**Problem Statement:**

You are tasked with creating an intelligent book management system using Python, a locally running Llama3 generative AI model, and cloud infrastructure. The system should allow users to add, retrieve, update, and delete books from a PostgreSQL database, generate summaries for books using the Llama3 model, and provide book recommendations based on user preferences. Additionally, the system should manage user reviews and generate rating and review summaries for books. The system should be accessible via a RESTful API and deployed on cloud.

**Requirements:**

1. **Database Setup:**
   * Use PostgreSQL to store book information.
   * Create a books table with the following fields: id, title, author, genre, year\_published, summary.
   * Create a reviews table with the following fields: id, book\_id (foreign key referencing books), user\_id, review\_text, rating.
2. **Llama3 Model Integration: (You can use O Llama 3 8GB Model/Hugging Face Model/ Groq OLAMA Hugging face etc.)**
   * Set up a locally running Llama3 generative AI model to generate summaries for books based on their content.
   * Integrate the Llama3 model to generate summaries for new book entries and review summaries for each book.
3. **RESTful API:**
   * Develop a RESTful API with the following endpoints:
     + POST /books: Add a new book.
     + GET /books: Retrieve all books.
     + GET /books/<id>: Retrieve a specific book by its ID.
     + PUT /books/<id>: Update a book's information by its ID.
     + DELETE /books/<id>: Delete a book by its ID.
     + POST /books/<id>/reviews: Add a review for a book.
     + GET /books/<id>/reviews: Retrieve all reviews for a book.
     + GET /books/<id>/summary: Get a summary and aggregated rating for a book.
     + GET /recommendations: Get book recommendations based on user preferences.
     + POST /generate-summary: Generate a summary for a given book content.
4. **Asynchronous Programming:**
   * Implement asynchronous operations for database interactions and AI model predictions using sqlalchemy[asyncio] and asyncpg.
5. **Cloud Deployment:**

Kindly push the code in Github/or create docker files or writes instructions for CI/CD workflow or setup the infrastructure workflow by preparing a deployment document and sharing it as Read Me file in GitHub repo.

* + Deploy the application on cloud using services such as EC2, Lambda, or ECS.
  + Ensure the database is hosted on AWS RDS.
  + Use AWS S3 for storing any model files if necessary.
  + Set up a CI/CD pipeline for automatic deployment.

1. **Authentication and Security:**
   * Implement basic authentication for the API.
   * Ensure secure communication with the database and API endpoints.

**Bonus:**

* Implement caching for the book recommendations using AWS ElastiCache.
* Add unit and integration tests for the API endpoints and AI model.
* Use AWS Sage Maker for deploying and managing the machine learning model.

**Instructions:**

1. **Database Schema:**
   * Define the schema for the books and reviews tables in PostgreSQL.
2. **Llama3 Integration: (You can use O Llama 3 8GB Model/Hugging Face/ Groq OLAMA Hugging face etc.)**
   * Set up a local instance of the Llama3 model for generating book and review summaries.
   * Develop an endpoint to generate summaries for new book entries and review summaries.
3. **API Development:**
   * Develop the RESTful API using Fast API or Flask.
   * Implement the necessary asynchronous operations for database and AI model interactions.
4. **Deployment on AWS or Azure or GCP:**

OPTION 1- Kindly push the code in Github/or create docker files or writes instructions for CI/CD workflow or setup the infrastructure workflow by preparing a deployment document and sharing it.

OPTION 2-

* + Set up an AWS account and use AWS services to deploy your application.
  + Ensure the API is accessible over the internet and securely connected to the PostgreSQL database.

1. **Testing and Documentation:**
   * Write unit tests for your API endpoints.
   * Document your API using Swagger or similar tools.
   * Provide clear instructions on how to set up and run your application.

**Deliverables:**

* Source code for the application, including the Llama3 integration, machine learning model, API, and database schema.
* Documentation on how to deploy and use the application.
* A link to the deployed application on cloud.