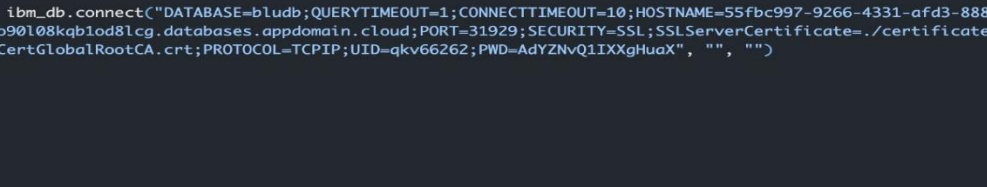


## Flask Code



```
1 import ibm_db
2
3 db = ibm_db.connect("DATABASE=bludb;QUERYTIMEOUT=1;CONNECTTIMEOUT=10;HOSTNAME=55fbc997-9266-4331-afd3-888b05e734c0.
bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31929;SECURITY=SSL;SSLServerCertificate=./certificates/
DigiCertGlobalRootCA.crt;PROTOCOL=TCP;UID=qkv66262;PWD=AdYZNvQ1IXXgHuaX", "", "")
```

```

15     })
16     if "password" not in args:
17         raise KeyError({
18             "status": "Error",
19             "message": "password is required field",
20             "data": None
21         })
22
23     def post(self):
24         payload = request.json
25         try:
26             self.__sanitizer(payload)
27             result = ibm_db.exec_immediate(db, "SELECT * FROM USERS WHERE EMAIL='{0}' AND VERIFIED=TRUE".format(payload["email"]))
28             value = ibm_db.fetch_tuple(result)
29             if value == False:
30                 return json.dumps({"status": "Error", "data": None, "message": "Invalid Credentials"})
31             if bcrypt.checkpw(bytes("{}").format(payload["password"]), 'utf-8'), bytes(value[3], 'utf-8')):
32                 payload = {
33                     "email": value[2],
34                     "token": value[5],
35                     "name": value[1],
36                     "currency": value[4]
37                 }
38                 return json.dumps({"status": "Success", "data": payload, "message": "success"})
39             else:
40                 return json.dumps({"status": "Error", "data": None, "message": "Invalid Credentials"})
41         except KeyError as e:
42             return json.dumps(e.args)
43         except:
44             return json.dumps({"status": "Error", "data": None, "message": "Server Error"})
45
46
47
48
49

```