**Отчет по Лабораторной работе 5**

Панфилов Группа 5

Код python

import psycopg2 import redis import json import random import string import time

def create\_token(length=16): alphabet = string.printable token = "".join([\*map(lambda \_: random.choice(alphabet),range(length))]) return token psql\_data = []

with open("C:/Users/andre/OneDrive/Рабочий стол/Схемы БД/лаба

3/conf2.json", "r") as file: psql\_data = json.load(file)

redis\_host = '169.254.10.60' redis\_port = 6379 redis\_db = 0

connection = psycopg2.connect(\*\*psql\_data['server'][0]) cursor = connection.cursor()

r\_connection = redis.StrictRedis(host=redis\_host, port= redis\_port, username ='client', password ='1234', charset='utf-8', decode\_responses=True,)

def signup(email: str, password: str):

cursor.execute('SELECT \* from personal\_data where email=%s',

(email,)) if cursor.fetchone():

raise Exception('Email is already registered') cursor.execute('insert into personal\_data (email, password) values (%s, %s)', (email, password)) connection.commit()

def signin(email: str, password: str): cursor.execute("select \* from personal\_data where email=%s and password= %s", (email, password))

auth\_user = cursor.fetchone() if not auth\_user: raise Exception('Invalid Email or Password')

token = create\_token() timestamp = time.time() r\_connection.hmset(token, {'email': email, 'timestamp':

timestamp}) return token

def verify\_token(token)-> bool: return r\_connection.exists(token)

def signout(token: str): if not r\_connection.exists(token): raise Exception('invalid token') r\_connection.delete(token)

def add\_account\_data(token: str, \*\*kwargs) -> None: if not r\_connection.exists(token): raise Exception('invalid token') r\_connection.hmset(token, kwargs)

def get\_account\_data(token: str) -> str: if not r\_connection.exists(token): raise Exception('invalid token') data = r\_connection.hgetall(token) data.pop('password', None) return json.dumps(data)

#signup('alel', '1234') a = signin('alal', '1234') print(verify\_token(a)) add\_account\_data(a, name = 'Alexey', sex = 'male', age = 15) print(get\_account\_data(a))

SQL запрос

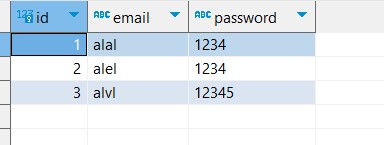
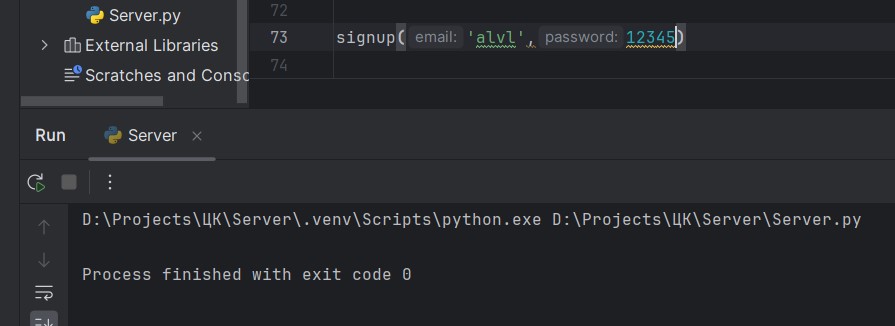
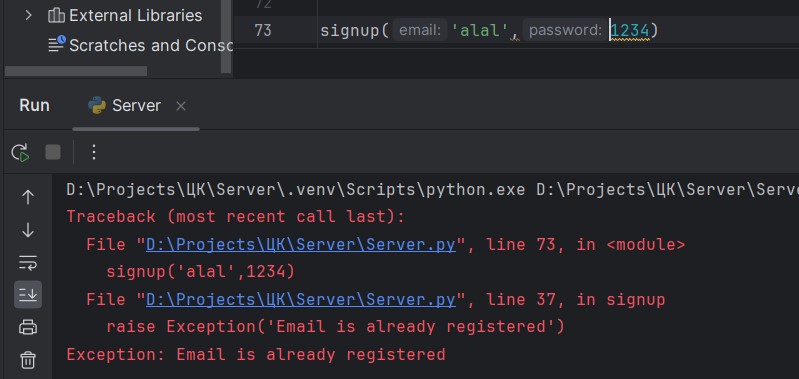
create table if not exists personal\_data( id serial primary key, email text, password text

);

Примеры выполнения функций

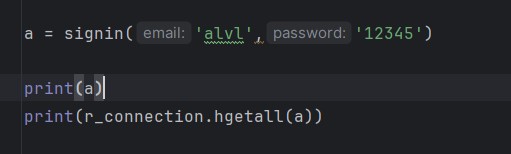
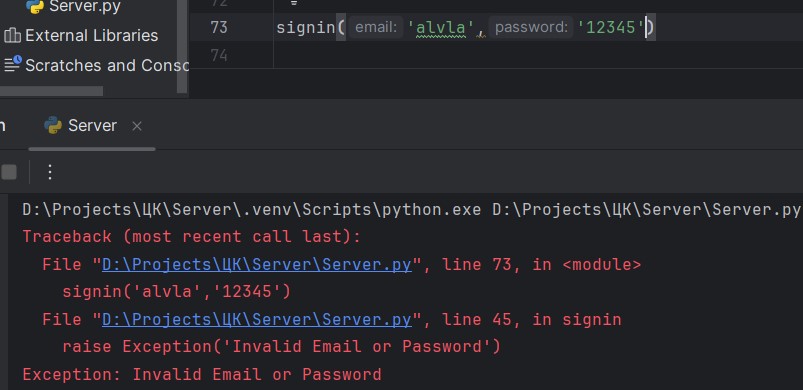
Signup function

Exception ввел существующий email

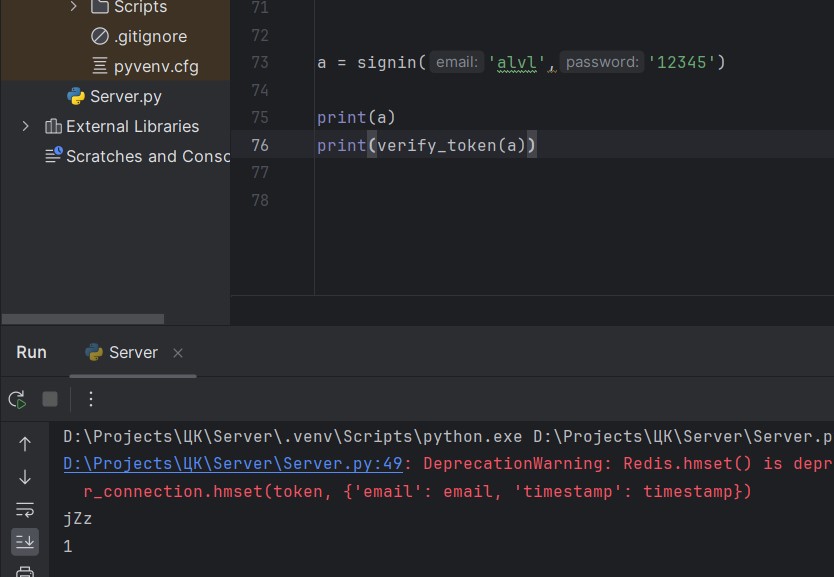


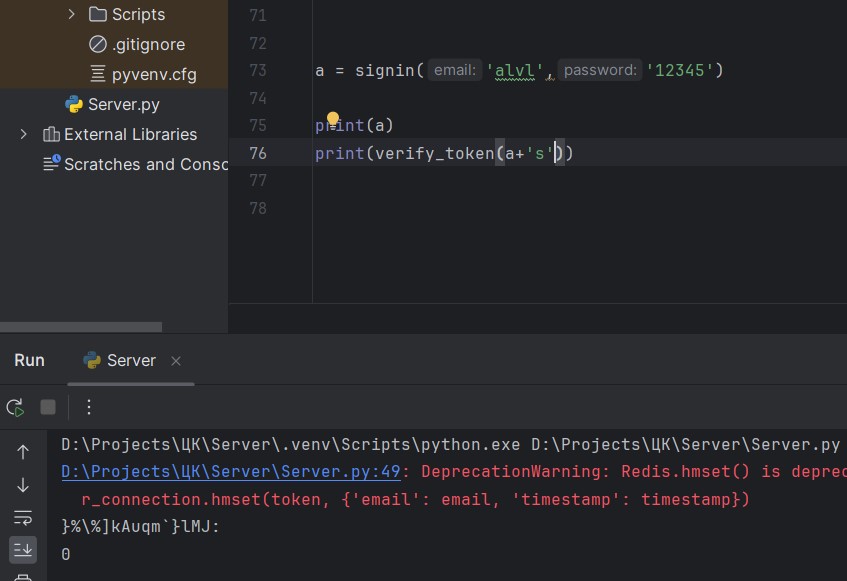
Signin function

Exception ввел несуществующий email

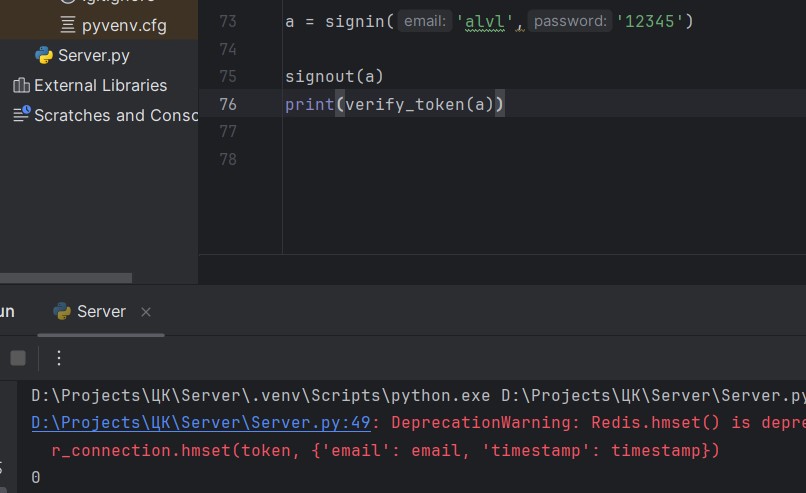


Verify token

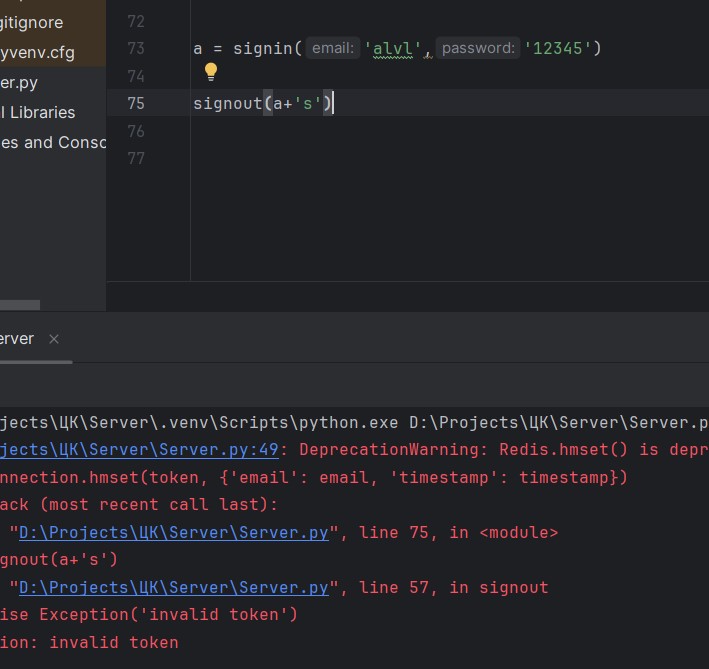




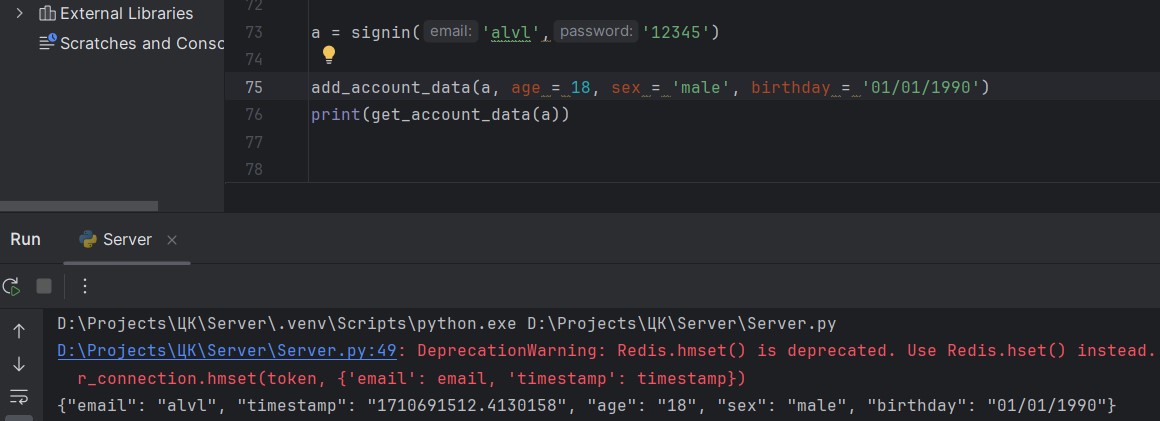
Signout



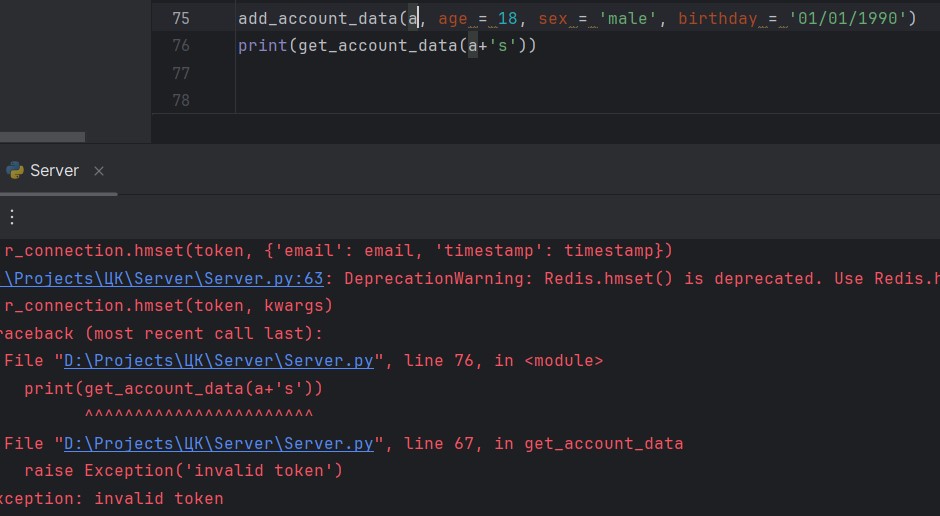
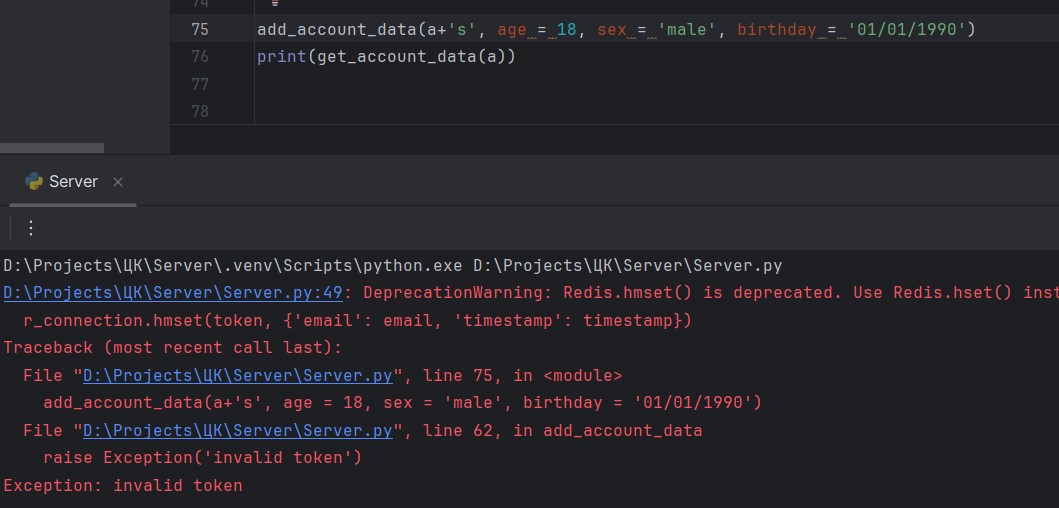
Exception испортил токен, чтобы проверить как реагирует программа на несуществующий токен



add\_account\_data/get\_account\_data



Exception



In [ ]: