**Views**

The code defines several view functions responsible for rendering different HTML templates when users navigate to specific URLs:

* dashboard, livestream, contact, about, employee\_main, index: These functions render different HTML templates for various pages like the dashboard, live stream page, contact page, about page, employee main page, and the homepage.

A screenshot of a computer screen

Description automatically generated

**User Authentication**

There's a function user\_login which handles user authentication. It checks the submitted username and password against the user database. If the credentials are valid, the user is redirected to the main dashboard (or another specified page). Otherwise, an error message is displayed.

A computer screen shot of a program code

Description automatically generated

**Video Streaming**

The heart of the application is the stream function:

1. **Capture Video**: It captures frames from a video feed (presumably from a webcam).
2. **Object Detection**: Each frame is processed using YOLOv5 for object detection.
3. **Object Tracking**: Detected objects are then passed to DeepSORT for tracking.
4. **Annotation**: Tracked objects are annotated on the frames.
5. **Streaming**: Annotated frames are streamed to the client.

A computer screen shot of a program code

Description automatically generated

**YOLOv5 and DeepSORT Integration**

* **YOLOv5 Integration**: YOLOv5 model is loaded from a checkpoint file (best\_1.pt). It is used to detect objects in each frame of the video feed.
* **DeepSORT Integration**: DeepSORT is initialized with specific configurations. It takes the detected objects and performs object tracking.

**Model Loading and Configuration**

* The YOLOv5 model is loaded using the yolov5.load function from the yolov5 module.
* DeepSORT is initialized with specific configurations loaded from a YAML file (deep\_sort/configs/deep\_sort.yaml).
* DEEPSORT:  
   MODEL\_TYPE: "osnet\_x\_25"  
   MAX\_DIST: 1 *# The matching threshold. Samples with larger distance are considered an invalid match* MAX\_IOU\_DISTANCE: 0.7 *# Gating threshold. Associations with cost larger than this value are disregarded.* MAX\_AGE: 300 *# Maximum number of missed misses before a track is deleted* N\_INIT: 1 *# Number of frames that a track remains in initialization phase* NN\_BUDGET: 100 *# Maximum size of the appearance descriptors gallery*

**Request Handlers**

* The video\_feed function returns a StreamingHttpResponse object. This object streams the processed video frames to the client in real-time.
* def video\_feed(request):  
   return StreamingHttpResponse(stream(), content\_type='multipart/x-mixed-replace; boundary=frame')

**Other Dependencies**

* Various libraries and modules are imported and used:
  + cv2 for video processing.
  + torch for deep learning operations.
  + pathlib for file path manipulation.
  + PIL for image processing.
* from django.shortcuts import render,redirect  
  from django.http import StreamingHttpResponse  
  import yolov5,torch  
  from utils.general import (check\_img\_size, non\_max\_suppression,  
   check\_imshow, xyxy2xywh, increment\_path)  
  from yolov5.utils.torch\_utils import select\_device, time\_sync  
  from yolov5.utils.plots import Annotator, colors  
  from deep\_sort.utils.parser import get\_config  
  from deep\_sort.deep\_sort import DeepSort  
  import cv2  
  from django.contrib.auth import authenticate, login  
  import pathlib

**Additional Notes**

* There's a modification of pathlib.PosixPath to pathlib.WindowsPath which seems specific to Windows operating system environment.
* The YOLOv5 model is configured with specific confidence (conf) and intersection over union (iou) thresholds, as well as the classes of objects to detect.
* The code handles the case where no object is detected in a frame by incrementing the ages of objects being tracked by DeepSORT.

In summary, this code sets up a Django web application with user authentication and a live video stream feature that performs real-time object detection and tracking using YOLOv5 and DeepSORT. It provides a seamless interface for users to access and interact with the video stream.

**Admin Panel**

The Django admin panel serves as an interface for managing the data stored in your Django application's database. It offers CRUD (Create, Read, Update, Delete) operations for your application's models without requiring you to write custom views or forms.

**Key Features:**

1. **Authentication and Authorization:**
   * The admin panel requires authentication, and users must log in with valid credentials.
   * Django provides a robust authentication system out of the box, including support for creating superusers with administrative privileges.
2. **User Management:**
   * Superusers can manage user accounts from the admin panel, including creating, editing, and deleting user accounts.
   * User authentication and permissions can be controlled through the admin interface, allowing administrators to grant or revoke access to specific users or groups.
3. **Model Management:**
   * The admin panel automatically generates forms based on your application's models, allowing you to manage data without writing custom HTML forms.
   * Administrators can view, add, edit, and delete records for each model registered with the admin interface.
   * Inline editing and adding related objects are supported for models with foreign key or many-to-many relationships.
4. **Customization:**
   * The admin interface can be customized to suit the needs of your application.
   * You can customize the appearance of the admin interface using CSS.
   * Model admin classes allow you to customize the behavior of the admin interface for each model, such as specifying fieldsets, list displays, search fields, filters, and more.
   * Custom admin actions can be defined to perform batch operations on selected objects.
5. **Security:**
   * Django's admin panel is designed with security in mind, protecting against common vulnerabilities such as cross-site scripting (XSS), cross-site request forgery (CSRF), and SQL injection.
   * Superusers have access to sensitive features, such as managing user accounts and modifying data, so it's essential to restrict access to the admin panel to authorized personnel only.
6. **Internationalization (i18n) and Localization (l10n):**
   * The admin panel supports internationalization and localization, allowing administrators to manage content in multiple languages.
   * Translations for built-in components of the admin interface are available in several languages, and custom translations can be added for application-specific content.