BIG DATA TERM PROJECT

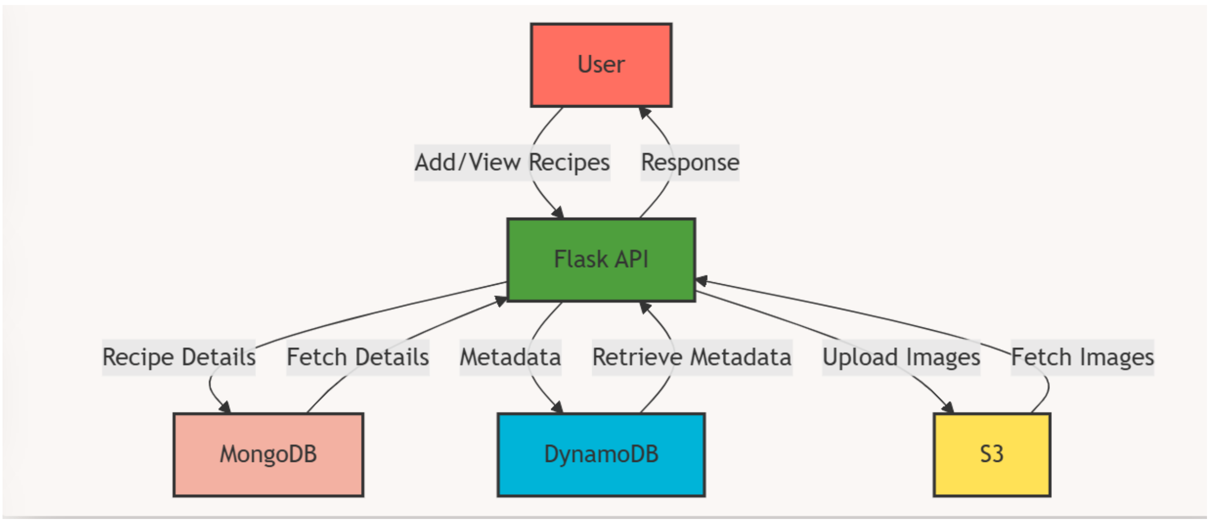
RECIPE MANAGEMENT SYSTEM

1. Project Description: Recipe Management System with Big Data Integration

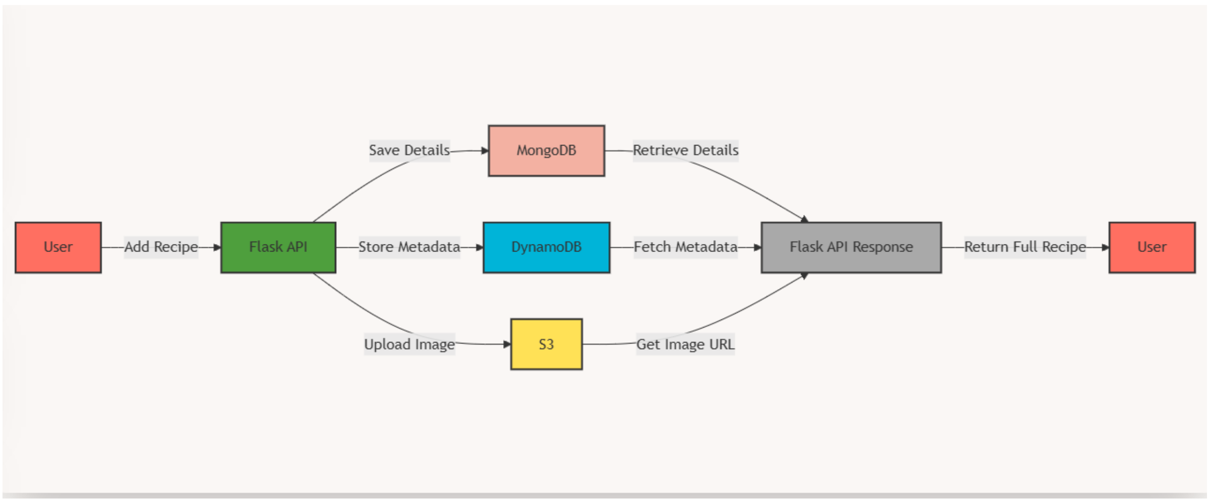
Overview: The **Recipe Management System** is designed to allow users to upload, view, and interact with recipes. The system handles different aspects of recipe management, such as storing recipe details, images, and metadata, while leveraging cloud services and big data tools for scalability and performance. It integrates several technologies like **AWS S3**, **DynamoDB**, **MongoDB**, and **Flask**, creating an efficient and responsive solution for users to manage a large dataset of recipes.

Key features:

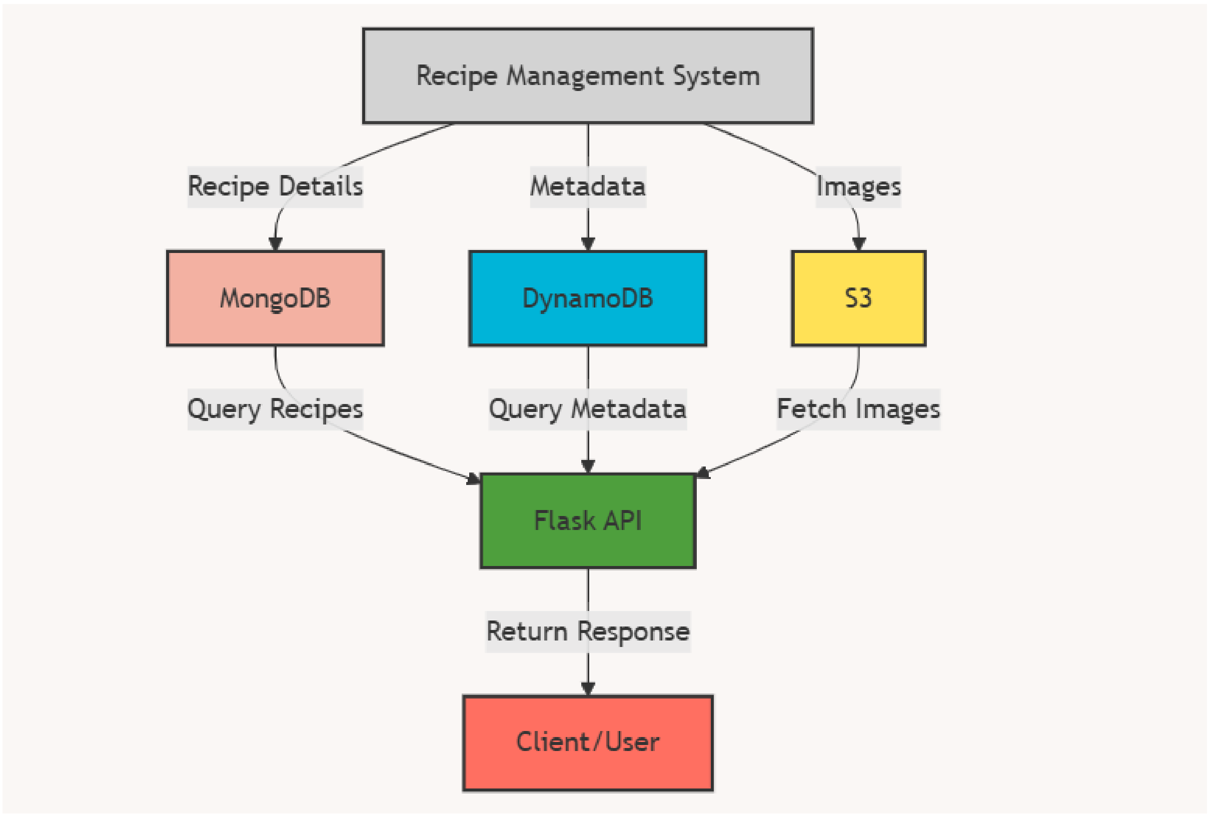
- **Add Recipe**: Users can add new recipes, including name, ingredients, instructions, and images.
- **View Recipes**: Users can fetch and view recipes from the system.
- **Upload Images**: Recipes can have associated images uploaded to Amazon S3.
- **Interaction Tracking**: Actions such as viewing or liking a recipe are logged and stored in MongoDB.
- **Metadata Storage**: Recipe metadata (such as views, likes, and tags) is stored in DynamoDB.



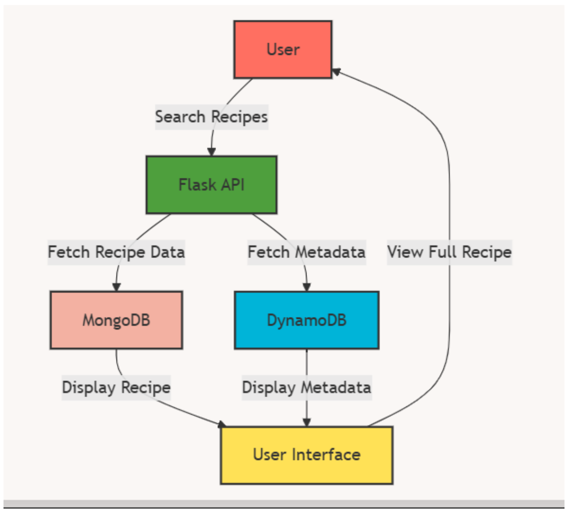
This diagram demonstrates the interaction between the **User**, **Flask API**, **MongoDB**, **DynamoDB**, and **Amazon S3** for adding/viewing recipes, uploading images, and handling metadata.



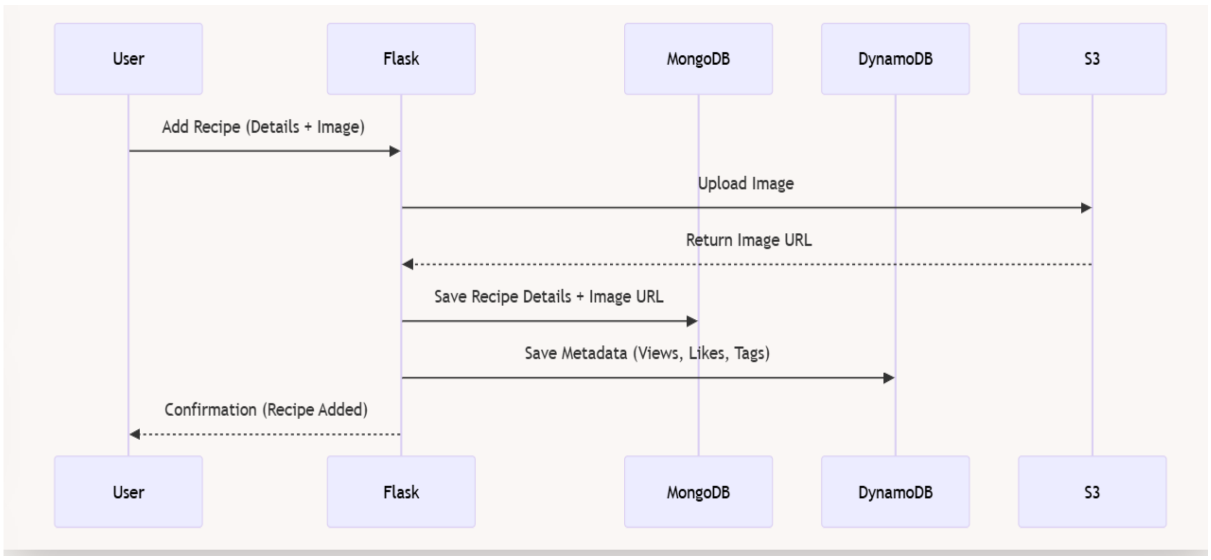
This flowchart shows how **recipes** are added, images are uploaded to **S3**, and metadata is stored and retrieved in **MongoDB** and **DynamoDB**, followed by a **response** to the **User**.



This diagram represents the **architecture** of the system, which utilizes **MongoDB**, **DynamoDB**, and **S3** for data storage, and **Flask API** for handling interactions with the **User**.



This diagram represents the **user interaction flow**, from searching and fetching recipes to viewing the full recipe and interacting with metadata like **views** and **likes**.


User
Flask
MongoDB
DynamoDB
S3
Add Recipe (Details + Image)
Upload Image
Return Image URL
Save Recipe Details + Image URL
Save Metadata (Views, Likes, Tags)
Confirmation (Recipe Added)

This sequence diagram shows the steps involved in **adding a recipe**, from uploading the image to saving the recipe details in **MongoDB**, **DynamoDB**, and **S3**, and finally confirming the addition to the **User**.

---

2. Design Challenges

- **Handling Large Volumes of Data**: Managing large datasets (e.g., images, recipe details, and metadata) poses a challenge. Efficient data storage and retrieval are critical, especially when scaling to millions of recipes and interactions.
- **Data Consistency**: Ensuring that the data in multiple databases (MongoDB, DynamoDB, and S3) remains consistent and up-to-date while minimizing latency and handling concurrency.

- **Cloud Integration**: Integrating with cloud services like S3 for file storage and DynamoDB for fast metadata retrieval requires careful configuration, including endpoints, bucket naming, and access management.
- **Scalability**: The system needs to be scalable to handle an increase in user requests, file uploads, and recipe entries over time. The architecture must support horizontal scaling.
- **Performance Optimization**: Minimizing query latency, particularly when retrieving large datasets from multiple sources (e.g., images from S3 and metadata from DynamoDB), was a key challenge.

---

3. Big Data Architecture and Tools Used in the Project

Architecture Overview: The architecture is designed to ensure scalability, availability, and fault tolerance. It leverages **cloud-native services** and **distributed systems** to handle the complexity of storing and managing large amounts of data in real-time.

1. **MongoDB (for Recipe Details)**:
   - MongoDB is used for storing recipe details (name, ingredients, instructions, etc.). It is a NoSQL database that is highly scalable and flexible for storing data in JSON-like documents.
   - **Rationale**: MongoDB's document-oriented structure is suitable for the semi-structured data of recipes. It allows quick and efficient retrieval of data, which is essential for recipes that may have varying ingredients, instructions, or metadata.

Example: Storing each recipe as a document in MongoDB, where each document contains details like ingredients and instructions.

2. **Amazon S3 (for Image Storage)**:
   - Amazon S3 is used for storing images associated with recipes. This cloud-based object storage service is highly scalable and designed to store large files.
   - **Rationale**: S3 allows the system to offload image storage, ensuring that the main database (MongoDB) is not overloaded with large files. It also supports high availability and durability.

Example: Images uploaded through the system are stored in an S3 bucket, and their URLs are saved in MongoDB to link images with recipe data.

3. **Amazon DynamoDB (for Metadata Storage)**:
   - DynamoDB is used for storing metadata such as views, likes, and tags related to each recipe. It is a managed NoSQL database that offers fast performance at scale.
   - **Rationale**: DynamoDB's ability to handle large amounts of read/write operations with low latency makes it ideal for storing and retrieving metadata quickly.

Example: For each recipe, metadata such as views, likes, and tags are stored in DynamoDB. This allows for quick retrieval of recipe metadata without impacting the performance of the main MongoDB database.

fetch

```
curl -X GET http://127.0.0.1:5000/get_recipes
```

```
python -m awscli --endpoint-url=http://localhost:4566 --region us-east-1 dynamodb get-item --table-
name recipe_metadata --key "{\"recipe_id\": {\"S\": \"1\"}}"
```

```
python -m awscli --endpoint-url=http://localhost:4566 --region us-east-1 s3 ls s3://my-bucket/
```

```
python -m awscli --endpoint-url=http://localhost:4566 --region us-east-1 s3 ls s3://recipe-files/
```

add

```
curl -X POST http://127.0.0.1:5000/add_recipe -H "Content-Type: application/json" -d "{\"recipe_id\":
\"2\", \"name\": \"Spaghetti Bolognese\", \"ingredients\": \"Spaghetti, Ground Beef, Tomato Sauce,
Onion, Garlic, Olive Oil, Salt, Pepper\", \"instructions\": \"Cook spaghetti. Prepare sauce with beef,
onions, garlic, and tomatoes. Combine.\" , \"image_url\": \"s3://my-bucket/spaghetti-bolognese.jpg\"}"
```

```
python -m awscli --endpoint-url=http://localhost:4566 --region us-east-1 dynamodb put-item --table-
name recipe_metadata --item "{\"recipe_id\": {\"S\": \"1\"}, \"views\": {\"N\": \"10\"}, \"likes\": {\"N\": \"5\"},
\"tags\": {\"L\": [{\"S\": \"Italian\"}, {\"S\": \"Pasta\"}]} }"
```

```
curl -X POST -F "file=@C:/Users/Lenovo/Downloads/spicy-spaghetti-arrabbiata.jpg"
http://127.0.0.1:5000/upload_file
```