Source Code: AI Generated Report

Generated on: 2026-02-07 13:45:35

```python
import json

import hashlib

import os

from datetime import datetime


DATA_FILE = 'finance_data.json'


def load_data():

    if os.path.exists(DATA_FILE):

        with open(DATA_FILE, 'r') as file:

            return json.load(file)

    return {"users": {}, "expenses": [], "budgets": {}}


def save_data(data):

    with open(DATA_FILE, 'w') as file:

        json.dump(data, file, indent=4)


def hash_password(password):

    return hashlib.sha256(password.encode()).hexdigest()


def register_user(username, password):

    data = load_data()

    if username in data['users']:
```

```python
        print("User already exists.")

        return False

    data['users'][username] = hash_password(password)

    save_data(data)

    print("User registered successfully.")

    return True


def authenticate_user(username, password):

    data = load_data()

    hashed_password = hash_password(password)

    if data['users'].get(username) == hashed_password:

        print("User authenticated successfully.")

        return True

    print("Authentication failed.")

    return False


def add_expense(username, amount, category, description):

    data = load_data()

    expense = {

        "username": username,

        "amount": amount,

        "category": category,

        "description": description,

        "date": datetime.now().isoformat()

    }

    data['expenses'].append(expense)
```

```python
        save_data(data)

        print("Expense added successfully.")


def edit_expense(expense_index, amount=None, category=None, description=None):

    data = load_data()

    try:

        expense = data['expenses'][expense_index]

        if amount is not None:

            expense['amount'] = amount

        if category is not None:

            expense['category'] = category

        if description is not None:

            expense['description'] = description

        save_data(data)

        print("Expense updated successfully.")

    except IndexError:

        print("Expense not found.")


def delete_expense(expense_index):

    data = load_data()

    try:

        del data['expenses'][expense_index]

        save_data(data)

        print("Expense deleted successfully.")

    except IndexError:

        print("Expense not found.")
```

```python
def set_budget(username, amount):

    data = load_data()

    data['budgets'][username] = amount

    save_data(data)

    print("Budget set successfully.")



def check_budget(username):

    data = load_data()

    budget = data['budgets'].get(username, 0)

        total_expenses = sum(expense['amount'] for expense in data['expenses'] if
expense['username'] == username)

    print(f"Budget: {budget}, Total Expenses: {total_expenses}")



def generate_report(username):

    data = load_data()

     user_expenses = [expense for expense in data['expenses'] if expense['username'] ==
username]

    total_expenses = sum(expense['amount'] for expense in user_expenses)

    print(f"Total expenses for {username}: {total_expenses}")

    for expense in user_expenses:

        print(f"{expense['date']}: {expense['description']} - ${expense['amount']}")



# Mock data for testing

register_user("test_user", "secure_password")

authenticate_user("test_user", "secure_password")
```

```
add_expense("test_user", 50, "Food", "Lunch")

add_expense("test_user", 20, "Transport", "Bus ticket")

edit_expense(0, description="Lunch at restaurant")

delete_expense(1)

set_budget("test_user", 100)

check_budget("test_user")

generate_report("test_user")
```