FINAL REPORT

ConnectHear DBMS

Arhum Ishtiaq (ai05182)

Owais Bin Asad (oa05007)

Introduction

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.

The idea is to develop a database system that helps ConnectHear manage their interpreters, clients (individuals and companies), students, content library, call records, and projects.

Each interpreter will have their complete name, gender, age, address, mobile number, and date of joining.

Each individual customer will have their complete name, address, gender, mobile number, and date of joining.

Each client (company) will have their complete name, address, point of contact's name, point of contact's mobile number, and date of joining.

Each content item in the library will have its title, the name of the interpreter who starred in it, the link to the content, and the date of release.

Each student will have their name, age, gender, mobile number, city, level, classification, occupation, and the trainer who has been assigned to them.

Each project will have the name of the interpreter assigned to it, the name of the client, date, starting time, ending time, location, payment, and payment status.

Each call record will have the name of the interpreter who hosted the call, the name of the customer in the call, starting time, ending time, the reason for the call, comments, customer's rating, and interpreter's rating.

Modules of the System

1. Interpreter look-up: A module that allows us to lookup an interpreter and their relevant details.
2. Client directory: A module that will allow us to look for any company and check for the project-based services offered to them.
3. Customer directory: A module that will allow us to search for any customer and check details for their interactions with the company.
4. Video playlist: A module that allows to look up any of the company's videos and provides links for them to view them.
5. Student records: A module allowing for a quick look at the status of a student and update student records.

Front-end Development

The target audience of this DBMS will be using it over the internet. This will require a web-based front-end. Going with a modern approach, we will be using a combination of HTML, CSS, and JavaScript along with Django to render all these. The main objectives of the front-end will be to:

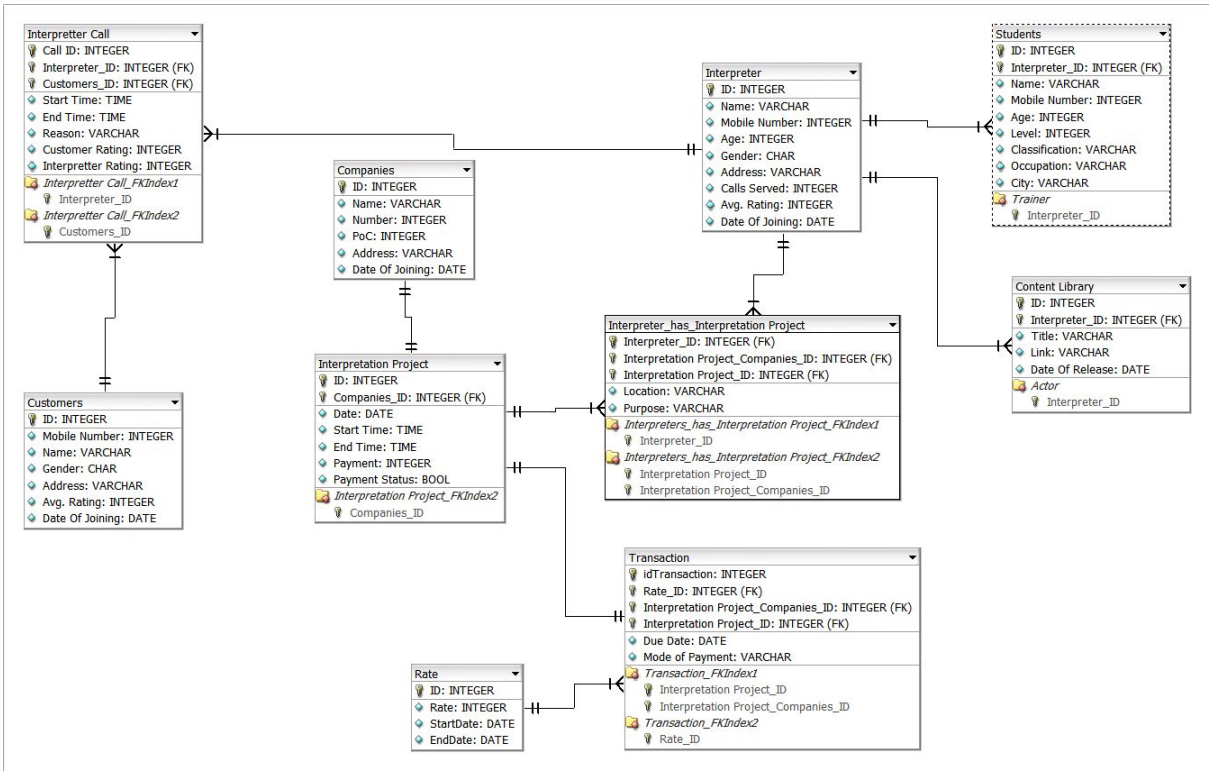
1. Provide the end-user with a clean, minimalistic, and intuitive representation of the form.
2. Provide appealing visuals to the end-user in order to make it easier for them to use the application.

Tools & Technologies

Back-end: SQL Server, Python, DB Browser, Django

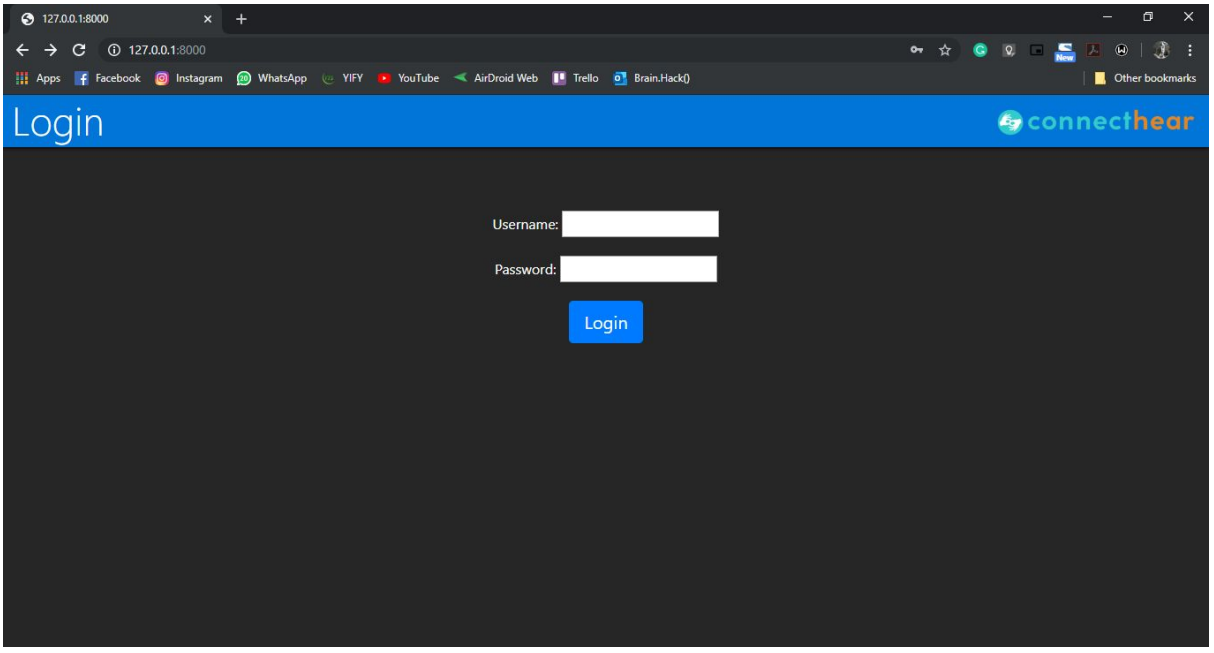
Front-end: Python, Javascript, CSS HTML, Bootstrap 4

ERD:

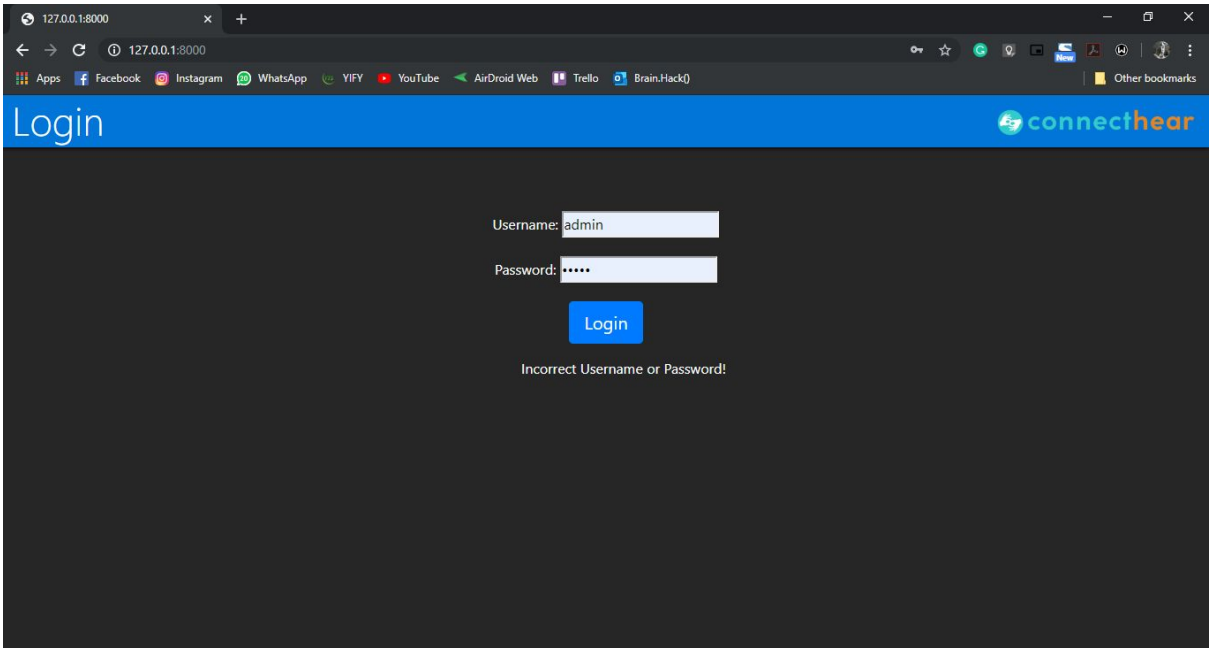

Interpretter Call
Call ID: INTEGER
Interpreter_ID: INTEGER (FK)
Customers_ID: INTEGER (FK)
Start Time: TIME
End Time: TIME
Reason: VARCHAR
Customer Rating: INTEGER
Interpretter Rating: INTEGER
Interpretter Call_FKIndex1
Interpreter_ID
Interpretter Call_FKIndex2
Customers_ID
Interpreter
ID: INTEGER
Name: VARCHAR
Mobile Number: INTEGER
Age: INTEGER
Gender: CHAR
Address: VARCHAR
Calls Served: INTEGER
Avg. Rating: INTEGER
Date Of Joining: DATE
Students
ID: INTEGER
Interpreter_ID: INTEGER (FK)
Name: VARCHAR
Mobile Number: INTEGER
Age: INTEGER
Level: INTEGER
Classification: VARCHAR
Occupation: VARCHAR
City: VARCHAR
Trainer
Interpreter_ID
Companies
ID: INTEGER
Name: VARCHAR
Number: INTEGER
PoC: INTEGER
Address: VARCHAR
Date Of Joining: DATE
Content Library
ID: INTEGER
Interpreter_ID: INTEGER (FK)
Title: VARCHAR
Link: VARCHAR
Date Of Release: DATE
Actor
Interpreter_ID
Interpreter_has_Interpretation Project
Interpreter_ID: INTEGER (FK)
Interpretation Project_Companies_ID: INTEGER (FK)
Interpretation Project_ID: INTEGER (FK)
Location: VARCHAR
Purpose: VARCHAR
Interpreters_has_Interpretation Project_FKIndex1
Interpreter_ID
Interpreters_has_Interpretation Project_FKIndex2
Interpretation Project_ID
Interpretation Project_Companies_ID
Interpretation Project
ID: INTEGER
Companies_ID: INTEGER (FK)
Date: DATE
Start Time: TIME
End Time: TIME
Payment: INTEGER
Payment Status: BOOL
Interpretation Project_FKIndex2
Companies_ID
Customers
ID: INTEGER
Mobile Number: INTEGER
Name: VARCHAR
Gender: CHAR
Address: VARCHAR
Avg. Rating: INTEGER
Date Of Joining: DATE
Transaction
idTransaction: INTEGER
Rate_ID: INTEGER (FK)
Interpretation Project_Companies_ID: INTEGER (FK)
Interpretation Project_ID: INTEGER (FK)
Due Date: DATE
Mode of Payment: VARCHAR
Transaction_FKIndex1
Interpretation Project_ID
Interpretation Project_Companies_ID
Transaction_FKIndex2
Rate_ID
Rate
ID: INTEGER
Rate: INTEGER
StartDate: DATE
EndDate: DATE

User Interface:

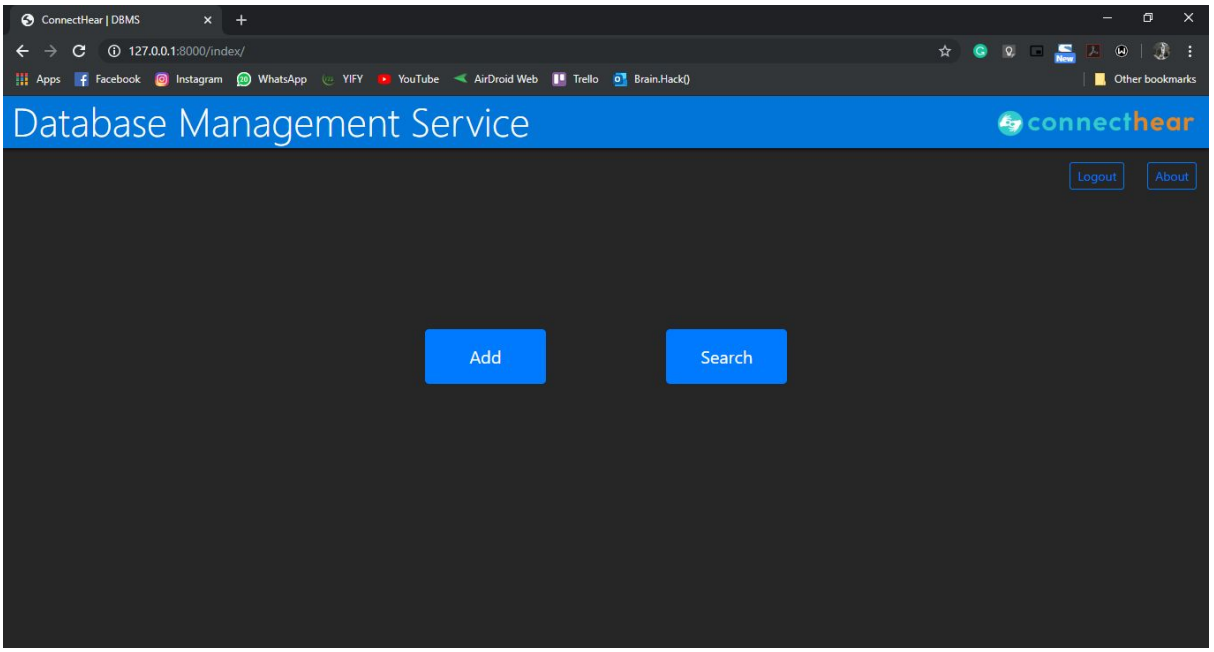
1. The user is presented with the login page as the landing page. The credentials are “admin” for username and “admin” for password by default.



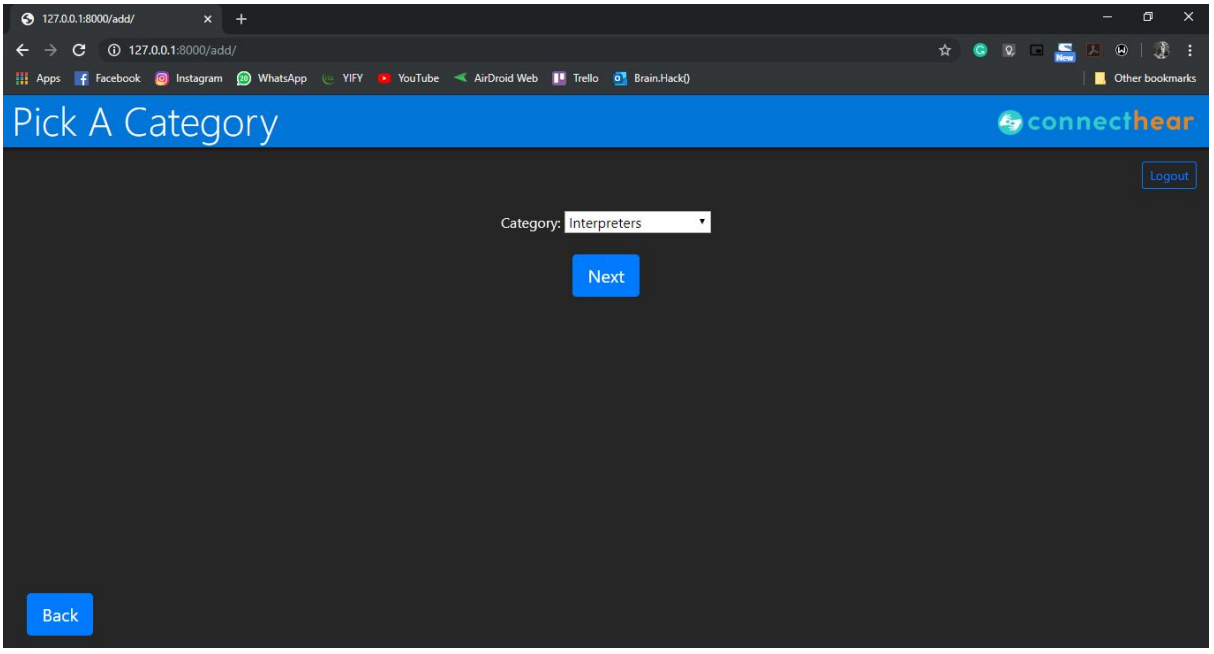
2. If incorrect credentials are added, the user is informed there and then.



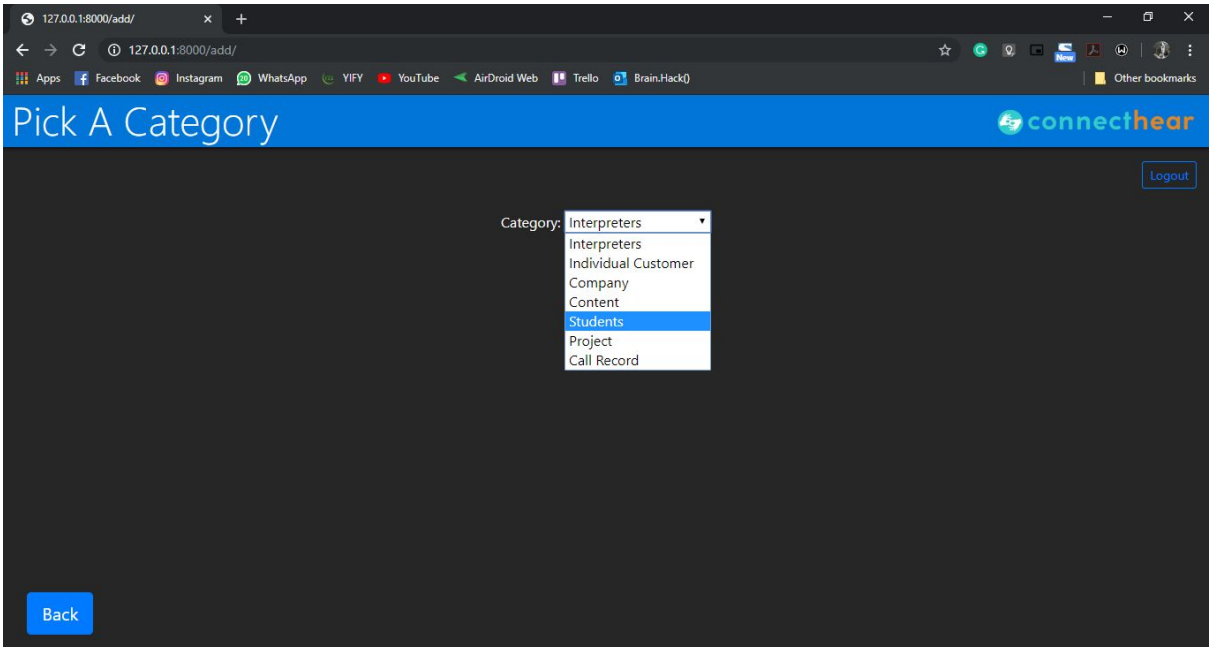
3. Upon entering valid credentials, the user is taken to the main page where they can opt between adding a new record or searching for one.



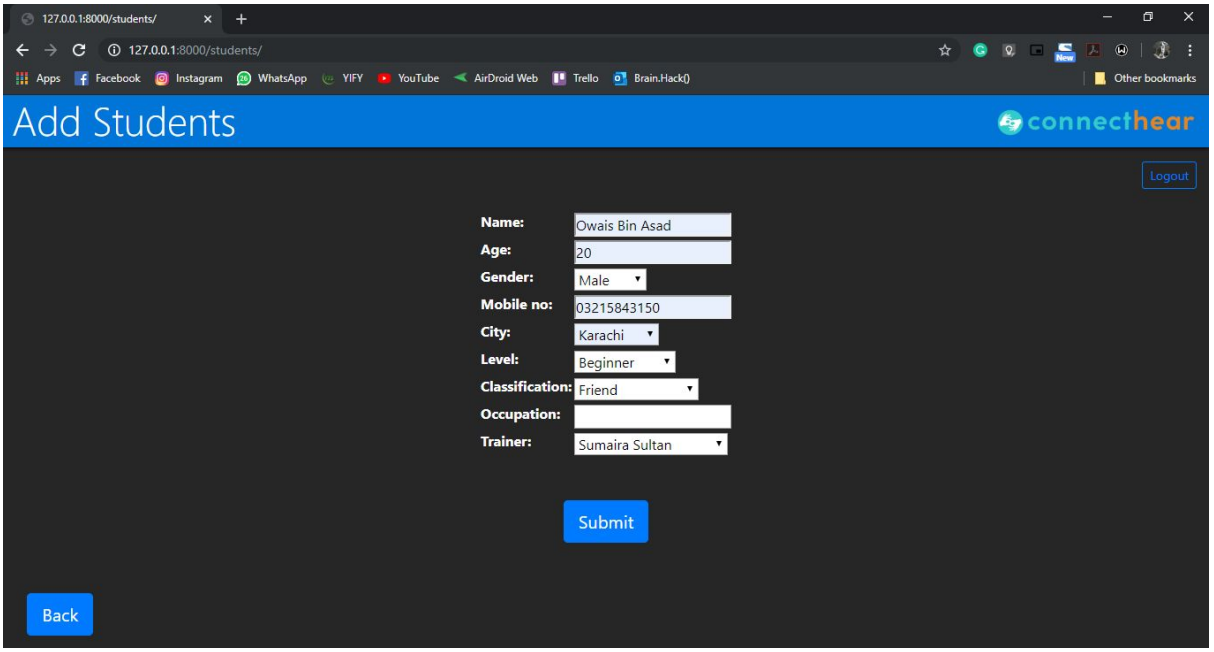
4. If they opt for adding, they have to choose a category in which they'd like to add a record.



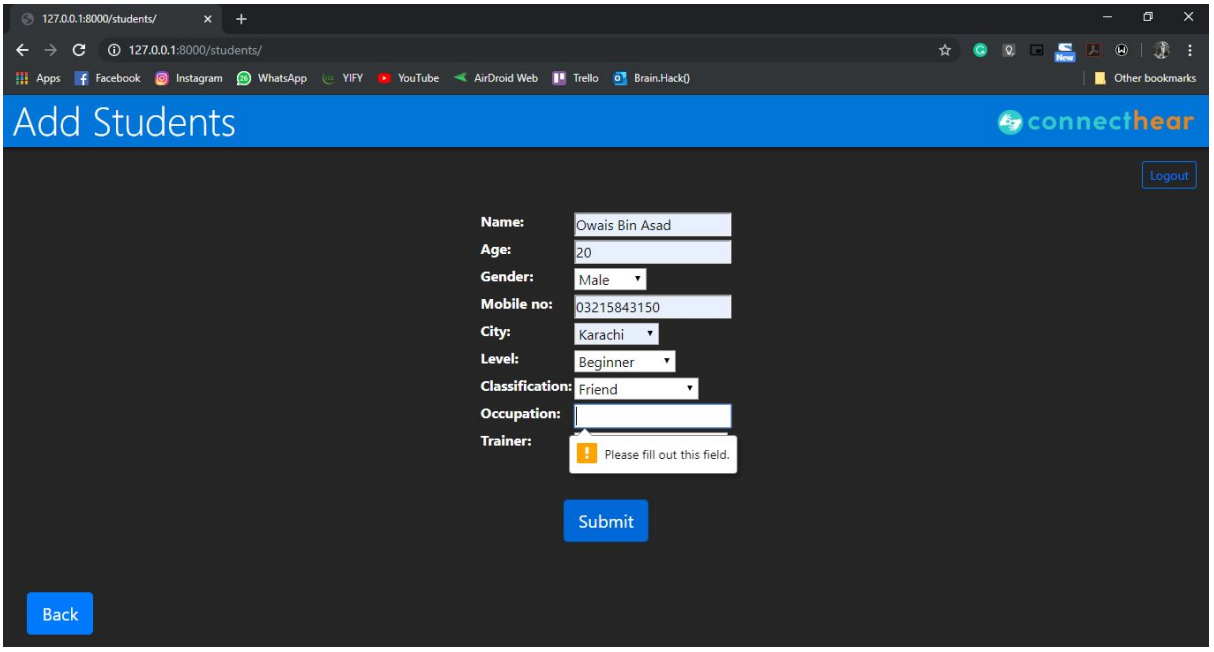
5. Each category corresponds to a separate table in the database.



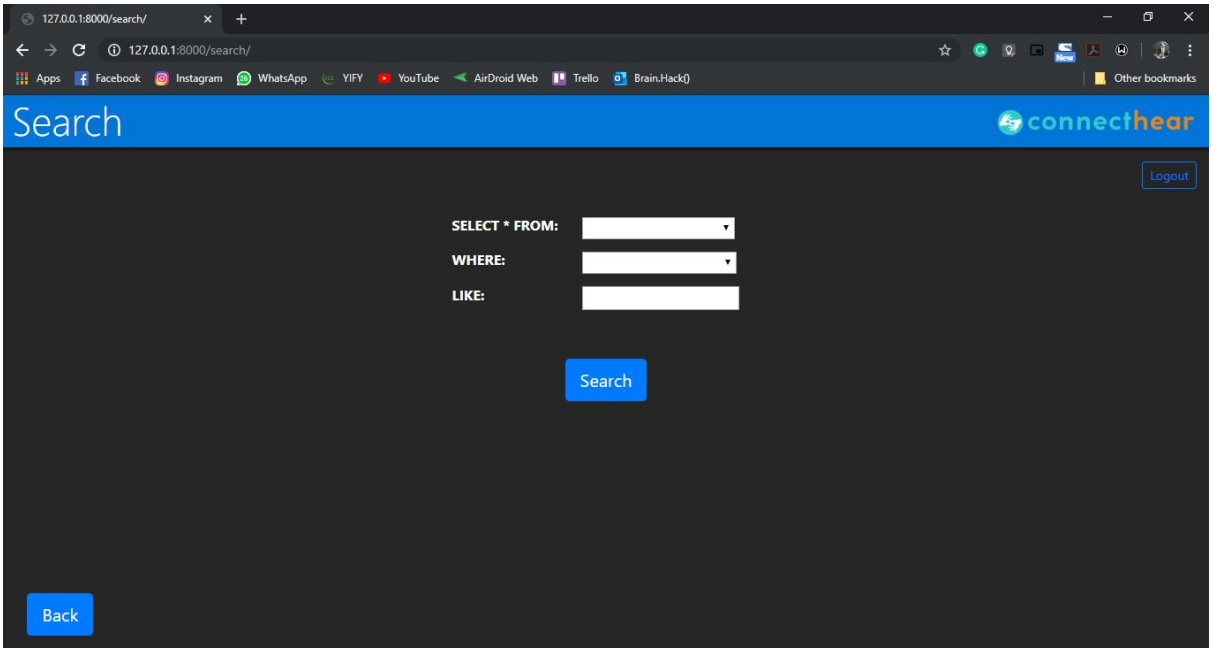
6. The user has to enter information in accordance with the data validation rules imposed for each form.



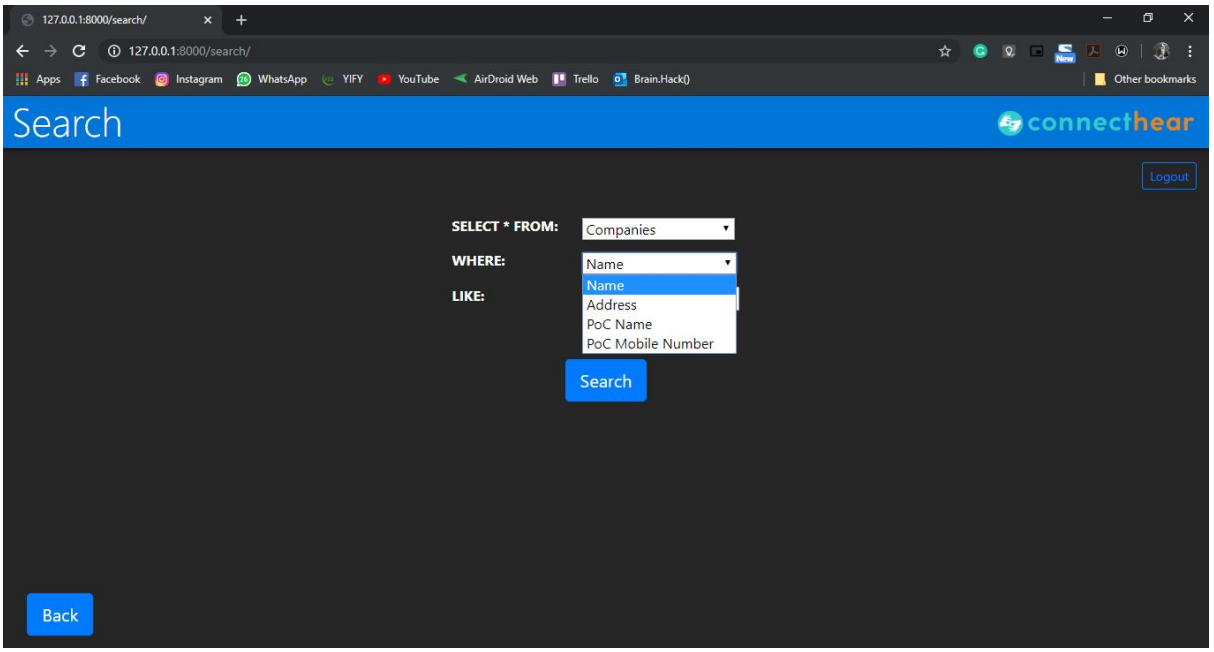
7. Failure to do so results in a warning message.



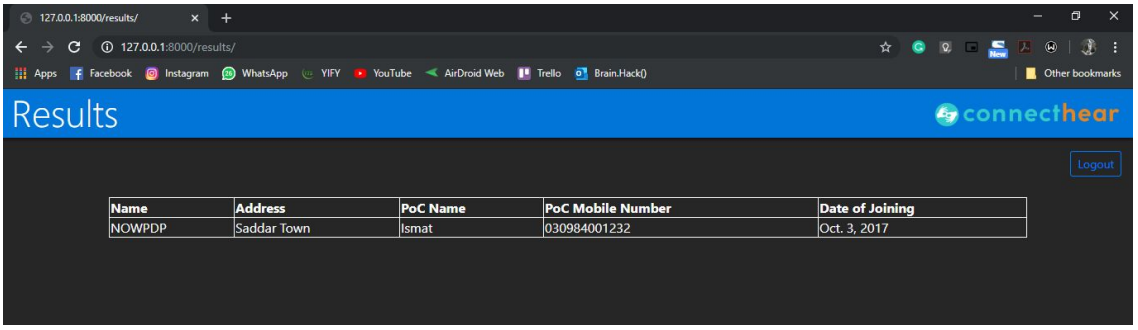
8. The user can also search for records. The search algorithm is quite powerful as it allows the user to search in any table by any field of that table.

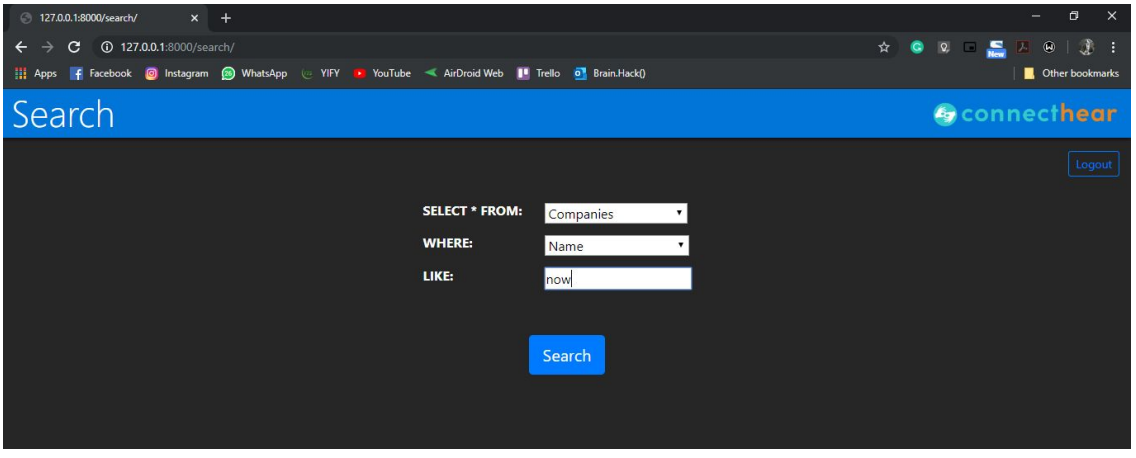


9. Upon choosing a category to search in, another drop-down menu is automatically populated with the fields of that table.

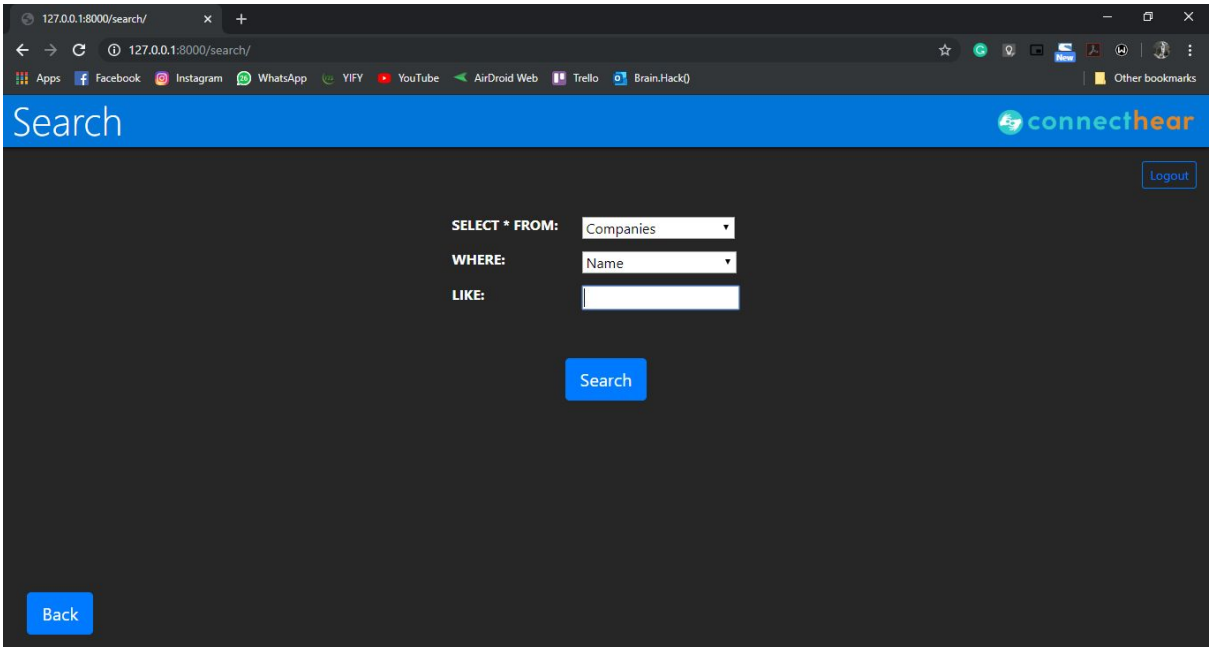


10. The user can enter the entire string they'd like to search for or just a part of it that they remember and the search would yield all results that have the passed string in them in the chosen field.

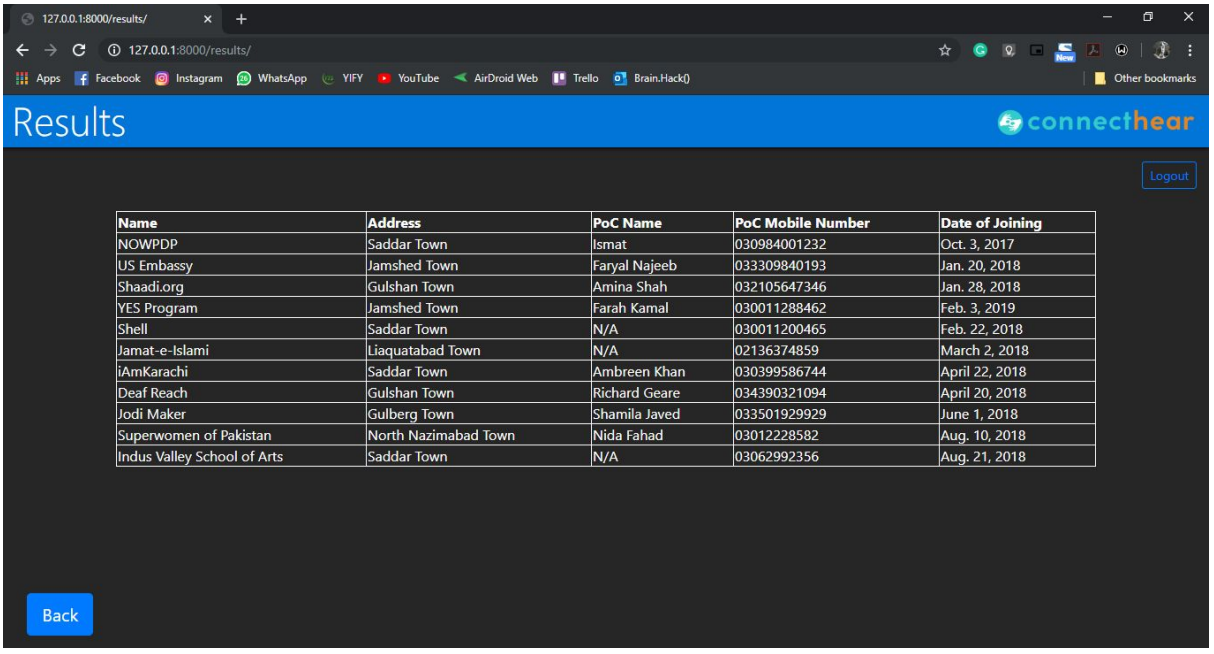




11. If the user wishes to do a blank search, they can by simply not entering anything into the **LIKE** field. This will yield all entries in the database without any constraints. In this case the chosen option in the **WHERE** field becomes irrelative.

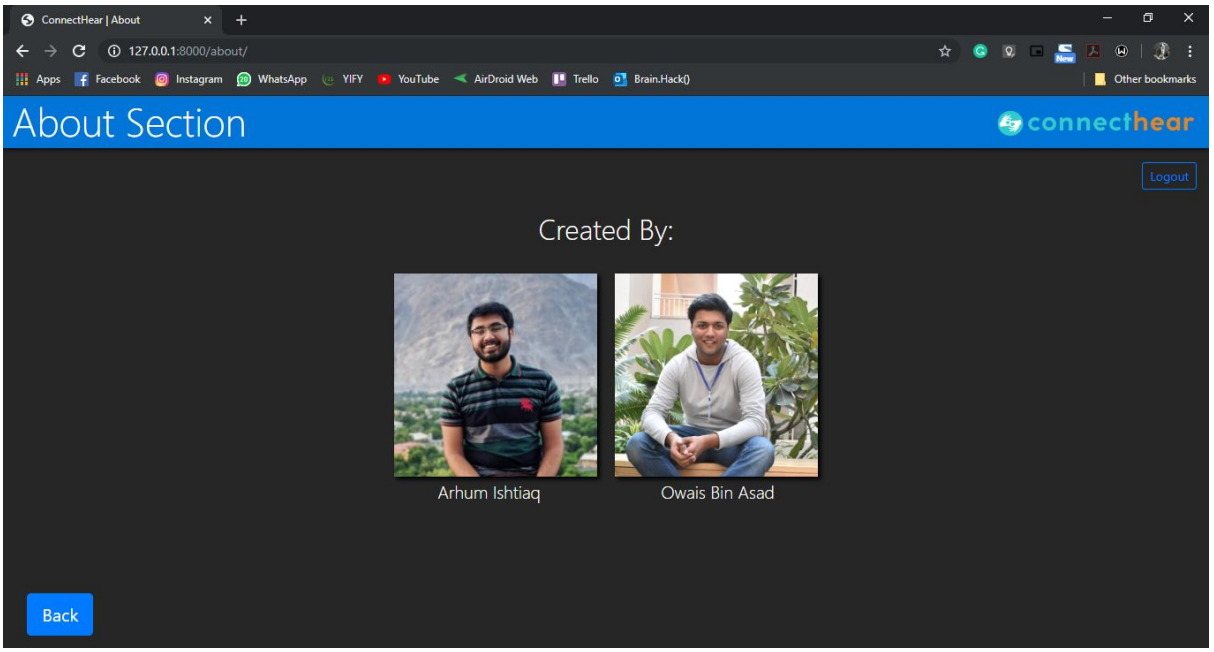


12. The results are displayed in a tabular form.

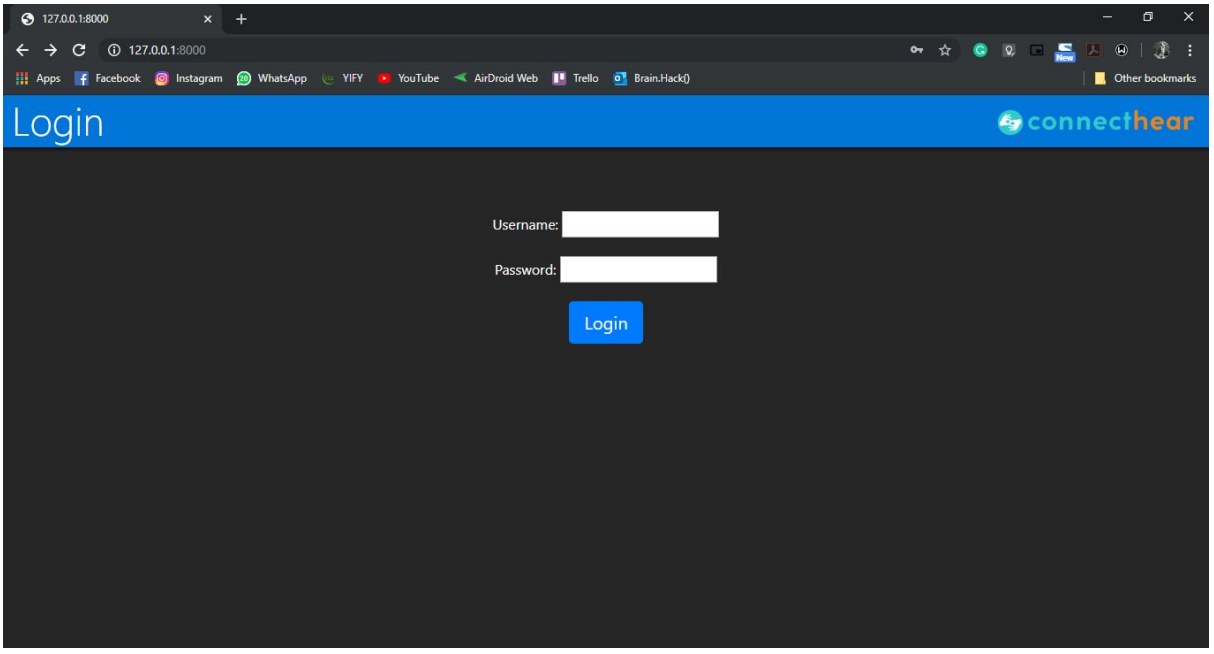


Name	Address	PoC Name	PoC Mobile Number	Date of Joining
NOWPDP	Saddar Town	Ismat	030984001232	Oct. 3, 2017
US Embassy	Jamshed Town	Faryal Najeeb	033309840193	Jan. 20, 2018
Shaadi.org	Gulshan Town	Amina Shah	032105647346	Jan. 28, 2018
YES Program	Jamshed Town	Farah Kamal	030011288462	Feb. 3, 2019
Shell	Saddar Town	N/A	030011200465	Feb. 22, 2018
Jamat-e-Islami	Liaquatabad Town	N/A	02136374859	March 2, 2018
iAmKarachi	Saddar Town	Ambreen Khan	030399586744	April 22, 2018
Deaf Reach	Gulshan Town	Richard Geare	034390321094	April 20, 2018
Jodi Maker	Gulberg Town	Shamila Javed	033501929929	June 1, 2018
Superwomen of Pakistan	North Nazimabad Town	Nida Fahad	0301228582	Aug. 10, 2018
Indus Valley School of Arts	Saddar Town	N/A	03062992356	Aug. 21, 2018

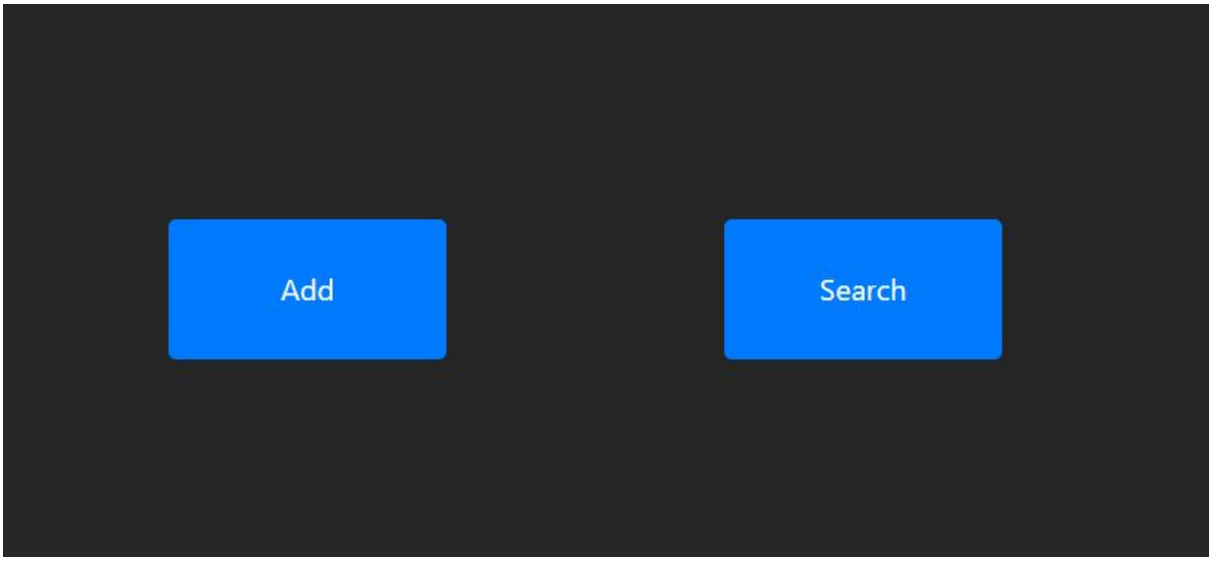
13. There is also an about section.



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



Queries:



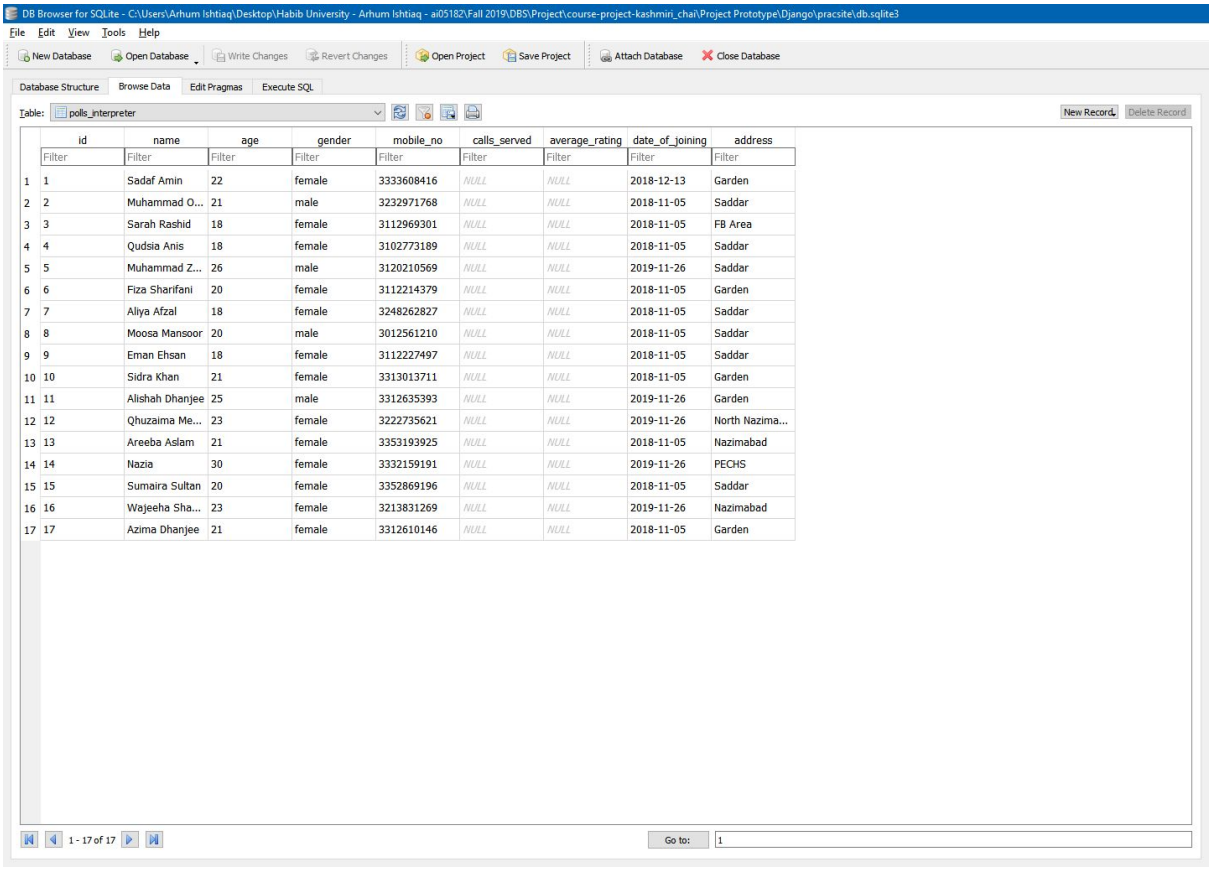
To begin with, each button leads to a form with different functionality. While "Add" leads to a form where we can get more relevant details about the personnel - leading to a schema request along the lines of:

INSERT INTO @tableName
VALUES (@inputFields)

Whereas, clicking on search would lead to a form that would be conducting the following query:

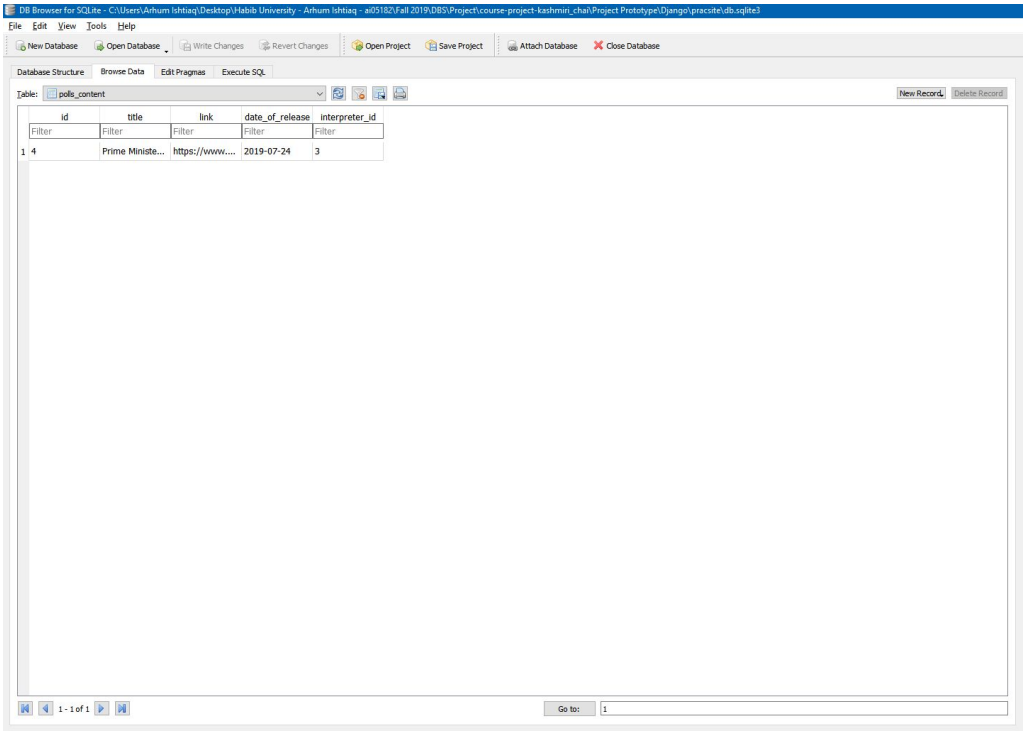
SELECT *
FROM @tableName
WHERE @tableField = @inputField

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:

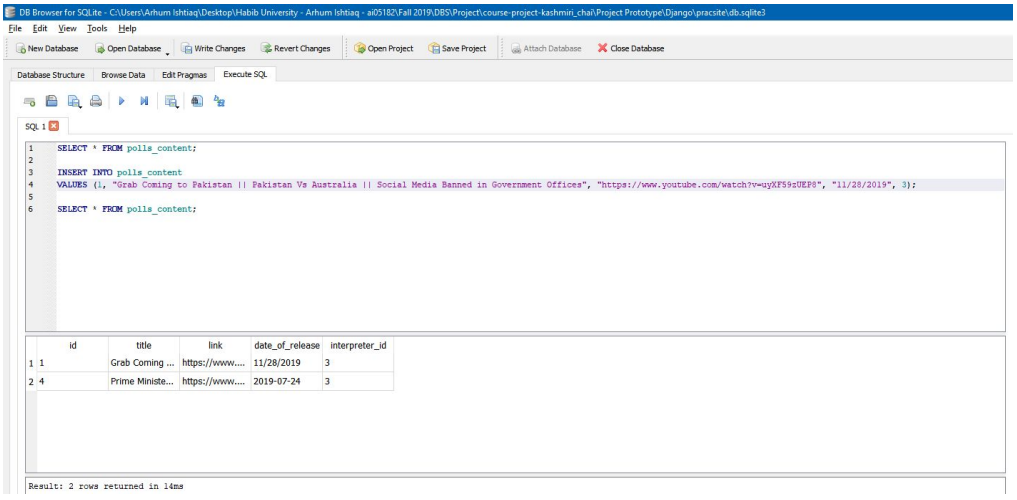


The DB Browser not only allows us to view the tables we've created but also dynamically add content through executing raw SQL commands, if the need arises, as shown below:

The initial state of the “content” database:



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



Test cases:

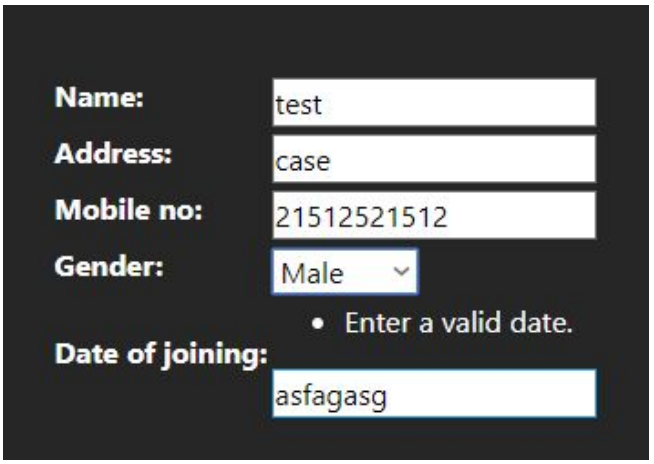
In order to ensure that we get the correct format, type, and amount of data into our forms that will be used to

1. Either search using the parameters provided by the users
2. Or save all requested parameters/inputs to the tables in our respective database

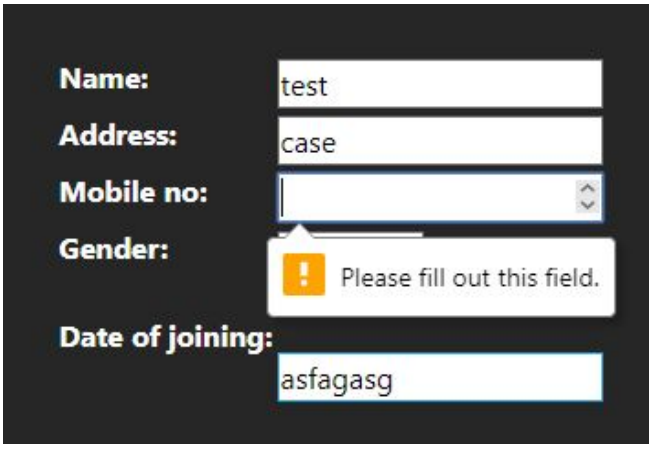
we have added custom form validation for each field depending on what input type we've set for the same field in the table. Following are the screenshots for certain scenarios where the user has either tried to put in the wrong type or no data at all so that we can initialize our form validation and prompt the user to enter the right type or form of data.

Page	**Table**	**Field**	**Input Value**	**Expected Output**	**Actual Output**
Add	All	Date of joining	asfagasg	Enter a valid date	Enter a valid date
Add	All	Mobile no	NULL	Please fill out this field.	Please fill out this field.
Add	Project	Interpreter	NULL	Please select an item in the list.	Please select an item in the list.
Search	All	Age	agasgas	Please only enter integers!	Please only enter integers!
Search	All	All	NULL	This field is required.	This field is required.
Search	All	Name	123131	Please only enter text values.	Please only enter text values.

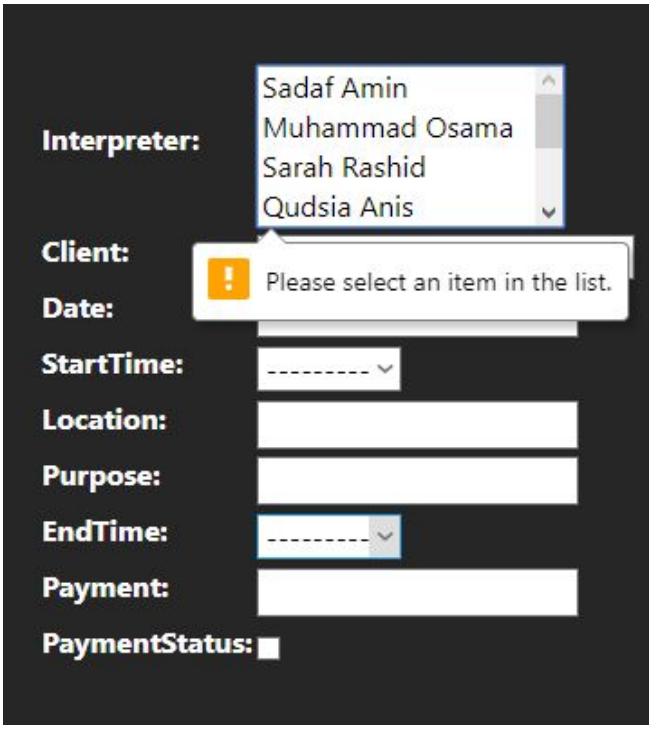
ADD form validation:



This screenshot shows that the date field will not accept any input that is not in a valid date format i.e. mm/dd/yyyy

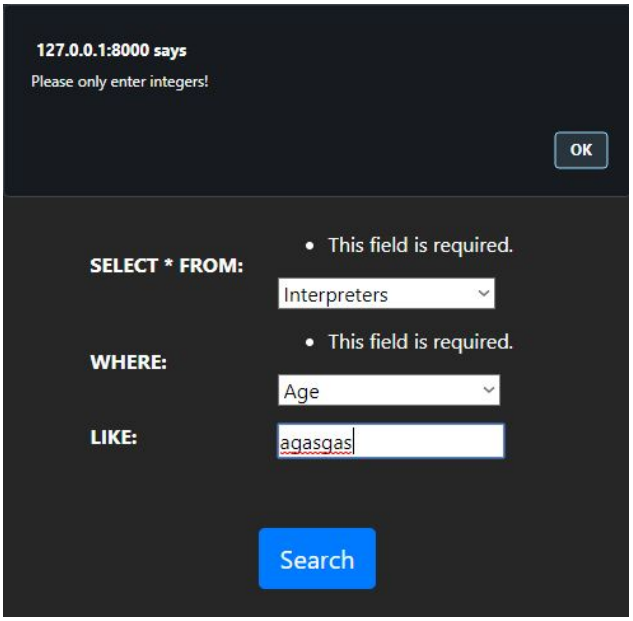


Here we're showing the fact that no form can be filled out with an empty field that is required for the table

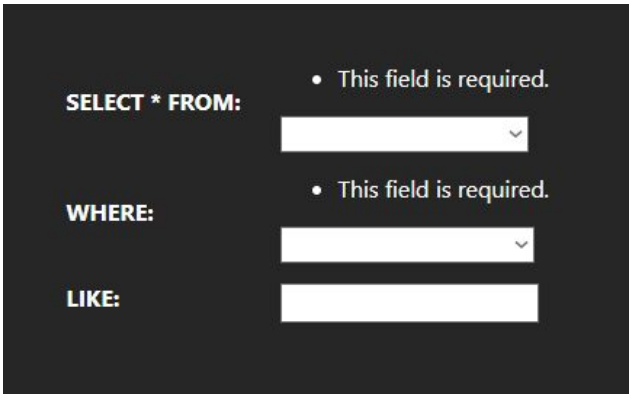


This screenshot depicts that for fields with drop-down or multiselection menus, the form will not submit until at least one selection is made to ensure that no NULL values are stored in the database

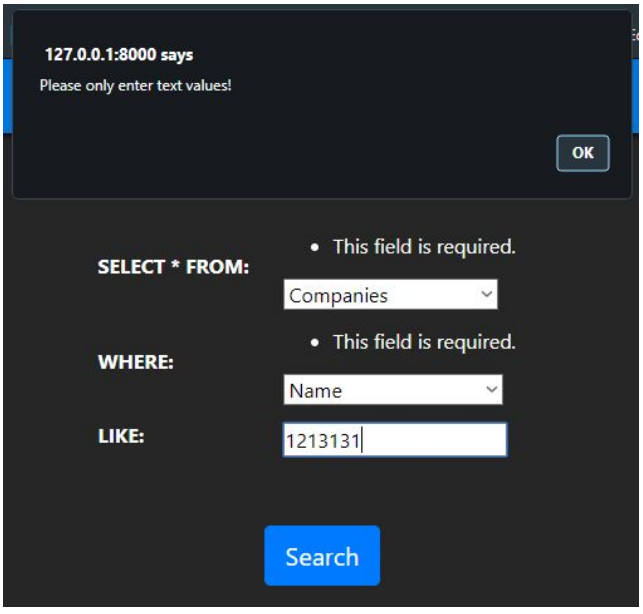
SEARCH form validation:



While searching for an entity by age, the user is not permitted to enter non-integers.



The user can not leave the fields blank. The only field that can be left blank is the LIKE field.



The user can not enter numerical values while searching for a particular entity by name.

The case against Stored Procedures and Views:

We did not use Stored Procedures as there was no single query that was complicated enough to warrant the use of a reusable procedure.

As for Views, all of our data is already normalized and arranged in a manner where we did not need to create a separate virtual table in order to get a certain collection of data.

Therefore, in light of the circumstances mentioned above, we implemented neither Stored Procedures nor Views.