Summer Fellowship Report

On

Workshop Integration

Submitted by

T Yochan Sandesh

Under the guidance of

Prof. Prabhu Ramachandran
Department of Aerospace Engineering
IIT Bombay

August 1, 2018

Acknowledgment

The time I spent in FOSSEE, IIT Bombay as an intern from May 2018 to July 2018 was a memorable one for me as it was rich in experience sharing and helped me discover my potential. I have had so many rich experiences and opportunities that I personally believe will forever shape and influence my professional life while fostering personal growth and development.
This report would not have been possible without the contribution and collaboration of others. My sincere gratitude:

- To The Prof. Kannan M. Moudgalya, Department of Chemical Engineering, IIT Bombay), the PI of FOSSEE, for giving us the opportunity to do an internship within the organization.
- To The Prof. Prabhu Ramchandran, Deparatment of Aerospace Engineering, IIT Bombay our internship guide for his constant support and supervision thorughout the internship.
- To the Senior Project Manager, FOSSEE, Mrs. Usha Viswanathan along with the FOSSEE team at IIT Bombay. With their patience and openness they created an enjoyable working environment.
- To the Project mentor Mr. Binson Babu and Project head Mr. Mahesh Gudi for their continous support, supervision motivation and guidance throughout the tenure of my project in spite of their hectic schedule who truly remained driving spirit in my project and their experience gave me the light in handling this project and helped me in clarifying the abstract concepts, requiring knowledge and perception, handling critical situations and in understanding the objective of my work.
- To the other research engineers Mr. Akshen Doke and Mr. P. Aditya for their constant support and guidance by providing me with the knowledge required to complete my project
- To all our fellow colleagues, with whom I have completed the fellowship. We experienced great things together.

To all of you, I extend my deepest gratitude.

Contents

Chapter 1

Introduction

Yaksh is an Online Test Interface for creating various courses and conducting online programming quizzes. It supports various programming languages like :- C, C++, Python and simple Bash. User can solve any questions by using these languages. Yaksh uses ”test cases” to test the the implementations of the students. It also supports simple multiple choice questions and file uploads so that user can easily submit his code. Not only you can practice the questions even you can also conduct a programming quiz that supports various languages.

Workshop Booking is an Online Platform for proposing, booking or creating workshops on various courses based on the availability of the instructors and the coordinators. It has flexibilty to prepone/postpone the date of booking.

In this project we are going to work on both the sites inorder to fulfill the integrity constraint.

Chapter 2

Area of concern

2.1 Integration

As the courses for any workshop should be created on Yaksh Portal, the workshop booking portal must have integration with Yaksh. This will help us in handling accounts of the users in both the platforms, and the key used here is the email id of a user.

2.2 Automation

After an instructor accepts, the proposed/booked workshops in the workshop booking he must create a course corresponding to that workshop on yaksh portal, and this must be done in a generic manner based on the workshop details. the following are the steps used by users on yaksh to create a course :

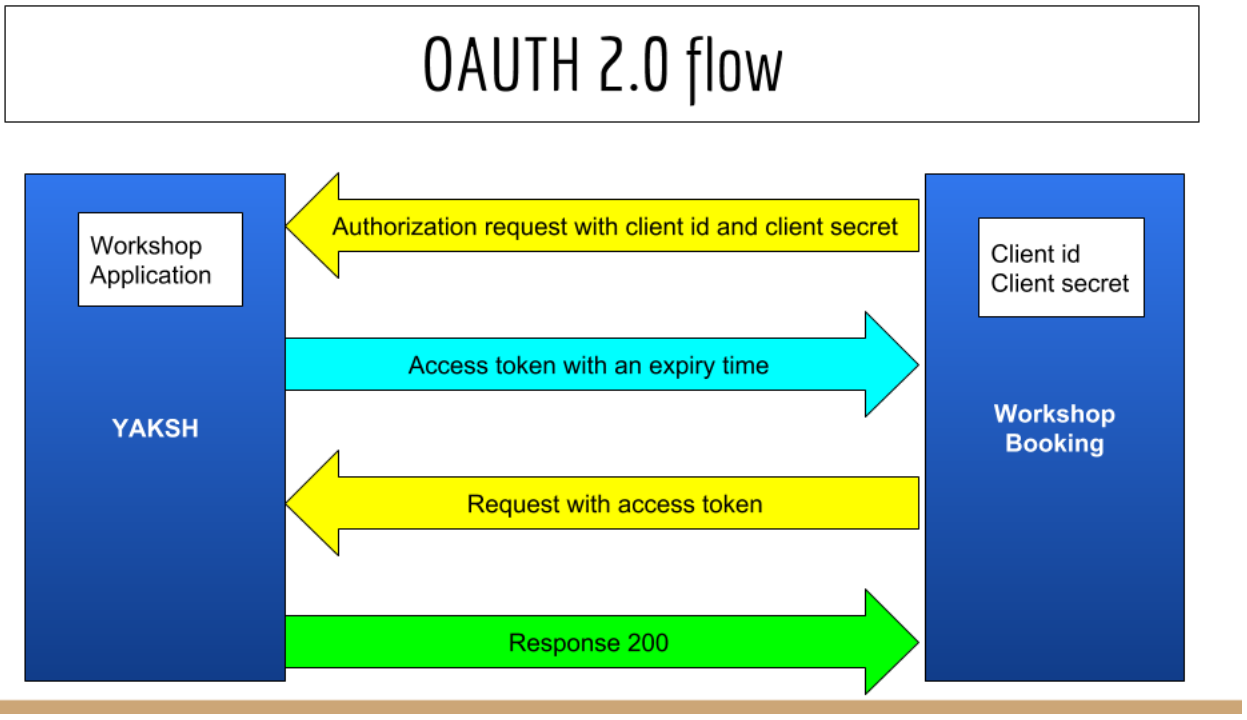
- **Coordinator proposes/books a workshop**
- **Instructor accepts the workshop**
- **Instructor logins to Yaksh**
- **Create a course**
- **Add lessons, quizzes, and modules to the Course created**
- **Alter Grading system, (If necessary)**

Chapter 3

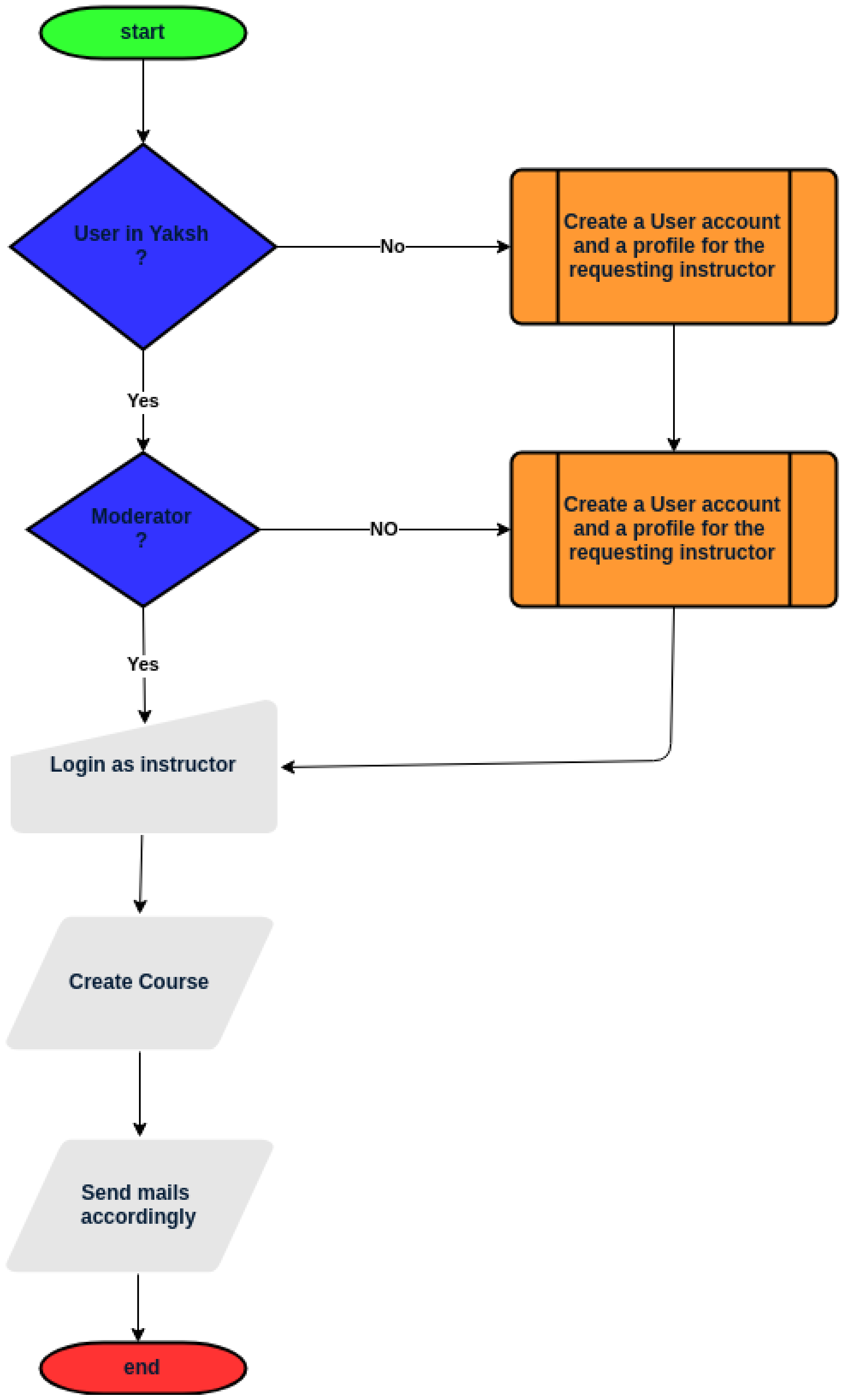
Design of the project

In this project we will use an API at Yaksh to receive the request to create a workshop along with the post-data required to create a course and user. Also inorder to make the request secure, we are using OAuth 2.0 toolkit, which is an open standard for token-based authentication and authorization on the Internet. The workflow can be illustrated in the following flow charts.

3.1 OAuth


OAUTH 2.0 flow
Workshop Application
YAKSH
Authorization request with client id and client secret
Access token with an expiry time
Request with access token
Response 200
Client id
Client secret
Workshop Booking

3.2 Yaksh


start
User in Yaksh ?
No
Create a User account and a profile for the requesting instructor
Yes
Moderator ?
NO
Create a User account and a profile for the requesting instructor
Yes
Login as instructor
Create Course
Send mails accordingly
end

Chapter 4

Implementation

4.1 Workshop Booking

- **workshop course request**
  This method is called when the instructor accepts the course. It manages the oauth authentication, later it sends the workshop details to the yaksh requesting to create a course based on the data.

```
def create_workshop_course(user,......., wtitle, user_position):
    """ requests yaksh portal to create a course """
    imail = User.objects.get(id=user.profile.user_id).email
    data = [('grant_type','client_credentials')]
    response=requests.post('http://localhost:8001/o/token/',
                           data= data,
                           auth=(settings.CLIENT_ID,settings.CLIENT_SECRET))
    if(response.status_code == 200):
        access_token=response.json()['access_token']
        workshop_data = dict(
                             instructor_username=auth_user.username,
                             ....
                             ....
                             position=user_position,
                             access_token=access_token
                             )
        yaksh_response = requests.post("http://127.0.0.1:8001/exam/workshop_co
                                       data=workshop_data)
        return yaksh_response
    else:
        return response
```

4.2 Online test master

- Workshop course
  - This method is called through the API from workshop booking, it is also decorated by protected resource(), which secures the API through a variable access code.
  - It then verifies about the user by fetching his email from workshop booking, If there is no user with that email a new account with profile is created.
  - Later It checks whether the user is of moderator group, if not it makes them a moderator and creates a course with this user.

```
@csrf_exempt
@protected_resource()
def workshop_course(request):
    """ authenticates the user from workshop booking """
    if request.method == 'POST':
        workshop_data = request.POST.dict()
        workshop_title = workshop_data['workshop_title']
        days = int(workshop_title[workshop_title.index("day") - 1])
        workshop_title = workshop_title[:workshop_title.index(',') - 5]
        if(workshop_data['position'] == 'instructor' and days != 0):
            user_created = False
            try:
                user = User.objects.get(email=...)
            except User.DoesNotExist:
                user_created = True
                user = None
                new_user = User.objects.create_user(.......)
                new_profile = Profile(user=new_user, is_email_verified=True)
                user = new_user
                if not is_moderator(user):
                    group = Group.objects.get(name="moderator")
                    user.groups.add(group)
            date_obj = datetime.strptime(workshop_data['workshop_date'])
            workshop_title_code = abbreviation(workshop_title)
            college_code =abbreviation(workshop_data['coordinator_Institute'])
            create_course(
                            workshop_data, days,
                            user=user, new_user=user_created,
                            .....)
            return HttpResponse(200)
        return Http404("Unauthorised user")
    return Http404("Unauthorised request")
```

- Create Course
  - This method creates a course based on the data received from workshop booking application.
  - It uploads the questions from the zip files such that no duplicates are uploaded.
  - The modules are designed based on the number of days described in the title of Workshop

```
def create_course(workshop_data, days, user,...):
    """ creates course for an user """
    course = Course.objects.create(...)
    # quizzes are created
    question_set = {
                    1: ['quiz_1', 'quiz_2', 'quiz_3'],
                    2: ['practice_basics', 'practice_control_flow', ..],
                    3: ['quiz_4', 'practice_functions', ....]
                }
    que = Question()
    for day in range(days):
        learning_module = LearningModule.objects.create(.......)
        for index, quiz_name in enumerate(question_set[day + 1], 0):
            quiz = Quiz.objects.create(........)
            zip_file_path = os.path.join(......... )
            files, extract_path = extract_files(zip_file_path)
            added_questions = que.read_yaml(extract_path, user, files)[1]
            question_paper = QuestionPaper.objects.create(quiz=quiz, ...)
            q_order = [str(que.id) for que in added_questions]
            question_paper.fixed_question_order = ",".join(q_order)
            question_paper.save()
            question_paper.fixed_questions.add(*added_questions)
            question_paper.update_total_marks()
            question_paper.save()
            quiz_unit = LearningUnit.objects.create(order=index, ....)
            learning_module.learning_unit.add(quiz_unit)
        course.learning_module.add(learning_module)
    send_workshop_course_mail(......................)
```

- This method is used to create an acronym for a string, here we use it to create course code from the title.

```
def abbreviation(input):
    """ returns the shorthand notation of a string """
    code = []
    for i in input.upper().split():
            code.append(i[0])
    return "".join(code)
```

Reference

- for Python built-in functions : https://docs.python.org/2/library/subprocess.html
- for Django OAuth toolkit : https://django-oauth-toolkit.readthedocs.io/en/latest/
- for Django REST API : http://www.django-rest-framework.org/