



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:

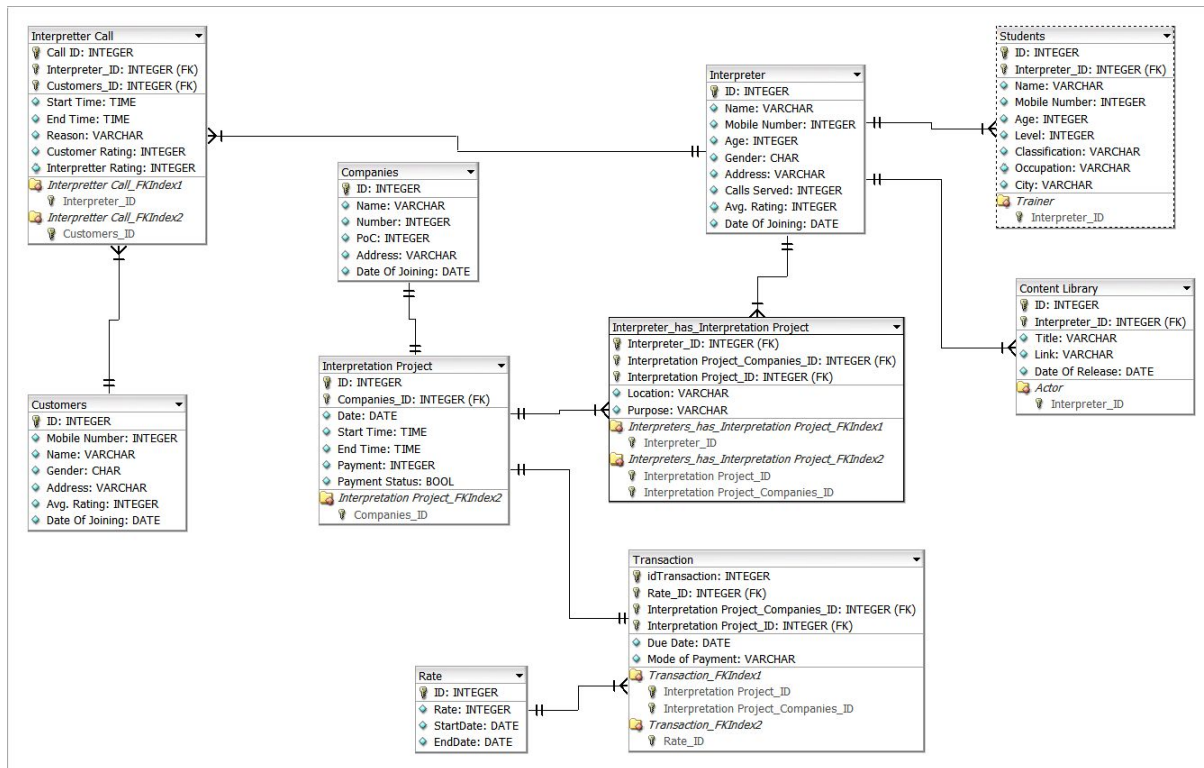
1. Provide the end-user with a clean, minimalistic, and intuitive representation of the form.
2. Provide appealing visuals to the end-user in order to make it easier for them to use the application.

Tools & Technologies

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

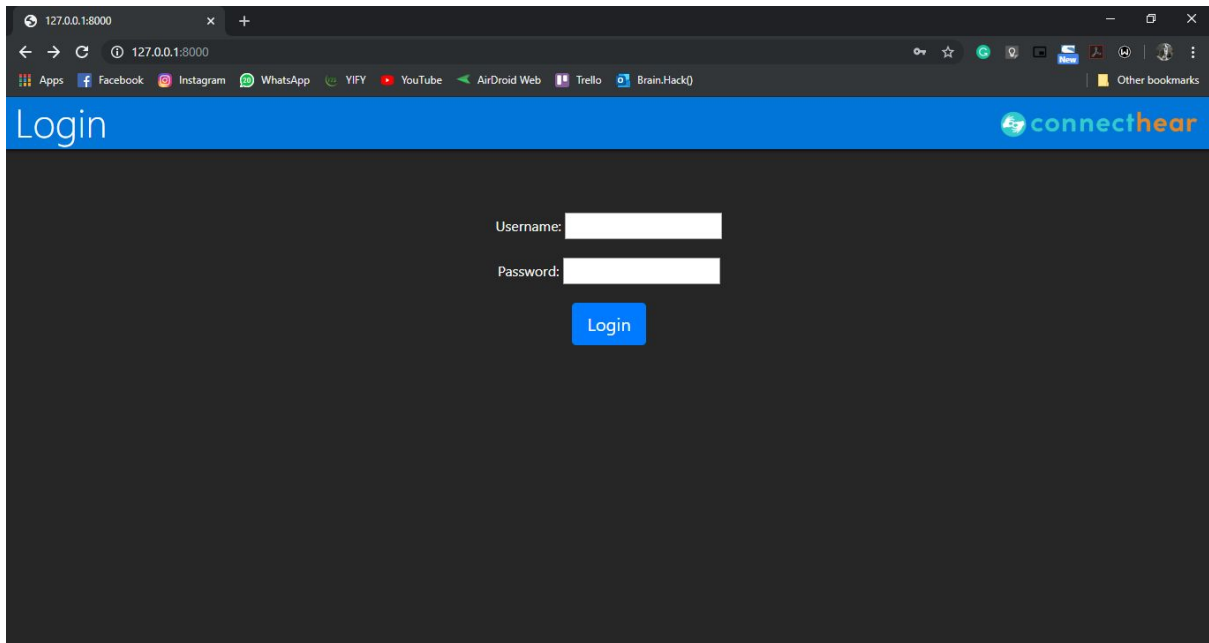
Front-end: Python, Javascript, CSS HTML, Bootstrap 4

ERD:

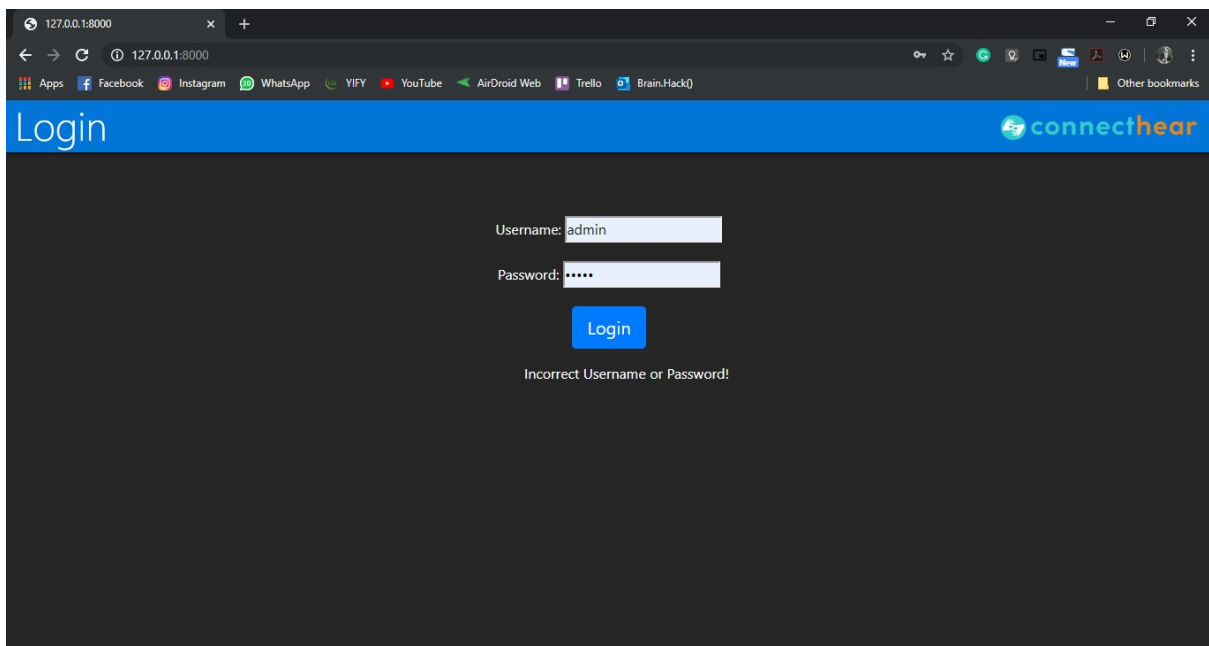


User Interface:

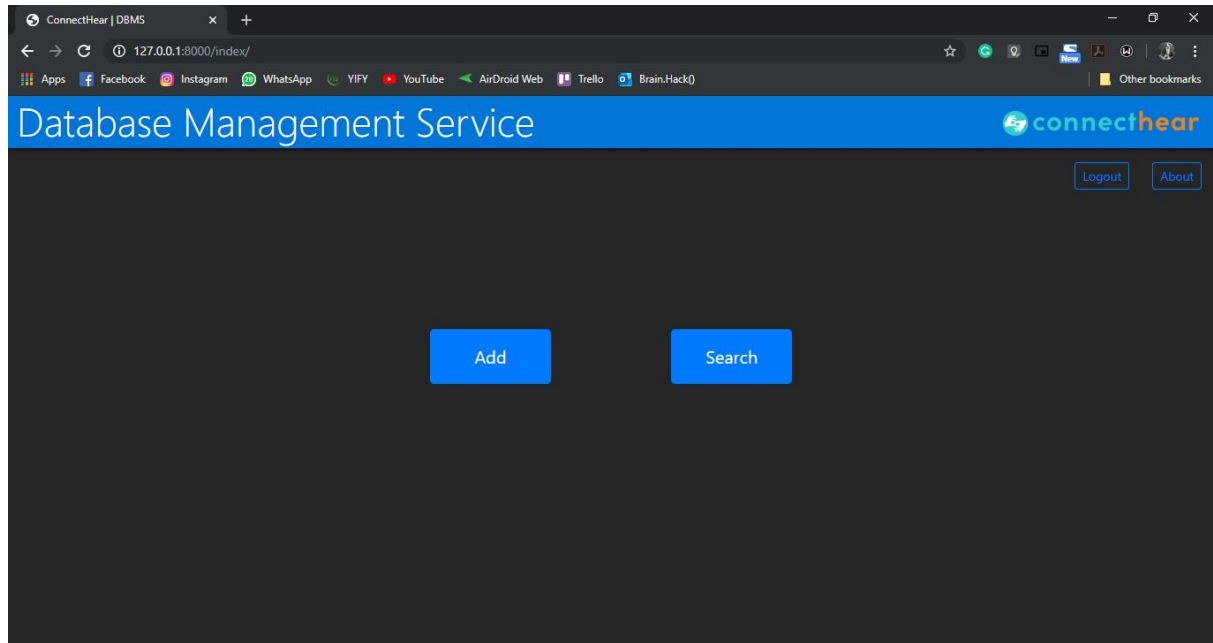
1. The user is presented with the login page as the landing page. The credentials are “admin” for username and “admin” for password by default.



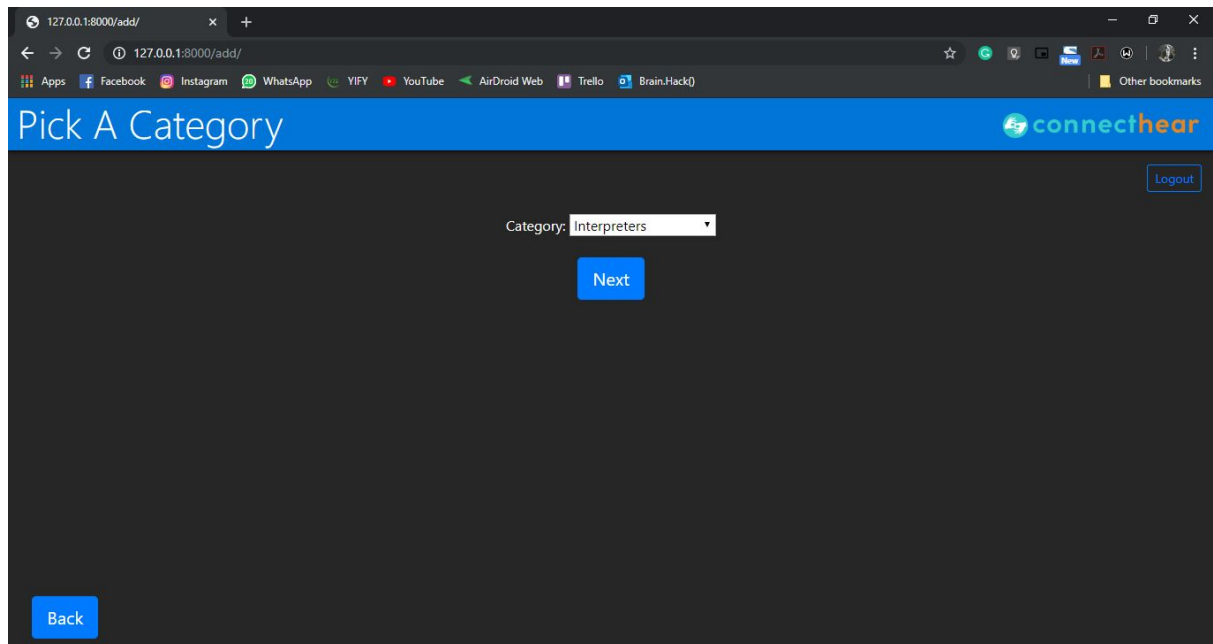
2. If incorrect credentials are added, the user is informed there and then.



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.

The screenshot shows a web browser at the address 127.0.0.1:8000/add/. The page has a blue header with the text "Pick A Category" and the "connecthear" logo. A "Logout" button is in the top right. The main content area is dark gray and contains a "Category:" label followed by a dropdown menu. The dropdown menu is open, showing a list of categories: Interpreters, Individual Customer, Company, Content, Students (highlighted in blue), Project, and Call Record. A "Back" button is located at the bottom left of the form area.

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

The screenshot shows a web browser at the address 127.0.0.1:8000/students/. The page has a blue header with the text "Add Students" and the "connecthear" logo. A "Logout" button is in the top right. The main content area is dark gray and contains a form with the following fields: "Name:" with the value "Owais Bin Asad", "Age:" with the value "20", "Gender:" with a dropdown menu showing "Male", "Mobile no:" with the value "03215843150", "City:" with a dropdown menu showing "Karachi", "Level:" with a dropdown menu showing "Beginner", "Classification:" with a dropdown menu showing "Friend", "Occupation:" with an empty text input field, and "Trainer:" with a dropdown menu showing "Sumaira Sultan". A "Submit" button is located below the form fields. A "Back" button is located at the bottom left of the form area.

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

The screenshot shows a web browser window with the URL `127.0.0.1:8000/students/`. The page has a blue header with the text "Add Students" and the "connecthear" logo. A "Logout" button is in the top right corner. The main content area is dark gray and contains a form with the following fields:

- Name:
- Age:
- Gender:
- Mobile no:
- City:
- Level:
- Classification:
- Occupation:
- Trainer:

A warning message is displayed below the Trainer field: "Please fill out this field." A blue "Submit" button is located below the form, and a blue "Back" button is in the bottom left corner.

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.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/search/`. The page has a blue header with the text "Search" and the "connecthear" logo. A "Logout" button is in the top right corner. The main content area is dark gray and contains three search criteria:

- SELECT * FROM:
- WHERE:
- LIKE:

A blue "Search" button is located below the search criteria, and a blue "Back" button is in the bottom left corner.

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

Search

connecthear

Logout

SELECT * FROM: Companies

WHERE: Name

LIKE: Name
Address
PoC Name
PoC Mobile Number

Search

Back

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.

Results

connecthear

Logout

Name	Address	PoC Name	PoC Mobile Number	Date of Joining
NOWPDP	Saddar Town	Ismat	030984001232	Oct. 3, 2017

Search

connecthear

Logout

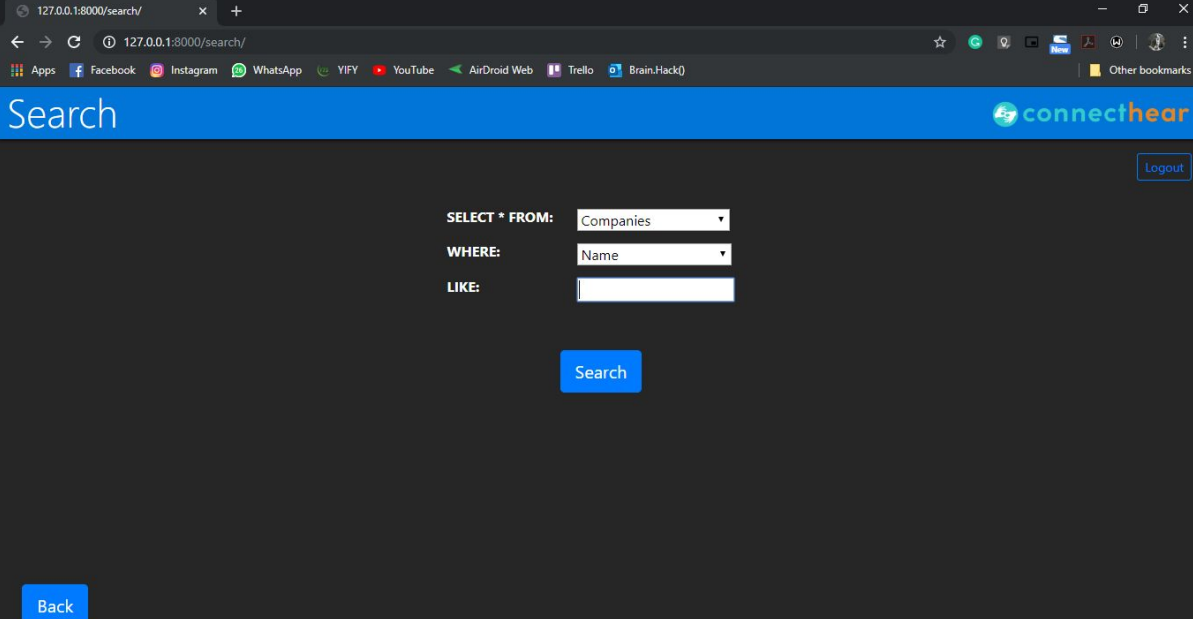
SELECT * FROM: Companies

WHERE: Name

LIKE: now

Search

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.



127.0.0.1:8000/search/

Search

connecthear

Logout

SELECT * FROM: Companies

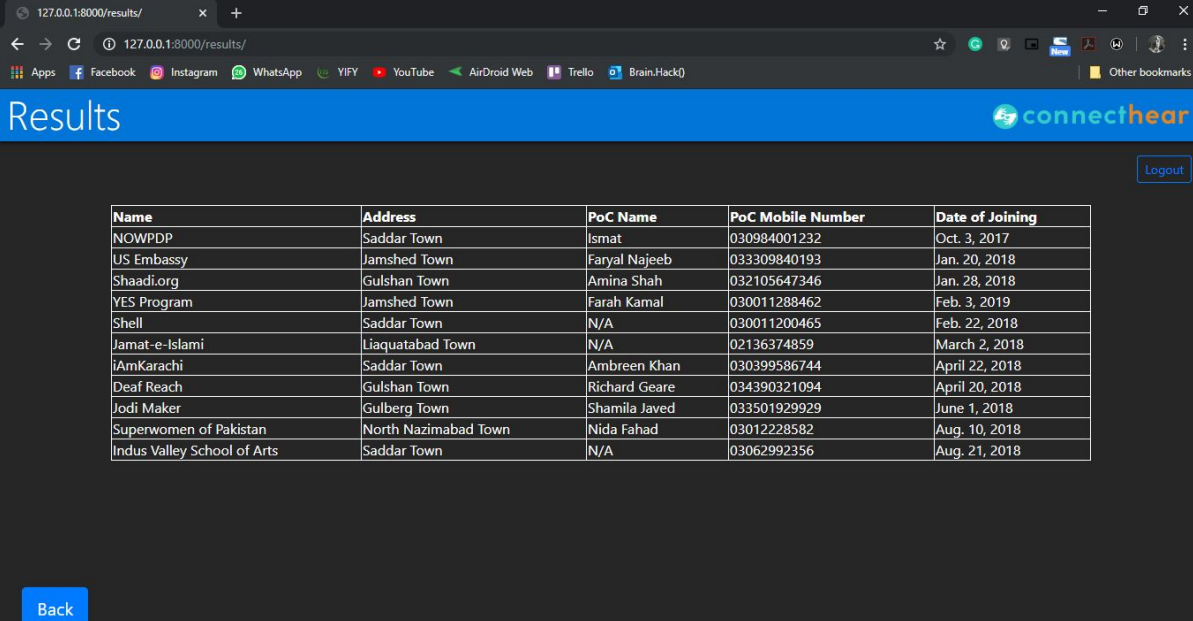
WHERE: Name

LIKE:

Search

Back

12. The results are displayed in a tabular form.



127.0.0.1:8000/results/

Results

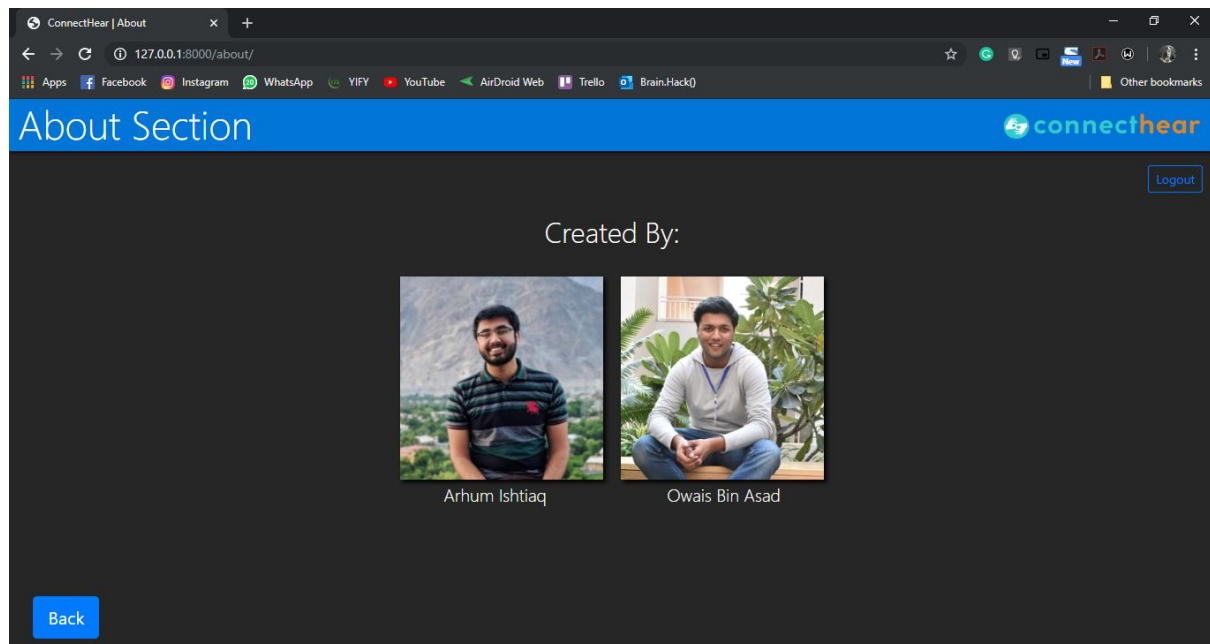
connecthear

Logout

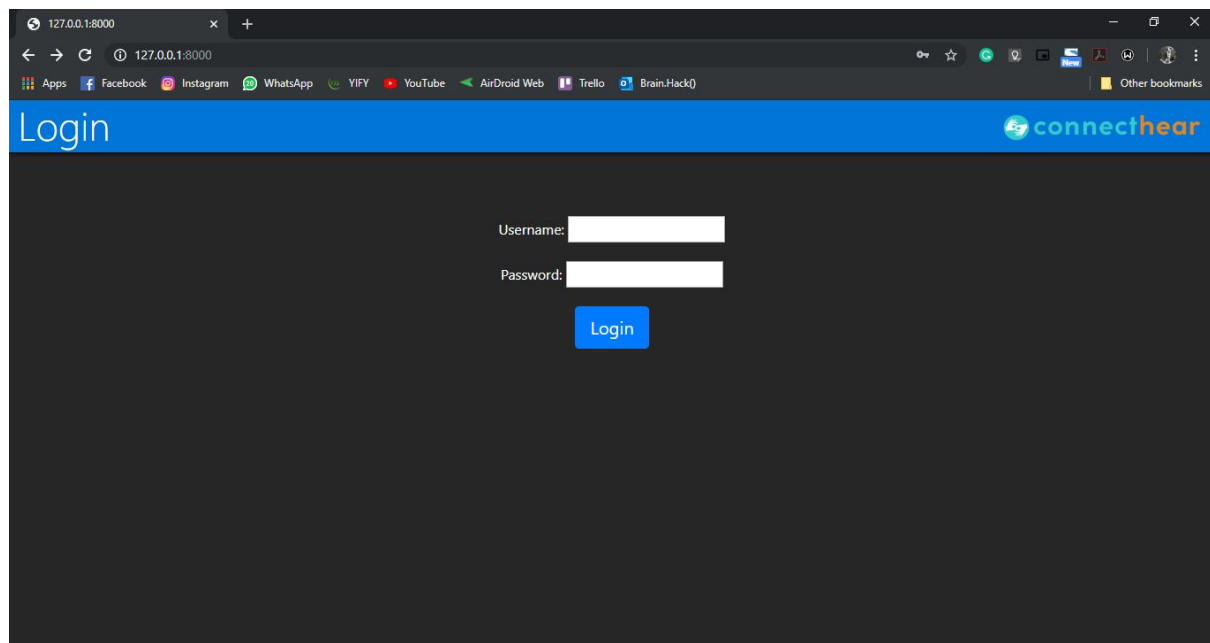
Name	Address	PoC Name	PoC Mobile Number	Date of Joining
NOWPDP	Saddar Town	Ismat	030984001232	Oct. 3, 2017
US Embassy	Jamshed Town	Faryal Najeeb	033309840193	Jan. 20, 2018
Shaadi.org	Gulshan Town	Amina Shah	032105647346	Jan. 28, 2018
YES Program	Jamshed Town	Farah Kamal	030011288462	Feb. 3, 2019
Shell	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	03012228582	Aug. 10, 2018
Indus Valley School of Arts	Saddar Town	N/A	03062992356	Aug. 21, 2018

Back

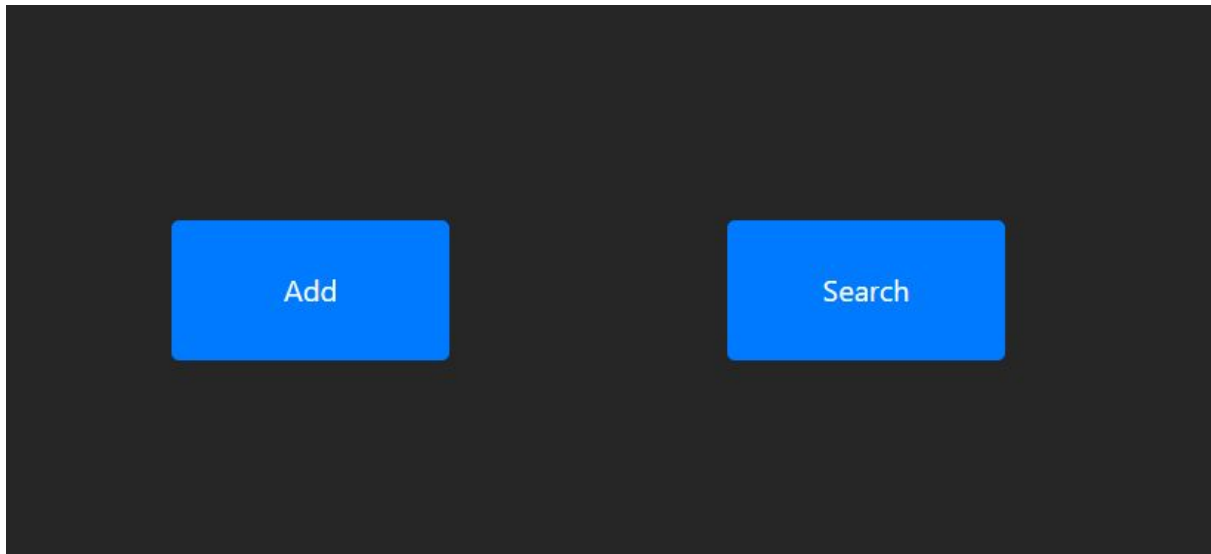
13. There is also an about section.



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:

DB Browser for SQLite - C:\Users\Arhum\Documents\Habib University - Arhum Ishtiaq - a05182\Fall 2019\DBS\Project\course-project-kashmiri_chai\Project Prototype\django\pracsite\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: polls_interpreter

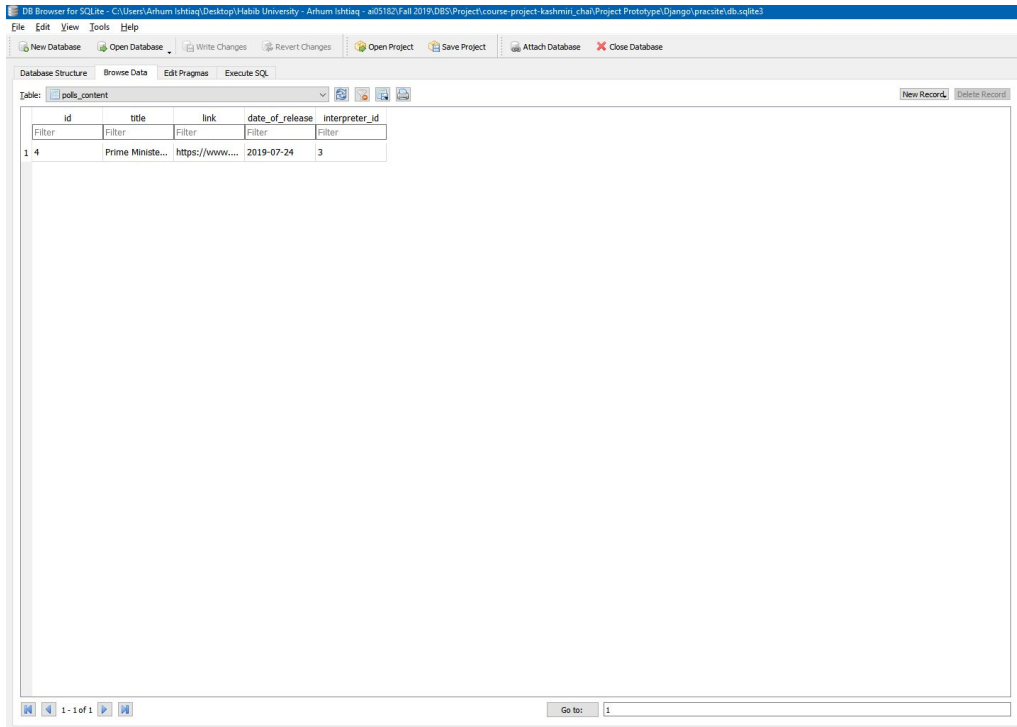
	id	name	age	gender	mobile_no	calls_served	average_rating	date_of_joining	address
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Sadaf Amin	22	female	3333608416	NULL	NULL	2018-12-13	Garden
2	2	Muhammad O...	21	male	3232971768	NULL	NULL	2018-11-05	Saddar
3	3	Sarah Rashid	18	female	3112969301	NULL	NULL	2018-11-05	FB Area
4	4	Qudsia Anis	18	female	3102773189	NULL	NULL	2018-11-05	Saddar
5	5	Muhammad Z...	26	male	3120210569	NULL	NULL	2019-11-26	Saddar
6	6	Fiza Sharifani	20	female	3112214379	NULL	NULL	2018-11-05	Garden
7	7	Aliya Afzal	18	female	3248262827	NULL	NULL	2018-11-05	Saddar
8	8	Moosa Mansoor	20	male	3012561210	NULL	NULL	2018-11-05	Saddar
9	9	Eman Ehsan	18	female	3112227497	NULL	NULL	2018-11-05	Saddar
10	10	Sidra Khan	21	female	3313013711	NULL	NULL	2018-11-05	Garden
11	11	Alishah Dhanjee	25	male	3312635393	NULL	NULL	2019-11-26	Garden
12	12	Qhuzaima Me...	23	female	3222735621	NULL	NULL	2019-11-26	North Nazima...
13	13	Areeba Aslam	21	female	3353193925	NULL	NULL	2018-11-05	Nazimabad
14	14	Nazie	30	female	3332159191	NULL	NULL	2019-11-26	PECHS
15	15	Sumaira Sultan	20	female	3352869196	NULL	NULL	2018-11-05	Saddar
16	16	Wajeelha Sha...	23	female	3213831269	NULL	NULL	2019-11-26	Nazimabad
17	17	Azima Dhanjee	21	female	3312610146	NULL	NULL	2018-11-05	Garden

1 - 17 of 17

Go to: 1

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:

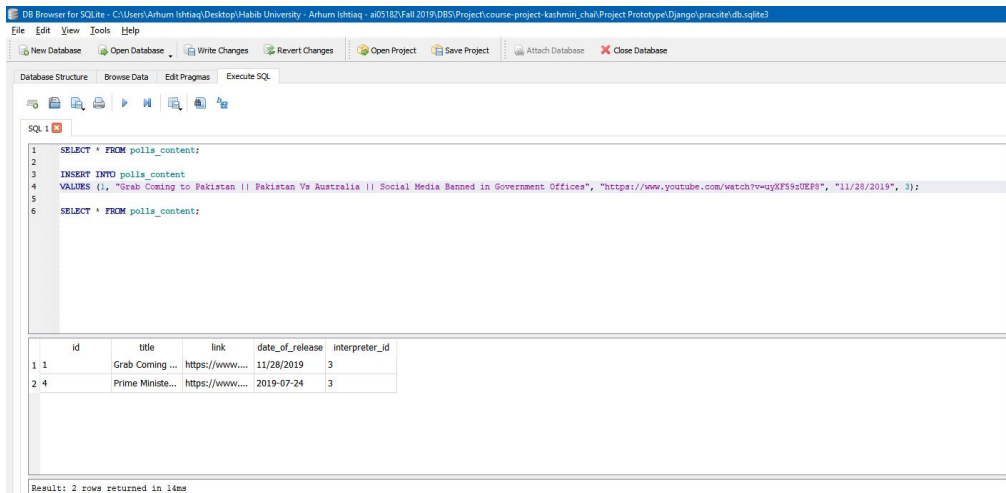
The initial state of the “content” database:



The screenshot shows the DB Browser for SQLite interface. The 'Database Structure' tab is active, displaying the 'polls_content' table. The table has five columns: id, title, link, date_of_release, and interpreter_id. The 'id' column has a filter set to '1'. The table contains one record with id 4, title 'Prime Ministe...', link 'https://www....', date_of_release '2019-07-24', and interpreter_id 3. The status bar at the bottom indicates '1 - 1 of 1' records.

id	title	link	date_of_release	interpreter_id
4	Prime Ministe...	https://www....	2019-07-24	3

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



The screenshot shows the DB Browser for SQLite interface after running an SQL query. The 'Execute SQL' tab is active, displaying the following query:

```
1 SELECT * FROM polls_content;
2
3 INSERT INTO polls_content
4 VALUES (1, "Grab Coming to Pakistan || Pakistan Vs Australia || Social Media Banned in Government Offices", "https://www.youtube.com/watch?v=yKF59dUEP8", "11/28/2019", 3);
5
6 SELECT * FROM polls_content;
```

 The result shows two records in the 'polls_content' table. The first record has id 1, title 'Grab Coming ...', link 'https://www....', date_of_release '11/28/2019', and interpreter_id 3. The second record is the same as the one in the previous screenshot. The status bar at the bottom indicates 'Result: 2 rows returned in 14ms'.

id	title	link	date_of_release	interpreter_id
1	Grab Coming ...	https://www....	11/28/2019	3
4	Prime Ministe...	https://www....	2019-07-24	3

Test cases:

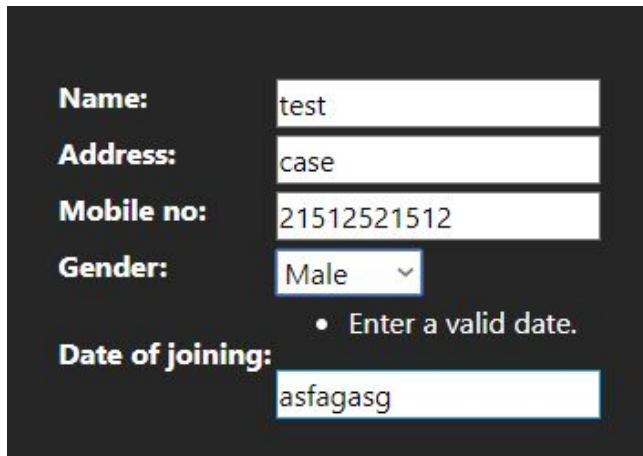
In order to ensure that we get the correct format, type, and amount of data into our forms that will be used to

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:



A screenshot of a form with a dark background. The fields are labeled 'Name:', 'Address:', 'Mobile no:', 'Gender:', and 'Date of joining:'. The inputs are 'test', 'case', '21512521512', 'Male' (selected in a dropdown), and 'asfagasg' respectively. A validation message 'Enter a valid date.' is shown next to the 'Date of joining:' field.

Name: test

Address: case

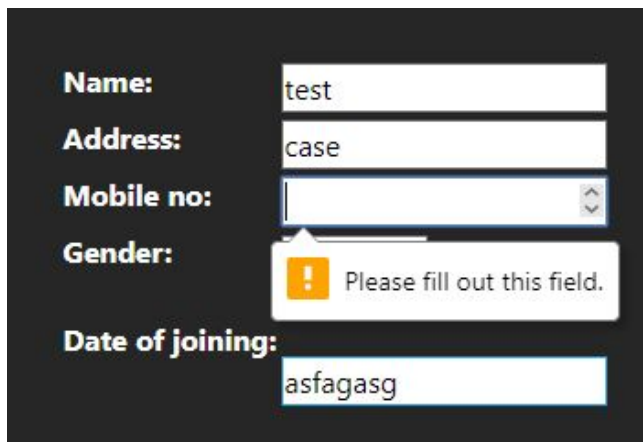
Mobile no: 21512521512

Gender: Male

Date of joining: asfagasg

- Enter a valid date.

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



A screenshot of the same form as above, but the 'Mobile no:' field is empty. A validation message 'Please fill out this field.' with an exclamation mark icon is shown next to the empty field.

Name: test

Address: case

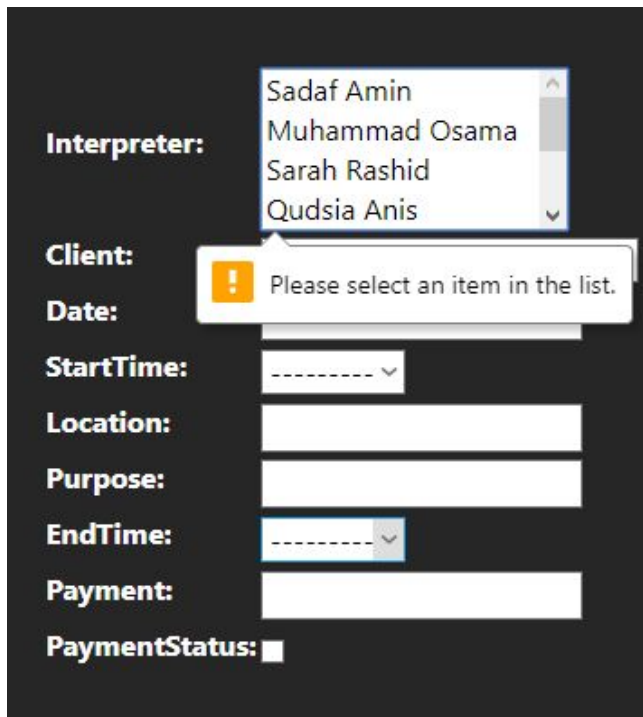
Mobile no:

Gender:

Date of joining: asfagasg

Please fill out this field.

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



A screenshot of a form with a dark background. The fields are labeled 'Interpreter:', 'Client:', 'Date:', 'StartTime:', 'Location:', 'Purpose:', 'EndTime:', 'Payment:', and 'PaymentStatus:'. The 'Interpreter:' dropdown menu is open, showing a list of names: Sadaf Amin, Muhammad Osama, Sarah Rashid, and Qudsia Anis. A validation message 'Please select an item in the list.' with an exclamation mark icon is shown next to the dropdown menu.

Interpreter:

Sadaf Amin
Muhammad Osama
Sarah Rashid
Qudsia Anis

Client:

Date:

StartTime:

Location:

Purpose:

EndTime:

Payment:

PaymentStatus:

Please select an item in the list.

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:

127.0.0.1:8000 says
Please only enter integers!

OK

SELECT * FROM: • This field is required.
Interpreters

WHERE: • This field is required.
Age

LIKE: agasgas

Search

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

SELECT * FROM: • This field is required.
▼

WHERE: • This field is required.
▼

LIKE:

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

127.0.0.1:8000 says
Please only enter text values!

OK

SELECT * FROM: • This field is required.
Companies

WHERE: • This field is required.
Name

LIKE: 1213131

Search

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.