SeeWise Django Full Stack- Assignment

# Part - I

**Create a video streaming application using django framework with the following requirements:**

1. **Account Management:**
    a. Users should be able to sign up and create an account on the platform.
    b. Once registered, users should securely log in to access all features and functionalities.
2. **Video Management:**
    a. Users should have the ability to manage their video content on the platform.
    b. This includes creating new video entries, editing existing ones, and deleting videos they no longer need.
    c. Each video entry should have a name and a corresponding video URL serving as the path to the video file.
3. **Video Streaming with OpenCV:**
    a. The platform should use the OpenCV library to provide seamless video streaming.
    b. Implement a multithreaded approach where each video is assigned a separate thread.
    c. Ensure smooth and responsive streaming, even when multiple videos are watched simultaneously.
4. **Search Functionality:**
    a. Users should be able to search for videos by their names.
    b. This feature should enhance user experience by enabling quick and easy access to specific videos of interest.
5. **Video Viewing:**
    a. A comprehensive list of all available videos should be accessible to users.

b. Users should simply click on the name of a video from the list to start watching it.
c. The interface should be intuitive, allowing users to browse through the video library effortlessly and enjoy their desired content with minimal effort.

Please provide a RESTful interface for interacting with the application, enabling users to perform various operations such as authentication, video management, searching, and profile management.

## ❖ Example

- ➢ `/api/videos` - GET and POST endpoints for creating and retrieving videos.
- ➢ `/api/videos/<id>` - GET, PUT, and DELETE endpoints for retrieving, updating, and deleting a specific video.

For authentication, use the Django built-in authentication system with token authentication.

Create a test case using the default Django test cases for each function and API.

## Evaluation:

We will evaluate your code based on the following criteria:

1. Correctness: Does the application meet the specified requirements?
2. Code quality: Is the code well-structured and easy to read?
3. Video streaming: Does the site successfully stream videos using OpenCV with multithreading?
4. API design: Is the API well-designed and easy to use?
5. Testing: Are there enough tests to ensure the functionality of the application?

SeeWise Django Full Stack- Assignment

# Part - II

**Build a Django application that accesses data from a SQLite database and computes machine efficiency using the OEE (Overall Equipment Effectiveness) formula, and provides a REST API for accessing the OEE data.:**

## 1. Model Descriptions:

a. **Machines**:

**machine_name**: Name of the machine.

**machine_serial_no**: Serial number of the machine.

**time**: Timestamp of machine creation/update.

b. **Production_log**:

**cycle_no**: Cycle number (e.g., CN001).

**unique_id**: Unique identifier for the production log entry.

**material_name**: Name of the material.

**machine**: Foreign key relationship with Machines.

**start_time**: Start time of production.

**end_time**: End time of production.

**duration**: Duration of production in hours.

SeeWise Django Full Stack-Assignment

## 2. The OEE formula is calculated as follows:

**OEE = Availability * Performance * Quality**

$$\text{AVAILABILITY} = \frac{\text{Available Time} - \text{Unplanned Downtime}}{\text{Available Time}} \times 100\ \%$$

$$\text{PERFORMANCE} = \frac{\text{Ideal Cycle Time} \times \text{Actual Output}}{\text{Available Operating Time}} \times 100\ \%$$

$$\text{QUALITY} = \frac{\text{No of good product}}{\text{Total no of product produced}} \times 100\ \%$$

## ❖ Note

- ➢ Based on the information below, kindly add a production entry to the database.
- ➢ **Available time** : Consider production company has totally 3 shift each shift has 8-hrs of Available time
- ➢ **Ideal cycle time** : Time taken to produce one part
- ➢ **Available operating time** : No of products * ideal cycle time
- ➢ **Unplanned Downtime** :( Available time - Available operating time)
- ➢ **Actual Output :** No of products produced
- ➢ **Good product :**Products produced exactly on 5 mins(Ideal cycle time)
- ➢ **Bad product :** Products produced lesser or greater than 5 mins(Ideal cycle time)
- ➢ **Total product:** Good products + Bad products

# SeeWise Django Full Stack-Assignment

**You should create a Django application that:**

a. Retrieves the data from the SQLite database and applies the OEE formula to calculate the efficiency of each machine.
b. Exposes a REST API to retrieve the OEE data.
c. Provides an endpoint to filter OEE data by Machines,Date ranges
d. Create a test case using the default Django test cases for each function and API.
e. Please push the code to GitHub and share us the repo link with clear instructions required to run the application
f. Please screen record your application and upload it to drive and share

## Evaluation

We will evaluate your code based on the following criteria:

1. Correctness: Does the application meet the specified requirements?
2. Code quality: Is the code well-structured and easy to read?
3. OEE calculation: Does the application successfully calculate the OEE values for each machine?
4. API design: Is the API well-designed and easy to use?
5. Testing: Are there enough tests to ensure the functionality of the application?