

Project Development Phase

Delivery of Sprint - 6

Date	19 November 2022
Team ID	PNT2022TMID29862
Project Name	AI-based discourse for Banking Industry

Creating Assistant & Integrate with Flask Web Page

You will be creating a banking bot in this activity that has the following capabilities

1. The Bot should be able to guide a customer to create a bank account.
2. The Bot should be able to answer loan queries.
3. The Bot should be able to answer general banking queries.
4. The Bot should be able to answer queries regarding net banking.
5. With the help of this bot, you can get all the required details related to banking.

Let us build our flask application which will be running in our local browser with a user interface.

In the flask application, users will interact with the chatbot, and based on the user queries they will get the outcomes.

Build Python Code

```
import email
from email import message
from importlib.resources import contents
from tkinter import S
from turtle import title
from flask import Flask, redirect,render_template, request,session, url_for, Flask
from pyexpat import model
from werkzeug.utils import secure_filename
import ibm_db
```

```

from flask_mail import Mail, Message

from markupsafe import escape

from flask import Flask,render_template,request

import requests

app = Flask(__name__)

app.secret_key = b'_5#y2L"F4Q8z\n\Xec]/'

mail = Mail(app)

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-
ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=
SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=hzj88231;PWD=z8f4ZiZ171T0FvR
1","")

print(conn)

print("connection successful...")

@app.route('/', methods = ['GET','POST'])

def signup():

    return render_template('signup.html')

@app.route('/login', methods=['GET','POST'])

def login():

    return render_template('login.html')

@app.route('/index')

def index():

    return render_template('index.html')

@app.route('/account')

def account():

    return render_template('account.html')

@app.route('/aboutus')

def aboutus():

    return render_template('aboutus.html')

```

```

@app.route('/services')

def services():

    return render_template('services.html')

@app.route('/register', methods=['GET', 'POST'])

def register():

    if request.method == 'POST':

        uname = request.form['uname']

        mail = request.form['email']

        phone = request.form['phone']

        password = request.form['password']

        sql = "SELECT * FROM customer WHERE email=?"

        stmt = ibm_db.prepare(conn, sql)

        ibm_db.bind_param(stmt,1,mail)

        ibm_db.execute(stmt)

        account = ibm_db.fetch_assoc(stmt)

        if account:

            return render_template('index.html', msg="You are already a member, please login using
your details....")

        else:

            insert_sql = "INSERT INTO customer VALUES (?, ?, ?, ?)"

            prep_stmt = ibm_db.prepare(conn, insert_sql)

            ibm_db.bind_param(prepare_stmt, 1, uname)

            ibm_db.bind_param(prepare_stmt, 2, mail)

            ibm_db.bind_param(prepare_stmt, 3, phone)

            ibm_db.bind_param(prepare_stmt, 4, password)

            ibm_db.execute(prepare_stmt)

            return render_template('login.html', msg="Student Data saved successfully..")

```

```

@app.route('/signin', methods=['GET', 'POST'])
def signin():
    sec = "

    if request.method == 'POST':
        mail = request.form['email']
        password = request.form['password']

        sql = f"select * from customer where email='{escape(mail)}' and password=
'{escape(password)}'"

        stmt = ibm_db.exec_immediate(conn, sql)

        data = ibm_db.fetch_both(stmt)

        if data:
            session["mail"] = escape(mail)
            session["password"] = escape(password)
            return redirect(url_for('index'))

        else:
            return render_template('login.html',msg = "Invalid email/ Password or Not registered!!?")

    return "not going to happen dickhead!!??"

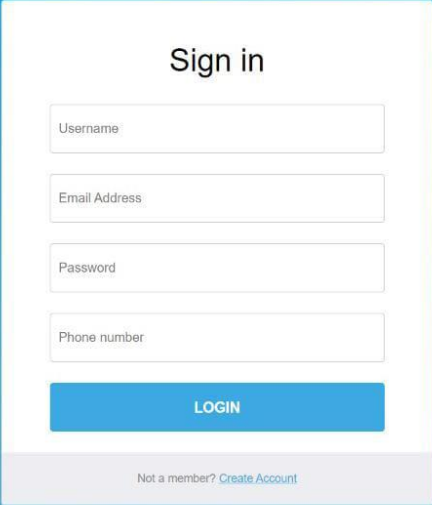
if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)

```

Build HTML Code

- We use HTML to create the front-end part of the web page.
- Here, we have created HTML page-index.html
- Chatbot.html displays the home page which integrates with Watson Assistant.
- A simple HTML page is created. Auto-generated source code from IBM Watson Assistants is copied and pasted inside the body tag

Run The Application



A sign-in form titled "Sign in" is centered on a blue gradient background. The form is a white card with a light gray shadow. It contains four input fields: "Username", "Email Address", "Password", and "Phone number". Below these fields is a blue "LOGIN" button. At the bottom of the card, there is a link: "Not a member? [Create Account](#)".

Sign in

Username

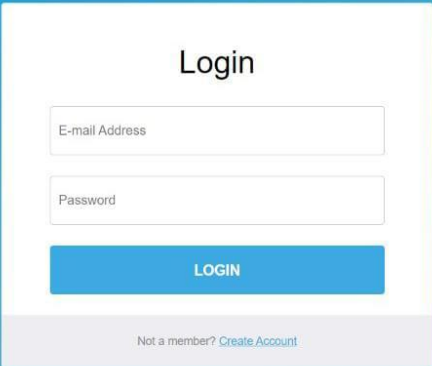
Email Address

Password

Phone number

LOGIN

Not a member? [Create Account](#)



A login form titled "Login" is centered on a blue gradient background. The form is a white card with a light gray shadow. It contains two input fields: "E-mail Address" and "Password". Below these fields is a blue "LOGIN" button. At the bottom of the card, there is a link: "Not a member? [Create Account](#)".

Login

E-mail Address

Password

LOGIN

Not a member? [Create Account](#)

