ConnectHear Database Management Service

Project Report

Project Contributors:

- Arhum Ishtiaq [ai05182]
- Owais Bin Asad [oa05007]

Overview:

ConnectHear is a social entrepreneurial start-up with the aim to connect the deaf community with the society using Sign Language training classes and interpretation services – in-person and via video call and aims to make a positive difference in the lives of millions of Pakistani deaf individuals.

This DBMS aims to provide a single-user the ability to add records to the database and search for records as well. The web-app has been designed to incorporate elements of minimalism and simplicity while ensuring maximum functionality.

A login control has been established as well ensuring security of the database and keep intruders away from sensitive company records.

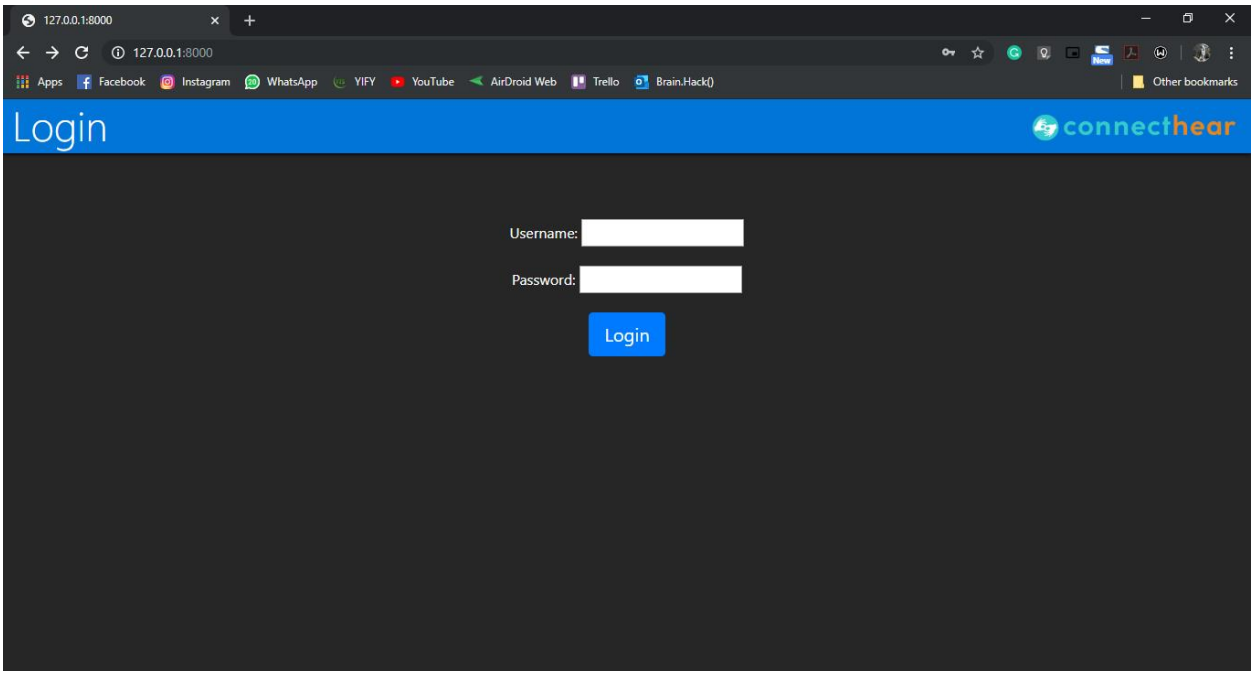
Technical Details:

The backend of this web-app is written in Python. The frontend is written in Django which is a web-framework for Python. This Python-centric design allows seamless linking between the two components without any need for parsing data.

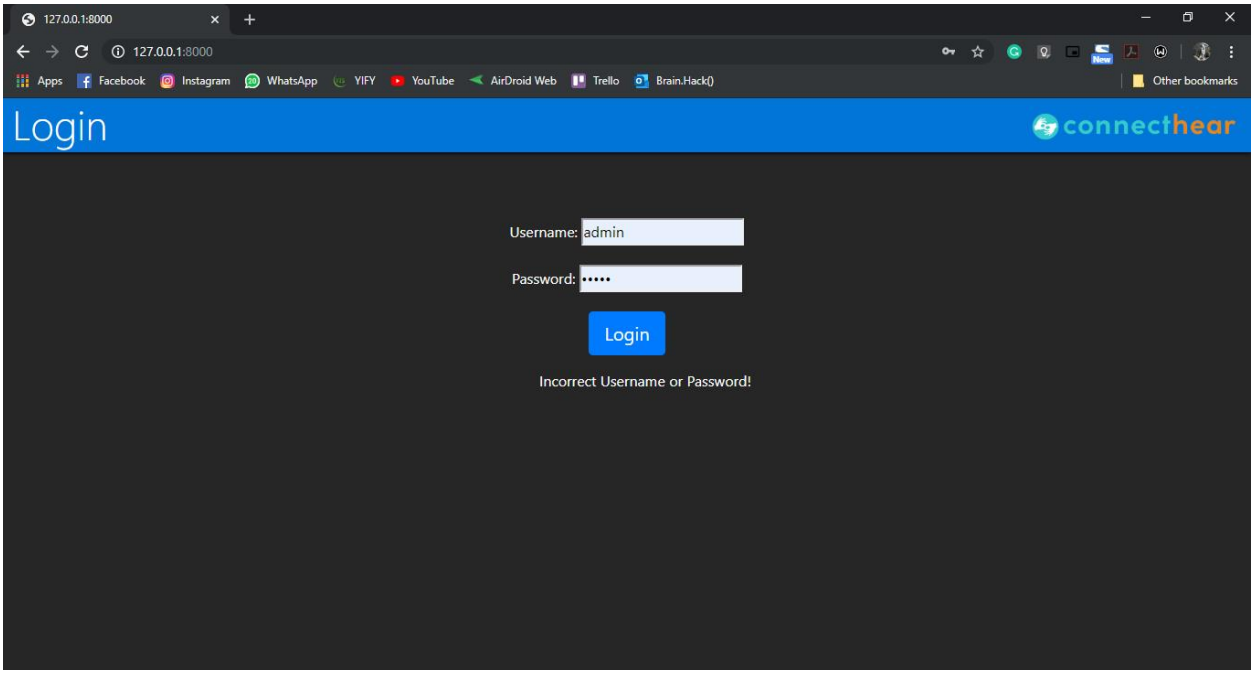
The project has been compiled in Python 3.8 and uses Django 2.2.7. The database used on the backend is SQLite 3.

User Interface:

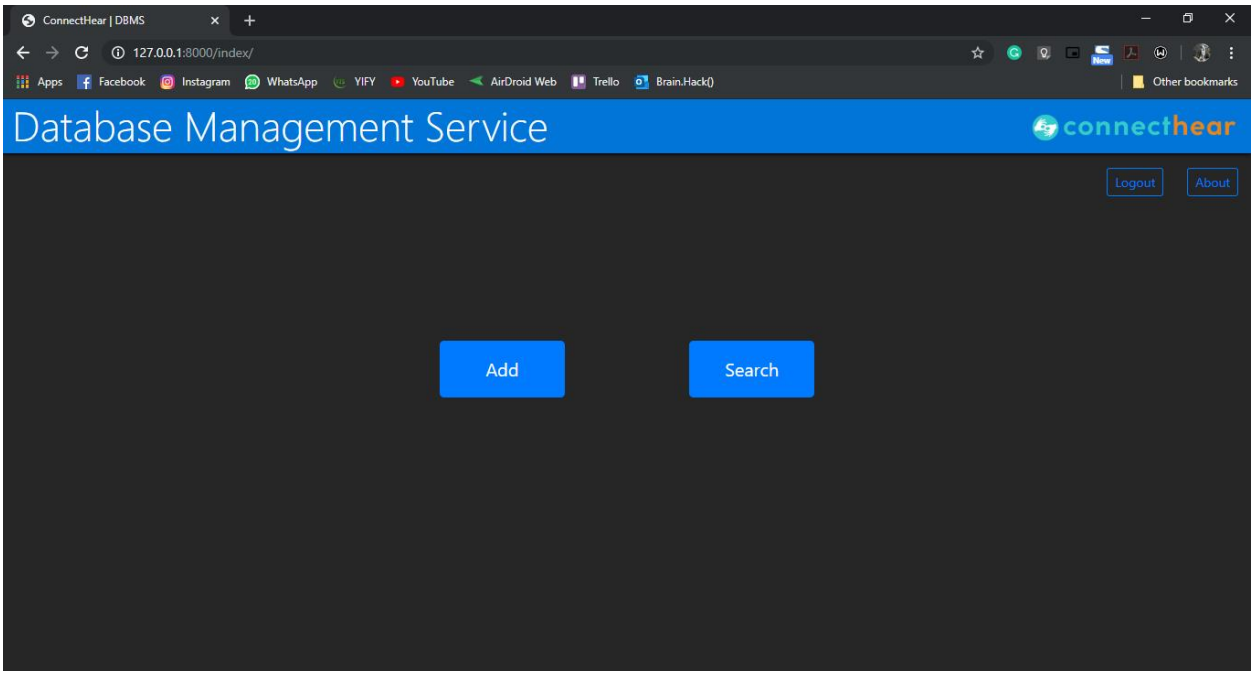
1. The user is presented with the login page as the landing page. The credentials are “admin” for username and “admin” for password by default.



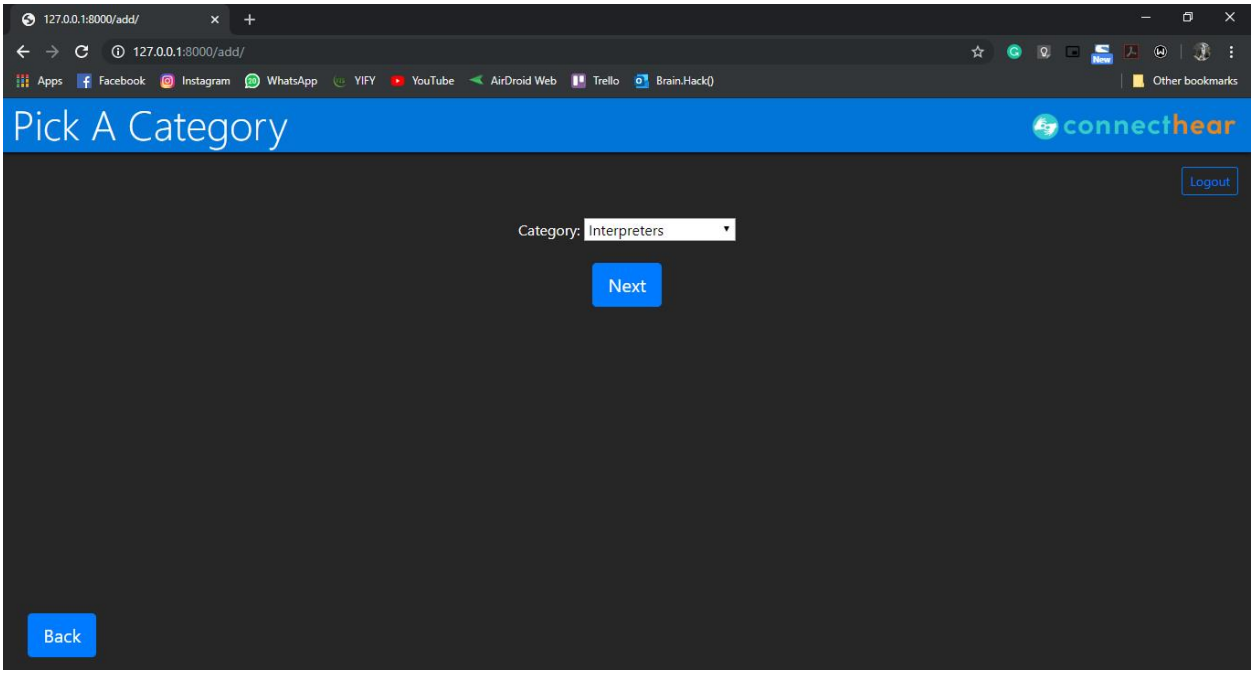
2. If incorrect credentials are added, the user is informed there and then.



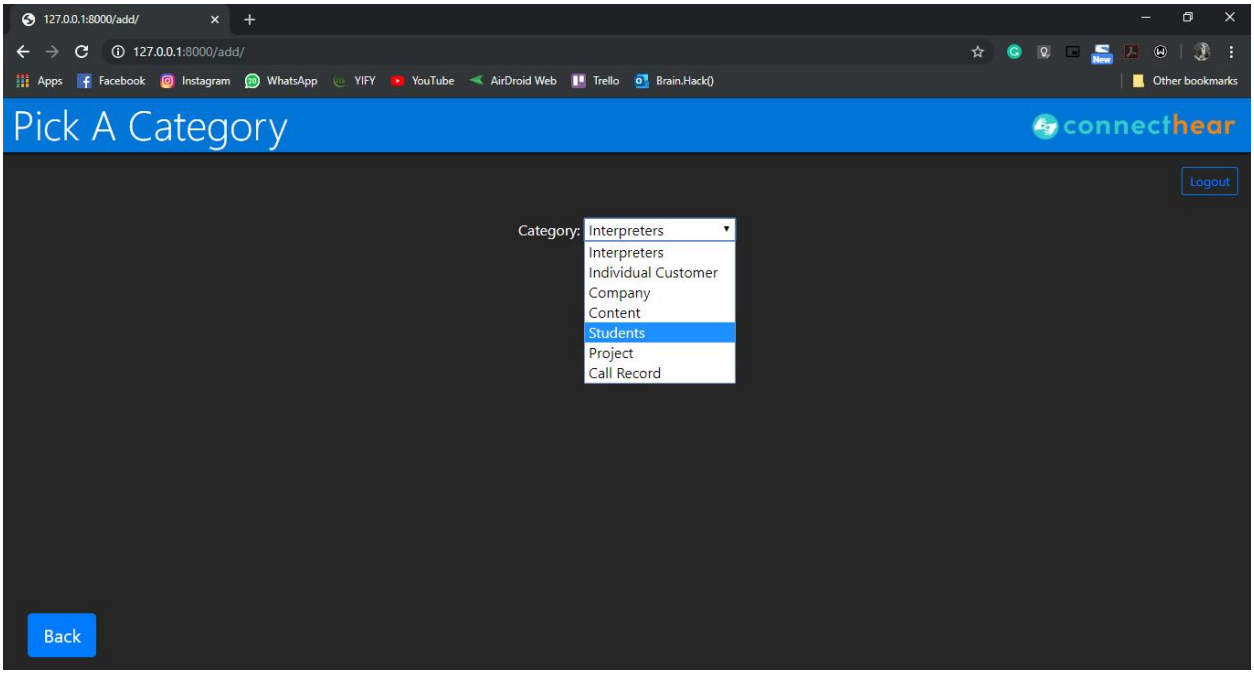
3. Upon entering valid credentials, the user is taken to the main page where they can opt between adding a new record or searching for one.



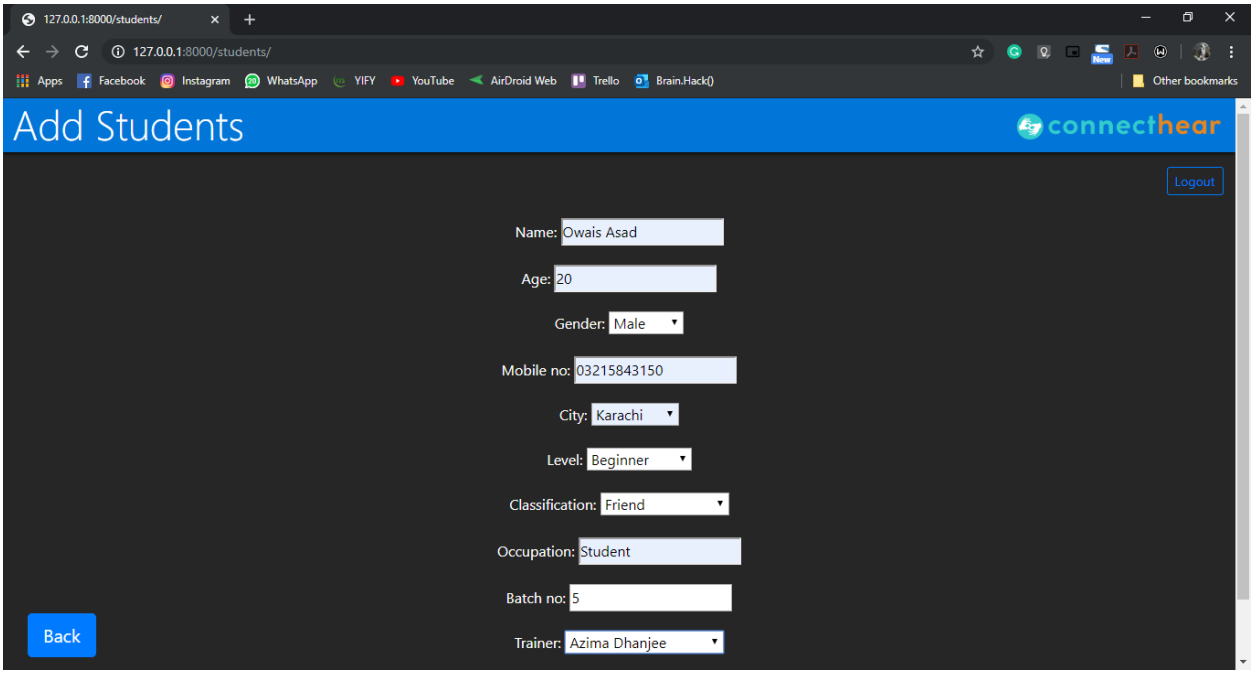
4. If they opt for adding, they have to choose a category in which they'd like to add a record.



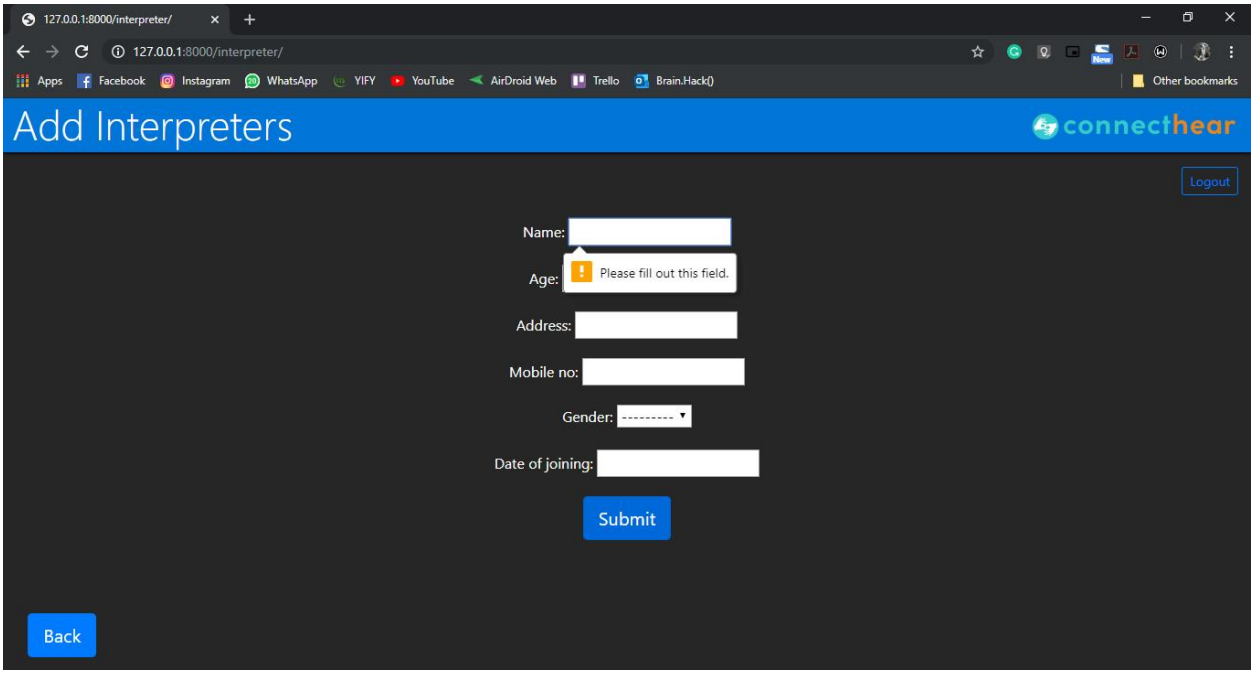
5. Each category corresponds to a separate table in the database.



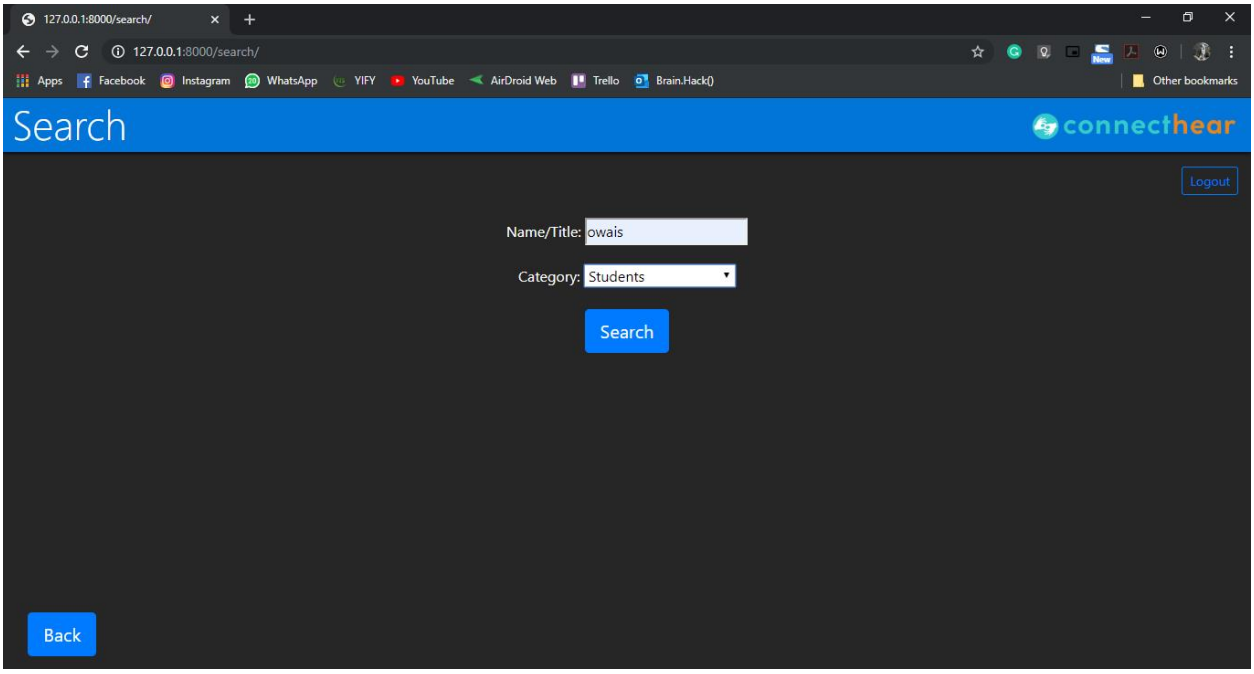
6. The user has to enter information in accordance with the data validation rules imposed for each form.



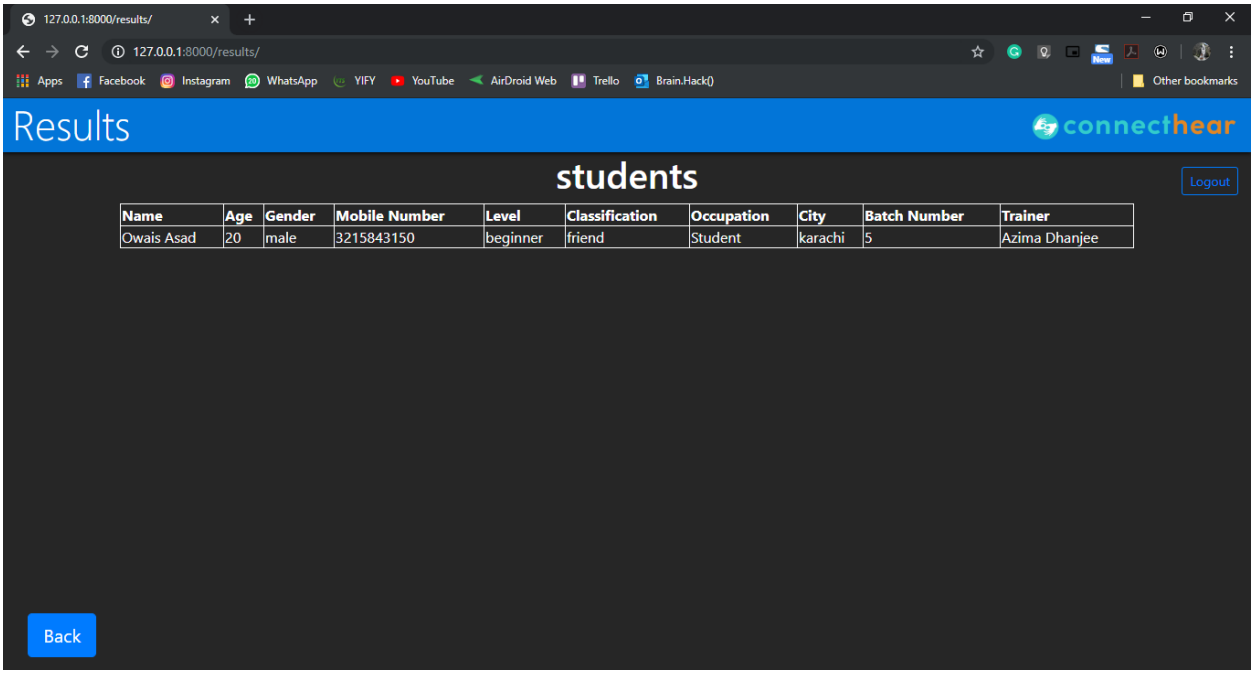
7. Failure to do so results in a warning message.



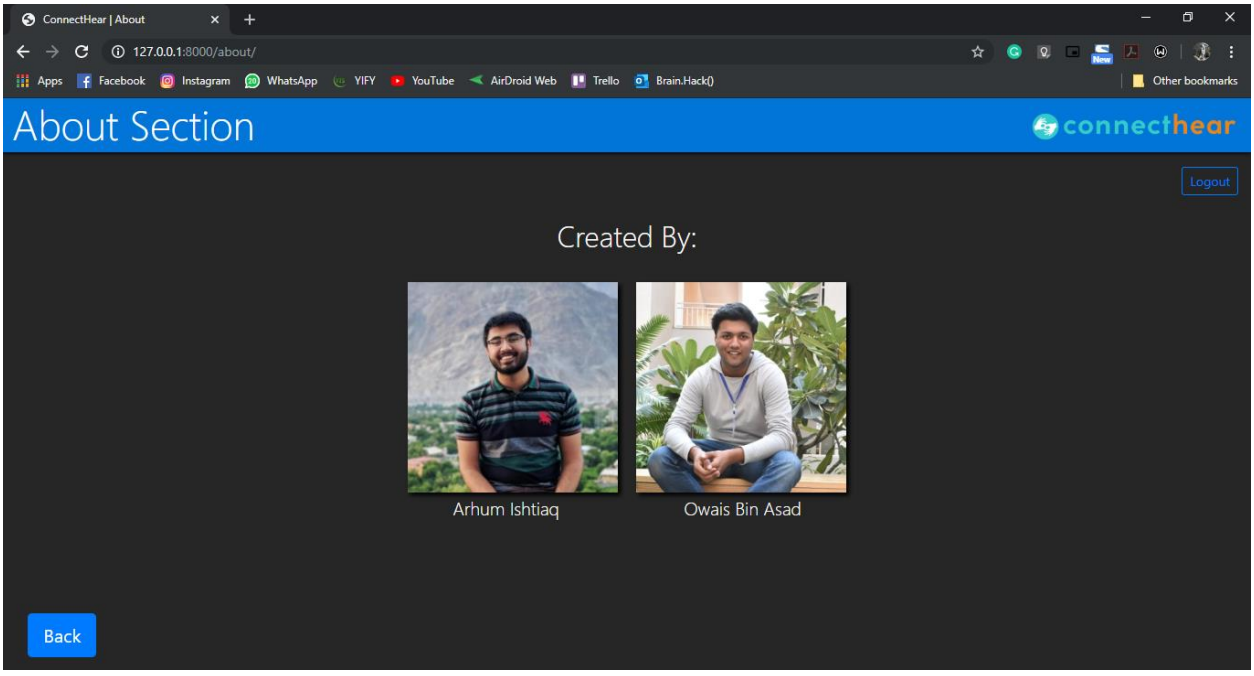
8. The user can also search for records. This sample shows a search for the same record that was previously added. Searches are conducted by name or title. The search algorithm is not case sensitive and will return any records which have the passed string in them.



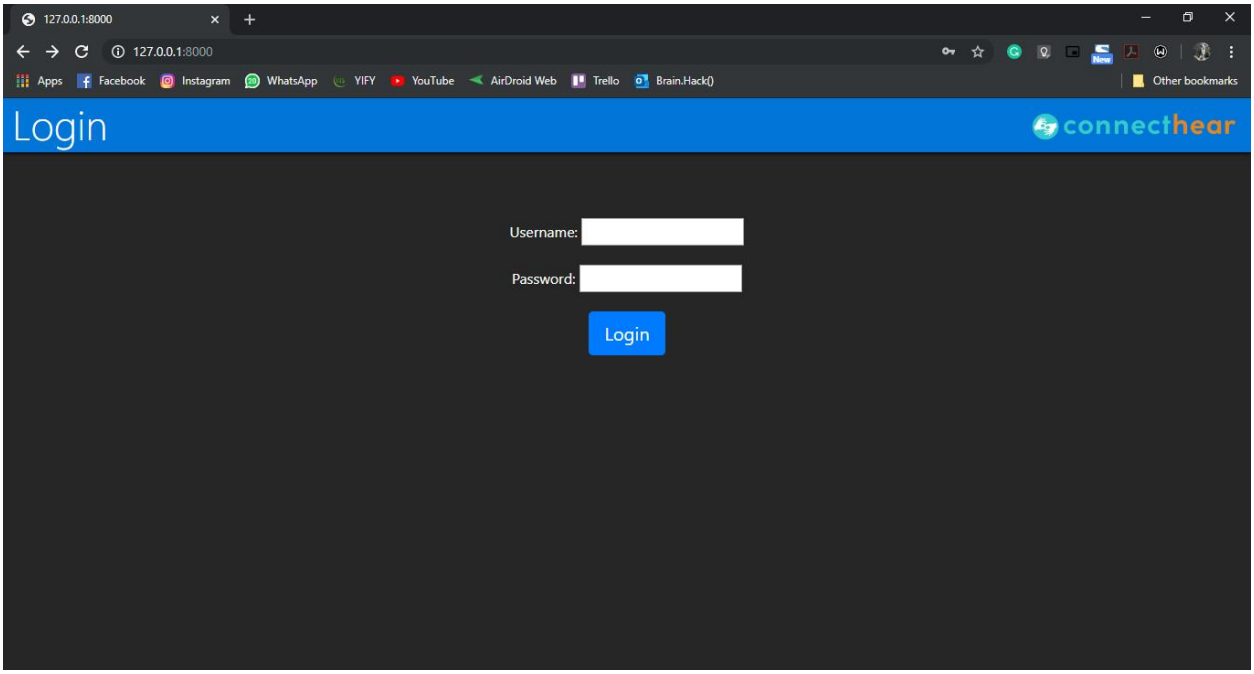
9. The results are displayed in a tabular form.



10. There is also an about section.



11. The user can click on the logout button from any screen to be logged out and redirected to the login page.



Queries:

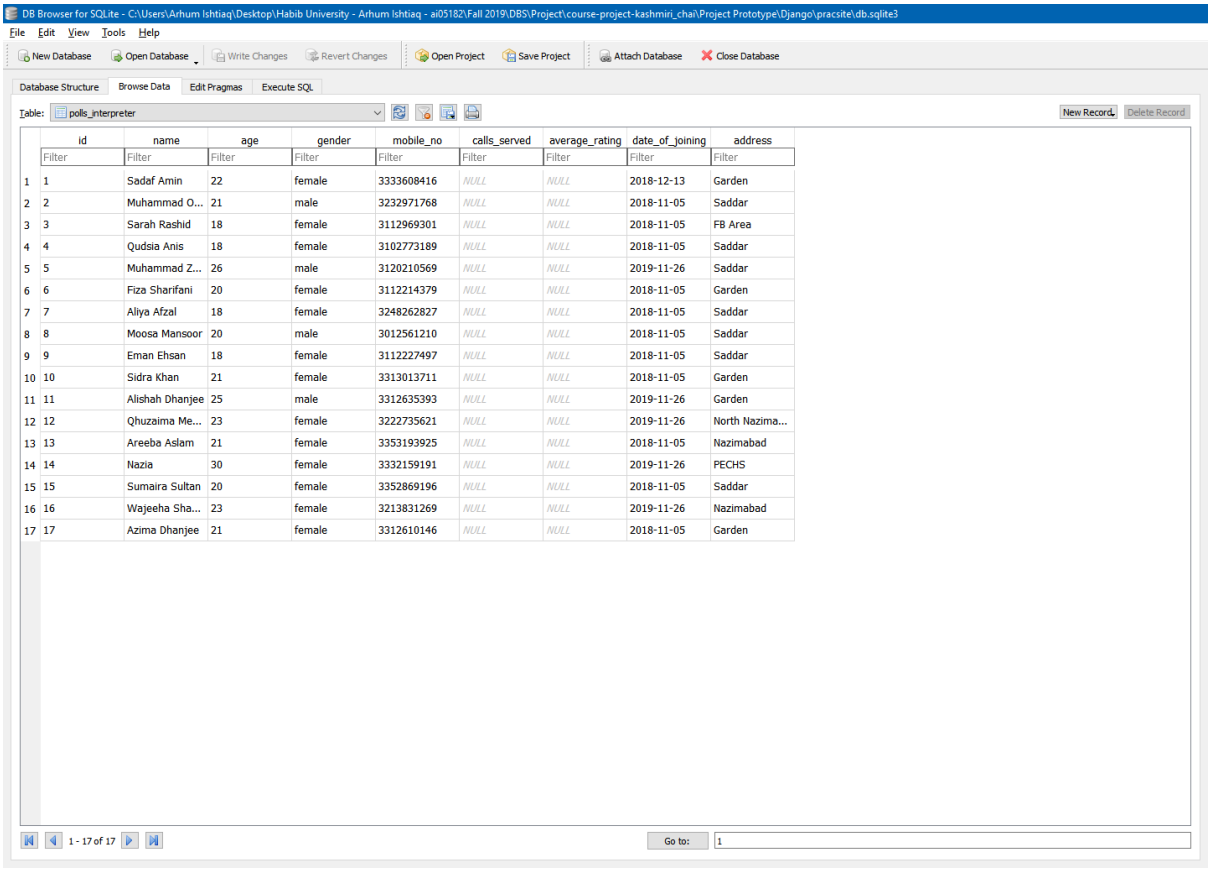
For addition of records:

INSERT INTO *@tableName*
VALUES (*@inputFields)*

For searching:

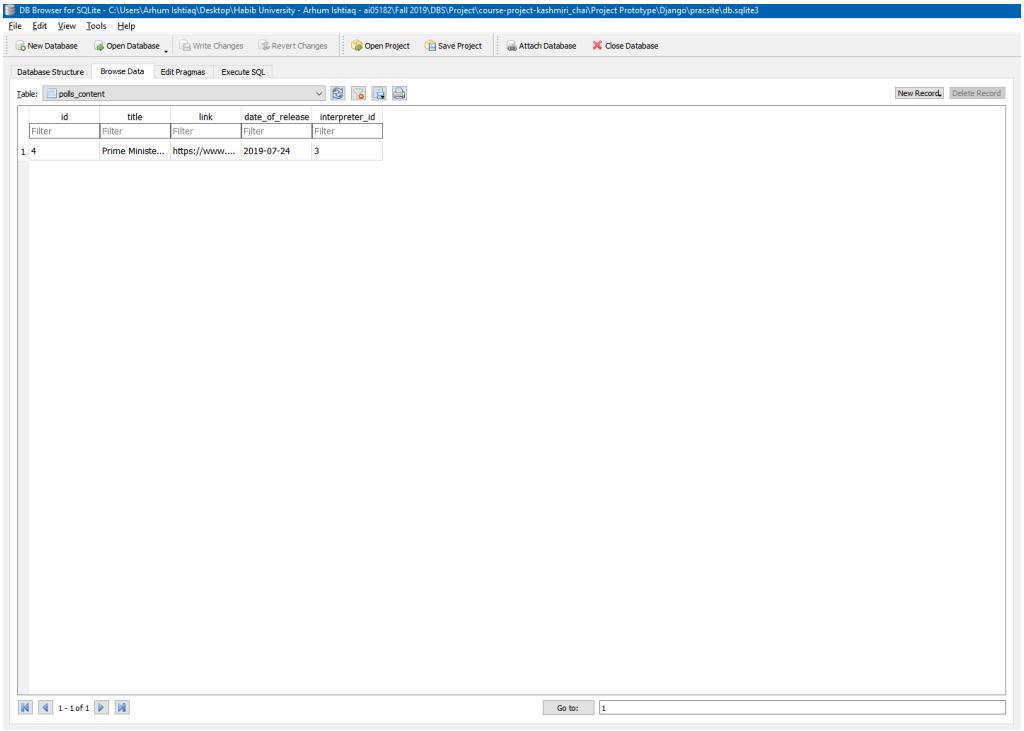
SELECT *
FROM *@tableName*
WHERE name/title = *@inputField*

To provide our program with some agility and versatility, we chose to work with SQLite 3, often referred to as a small, fast, self-contained, high-reliability, full-featured, SQL database engine. To interface with the database, we used DB Browser, screenshot below:



The DB Browser not only allows us to view the tables we've created but also dynamically add content through executing raw SQL commands, if the need arises, as shown below:

The initial state of the *"content"* database:



The successful result after running an INSERT query to add to the database: