

# Django Training - 4. Class-based Views, User class, Authentication

---

## Materials

---

1. [User Authentication](#)
2. [Class-based views](#)
3. [Django ImageKit](#)
4. [Static Files](#)
5. [Singals](#)

## Task

---

1. Change all the current views you have to class based views, from now on we'll only be creating class based views
2. Add a sign in page using which a user can provide their username and password to get authenticated
3. Allow unauthenticated users to access the endpoint `https://localhost:8000/artists/`, but only authenticated users can access the creation form pages

## Refresher Task

---

1. Remove the album inline from the artist admin form and make sure you've added the album model to django admin.
2. We received a requirement that each album **must have at least one song**. In the `albums` app, create a song model that consists of:
  - A name (if no name is provided, the song's name defaults to the album name)
  - An image (required)
  - An image thumbnail with JPEG format (hint: use `ImageKit`)
    - Do you think this field is useful? share with me your answer to this question whether you agree or disagree
  - An audio file with `.mp3` or `.wav` extensions (required)
  - Setup your server to serve the uploaded media files, for example, I should be able to view a song's image by accessing its url: `http://127.0.0.1:8000/YOUR_MEDIA_PATH/image.jpg`
  - You should add the directory where the images and audio files are stored to `.gitignore` because user uploaded media isn't part of the codebase
  - We know that `models.ForeignKey` achieves a one-to-many relationship, so if we create a model `Song` with a foreign key to `Album`, we'll guarantee that any album instance has 0 or more songs, but we want to enforce a 1-or-more relationship as much as we can, at least on django admin so that any admin can't create an album without first uploading a song, how can this be done?
    - hint: You'll need to perform custom formset validation
    - hint: Don't forget to handle the delete case, an admin user shouldn't be able to delete all songs of an album

## Guidelines

---

1. Use this gitignore [template](#)
2. List your environment variables (if any) in a `.env.example` file
3. Don't forget to run `manage.py makemigrations` and `python manage.py migrate`
4. You are allowed to refer to any articles, documentation source, questions on StackOverFlow
5. You are not allowed to consult other people through any medium of communication
6. Your grade will be affected if your code isn't clean, you should maintain code cleanliness as much as you can