

index.html

```
const signUpButton = document.getElementById('signUp');
```

```
const signInButton = document.getElementById('signIn');
```

```
const container = document.getElementById('container');
```

```
const switchone = document.getElementById('c1');
```

```
const switchtwo = document.getElementById("c2");
```

```
const switchthree = document.getElementById('c3');
```

```
const switchfour = document.getElementById('c4');
```

```
const Fswitchone = document.getElementById('l1');
```

```
const Fswitchtwo = document.getElementById("l2");
```

```
const Fswitchthree = document.getElementById('l3');
```

```
const Fswitchfour = document.getElementById('l4');
```

```
const space = document.getElementById('infos'); var
```

```
pre_state = 0; var stateone = 0; var statetwo = 0;
```

```
var statethree = 0;
```

```
signUpButton.addEventListener('click', () => {
```

```
  container.classList.add("right-panel-active");
```

```
});
```

```
  signInButton.addEventListener('click', () => { container.classList.remove("right-panel-  
active"); });
```

```
switchone.addEventListener('click', remover);

switchtwo.addEventListener('click', signin); switchthree.addEventListener('click',
Signup) switchfour.addEventListener('click', about);
```

```
Fswitchone.addEventListener('click', remover);

Fswitchtwo.addEventListener('click', signin);

Fswitchthree.addEventListener('click', Signup)

Fswitchfour.addEventListener('click', about);
```

```
function remover() {

    if(pre_state == 1){ pre_state = 0;

space.classList.remove("spaceimp");

document.getElementById("abouts").style.display = "none";

document.getElementById("logins").style.display = "none";

document.getElementById("l1").style.display = "flex";

document.getElementById("l2").style.display = "flex";

document.getElementById("l3").style.display = "flex";

document.getElementById("l4").style.display = "flex";

    }

}
```

```

function div_adder () {  space.classList.add("spaceimp");

document.getElementById("abouts").style.display = "none";

document.getElementById("logins").style.display = "block";

document.getElementById("l1").style.display = "none";

document.getElementById("l2").style.display = "none";

document.getElementById("l3").style.display = "none";

document.getElementById("l4").style.display = "none";

}

```

```

function about_adder () {

    //space.classList.add("spaceimp");

    // remover();

document.getElementById("abouts").style.display = "block";

document.getElementById("l1").style.display = "none";

document.getElementById("l2").style.display = "none";

document.getElementById("l3").style.display = "none";

document.getElementById("l4").style.display = "none";

}

```

```

function signin() {

if(pre_state == 0) { pre_state

= 1; stateone = 1; statetwo =

0;  statethree = 0;

container.classList.remove("ri

```

```

ght-panel-active");

div_adder(); }else {

if(stateone == 0) {

pre_state = 1;    stateone =

1;    statetwo = 0;

statethree = 0;

container.classList.remove("ri

ght-panel-active");

div_adder(); }else {

remover();

}

}

}

```

```

function Signup() {

if(pre_state == 0) { pre_state

= 1; stateone = 0; statetwo

= 1;    statethree = 0;

container.classList.add("rig

ht-panel-active");

div_adder(); }else {

if(statetwo == 0) {

```

```
pre_state = 1;    stateone
= 0;    statetwo = 1;

statethree = 0;

container.classList.add("rig
ht-panel-active");

div_adder();    }else {

remover();

}

}

}
```

```
function about() {

if(pre_state == 0){

pre_state = 1;

stateone = 0; statetwo =

0; statethree = 3;

about_adder();

}else{

if(statethree == 0){

remover();

pre_state = 1;

stateone = 0;

statetwo = 0;

statethree = 3;
```

```
about_adder();

}else{    remover();

    }

}

}
```

```
function invisible(x) {    if(pre_state == 0) {

document.getElementById("l1").style.display = "none";

document.getElementById("l2").style.display = "none";

document.getElementById("l3").style.display = "none";

document.getElementById("l4").style.display = "none";

    }

}
```

```
function visible(x){    if(pre_state == 0) {

document.getElementById("abouts").style.display = "block";

    //space.classList.add("spaceimp");    container.classList.add("right-panel-

active");    document.getElementById("l1").style.display = "none";

document.getElementById("l2").style.display = "none";

document.getElementById("l3").style.display = "none";

document.getElementById("l4").style.display = "none";

}
```

```
}
```

```
}
```

```
function unsignin(x) {  if(pre_state == 0){  
container.classList.remove("right-panel-active");  
  
space.classList.remove("spaceimp");  
  
document.getElementById("logins").style.display = "none";  
document.getElementById("abouts").style.display = "none";  
  
document.getElementById("l1").style.display = "flex";  
document.getElementById("l2").style.display = "flex";  
document.getElementById("l3").style.display = "flex";  
document.getElementById("l4").style.display = "flex";  
  
}
```

```
}
```

```
function signinOne(x){  if(pre_state == 0) {  
container.classList.remove("right-panel-active");  
  
space.classList.add("spaceimp");  
  
document.getElementById("logins").style.display = "block";  
document.getElementById("l1").style.display = "none";  
document.getElementById("l2").style.display = "none";
```

```
document.getElementById("l3").style.display = "none";

document.getElementById("l4").style.display = "none";

}

}
```

```
function signinTwo(x){  if(pre_state == 0) {

document.getElementById("logins").style.display = "block";

space.classList.add("spaceimp");

container.classList.add("right-panel-active");

document.getElementById("l1").style.display = "none";

document.getElementById("l2").style.display = "none";

document.getElementById("l3").style.display = "none";

document.getElementById("l4").style.display = "none";

}

}
```

```
function setcon(x) {  if(pre_state == 0) {

document.getElementById("abouts").style.display = "block";

    //space.classList.add("spaceimp");    container.classList.add("right-panel-
active");    document.getElementById("l1").style.display = "none";

document.getElementById("l2").style.display = "none";

document.getElementById("l3").style.display = "none";

document.getElementById("l4").style.display = "none";

}

}
```



```
}
```

```
}
```

Footer

Index.css

```
@media only screen and (max-width :768px) {
```

```
  .colh {
```

```
    height: auto;
```

```
  }
```

```
  .lists {
```

```
    height: 330px;
```

```
    overflow: auto;    flex-
```

```
    direction: column;
```

```
  }
```

```
}
```

**App.py**

```
from flask import Flask, render_template, url_for, request
```

```
import ibm_db import sendgrid from
```

```
sendgrid.helpers.mail import Mail, Email, To, Content
```

```
SENDGRID_API_KEY =
```

```
"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" # sendgrid conn
```

```
= ibm_db.connect(
```

```
    "DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-
```

```
    7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT"
```

```

"=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=wqq31800;PWD=tOIYo6K1IKAc0XhU",

", ")

print(conn) app = Flask(__name__) app.secret_key =

"\xfd{H\xe5<\x95\xf9\xe3\x96.5\xd1\x01O<!\xd5\xa2\xa0\x9fR"

# sendgrid def

send_mail(email):

    sg = sendgrid.SendGridAPIClient(SENDGRID_API_KEY)

    from_email = Email("xxxxxxxxxxxxxxxxx@gmail.com") # Change to your verified sender

to_email = To(email) # Change to your recipient    subject = "Nutrition is a basic human

need and a prerequisite for healthy life"    content = Content("text/plain",

        "Thank you for creating an account on our platform. Now you can utilise our platform "

        "to maintain a healthier life.")    mail =

Mail(from_email, to_email, subject, content)    # Get a

JSON-ready representation of the Mail object

mail_json = mail.get()

    # Send an HTTP POST request to /mail/send    response =

sg.client.mail.send.post(request_body=mail_json)

print(response.status_code)    print(response.headers)

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

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

def homepage():

    if request.method == 'POST' and 'email' in request.form and 'pass' in request.form:

```

```
return render_template('index.html', error="Wrong Password!")
```

```
return render_template('index.html')
```

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

```
register():
```

```
    if request.method == 'POST' and 'name' in request.form and 'email' in request.form and 'pass' in request.form:
```

```
        name = request.form['name']
```

```
email_up = request.form['email']
```

```
pass_up = request.form['pass']    if
```

```
name == "":
```

```
    error = 'Enter a valid Name.'    return
```

```
render_template('index.html', error=error)    if
```

```
email_up == "":
```

```
    error = 'Enter a valid E-mail.'    return
```

```
render_template('index.html', error=error)    if pass_up
```

```
== "":
```

```
    error = 'Enter a valid Password.'    return
```

```
render_template('index.html', error=error)    sql =
```

```
"SELECT * FROM USER WHERE email =?"    stmt =
```

```
ibm_db.prepare(conn, sql)
```

```
ibm_db.bind_param(stmt, 1, email_up)
```

```
ibm_db.execute(stmt)    account =
```

```
ibm_db.fetch_assoc(stmt)
```

```

    if account:

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

    else:

try:

        insert_sql = "INSERT INTO USER VALUES ,?,?)"

prep_stmt = ibm_db.prepare(conn, insert_sql)

ibm_db.bind_param(prepare_stmt, 1, name)

ibm_db.bind_param(prepare_stmt, 2, email_up)

ibm_db.bind_param(prepare_stmt, 3, pass_up)

ibm_db.execute(prepare_stmt)          send_mail(email_up)

return render_template('index.html', error="Successfully
created")          except ibm_db.stmt_error:

print(ibm_db.stmt_error())          return

render_template('index.html', error="Failed to create Account")

return render_template('index.html') if __name__ == '__main__':

    app.debug = True    app.run() prediction from flask

import Flask, render_template, url_for, request import

ibm_db import sendgrid from sendgrid.helpers.mail

import Mail, Email, To, Content

SENDGRID_API_KEY =
    "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" # sendgrid conn

= ibm_db.connect(

    "DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-
7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT"

```

```

"=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=wqq31800;PWD=tOIYo6K1IKAc0XhU",

", ")

print(conn) app = Flask(__name__) app.secret_key =

"\xfd{H\xe5<\x95\xf9\xe3\x96.5\xd1\x01O<!\xd5\xa2\xa0\x9fR"

# sendgrid def

send_mail(email):

    sg = sendgrid.SendGridAPIClient(SENDGRID_API_KEY)    from_email =

Email("xxxxxxxxxxxxxxxxxx@gmail.com") # Change to your verified sender    to_email =

To(email) # Change to your recipient    subject = "Nutrition is a basic human need and a

prerequisite for healthy life"    content = Content("text/plain",

        "Thank you for creating an account on our platform. Now you can utilise our platform "

        "to maintain a healthier life.")    mail =

Mail(from_email, to_email, subject, content)


# Get a JSON-ready representation of the Mail object

mail_json = mail.get()

# Send an HTTP POST request to /mail/send    response =

sg.client.mail.send.post(request_body=mail_json)

print(response.status_code)    print(response.headers)

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

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

def homepage():

```

```

if request.method == 'POST' and 'email' in request.form and 'pass' in request.form:

    return render_template('index.html', error="Wrong Password!")

return render_template('index.html') @app.route('/register',

methods=['GET', 'POST']) def register():

    if request.method == 'POST' and 'name' in request.form and 'email' in request.form and 'pass' in
request.form:

        name = request.form['name']

email_up = request.form['email']

pass_up = request.form['pass']    if

name == "":

    error = 'Enter a valid Name.'        return

render_template('index.html', error=error)    if

email_up == "":

    error = 'Enter a valid E-mail.'        return

render_template('index.html', error=error)    if pass_up

== "":

    error = 'Enter a valid Password.'        return

render_template('index.html', error=error)    sql =

"SELECT * FROM USER WHERE email =?"    stmt =

ibm_db.prepare(conn, sql)

ibm_db.bind_param(stmt, 1, email_up)

ibm_db.execute(stmt)    account =

ibm_db.fetch_assoc(stmt)

```

```

    if account:

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

    else:

try:

        insert_sql = "INSERT INTO USER VALUES (?,?)"          prep_stmt =

ibm_db.prepare(conn, insert_sql)          ibm_db.bind_param(prepare_stmt, 1,

name)          ibm_db.bind_param(prepare_stmt, 2, email_up)

ibm_db.bind_param(prepare_stmt, 3, pass_up)

ibm_db.execute(prepare_stmt)          send_mail(email_up)          return

render_template('index.html', error="Successfully created")          except

ibm_db.stmt_error:          print(ibm_db.stmt_error())          return

render_template('index.html', error="Failed to create Account")          return

render_template('index.html') if __name__ == '__main__':

    app.debug = True

app.run()

```

## Upload.html

```

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Upload Face with ID</title>

</head>

<body>

```

```
<div class="container">
```

```
<div class="row">
```

```
<div class="col">
```

```
<h1>Upload Face (filename = face's name (i.e. John_Smith.jpg)</h1>
```

```
<hr>
```

```
<form action="/upload-image" method="POST" enctype="multipart/form-data">
```

```
<div class="form-group">
```

```
<label>Select image</label>
```

```
<div class="custom-file">
```

```
<input type="file" class="custom-file-input" name="image"
```

```
id="image">
```

```
<label class="custom-file-label" for="image">Select image...</label>
```

```
</div>
```

```
</div>
```

```
<button type="submit" class="btn btn-primary">Upload</button>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</div>
```

```

```



```
</body>
```

```
</html>
```

```
Upload.py
```

```
import os
```

```
from flask import Flask, redirect, jsonify, request, url_for, render_template, flash
```

```
app = Flask(__name__)
```

```
app.config["IMAGE_UPLOADS"] = "C:/Flask/Upload/"
```

```
@app.route("/")
```

```
def home():
```

```
    return render_template("index.html")
```

```
# Route to upload image
```

```
@app.route('/upload-image', methods=['GET',  
'POST']) def upload_image(): if request.method ==
```

```
"POST": if request.files:
```

```
    image = request.files["image"]
```

```
        # print(image + "Uploaded to Faces")
```

```
        # flash('Image successfully Uploaded to Faces.')
```

```
image.save(os.path.join(app.config["IMAGE_UPLOADS"], image.filename)) filename  
= os.path.join(app.config["IMAGE_UPLOADS"], image.filename) print("stored as:" +  
filename)
```

```
return render_template("upload_image.html",
```

```
uploaded_image=filename)
```

```
return render_template("upload_image.html") if __name__ ==
```

```
"__main__":
```

```
app.run()
```

output:



