Assignment Documentation

1. Flask Application (`app.py`)

Overview

The Flask application (`app.py`) is designed to manage expenses among users, allowing them to add expenses and track amounts owed between friends.

Features

1. **User Management**
    - Endpoint: **`/add_user`**
    - Allows adding new users to the system.
    - Checks for existing users before adding.
2. **Expense Management**
    - Endpoint: **`/add_expense`**
    - Allows users to add expenses, specifying the payer, amount, and shares among friends.
    - Validates payer existence and friend IDs before adding the expense.
3. **Amount Calculation**
    - Endpoints: **`/amount_owed_by_friends/<user_id>`** **and** **`/amount_owed_to_friends/<user_id>`**
    - Calculates amounts owed by a user to friends (`amount_owed_to_friends`) and amounts owed by friends to a user (`amount_owed_by_friends`).

Implementation Details

- **Logging**: Utilizes logging to record key events such as user additions, expense additions, and errors (e.g., user not found).
- **Data Structures**: Uses dictionaries (`users` and `expenses`) to store user and expense data in memory.
- **Error Handling**: Implements error handling for invalid requests (e.g., existing user, non-existing user).
- **Integration with MongoDB**: Demonstrates integration with MongoDB via `mongoengine` for persistent storage (not implemented in this example but recommended for real-world scenarios).

Submitted By: Utkarsh Kuchhal

Proposed Improvements

1. **Cache Integration with Redis**
    - Implement Redis caching to store frequently accessed user data and calculations.
    - Benefits: Improves response times by reducing database queries for repeated requests.
2. **Optimization Using Graph Algorithms**
    - Apply graph algorithms (e.g., cycle detection, shortest path) to minimize cash flow among a group of friends.
    - Benefits: Reduces the number of transactions needed to settle debts, optimizing cash flow and reducing complexity.

2. MongoDB Models (`models.py`)

Overview

The `models.py` file defines MongoDB models using `mongoengine` for user and expense management.

Models

1. **User Model**
    - Attributes: `user_id`, `expenses_paid`, `shares_received`
    - Represents a user with attributes to track expenses paid and shares received.
2. **Expense Model**
    - Attributes: `payer_id`, `amount`, `shares`
    - Represents an expense with attributes for the payer, amount paid, and shares among friends.