**Assignment Report**

* **Introduction:**
* Technologies used : Django, Django REST framework.
* REST framework’s SimpleJWT for User Authentication using JWT tokens.
* The project(“LMS”) contains 2 apps:
  + 1. myapp : containing endpoints for all operations related to assignments, questions and classes.
  + 2. user : containing endpoints for user registration,token generation and refresh for Json Web Tokens(JWT).
* **API design:**
* APIs have the following design:  
  - **ModelList** : contains get and post method to get a list of all model objects and create a new model object respectively.  
  - **ModelDetail :** contains get(retrieve), patch(update) and delete methods to fetch, update and delete a specific model object respectively.
* All API endpoints are secured by JWT authentication.
* **API Contracts:**
  + **GET /home/classes/   
    Returns a list of classes.**
* **URL Params:** None
* **Data Params:** None
* **Headers:**
  + Content-Type: application/json
  + Authorization: Bearer <Access Token>
* **Success Response Code:** 200
* **Response:**
  + Content Type : ‘application/json’
  + **Content :**
  + [
  + {
  + "id": 1,
  + "name": "Math-1",
  + "description": "This is a level 1 math course",
  + "teachers": [
  + 1
  + ]
  + }
  + ]
  + **POST /home/classes/   
    Creates a new class.**
* **URL Params:** None
* **Data Params:**
  + {
  + "name":"Physics-1",
  + "description":"This is level 1 phy course",
  + "teachers":[1]
  + }
* **Headers:**
  + Content-Type: application/json
  + Authorization: Bearer <Access Token>
* **Success Response Code:** 201 created
* **Response:**
  + Content Type : ‘application/json’
  + **Content :**
    - {
    - "id": 2,
    - "name": "Physics-1",
    - "description": "This is level 1 phy course",
    - "teachers": [
    - 1
    - ]
    - }
* **Error Response:**
  + **Bad Request**: Status code 400 – Invalid Input
  + **GET /home/classes/{class\_id}   
    Returns the details of the class with id=class\_id.**
* **URL Params:** *Required:* class\_id=[integer]
* **Data Params:** None
* **Headers:**
  + Content-Type: application/json
  + Authorization: Bearer <Access Token>
* **Success Response Code:** 200
* **Response:**
  + Content Type : ‘application/json’
  + **Content :**
    - {
    - "id": 2,
    - "name": "Physics-1",
    - "description": "this desc has been updated using patch api",
    - "teachers": [
    - 1
    - ]
    - }
  + **PATCH /home/classes/{class\_id}  
    Updates the class with id=class\_id.**
* **URL Params:** *Required:* class\_id=[integer]
* **Data Params:**
  + {
  + "description": "this desc has been updated using patch api"
  + }
* **Headers:**
  + Content-Type: application/json
  + Authorization: Bearer <Access Token>
* **Success Response Code:** 200
* **Response:**
  + Content Type : ‘application/json’
  + **Content :**
    - {
    - "id": 2,
    - "name": "Physics-1",
    - "description": "this desc has been updated using patch api",
    - "teachers": [
    - 1
    - ]
    - }
  + **DELETE /home/classes/ {class\_id}  
    Deletes the class object with id=class\_id.**
* **URL Params:** *Required:* class\_id=[integer]
* **Data Params:** None
* **Headers:**
  + Content-Type: application/json
  + Authorization: Bearer <Access Token>
* **Success Response Code:** 204 No content
* **Response:** (empty)
  + **\*\*OTHER APIs FOR ASSIGNMENT AND QUESTION MODEL ARE CREATED IN SIMILAR FORMATS**
  + **GET /home/assignments/**
  + **POST /home/assignments/**
  + **GET /home/assignments/{assignment\_id}**
  + **PATCH /home/assignments/{assignment\_id}**
  + **DELETE /home/assignments/{assignment\_id}**
  + **GET /home/questions/**
  + **POST /home/questions/**
  + **GET /home/question/{question\_id}**
  + **PATCH /home/question/{question\_id}**
  + **DELETE /home/question/{question\_id}**
* **User Authentication and registration:**
* Project utilizes a custom user model, initialized in a separate app - “user” containing login, register, refresh endpoints as well.
  + **POST /user/register/   
    To register a new user.**
  + **POST /user/token/   
    To generate tokens using login credentials.**
  + **POST /user/token/refresh/   
    Endpoint to generate fresh access token.**