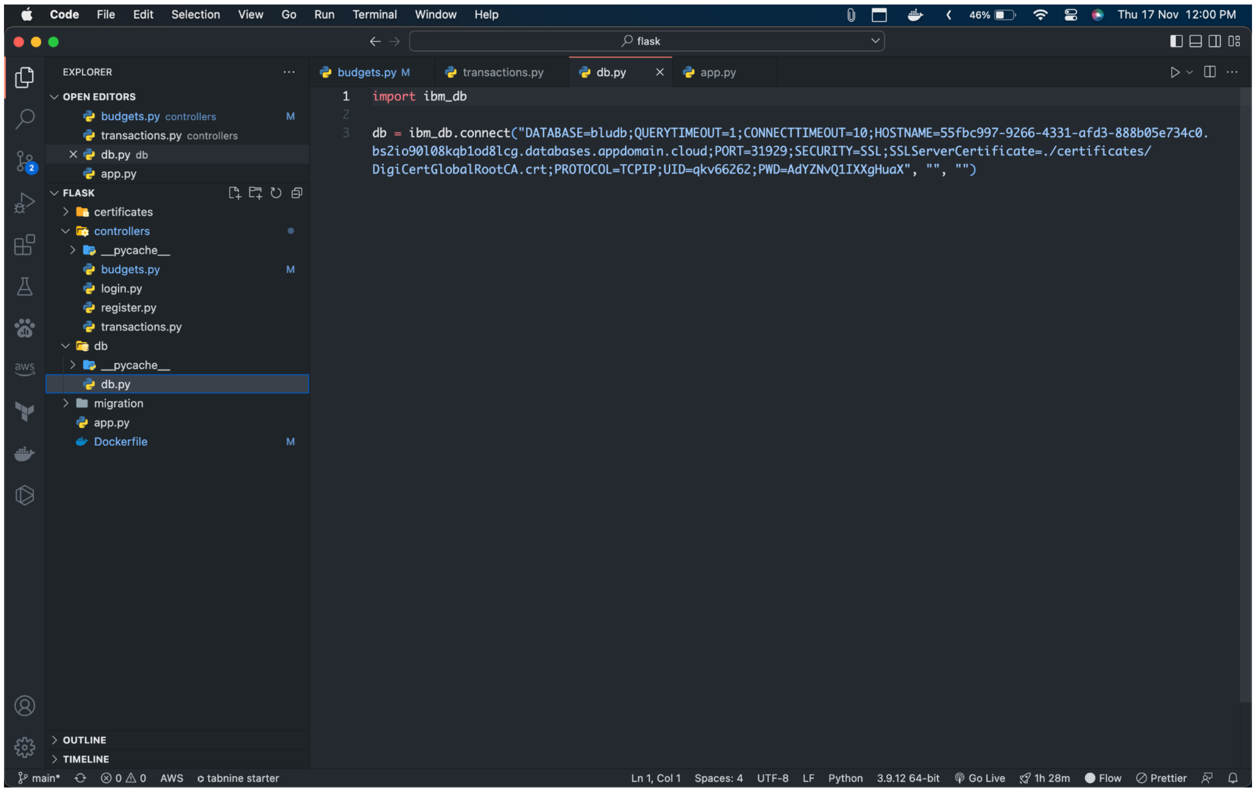
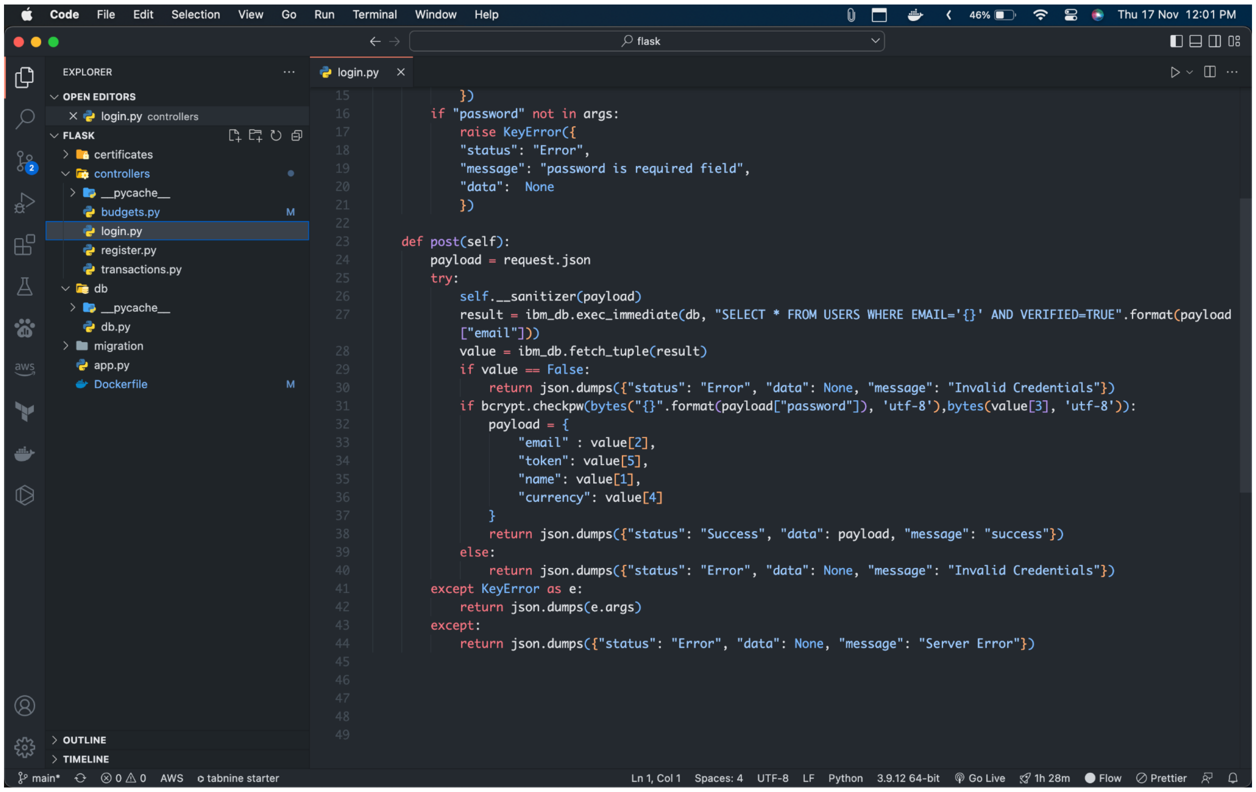
Create IBM DB2 and connect with python

Flask Code



```python
import ibm_db

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=TCPIP;UID=qkv66262;PWD=AdYZNvQ1IXXgHuaX", "", "")
```



```python
            })
        if "password" not in args:
            raise KeyError({
            "status": "Error",
            "message": "password is required field",
            "data":  None
            })

    def post(self):
        payload = request.json
        try:
            self.__sanitizer(payload)
            result = ibm_db.exec_immediate(db, "SELECT * FROM USERS WHERE EMAIL='{}' AND VERIFIED=TRUE".format(payload["email"]))
            value = ibm_db.fetch_tuple(result)
            if value == False:
                return json.dumps({"status": "Error", "data": None, "message": "Invalid Credentials"})
            if bcrypt.checkpw(bytes("{}".format(payload["password"]), 'utf-8'),bytes(value[3], 'utf-8')):
                payload = {
                    "email" : value[2],
                    "token": value[5],
                    "name": value[1],
                    "currency": value[4]
                }
                return json.dumps({"status": "Success", "data": payload, "message": "success"})
            else:
                return json.dumps({"status": "Error", "data": None, "message": "Invalid Credentials"})
        except KeyError as e:
            return json.dumps(e.args)
        except:
            return json.dumps({"status": "Error", "data": None, "message": "Server Error"})
```