There are 2 types of users: Teacher and Student

Create API for Registration and Login.

You can use Session base Authentication or Token base Authentication

Registration API input params:

1. username : string
2. first_name: string
3. last_name: string
4. email_id: string
5. password: string
6. type: teacher or student

Login API input params:

1. username: string
2. Password: string

API to create, get, list, update and delete Course, CourseChapter and Assignment

Teacher can have access to create, get, list, update and delete for these API

Student can have access only to get, list for these API

Below is Model Schema:

```
TEACHER = 'TE'

STUDENT = 'ST'

UNKNOWN = 'UN'

TYPE_CHOICES = (

  (TEACHER, 'Teacher'),

  (STUDENT, 'Student'),

  (UNKNOWN, 'Unknown'),

)
```

UserProfile(models.Model)

```
        user = models.OneToOneField(User, on_delete=models.CASCADE)

        type = models.CharField(choices=TYPE_CHOICES, default=TYPE_UNKNOWN)
```

```
import uuid

class Course(models.Model):
    id = models.UUIDField(primary_key=True, default=uuid.uuid4)
    name = models.CharField(max_length=1000)
    description = models.CharField(blank=True, null=True, max_length=5000)
    order = models.IntegerField(default=0, blank=False, null=False)
    created_utc = models.DateTimeField(null=True, blank=True)
    modified_utc = models.DateTimeField(null=True, blank=True)



class CourseChapter(models.Model):
    id = models.UUIDField(primary_key=True, default=uuid.uuid4)
    course = models.ForeignKey('Course', related_name='course_chapter',
on_delete=models.CASCADE)
    name = models.CharField(max_length=256, null=False, blank=False)
    order = models.IntegerField(default=0, blank=False, null=False)
    created_utc = models.DateTimeField()
    modified_utc = models.DateTimeField(null=True, blank=True)



class Assignment(models.Model):
    id = models.UUIDField(primary_key=True, default=uuid.uuid4)
    title = models.CharField(max_length=1000, null=True, blank=True)
    description = models.CharField(blank=True, null=True, max_length=5000)
    course = models.ForeignKey('Course', related_name='course_assignments',
on_delete=models.CASCADE)
    course_chapter = models.ForeignKey('CourseChapter',
related_name='course_chapter_assignments', on_delete=models.CASCADE)
```