

Personal Finance Management System

Objective:

Develop a "Personal Finance Management System" using Django, Django REST Framework, and PostgreSQL. This application will help users track their income, expenses, and savings goals. The primary goal is to assess your backend development skills with Django, your understanding of RESTful API design, and your ability to implement token based authentication.

Requirements:

Project Setup and Structure:

- Initialize a new Django project and configure it to use PostgreSQL as the database.
- Use Django REST Framework for creating API endpoints.
- Organize the project in a clean and modular architecture, separating concerns appropriately (e.g., models, serializers, views).

Core Features:

- **Income and Expense Models:** Define models for tracking income and expenses. Each transaction should include fields like amount, category, date, and a description.
- **Savings Goal Model:** Define a model for savings goals, including fields like target amount, current amount, and deadline.
- **Transaction API:** Create RESTful API endpoints to list, create, update, and delete transactions. Ensure these endpoints require authentication.
- **Savings Goal API:** Implement API endpoints to view and manage savings goals.
- **User Authentication:** Implement a custom user model and use Django REST Framework to create a login API. Authentication should be based on tokens, returning an access token and a refresh token upon successful login.
- **Registration API:** Allow new users to register through an API endpoint, creating a new user profile.

Authentication and Permissions:

- Utilize for managing user sessions and securing API endpoints.
- Ensure that only authenticated users can create, update, or delete transactions and savings goals.
- Allow users to manage their profiles and financial data only.

Database Design:

- Design the database schema with PostgreSQL, ensuring relationships between users, transactions, and savings goals are efficiently modeled.
- Implement migrations for your database models.

Coding Practices and Documentation:

- Write clean, modular, and reusable code following PEP 8 style guide.

- Use Django best practices for models, views, serializers, and URL routing.
- Include unit tests for your models and API endpoints using Django's testing framework.
- Document your API endpoints and provide a README file with instructions on how to set up and run your project, including database setup and any initial configuration steps.

Prohibited Use of AI Services:

- The use of external AI services or assistance for code generation or problem-solving in this task is strictly prohibited. Your submission will be reviewed for originality and compliance with this rule.

Deliverables:

- A GitHub repository containing the source code for the "Personal Finance Management System."
- Documentation covering setup instructions, API endpoint details, and a brief explanation of your project structure and design choices.
- Unit test cases for your models and API endpoints. (Optional)

Evaluation Criteria:

- Adherence to the project requirements and objectives.
- Quality of code, including organization, readability, and adherence to best practices.
- Functionality and reliability of API endpoints, especially the authentication flow with tokens.
- Design and implementation of database models.
- Documentation quality and completeness.
-