

## 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 [Canvas Website 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 Django 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:

### Access Token Details

Access tokens can be used to allow other applications to make API calls on your behalf. You can also generate access tokens and [use the Canvas Open API](#) to come up with your own integrations.

Token:	7~tYRfA...
App:	User-Generated
Purpose:	Testing Canvas API
Created:	Oct 4 at 7:03pm
Last Used:	--
Expires:	Dec 6 at 12am

Regenerate Token

### Edit App

Name

LTI app

Consumer Key

consumerKey

Shared Secret

[Unchanged]

Launch URL

https://127.0.0.1:8000/lti/

Domain

Privacy Level

Anonymous

Custom Fields

course\_name=\$Canvas.course.name  
course\_id=\$Canvas.course.id  
user\_id=\$Canvas.user.id

Description

## Code from the views file:

```
import requests
from django.http import HttpResponse

def index(request):
    api_url = "https://canvas.instructure.com/api/v1/"
    # Authentication headers (replace with your API token)
    headers = {
        "Authorization": "Bearer
7~tYRfABEynHxxy2cryrQzeLN6DW2BaKzvKcEuB9zmRFXy2nEtBaLYXZAwecVMLcDt"
    }

    # Retrieve course information from the request
    course_id = request.POST.get("custom_course_id")
    course_name = request.POST.get("custom_course_name")

    # Check for missing course information
    if not course_id or not course_name:
        return HttpResponse("Course ID or Course Name is missing.", status=400)

    # Filter only student roles
    roles = ["student"]

    try:
        # Fetch users enrolled in the course with the role of 'student'
        user_response = requests.get(f"{api_url}courses/{course_id}/users",
headers=headers,
                                params={"enrollment_type[]": roles}, timeout=10)

        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()
        user_list = [(user['id'], user['name']) for user in users_data]

        # Fetch assignments for the course
        assignments_response = requests.get(f"{api_url}courses/{course_id}/assignments",
headers=headers, timeout=10)

        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:
        return HttpResponse("The request to the Canvas API timed out.", status=504)

    result = []

    # Iterate through users and their assignment submission status
    for user_id, user_name in user_list:
        user_assignments = []

        for assignment in assignments_data:
```

```

        assignment_id = assignment['id']
        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()
            print(submission_data) # Debugging: print the full submission data
            # Adjust submission status check to handle multiple states
            submitted = submission_data.get('workflow_state') in ['submitted',
'graded']
        else:
            submitted = False

        user_assignments.append({
            'Assignment Name': assignment['name'],
            'Submission Status': submitted
        })

        result.append({
            'Student Name': user_name,
            'Assignments': user_assignments
        })

# Prepare the response text
response_text = f"List of Students with Their Assignment Status in Course
{course_name}:\n\n"

for user in result:
    response_text += f"Student Name: {user['Student Name']}\nAssignments:\n"
    for assignment in user['Assignments']:
        assignment_name = assignment.get('Assignment Name', 'Name not available')
        submission_status = 'Submitted' if assignment.get('Submission Status',
False) else 'Not Submitted'
        response_text += f" - Assignment Name: {assignment_name}\n"
        response_text += f"    Submission Status: {submission_status}\n"
        response_text += "\n"

return HttpResponse(response_text, content_type="text/plain")

```

## Results displayed on Website:

```
<  →  ↻  Not secure  https://127.0.0.1:8000/iti/

List of Students with Their Assignment Status in Course CS Education -Tarun Fall 2024:

Student Name: Luiz Aceves
Assignments:
- Assignment Name: Quiz
  Submission Status: Not Submitted
- Assignment Name: Sample Assignment
  Submission Status: Not Submitted

Student Name: Md Ashraful Alam
Assignments:
- Assignment Name: Quiz
  Submission Status: Not Submitted
- Assignment Name: Sample Assignment
  Submission Status: Not Submitted

Student Name: Kushyanth Buddala
Assignments:
- Assignment Name: Quiz
  Submission Status: Submitted
- Assignment Name: Sample Assignment
  Submission Status: Not Submitted

Student Name: Damyenn Cardenas
Assignments:
- Assignment Name: Quiz
  Submission Status: Submitted
- Assignment Name: Sample Assignment
  Submission Status: Not Submitted

Student Name: Rajeshwari Deoraj
Assignments:
- Assignment Name: Quiz
  Submission Status: Not Submitted
- Assignment Name: Sample Assignment
  Submission Status: Not Submitted

Student Name: Aidan Garcia
Assignments:
- Assignment Name: Quiz
  Submission Status: Not Submitted
- Assignment Name: Sample Assignment
  Submission Status: Not Submitted
```

```
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: Quiz
  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
```