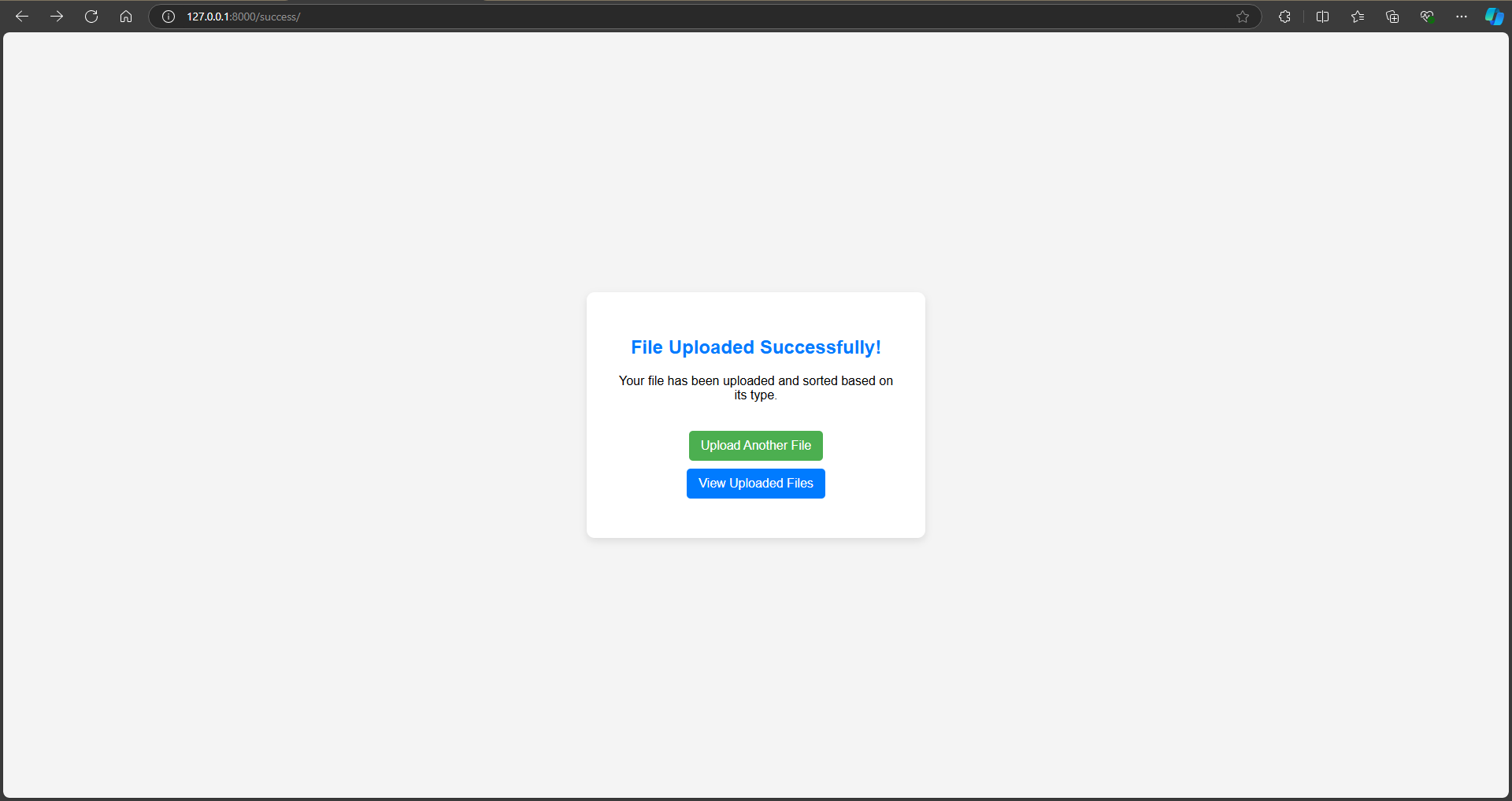
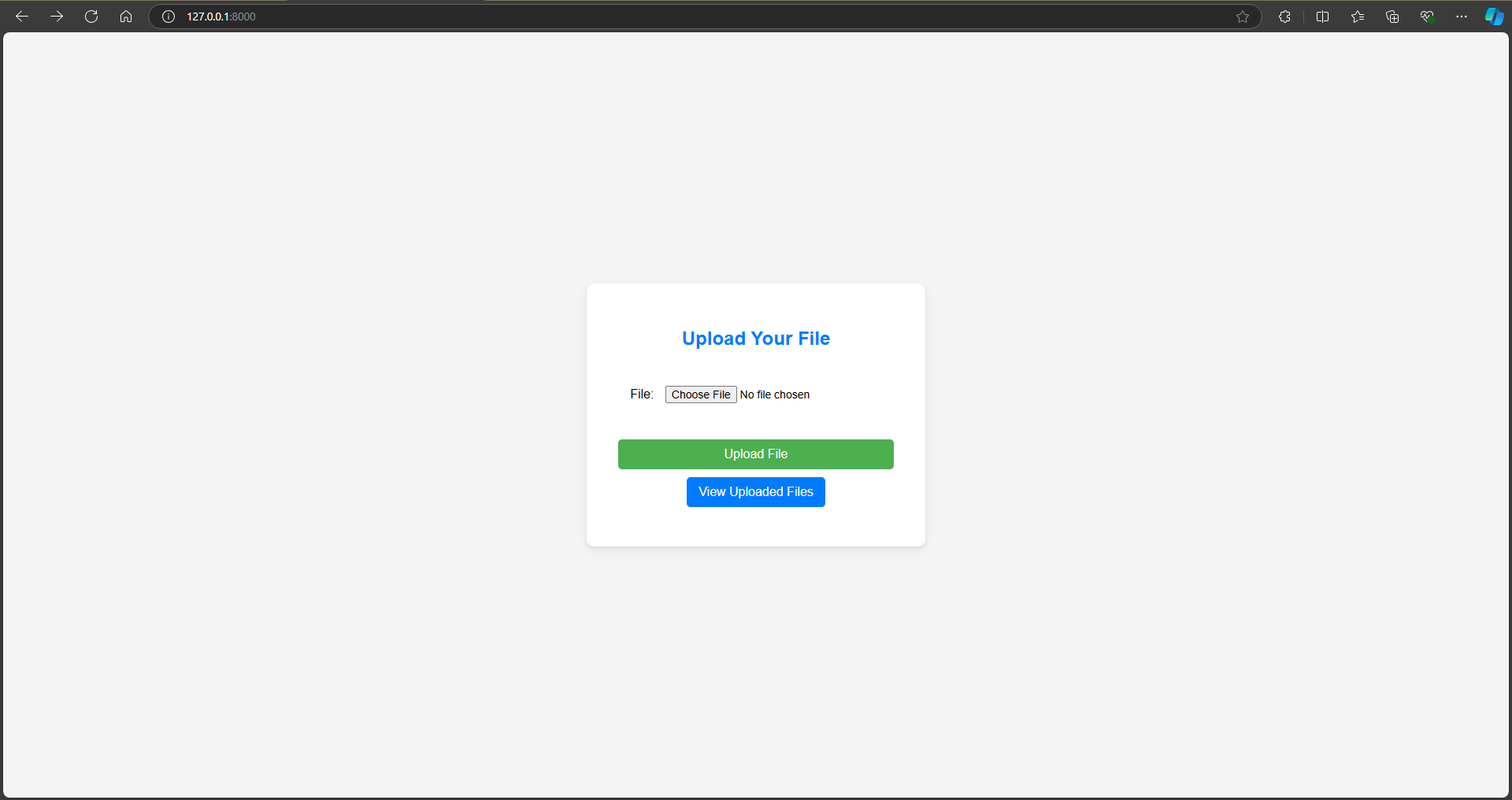
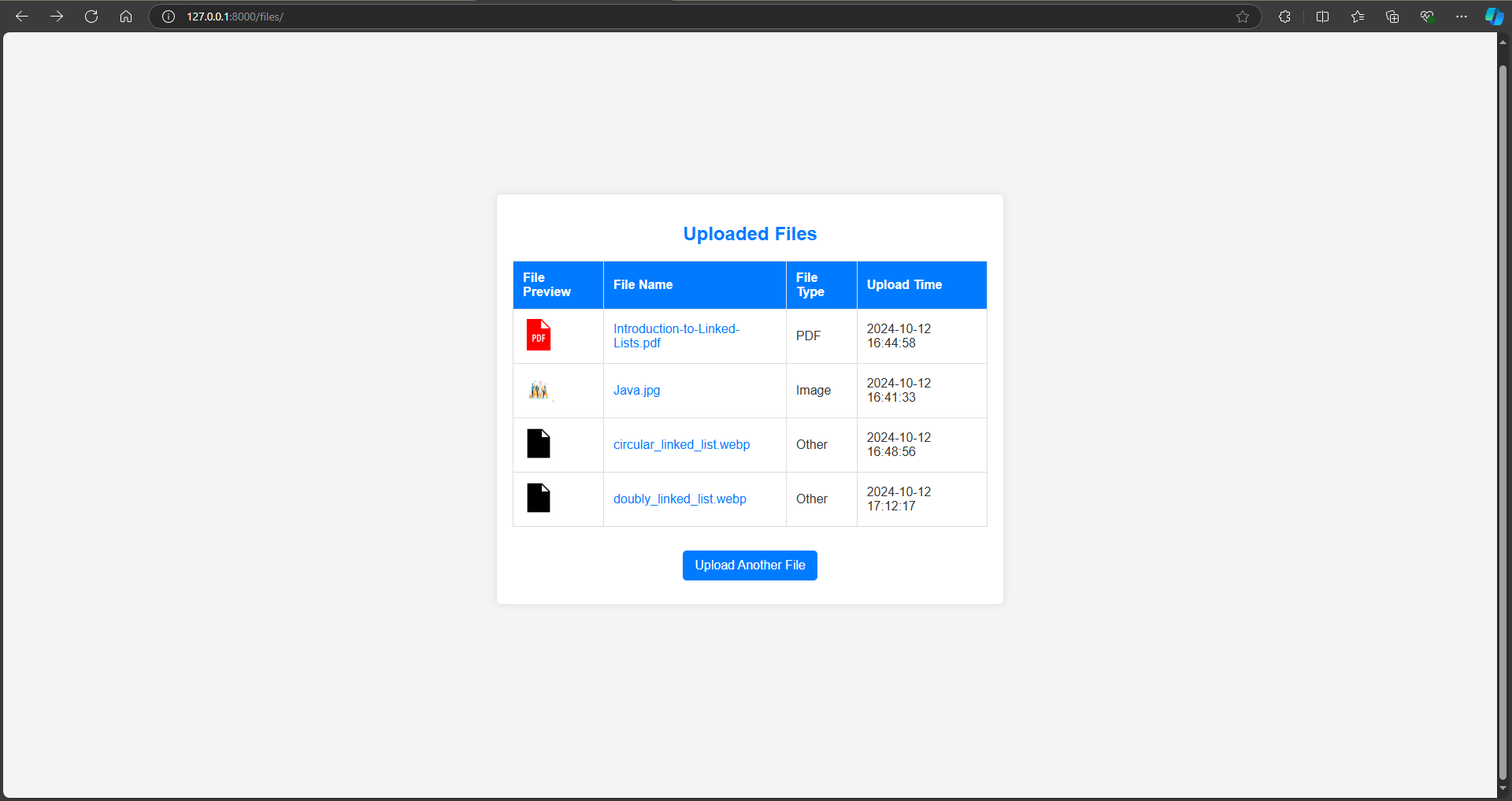
**Build a Model :**

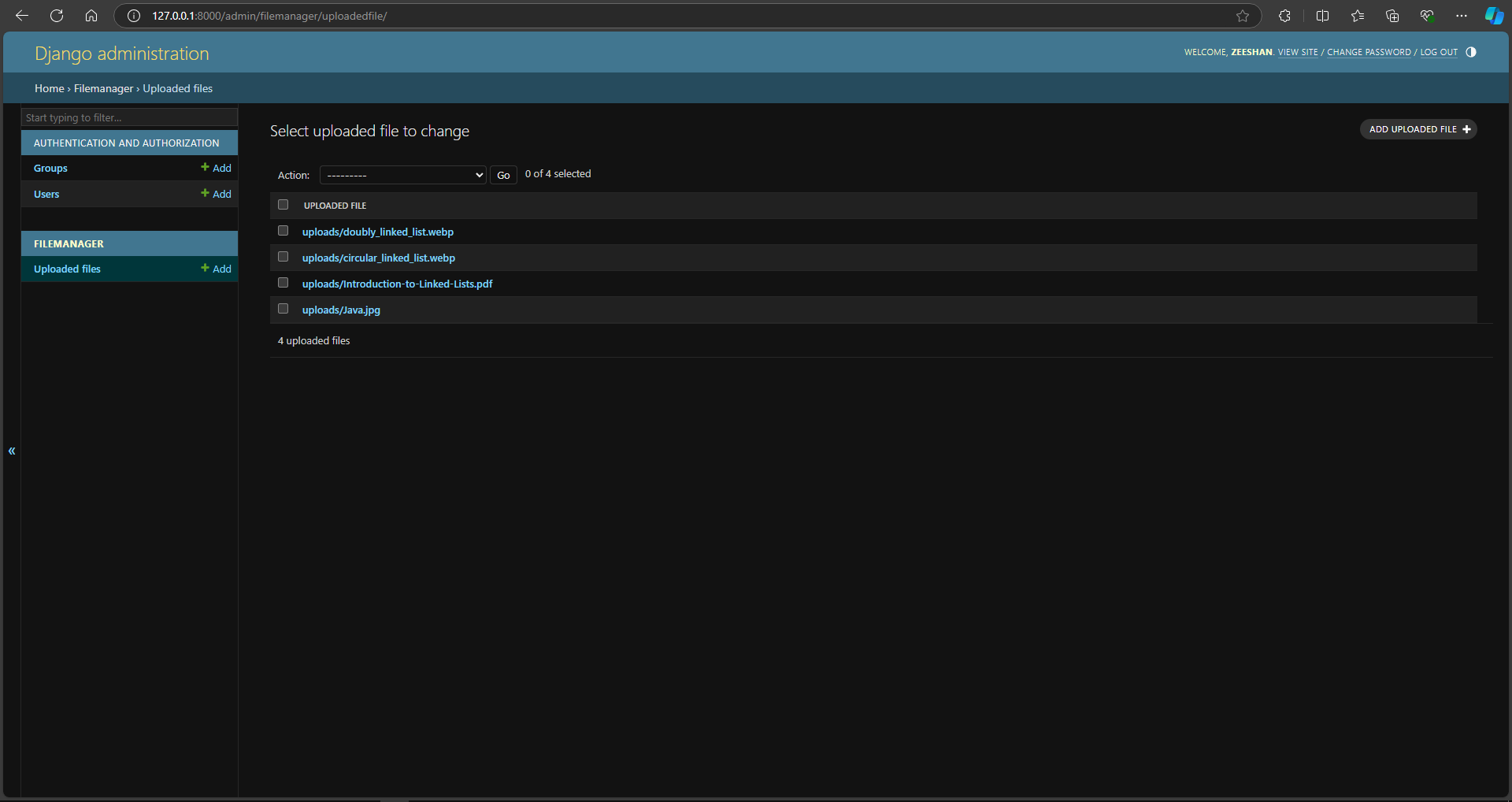
1. **Task Description**

The application allows users to upload files, which are then categorized based on their types (PDFs, images, and other file types). Additionally, the system displays a list of uploaded files along with relevant metadata.

1. **Task Output Screenshot**







## Project Setup

### Starting the Project-

I began by setting up a new Django project and application. I configured the project settings to manage media files (the files users upload) and static files (like images and styles). This included specifying where the files would be stored on the server.

### Creating the Model-

Next, I created a model called UploadedFile. This model helps manage the files that users upload. It includes a field for the uploaded file and a timestamp to record when each file was uploaded. This makes it easy to keep track of all uploaded files.

## User Interface Development

### File Upload Form-

To make it easy for users to upload files, I designed a simple HTML form. This form allows users to select a file from their computer and submit it. I used CSS to make the form look nice and to center it on the page.

### Success Page-

After a user uploads a file successfully, they are taken to a confirmation page. This page thanks them for their upload and provides a link to view all uploaded files, guiding them to the next step.

### Displaying Uploaded Files-

I created a page to display all uploaded files in a table format. Each row in the table shows important details about the files, including:

* **File Preview**: A small image or icon that represents the file type.
* **File Name**: The name of the uploaded file without any folder paths.
* **File Type**: The category of the file (like PDF, image, or others).
* **Upload Time**: The date and time when the file was uploaded, shown in the user’s local timezone.

### Icons and Previews-

To make the file list more visually appealing, I added icons for different file types. This helps users quickly see what type of file they uploaded. For image files, I displayed a small thumbnail so users can see a preview of the image.

## Functionality Improvements

### Categorizing Uploaded Files-

I added functionality to automatically categorize the uploaded files into three groups:

* **PDFs**: Files with a .pdf extension.
* **Images**: Files with .jpg or .png extensions.
* **Others**: Any other file types that don’t fit into the first two categories.

This helps keep everything organized.

### Admin Interface-

I enabled the Django admin interface so that I (and any admin users) could upload files directly through the admin panel. This means all uploaded files, whether through the user interface or the admin panel, are displayed in the same file list.

### Managing Time Zones-

To ensure that the upload times are displayed correctly, I set up the timezone settings in the project. This means the application shows the correct upload times based on the user’s local timezone.

## Conclusion

Overall, I created a file upload management system that is functional and easy to use. The system allows users to upload files, see a list of their uploads, and view important details about each file, all in a user-friendly way.