.com/IMRAN-Anisha/CMSCI-DB_PRO)

## Deployment Instructions

1. Right click on “app.py” located in **CMSCI-DB\_PRO/CommunityConnect/app/app.py**.
2. Select “Open in **integrated** terminal”.
3. Enter “**python app.py**” in the terminal and press enter.
4. Open development server link.
5. Flask app will be running, then navigate through the website.

## ACID Properties

- **Atomicity:** Transactions (INSERT/UPDATE/DELETE) are either fully applied or not at all.  
Example: when registering a volunteer, their user record and volunteer profile must both be created.
- **Consistency:** Constraints ensure that the database moves from one valid state to another.  
Example: primary keys, foreign keys, NOT NULL
- **Isolation:** Transactions run independently.  
For example, two volunteers registering simultaneously do not interfere with each other's data.
- **Durability:** Once committed, changes are stored in the database even after failure.

## SQL Table Creation

- Nine tables were created, please refer to the create\_tables.sql file.  
**Users, Volunteers, Companies, Events, VolunteerEvents, VolunteerProfile, Notifications, Skills, EventRequirements**

- Primary keys, foreign keys, uniqueness, NOT NULL, and DEFAULT constraints were set.
- Referential integrity enforced using FOREIGN KEY.
- ON DELETE CASCADE added where necessary to maintain consistency.
- CHECK constraint added for status in VolunteerEvents.

## Data Population

After creating the Users table and writing its initial INSERT statements, these statements were used to prompt the AI to generate dummy data. For each table in the database, 5 rows of sample data were created and saved in the insert\_data.sql file.

## Web Application

### Technology Stack:

**Backend:** Python Flask

**Database:** SQLite

**Frontend:** HTML, CSS, basic JavaScript

**Web Pages:** Please find them in the templates folder.

- index.html
- Login.html
- Register\_volunteer.html
- register\_organisation.html
- communityConnectHomePage.html
- VolunteerDashboard.html
- organisationDashboard.html
- EditPostEvents.html
- EventSearch.html
- VolunteerProfilePage.html
- EditVolunteerProfilePage.html
- Notifications.html
- Accountsettings.html
- PrivacyPolicy.html

## CRUD Demonstration

- **Create:** Registration form inserts a new volunteer into the database.

### Register as a Volunteer

First Name:

Last Name:

Date of Birth:

 

Address:

Phone Number:

Email Address:

Password:

Confirm Password:

**Register**

Already have an account? [Log in here.](#)

## Register as an Organisation

Company Name:

Company Email:

Company Phone:

Location:

Password:

Confirm Password:

**Register**

Already have an account? [Log in here.](#)

- **Read:** Organisations list displayed using SELECT.
- **Update:** Volunteers can update their contact number or profile details.

**About Me**  
Add an About Me!

**Skills**

- First Aid
- Event Planning
- Community Outreach

**Edit Profile**

Full Name:

Title / Position:

Bio:

- **Delete:** Events can be deleted by organisations.

**Community Connect**      Dashboard    Events    Notifications    Account Settings

**Account Settings**  
Manage your account details securely.

**Email Address**

Current Email:

New Email:

**Update Email**

**Password**

Current Password:

New Password:

Confirm Password:

**Update Password**

**Delete Account**

**Warning:** This action cannot be undone.

**Delete Account**

© 2025 Community Connect. All rights reserved.

## **Extra Features in Progress** : not assessed

- Profile image update with dynamic JavaScript refresh.
- Volunteer search feature for organisations.
- Notifications per volunteer, updated individually.
- Error handling with user-friendly messages (e.g. “username already taken”).

## **Error Handling in Development**

- A circular import error was encountered (from app import create\_app).
  - Solution: removed app\_routes.py and \_\_init\_\_.py and simplified to app.py only
- Database Locked
  - Restarted the computer, later confirmed with the teacher that the database cannot be opened while performing operations and will cause the database to lock itself.

## **Testing**

- **Database** tested by inserting, updating, and deleting sample records.
- **CRUD operations** verified on the web interface.
- **Error case handling** tested by trying duplicate usernames and invalid inputs.
- **Referential integrity** tested with ON DELETE CASCADE.