Backend Developer Practical : Fund Management System

Task 8: Documentation

Name: Nur Irdina Izzati Binti Khairuzaman

Github: irdinazati/Backend-Developer-Assessment (github.com)

API Documentation

The API allows you to manage investment funds, including creating, retrieving, updating, and deleting funds.

Base URL

```
http://localhost:5000/
```

Endpoints

1. Get All Funds

- Endpoint: /funds
- Method: GET
- Description: Retrieve a list of all funds.
- Sample Request:

```
curl -X GET http://localhost:5000/funds
```

- Sample Response:

```
[
  {
    "fund_id": 1,
    "fund_name": "Growth Fund",
```

```
    "fund_manager_name": "Irdina Izzati",
    "fund_description": "A fund focusing on growth stocks.",
    "fund_nav": 500000.0,
    "fund_date_of_creation": "2024-07-04",
    "fund_performance": 8.2
  }
]
```


2. Create a New Fund

- Endpoint: /funds
- Method: POST
- Description: Create a new fund.
- Sample Request:

```
curl -X POST http://localhost:5000/funds -H "Content-Type:
application/json" -d '{
  "fund_id": "F02",
  "fund_name": "Income Fund",
  "fund_manager_name": "Shaufy Yana",
  "fund_description": "A fund focusing on income stocks.",
  "fund_nav": 300000.0,
  "fund_date_of_creation": "2024-07-26",
  "fund_performance": 5.6
}'
```

- Sample Response:


```
{
  "fund_id": "F02",
  "fund_name": "Income Fund",
  "fund_manager_name": "Shaufy Yana",
  "fund_description": "A fund focusing on income stocks.",
  "fund_nav": 300000.0,
  "fund_date_of_creation": "2024-07-26",
```

```
    "fund_performance": 5.6
}
```

3. **Get a Specific Fund**
    - Endpoint: /funds/<fund_id>
    - Method: GET
    - Description: Retrieve details of a specific fund using its ID.
    - Sample Request:

```
curl -X GET http://localhost:5000/funds/F01
```

    - Sample Response:

```
{
    "fund_id": "F01",
    "fund_name": "Growth Fund",
    "fund_manager_name": "Irdina Izzati",
    "fund_description": "A fund focusing on growth stocks.",
    "fund_nav": 500000.0,
    "fund_date_of_creation": "2024-07-04",
    "fund_performance": 8.2
}
```

4. **Update Fund Performance**
    - Endpoint: /funds/<fund_id>
    - Method: PUT
    - Description: Update the performance of a specific fund using its ID.
    - Sample Request:

```
curl -X PUT http://localhost:5000/funds/F01 -H "Content-Type:
application/json" -d '{
    "fund_performance": 9.0
```

```
}'
```

- Sample Response:

```
{
   "fund_id": "F01",
   "fund_name": "Growth Fund",
   "fund_manager_name": "Irdina Izzati",
   "fund_description": "A fund focusing on growth stocks.",
   "fund_nav": 500000.0,
   "fund_date_of_creation": "2024-07-04",
   "fund_performance": 9.0
}
```

5. **Delete a Fund**
   - Endpoint: /funds/<fund_id>
   - Method: DELETE
   - Description: Delete a specific fund using its ID.
   - Sample Request:

```
curl -X DELETE http://localhost:5000/funds/F01
```

   - Sample Response:

```
{
   "message": "Fund deleted"
}
```

SQL Database Schema

This section describes the SQL database schema used for storing investment fund data.

Table: fund

- Description: Stores details of investment funds.
- Columns:

fund_id (INTEGER): Unique identifier for the fund (Primary Key, Auto Increment).
fund_name (TEXT): Name of the fund (Not Null).
fund_manager_name (TEXT): Name of the fund manager (Not Null).
fund_description (TEXT): Description of the fund (Not Null).
fund_nav (REAL): Net Asset Value (NAV) of the fund (Not Null).
fund_date_of_creation (TEXT): Date when the fund was created (Not Null).
fund_performance (REAL): Performance of the fund as a percentage (Not Null).

SQL Statements

- Create Table

```
CREATE TABLE fund (
    fund_id INTEGER PRIMARY KEY AUTOINCREMENT,
    fund_name TEXT NOT NULL,
    fund_manager_name TEXT NOT NULL,
    fund_description TEXT NOT NULL,
    fund_nav REAL NOT NULL,
    fund_date_of_creation TEXT NOT NULL,
    fund_performance REAL NOT NULL
);
```

- Insert Data

```
INSERT INTO fund (fund_name, fund_manager_name, fund_description,
fund_nav, fund_date_of_creation, fund_performance)
VALUES ('Growth Fund', 'Irdina Izzati', 'A fund focusing on growth stocks.',
500000.0, '2024-07-04', 8.2);
```

- Select All Funds

```
SELECT * FROM fund;
```

- Select Fund by ID

```
SELECT * FROM fund WHERE fund_id = 1;
```

- Update Fund Performance

```
UPDATE fund
SET fund_performance = 9.0
WHERE fund_id = 1;
```

- Delete Fund by ID

```
DELETE FROM fund WHERE fund_id = 1;
```

How to Use the API and Database

1. **Set Up the API:**
   - Make sure you have Python and Flask installed.
   - Save the provided API code in a file named Task3.py.
   - Run the API using the command: python Task3.py.
   - The API will be accessible at http://localhost:5000/

2. **Set Up the Database:**
   - Use the provided SQL statements to create the database and table.
   - Use an SQLite client or integrate it with the Flask API for data persistence.
   - For data migration, use the provided `Task5.py` script to move data between databases.