Documentation

A. Setup Instructions

1. **Install Dependencies**:
   Ensure you have **Node.js** and **npm** installed. Then, install dependencies:
   ```
   npm install
   ```
2. **Create `.env` file**:
   Add your MongoDB connection URI and Redis configuration:
   ```
   MONGO_URI=mongodb://localhost:27017/wellnessTracker
   REDIS_HOST=localhost
   REDIS_PORT=6379
   ```

B. Run the Service:

1. Start the backend service:
   ```
   node server.js
   ```

C. API Usage

1. **Log Wellness Activity**: **Endpoint**: `POST /activity`

Request Body:

```
{
  "user_id": "1234",
  "date": "2023-08-01",
  "hydration_liters": 2.0,
  "sleep_hours": 7.5,
  "exercise_minutes": 30,
  "meditation_minutes": 10,
  "source": "manual"
}
```

Response: A JSON object of the newly logged activity.

2. **Get Activity Logs for a User**:
   - **Endpoint**: `GET /activity/:userId?start=YYYY-MM-DD&end=YYYY-MM-DD`
   - **Example**: `GET /activity/1234?start=2023-08-01&end=2023-08-07`
   - **Response**: A JSON array of the user's activity logs within the date range.

C. Assumptions Made

- User data is identified by `user_id`.
- The date format for activity logs is `YYYY-MM-DD`.
- Redis stores cached data indefinitely (no TTL).
- The system doesn't have authentication or authorization mechanisms (this can be added if needed).

Explanation of Tracking Logic

1. **Logging Activities**:
    - When a user logs wellness data, the backend stores the data in MongoDB.
    - If there is a valid source, it is stored as either `'manual'` or `'device'`.
2. **Caching**:
    - **Cache Miss**: When the `GET` request is made, Redis is queried for cached activity logs.
    - **Cache Hit**: If data exists in Redis, it