

auth.py

```
import jwt
from . import validation_error

def check_auth(request):
    token = request.headers['Authorization']
    if (not token):
        return validation_error.throw_validation("Please login",401)
    decoded = jwt.decode(token,"secret",algorithms=["HS256"])
    return decoded['id']
```

db.py

```
import ibm_db

conn = None

def get_db():
    global conn
    print(conn)
    if conn == None:
        conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0t
gtu0lqde00.databases.appdomain.cloud;PORT=32459;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRoot
CA.crt;UID=yvp21124;PWD=jNDakHSrZNopa2oe","")
    return conn
```

exception.py

```
from flask import json

def handle_exception(e):
    exception_name = type(e).__name__
    exception_str = e.__str__()
    print(exception_name,exception_str.find('803'))
    status_code=500
    response={
        "status":"fail",
        "message":"Something went wrong"
    }
    if exception_name=='KeyError':
        status_code=400
        response["message"] = "Please fill all fields"

    if exception_name=='ValidationException':
        print(e.args)
        status_code=400
        response["message"] = "Please fill all fields"
```

```

if exception_str.find('803') >=0:
    print('Db')
    status_code=400
    response["message"] = "Invalid value provided"

if exception_name == 'DecodeError':
    print("token error")
    status_code=401
    response['message']='Token expired.Please login'

return json.dumps(response), status_code, {'ContentType':'application/json'}

```

sendemail.py

```

import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
from ..lib import exception

```

```

def sendemail():

```

```

    message = Mail(
        from_email='from_email@example.com',
        to_emails='to@example.com',
        subject='ORDER CONFIRMATION',
        html_content='<strong>Your order placed</strong>')
    try:
        sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
        response = sg.send(message)
        print(response.status_code)
        print(response.body)
        print(response.headers)
    except Exception as e:
        return exception.handle_exception(e)

```

validation_error.py

```

from flask import json

```

```

def throw_validation(msg,code):
    return json.dumps({"status":"fail","message":msg}),code,{'ContentType':'application/json'}

```