IT 309 SOFTWARE ENGINEERING

PROJECT DOCUMENTATION

Finance Tracker

Prepared by:
Ljoljic Filip
Begić Muhamed

Proposed to:
Nermina Durmić, Assist. Prof. Dr.
Aldin Kovačević, Teaching Assistant

20.06.2023

TABLE OF CONTENTS

Contents

1. Introduction

Finance tracker is a web application project designed to help you manage your personal finances effectively. You can track your income and expenses all in one place, set financial goals, gain insight and make informed decisions to better organize your finances.

1.1. About the Project

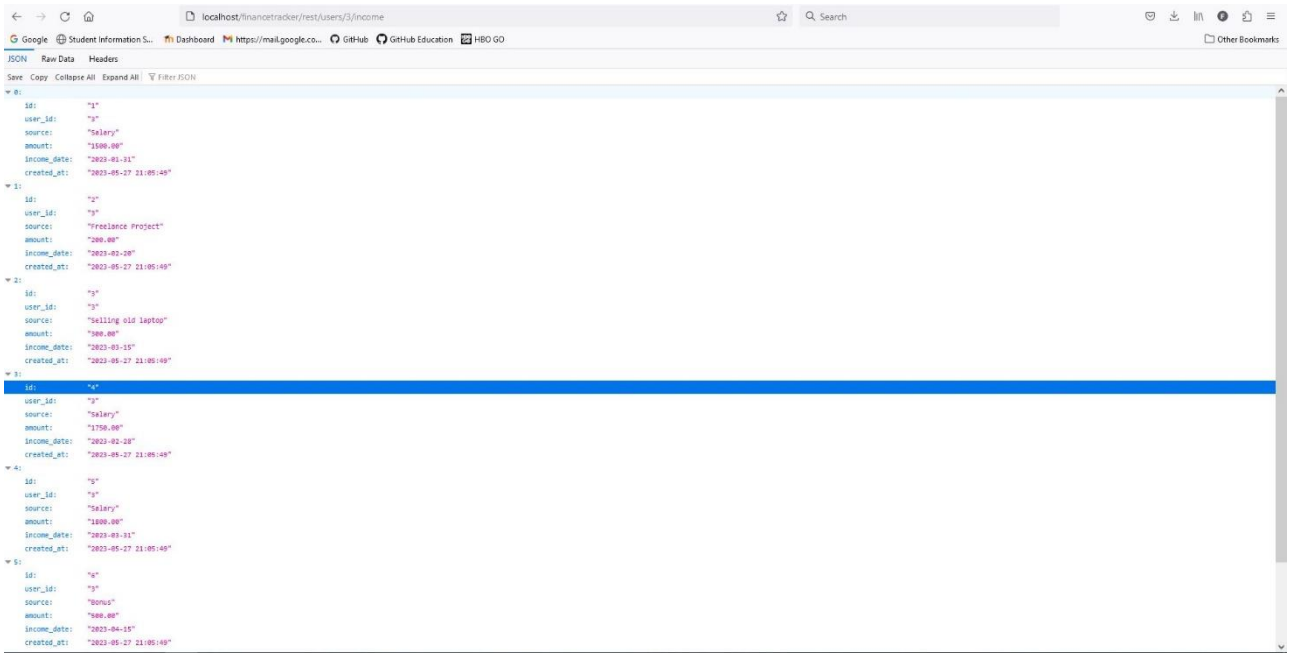
The Finance Tracker application is a web-based solution developed in PHP. Its main goal is to help users manage their personal finances and make informed financial decisions. The application provides an easy-to-use interface that allows individuals to enter and categorize their income and expenses and keep a clear record of their financial activities. Some of key features of this application include transaction management, categorization, budget setting and tracking and reports.

Link to the application: https://filip-finance-tracker.herokuapp.com/

1.2. Project Functionalities and Screenshots

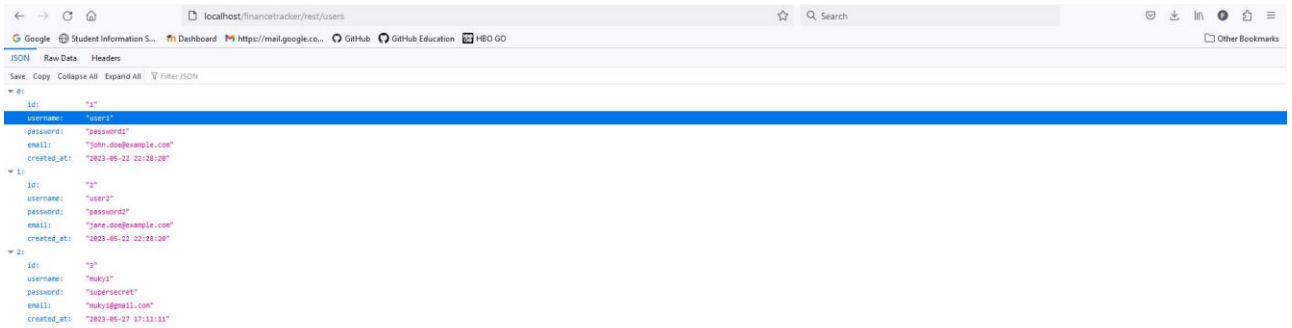
Regarding some main features of the application, we have methods to delete users, add users, update users and create users. Also we have some features for income and that is to add get update and delete income for a specific user. Here we have some screenshots.

In this screenshot we can see data when we use get single user income.

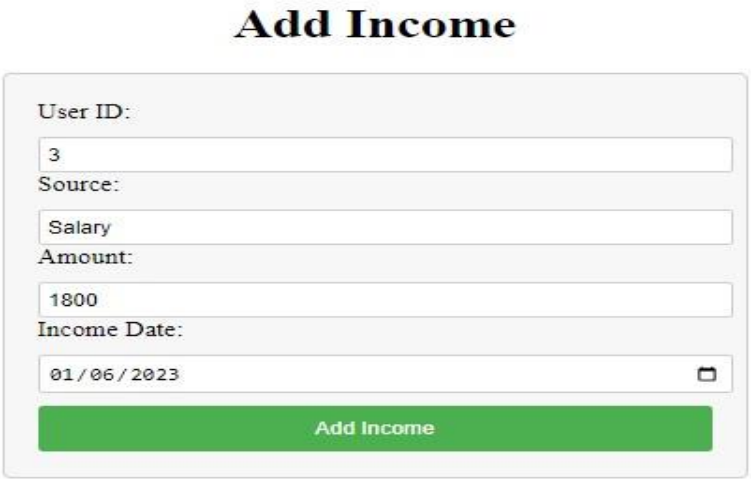


In this screenshot we can see data when we get single user.

In this screenshot we can see all users being displayed.



In the next screenshot we have a from for adding a new income. And the image after that we can se that the request has passed.



Name	Status	Type	Initiator	Size	Time	Waterfall
income	200	xhr	jquery-3.6.3.min....	346 B	6.07 s	

2. Project Structure

2.1. Technologies

Regarding technologies we used WAMP stack, so we used Apache server, MySQL database and PHP. Besides all these technologies we used JavaScript, html, and bootstrap for frontend part of the application. For deploying the frontend part of the application we used Heroku and for the database deploy we used the db4free.

For coding standards, we used PHP PSR-12 coding standards.

2.2. Database Entities

The tables in the database that we have are USERS where we have all users stored than we also have INCOME table where we have all income for all users and last table is EXPENSES where we have all expenses users had. Additionally, to these we have another two tables witch are categories and expense_categories but they aren't filled with any data.

2.3. Design Patterns

Regarding design patterns, the ones that we used in our application are DAO (Data Access Object) Pattern, Dependency Injection Pattern and Singleton Pattern.

We used DAO pattern because we wanted to separate the business logic of an application from the underlying data access logic. The DAO pattern provides a standardized interface for performing CRUD (Create, Read, Update, Delete) operations. So, an example from our project for DAO pattern is The BaseDao class serves as a base implementation of the DAO pattern. It provides common methods such as get_all, get_by_id($id), add($entity), update($entity, $id), and delete($id). These methods encapsulate the database operations related to retrieving, adding, updating, and deleting entities. The UserDao class extends BaseDao to specifically handle user-related operations, utilizing the common methods inherited from BaseDao. Also the UserDao class extends BaseDao and focuses on specific operations related to the "users" table, such as retrieving users from the database.

For Dependency Injection (DI) Pattern we used it because it allows you to inject dependencies (such as database connections or other objects) into a class rather than creating them directly within the class. In your code, the BaseDao class takes a table name as a constructor parameter, and the UserDao class extends BaseDao and sets the table name to "users" in its constructor. An example from our code could be that when we are passing table name to BaseDao and when UserDao is extending the BaseDao it gets the table name automatically. This is a good approach because we can reuse BaseDao class for different tables.

We have decided to use Singleton Pattern as well because with this pattern we can ensure that only one instance of a class can be created in the application providing a global point of access to the instance. Best file from our application to describe singleton pattern could be Config.class.php. Even though it is not implemented exactly as singleton pattern is implemented in Config file we have methods DB_HOST(), DB_USERNAME(), DB_PASSWORD() and DB_SCHEMA() that can be accessed globally without instantiating the Config class. These static methods provide access to configuration values related to the database connection, allowing you to retrieve the values without creating multiple instances of the Config class.

2.4. Tests

We have wrote four unit tests using UnitPHP. Test are located in tests folder inside the project. To be precise we have two files inside the test folder but the test we are using are in UsersRoutesTest.php file. Other file in test folder are commented tests but not meaningful. They were just used to give us an example how can we build some more meaningful tests. Test that we created are called testGetUserRoute();testGetAllUsersRoute();testGetIncomeByUserIdRoute();testDeleteUserRoute(). We have four tests in file but that is because the first and second tests do the same thing but are just differently called so because of that we have three different tests.

3. Conclusion

In conclusion, the finance tracker application built using PHP offers individuals a convenient and efficient way to track their personal finances. This application could really help people track their expenses and potentially in the future some of the features could be some enhance analysis where users could get some tips for saving money or being more efficient with their account balance. Another thing that could be improved in the future is some modified user interface with some charts and diagrams of expends and incomes.