

# Project Explanation

This document provides an overview of the code in the `views.py` file of our Django project. The code in this file is responsible for handling various API endpoints related to user registration, login, product management, and pagination.

## User Registration register

Endpoint: register

- This view allows new users to register by providing their first name, last name, username, and password.
- The provided information is validated, and a new user is created using Django's built-in `User` model.
- Successful registration results in a JSON response with a success message.

## User Login user\_login

Endpoint: login

- Users can log in by providing their username and password.
- The provided credentials are authenticated using Django's authenticate method.
- If authentication is successful, the user is logged in, and a success message is returned in the response.
- In case of failed authentication, an error message is returned with a status code.

## CSV File Upload upload\_product\_csv

Endpoint: upload\_product\_csv

- Admin users can upload CSV files containing product data.
- The uploaded CSV file is processed using the `pandas` library to read and handle the data.
- The code expects the CSV file to be submitted as a part of the POST request.
- If the user is not an admin or if no CSV file is provided, appropriate error messages are returned.
- Successful processing of the CSV file results in a success message.

## Product Review Submission add\_product\_review

Endpoint: add\_product\_review

- Users can submit product reviews along with ratings.
- The review data, including the product ID, review text, and rating, is expected in a POST request.
- The code handles validation, associates the review with the relevant product and user, and returns a success message upon successful submission.

## **Product Pagination and Sorting `paginate_products`**

Endpoint: `paginate_products`

- This view provides product pagination and sorting based on reviews.
- Users can specify the page number and sorting criteria (default is by rating) in the GET request parameters.
- The code fetches products from the database, applies to sort, and paginates the results using Django's `Paginator`.
- The paginated products are serialized and returned in a JSON response, along with information about the total number of pages and the current page number.

This document serves as a high-level explanation of the functionality implemented in the `views.py` file. It is essential to refer to this documentation when working with the API endpoints provided by this Django project.