

Sprint – 1

Login.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Log In</title>
    <link
      rel="stylesheet"
      href="{{ url_for('static', filename='style.css') }}"
    />
  </head>
  <body>
    <div class="card">
      <h1>LOG IN</h1>
      <strong> Don't have an account? <a href="/"><button>Sign
in</button></a></strong>
      <form method="POST" action="">
        <div> <input type="text" placeholder="Username" name="email" required
/></div>
        <div> <input type="password" placeholder="Password" name="password"
required/></div>
        <input type="submit" value="Log In" />
      </form>

      <br />
      <strong>{{msg}}</strong>
    </div>
  </body>
</html>
```

Signup.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Sign Up</title>
    <link
      rel="stylesheet"
      href="{{ url_for('static', filename='style.css') }}"
    />
  </head>
  <body>
```

```

<div class="card">
  <h1>SIGN UP</h1>
  <strong>Already have an account? <a href='/login'><button>Log
in</button></a></strong>
  <form method="POST" action="">
    <div> <input type="text" placeholder="Username" name="user" required/>
</div>
    <div> <input type="text" placeholder="Email" name="email" required/>
</div>
    <div> <input type="password" placeholder="Password" name="password"
required/> </div>
    <input type="submit" value="Sign Up" />
  </form>
  <br/>
  <strong>{{msg}}</strong>
</div>
</body>
</html>

```

Style.css

```

html {
  background: linear-gradient(#85FFBD, #87CEEB);
  height: 100%;
  font-family: 'roboto', sans-serif;
  background-repeat: no-repeat;
  display: flex;
  justify-content: center;
  text-align: center;
}

form {
  text-align: center;
  box-sizing: border-box;
}

form input[type="submit"], button {
  height: 50px;
  width: 200px;
  background: linear-gradient(#85FFBD, #87CEEB);
  border: 1px solid #f2f2f2;
  text-transform: uppercase;
  cursor: pointer;
}

form input[type="text"],
form input[type="password"] {
  max-width: 400px;

```

```

width: 80%;
line-height: 3em;
margin: 1em 2em;
border-radius: 5px;
border: 2px solid #f2f2f2;
outline: none;
padding-left: 10px;
}

input[type="file"] {
display: none;
}

.custom-file-upload {

display: block;
height: 50px;
width: 200px;
background: linear-gradient(#E61F20, #C9000B);
border: 1px solid #f2f2f2;
text-transform: uppercase;
cursor: pointer;
max-width: 400px;
width: 80%;
line-height: 3em;
margin: 1em 4em;
border-radius: 5px;
border: 2px solid #f2f2f2;
outline: none;
margin-left: 300px;
}

}

.card {
margin-top: 100px;
background-color: white;
width: 550px;
height: inherit;
padding: 20px;
border-radius: 12px;
}

```

```

App.py
import ibm_db as db
from flask import Flask, render_template, request, redirect, session, abort
import os
import pathlib
import requests
from dotenv import load_dotenv

```

```

from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
from google.oauth2 import id_token
from google_auth_oauthlib.flow import Flow
from pip._vendor import cachecontrol
import google.auth.transport.requests

# Configure Flask app
app = Flask(__name__)
SECRET_KEY = os.urandom(32)
app.config['SECRET_KEY'] = SECRET_KEY

# Load .env file
load_dotenv()

conn = db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-
a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30756;SECURIT
Y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=wms89097;PWD=tjzW7bTU8
RTK031d",'', '')

# Frequently used variables
SIGN_UP_PAGE_URL = '/'
LOG_IN_PAGE_URL = '/login'

def execute_sql(statement, **params):
    global conn
    stmt = db.prepare(conn, statement)

    param_id = 1
    for key, val in params.items():
        db.bind_param(stmt, param_id, val)
        param_id += 1

    result = ''
    try:
        db.execute(stmt)
        result = db.fetch_assoc(stmt)
    except:
        pass

    return result

@app.route('/signup', methods=['GET', 'POST'])
def signup():
    msg = ''

```

```

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

    duplicate_check = "SELECT * FROM user WHERE email=?"
    account = execute_sql(statement=duplicate_check, email=email)

    if account:
        msg = "There is already an account with this username!"
    else:
        insert_query = "INSERT INTO user values(?, ?, ?)"
        execute_sql(statement=insert_query,
name=user,email=email,password=password)
        print("account creating")
        send_confirmation_mail(email)
        return redirect('/login')
    return render_template('signup.html', msg=msg)

@app.route('/login', methods=['GET', 'POST'])
def login():
    msg = ''
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']

        duplicate_check = "SELECT * FROM user WHERE email=? and password=?"
        account = execute_sql(statement=duplicate_check,
email=email,password=password)
        if account:
            msg = "There is already an account with this username!"
            print(account)
            session['email'] = email
            duplicate_check = "SELECT name FROM user WHERE email=? and
password=?"
            account = execute_sql(statement=duplicate_check,
email=email,password=password)
            session['user']=account["NAME"]

            return redirect(HOME_PAGE_URL)
        else:
            duplicate_check = "SELECT * FROM user WHERE email=?"
            account = execute_sql(statement=duplicate_check, email=email)
            if account:
                msg="Invalid Password"
            else:
                msg="Account not Found"
            return render_template('signup.html',msg=msg)

```

```
return render_template('login.html', msg=msg)
```

SIGN UP

Already have an account? [LOG IN](#)

[SIGN UP](#)

LOG IN

Don't have an account? [SIGN IN](#)

[LOG IN](#)