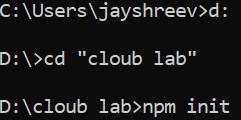
Go to nodejs cmd prompt🡺select the path were the scriptjs is located

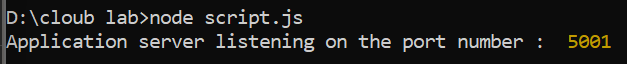
🡺npm init



🡺npm i express

🡺npm i -g nodemon

🡺node script.js



Script file:script.js

const express = require('express');

const app = express();

app.use(express.json());

var lab = [

{device : 'Fan', id : 'F1',state : 'ON'},

{device : 'Fan', id : 'F2',state : 'OFF'},

{device : 'Fan', id : 'F3',state : 'OFF'},

{device : 'Fan', id : 'F4',state : 'ON'},

{device : 'Fan', id : 'F5',state : 'ON'},

{device : 'Light', id : 'L1',state : 'ON'},

{device : 'Light', id : 'L2',state : 'ON'},

{device : 'Light', id : 'L3',state : 'ON'},

{device : 'Light', id : 'L4',state : 'OFF'},

{device : 'Light', id : 'L5',state : 'OFF'}

]

app.get('/',(req,res)=>{

res.send('<center><h1 style = "color : blue;">Welcome to Smart Podium ! ! ! </h1><br><br/><h2>You can control your appliances in a lab and see the status</center>');

});

app.get('/lab',(req,res)=>

{

res.send(lab);

});

//Get the state of the device based on the ID.

app.get('/lab/:id',(req,res)=>

{

const result = lab.find(x => x.id === req.params.id);

if(!result)

{

res.status(404).send('<h3>Sorry please check device ID, No device with the given ID');

}

res.send(result.state);

});

//Adding a new device

app.post('/add',(req,res)=>

{

var devi = req.body;

lab.push(devi);

res.send(lab);

});

//Update the state based on the device id

app.put('/labu/:id',(req,res)=>

{

const f = lab.find(x=>x.id === req.params.id);

if(!f)

{

res.status(404).send('Sorry unable to find record');

}

//Updating state of the device

f.state = req.body.state;

res.send(f);

});

//Deleting a device for the given id

app.delete('/del/:id',(req,res)=>

{

var p = req.params.id;

const result = lab.find(x => x.id === p);

if(!result)

{

res.status(404).send('<h3>Sorry please check device ID, No device with the given ID');

}

const ind = lab.indexOf(result);

lab.splice(ind,1);

res.send(lab);

});

var server = app.listen(5001,function(){

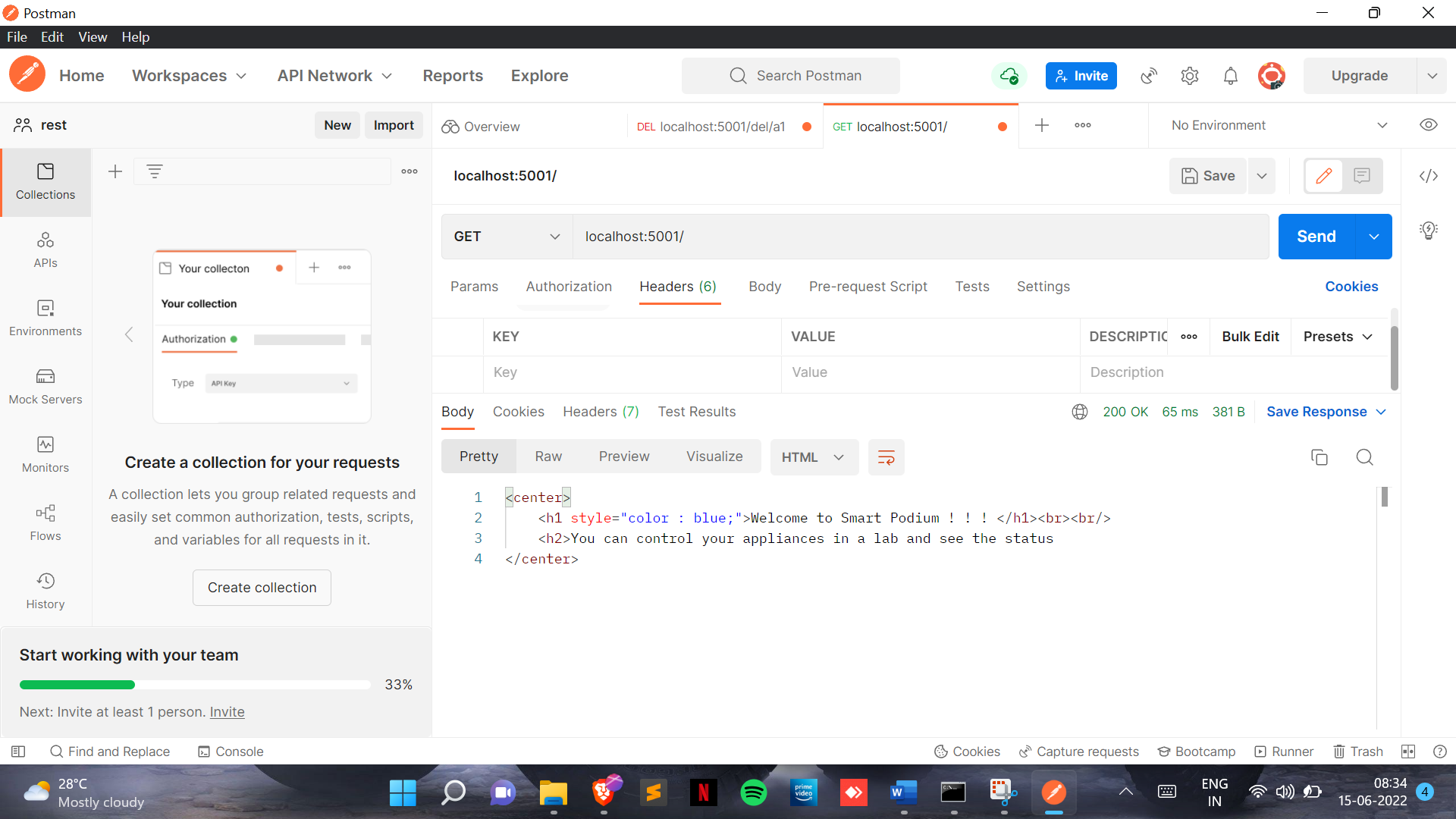
{

var port = server.address().port;

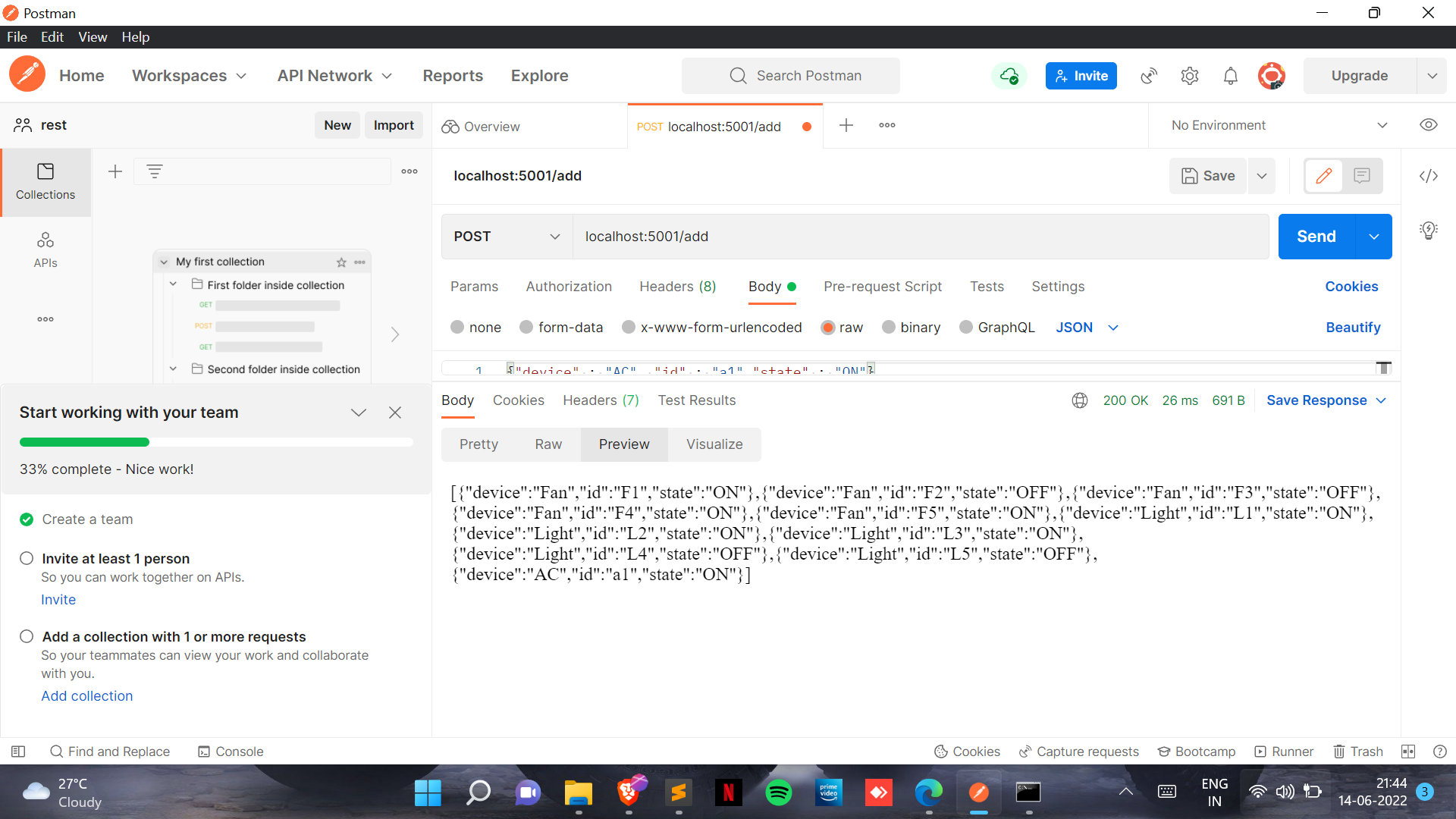
console.log("Application server listening on the port number : ",port);

}})

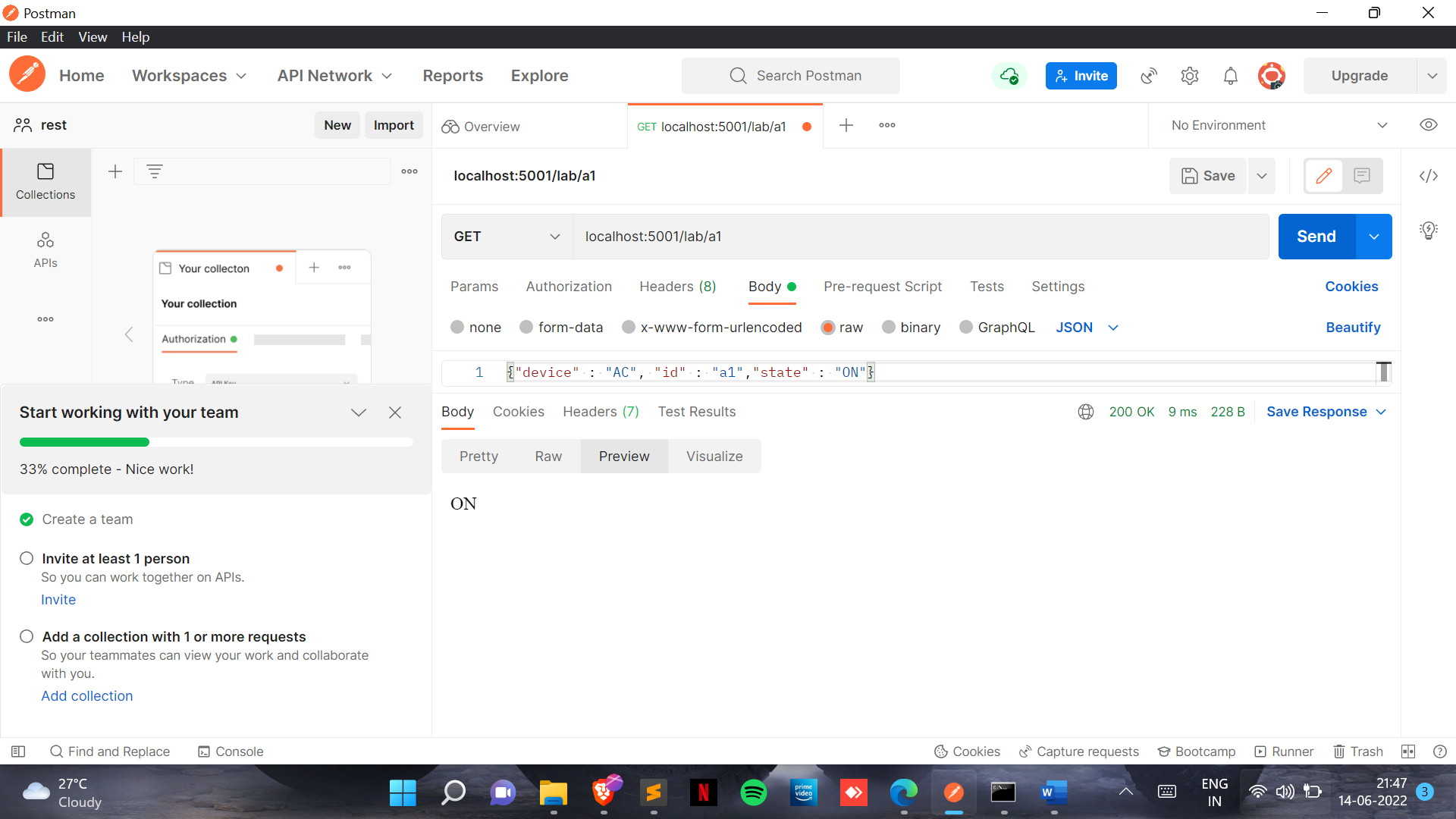
Go to postman and execute the below commands



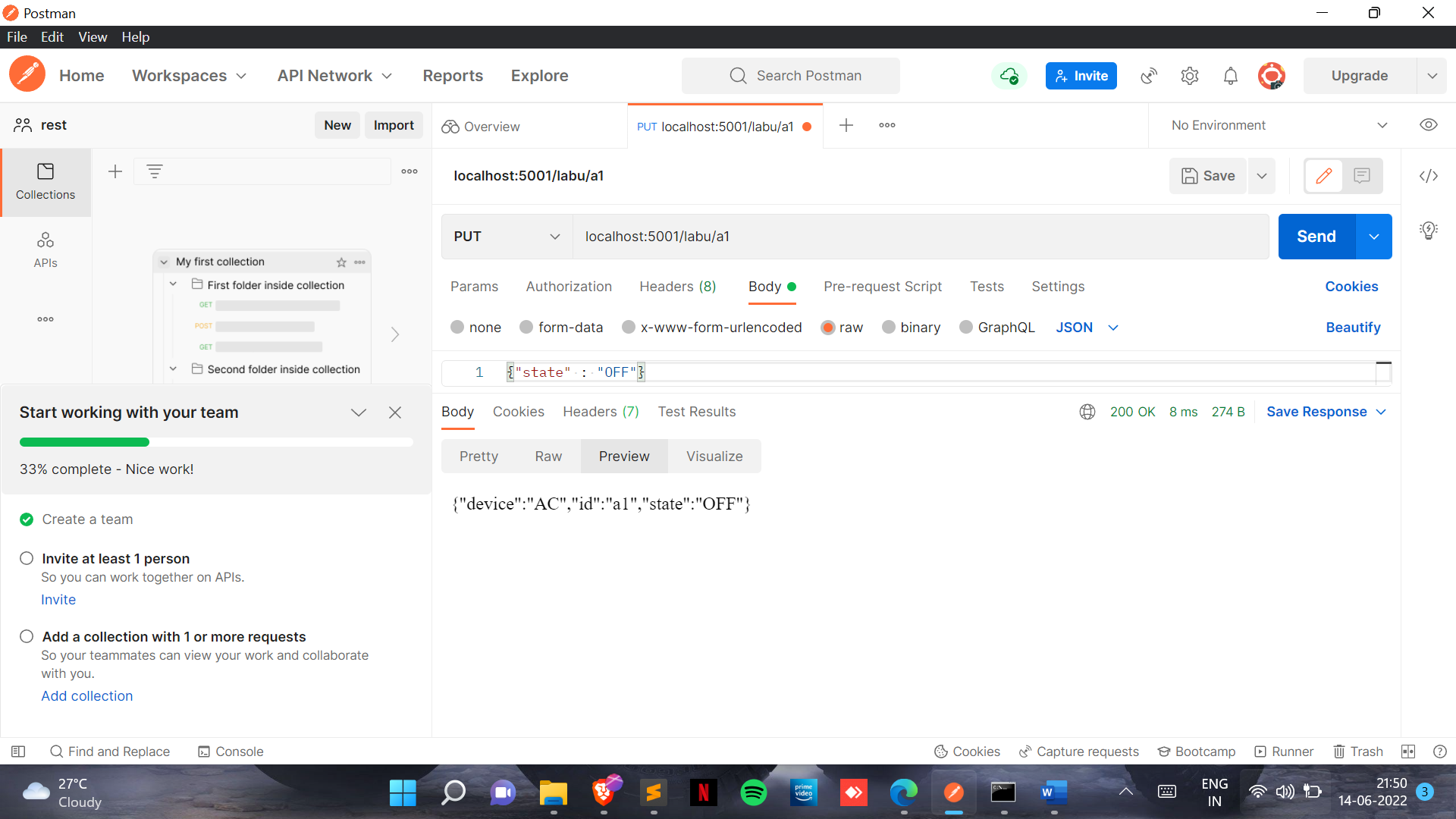
add



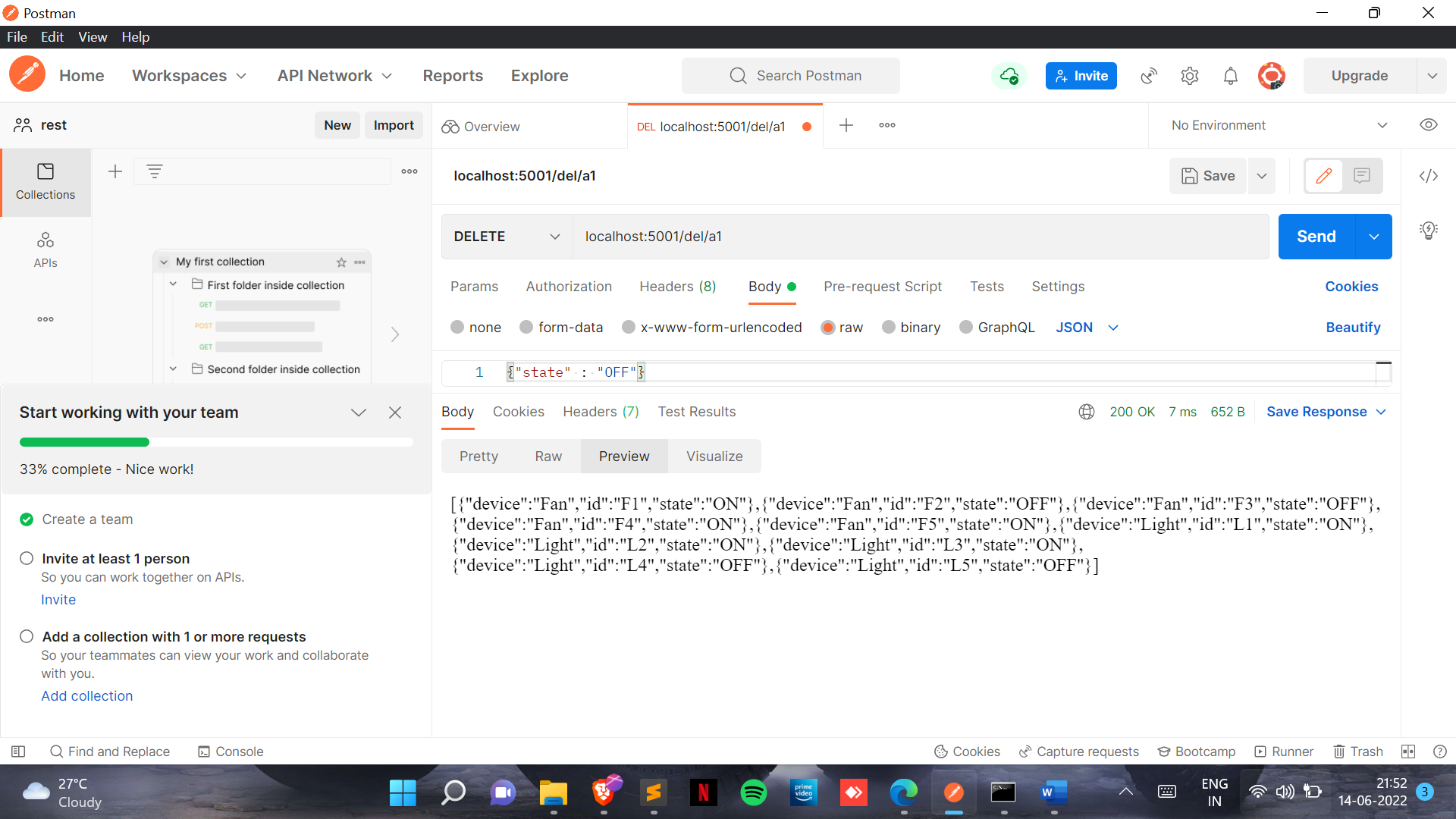
get



Put:update



Delete:

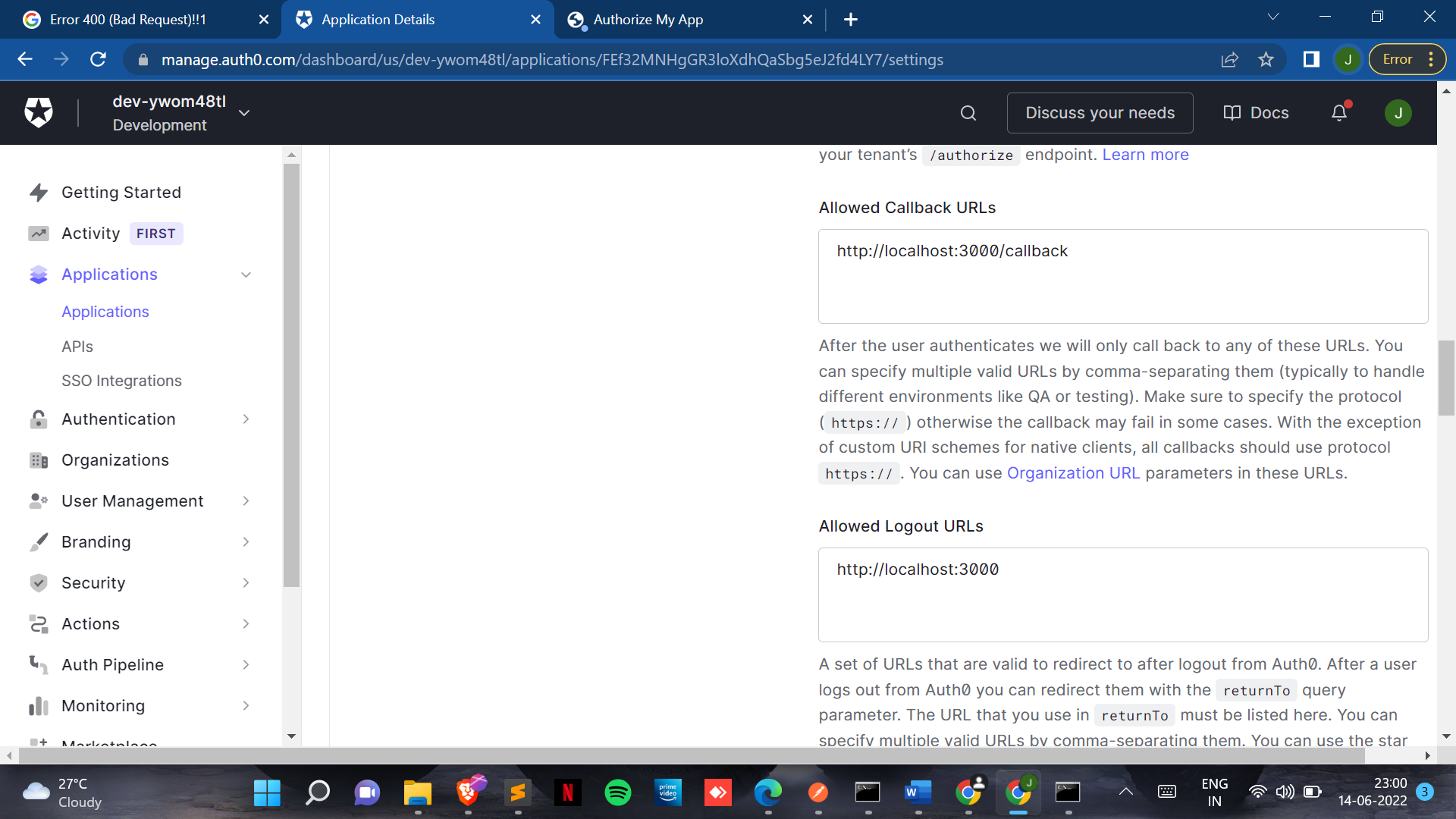


Windows,preferences,server,runtime enviroinment

IAAS

Cmd: pip install -r requirements.txt

Python server.py



SAAS:

<https://dropbox.com/developers/apps>

import dropbox

dropbox\_access\_token= "sl.BIwBWSa\_uyhrjGsNIPIAky5PjGLvmP9A\_8-

UrnHOj63hG9vnZ8lO1L2h6TRNKZ0BRYkJHlwRYHQRvUgBxoX7dIy36CxIF1yur\_IP9DgLjr9WPBYRuCN29-cZNG1pyWl-TbzDxoseWze"

dropbox\_path= "/Ex 05.pdf";

computer\_path=r"E:\Sem 6\18IT670 CCL\EX 05.pdf"

client = dropbox.Dropbox(dropbox\_access\_token)

print("[SUCCESS] dropbox account linked")

client.files\_upload(open(computer\_path, "rb").read(), dropbox\_path)

print("[UPLOADED] {}".format(computer\_path))

metadata, f = client.files\_download("/EX 05.pdf")

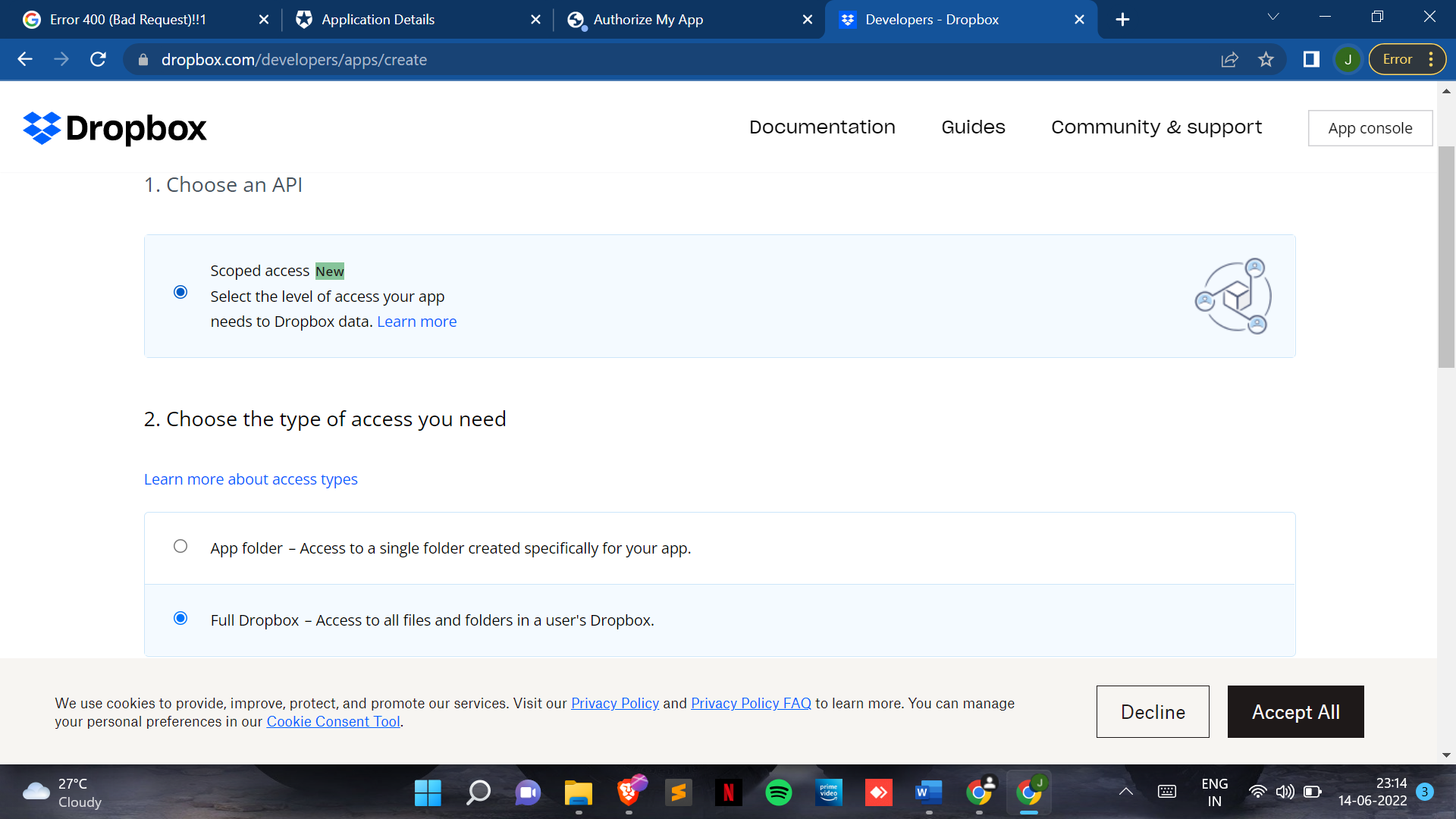
out = open("EX 05\_download.pdf", "wb")

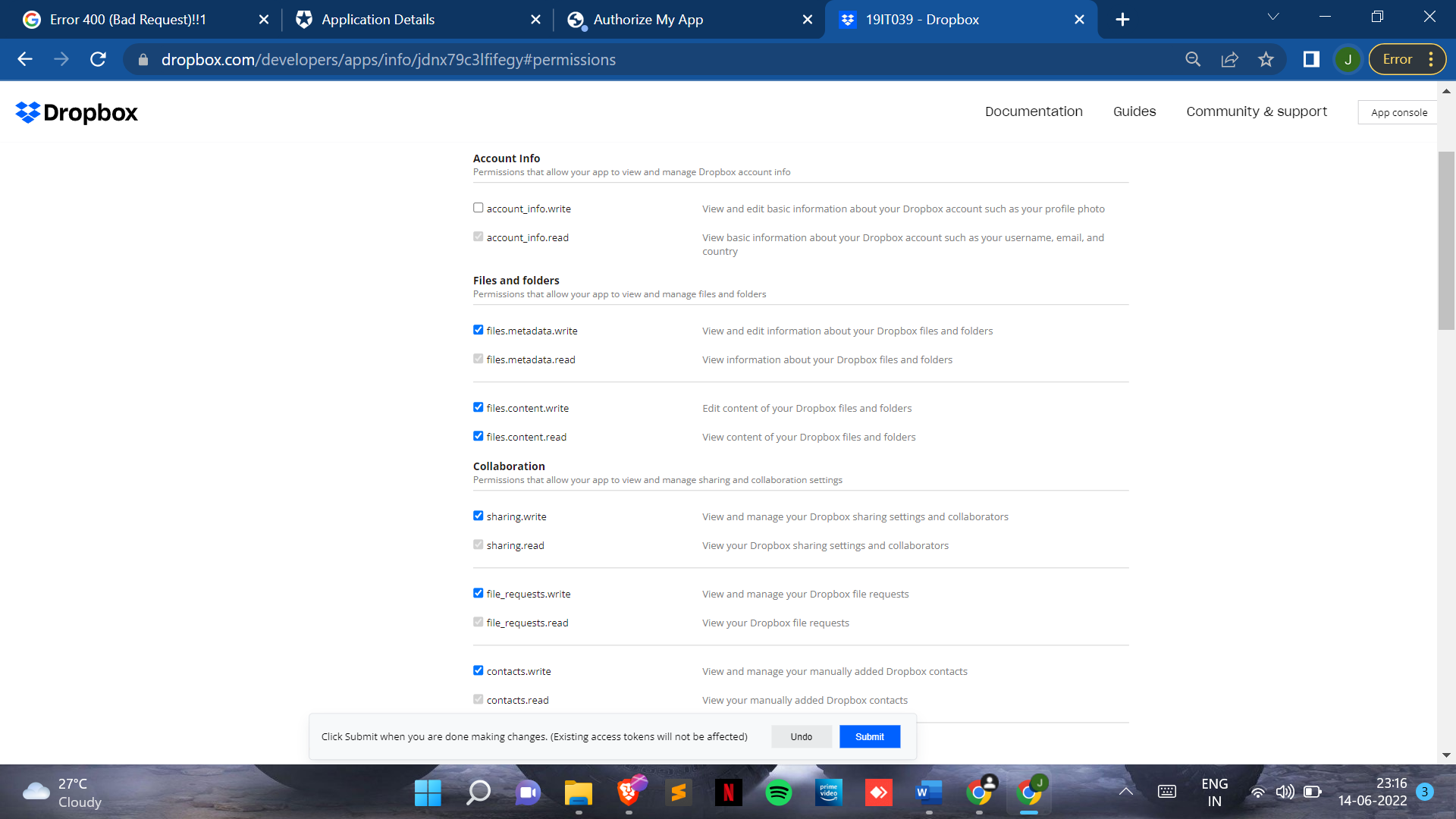
out.write(f.content)

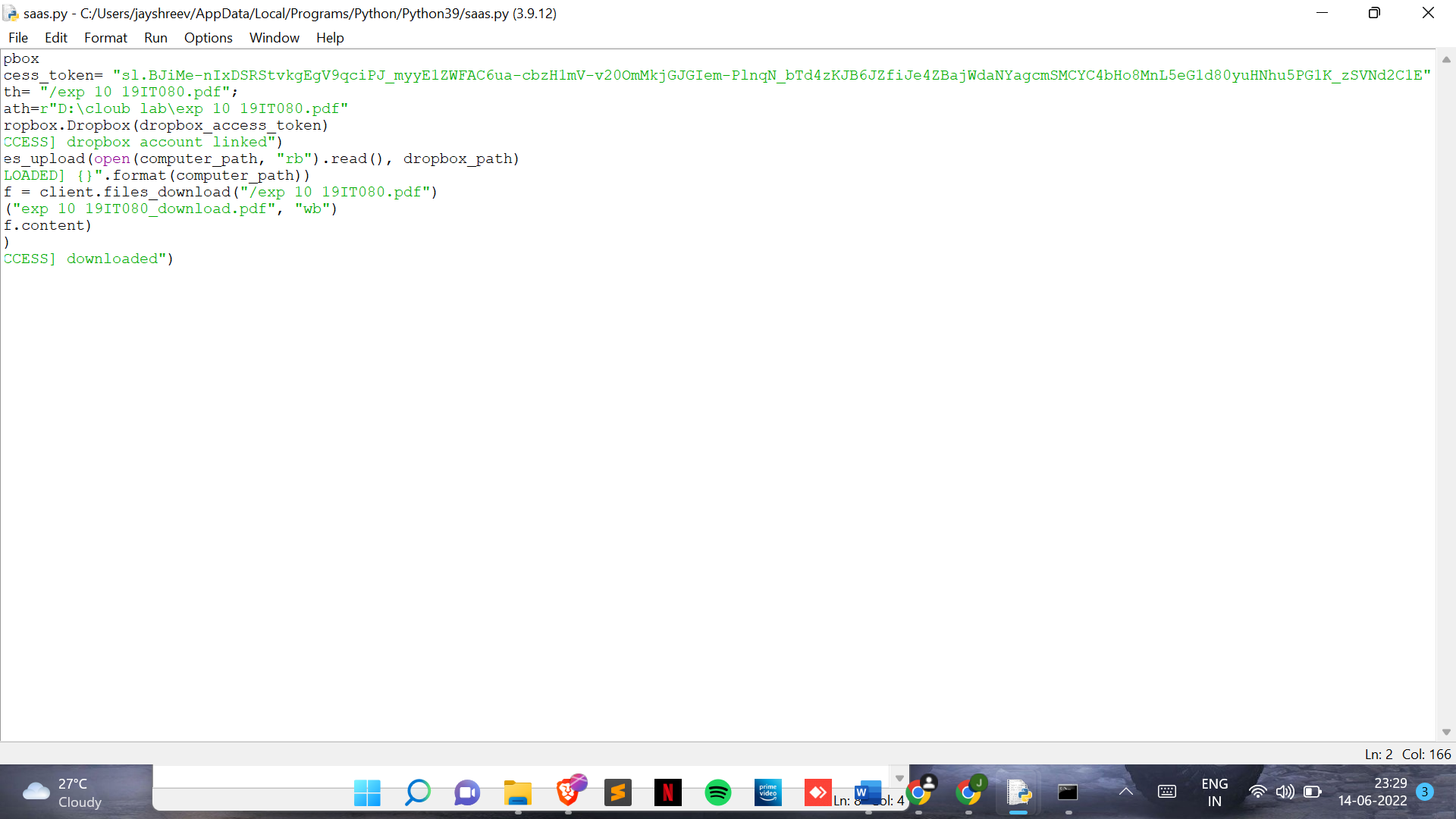
out.close()

print("[SUCCESS] downloaded")

go to IDLE







Dropbox.com/h

**PAAS:**

Ibm cloud foundry

Menu public

**DBAAS:**

https://gist.github.com/nikhilkumarsingh/2c39d161f3d5fe4a40c24d7c9a7a11c3

Search database atlas🡺mongodb atlas🡺signin🡺browse collection🡺create database(once the db is created successfully)🡺database access(delete all the old user)🡺add new user(with username and pass)🡺next click network access(del all the old entries)🡺add new ip🡺allow access from anywhere🡺go to database🡺connect your application🡺open cmd🡺type jupyter notebook(it will be opened) 🡺click new🡺python kernel3🡺copy paste the code🡺check the result in browse collection

Paas:

Ibm cloud foundry

Abc12345678900

Cloud founfry🡺overview🡺create🡺php🡺name🡺create🡺overview🡺enable continuous delivery🡺delivery pipelines🡺new🡺create🡺eclipse orion web id🡺index.php🡺choose html file🡺git🡺go to commit message🡺type commit🡺click commit🡺click push🡺sync🡺clound foundry🡺public🡺click your application🡺visit url

**KVM:**

//kvm

sudo -s

lsb\_release -a

ip a

ip a|more

uname -a

free -m

top

apt-get update

apt-cache search net-tools

apt-get install net-tools

netstat -ntlp

apt-get install qemu-kvm

apt-get install qemu-kvm virt-manager virt-viewer virtinst

apt-get install libvirt-bin qemu-system virt-manager bridge-utils

/etc/init.d/libvirtd restart

/etc/init.d/libvirtd status

virsh

wget "http://download.cirros-cloud.net/0.3.2/cirros-0.3.2-i386-diskk.img

"

**Soap:**

**Windows -prefrence -server- run time -addapache -apache v9.0 -apply and close -window-prefetence-java -userlibrarie-add external jar -(tomcatlib)- new -dynamicwebproject-select project -new jsp file -new-other -wsdl file -loginsoap.wsdl -new -class file -loginsoap.java -rit click on java file -webservice -create web service -high, bottomup java bean -**

**Loginsoap.java**

public class loginsoap {

public String login(String username, String password) { int domainPos =

username.indexOf("@");

String domain = username.substring(domainPos+1,domainPos+4); if(domain.equals("tce")

&& password.equals("password")) {

return "Login Successful"; }

return "Invalid Credentials"; } }

**loginsoap.wsdl**

<?xml version="1.0" encoding="UTF-8"?>

<wsdl:definitions targetNamespace="http://com"

xmlns:apachesoap="http://xml.apache.org/xml- soap" xmlns:impl="http://com"

xmlns:intf="http://com" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<!--WSDL created by Apache Axis version: 1.4 Built on Apr 22, 2006 (06:55:48 PDT)-->

<wsdl:types>

<schema elementFormDefault="qualified" targetNamespace="http://com"

xmlns="http://www.w3.org/2001/XMLSchema">

<element name="login">

<complexType>

<sequence>

<element name="username" type="xsd:string"/>

<element name="password" type="xsd:string"/>

</sequence>

</complexType>

</element>

<element name="loginResponse">

<complexType>

<sequence>

<element name="loginReturn" type="xsd:string"/>

</sequence>

</complexType>

</element>

</schema>

</wsdl:types>

<wsdl:message name="loginRequest">

<wsdl:part element="impl:login" name="parameters">

</wsdl:part>

</wsdl:message>

<wsdl:message name="loginResponse">

<wsdl:part element="impl:loginResponse" name="parameters">

</wsdl:part>

</wsdl:message>

<wsdl:portType name="loginsoap">

<wsdl:operation name="login">

<wsdl:input message="impl:loginRequest" name="loginRequest">

</wsdl:input>

<wsdl:output message="impl:loginResponse" name="loginResponse">

</wsdl:output>

</wsdl:operation>

</wsdl:portType>

<wsdl:binding name="loginsoapSoapBinding" type="impl:loginsoap">

<wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

<wsdl:operation name="login">

<wsdlsoap:operation soapAction=""/>

<wsdl:input name="loginRequest">

<wsdlsoap:body use="literal"/>

</wsdl:input>

<wsdl:output name="loginResponse">

<wsdlsoap:body use="literal"/>

</wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:service name="loginsoapService">

<wsdl:port binding="impl:loginsoapSoapBinding" name="loginsoap">

<wsdlsoap:address location="http://localhost:8081/SOAP/services/loginsoap"/>

</wsdl:port>

</wsdl:service>

</wsdl:definitions>

**RMI:**

import java.rmi.\*;

public interface Adder extends Remote{

public int add(int x,int y)throws RemoteException;

}

import java.rmi.\*;

import java.rmi.server.\*;

public class AdderRemote extends UnicastRemoteObject implements Adder{

AdderRemote()throws RemoteException{

super();

}

public int add(int x,int y){return x+y;}

}

import java.rmi.\*;

import java.rmi.registry.\*;

public class MyServer{

public static void main(String args[]){

try{

Adder stub=new AdderRemote();

Naming.rebind("rmi://localhost:5000/sonoo",stub);

}catch(Exception e){System.out.println(e);}

}

}

import java.rmi.\*;

public class MyClient{

public static void main(String args[]){

try{

Adder stub=(Adder)Naming.lookup("rmi://localhost:5000/sonoo");

System.out.println(stub.add(34,4));

}catch(Exception e){}

}

}

javac \*.java

rmic AdderRemote

rmiregistry 5000

java MyServer

java MyClient