CS 5200: Database management systems

Project Report

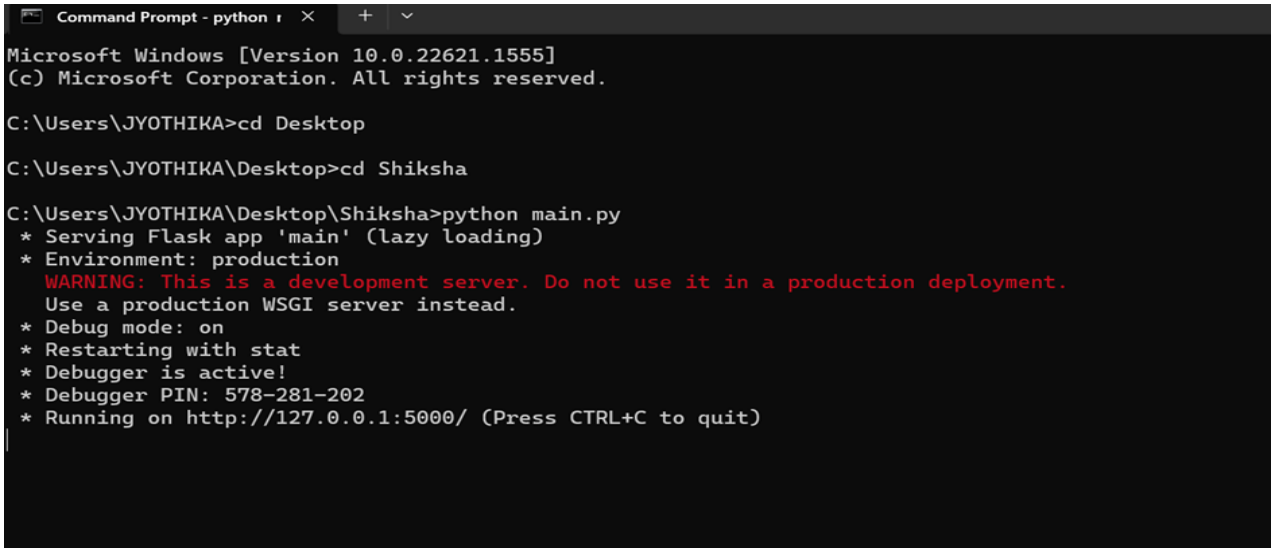
Group name: NittalaDJallaJ

Group members: Jyothika Jalla, Devi Phani Sravanthi Nittala

README

The application requires the python and any suitable IDE that supports Python. The below guide details the usage of Python with VSCode, however the application can be run on any other platform that supports Python as well.

1. Download visual studio code https://code.visualstudio.com/download
2. Install python from https://www.python.org/downloads/ and download the latest version.
3. Open Visual studio and upload the Shiksha folder into the studio.
4. Install the python extension in visual studio if any error is displayed.
5. Install following packages in the command prompt
   - pip install flask
   - pip install pymysql
6. After running the program in the visual studio if there is any error showing
like "'flask module error" even after installing flask in the command prompt. In case of this issue, it can be sidestepped by using the command prompt directly . Open command prompt and navigate to the location of the Shiksha folder.
7. After navigating to the path run the command "python main.py" in the command prompt.
The user is prompted to enter the database username and password in the console itself.
Should error arise, the user is prompted once again to enter the login credentials for the database.
After entering the command, a url is generated. Refer to below image for reference.



8. Copy the URL http://127.0.0.1:5000/ on the chrome browser and paste it.
9. Use the login credentials to login and continue accordingly.

    Admin login details examples:

        1. username: mdavid

           password: pass1

        2. username: apatel

           password: password

    Users can be created via admin profile, but existing user profiles include

        1. username: smehta

           password: pass1

Admin users, new shelter homes, cities, details of children must be directly in the database and cannot be done through the application. This was done to ensure more control over the data, in the future we plan to introduce more secure ways of doing this through the application itself by introducing more authentication for admin users.

Introduction

Shiksha, our application, is used to represent a non-profit organization that consists of volunteers who provide free tuition to children living in government-run shelter homes. The organization teaches children in the age groups of 5-18 years and covers both mainstream subjects taught at schools as well as life skills and career preparation classes. It aims to provide a holistic education and help them be on equal footing with their peers who have access to better resources. This project is a volunteer dashboard application where the volunteers can get information regarding classes, donations and their profile.

Technical Description

We intend to use a relational database **MySQL** to build our application. The front end of the website is built using **HTML5 and CSS**, which provide the structure and visual design of web pages. To enhance the website's functionality, **JS** and **charts.js** library will be utilized for scripting and creating interactive visualizations. Additionally, **Python** will be used as the primary scripting language for back-end development. We will be importing mysql.connector python library to work with the databases. Finally, the **Flask** framework will be employed to provide a web application framework for Python and to integrate the front end and back end components of the website. We will be implementing our database using the **MySQL Workbench** application.

Database Description

The database consists of information about the staff working at the organization. All staff members have a unique ID, name, age, email address, a unique username, password and date of joining. The staff members may either be volunteers or full time employees.

Full time employees have a defined salary and department. Each department has a unique name. Departments may consist of zero or more volunteers.

Volunteers work in a single department and are associated with a single shelter home in their city. They conduct classes at their shelter home or may substitute at other shelter homes when the need arises.

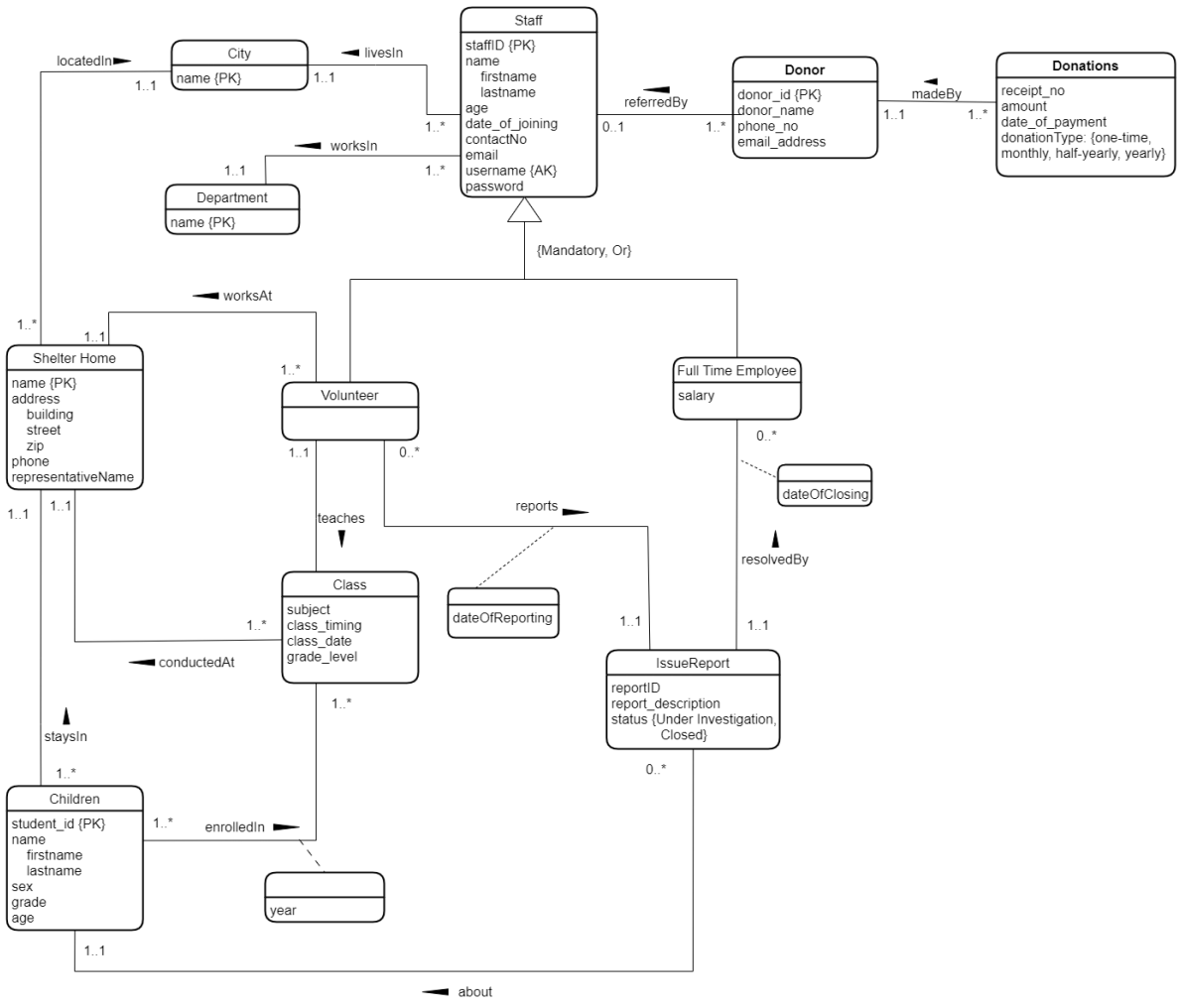
There are multiple shelter names in each city, each of which have a unique name, an address, the contact number, and the name of the government representative who oversees the running of the shelter home. There are multiple classes conducted in each shelter home that are taught by the volunteers and attended by the children in the shelter home. Each class is for a specific subject taught on a particular date and time. The classes are aimed at specific school grade levels.

The database also consists of the children staying at the shelter home and enrolled in a teaching program for a particular academic year. The children have unique student IDs, names, age and sex.

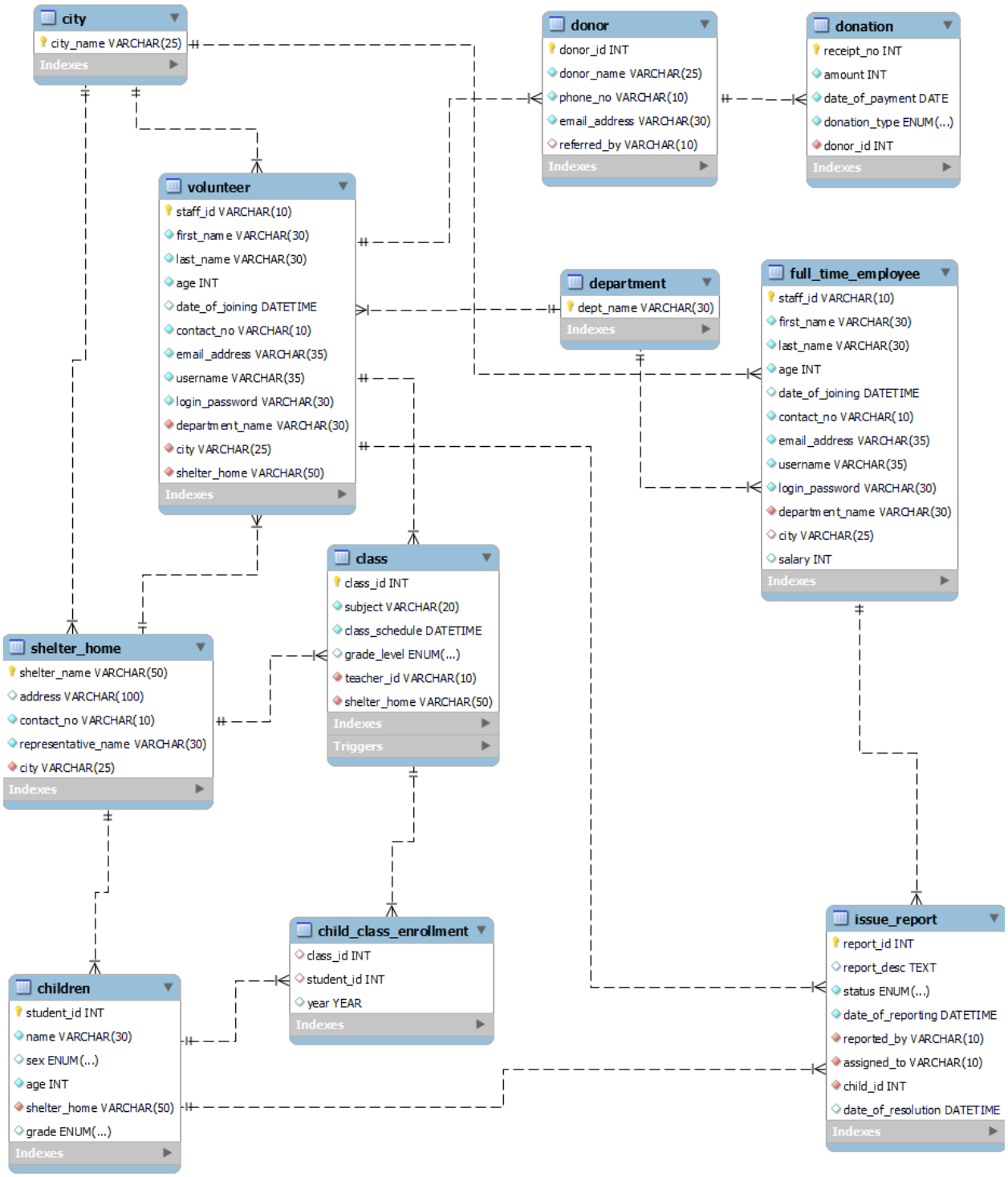
Keeping in tune with the child-centric approach of the organization, the volunteers can report any issues that they feel a particular child may be facing, this includes potential cases of neglect, abuse, etc. that the child may be undergoing. These issues are reported by the volunteers and resolved by a full-time employee. Each report has a report ID, a description and a completion status which may be under investigation or closed. This functionality ensures that the children are giving prompt and effective care in a transparent manner.

In order to maintain the day-to-day operations of the organization and provide resources such as books, stationary and technology to the children in the shelter homes, the organization collects donations. The donors may be brought in by a volunteer or may have heard of the organization through other channels. Each donor has a unique donor id, name, phone number and email address. Each donation is tracked via a receipt number, amount, date of payment and information about whether it is a recurrent donation or a one time donation. Recurrent donations may occur monthly, half-yearly or yearly.

UML Diagram



Logical Design



Functionalities

The main aim of this project is to simplify different activities done by the user and admin in an efficient way rather than storing information in an unorganized way which makes it complicated to access data and often leads to cascading effects in the organization's working.

The application is accessed by users who are volunteers in the teaching staff and admin who are also volunteers given certain admin rights to add/update/delete users. Admins may also be full-time employees.

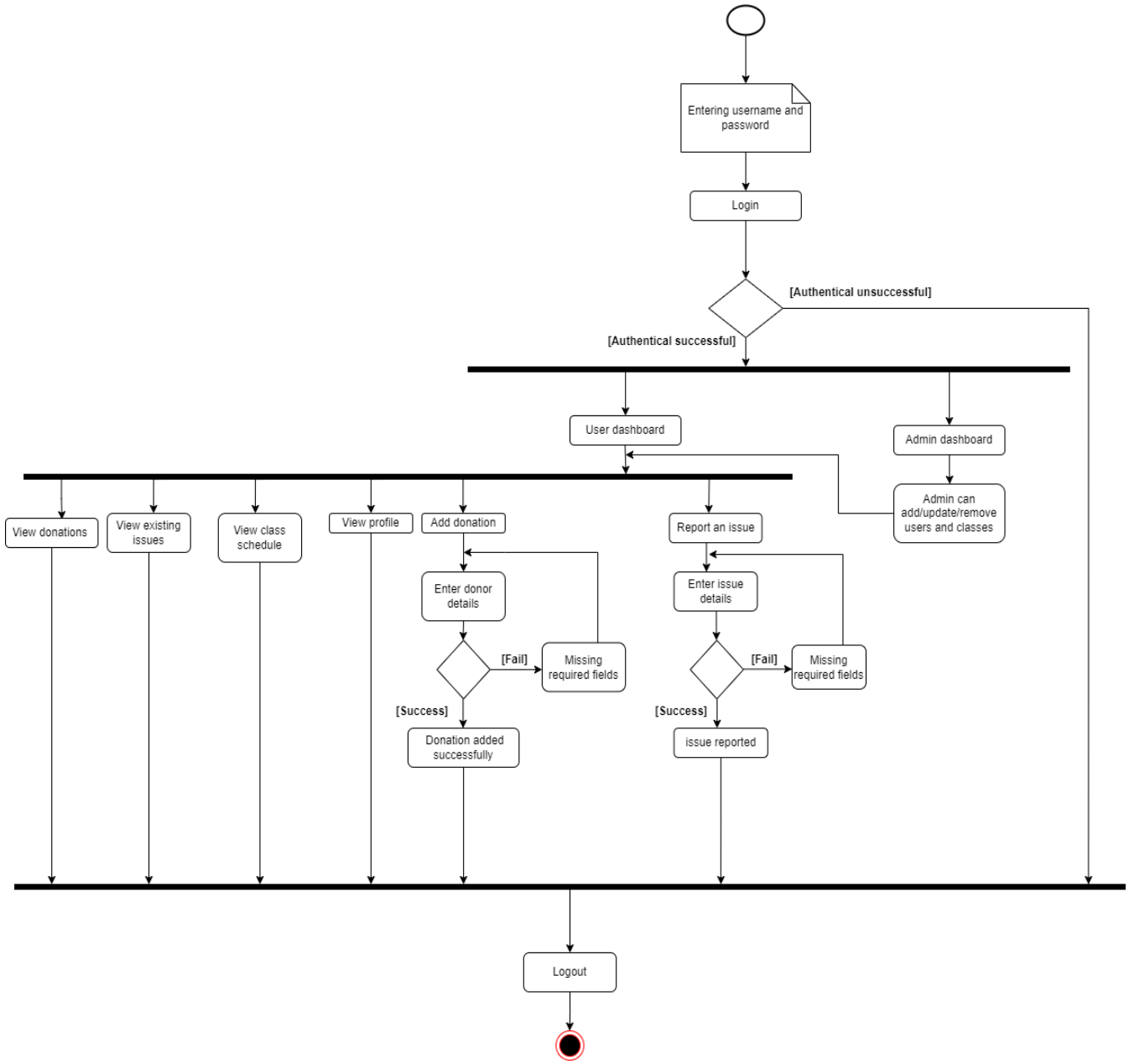
Admin : As an Admin is also a volunteer, admin can do all tasks that a user can. Additionally, Admin can:

1. Add a user.
2. Update information regarding the user
3. Delete or remove access from the application.
4. Create or remove classes
5. Resolve any reported issues assigned to them. The issues are assigned to any full time employee who is in the same department and city as the reporting employee.

User :

1. Users can view their profile which includes their name, Department/Program, and date of joining.
2. Users can add donations sourced by them by filling out basic information of the donor which contains donor name, amount to be raised, donation type which could be one-time donation, monthly or yearly.
3. Users can report an issue if any of the children is facing any sort of issue. The user has to fill out required information which is the subject and can also fill out a description which is an optional field if the issue is to be described in detail. This issue is tackled by full-time employees.
4. Users can view the issues reported so far with reportID and the status of the issue whether it is under investigation or closed.

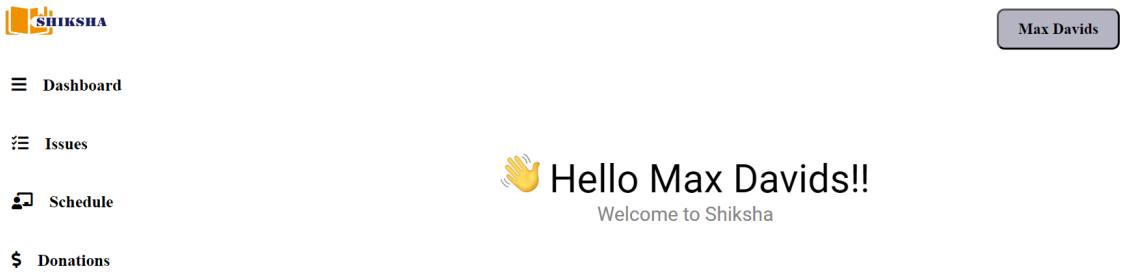
Activity Diagram



Using the application:

1. Login with credentials. For incorrect login, a pop appears showing that the credentials entered are incorrect. This appears as a separate pop up window in the taskbar.

2. On the first page (dashboard), there will be a menu on the left with the options as shown below in the screenshot. Click on the desired option to navigate to that page.



3. In the classes section, a list of all the upcoming are listed for all users. If user is logged in as admin, there is an additional option to Add Class, this creates a new class in the schedule.

4. In the report issues page, users can view or report issues they or any child they are teaching is facing. Admins can resolve and update the status of open issues.

5. Users can view and add donors and donations that the organization received.

6. Users can view their own profile by clicking on the profile button in the top right corner. Admins will have an additional option of Manage Users which is where they can add/remove/update users that can access the application.

Lessons Learned

Technical Expertise:

Completing this project has helped us gain a better understanding of the process of designing a database from scratch.

We learned how to work with frameworks such as Flask that helped host our application. We gained a lot of insight into what kind of data can be exposed in the code and what best approach we have to take to ensure that there is minimal amount of SQL querying done in the host language program, in our case the Python program.

Insights:

This project helped us learn how to allocate the proper time to different portions of the project. While there were certain aspects of the project that could not be completed concurrently, we

managed to divide the work and set up a timeline that ensured to the best of our ability that the delay of one section doesn't cascade forward.

This project also highlighted the need for more secure ways to store sensitive data, in the case of our chosen domain for example, it is important to ensure that the personal details of the children and minors remains inaccessible. While this was something we were unable to incorporate fully into our project, this falls firmly under future work we need to implement for a more secure system. For now, we have just made sure that no personal details of any child is directly accessible from the application interface.

Future Work

The database can be expanded in such a way that it not only supports inclusion of different cities but can also be applied across different organizations which have a similar organizational structure. This would mean supporting atleast one entity that represents the organization as a whole.

Future functionalities would include supporting the following features:

- Allowing the teachers themselves to raise and accept substitution requests for classes.
- Tracking the progress of students in class to allow for more specialized attention where required.
- Support for lesson plans and class notes.

Citations:

"Children Images – Browse 15,647,438 Stock Photos, Vectors, and Video." *Adobe Stock*, https://stock.adobe.com/search?k=children&asset_id=275015611.