

CLIENT – SERVER WORKING

CLIENT SIDE:

FEATURES

- Client can search through any of the attribute present in the dataset.
- Client can decide which attributes he wants in the output.
- Will give all the records in output which are starting with the searched value.
- A unique id is maintained for all query response packet transactions between client and server.
- 'Time out' time for the client is 3 seconds.

WORKING

```
PS C:\Users\User\Desktop\server-client> python3 client.py 'ja' 0 3
Name      : Jatin Kumar Tomar
Email Id   : 2018ucs0072@iitjammu.ac.in

Name      : Jayesh Khistariya
Email Id   : 2018ucs0084@iitjammu.ac.in

Name      : Jaswant Patel
Email Id   : 2018uee0116@iitjammu.ac.in
```

In the above example the command is : “python3 client.py ‘ja’ 0 3” the parameter passed is ” ‘ja’ - 0 - 3 ” < = > “ ‘search_value’ - search_attribute - response_attribute ”

search_value: value you want to search. To search name or email or phone number just enter the value but to search the blood group or the department follow the following mapping:

blood_group_mapping = {0:"A+", 1:"B+", 2:"A-", 3:"B-", 4:"AB-", 5:"AB+", 6:"O+", 7:"O-"}

department_mapping = {0:"Chemical Engineering", 1:"Civil Engineering", 2:"Computer Science Engineering", 3:"Electrical Engineering", 4:"Mechanical Engineering"}

Search_attribute: attribute under which you want to search the value, their mapping is as follows :

- NAME_SEARCH = 0
- EMAIL_SEARCH = 1
- PHONE_SEARCH = 2

- DEPARTMENT_SEARCH = 3
- REG_NO_SEARCH = 4
- BLOOD_GROUP_SEARCH = 5

response_attribute: The response_attribute helps you to set the attributes you want in the response, it is the decimal representation of a byte whose bit's are set '1' and '0', '1' means that attribute is required and '0' otherwise. The bits are assigned the following attribute from the least significant to most significant: Name, Email, Phone no., department, Registration no., Blood group.

In the example the search_value is set as 'ja' and the search_attribute is '0' and '0' is mapped to 'name' which means it will search all names and get the names starting with 'ja', search_attribute is set as '3' < = > '00000011' here name and email bit's are set so in response we will get the name and email of all the names starting with 'ja' .

With this parameters a query packet is made and sent to server, the server send response packet's if the id is matching no packet is lost then all the data from response packets is combined and printed and in other case query packet is sent again to server.

QUERY PACKET

Here id is binary encoded unique id_number maintained between server and client, query_type is binary encoding of search_attributr, response_type is the binary encoding of response_attribute and value is binary encoding of search_value.

ID	4 bytes
query_type	1 byte
response_type	1 byte
value	-

SERVER SIDE:

FEATURES

- The size of the response packet is set as 200 bytes.
- We can send up to 255 response packets for a single query.
- A unique id is maintained for all query response packet transactions between client and server.
- Response is sent through multiple response packets each of size 200 bytes.

WORKING

Server receives the query packet and decodes the packet getting `Id`, `search_value`, `search_attribute` and `response_attribute`, based on this it computes data to send.

After encoding this data to binary it is divided in chunks of 194 bytes and all chunks are added to the packets and are sent to client's address.

RESPONSE PACKET

Here `id` is binary encoded unique `id_number` maintained between server and client, `Total packets` is an binary encoded integer denoting total number of packets client will receive for a query, `Remaining packets` is an binary encoded integer denoting number of packets client will receive after this packet, and `data` is the binary encoded data chunk.

ID	4 byte's
Total packets	1 byte
Remaining packets	1 byte
Data	194 byte's