

Personal Finance Tracker

Back-end webApp

By: Mohammed Abdullah Al Adhary

Remember...
Hazlo siempre lo mejor
que puedes
....Always do your best



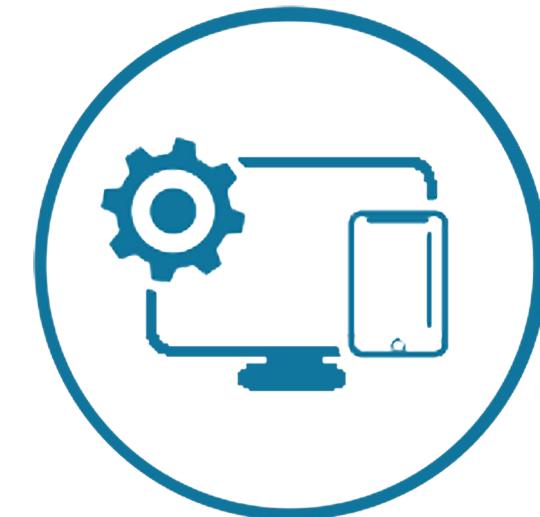


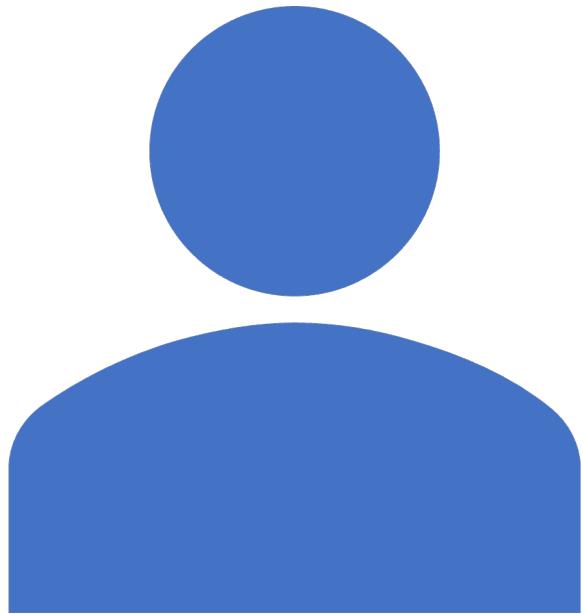
Taking Control of Your Finances

- Empowering Individuals for Financial Well-being
- A tool to help individuals manage their finances more effectively and make informed financial decisions.

Project description

- Create a web application that helps users manage their personal finances, track expenses, and set budget goals.
- This project can help individuals better understand their financial habits and make informed decisions.
- Trello:
<https://trello.com/b/DTZck8dS/personal-finance-tracker-webapp>





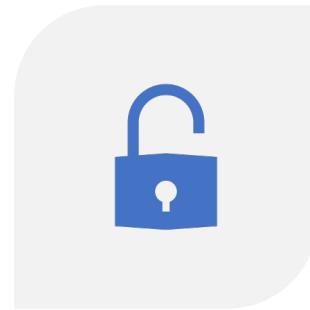
Key Features

1. User Registration and Authentication
2. Expense Tracking
3. Income Tracking
4. Report Analytics
5. Dashboard menu

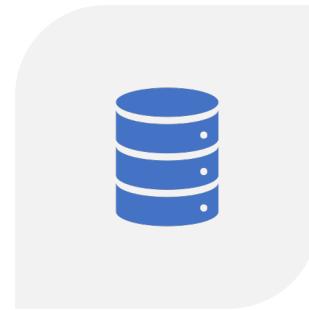
Technologies Used



JAVA FOR THE BACKEND
USING SPRING BOOT.



SPRING SECURITY FOR
USER AUTHENTICATION
AND AUTHORIZATION.



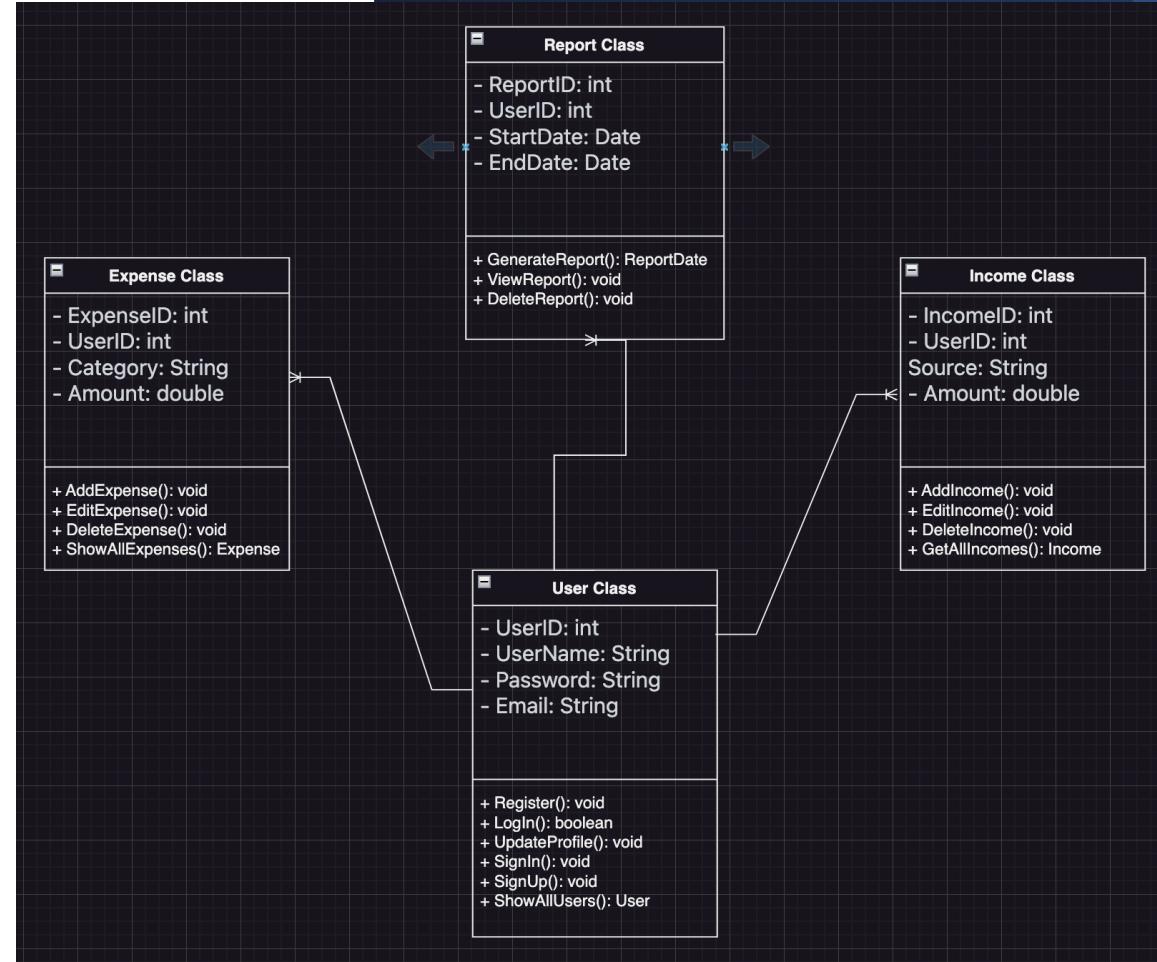
MYSQL WORKBENCH FOR
DATA STORAGE.



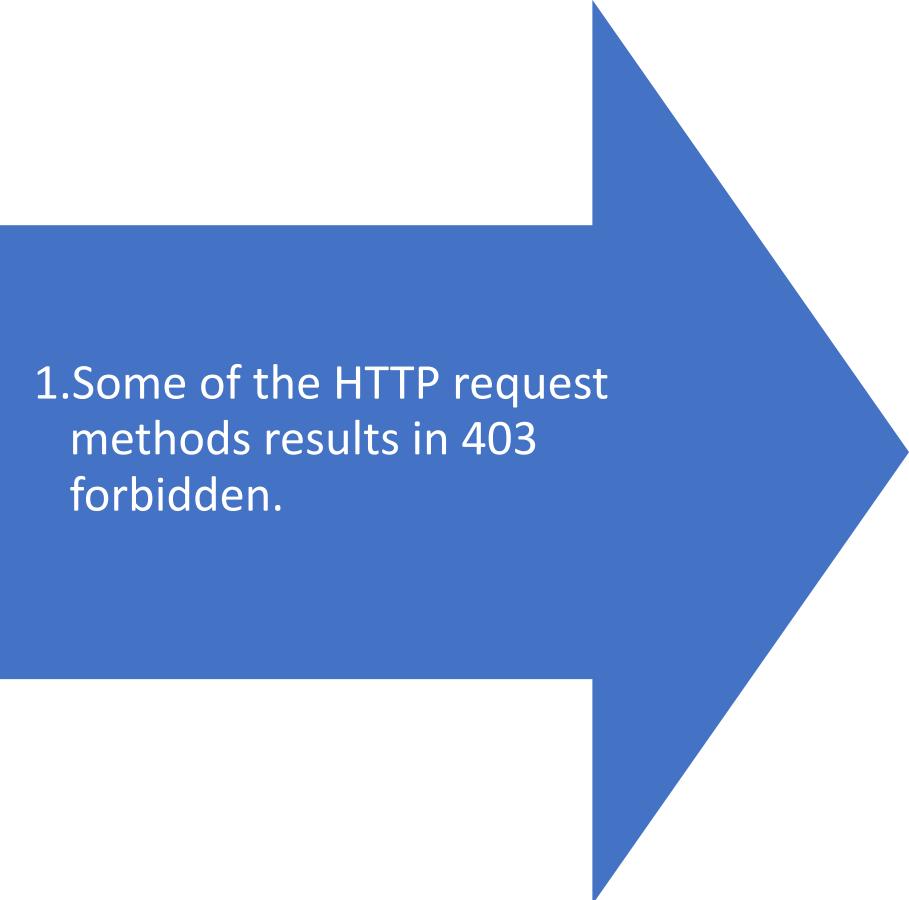
UML FOR SYSTEM DESIGN.

UML Diagram

- **User Class:**
 - **Has Many Expenses:** A User can have multiple Expense records. This is a one-to-many relationship, indicating that a single user can record multiple expenses.
 - **Has Many Incomes:** Similarly, a User can have multiple Income records, representing various sources of income.
 - **Generates Many Reports:** Users can generate different financial reports, so there's a one-to-many relationship between User and Report.
- **Expense Class:**
 - **Belongs to One User:** An Expense record belongs to a single User, indicating a many-to-one relationship between Expense and User.
- **Income Class:**
 - **Belongs to One User:** Similarly, an Income record belongs to a single User, indicating a many-to-one relationship between Income and User.



Technical Challenges:

- 
1. Some of the HTTP request methods results in 403 forbidden.

Possible Solutions:

1. Drop the database and then create it again.
2. Review and update your Spring Security configurations.

Future Improvements

- **1. Enhanced Security:** Strengthen the security of your application by fine-tuning authentication and authorization mechanisms. Regularly review and update security configurations to adapt to evolving threats.
- **2. Robust Error Handling:** Implement comprehensive error handling and logging to provide meaningful error messages to users and developers. This can significantly improve the debugging process.
- **3. Continuous Testing:** Implement a robust testing strategy, including unit tests, integration tests, and end-to-end tests. Regularly test your application to identify and resolve issues early in the development process.





Future work

- Implement Budget and Notification Section.
- Developing a mobile app version of the Personal Finance Tracker to make it even more accessible to users on the go.
- Explore the integration of the Personal Finance Tracker with users' bank accounts and credit cards.
- Add features for tracking investments, including stocks, mutual funds, and other assets.

just
do
your
BEST.

Conclusion

- In conclusion, the Personal Finance Tracker is more than just an application; it's a path to financial well-being and personal empowerment.
- It helps users take control of their finances, set and achieve financial goals, and make informed decisions. The future of financial management is brighter and more accessible with this powerful tool.
- Empower yourself. Control your finances. Flourish with the Personal Finance Tracker.

Thank You

- Wish you all the best
- Next slide: Demonstration

Any
Questions

Demonstration

- Refer to [README.md](#) file in my Github for testing EndPoints requests using [POSTMAN](#) App
- GitHub: <https://github.com/mohammedaladhary/Personal-Finance-Tracker-webApp/blob/main/README.md> for more Endpoints.
- SignUp(POST)/SignIn User(POST):
- localhost:8080/financeTracker/auth/dashBoard/signUp
- localhost:8080/financeTracker/auth/dashBoard/signIn
- AddUser(POST)/(GET)ShowAllUsers/(DEL)DeleteUser/(PUT)UpdateUser/(PUT)UpdateBalance:
- localhost:8080/financeTracker/dashBoard/users
- localhost:8080/financeTracker/dashBoard/users9/delete/{userId}
- localhost:8080/financeTracker/dashBoard/users/updateBalance/{userId}
- AddIncome(POST)/(GET)ShowAllIncomes/(DEL)DeleteIncome/(PUT)UpdateIncome/(GET)findIncomeById:
- localhost:8080/financeTracker/dashBoard/users/incomes/add
- localhost:8080/financeTracker/dashBoard/users/incomes/updateIncome/{userId}
- localhost:8080/financeTracker/dashBoard/users/incomes/findById/{userId}
- AddExpense(POST)/(GET)ShowAllExpenses/(DEL)DeleteExpense/(GET)findExpenseById:
- localhost:8080/financeTracker/dashBoard/users/expenses/add
- localhost:8080/financeTracker/dashBoard/users/expenses/delete/{userId}
- AddReport(POST)/(GET)ShowAllReports/(DEL)DeleteReport:
- localhost:8080/financeTracker/dashBoard/users/reports/generateReportById/{userId}
- localhost:8080/financeTracker/dashBoard/users/reports/delete/{userId}

