

**PROJECT**

**Budget Tracker**

Graduate:

Cloșcă Marius-Cătălin

Coordinators:

Cristi Macoviciuc

Iași

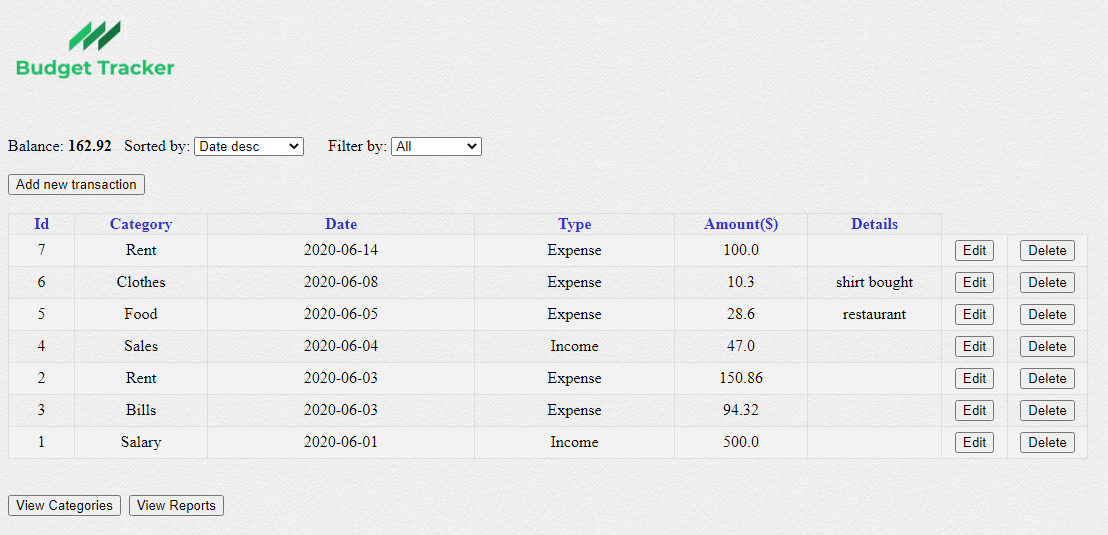
2020

1. **Description**

Budget Tracker is designed to help your budget, track and possibly control your monthly expenses. It provides an integrated set of features to help you manage your expenses and cash flow. Budget Tracker supports tracking of both your expenses and income and provides the ability to distribute your expenses and income into categories.

**Transactions**

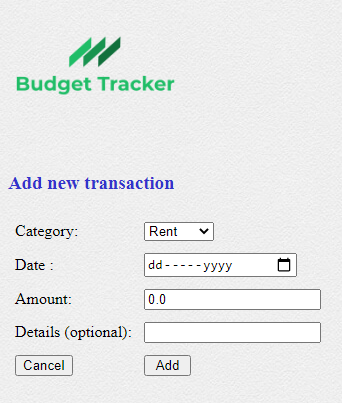
When you launch Budget Tracker, you are presented with a home screen that includes the balance and a table with all of your transactions. You can have the possibility to sort the transactions by date, amount or category and filter them by types or categories.



The main page provides the ability to add, update and delete transactions.

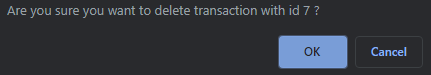
*Adding a new transaction/Update/Delete a transaction*

To add a new transaction, click the “Add new transaction” button at the top left of the main page.

**The page will load and you will need to:

* Choose the category
* Enter the transaction date
* Enter the amount
* You can optionally add some brief details for this transaction entry.

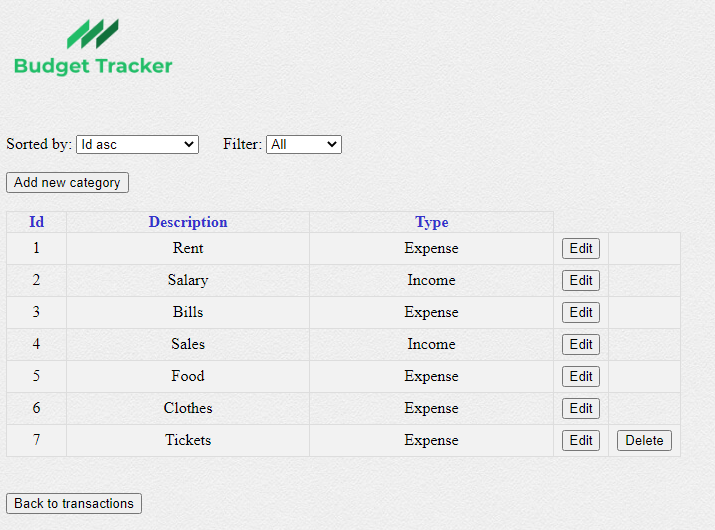
You can update an existing transaction. On the main page click the “Edit” button on the row of the transaction which you want to update. Make any changes and update.

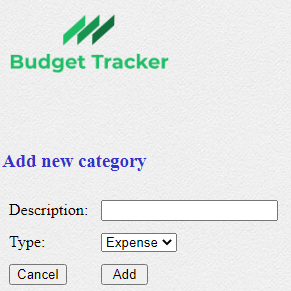
Also, you can choose to delete a transaction by clicking the “Delete” button for the transaction you want to remove and confirm OK.

**Categories**

To see the recorded categories you can access them by clicking the “View Categories” button on the left bottom of the main page under the transactions table.

Here you can sort the categories by Id and description, and also filter them by type (Expense/Income).



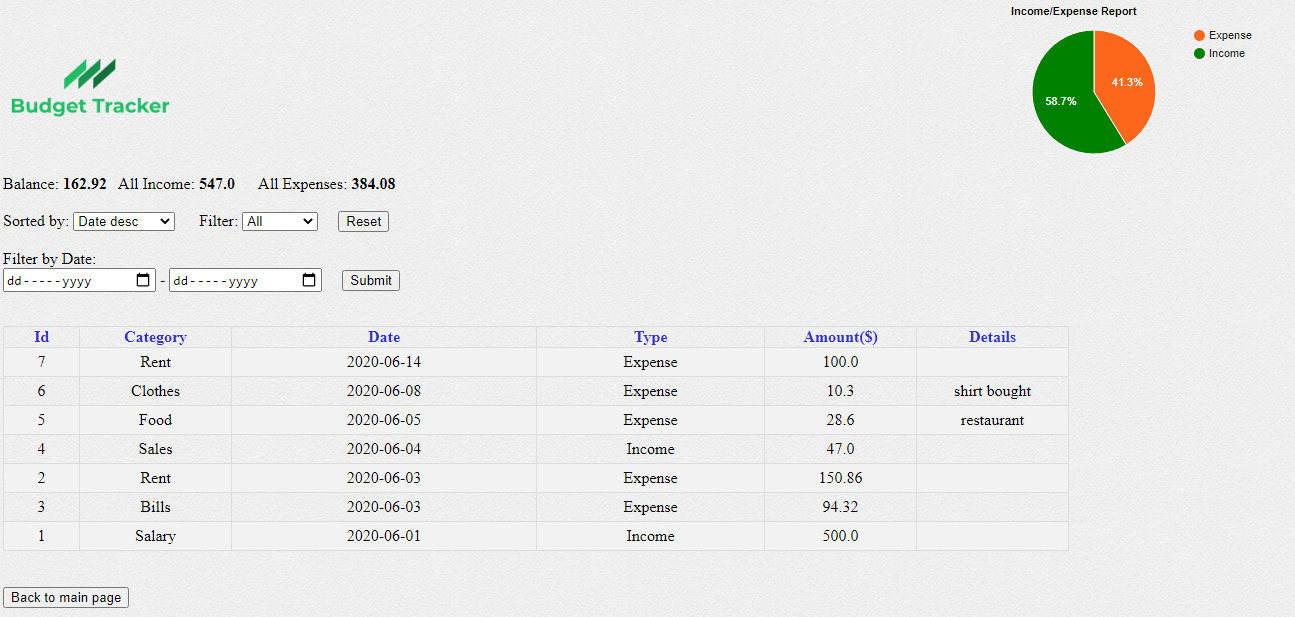
As well as the transactions page (main page) you can also add a new category, update/delete an existing one.

For adding a new category you just have to enter a description for the new description and also choose the type.

The procedure for updating or deleting an existing category is the same as in transactions.

**Note**: You can delete only those categories which are not linked to a transaction.

**Reports**

To access the reports page just click the “View Reports” button in the main page under the transactions table.

Here you can see some part of the transaction page without the add/update/delete forms and with more options, like filter by date, where you can get transactions from a date interval (by completing both date inputs), from a day to present (completing just first date input), or get past transaction to a certain day (completing just last date input) and clicking the “Submit” button. If you want to get all the transactions just click the “Reset” button.

While you are filtering by date, these statistic elements: ”Balance”, “All Income” and “All Expenses” and also, the pie charts are modifying only by the shown transactions in the table.

There are three charts in the reports page, one at the top right of the page showing amounts per type (Expense/Income), and the other two at the bottom of the page showing amounts per category, each one for every type.

1. **Setup**

No setup is needed, just start the application. If the database is missing (like on first startup), it will create a new database (of type SQLite, stored in a local file named 'budget\_tracker.db'), and use it to save the future data.

Once started, navigate with a web browser at url: http://localhost:4567/main.

1. **Technical details**

The project includes a web app (started with WebApp class) user interface.

Technologies used to create it:

* main code is written in Java (work with Java of minimum version 8)
* it uses a small embedded database of type SQLite, using SQL and JDBC to connect the Java code to it
* it uses Spark micro web framework (which includes an embedded web server, Jetty)
* web pages: using the Velocity templating engine, to separate the UI code from Java code; UI code consists of basic HTML and CSS code (and a little Javascript)
* charts: using JavaScript library Google Charts (https://developers.google.com/chart)
* includes some unit tests for DB part (using JUnit library).

Code structure:

* java code is organized in packages by its role, on layers:
* db - database part, including DTOs and DAOs, as well as the code to init and connect to the db
* ui - code related to the interface/presentation layer
* root package - the main classes for the types of interfaces it supports
* web resources are found in main/resources folder:
* under /public folder - static resources to be served by the web server directly (images, css files)
* all other (directly under /resources) - the Velocity templates.

Note: the focus of this project is on the back-end part, not so much on front-end part.