Project Development Phase
Sprint 4

Date	18 November 2022
Team ID	PNT2022TMID35942
Project Name	Project – AI based discourse for Banking Industry

In sprint 4, we have focused on developing a database to store user data and improving the features of our chatbot by adding new actions to it.

We have used **IBM cloudant** to create the required databases for our model.

Offers, current account balance, feedback data storage and net banking registration are the new features added in this sprint. Links for current account balance, feedback form and net banking registration form is provided by the chatbot as response and they are also provided in the webpage created.

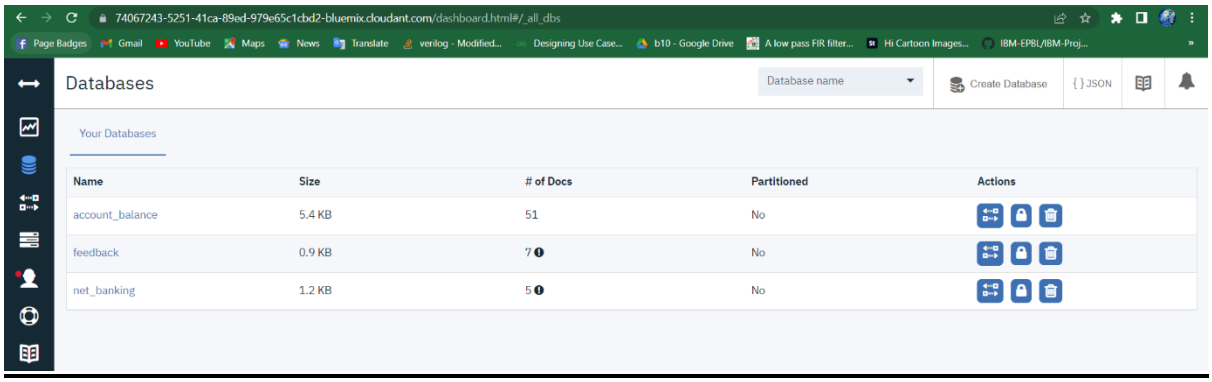
IMPORTING REQUIRED LIBRARIES:

```
from flask import Flask,render_template
from flask import *
from cloudant.client import Cloudant
from cloudant.result import Result
```

INTEGRATING IBM CLOUDANT DATABASE:

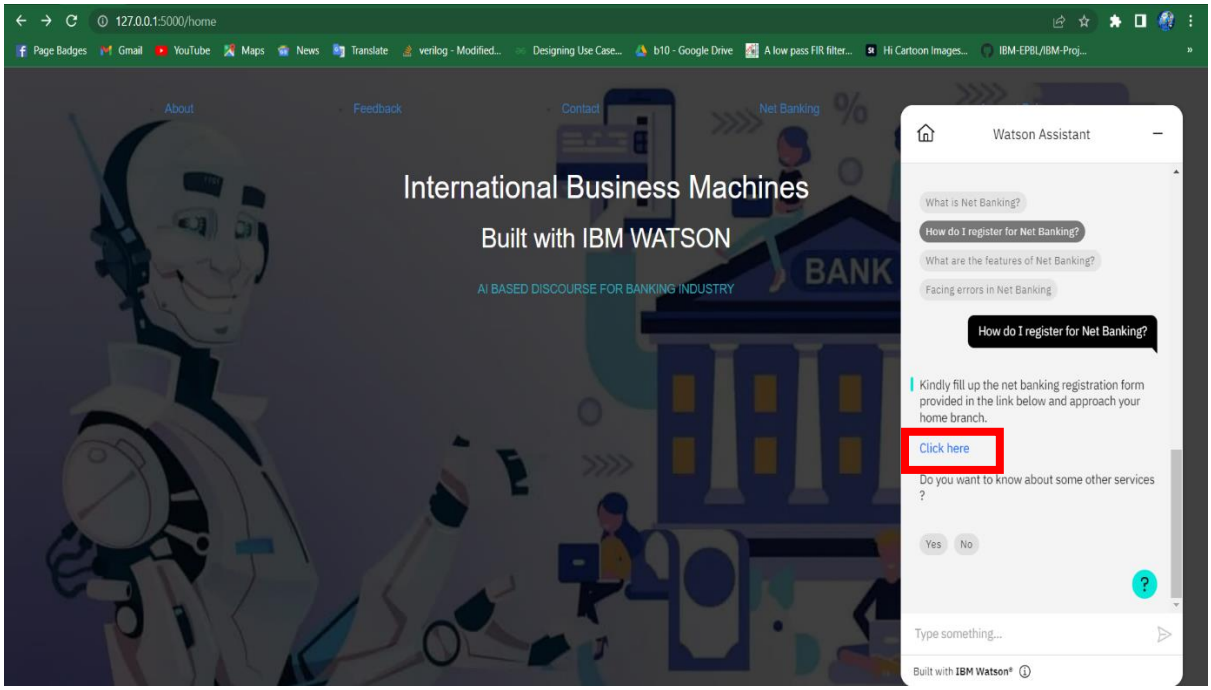
```
ACCOUNT_NAME = "74067243-5251-41ca-89ed-979e65c1cbd2-bluemix"
API_KEY = "IqWgY3pe1n9DGN4pN_9uSXzmlCs27ML4DkcTAePwkFbv"

client = Cloudant.iam(ACCOUNT_NAME,API_KEY,connect=True)
```

DIFFERENT DATABASES CREATED:



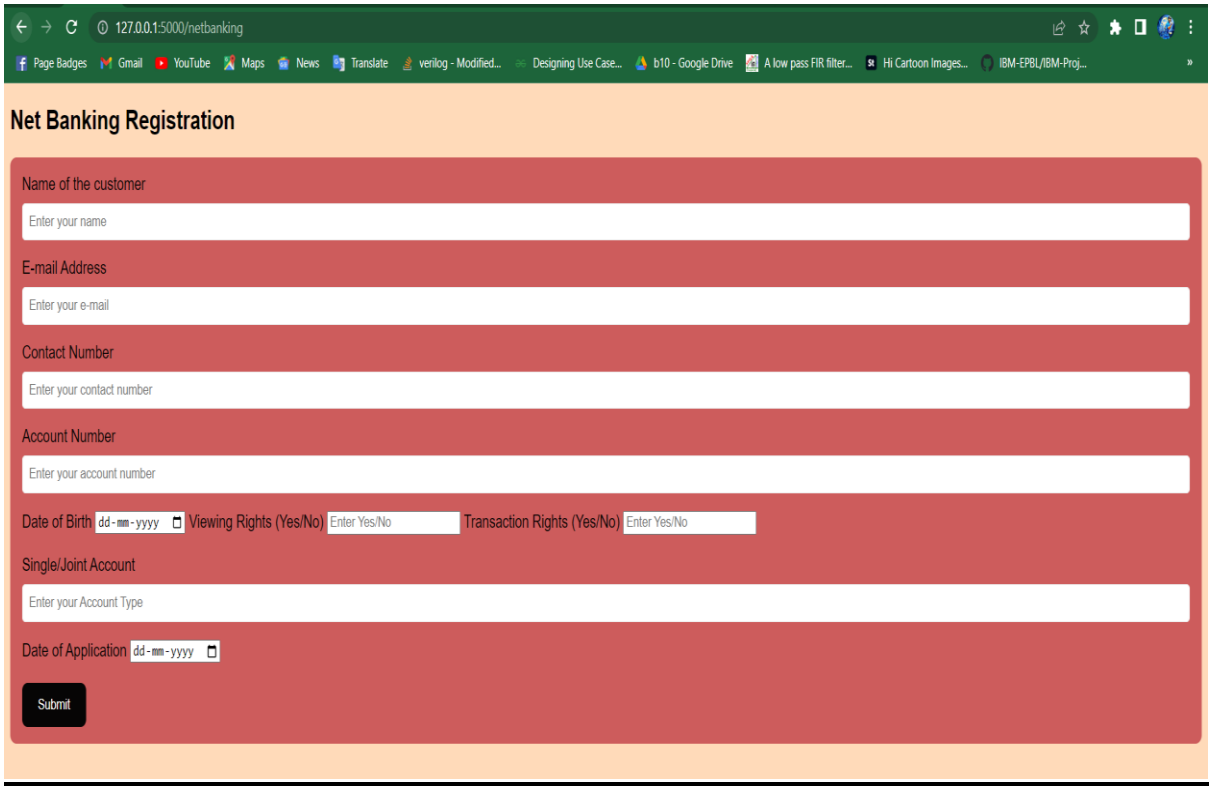
NET BANKING:

We have added a link in the chatbot which is given as a response when the user asks about net banking registration.



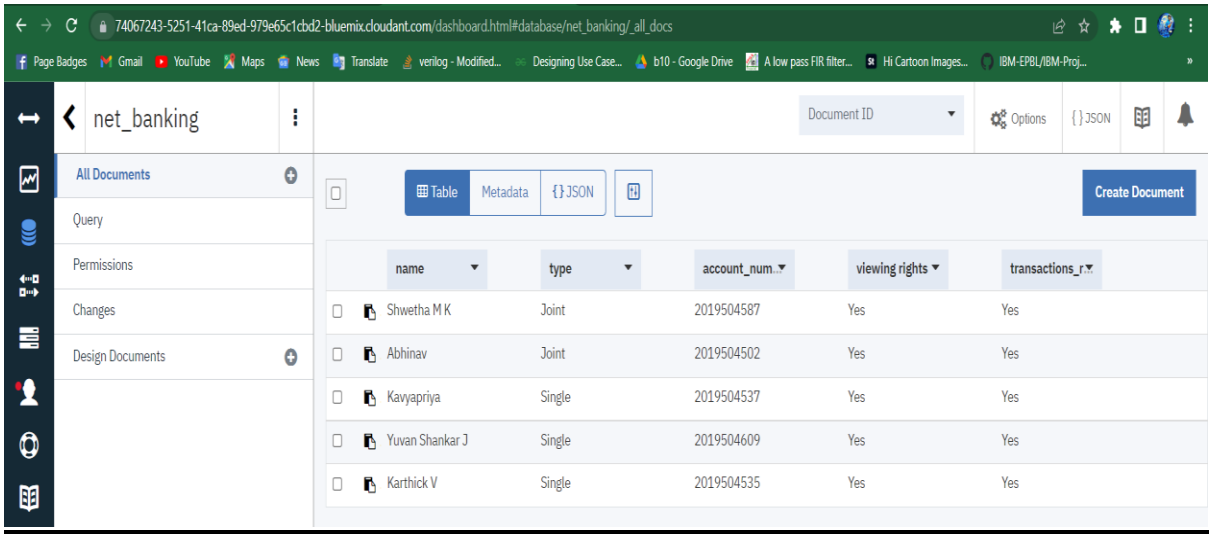
The link “click here” will direct the user to net banking registration form.

NET BANKING REGISTRATION FORM:



STORAGE OF USER DATA PROVIDED IN IBM CLOUDANT DATABASE:

The data provided by the users in the registration form is stored in the net_banking database as shown below.



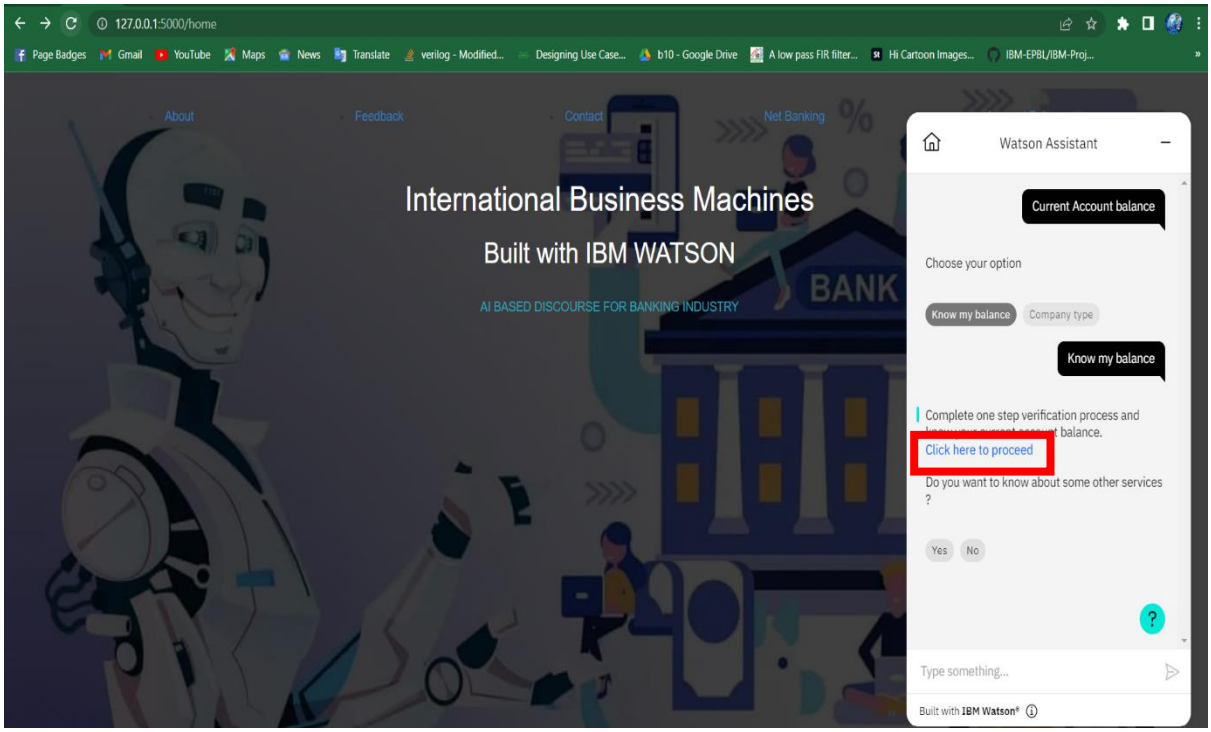
PYTHON CODE FOR USER DATA STORAGE:

```
@app.route('/netbanking',methods = ['GET','POST'])
def netbanking():
    print("******")
    if request.method == "POST":
        email = request.form['mail']
        name = request.form['name']
        acc_num = request.form['num']
        contact_number=request.form['number']
        date_of_birth=request.form['dob']
        date_of_application=request.form['doa']
        viewing_rights=request.form['viewing']
        transaction_rights=request.form['transaction']
        type_of_account=request.form['type']
        jsonDocument= {
            'email':email,
            'name':name,
            'account_number':acc_num,
            'contact_numer':contact_number,
            'viewing rights':viewing_rights,
            'transactions_rights':transaction_rights,
            'date_of_birth':date_of_birth,
            'date_of_application':date_of_application,
            'type':type_of_account
            }
        newDocument = mydatabase1.create_document(jsonDocument)
        result = Result(mydatabase1.all_docs,include_docs=True)
        print(result[0])
        return redirect(url_for('home'))
        print('#####')
    return render_template('netbanking.html')
```

ACCOUNT BALANCE:

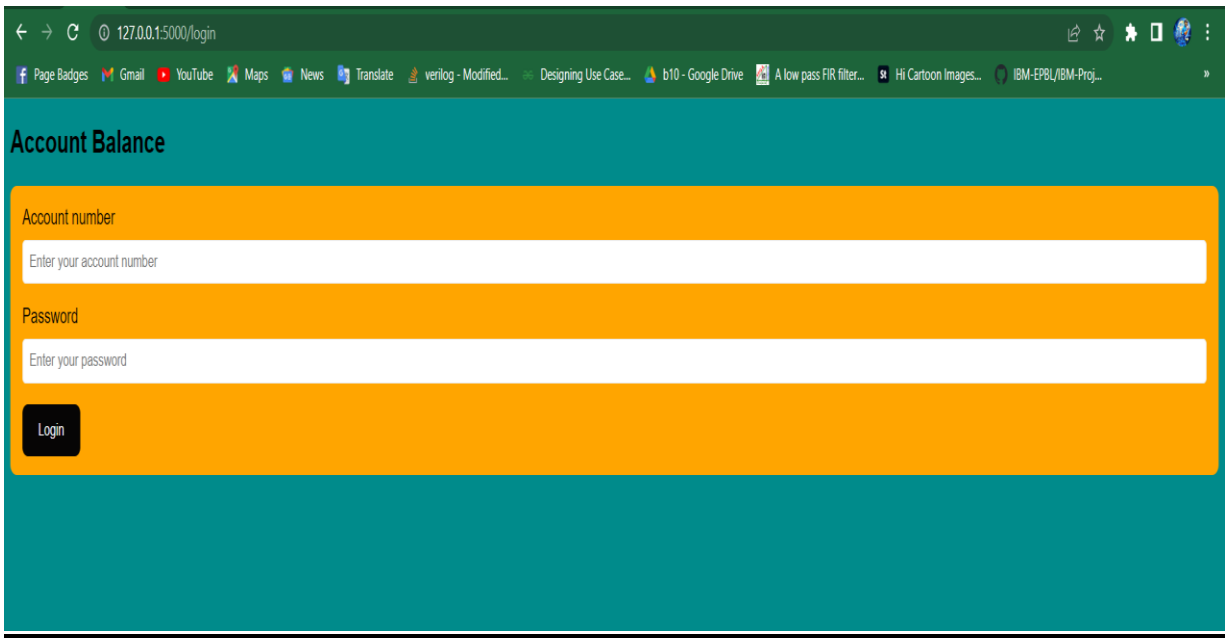
We have added a new feature for the users to find their account balance with one step authentication.

When the user asks for his account balance, a link appears as a response which will direct the user to the login page.



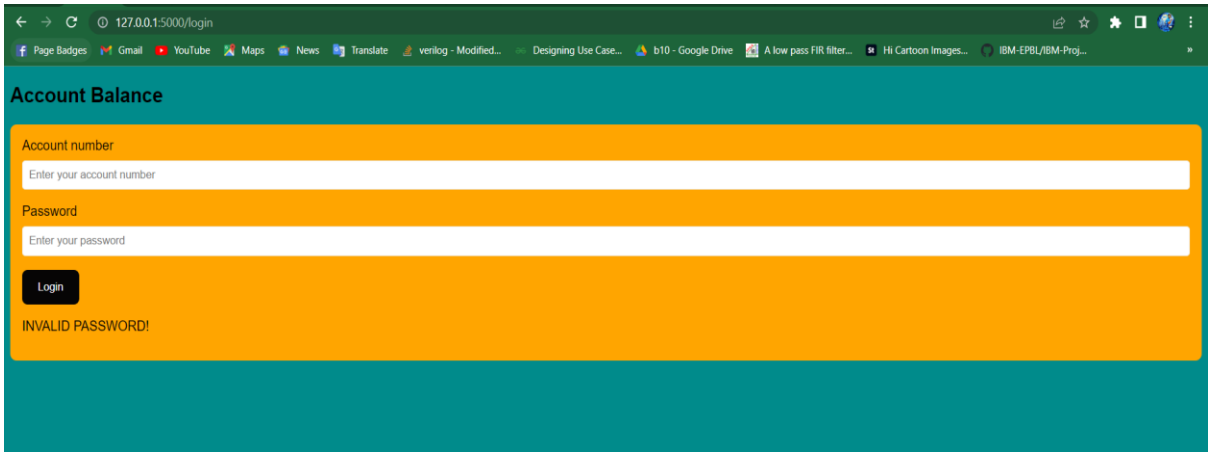
ACCOUNT BALANCE WEBPAGE:

The ‘click here’ link shown in the above image directs the user to the login page.

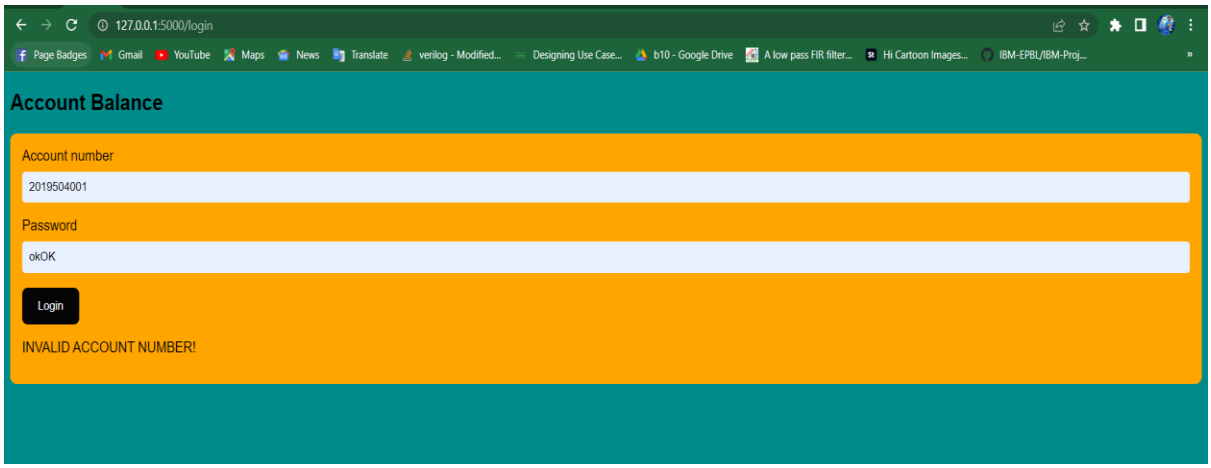


The information given by the user is validated and the following messages are shown based on the result of validation.

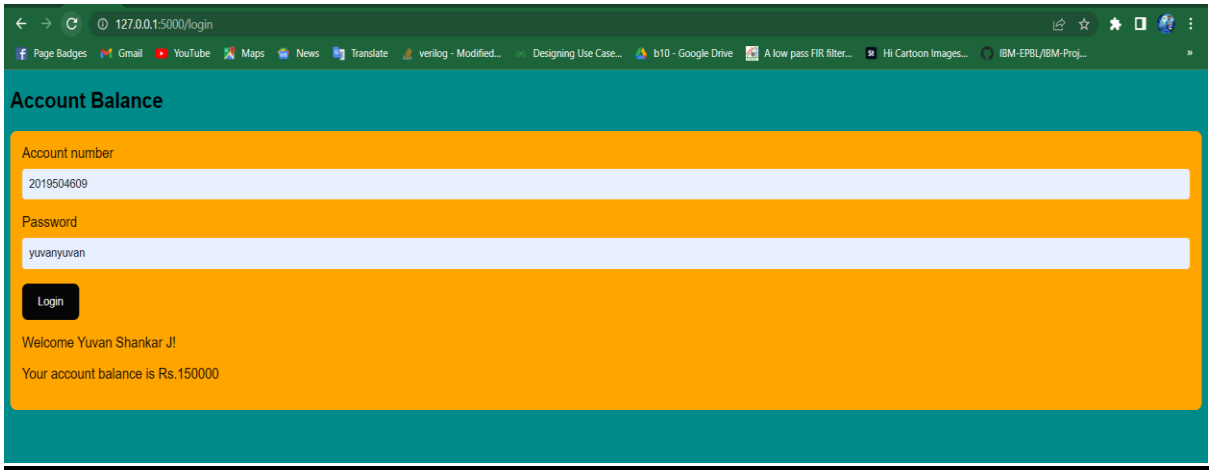
CASE 1: ACCOUNT NUMBER AND PASSWORD MISMATCH:



CASE 2: ACCOUNT NUMBER IS INVALID



CASE 3: ACCOUNT NUMBER AND PASSWORD MATCH

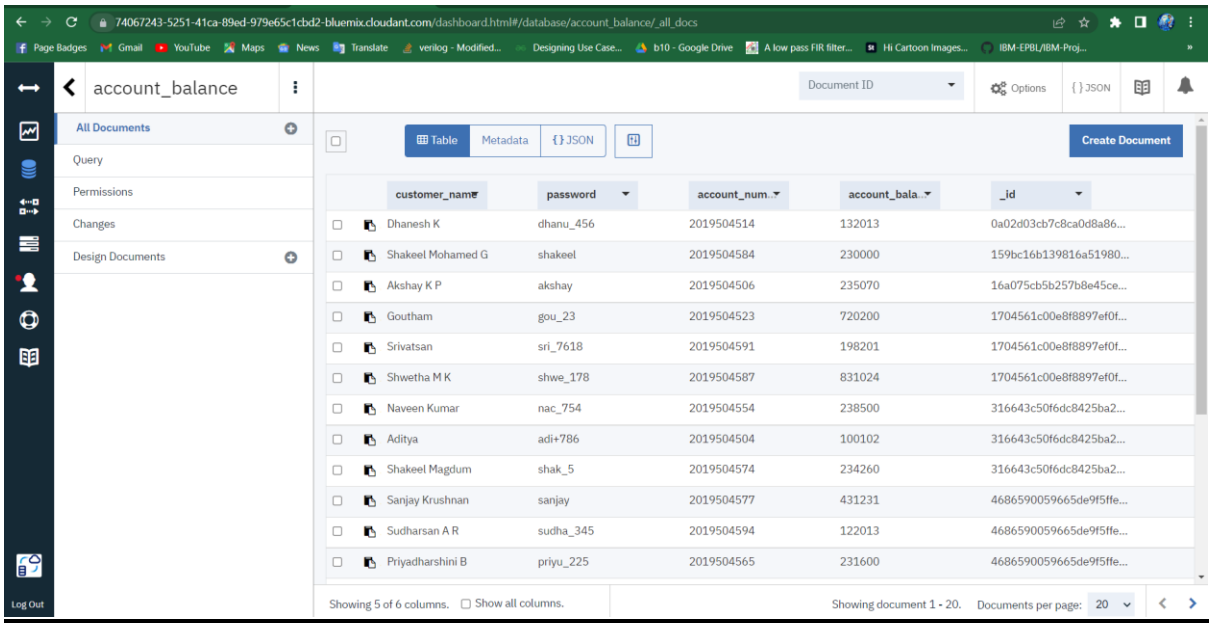


PYTHON CODE FOR USER DATA VALIDATION:

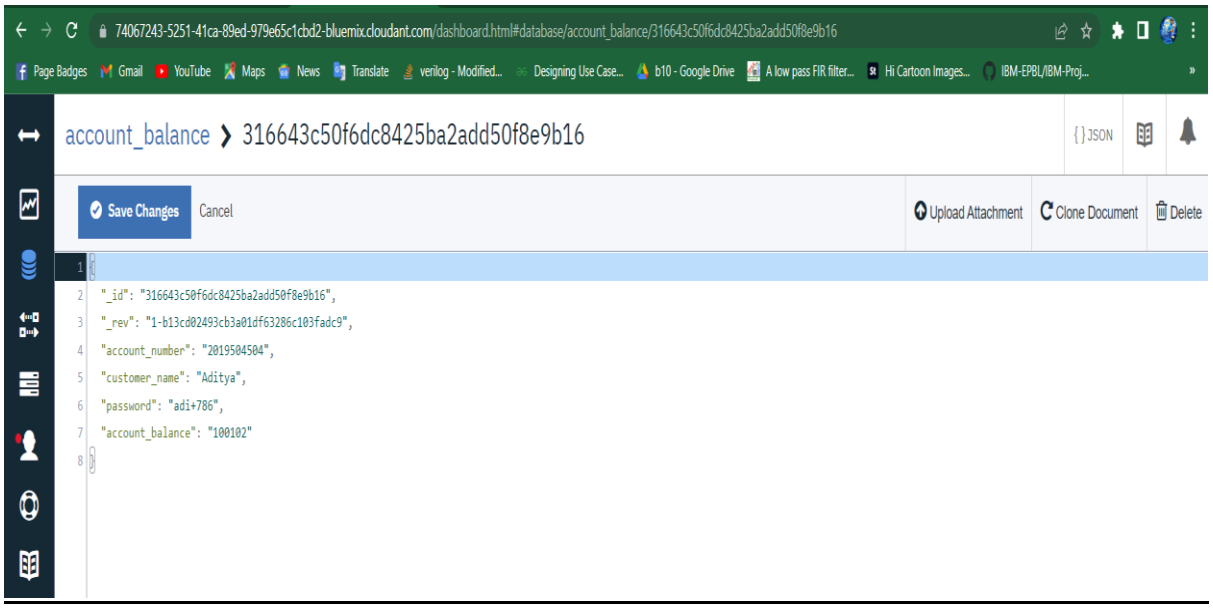
```python
mydatabase2=client.create_database('account_balance')
@app.route('/login',methods=['GET','POST'])
def login():
    if request.method=='POST':
        account_number=request.form['accnum']
        password=request.form['pass']
        account_found=False
        for documents in mydatabase2:
            if documents['account_number']==account_number:
                if documents['password']==password:
                    flash_name='Welcome'+' '+documents['customer_name']+'!'
                    flash(flash_name)
                    flash_text='Your account balance is'+' '+"Rs."+documents['account_balance']
                    flash(flash_text)
                else:
                    flash('INVALID PASSWORD!')
                account_found=True
                break
        if not account_found:
            flash('INVALID ACCOUNT NUMBER!')
    return render_template('login.html')
```

ACCOUNT BALANCE DATABASE:

The input data provided by the user is validated using the python code above. Validation is done by comparing the inputs given and the data available in the database. If the account number and password provided matches with account number and password of any document in the database, the corresponding account balance is given as output.

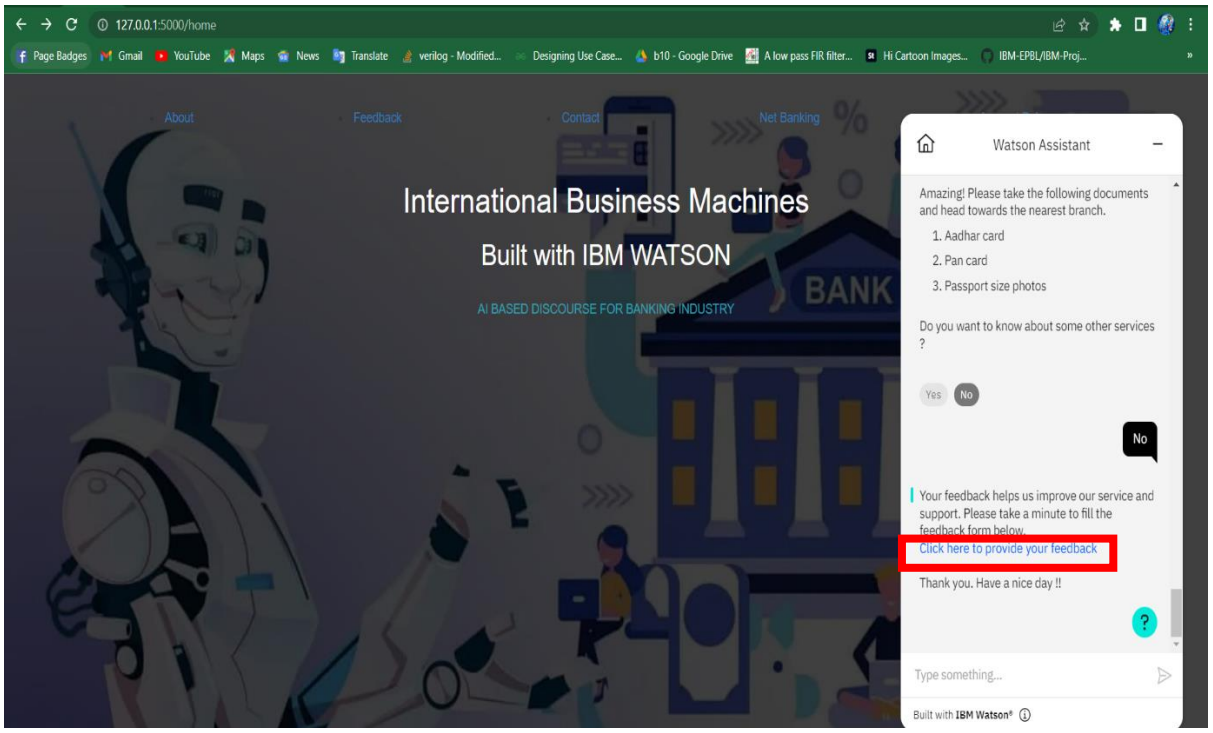


A SAMPLE DOCUMENT STORED IN THE ACCOUNT BALANCE DATABASE:



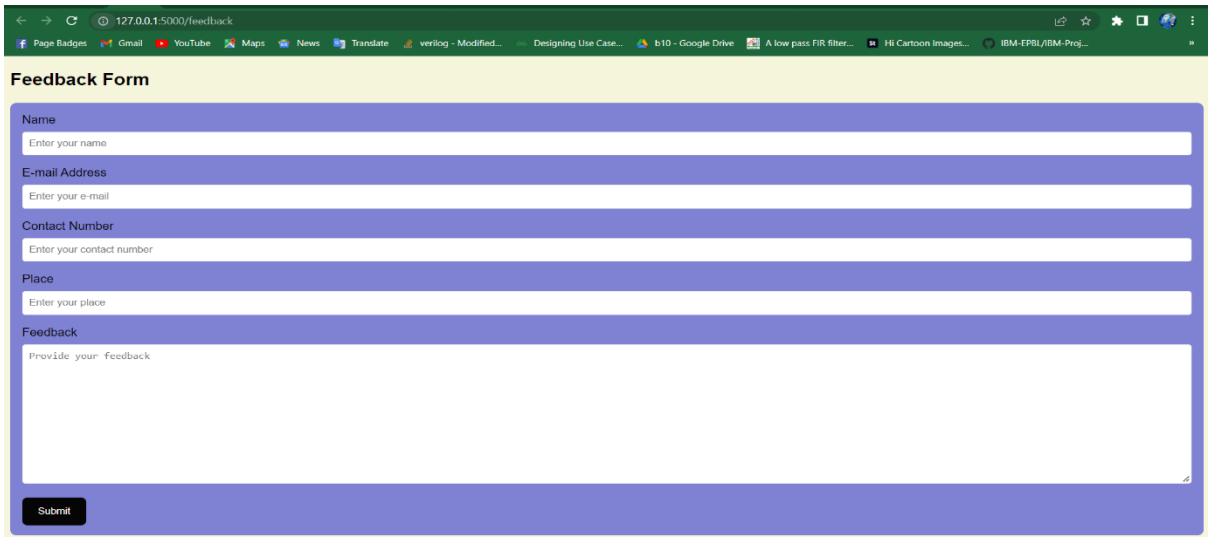
FEEDBACK:

A feedback form has been provided for the users at the end of a conversation to share their experience and mention any difficulties faced by them while they use the chatbot.



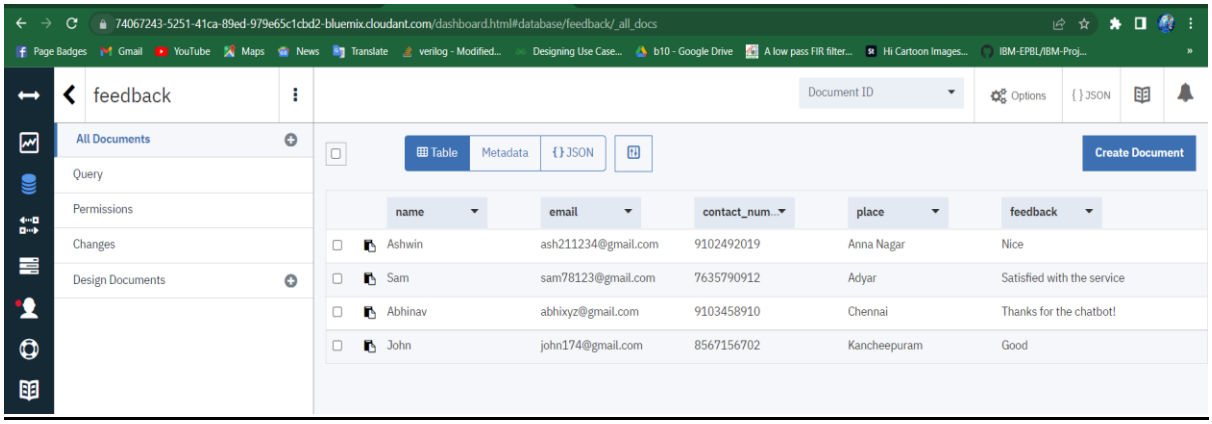
The link ‘Click here to provide your feedback’ will direct the user to feedback form webpage.

FEEDBACK FORM WEBPAGE:



STORAGE OF USER FEEDBACKS IN IBM CLOUDANT DATABASE:

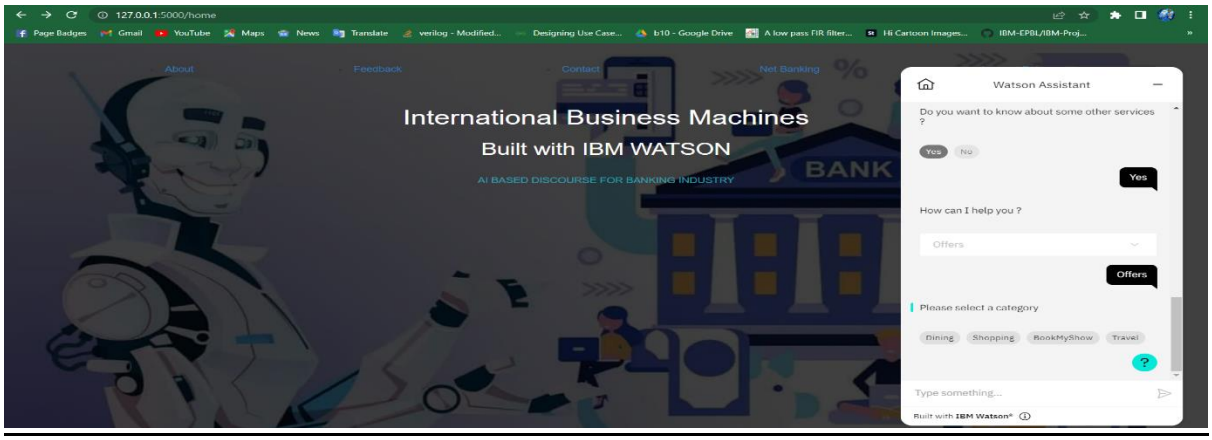
Data provided by the users is stored in the feedback database as shown below.



PYTHON CODE FOR FEEDBACK DATA COLLECTION:

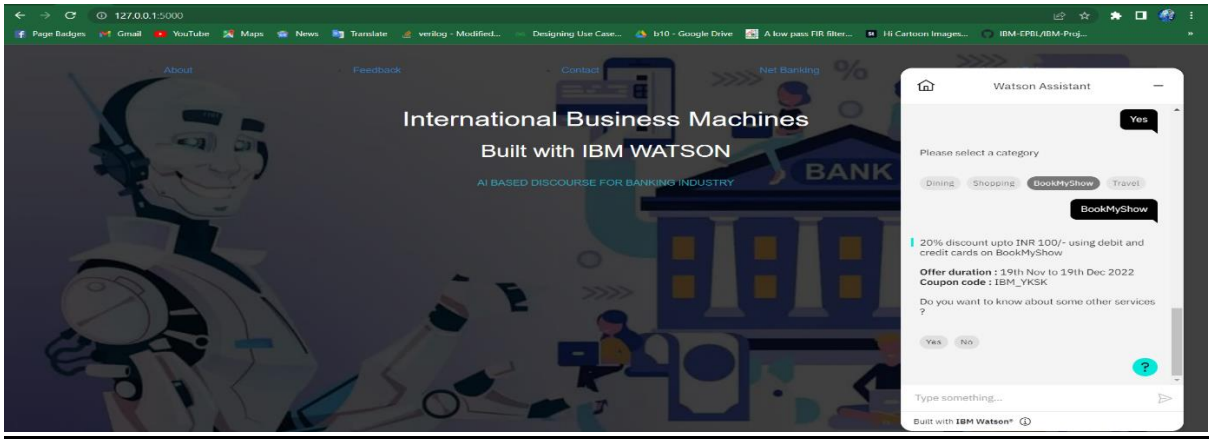
```
@app.route('/feedback',methods = ['GET','POST'])
def feedback():
    print("******")
    if request.method == "POST":
        email = request.form['mail']
        name = request.form['name']
        num = request.form['num']
        place= request.form['place']
        msg = request.form['message']
        #print(email,name,num,place,msg)
        jsonDocument= {
            'email':email,
            'name':name,
            'contact_number':num,
            'place':place,
            'feedback':msg
            }
        newDocument = mydatabase.create_document(jsonDocument)
        result = Result(mydatabase.all_docs,include_docs=True)
        print(result[0])
        return redirect(url_for('home'))
        print('#####')
    return render_template('feedback.html')
```

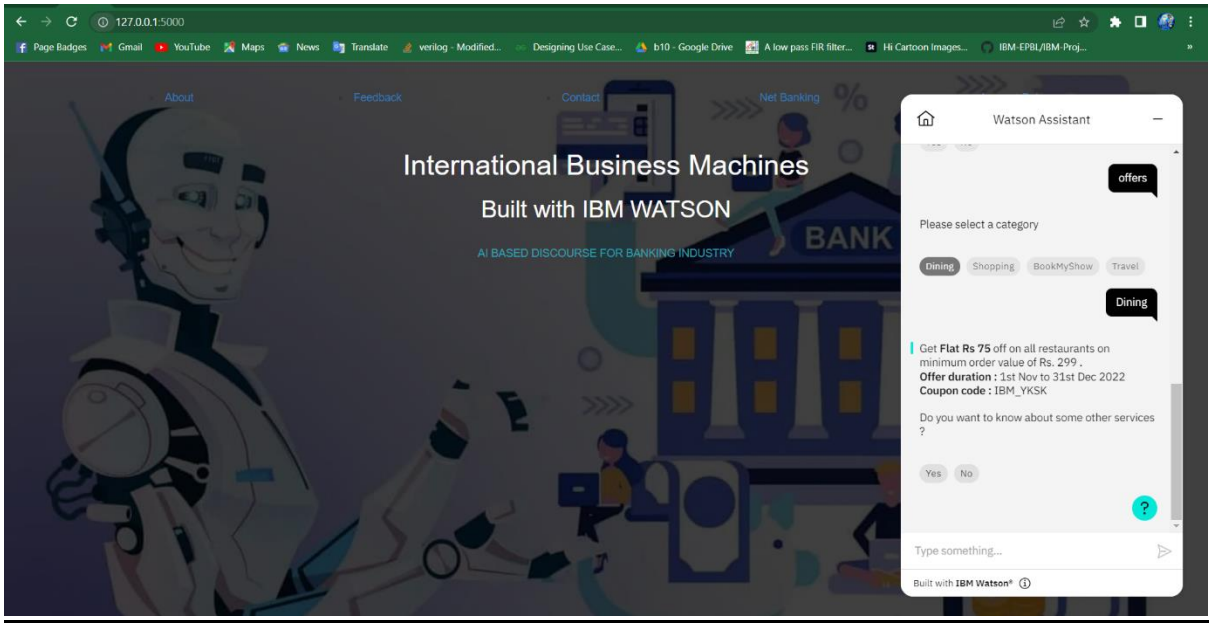
OFFERS:


127.0.0.1:5000/home
About
Feedback
Contact
Net Banking
International Business Machines
Built with IBM WATSON
AI BASED DISCOURSE FOR BANKING INDUSTRY
Watson Assistant
Do you want to know about some other services ?
Yes
No
Yes
How can I help you ?
Offers
Offers
Please select a category
Dining
Shopping
BookMyShow
Travel
Type something...
Built with IBM Watson


127.0.0.1:5000
About
Feedback
Contact
Net Banking
International Business Machines
Built with IBM WATSON
AI BASED DISCOURSE FOR BANKING INDUSTRY
Watson Assistant
Please select a category
Dining
Shopping
BookMyShow
Travel
Travel
Upto 15% Instant Discount on Flights and Hotels.
The offer is applicable on all Variants of Credit Cards.
Offer duration : 1st Nov 2022 to 31st Dec 2022
Coupon code : IBM_YKSK
Do you want to know about some other services ?
Yes
No
Type something...
Built with IBM Watson


About
Feedback
Contact
Net Banking
International Business Machines
Built with IBM WATSON
AI BASED DISCOURSE FOR BANKING INDUSTRY
Watson Assistant
Offers
Please select a category
Dining
Shopping
BookMyShow
Travel
Shopping
Flat 15% Instant Discount upto 100 INR on all Debit Card and Credit Card
Offer duration : 1st Nov 2022 to 31st Jan 2023
Coupon code : IBM_YKSK
Do you want to know about some other services ?
Yes
No
Type something...
Built with IBM Watson


127.0.0.1:5000
About
Feedback
Contact
Net Banking
International Business Machines
Built with IBM WATSON
AI BASED DISCOURSE FOR BANKING INDUSTRY
Watson Assistant
Yes
Please select a category
Dining
Shopping
BookMyShow
Travel
BookMyShow
20% discount upto INR 100/- using debit and credit cards on BookMyShow
Offer duration : 19th Nov to 19th Dec 2022
Coupon code : IBM_YKSK
Do you want to know about some other services ?
Yes
No
Type something...
Built with IBM Watson


127.0.0.1:5000
About
Feedback
Contact
Net Banking
International Business Machines
Built with IBM WATSON
AI BASED DISCOURSE FOR BANKING INDUSTRY
Watson Assistant
offers
Please select a category
Dining
Shopping
BookMyShow
Travel
Dining
Get Flat Rs 75 off on all restaurants on minimum order value of Rs. 299 .
Offer duration : 1st Nov to 31st Dec 2022
Coupon code : IBM_YKSK
Do you want to know about some other services ?
Yes
No
Type something...
Built with IBM Watson®