Crest Infosystems Pvt. Ltd

#411-414, Wood Square, Opp. TGB Restaurant, L. P. Savani Road, Adajan, Surat – 395009, Gujarat, India.

Python Practical Task

Kindly review the following requirements and implement a demo.
Technology: Python
Duration : 3 hours

Objective: Develop APIs using the Django Framework to perform CRUD operations and implement an authentication system with role-based access control.

Requirements:

Functional Requirements

CRUD Operations for Product Table:

1. Create
    - Implement an API to create a new product.
    - Required fields: `title`, `description`, `price`, `image`, `ssn`, `discount`.
    - Optional fields: None.
2. Update
    - Implement an API to update an existing product by its ID.
    - Ensure partial updates (PATCH) and full updates (PUT) are supported.
3. Delete
    - Implement an API to delete a product by its ID.
    - Soft delete functionality: Set `is_active` to `False` instead of removing the record from the database.
4. View
    - Implement an API to list all products.
    - Support pagination.
    - Add filtering by `title`, `description`, and `price_range` (ascending and descending).
    - Allow sorting by `created_on` and `updated_on`.
5. Disable
    - Implement an API to disable a product by its ID.
    - Set `is_active` to `False`.

Authentication System:

1. **Token-Based Authentication**
   - **Use JWT (JSON Web Token) for authentication.**
   - **Ensure all CRUD operations require a valid token for access.**
2. **Login and Registration**
   - **Implement APIs for user registration and login.**
   - **Return a JWT upon successful login.**
3. **Role-Based Access Control**
   - **Define two roles: `admin` and `user`.**
   - **Permissions:**
     - **`admin`: Full access to all CRUD operations.**
     - **`user`: Read-only access (can only view products).**

Export functionality:

- **Provide an API endpoint with which the user can export all products in an excel sheet.**

Database Table:

Product

- `title`
- `description`
- `price`
- `discount`
- `image`
- `ssn`
- `is_active`
- `created_on`
- `updated_on`

[Note: Create other tables as per the requirement]

Bonus Points:

1. **Implement a search endpoint for products, allowing keyword search across `title` and `description`.**
2. **Add support for bulk product creation.**
3. **Use Django Signals to log changes in the `Product` table.**
4. **Implement rate-limiting for all endpoints.**

Submission Guidelines:

- **Provide the source code in a GitHub repository.**
- **Include detailed instructions for setting up and running the project locally**

- **Ensure code quality and adhere to PEP 8 standards.**

We believe in growing together