**Student Support Ticket System using ServiceNow and Python**

**Python Code: Student Support Ticket System**

import requests   
from requests.auth import HTTPBasicAuth

# Configuration   
INSTANCE = 'your\_instance\_name' # e.g., 'dev12345'   
USER = 'your\_username'   
PASSWORD = 'your\_password'   
TABLE = 'incident' # Or your custom table like 'x\_edu\_support\_ticket'

# Base URL   
BASE\_URL = f'https://{INSTANCE}.service-now.com/api/now/table/{TABLE}'

# Headers   
HEADERS = {  
 "Accept": "application/json",  
 "Content-Type": "application/json"   
}

# 1. Fetch tickets for students   
def get\_student\_tickets(limit=5):  
 params = {'sysparm\_limit': limit}  
 response = requests.get(BASE\_URL, auth=HTTPBasicAuth(USER, PASSWORD), headers=HEADERS, params=params)

if response.status\_code == 200:  
 data = response.json()  
 for ticket in data['result']:  
 print(f"Ticket: {ticket['number']} - {ticket['short\_description']}")  
 else:  
 print(f"Failed to fetch tickets: {response.status\_code} - {response.text}")

# 2. Create a new student support ticket   
def create\_student\_ticket(student\_name, issue\_description):  
 payload = {  
 "short\_description": f"Issue reported by {student\_name}",  
 "description": issue\_description,  
 "caller\_id": student\_name # must match a valid user in ServiceNow }

response = requests.post(BASE\_URL, auth=HTTPBasicAuth(USER, PASSWORD), headers=HEADERS, json=payload)

if response.status\_code == 201:  
 result = response.json()['result']  
 print(f"Ticket created: {result['number']} - {result['sys\_id']}")

**Student Support Ticket System using ServiceNow and Python**

else:  
 print(f"Failed to create ticket: {response.status\_code} - {response.text}")

# Run sample   
if \_\_name\_\_ == "\_\_main\_\_":  
 print("Fetching student tickets...\n")  
 get\_student\_tickets()

print("\nCreating a new student ticket...\n")  
 create\_student\_ticket("Student A", "Unable to access learning portal")