Paribus

Paribus Python Challenge

Hi! We're very excited that you're interviewing at Paribus. We created this task so that we can evaluate your ability to learn and code. We hope that you learn something new working on this. Good luck! ☺️

Task: Hospital Bulk Processing System

You have been given access to a deployed Hospital Directory API that manages individual hospital records. Your task is to build a separate bulk processing system that integrates with this existing API to handle CSV uploads and process hospital records.

Estimated Completion Time: 3-5 days

Hospital Directory API (Given)

We have deployed a Hospital Directory API for you to use. This API handles individual hospital operations and includes batch management features specifically designed for bulk processing scenarios.

Base URL: `https://hospital-directory.onrender.com/docs`

Key Features:

- Individual hospital CRUD operations
- Batch processing support with unique batch IDs
- Hospitals created with a batch ID are marked as inactive
- Batch activation to make all hospitals in a batch active simultaneously

Available Endpoints:

- `POST /hospitals/` - Create individual hospital
- `GET /hospitals/` - Get all hospitals
- `GET /hospitals/batch/{batch_id}` - Get hospitals by batch ID
- `PATCH /hospitals/batch/{batch_id}/activate` - Activate all hospitals in batch
- `DELETE /hospitals/batch/{batch_id}` - Delete all hospitals in batch

Hospital Model:

```json
{
  "id": 123,
  "name": "Hospital Name",
  "address": "123 Main St",
  "phone": "555-1234",
  "creation_batch_id": "uuid",
  "active": false,
  "created_at": "2025-09-19T10:30:00Z"
}
```

Functional Requirements

Your bulk processing API should support the following operations:

- **Bulk Create Hospitals:**
  - **Endpoint:** `POST /hospitals/bulk`
  - **Request:** Multipart form data with CSV file
  - **CSV Format:** `name,address,phone` (phone is optional)
  - **Response:** Processing status with batch ID and progress information

Processing Workflow:

1. Accept and validate CSV file upload
2. Generate unique batch ID (UUID)
3. Make HTTP calls to `POST /hospitals/` for each hospital with the batch ID
4. Once all hospitals are created successfully, call `PATCH /hospitals/batch/{batch_id}/activate`
5. Return comprehensive processing results

Expected Response:

```
{
  "batch_id": "550e8400-e29b-41d4-a716-446655440000",
  "total_hospitals": 25,
  "processed_hospitals": 25,
  "failed_hospitals": 0,
  "processing_time_seconds": 250,
  "batch_activated": true,
  "hospitals": [
    {
      "row": 1,
      "hospital_id": 101,
      "name": "General Hospital",
      "status": "created_and_activated"
    }
  ]
}
```

Technical Constraints

- Maximum CSV size: 20 hospitals

Expected Tech Stack

- **Language:** Python 3.8+
- **Web Framework:** FastAPI or Flask (preferred for its minimalism)
- **Data Persistence:** In-memory storage is acceptable
- **Deployment:** Deploy the app online. Suggested platforms: Render

Optional Tasks (Bonus Points)

If you have extra time or want to showcase additional skills, consider implementing:

- **Performance Optimization**
- **Progress Tracking:** Real-time progress updates via WebSocket or polling endpoint
- **Resume Capability:** Ability to resume failed bulk operations
- **CSV Validation:** Endpoint to validate CSV format before processing
- **Comprehensive Testing:** Unit tests, integration tests, and error scenarios

- **Dockerization:** Provide a `Dockerfile` and `docker-compose.yml` to easily run your application in a Docker container

Evaluation Rubric

Your submission will be evaluated based on the following criteria:

- **System Design (25%)**
- **Functionality (20%)**
- **Performance & Scalability (25%)**
- **Code Quality (20%)**
- **Documentation & Testing (10%)**

Submission Instructions

Please include the below in your assignment submission:

- A Git repository (e.g., GitHub, GitLab, Bitbucket) including all source code
- A public URL for the hosted application

Good luck, and we look forward to reviewing your work!