

# IMPLEMENTING WEB APPLICATION

## CREATE IBM DB2 AND CONNECT WITH PYTHON

Date	3 <sup>rd</sup> November, 2022
Team ID	PNT2022TMID27090
Project Name	Plasma Donor Application

The screenshot shows the IBM Cloud console interface for a Db2-cs resource. The top navigation bar includes the IBM Cloud logo, a search bar, and various icons. The main content area is divided into a left sidebar with 'Manage' options (Getting started, Service credentials, Connections) and a main panel. The 'Getting started' section provides instructions on finding credentials and includes buttons for 'Go to UI', 'Getting started docs', and 'Support case'. The 'Need help?' section includes a link to submit a support case.

The screenshot shows the IBM Db2 on Cloud console interface. The top navigation bar includes the IBM Db2 on Cloud logo and various icons. The main content area is divided into a left sidebar with 'Tables' and 'Views' options and a main panel. The 'Tables' section displays a table named 'NYB16417.LOGIN'. The table has four columns: USERNAME, USERMAIL, USERCONTACT, and PASSWORD. The data is displayed in a table format with a 'Back' button and an 'Export to CSV' button.

USERNAME	USERMAIL	USERCONTACT	PASSWORD
NandhiniL	nandhini02lak@gmail.com	1234	nandhini
NandhiniLakshmi	nandhini02lak@gmail.com	1234	dass
NandhiniLak	nandhini02lak@gmail.com	12345678	nan
Sharath	nandhiniLakshmi1307@gmail.com	7334567898	Nandhy
Sharath Kumar	nandhini02lak@gmail.com	7339286572	123456
Sharathtest	chinnukool72@gmail.com	1234	sharath

IBM Db2 on Cloud									
Load Data Load History Tables Views Indexes Aliases MQTs Sequences Application objects									
NYB16417.REQUEST2									Back
									Export to CSV
DRMAIL	HOSPITALNAME	RECNAME	RECMOBILE	RECMAIL	RECAGE	RECGENDER	RECBLOODGROUP	RECAREA	RECCITY
chinnukool72@gmail.com	kmc	nandhini	123345	chinnukool72@gmail.com	21	Male	A+	chennai	chennai
nandhini@gmail.com	kmc	nandhini	123456	nandhini@gmail.com	21	Female	A-	chennai	chennai

IBM Db2 on Cloud									
Load Data Load History Tables Views Indexes Aliases MQTs Sequences Application objects									
NYB16417.DONOR2									Back
									Export to CSV
NAME	MOBILE	EMAIL	AGE	GENDER	BLOOD	AREA	CITY	DISTRICT	
Nandhinilak	1234	nandhini02lak@gmail.com	21	Female	O+	chennai	chennai	tn	
Sharath	123456	chinnukool72@gmail.com	21	Male	B+	chennai	chennai	chennai	

```

1 from flask import Flask, render_template, request, redirect, url_for
2 import ibm_db
3 import sendgrid
4 import os
5 import re
6 from sendgrid.helpers.mail import Mail
7
8 app = Flask(__name__)
9 app.secret_key = "1"
10
11 conn = ibm_db.connect("DATA")
12
13 @app.route("/")
14 def index():
15     return render_template("index.html")
16
17 @app.route("/home")
18 def home_page():
19     return render_template("home.html")
20
21 #-----
22 @app.route("/login", methods=['GET', 'POST'])
23 def login():
24     global userid
25     msg = ""
26     if request.method == 'P':
27         username = request.form.get('username')
28         password = request.form.get('password')
29         sql = "SELECT * FROM user WHERE username = '%s' AND password = '%s'" % (username, password)
30         stmt = ibm_db.prepare(conn, sql)
31         ibm_db.execute(stmt)
32         ibm_db.bind_param(stmt, 1, username)
33         ibm_db.execute(stmt)
34         account = ibm_db.fetch_row()
35         print(account)
36         if account:
37             session['logged_in'] = True
38             session['id'] = account[0]
39             session['username'] = account[1]
40             session['password'] = account[2]
41             msg = 'Logged in successfully'
42         else:
43             msg = 'Incorrect username or password'
44     return render_template("login.html", msg=msg)
45
46 return render_template("login.html", msg=msg)
47
48 #-----
49 # After login
50 @app.route("/afterlogin")
51 def afterlogin():
52     return render_template("afterlogin.html")
53
54 #-----
55 @app.route("/signin", methods=['GET', 'POST'])
56 def signin():
57     msg = ""
58     if request.method == 'P':
59         username = request.form.get('username')
60         password = request.form.get('password')
61         sql = "SELECT * FROM user WHERE username = '%s' AND password = '%s'" % (username, password)
62         stmt = ibm_db.prepare(conn, sql)
63         ibm_db.execute(stmt)
64         account = ibm_db.fetch_row()
65         print(account)
66         if account:
67             session['logged_in'] = True
68             session['id'] = account[0]
69             session['username'] = account[1]
70             session['password'] = account[2]
71             msg = 'Logged in successfully'
72         else:
73             msg = 'Incorrect username or password'
74     return render_template("signin.html", msg=msg)
75
76 #-----
77 @app.route("/register", methods=['GET', 'POST'])
78 def register():
79     msg = ""
80     if request.method == 'P':
81         username = request.form.get('username')
82         password = request.form.get('password')
83         sql = "SELECT * FROM user WHERE username = '%s'" % (username)
84         stmt = ibm_db.prepare(conn, sql)
85         ibm_db.execute(stmt)
86         account = ibm_db.fetch_row()
87         print(account)
88         if account:
89             msg = 'Username already exists'
90         else:
91             sql = "INSERT INTO user (username, password) VALUES ('%s', '%s')" % (username, password)
92             stmt = ibm_db.prepare(conn, sql)
93             ibm_db.execute(stmt)
94             msg = 'Registration successful'
95     return render_template("register.html", msg=msg)
96
97 #-----
98 @app.route("/addonor", methods=['GET', 'POST'])
99 def addonor():
100     msg = ""
101     if request.method == 'P':
102         name = request.form.get('name')
103         mobile = request.form.get('mobile')
104         email = request.form.get('email')
105         age = request.form.get('age')
106         gender = request.form.get('gender')
107         blood = request.form.get('blood')
108         area = request.form.get('area')
109         city = request.form.get('city')
110         district = request.form.get('district')
111         sql = "SELECT * FROM donor WHERE name = '%s' AND mobile = '%s'" % (name, mobile)
112         stmt = ibm_db.prepare(conn, sql)
113         ibm_db.execute(stmt)
114         account = ibm_db.fetch_row()
115         print(account)
116         if account:
117             msg = 'Donor already exists'
118         else:
119             sql = "INSERT INTO donor (name, mobile, email, age, gender, blood, area, city, district) VALUES ('%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s')" % (name, mobile, email, age, gender, blood, area, city, district)
120             stmt = ibm_db.prepare(conn, sql)
121             ibm_db.execute(stmt)
122             msg = 'Donor added successfully'
123     return render_template("addonor.html", msg=msg)
124
125 #-----
126 @app.route("/request_page", methods=['GET', 'POST'])
127 def request_page():
128     msg = ""
129     if request.method == 'P':
130         drmail = request.form.get('drmail')
131         hospitalname = request.form.get('hospitalname')
132         recname = request.form.get('recname')
133         recmobile = request.form.get('recmobile')
134         recmail = request.form.get('recmail')
135         recage = request.form.get('recage')
136         recgender = request.form.get('recgender')
137         recbloodgroup = request.form.get('recbloodgroup')
138         recarea = request.form.get('recarea')
139         reccity = request.form.get('reccity')
140         recdistrict = request.form.get('recdistrict')
141         sql = "SELECT * FROM request WHERE drmail = '%s' AND hospitalname = '%s'" % (drmail, hospitalname)
142         stmt = ibm_db.prepare(conn, sql)
143         ibm_db.execute(stmt)
144         account = ibm_db.fetch_row()
145         print(account)
146         if account:
147             msg = 'Request already exists'
148         else:
149             sql = "INSERT INTO request (drmail, hospitalname, recname, recmobile, recmail, recage, recgender, recbloodgroup, recarea, reccity, recdistrict) VALUES ('%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s')" % (drmail, hospitalname, recname, recmobile, recmail, recage, recgender, recbloodgroup, recarea, reccity, recdistrict)
150             stmt = ibm_db.prepare(conn, sql)
151             ibm_db.execute(stmt)
152             msg = 'Request added successfully'
153     return render_template("request_page.html", msg=msg)
154
155 #-----
156 @app.route("/logout")
157 def logout():
158     session.clear()
159     return redirect(url_for("home_page"))
160
151 if __name__ == '__main__':
152     app.run(debug=True)

```

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <!-- Required meta tags -->
5 <meta charset="utf-8">
6 <meta http-equiv="X-UA-Compatible" content="IE=edge">
7 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
8 <!-- font awesome -->
9 <script src="https://kit.fontawesome.com/15af226b72.js" crossorigin="anonymous"></script>
10
11 <!-- Bootstrap CSS -->
12 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css">
13
14 <!-- Google Font -->
15 <style>
16 @import url('https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100..700;1,100..700&display=block');
17 </style>
18 <!-- contains style for all pages -->
19 <link rel="stylesheet" href="/static/style.css">
20 <script src="https://kit.fontawesome.com/000fb23390.js" crossorigin="anonymous"></script>
21
22 <!-- block link -->
23 <div class="block-link">
24 </div>
25 </head>
26 <body>
27
28 <!-- Header -->
29 <header>
30 <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
31 <div class="container">
32 <a href="{{ url_for('index') }}" class="navbar-brand"><div class="collapse navbar-collapse">
33 <a href="{{ url_for('home') }}" class="nav-link">Home</a>
34 <a href="{{ url_for('signin') }}" class="nav-link">Register</a>
35 <a href="{{ url_for('login') }}" class="nav-link">Login</a>
36 </div>
37 </div>
38 </nav>
39
40 <!-- Future contents -->
41 <div class="future-content">
42 <div class="block-content">
43 </div>
44 </div>
45
46 <!-- Optional JavaScript -->
47 <!-- jQuery first, then Popper.js, then Bootstrap JS -->
48 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+9654zq87zP8tBRNNtAZUhtTd" crossorigin="anonymous"></script>
49 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js" integrity="sha384-Uo2eK++S6pMgUxW9pALhMDDX+tZEeNMDg+JQ8v6V80gY" crossorigin="anonymous"></script>
50 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js" integrity="sha384-gdbQ4e6KNBJtz9WttlC79Hw4gn2QmV2hsDlAlXoJ18t13xXrSMW96SQ0Nw3m/48"></script>
51
52 <!-- Watson Assistant Chat Options -->
53 <script>
54 window.watsonAssistantChatOptions = {
55   integrationID: "18cd3fb-fde-4fb9-b9d2-a7c1895e980b", // The ID of this region: "au-syd", // The region your integration is hosted in.
56   serviceInstanceID: "60ba66a6-d37f-412f-b60f-5a5d73de97", // The ID of your service instance
57   onLoad: function(instance) { instance.render(); }
58 };
59
60 <script>
61 const t=document.createElement('script');
62 t.src="https://web-chat.global/assistant/watson.appdomain.cloud/versions/1.8.0/watson-chat-1.8.0.js";
63 document.head.appendChild(t);
64 </script>
65
66 </body>
67 </html>

```







```
File Edit Selection View Go Run Terminal Help
sprint2

EXPLORER
template
  base.html
  donor.html
  home.html
  login.html
  register.html
  request.html
  success.html
  user_profile.html
  app.py

user_profile.html
1 1 {% extends 'base.html' %}
2
3 <!-- title tag -->
4 <!-- block title % -->
5 <title>Plasma-Profile</title>
6 <style>
7   .profile-area{
8     padding:70px 0;
9     border: 2px solid #232323;
10    margin-right: 190px;
11    margin-left: 190px;
12    margin-top: 50px;
13  }
14  .card{
15    box-shadow: 0 0 30px #232323;
16    overflow: hidden;
17    border-radius: 15px;
18    margin-top: 30px;
19    background-image: linear-gradient(to top right, #f5f5f5, #f5f5f5);
20    height: 200px;
21  }
22  .card: hover{
23    border-color: #000000;
24    transform: rotate(1deg);
25    transition: 7s;
26  }
27  .main-text, .card-body{
28    text-align: center;
29    padding-top: 60px;
30  }
31  .btn{
32    padding: 10px 10px;
33    margin-bottom: 20px;
34  }
35 </style>
36 <!-- endblock % -->
37 <!-- block link % -->
38
39 <!-- endblock % -->
40
41 <!-- endblock % -->
42 <div class="page-header text-center" style="margin-top: 30px; color: black">Your P
43
44
45 <!-- User Content -->
46 <!-- block content % -->
47 <div class="container">
48   <div class="text-danger" style="margin-top: 50px;">Welcome : {{session["username"]}}
49 </div>
50 <div class="profile-area">
51   <div class="container">
52     <div class="msg">{{ msg }}</div>
53   <div class="row">
54
55     <div class="col-12 col-md-6 col-lg-6">
56       <div class="card">
57         <div class="main-text card-body">
58           <div><a href="{{url_for('register')}}" class="btn btn-light">Donate P
59         </div>
60       </div>
61     </div>
62     <div class="col-12 col-md-6 col-lg-6">
63       <div class="card">
64         <div class="main-text card-body">
65           <div><a href="{{url_for('donorlist')}}" class="btn btn-light">Request
66         </div>
67       </div>
68     </div>
69   </div>
70 </div>
71 </div>
72 <div class="container">
73   <a href="{{url_for('logout')}}" class="btn btn-danger">Log Out</a></div>
74
75
76 <!-- endblock % -->
77
```

```
File Edit Selection View Go Run Terminal Help
sprint2

EXPLORER
template
  base.html
  donor.html
  home.html
  login.html
  register.html
  request.html
  success.html
  user_profile.html
  app.py

login.html
1 1 {% extends 'base.html' %}
2
3 <!-- title tag -->
4 <!-- block title % -->
5 <title>Plasma-Login</title>
6 <!-- endblock % -->
7
8 <!-- login Content -->
9 <!-- block content % -->
10 <!-- login form -->
11 <div class="container">
12   <div class="text-center mt-5"><h3>Login using UserName and Password</h3></div>
13 </div>
14
15
16 <div class="container mt-5">
17   <div class="row justify-content-center">
18     <div class="col-sm-6">
19       <div class="card">
20         <div class="card-body">
21           <!-- form content -->
22           <form action="/login" method="POST">
23             <div class="msg">{{ msg }}</div>
24
25             <div class="form-group">
26               <label for="username">User Name</label>
27               <input type="text" class="form-control" name="username">
28             </div>
29             <div class="form-group">
30               <label for="password">Password</label>
31               <input type="password" class="form-control" name="password">
32             </div>
33
34             <div>
35               <a href="{{url_for('forgetpassword')}}">Forget Password
36             </div>
37             <div>
38               <div class="form-group text-center">
39                 <input type="submit" value="Login" class="btn btn-success">
40               </div>
41               <div style="text-align: center;">
42                 <p>Don't have an account <a href="{{url_for('signin')}}">Signin
43               </div>
44             </div>
45           </form>
46         </div>
47       </div>
48     </div>
49   </div>
50 </div>
51 <!-- endblock % -->
52
53
54
```