

Sprint 2

Team Mates

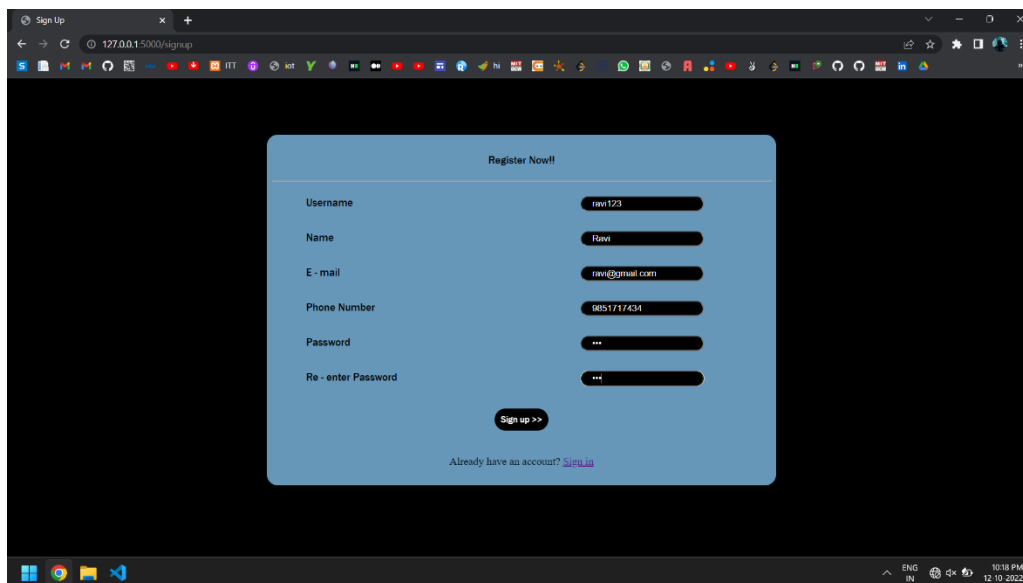
- 1.Akalya.C
- 2.Anusha.S
- 3.Arockia Jersha Tharshali.T
- 4.Rasi Binu.R

Sprint 2:

Completed Tasks:


- Login page
- Sign Up page
- Dashboard page
- Add tickets page
- Sendgrid integration

1. Create an UI to interact with Application



2. Connect with IBM DB2

Agent Table:

USERNAME	NAME	EMAIL	PHN	PASSWORD
 <p>You don't have any data currently</p>				

Customer Table:

Result set - Run SQL - IBM Db2 on Cloud - select * from agent - Google Chrome						
about:blank						
Result set 1						
USERNAME	NAME	EMAIL	PHN	PASSWORD	DOMAIN	STATUS
You don't have any data currently						

Code:

App.py:

```
from flask import Flask, render_template, request, redirect, session
import ibm_db
import re

app = Flask(__name__)

# for connection

try:
    app.secret_key = 'a'
```

```

    conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b70af05b-76e4-4bca-a1f5-
23dbb4c6a74e.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32716;SECURIT
Y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=xyn12614;PWD=oXjOuam0A
0YeLVKq", '', '')
except:
    print("can't connect")

import os

from sendgrid import SendGridAPIClient

from sendgrid.helpers.mail import Mail

message = Mail(

    from_email='from_email@example.com',

    to_emails='to@example.com',

    subject='Sending with Twilio SendGrid is Fun',

    html_content='<strong>and easy to do anywhere, even with Python</strong>')

try:

    sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))

    response = sg.send(message)

    print(response.status_code)

    print(response.body)

    print(response.headers)

except Exception as e:

    print(e.message)

@app.route('/signup', methods=['GET', 'POST'])
def signup():
    global userid
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        name = request.form['name']
        email = request.form['email']
        phn = request.form['phn']

```

```

password = request.form['pass']
repass = request.form['repass']
print("inside checking")
print(name)
if len(username) == 0 or len(name) == 0 or len(email) == 0 or len(phn)
== 0 or len(password) == 0 or len(repass) == 0:
    msg = "Form is not filled completely!!"
    print(msg)
    return render_template('signup.html', msg=msg)
elif password != repass:
    msg = "Password is not matched"
    print(msg)
    return render_template('signup.html', msg=msg)
elif not re.match(r'[a-z]+', username):
    msg = 'Username can contain only small letters and numbers'
    print(msg)
    return render_template('signup.html', msg=msg)
elif not re.match(r'^@+@[^@]+\.[^@]+', email):
    msg = 'Invalid email'
    print(msg)
    return render_template('signup.html', msg=msg)
elif not re.match(r'[A-Za-z]+', name):
    msg = "Enter valid name"
    print(msg)
    return render_template('signup.html', msg=msg)
elif not re.match(r'[0-9]+', phn):
    msg = "Enter valid phone number"
    print(msg)
    return render_template('signup.html', msg=msg)

sql = "select * from users where username = ?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt, 1, username)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print(account)
if account:
    msg = 'Account already exists'
else:
    insert_sql = "insert into users values(?,?,?,?,?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prepare_stmt, 1, username)
    ibm_db.bind_param(prepare_stmt, 2, name)
    ibm_db.bind_param(prepare_stmt, 3, email)
    ibm_db.bind_param(prepare_stmt, 4, phn)
    ibm_db.bind_param(prepare_stmt, 5, password)
    ibm_db.execute(prepare_stmt)
    print("successs")

```

```

        msg = "succesfully signed up"
        return render_template('dashboard.html', msg=msg, name=name)
    else:
        return render_template('signup.html')

@app.route('/dashboard')
def dashboard():
    return render_template('dashboard.html')

@app.route('/login', methods=["GET", "POST"])
def login():
    global userid
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['pass']
        sql = "select * from users where username = ? and password = ?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin'] = True
            session['id'] = account['USERNAME']
            userid = account['USERNAME']
            session['username'] = account['USERNAME']
            msg = 'logged in successfully'
            return render_template('dashboard.html', name=account['name'])
        else:
            msg = 'Incorrect user credentials'
            return render_template('dashboard.html', msg=msg)
    else:
        return render_template('login.html')

@app.route('/admin')
def admin():
    return render_template('admin.html')

@app.route('/about')
def about():
    return render_template('about.html')

```



```

        Password
    </div>
    <div class="textinformright">
        <input type="password" name="pass">
    </div>
</div>

    <br>
    <div>
        <button class="forbutton" type="submit"> Sign In
>></button>
    </div>
</form>
<br>

    <div>
        New user? <a href="/signup">Sign up</a>
    </div>
    <br>
</div>

</div>
</div>

{% endblock %}

```

Dashboard.html:

```

{% extends 'base.html' %}

{% block head %}

<title>
    Dashboard
</title>

{% endblock %}

{% block body %}

<!-- things

    div 1
welcome jetson,    sign out

```

```

    <div 2
your complaints status

add new complaint -->
<br>
<div class="fordashboardtop">
    <div class="fordashboardtopelements1">
        Welcome Ravi,
    </div>
    <div class="fordashboardtopelements2">
        <a href="/login"><button class="forbutton">Sign out</button></a>
    </div>
</div>
<br>
<div class="outerofdashdetails">

    <div class="fordashboarddetails">
        <br>
        <!-- table of customers complaints -->
        <table class="fortable">
            <thead>
                <th class="pad">Complaint Detail</th>
                <th>Assigned Agent</th>
                <th>Status</th>
                <th>More details</th>
            </thead>
            <tbody>
                <tr>

                    <td class="pad">
                        Network Issue
                    </td>
                    <td>
                        Dr.Raj
                    </td>
                    <td>
                        Not Completed
                    </td>
                    <td>
                        <a href="">Click Here >></a>
                    </td>
                </tr>
            </tbody>

        </table>
        <br>

```

```

        <center>
            <div class="fordashboardtopelements2">
                <a href="/login"><button class="forbutton2">Add new complaint
+ </button></a>
            </div>
        </center>
    </div>

</div>

{% endblock %}

```

Base.html:

```

<!DOCTYPE html>

<head>

    <link rel="stylesheet" href="static/css/main.css"/>
    {% block head %} {% endblock %}
</head>

<body>
    {% block body %}

    {% endblock %}
</body>

</html>

```