

Assignment: Build a Secure API for a Personal Finance Management Application

Candidates are advised to use AI tools for solving the problem, leverage AI and solve the problem creatively. We recommend submitting the assignment even if you are stuck somewhere in between.

Overview

Develop a RESTful API using Node.js for a personal finance management application. The API will allow users to track their incomes, expenses, and provide insights on their spending habits. The primary focus is on backend development, with an emphasis on security, data integrity, and efficient data retrieval.

Requirements

API Endpoints: Implement the following RESTful endpoints:

- POST /transactions: Add a new transaction (income or expense).
- GET /transactions: Retrieve a list of transactions for a given period.
- GET /transactions/summary: Retrieve a summary of transactions (total income, total expenses, and savings) for a given period.
- DELETE /transactions/:id: Delete a specific transaction.

Data Validation and Sanitization: Ensure all user inputs are properly validated and sanitized to prevent common vulnerabilities such as SQL injection and cross-site scripting (XSS).

Authentication and Authorization:

- Implement JWT-based authentication.
- Ensure that users can only access their own transactions.

Database Design: Design a relational database schema to store user accounts and transaction data. Provide an ER diagram and SQL schema definitions.

Efficient Data Retrieval: Implement efficient queries for retrieving transaction summaries. Consider the use of indexes or aggregation queries to optimize performance.

Security Measures:

- Implement rate limiting to prevent brute force attacks.
- Secure sensitive data at rest using appropriate encryption techniques.

Testing: Write unit tests for your API endpoints, focusing on testing the business logic, data layer, and security considerations.

Documentation: Provide comprehensive API documentation, including endpoint descriptions, request/response formats, and example use cases.

Evaluation Criteria

- **Functionality:** The API meets all the specified requirements and functions correctly.
- **Code Quality:** Code is clean, well-organized, and properly commented.
- **Security Practices:** Implementation follows best practices for web security.
- **Database Design:** The database schema is well-designed and supports efficient data retrieval.
- **Testing:** The application includes a thorough set of unit tests.
- **Documentation:** The API documentation is clear, concise, and complete.

Deliverables

- Source code for the API.
- Database schema (ER diagram and SQL definitions).
- API documentation.
- Instructions for setting up and running the application, including any necessary environment setup.

Submissions to rrllcwy@gmail.com with required deliverables along with public github repo link and deployment links with Subject: Nodejs intern assignment-2 submission