CoLab User Manual

Harvey Bellini, Michael Hills, Theo Dal Pozzo Dos Santos,
Pedro Pereira, Joshua Prout

2023/2024

Table of Contents

Preface

This project would not have been made possible without the continued efforts of Dr Majeed Soufian and John Stammers.

This document provides everything a user should know about the “CoLab” project. It details full installation steps, troubleshooting and usage guides.

This is aimed at users from a non-technical background and not at developers. Developers should seek the related “developer guide” in the project directory.

System Requirements

The program has some recommended system requirements that should be met in order for the program to execute as intended. Note: no minimum system requirements can be generated for this program.

The operating system used must be Windows 10/11. Functionality beyond these versions are expected to work, but are untested and **not recommended.**

Hardware recommended requirements

Item	Recommended	Confirmed working with
CPU	I3 7th Gen	AMD Ryzen 5-5600H
RAM	8GB	32GB
Storage	10GB	10GB

The storage requirements for this project are 10GB. This is due to Python 3.10 requiring around 6GB on its own, and the program (without data in the database) comes out to around 5MB. The program dependencies combined require about 250MB of storage. Therefore, 10GB provides suitable overhead for not only installation, but further project development.

The program relies on a Django backend, which is a lightweight framework for web applications and has low requirements, hence the modest CPU and RAM recommendations.

Software recommended requirements

Name	Version	Note
Python	3.10	The program will need modification to work with newer versions of Python
Django	4.1.6	Framework for development
matplotlib	3.6.3	Package used for generating graphs
pandas	1.5.3	Package for data forms
seaborn	0.13.2	Package for graph generation
django-multiselectfield	0.1.12	Plugin for multi-select fields in Django forms
django-filter	23.5	Library for filtering Django QuerySets based on user selections
reportlab	4.1.0	Library for PDF generation in Django

django-auto-logout	0.5.1	Middleware for automatically logging out inactive users
django-cryptography	1.1	Provides cryptography tools for Django models
django-admin-logs	1.2.0	Plugin for tracking admin site activities
numpy	1.26.4	Fundamental package for scientific computing in Python

It is **strongly** recommended to use the versions listed, as only these versions have been fully tested and updated versions are **not** supported. Full details for installation of the correct versions are listed below. Additionally, a web browser is required to access the program. Any web browser can be used with the program.

Installation

This section details the procedure for installing prerequisite software, and installing the application itself with both client and developer options. Python 3.10 must be installed on all systems, as this is the language the project is mainly written in. Additionally, the code was written specifically to be run and operated on a Windows machine, thus, full functionality is not guaranteed on either Mac or Linux systems.

Pre-requisites

Python 3.10

Python 3.10 is the version of Python the program is written in. It can be downloaded here:

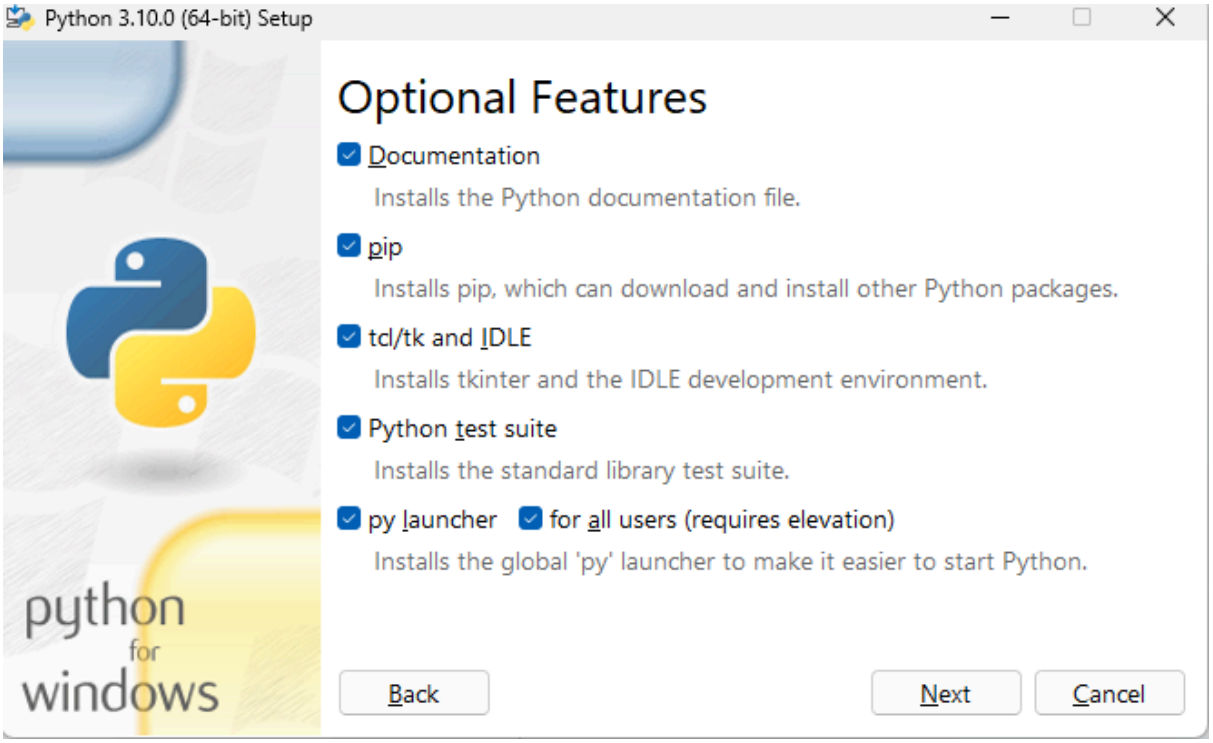
https://www.python.org/downloads/release/python-3100/

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		729e36388ae9a832b01cf9138921b383	25007016	SIG
XZ compressed source tarball	Source release		3e7035d272680f80e3ce4e8eb492d580	18726176	SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later (updated for macOS 12 Monterey)	8575cc983035ea2f0414e25ce0289ab8	39735213	SIG
Windows embeddable package (32-bit)	Windows		dc9d1abc644dd78f5e48edae38c7bc6b	7521592	SIG
Windows embeddable package (64-bit)	Windows		340408540eeff359d5eaf93139ab90fd	8474319	SIG
Windows help file	Windows		9d7b80c1c23cfb2cecd63ac4fac9766e	9559706	SIG
Windows installer (32-bit)	Windows		133aa48145032e341ad2a000cd3bff50	27194856	SIG
Windows installer (64-bit)	Windows	Recommended	c3917c08a7fe85db7203da6dcaa99a70	28315928	SIG

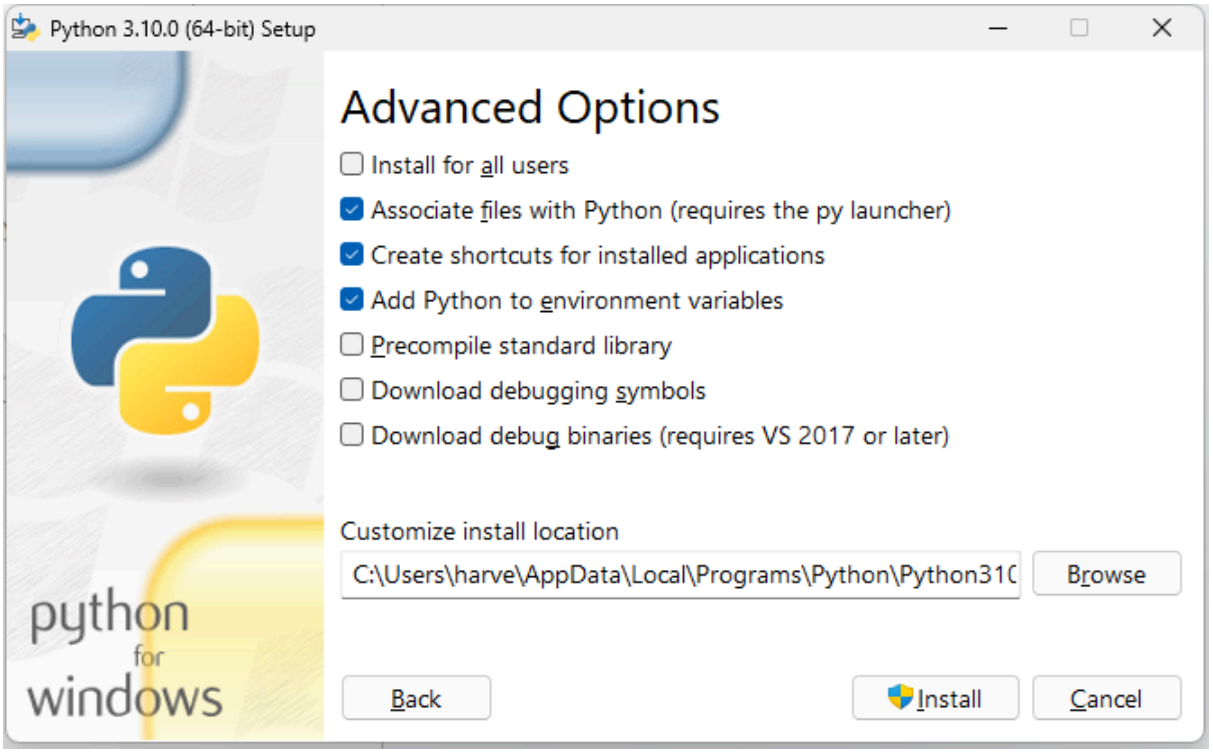
It is recommended to choose the last option, the 64-bit installer for Windows. Click on “Windows Installer (64-bit)” and an EXE will download. Run the downloaded EXE. (You may be asked to run the application as administrator.)

You will be greeted with the following image. Select “Add Python 3.10 to PATH” then select “Customize installation”

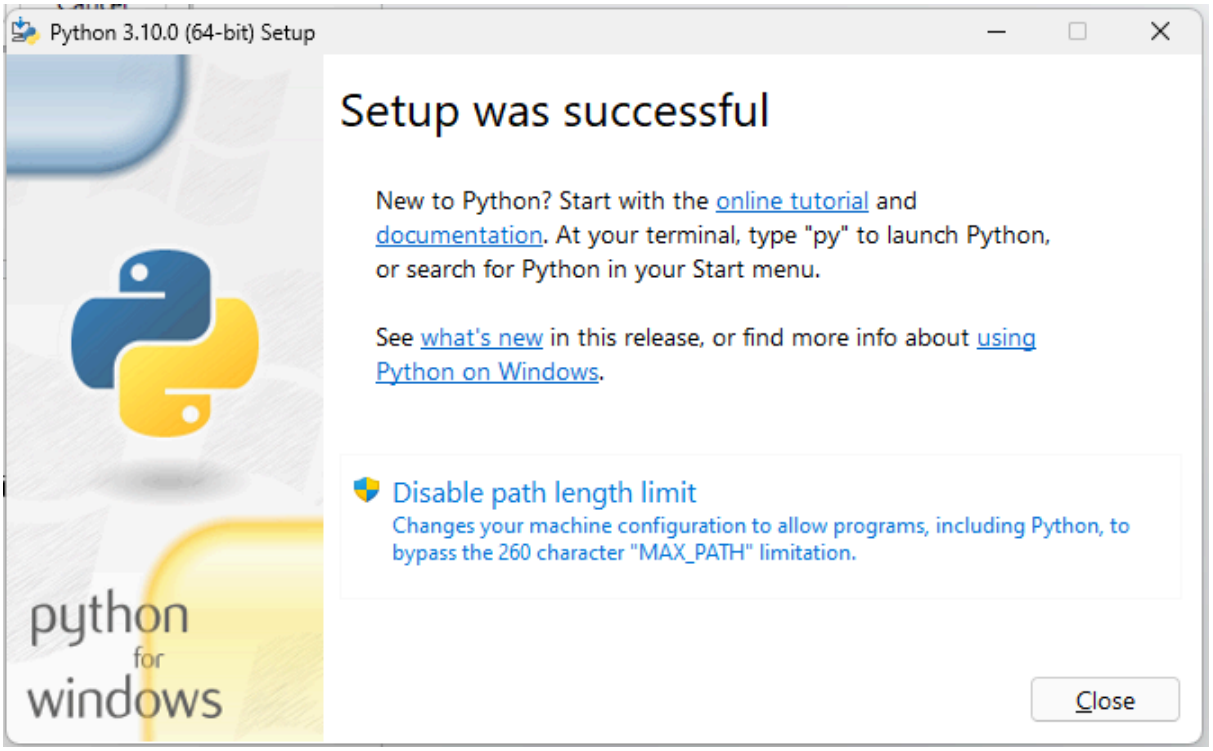
Select all optional features then press “Next”.



After selecting “Next”, you MUST select the second and fourth options: “Associate files with Python” and “Add Python to environment variables.” The other options are optional and will not impact the program. Additionally, you may change the install location, although this is not recommended. Press “Install”

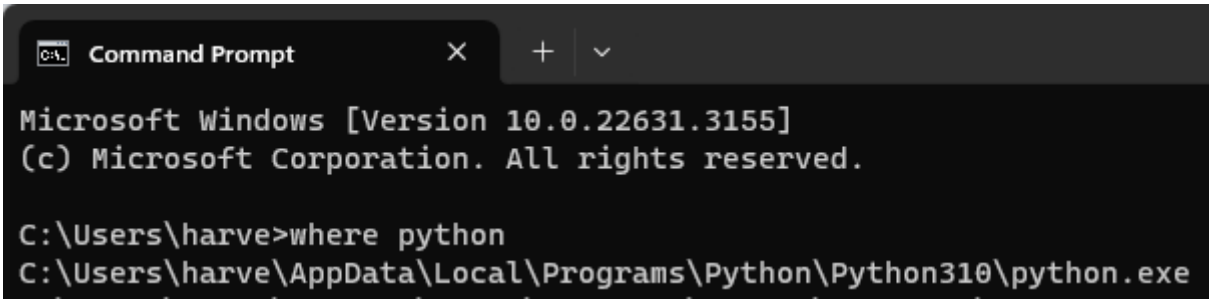


You will be greeted with an administrator privileges page, select “Yes” or “Continue” and enter a password if necessary. After this, simply wait for the program to install and you will be greeted with the following:

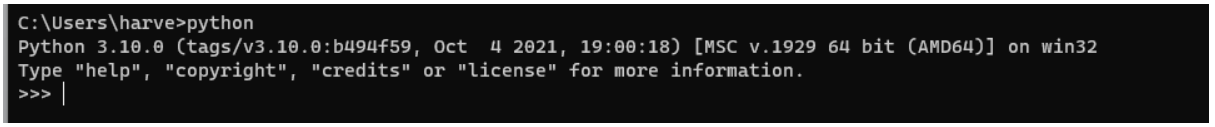


Select “Disable path length limit” and accept the administrator privileges. After this, the setup of Python 3.10 has been completed.

Installation can be verified by opening a command prompt window and typing “where python”. If python was installed correctly you will be given the path to python.exe.



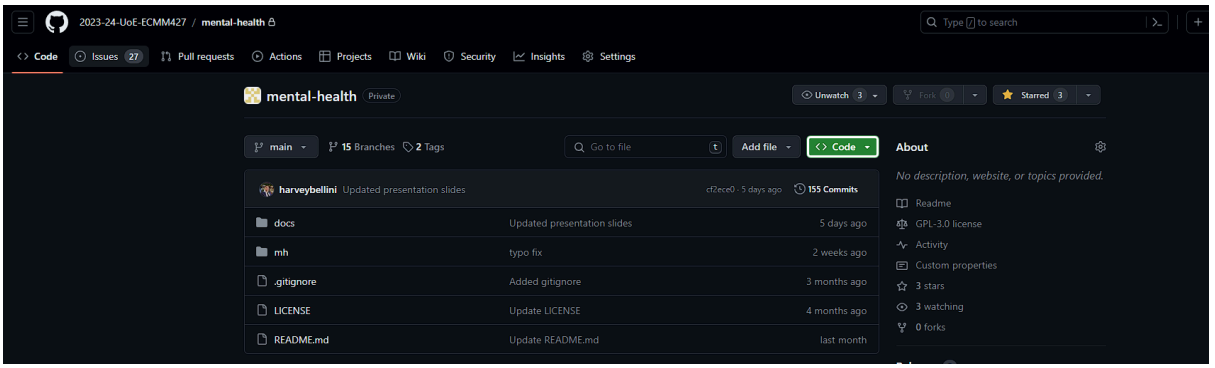
You can further verify by typing “python” and pressing enter:



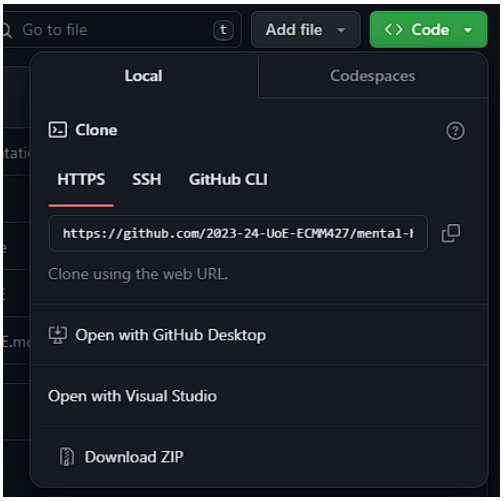
If neither of the two work, you will need to uninstall Python and restart the process.

Program Files

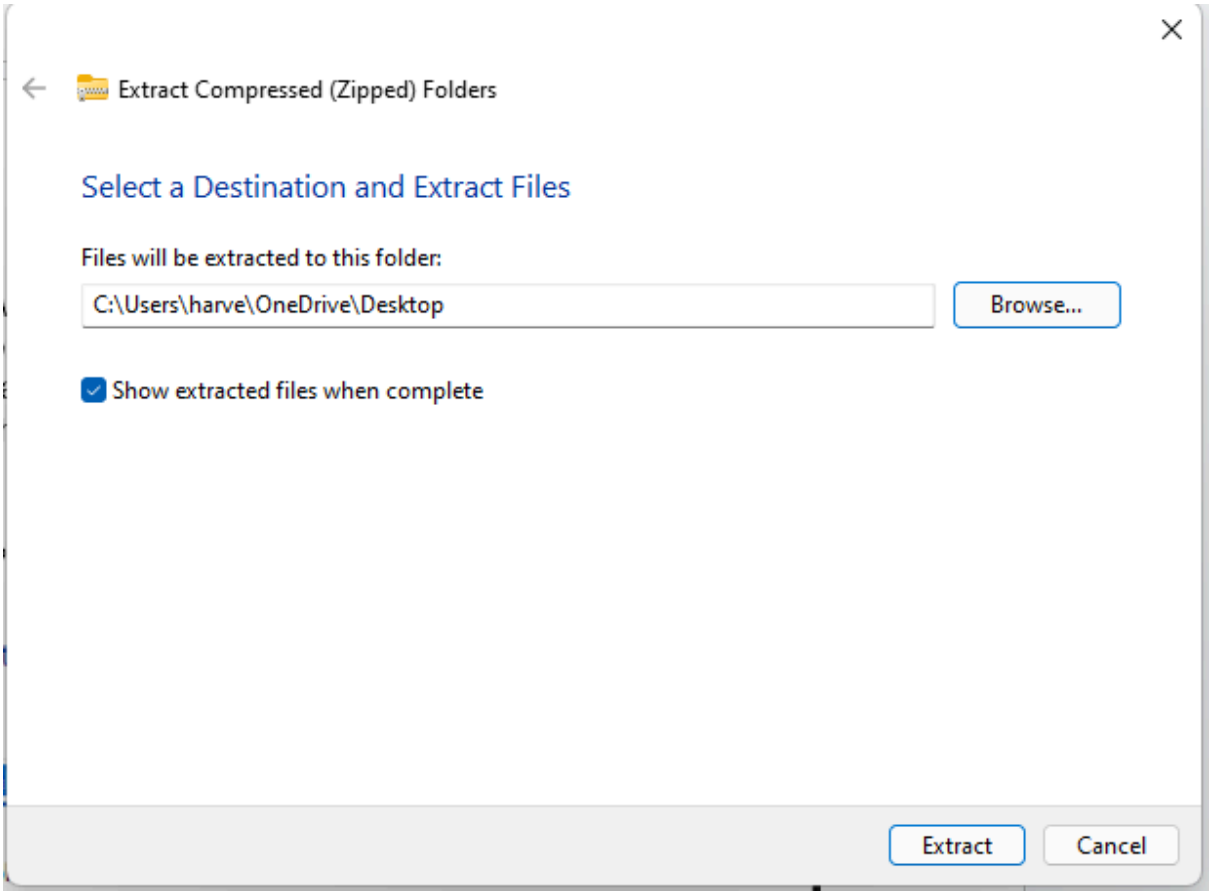
Next, we need to download the necessary files for the program to run. These will either be sent as a Zip File, or alternatively can be downloaded from Github. If access to the Github is provided, it will look similar to this:



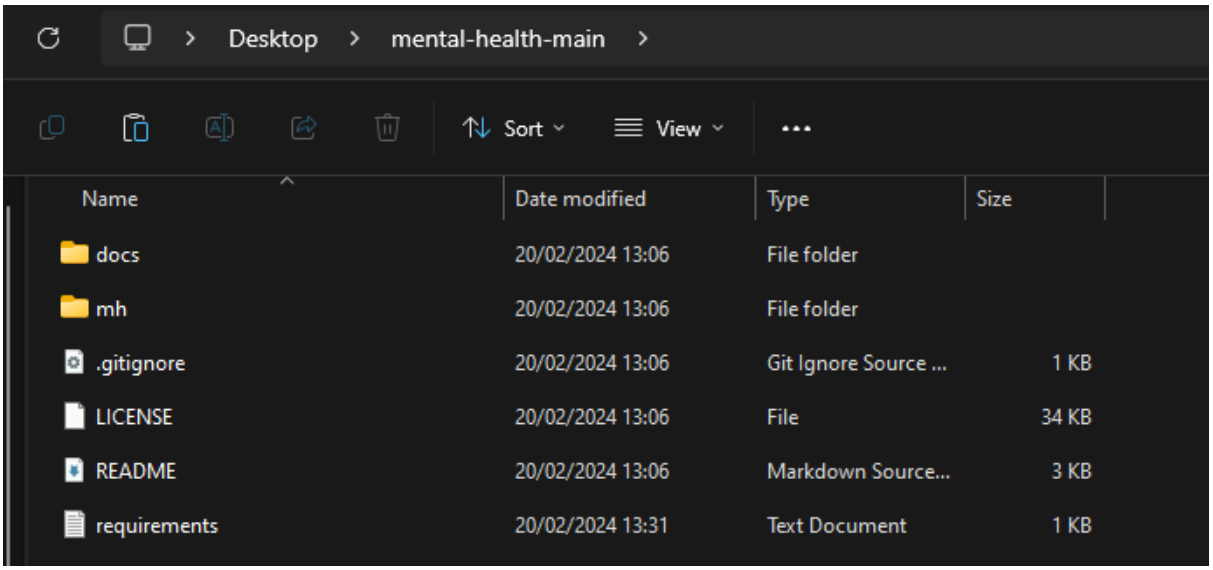
Click the green “Code” button:



Then select “Download ZIP”. This will download a ZIP file containing the code. Navigate to the Downloads folder. Right click on the downloaded ZIP file and select “Extract All”. This will create a pop-up confirming the location of the extracted files. This location can be anywhere on the PC, but for the purpose of this documentation, it will be placed on the local users Desktop. Note this path location down in this instance “C:\Users\harve\OneDrive\Desktop”. Then, click “Extract”.



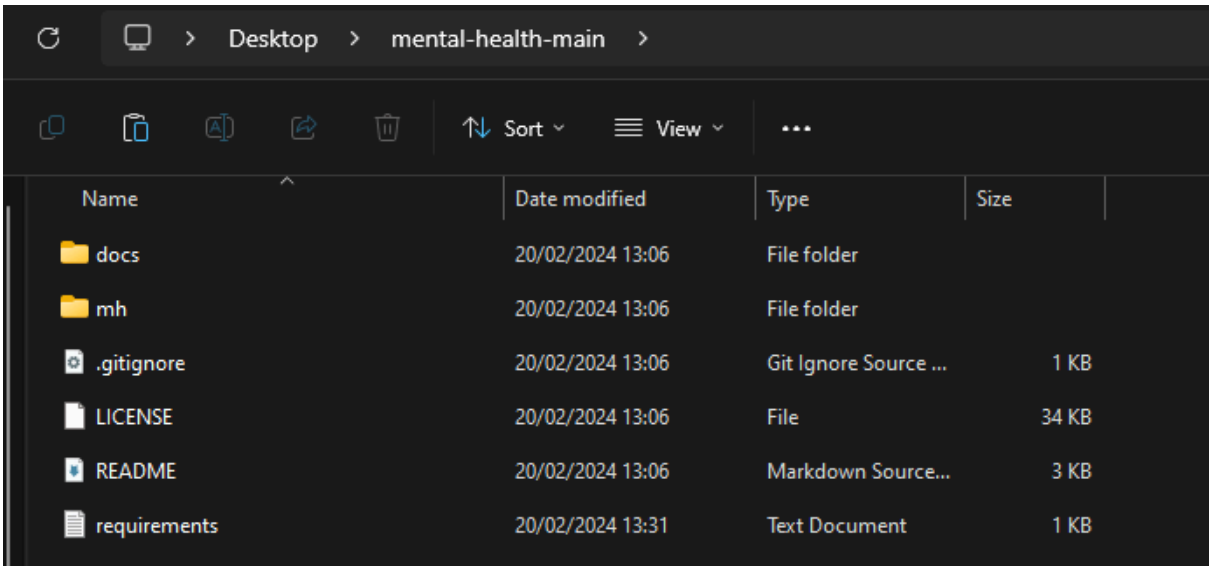
This will extract the files to the selected location and the created folder will be present.



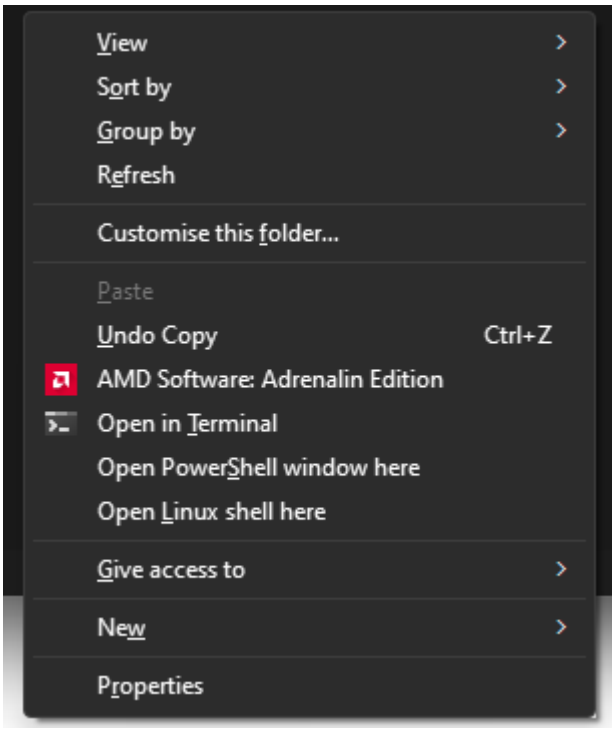
The structure of the repository and files will be explained in a later section.

Pip Requirements

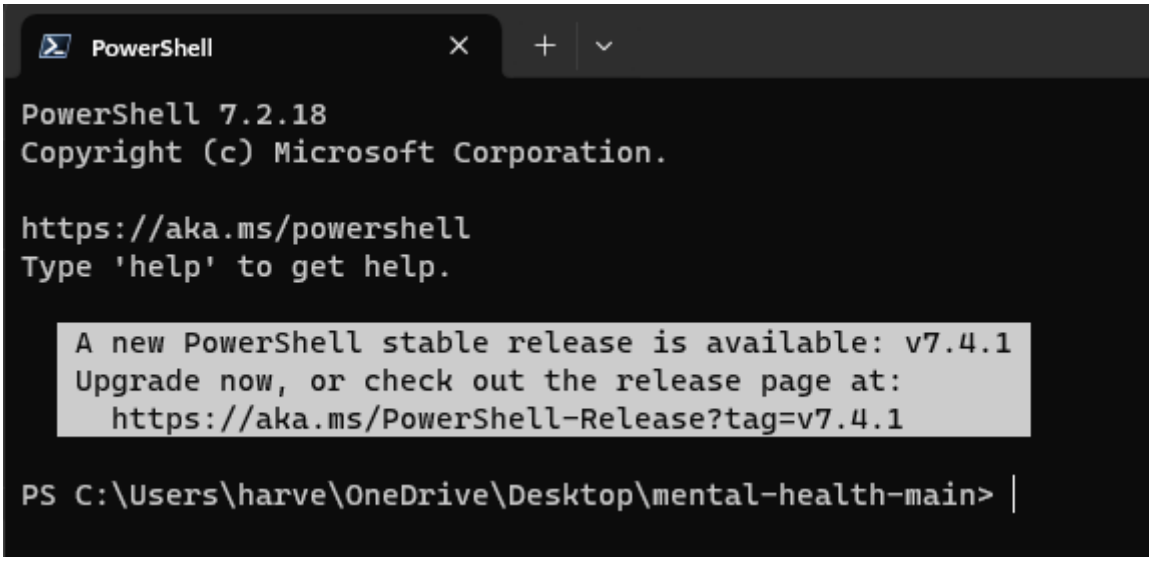
The program relies on several dependencies, for various functionality, these are all noted in the file “requirements.txt”. To install the necessary requirements, complete the following steps. In File Explorer, navigate to the folder where the program files are located.



Hold “SHIFT” and right click on empty space within the folder. Select “Open in terminal”:



This will open a terminal:



Enter “pip install -r requirements.txt” then press enter. The terminal will then use the pip package manager, as installed in Python, to download and install all the necessary software components for this project. When the following message is shown, the operation has completed:

```
Successfully installed Django-4.1.6 matplotlib-3.6.3 pandas-1.5.3 seaborn-0.13.2
PS C:\Users\harve\OneDrive\Desktop\mental-health-main>
```

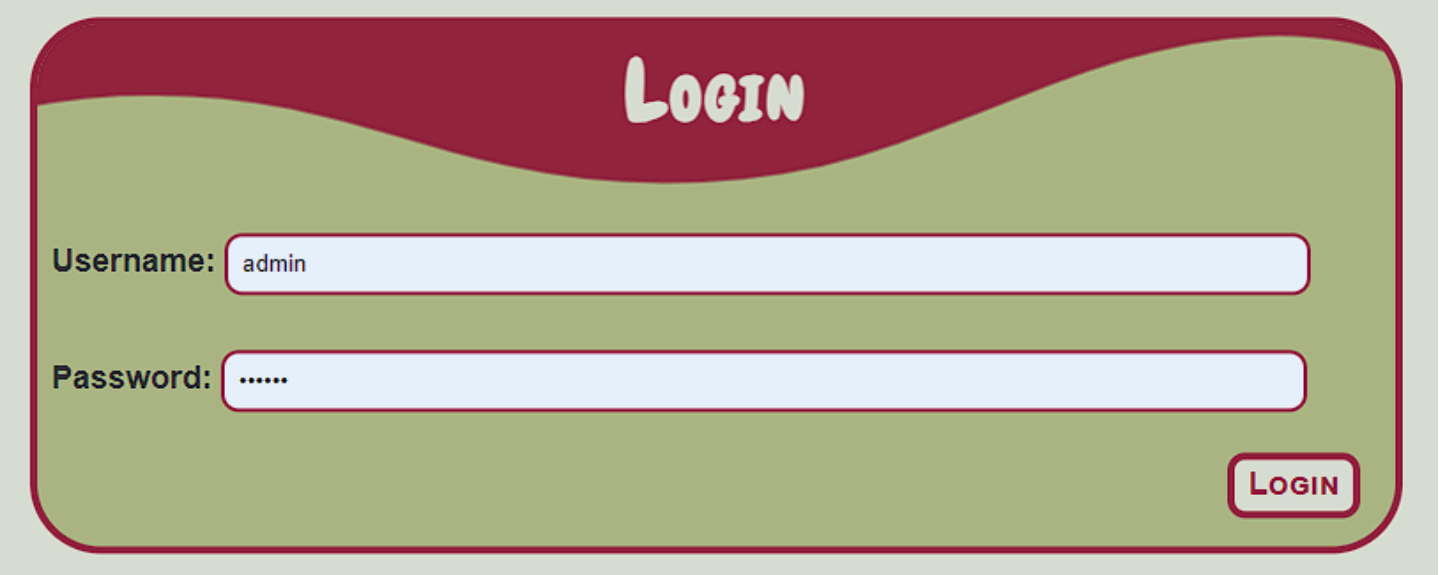
Now, the program can be run directly from the terminal, to ensure installation has been completed successfully. Full details on operation are detailed below.

To ensure successful installation, enter “cd mh” followed by “python manage.py runserver”. The following message indicated the program is successfully running:

```
PS C:\Users\harve\OneDrive\Desktop\mental-health-main> cd mh
PS C:\Users\harve\OneDrive\Desktop\mental-health-main\mh> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
February 20, 2024 - 13:42:25
Django version 4.1.6, using settings 'mh.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

CTRL + left click on “http://127.0.0.1:8000/” will open the program in your default web browser. If you are greeted with the following page, installation has been successful:

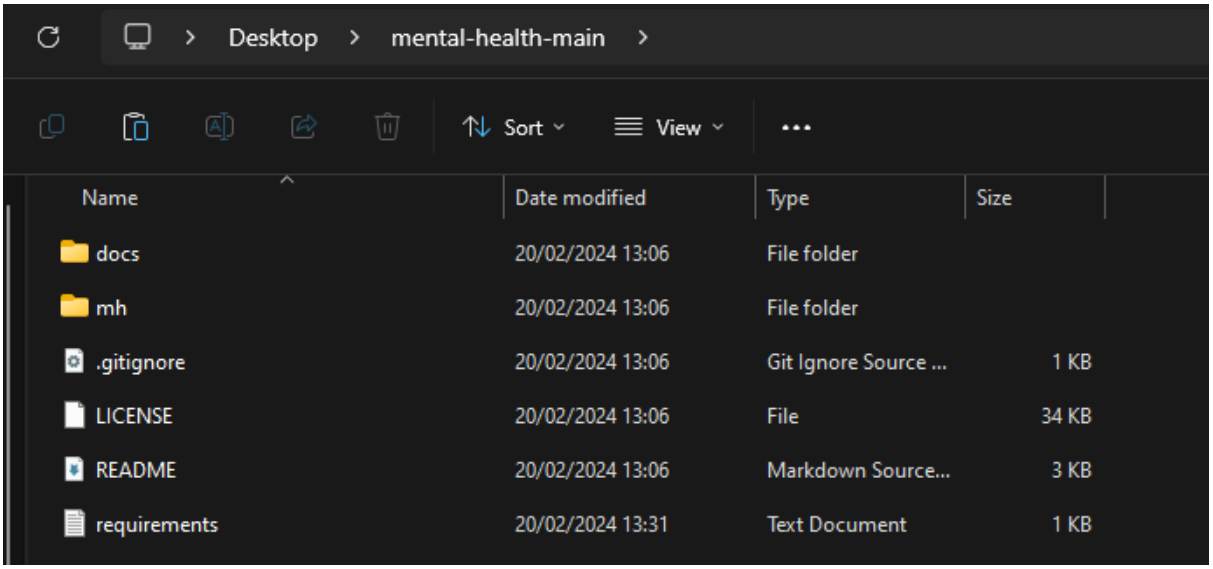


Usage Guide

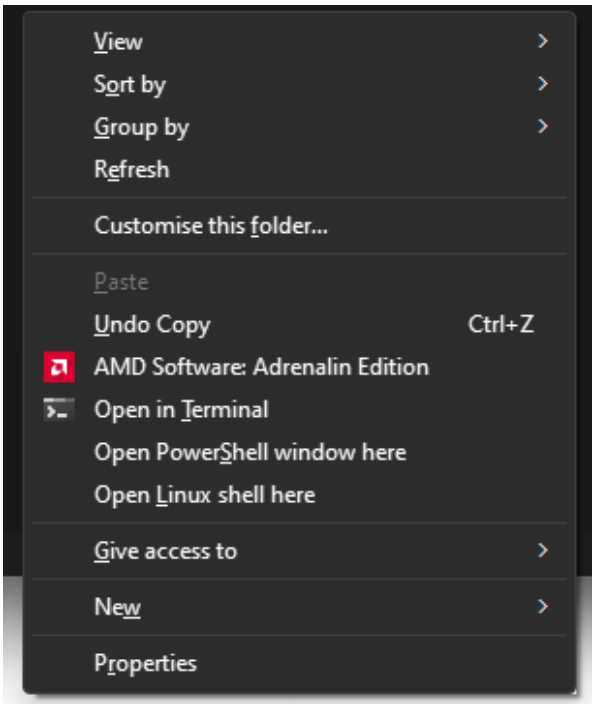
This section details how to use the program and details **ALL** the program's functionality. There are five main areas to detail: Login, Profiles, Workshops, Data Analysis and RAC form.

Start the Program

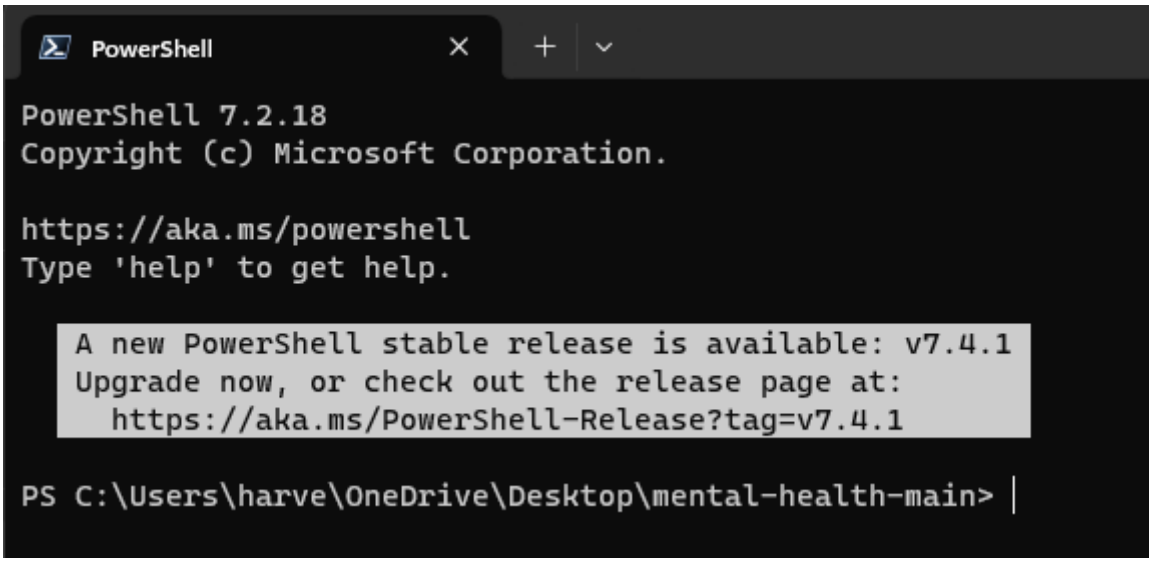
Navigate to the folder containing the program:



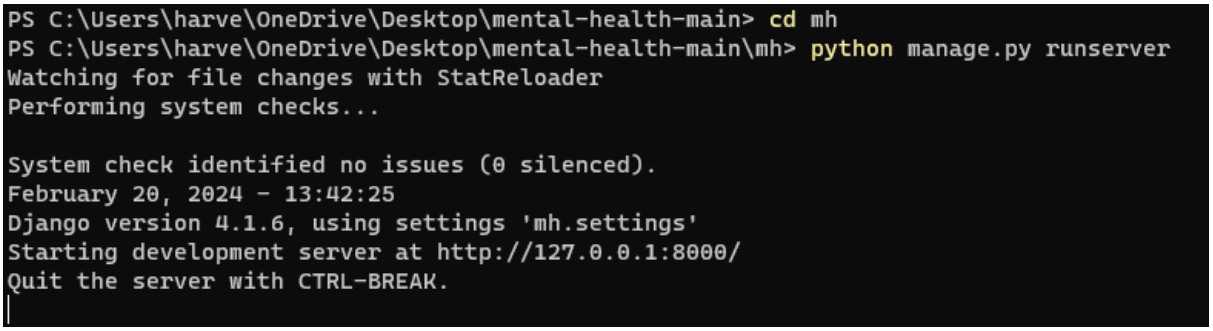
Hold "SHIFT" and right click on empty space within the folder. Select "Open in terminal":



This will open a terminal:



enter “cd mh” followed by “python manage.py runserver”. The following message indicated the program is successfully running:



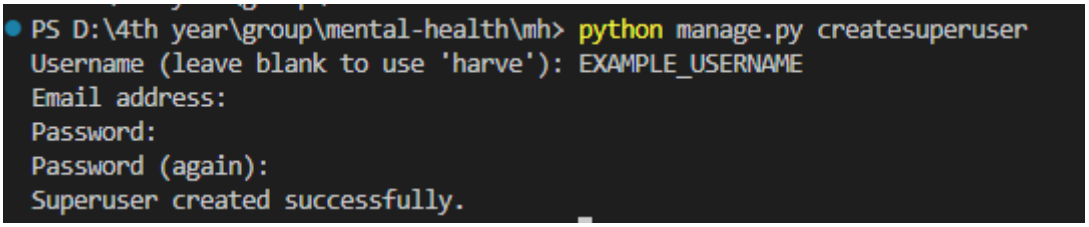
CTRL + left clicking on “http://127.0.0.1:8000/” will open the program in your default web browser. If you are greeted with the login page, the program is running successfully.

A common issue to encounter at this stage is the “PORT IN USE” issue, a solution can be found below in FAQ’s.

Creating a user

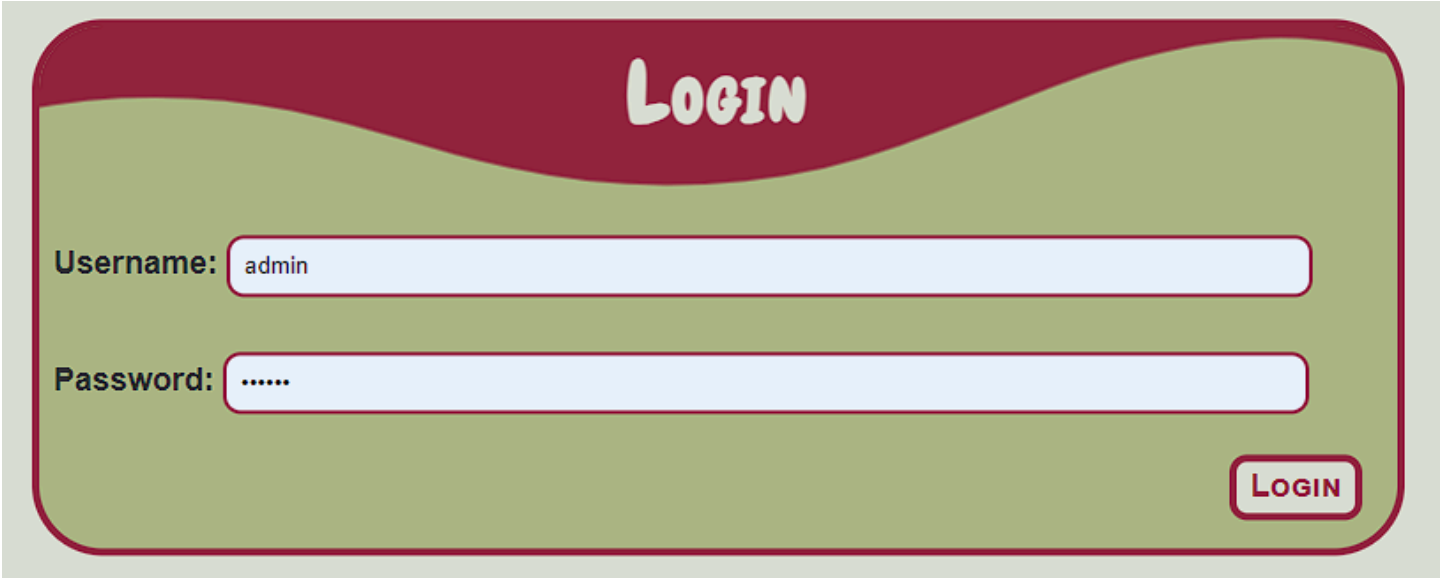
In order to create a user account, replicate the steps above, but instead of entering “python manage.py runserver” enter “python manage.py createsuperuser”. This will ask the user to enter a “username” (this must be filled in. It will also ask for “email” (this can be left blank) and then a double entry of the password. Note: the user will not see the password entry for security purposes:

The message “Superuser created successfully” indicates that a valid username and password has been created. This will create a user with maximum permissions.



Login

Before the program can be used, the user must login. The login page looks like the following:

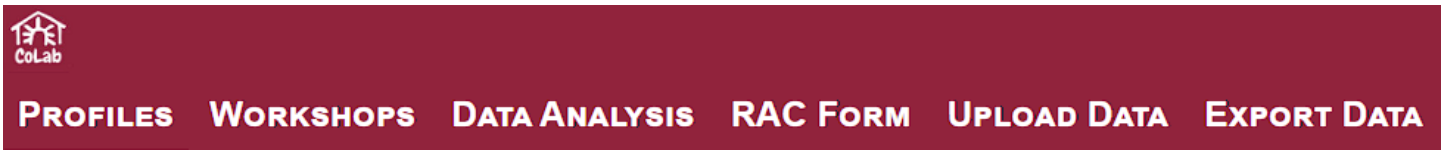


The user should enter their username and password and press “LOGIN”, this will redirect the user, upon successful validation, to the program functionality. The user lands on the “Profiles” page by default.

The “Create Profile” button is currently a placeholder to add more users, however this feature wasn’t implemented due to our number of users being one.

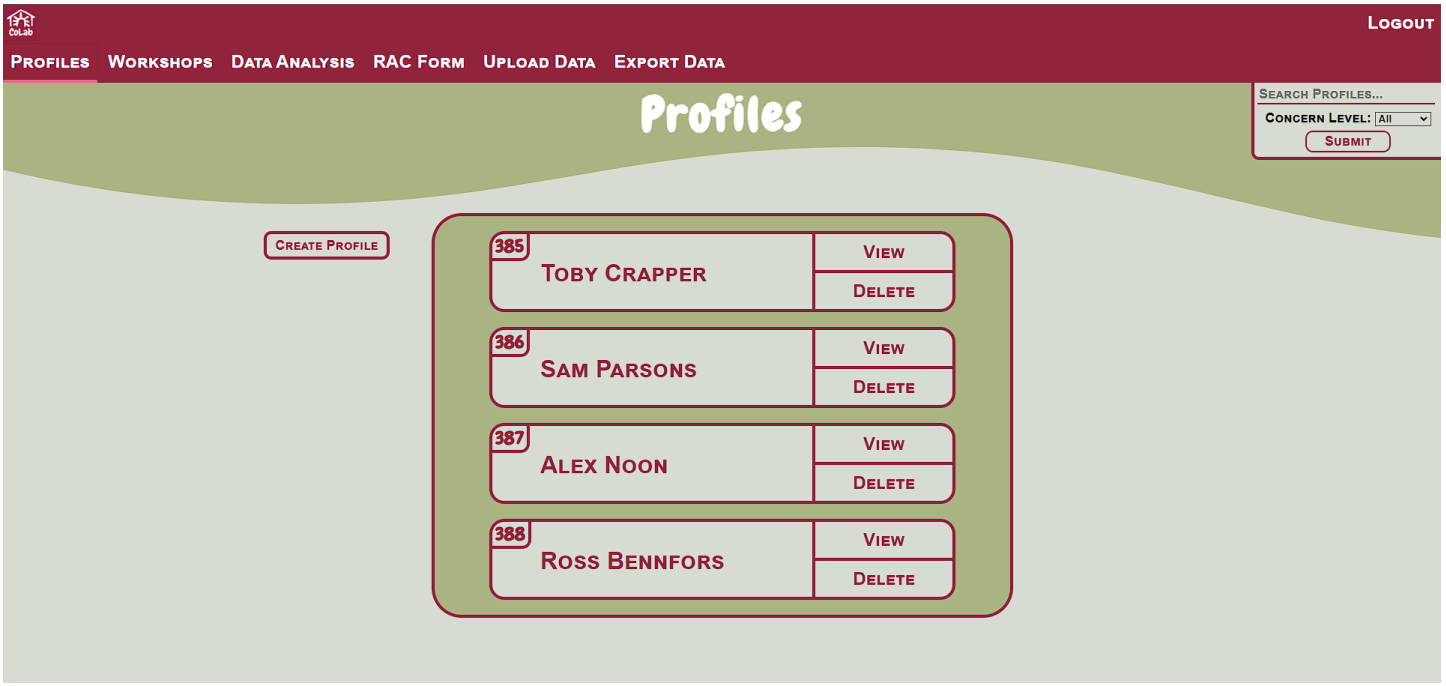
Navbar

The navbar remains present across all main pages. It provides the user with a quick switch method between the main functionality of the program. Each button: “Profiles”, “Workshops”, “Data Analysis”,“RAC Form”, “Upload Data” and “Export Data” navigate the user to each module of the program, each explained below.

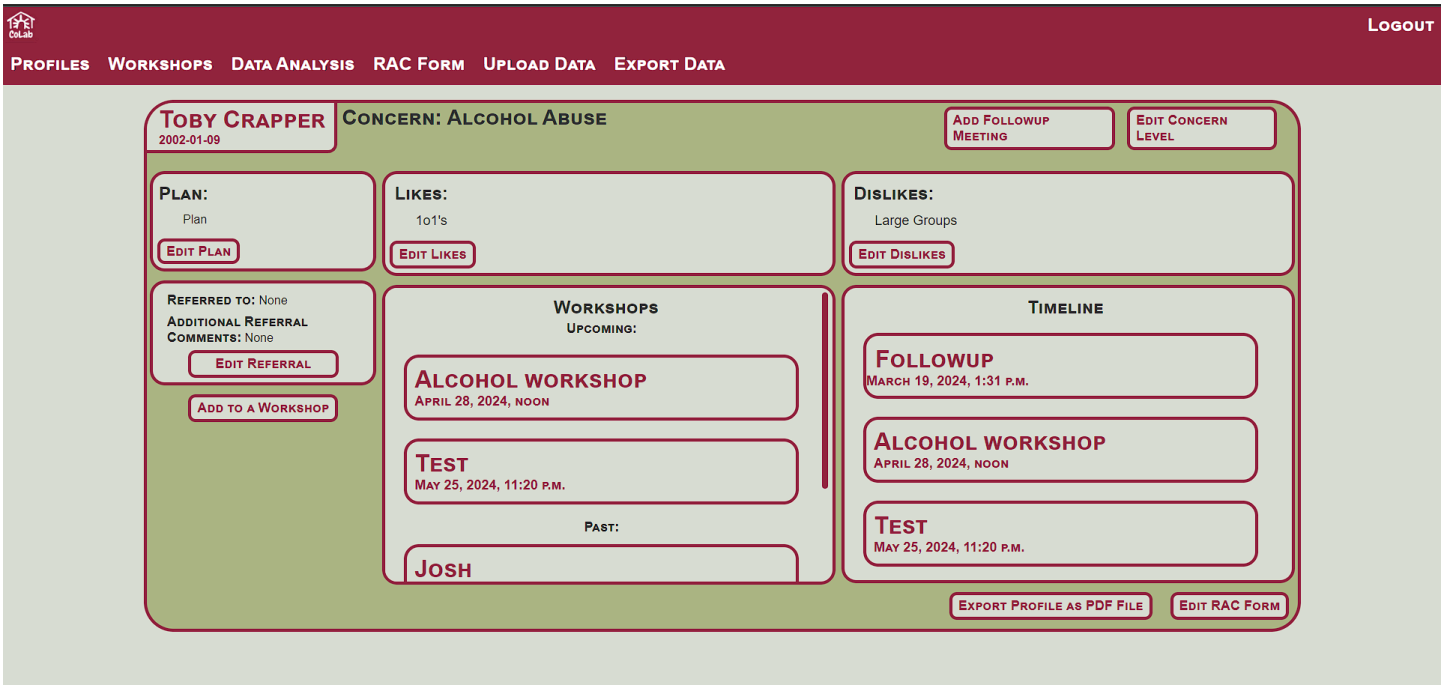


Profiles

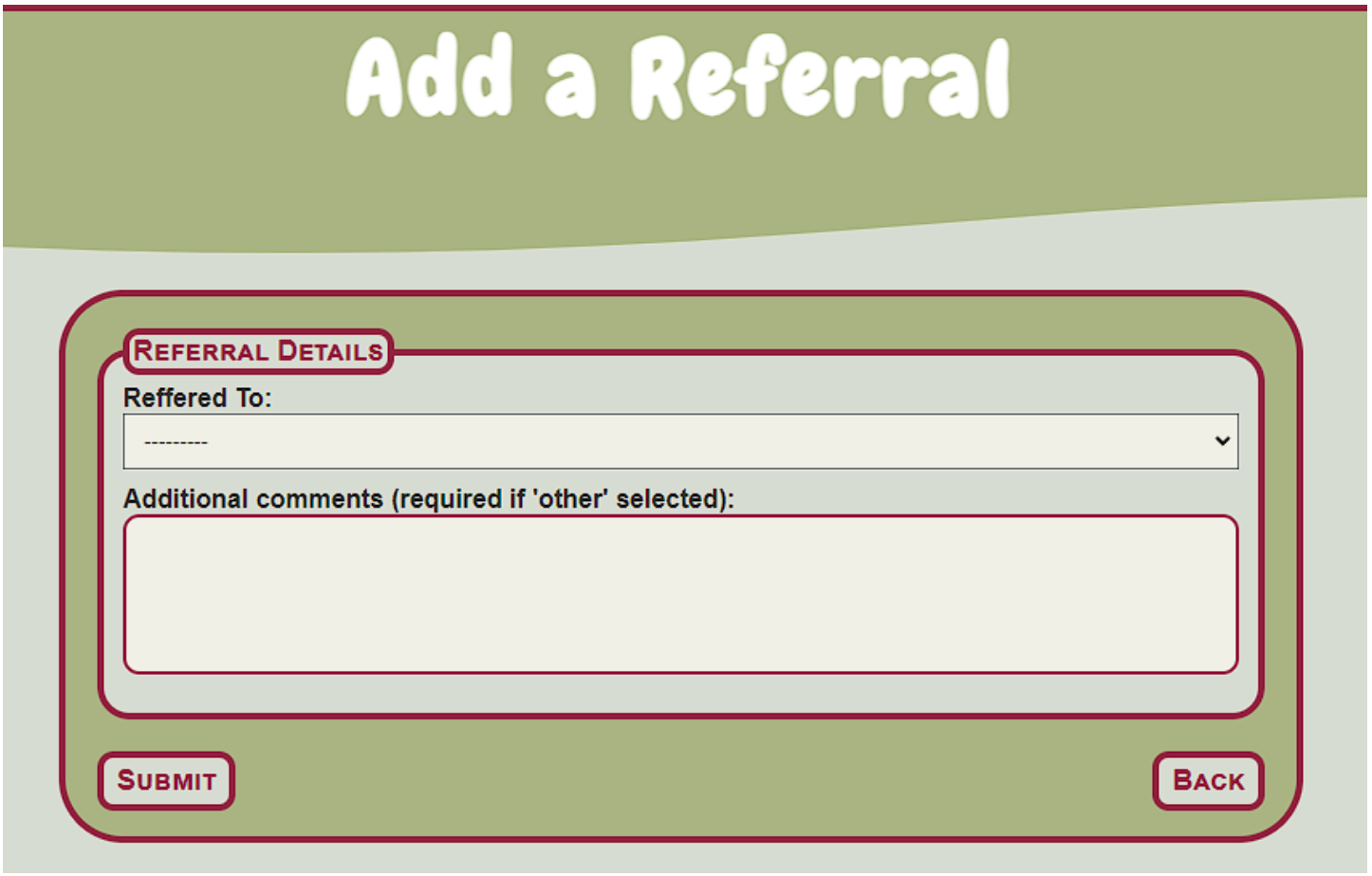
The profiles page lists all the profiles of CoLab’s clients that are stored in the database. The search bar title “Search Profiles” allows for a user to search a client by name. As each profile has an associated concern level, we also provide an easy method to filter between the profiles.



"Create profile" redirects the user to the RAC form for Profile entry. A user can also select a client to view their profile, either by clicking on their name or the "view" button. Doing so redirect a user to the profile, example below:

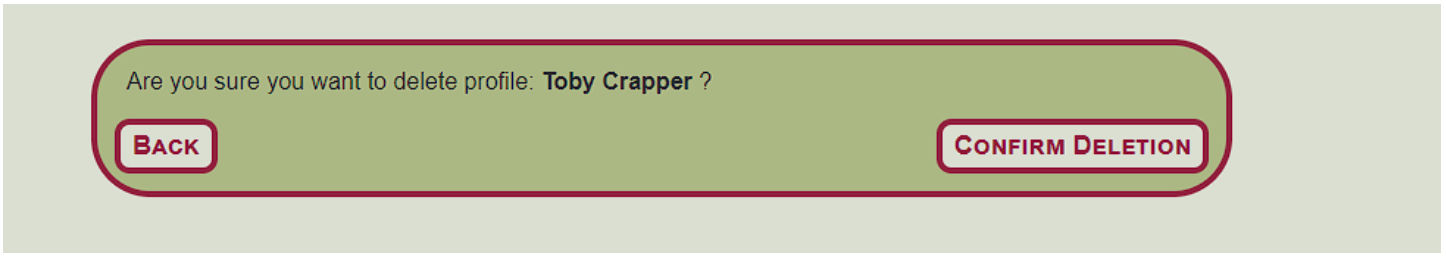


It is expected that a majority of time spent using the program will be on this page. At the top of the page we see "Concern: Alcohol Abuse". This entry is determined by the RAC form entry for the user. "Plan" indicates the current plan for the user and can be updated at any time by pressing "edit plan". The "likes" and "dislikes" sections are for user's stored likes and dislikes. The list can be edited at any time in mh/base/models.py. Next is referral data, which can be edited by selecting the "edit referral" button:



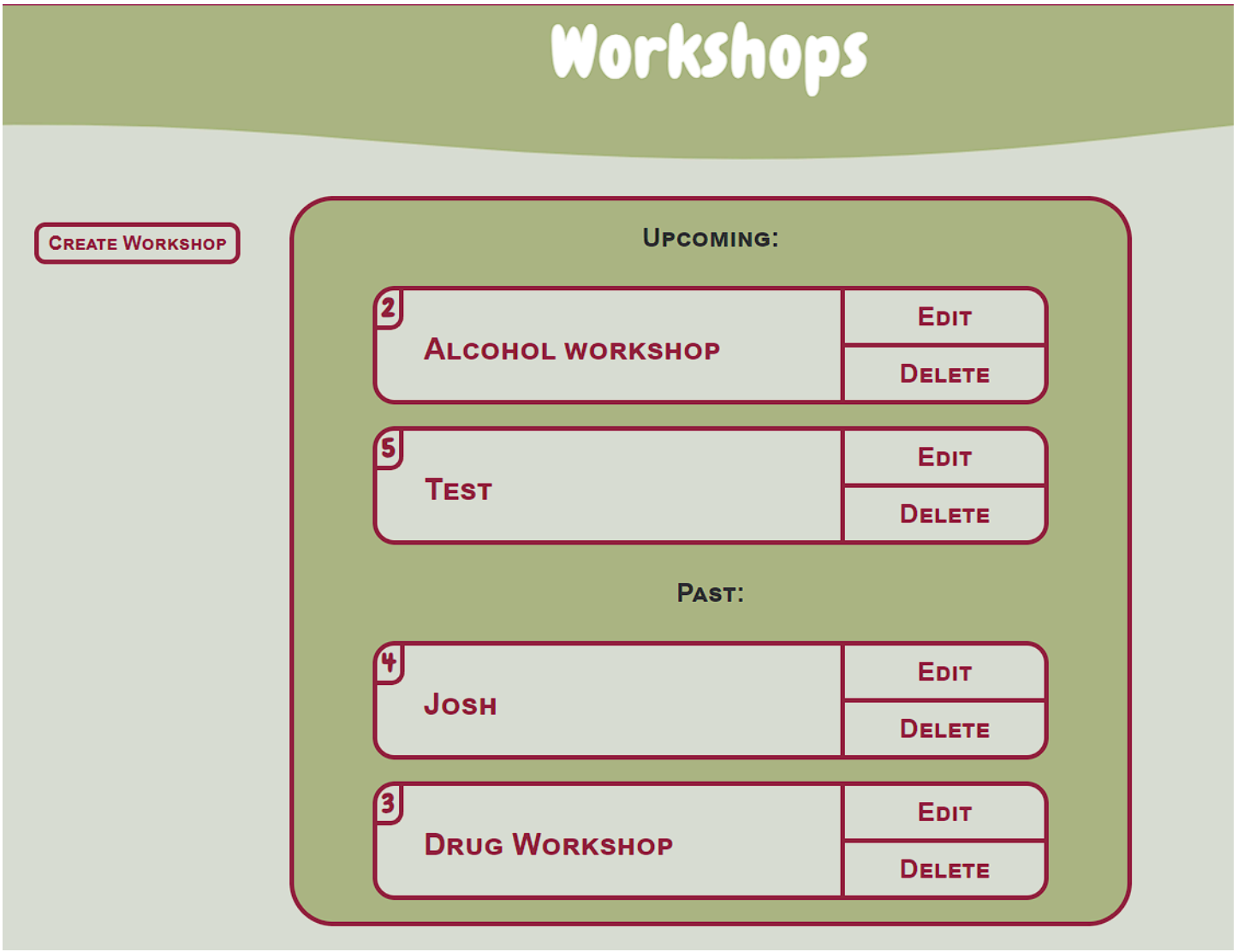
We then see workshops, which details the user's past and future workshops they are selected to attend. Lastly, we see a timeline of events. The "export profile as pdf" button will export the clients data as a pdf for viewing or sharing. "Edit RAC form" allows the user to edit the RAC form originally filled in in case any new data is available.

A profile can also be deleted by using the "delete" button, which redirects to a confirmation form, to ensure the delete is not a misclick. Pressing "confirm" removes the data entirely.

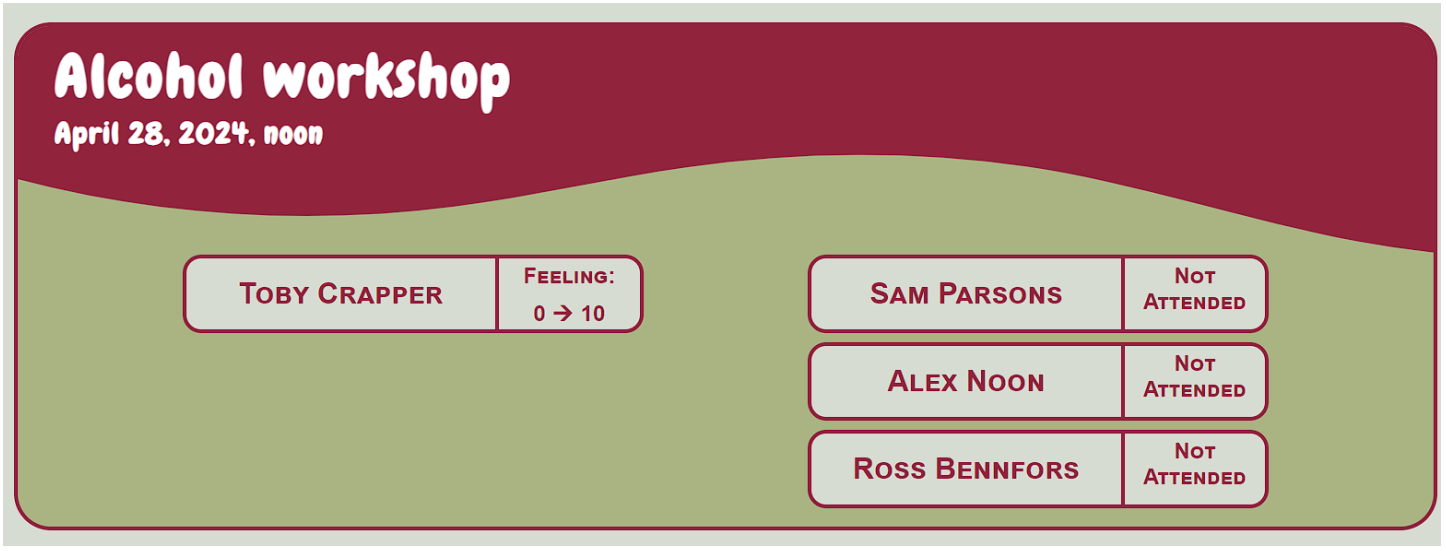


Workshops

"Workshops" follow the same style of Profiles, with each workshop being listed.



Clicking on a workshop shows the user the date and time of the event, and also the participants list.

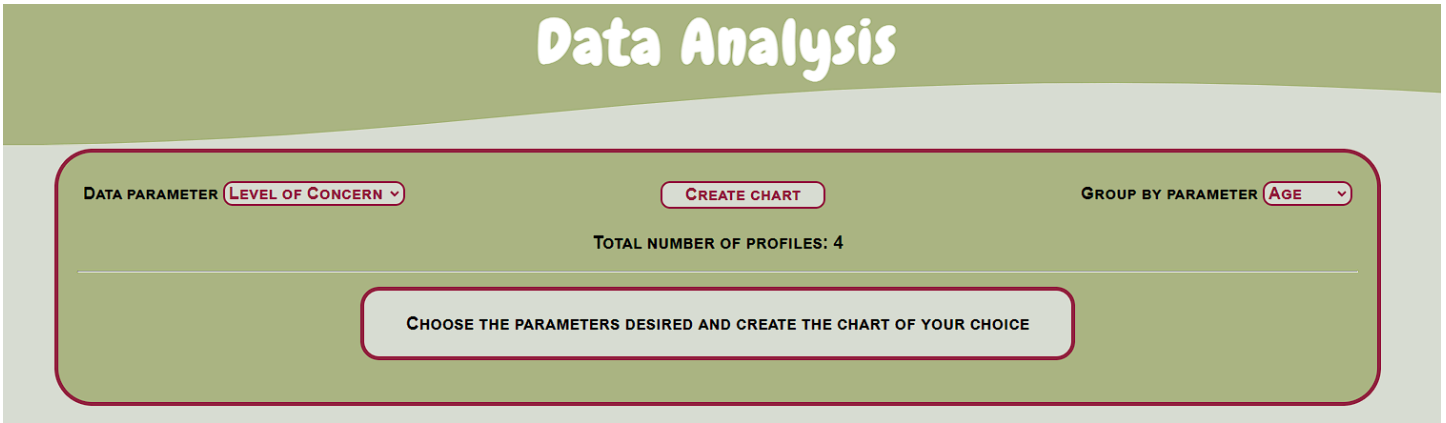


A workshop can be added using the “Create Workshops” button, which redirect a user to the following page:

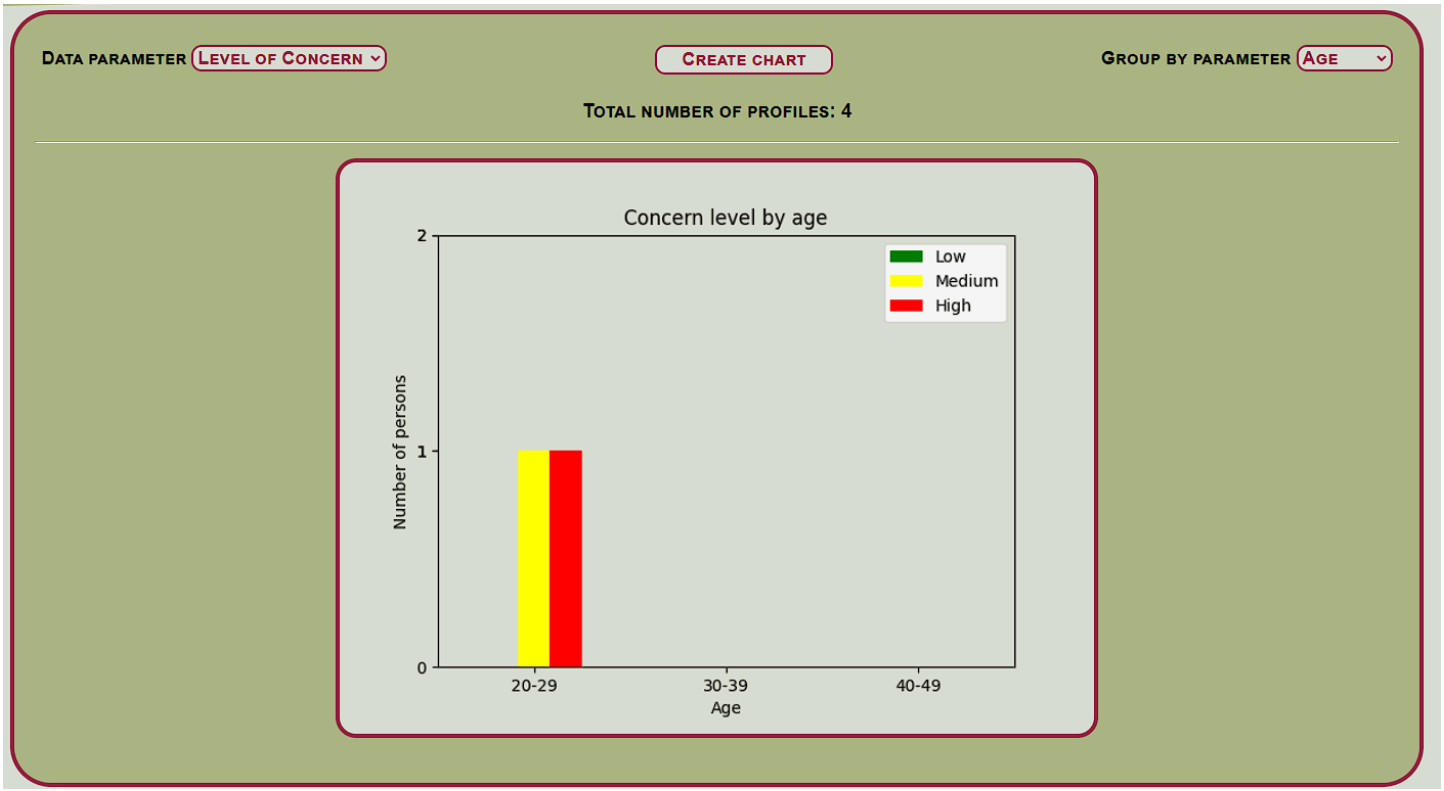
Pressing “Submit” adds the workshop to the database.

Data Analysis

The data analysis page allows for a user to choose two parameters, “Data Parameter” and “Group by Parameter”, and create a chart based upon this data.

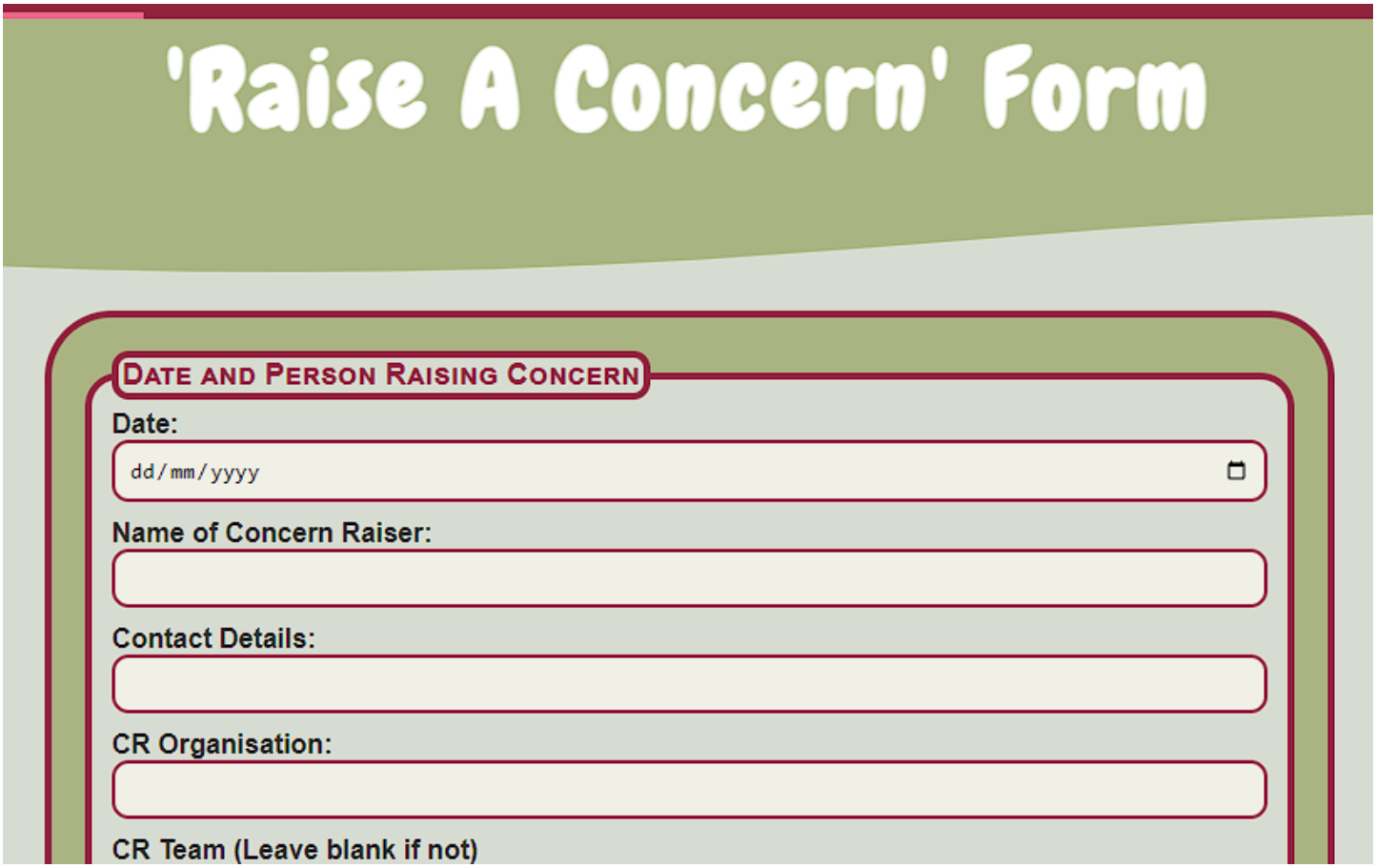


In this instance, clicking create graph to generate a graph based upon “level on concern” and “age” gives us the following:



RAC Form

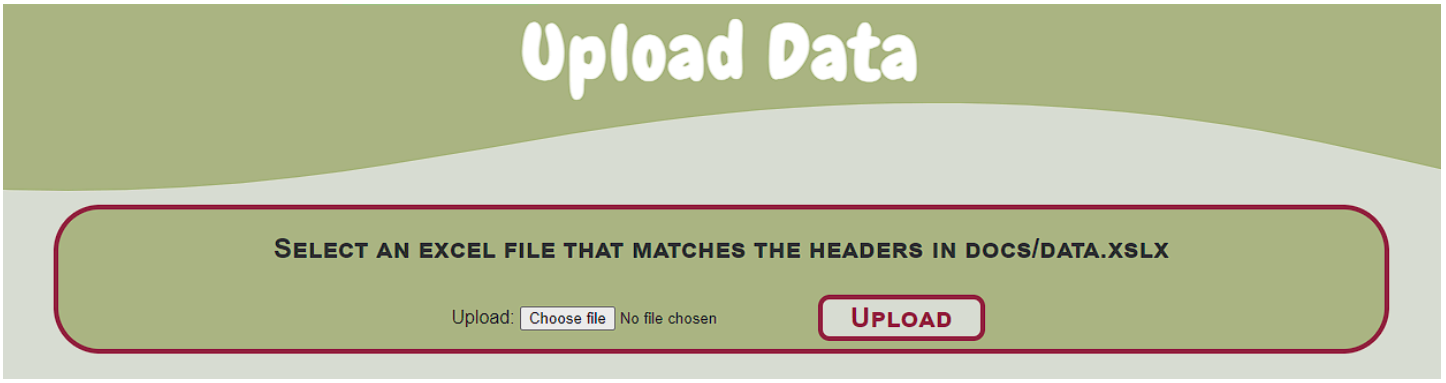
The RAC form is the main data entry point for the program, in terms of initial user data. The fields were copied directly from the physical form. Pressing “submit concern” adds the profile to the database, and the profile appears in “Profiles”



Certain fields of the form are required in order to submit the data to the database. These are “Date”, “Name of Concern Raiser”, “Contact Details”, “CR Organization”, “Concern Description”, “Consent Checkbox”, “Signature” and “Signature Date.” The rest of the fields can be left blank, where data is not available, but it is strongly recommended to add all of the available data to the database. This will ensure that the “Data Analysis” section can provide fruitful visualizations.

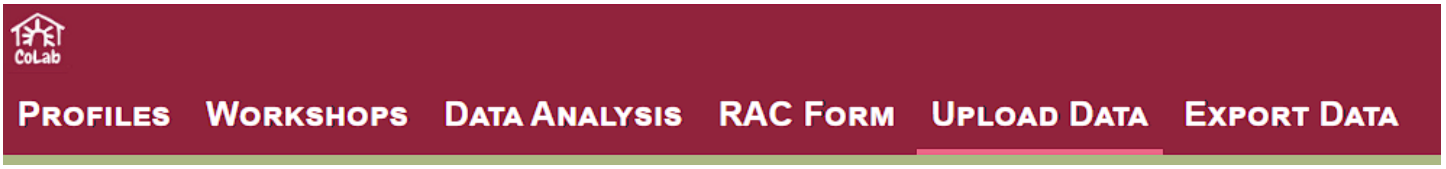
Upload Data

Upload data allows for the client to add their existing data to our modernized system in one simple step. For development we were provided an anonymised spreadsheet detailing the data available. If the real data is in the same format, the client can “upload” the spreadsheet and it will add all the profiles to the database. Note: this doesn’t “upload” anywhere as the system is run locally.



Export Data

In a similar theme, if at any point the client decides that they do not wish to use the software anymore, clicking “Export data” on the nav-bar will export the data as a csv file. Note: the client must also delete the project files in its entirety to ensure data is not stored outside of GDPR laws.



Maintenance

Maintenance for this project is low, and the design and coding approaches focused on ensuring the software would stand alone for a number of years.

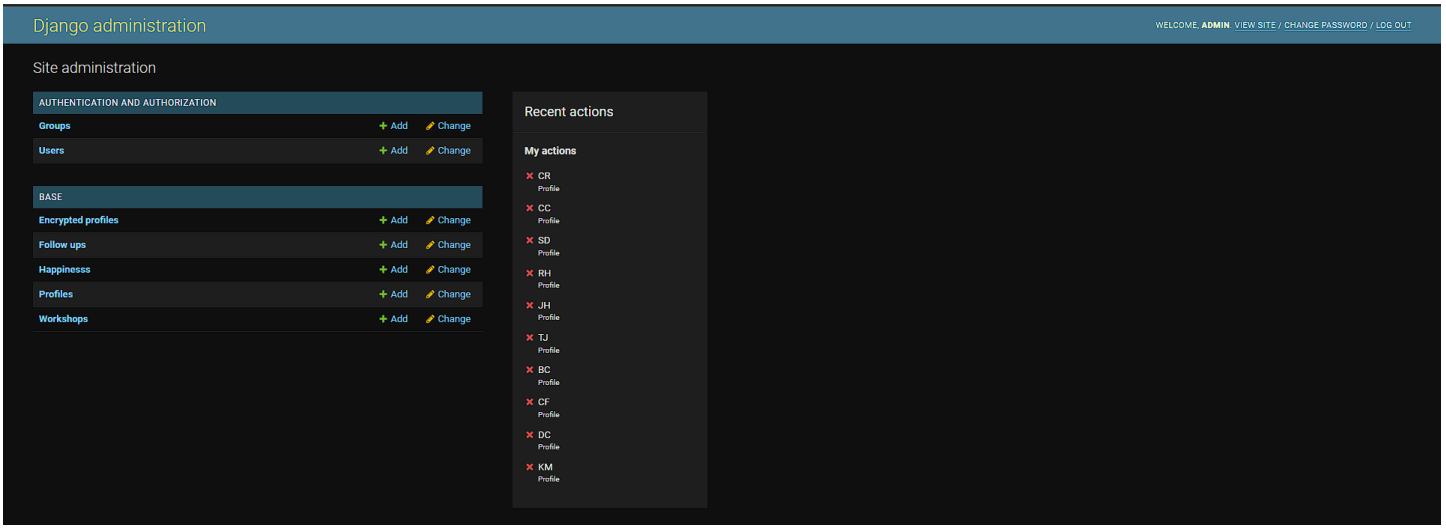
Database backups

It is recommended to back-up the database manually every 24 hours as well as every time a user requests their data be deleted. This is to ensure that a recent copy of the database is always stored, should failure occur. Due to the nature of the data, it is recommended to be backed up locally.

To restore the data, simply replace the broken database file with the most recent backup and the software will work as anticipated.

Managing users

The program is designed to be used by a single user at a time, as is the limitation of running it on a single machine. However, multiple users can be created and it's recommended to perform a weekly check that no unauthorized users have been created. This follows standard Django admin usage.



FAQ

This section details frequently asked questions, or issues that may arise during using the project, such as known bugs at the time of submission, or simply errors that we've encountered along the way

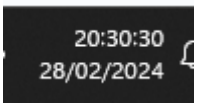
Port in Use

The Port in Use issue occurs when an localhost application is running on the same port as the django project is attempting to run on.

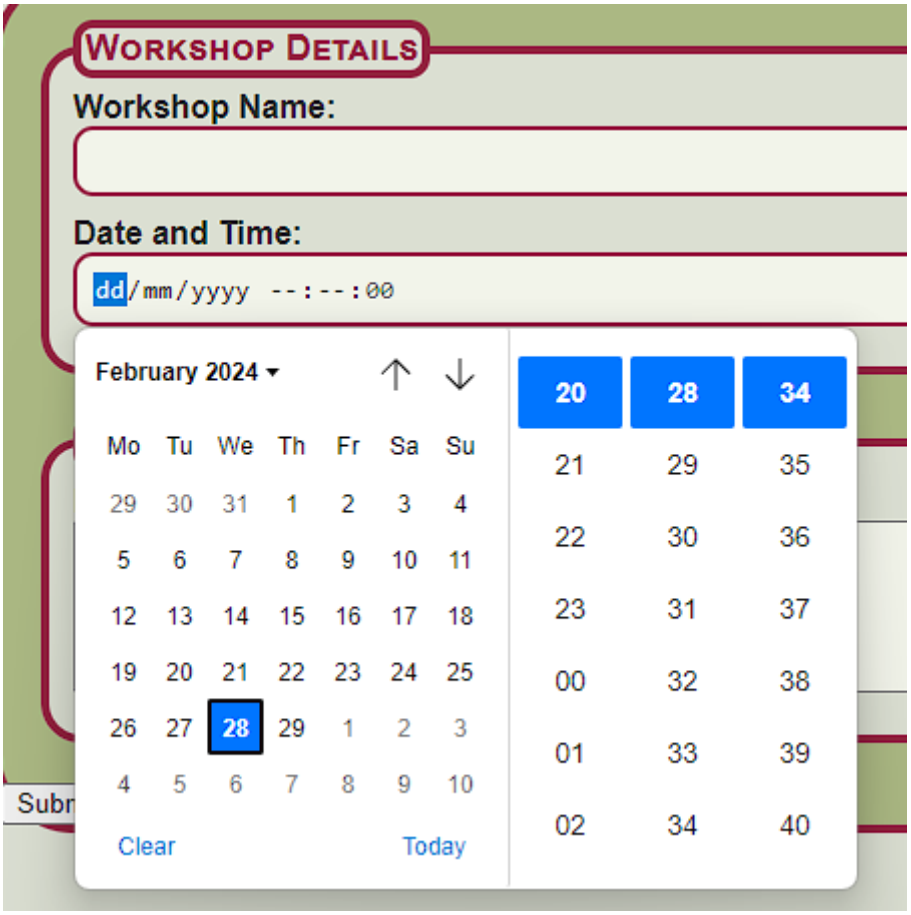
The solution is to change the port the django application is attempting to run on. This has no impact on any functionality of the program. To fix the issue, instead of running "python manage.py runserver" run "python manage.py runserver XXXX" where XXXX denotes 4 numbers of your choice. Recommended is 8080, however, most combinations of numbers will work.

Calendar Bug

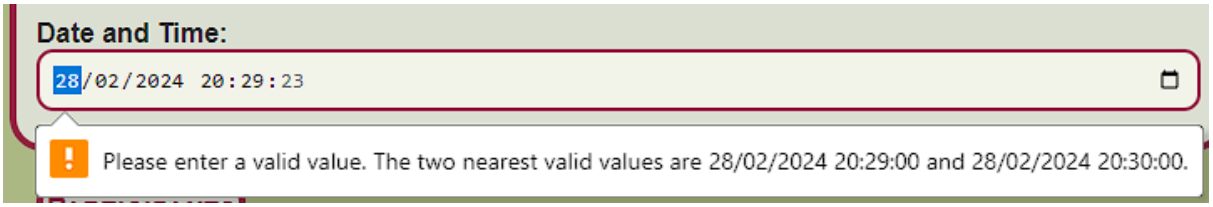
When creating a workshop, depending on the user's local Windows settings, they may be given the option to modify the seconds item in the calendar. This is the case when the user has the "seconds" options enabled on Windows, and the clock presents the seconds. For example:



This replicates itself in the Django application as the following image in the workshop page.



Attempting to add to the workshop list gives the following error:



The solution: change the “seconds” value to equal to zero (currently indicated as 23 in the example photo). This will solve the issue.