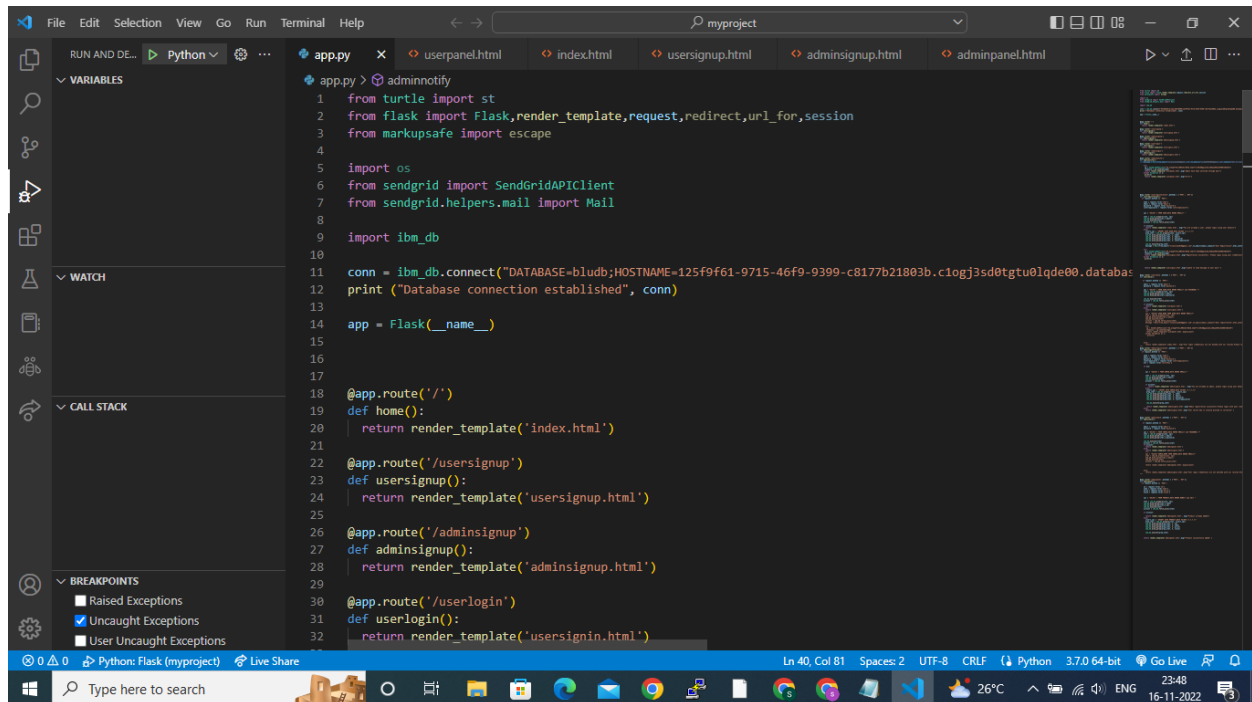


Delivery of sprint 3

| | |
|--------------|---|
| DATE | 3 November 2022 |
| TEAM ID | PNT2022TMID26994 |
| PROJECT NAME | Inventory Management System for Retailers |



The screenshot shows a Python IDE with a dark theme. The main editor displays the code for `app.py`, which is a Flask application. The code includes imports for `turtle`, `Flask`, `render_template`, `request`, `redirect`, `url_for`, `session`, `os`, `SendGridAPIClient`, `Mail`, and `ibm_db`. It also shows a database connection setup, a Flask app instance, and several routes: `home()`, `usersignup()`, `adminsingup()`, and `userlogin()`. The left sidebar contains panels for `VARIABLES`, `WATCH`, `CALL STACK`, and `BREAKPOINTS`. The bottom status bar indicates the file is `app.py` in the `myproject` directory, with a Python 3.7.0 64-bit interpreter.

```
1 from turtle import st
2 from flask import Flask,render_template,request,redirect,url_for,session
3 from markupsafe import escape
4
5 import os
6 from sendgrid import SendGridAPIClient
7 from sendgrid.helpers.mail import Mail
8
9 import ibm_db
10
11 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases
12 print ("Database connection established", conn)
13
14 app = Flask(__name__)
15
16
17
18 @app.route('/')
19 def home():
20     return render_template('index.html')
21
22 @app.route('/usersignup')
23 def usersignup():
24     return render_template('usersignup.html')
25
26 @app.route('/adminsingup')
27 def adminsingup():
28     return render_template('adminsingup.html')
29
30 @app.route('/userlogin')
31 def userlogin():
32     return render_template('usersignin.html')
```

The screenshot shows a VS Code editor window with a Python project named 'myproject'. The editor displays the 'app.py' file, which contains the following code:

```
45     return render_template('userpanel.html',msg="Admin have been notified through mail")
46 except Exception as e:
47     print(e)
48     return render_template('userpanel.html',msg="Error")
49
50
51
52
53
54
55
56
57
58 @app.route('/userregistration',methods = ['POST', 'GET'])
59 def userregistration():
60     if request.method == 'POST':
61
62         name = request.form['name']
63         email = request.form['email']
64         password = request.form['password']
65         confirmpassword = request.form['confirmpassword']
66
67
68         sql = "SELECT * FROM USER_DATA WHERE EMAIL=? "
69
70         stmt = ibm_db.prepare(conn, sql)
71         ibm_db.bind_param(stmt,1,email)
72         ibm_db.execute(stmt)
73         account = ibm_db.fetch_assoc(stmt)
74
75         if account:
76             return render_template('index.html', msg="You are already a user, please login using your details")
```

The left sidebar shows the 'VARIABLES' panel, which is currently empty. The 'WATCH' panel is also empty. The 'CALL STACK' panel is empty. The 'BREAKPOINTS' panel shows 'Uncaught Exceptions' checked. The status bar at the bottom indicates the file is at line 40, column 81, with a UTF-8 encoding and CRLF line endings. The system tray shows the date and time as 23:48 on 16-11-2022.

The screenshot shows a VS Code editor window with a Python project named 'myproject'. The editor displays the 'app.py' file, which contains the following code:

```
84
85     ibm_db.execute(prepare_stmt)
86     message = Mail(from_email="sivasurya1010@gmail.com",to_emails=email,subject="User Registration",html_content=html_content)
87
88
89     sg = SendGridAPIClient("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")
90     response = sg.send(message)
91     return render_template('usersignin.html',msg="Registration successfull. Please login using your credentials")
92 except Exception as e:
93     print(e)
94
95
96
97
98     return render_template('usersignin.html',msg="unable to send message to your mail")
99
100
101
102 @app.route('/usercheck',methods = ['POST', 'GET'])
103 def usercheck():
104
105     if request.method == 'POST':
106
107         email = request.form['email']
108         password = request.form['password']
109
110         sql = "SELECT * FROM USER_DATA WHERE EMAIL=? and PASSWORD=? "
111         stmt = ibm_db.prepare(conn, sql)
112         ibm_db.bind_param(stmt,1,email)
113         ibm_db.bind_param(stmt,2,password)
114
115         ibm_db.execute(stmt)
```

The left sidebar shows the 'VARIABLES' panel, which is currently empty. The 'WATCH' panel is also empty. The 'CALL STACK' panel is empty. The 'BREAKPOINTS' panel shows 'Uncaught Exceptions' checked. The status bar at the bottom indicates the file is at line 40, column 81, with a UTF-8 encoding and CRLF line endings. The system tray shows the date and time as 23:48 on 16-11-2022.

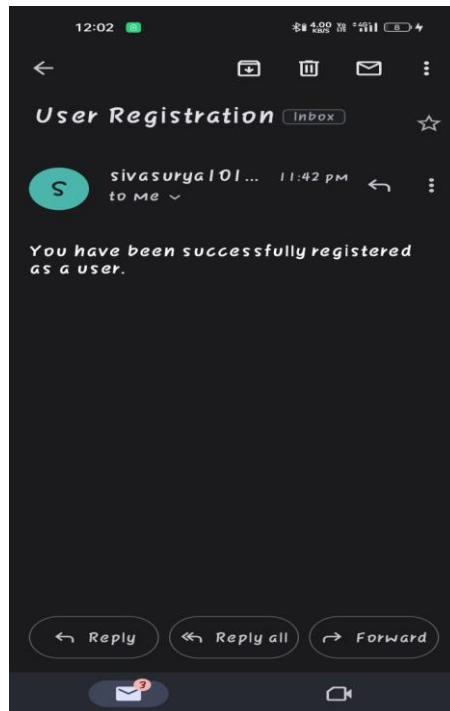
This screenshot shows the Visual Studio Code editor with a Python project named 'myproject'. The file explorer on the left shows a directory structure with files like 'app.py', 'userpanel.html', 'index.html', 'usersignup.html', 'adminsingup.html', and 'adminpanel.html'. The 'app.py' file is open, showing a function 'adminregistration()' that handles POST requests. The code includes database operations using 'ibm_db' to check for existing users and insert new ones into the 'ADMIN_DATA' table. The status bar at the bottom indicates the current position is at line 40, column 81, with a UTF-8 encoding and CRLF line endings.

```
138 else:
139     return render_template('index.html', msg="Your login credentials are not mached with our records.Please l
140
141
142 @app.route('/adminregistration',methods = ['POST', 'GET'])
143 def adminregistration():
144     if request.method == 'POST':
145
146         name = request.form['name']
147         email = request.form['email']
148         password = request.form['password']
149         confirmpassword = request.form['confirmpassword']
150         ece = request.form['secretkey']
151
152     if ece:
153
154
155         sql = "SELECT * FROM ADMIN_DATA WHERE EMAIL=? "
156
157         stmt = ibm_db.prepare(conn, sql)
158         ibm_db.bind_param(stmt,1,email)
159         ibm_db.execute(stmt)
160         account = ibm_db.fetch_assoc(stmt)
161
162     if account:
163         return render_template('adminsigin.html', msg="You are already an admin, please login using your deta
164     else:
165         insert_sql = "INSERT INTO ADMIN_DATA VALUES (?,?,?,?)"
166         prep_stmt = ibm_db.prepare(conn, insert_sql)
167         ibm_db.bind_param(prepare_stmt, 1, name)
168         ibm_db.bind_param(prepare_stmt, 2, email)
169         ibm_db.bind_param(prepare_stmt, 3, password)
```

This screenshot shows the Visual Studio Code editor with the same 'myproject' project. The 'app.py' file is open, showing a function 'adminnotify()' that handles POST requests for adding products. The code includes database operations using 'ibm_db' to check for existing products and insert new ones into the 'PRODUCT_DATA' table. The status bar at the bottom indicates the current position is at line 40, column 81, with a UTF-8 encoding and CRLF line endings.

```
220 price = request.form['price']
221 stock = request.form['stock']
222
223
224
225
226 sql = "SELECT * FROM PRODUCT_DATA WHERE NAME=? and ID=? "
227
228 stmt = ibm_db.prepare(conn, sql)
229 ibm_db.bind_param(stmt,1,name)
230 ibm_db.bind_param(stmt,2,id)
231 ibm_db.execute(stmt)
232 account = ibm_db.fetch_assoc(stmt)
233
234 if account:
235     return render_template('adminpanel.html', msg="Product already added")
236 else:
237     insert_sql = "INSERT INTO PRODUCT_DATA VALUES (?,?,?,?)"
238     prep_stmt = ibm_db.prepare(conn, insert_sql)
239     ibm_db.bind_param(prepare_stmt, 1, id)
240     ibm_db.bind_param(prepare_stmt, 2, name)
241     ibm_db.bind_param(prepare_stmt, 3, price)
242     ibm_db.bind_param(prepare_stmt, 4, stock)
243
244     ibm_db.execute(prepare_stmt)
245
246 return render_template('adminpanel.html',msg="Product successfully added" )
247
248
249
250
251
```

Sendgrid verification(user):



Sendgrid verification(Admin):

