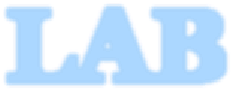


**FACULTY OF COMPUTING**

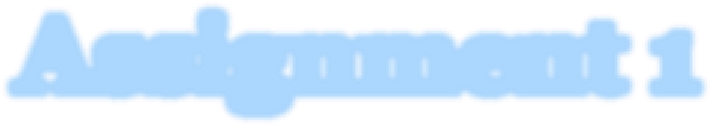
# RIPHAH INTERNATIONAL UNIVERSITY

**Submitted by**

**CMS No.**



**LAB**



**Assignment 1**

Hussnain Ahmad10746

**COURSE CODE – Information Security**

## BS-CS (7C)

**INSTRUCTOR(S): Tanveer Younis**

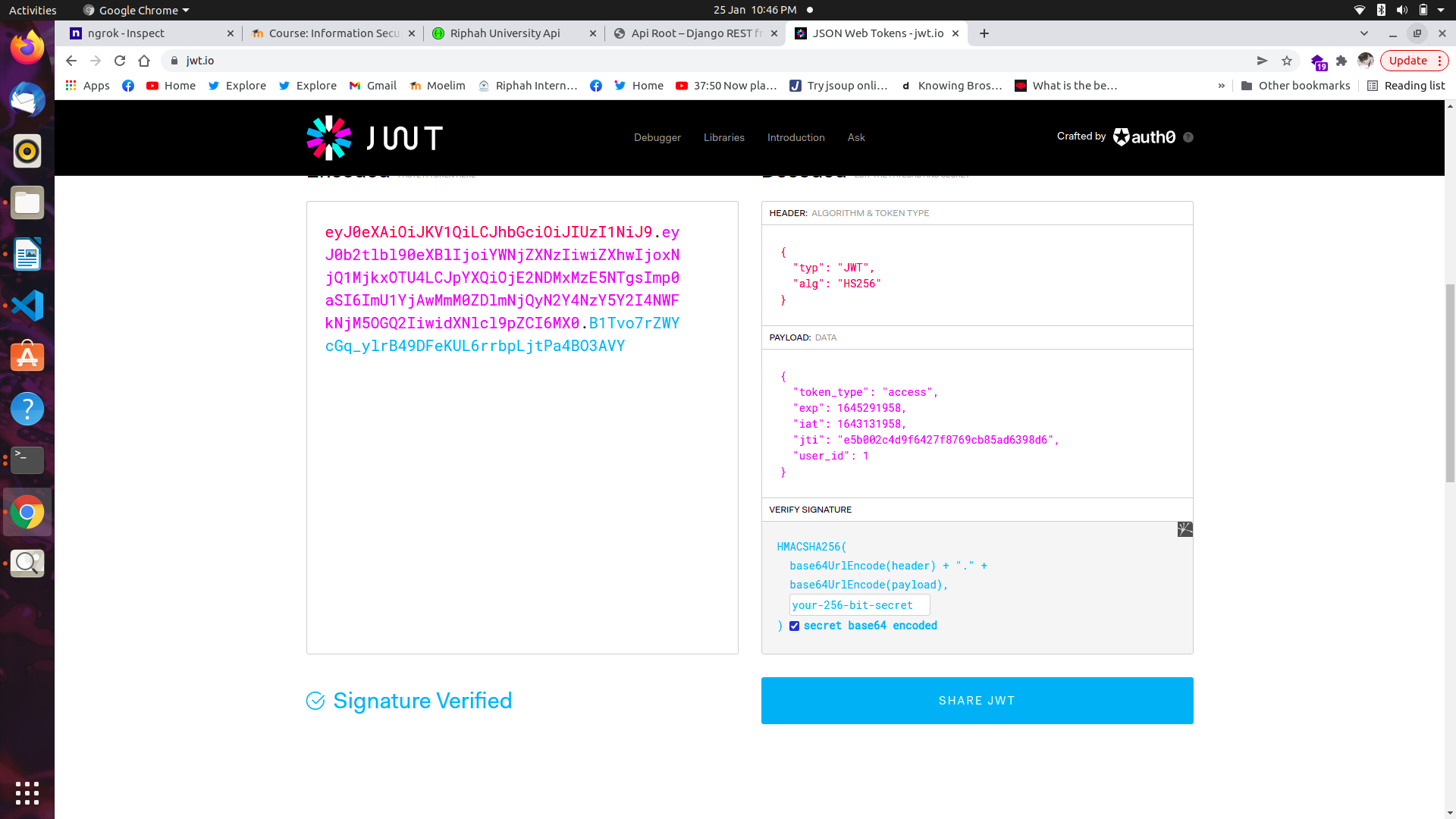
DATE: 25/01/2022

**Q1:** Develop a user authentication system in any language by implementing the knowledge you have learned so far about user authentication.

Solution :-

I am using Django Rest framework technolgies for this project in this I will be using JWT(JSON Web Tokens ) for authentication and the password will be in hash stored in database The JWT will be using HS256 algoithm with secret key and a payload

the Token will look like this



The Jwt consist of three parts Header , Payload that contains user login information as shown in image , and signature

if the user iss not authenticated he will not be given access to anything

Code:-

**Models.py**

Database model creatation file

class User(AbstractUser):

username = models.CharField(max\_length=150,unique=True)

email = models.EmailField(unique=True)

is\_admin = models.BooleanField(default=False)

is\_student = models.BooleanField(default=False)

is\_faculty = models.BooleanField(default=False)

profile\_image = models.ImageField(upload\_to='uploads',null=True)

class Admin(models.Model):

"""Model definition for Admin."""

user = models.OneToOneField(User,on\_delete=CASCADE,primary\_key=True)

portal = models.CharField(null=True,max\_length=200)

class Meta:

"""Meta definition for Admin."""

verbose\_name = 'Admin'

verbose\_name\_plural = 'Admins'

def \_\_str\_\_(self):

"""Unicode representation of Student."""

return self.user.username

def create\_admin(sender,instance,created,\*\*kwargs):

if created:

if instance.is\_admin == True:

Admin.objects.create(user=instance)

print("Admin Created")

post\_save.connect(create\_admin,sender=User)

class Student(models.Model):

"""Model definition for Student."""

the\_class = models.ForeignKey(Class,on\_delete=CASCADE,related\_name='studentsforclass',null=True,blank=True)

user = models.OneToOneField(User,on\_delete=CASCADE,primary\_key=True,related\_name='student')

#gender

roll\_num = models.IntegerField(default=100)

address = models.TextField(null=True)

# course\_id =

class Meta:

"""Meta definition for Student."""

verbose\_name = 'Student'

verbose\_name\_plural = 'Students'

def \_\_str\_\_(self):

"""Unicode representation of Student."""

return self.user.username

def save(self, \*args, \*\*kwargs):

self.roll\_num = self.roll\_num + 1

super().save(\*args, \*\*kwargs) # Call the "real" save() method.

def create\_student(sender,instance,created,\*\*kwargs):

if created:

if instance.is\_student == True:

Student.objects.create(user=instance)

print("Student Created")

post\_save.connect(create\_student,sender=User)

class Faculty(models.Model):

"""Model definition for Faculty."""

user = models.OneToOneField(User,on\_delete=CASCADE,primary\_key=True)

class Meta:

"""Meta definition for Faculty."""

verbose\_name = 'Faculty'

verbose\_name\_plural = 'Facultys'

def \_\_str\_\_(self):

"""Unicode representation of Faculty."""

return self.user.username

def create\_faculty(sender,instance,created,\*\*kwargs):

if created:

if instance.is\_faculty == True:

Faculty.objects.create(user=instance)

print("Faculty Created")

post\_save.connect(create\_faculty,sender=User)

**Serializer.py**

username = serializers.CharField(max\_length=255,min\_length=2)

password = serializers.CharField(max\_length=65,min\_length=8,write\_only=True)

class Meta:

model = User

fields = ['username','password']

**Views.py**

class LoginView(GenericAPIView):

serializer\_class = LoginSerializer

def post(self,request):

data = request.data

username = data.get('username', '')

password = data.get('password', '')

user = auth.authenticate(username=username, password=password)

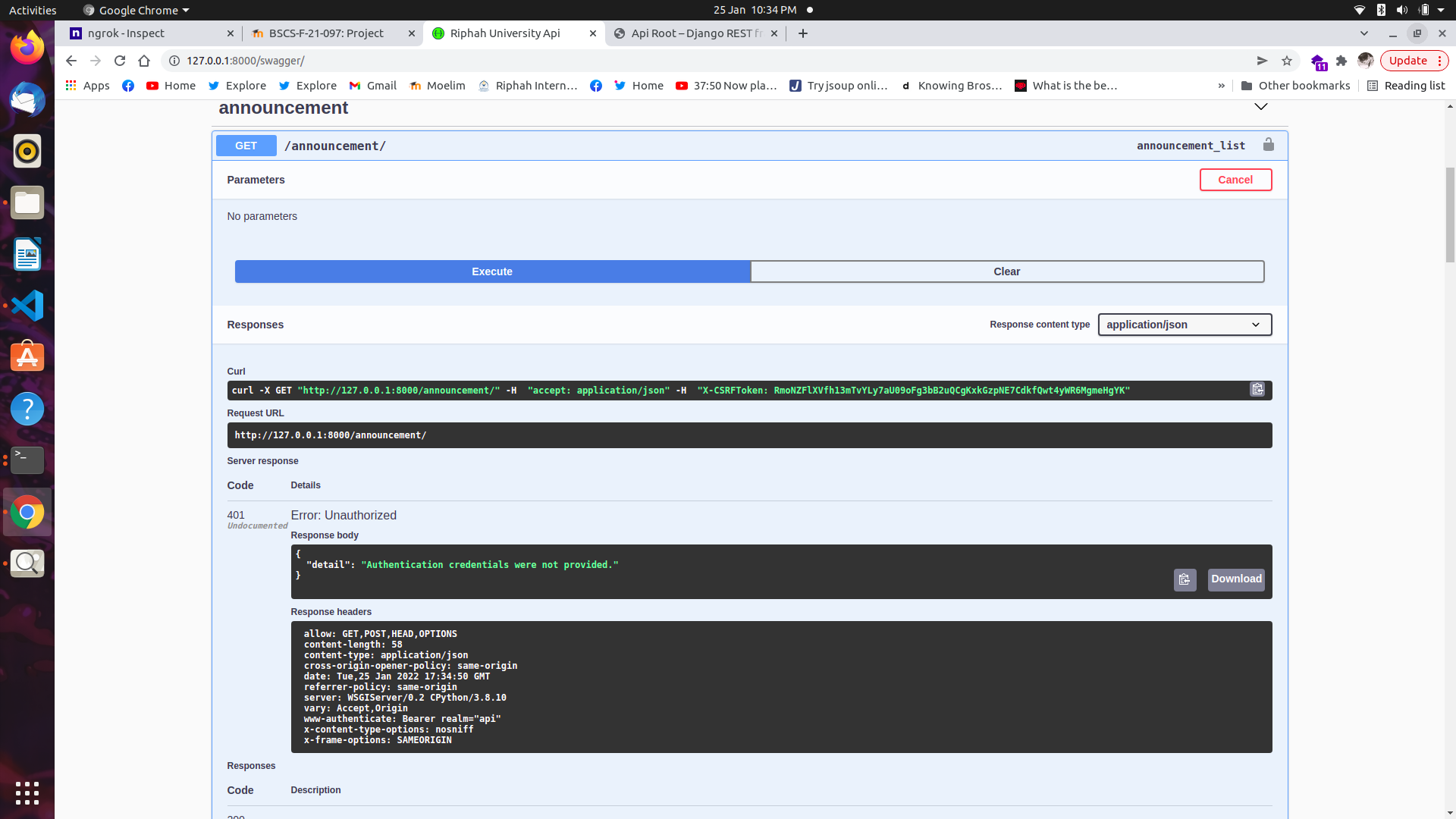
if user:

serializer = UserSerializer(user)

data = {'user':serializer.data,'jwt': get\_tokens\_for\_user(user)}

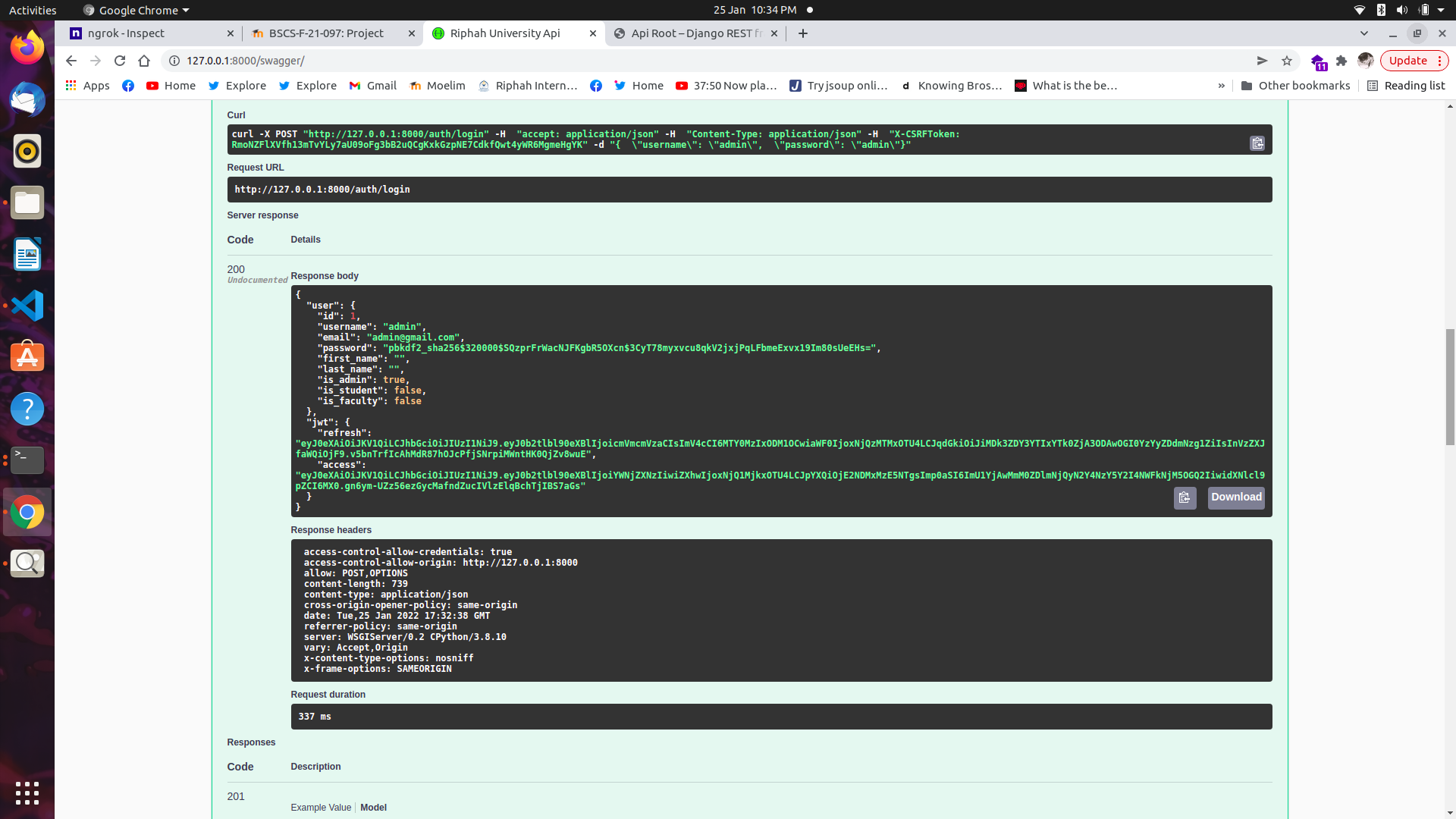
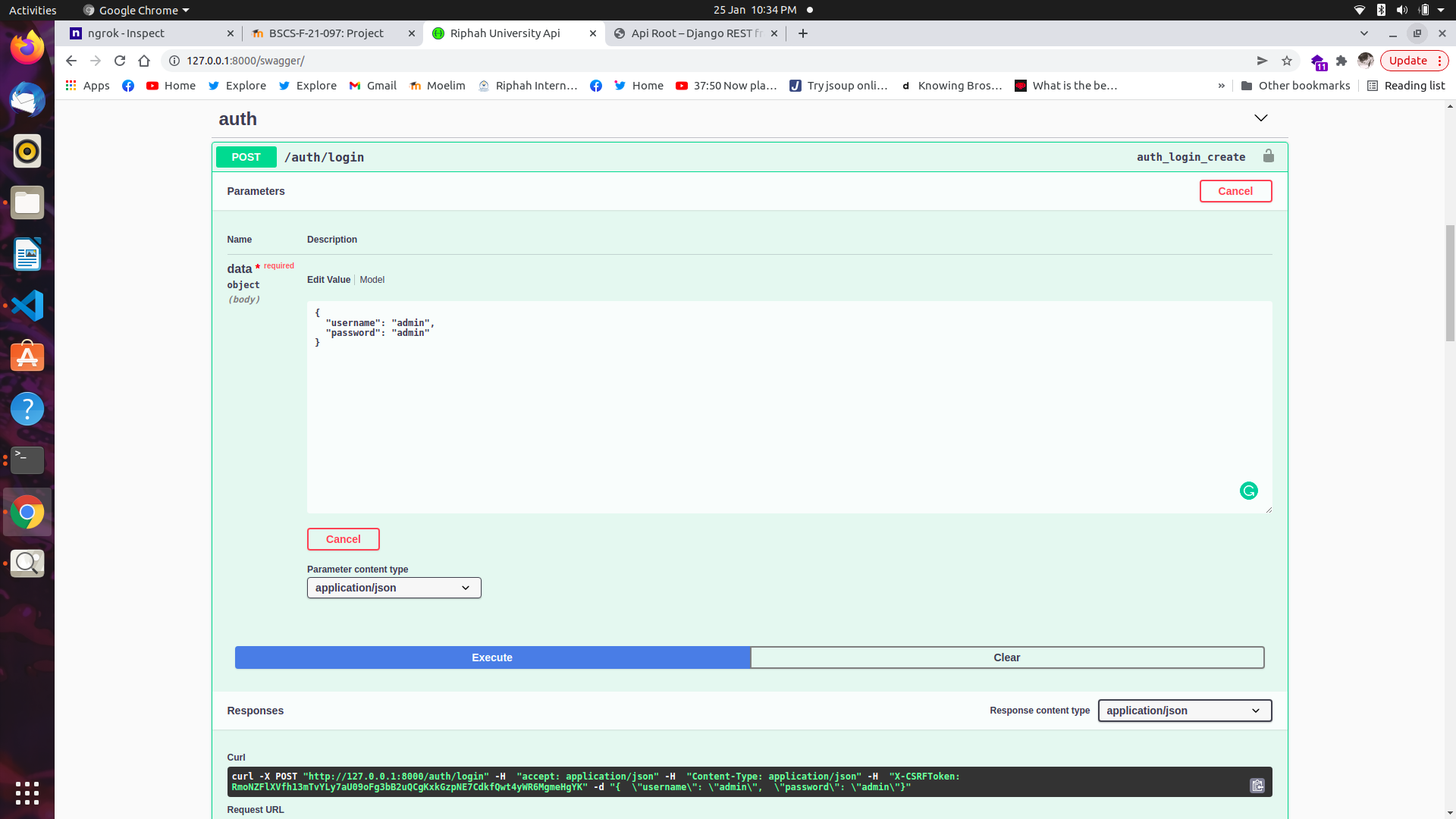
return Response(data, status=status.HTTP\_200\_OK)

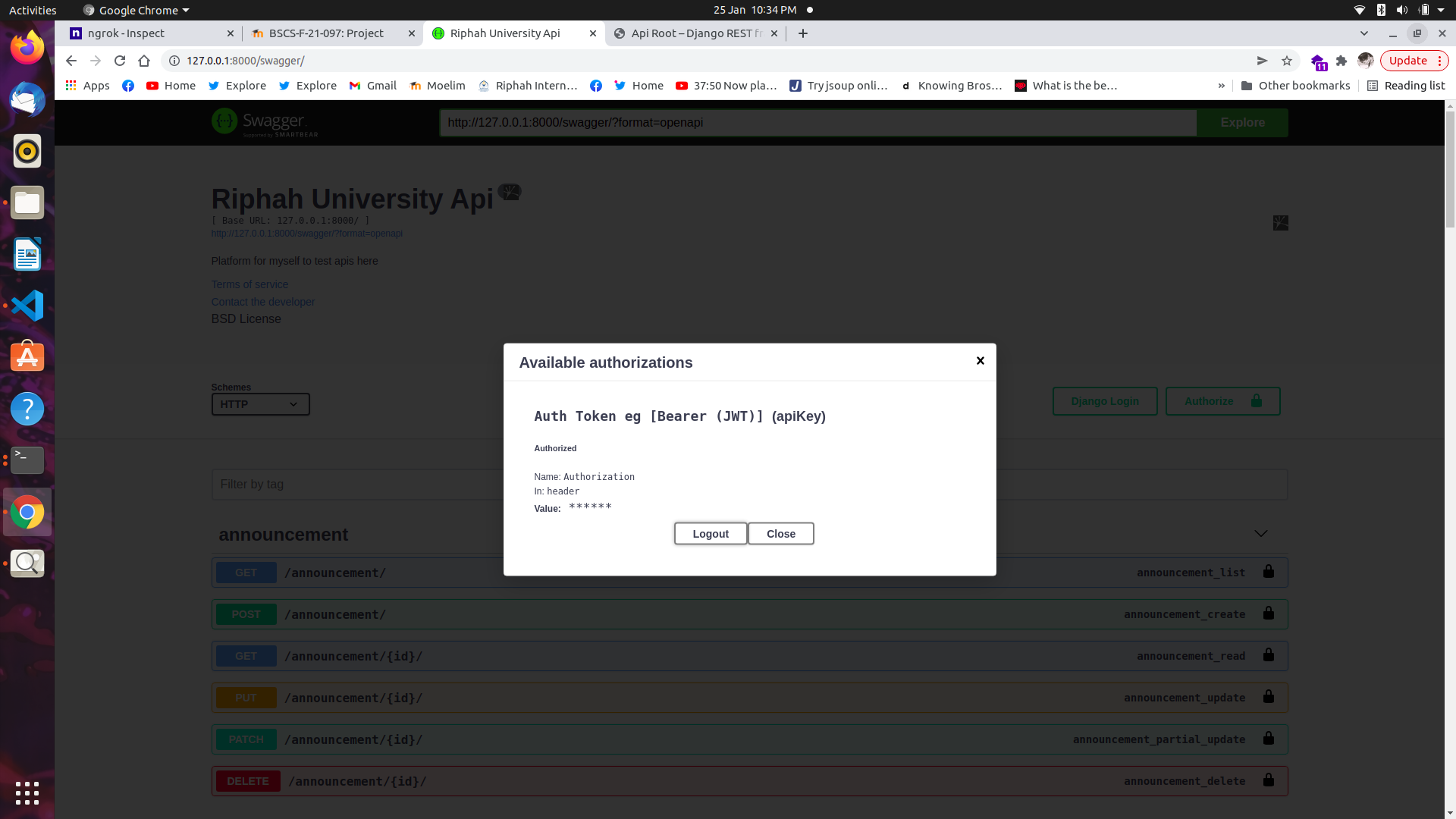
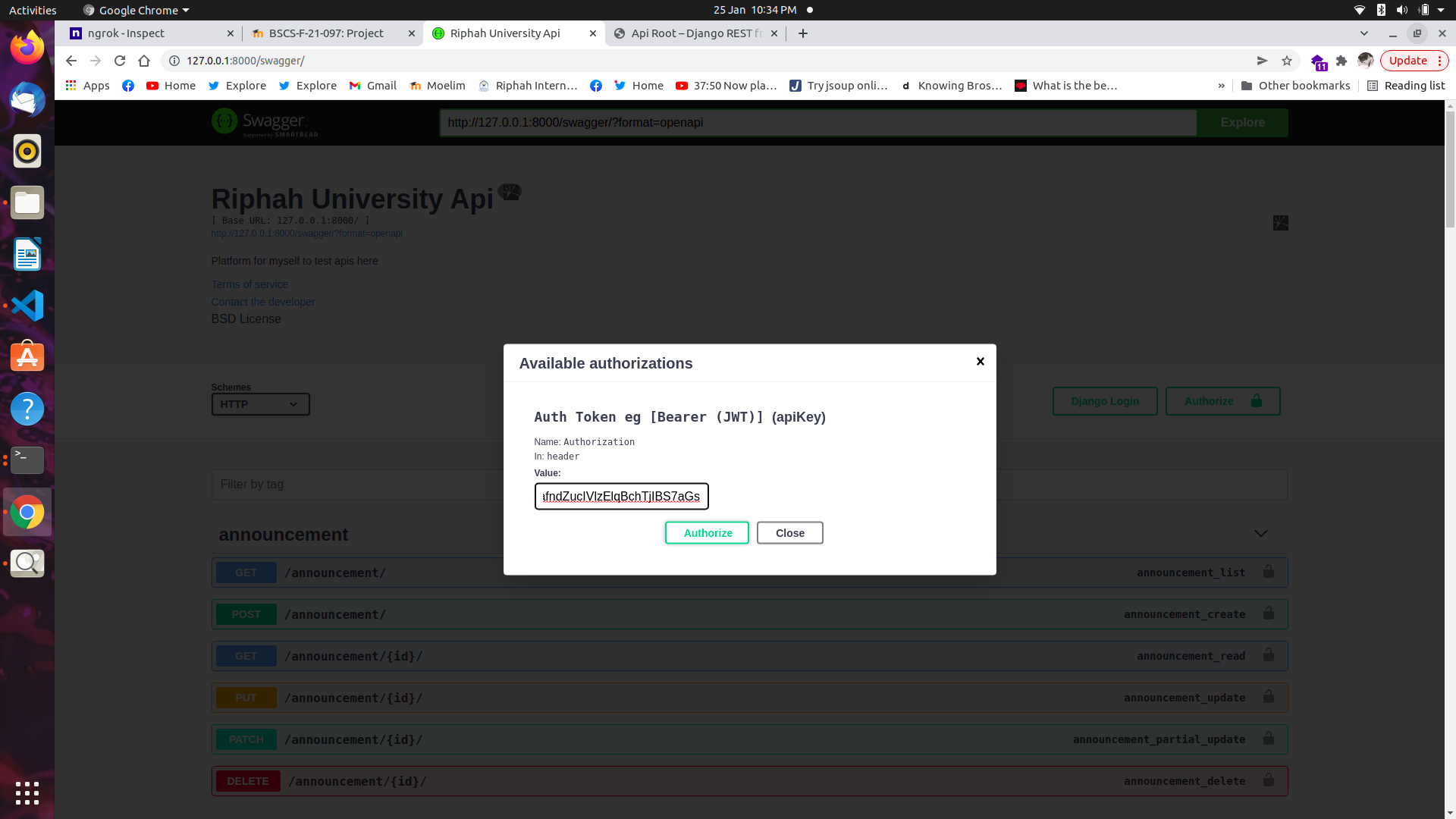
# SEND RES

return Response({'detail': 'Invalid credentials'}, status=status.HTTP\_401\_UNAUTHORIZED)

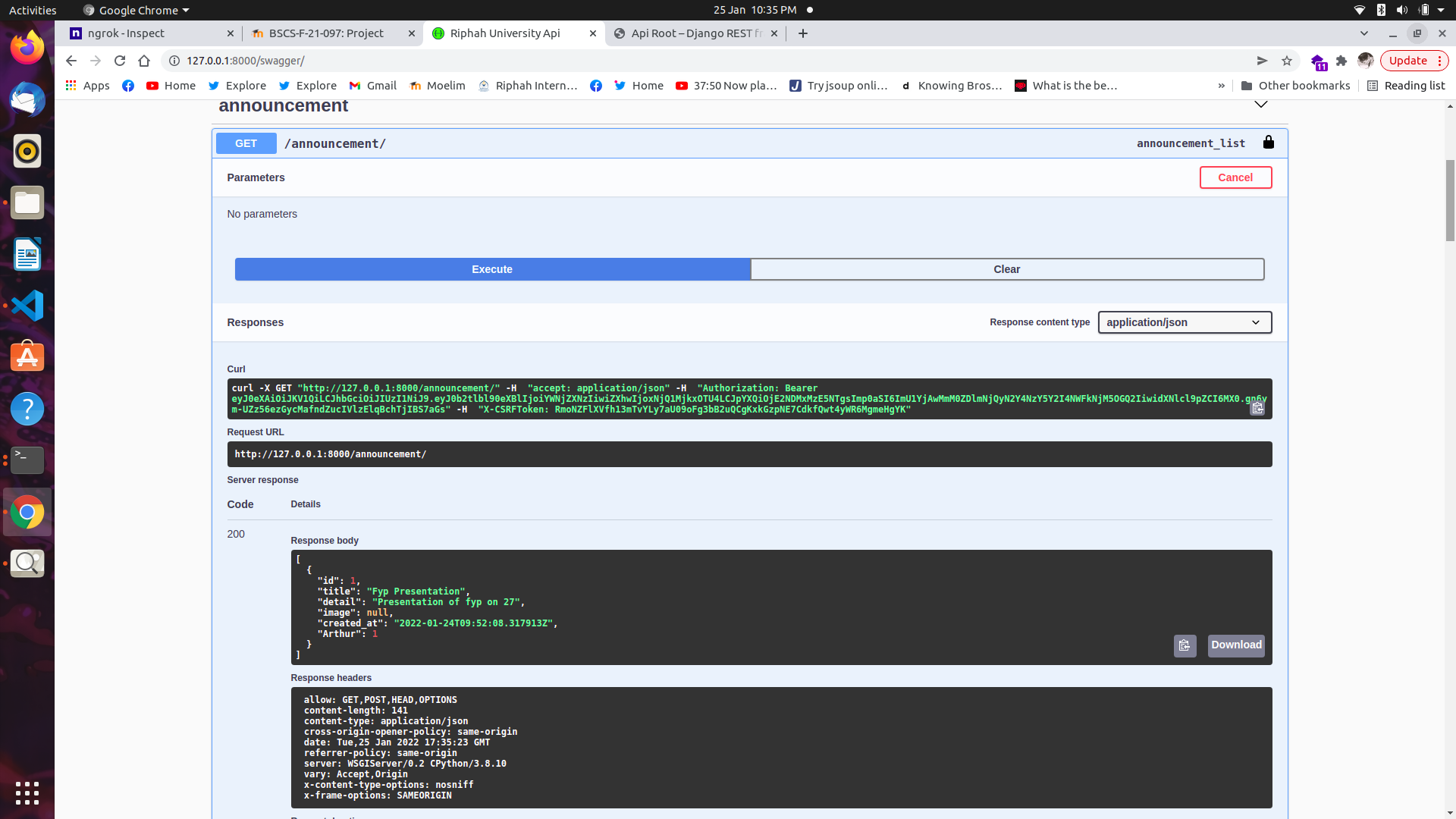
Output Screen Shot:-

if some one try get acces it wil show you are not authenticated

Once you provide credentials it will genrate the token acces frm jwt



and after you provide the credentials it will alow acces for the specif time



Now you can see the data