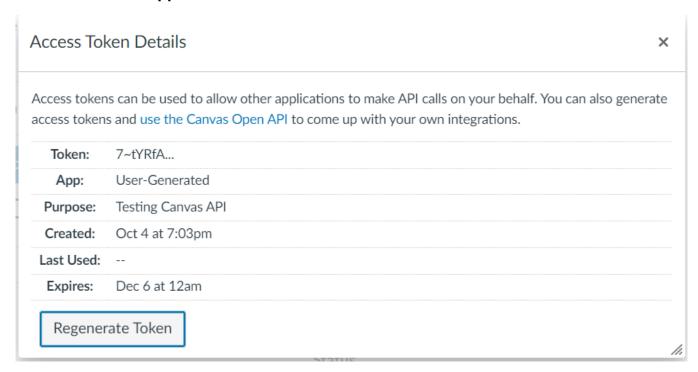
CANVAS LTI ASSIGNMENT

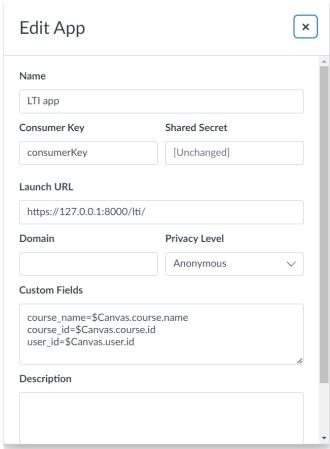
Building a Canvas LTI (Learning Tools Interoperability) application using the Django or Flask framework provides an effective approach to incorporate external tools and services into the Canvas LMS (Learning Management System). For this activity, I have opted to use the Django framework to develop the integration of Learning Interoperability Tools. The purpose of this project is to construct an LTI application that obtains the list of students with their assignment status in the course, and displays the following information about a list of users registered in the course. And also shows about each user, with their assignments in the course, noting whether they have submitted the assignment or not.

STEPS TO DEVELOP CANVAS LTI APPLICATION:

- 1. **Django**: Ensure Django is installed. If not, you can install it using pip: **pip install Django==4.1.1**
- 2. **Canvas LMS Account**: You'll need access to a Canvas LMS instance where the LTI tool can be configured and used. Be sure to create an instructor account. Here is the <u>Canvas Website</u> link.
- 3. For project setup:
 - Firstly, create new Django project file and also create an App file. (Use below code)
 django-admin startproject ProjectName
 cd ProjectName
 python manage.py startapp AppName
- 4. Integrate LTI with Canvas by customized settings mentioned in the LTI Diango Step by Step pdf.
- 5. To create the application, make the below changes in views file:
 - First, configure the API URL in the Django LTI application. This will allow the application to make API calls. The application will fetch data from Canvas LMS.
 - To authenticate the API requests, I create an API Token in Canvas LMS. I include this
 token in the LTI code. This allows the code to have the right authorization to access
 data.
 - I set the course ID and course name as custom fields. The application will retrieve these fields from the Canvas LMS. Next, I filter the users based on their enrollment role. I only list the students. I make an API call to the Canvas LMS Users endpoint (/users) for this. Then, I parse the JSON data to show the list of students.
 - Next, I make an API call to retrieve the assignment data. For each student, I check their submission status for every assignment by accessing the user's submission endpoint. Then, I process the assignment JSON data. Finally, the results are displayed on the website.
- VIDEO URL: Canvas LTI application.

Screenshots of LTI Application in Action:



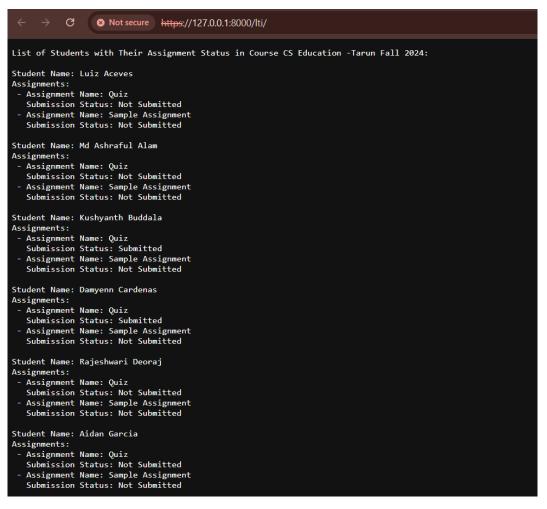


Code from the views file:

```
import requests
from django.http import HttpResponse
def index(request):
   api url = "https://canvas.instructure.com/api/v1/"
   course id = request.POST.get("custom course id")
   course name = request.POST.get("custom course name")
       user response = requests.get(f"{api url}courses/{course id}/users",
eaders=headers,
       if user response.status code != 200:
           return HttpResponse(f"API call failed with status code:
user response.status code}.", status=user response.status code)
       users data = user response.json()
       assignments response = requests.get(f"{api url}courses/{course id}/assignments",
       if assignments response.status code != 200:
           return HttpResponse (f"Failed to retrieve assignments with status code:
assignments response.status code}.", status=assignments response.status code)
       assignments data = assignments response.json()
   except requests.exceptions.Timeout:
```

```
submission url =
f"{api url}courses/{course id}/assignments/{assignment id}/submissions/{user id}"
            submission response = requests.get(submission url, headers=headers)
            if submission response.status code == 200:
               submission data = submission response.json()
           user assignments.append({
                'Submission Status': submitted
       result.append({
   for user in result:
       response text += f"Student Name: {user['Student Name']}\nAssignments:\n"
           submission status = 'Submitted' if assignment.get('Submission Status',
           response text += f" - Assignment Name: {assignment name}\n"
   return HttpResponse(response text, content type="text/plain")
```

Results displayed on Website:



```
Student Name: Malin Jackson
Assignments:
 - Assignment Name: Quiz
   Submission Status: Not Submitted
 - Assignment Name: Sample Assignment
   Submission Status: Not Submitted
Student Name: Sai Kiran Ganesh Kumar
Assignments:
 - Assignment Name: Quiz
 Submission Status: Submitted
- Assignment Name: Sample Assignment
   Submission Status: Not Submitted
Student Name: Jay Kurivella
Assignments:
 - Assignment Name: Ouiz
   Submission Status: Not Submitted
 - Assignment Name: Sample Assignment
   Submission Status: Not Submitted
Student Name: Mounika Makkena
Assignments:
 - Assignment Name: Quiz
 Submission Status: Not Submitted
- Assignment Name: Sample Assignment
   Submission Status: Not Submitted
```