

# **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.

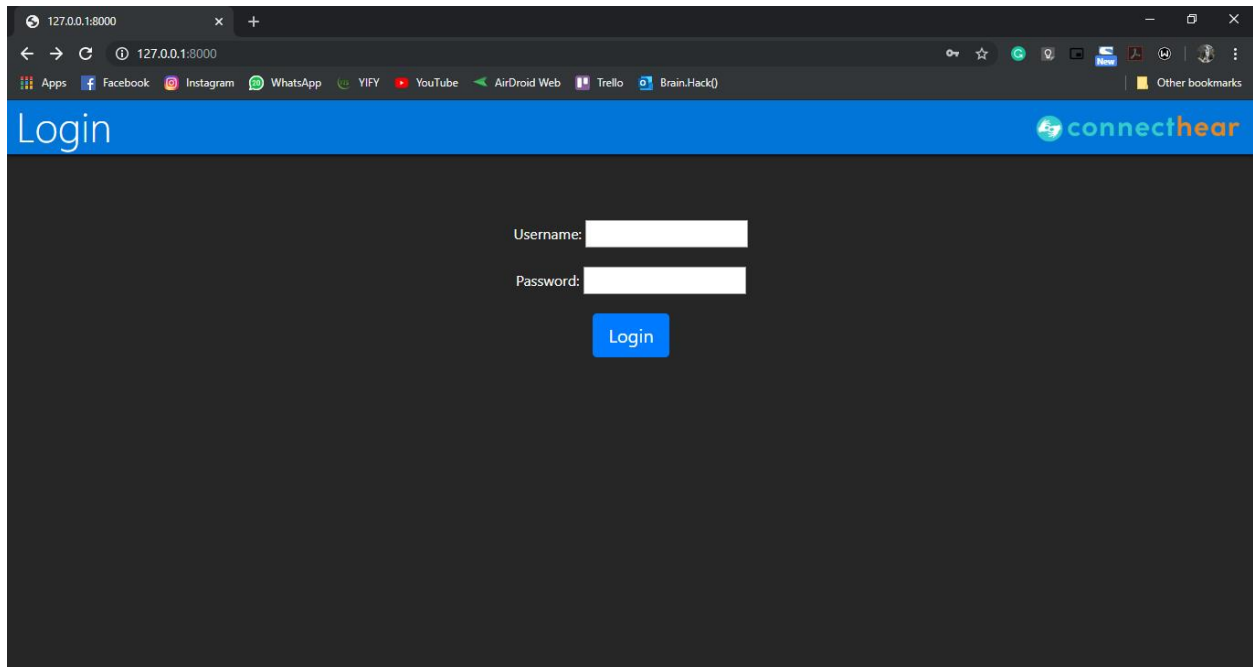
### **Technical Details:**

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.

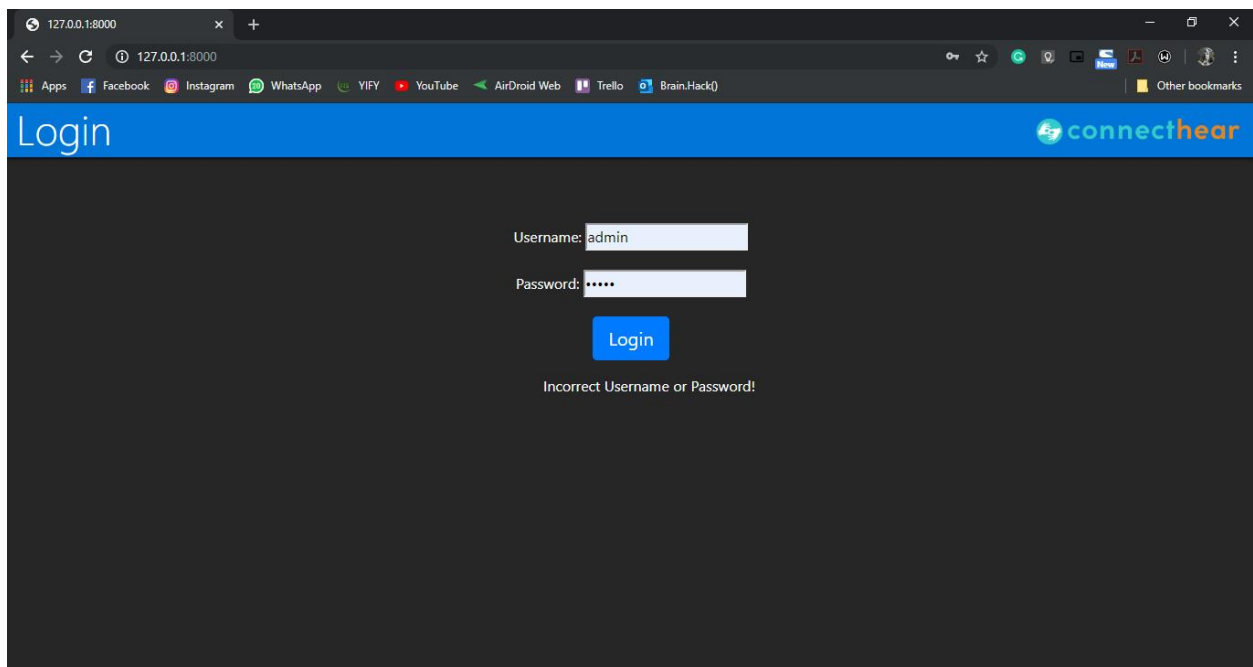
The project has been compiled in Python 3.8 and uses Django 2.2.7. The database used on the backend is SQLite 3.

## User Interface:

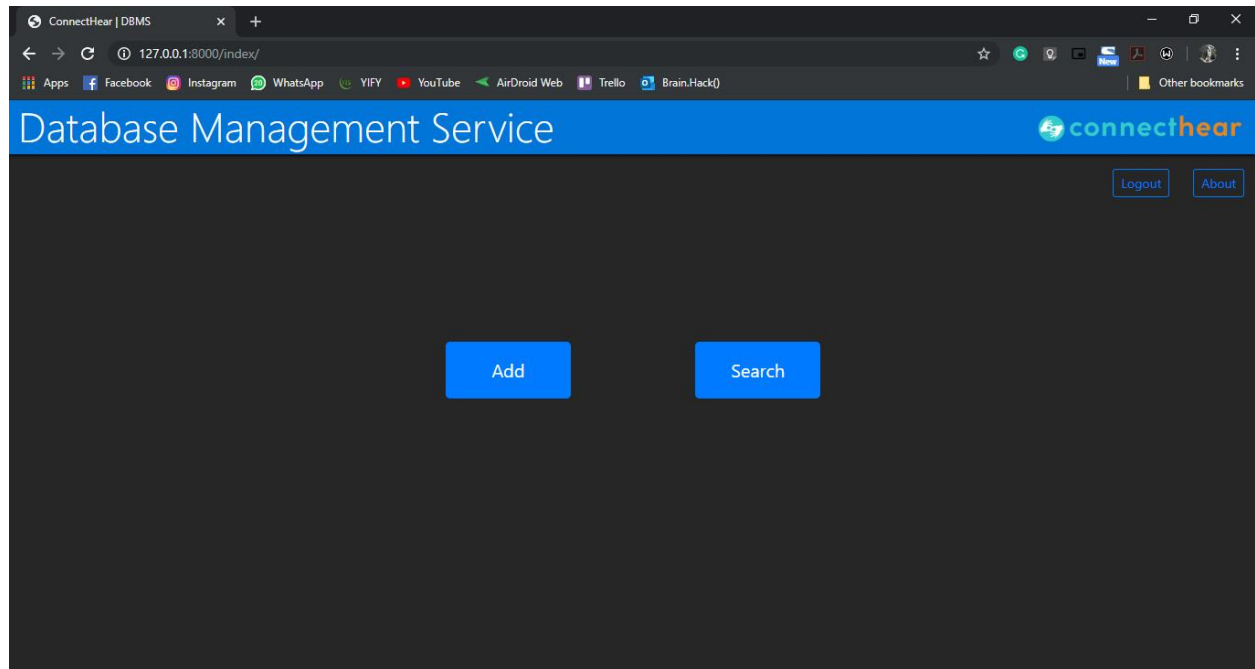
1. The user is presented with the login page as the landing page. The credentials are “admin” for username and “admin” for password by default.



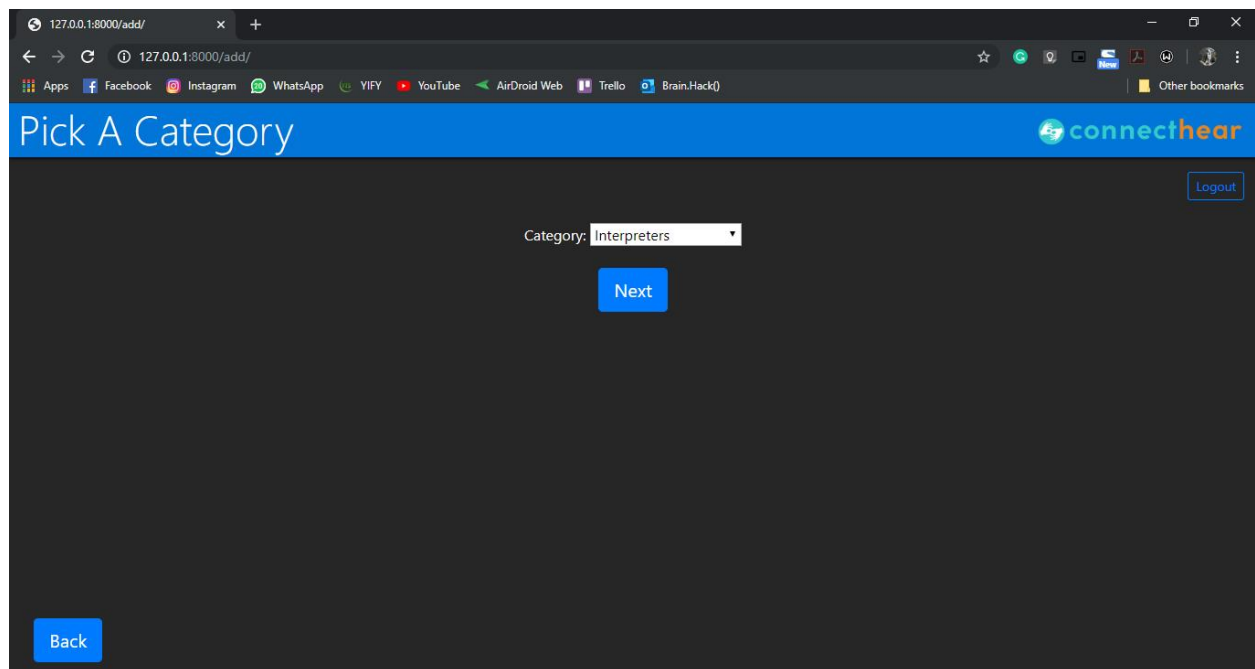
2. If incorrect credentials are added, the user is informed there and then.



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.

The screenshot shows a web browser window with the URL `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" (which is highlighted in blue), "Project", and "Call Record". At the bottom left, there is a blue "Back" button.

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

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. The main content area is dark gray and contains a form with the following fields: "Name:" with the value "Owais 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 the value "Student", "Batch no:" with the value "5", and "Trainer:" with a dropdown menu showing "Azima Dhanjee". At the bottom left, there is a blue "Back" button.

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

127.0.0.1:8000/interpreter/

connecthear

Logout

Name:

Age:  Please fill out this field.

Address:

Mobile no:

Gender:

Date of joining:

Submit

Back

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.

127.0.0.1:8000/search/

connecthear

Logout

Name/Title:

Category:

Search

Back

9. The results are displayed in a tabular form.

127.0.0.1:8000/results/

Results

connecthear

students

Logout

Name	Age	Gender	Mobile Number	Level	Classification	Occupation	City	Batch Number	Trainer
Owais Asad	20	male	3215843150	beginner	friend	Student	karachi	5	Azima Dhanjee

Back

10. There is also an about section.


ConnectHear | About

About Section


connecthear

Logout

Created By:



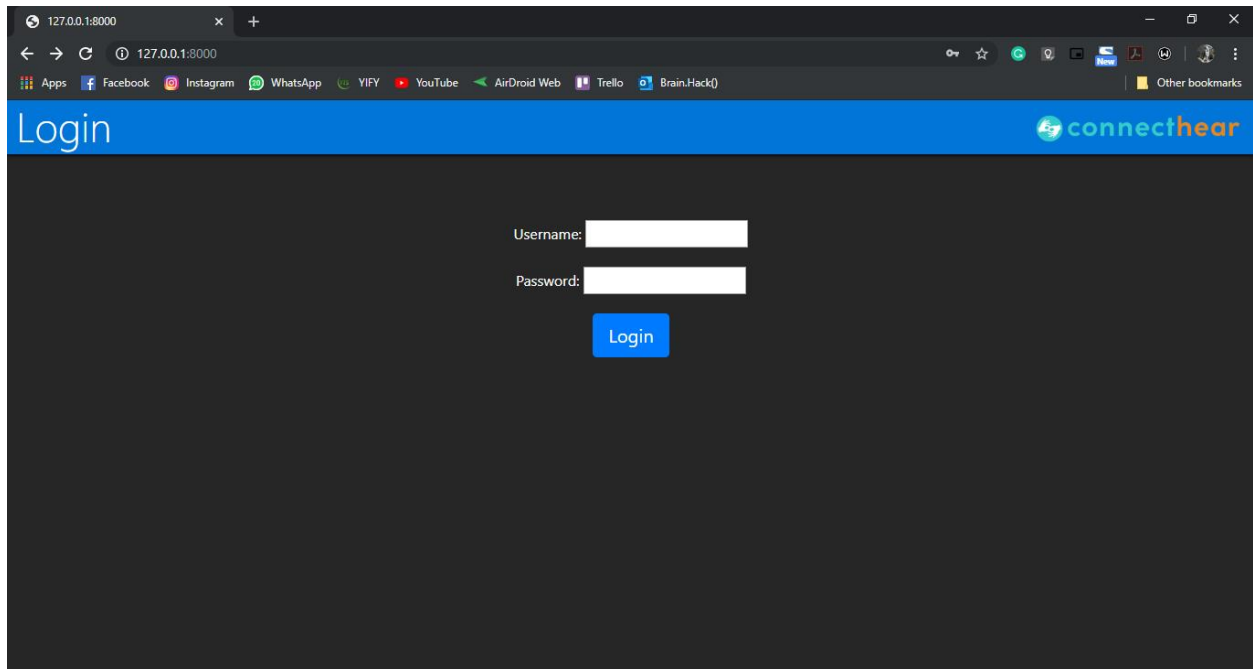
Arhum Ishtiaq



Owais Bin Asad

Back

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:

DB Browser for SQLite - C:\Users\Arhum Ishtiaq\Desktop\Habib University - Arhum Ishtiaq - ai05182\Fall 2019\DBS\Project\course-project-kashmiri\_chah\Project Prototype\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	Nazia	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

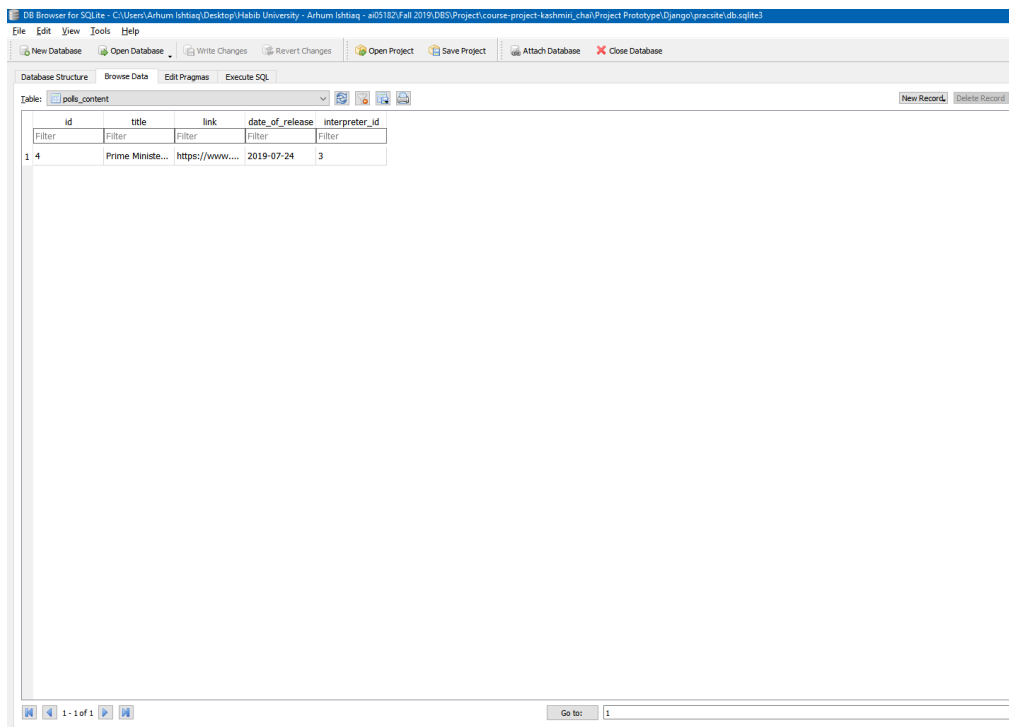
1 - 17 of 17

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:



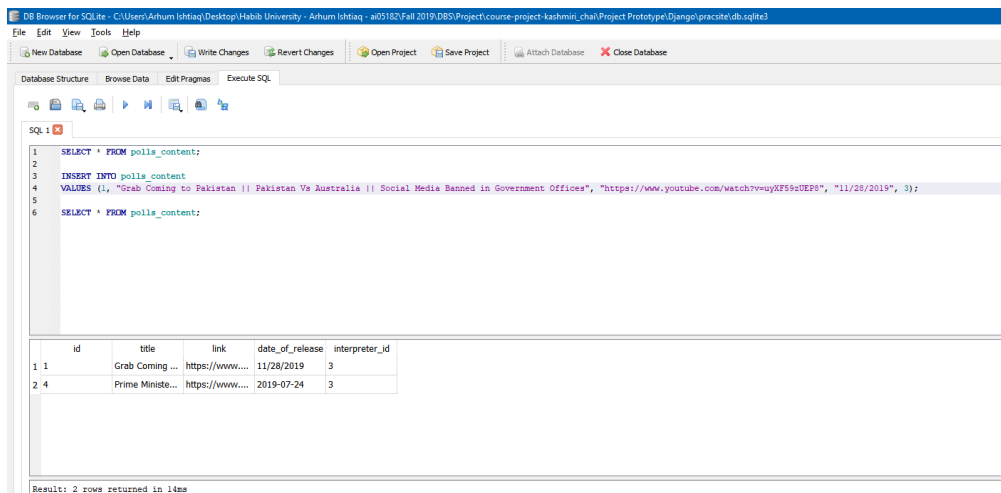
The initial state of the “*content*” database:



The screenshot shows the DB Browser for SQLite interface. The table 'polls\_content' is selected, and its structure is displayed. The table has five columns: id, title, link, date\_of\_release, and interpreter\_id. The data is as follows:

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 INSERT query. The SQL editor contains the following code:

```
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=yuXF59dUEP8", "11/28/2019", 3);
5
6 SELECT * FROM polls_content;
```

The result set shows two rows:

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

Result: 2 rows returned in 14ms