EXPENSE TRACKER - WORKFLOW

(VERSION 1.0)

**OBJECTIVE**:

To maintain a webpage to add daily expenses

**SCOPE:**

This helps prevent manual entry and keep track of all daily expenses.

**PROCESS WORKFLOW:**

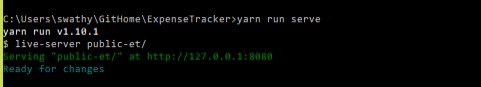
**To open the application on a webpage, follow the steps given below:**

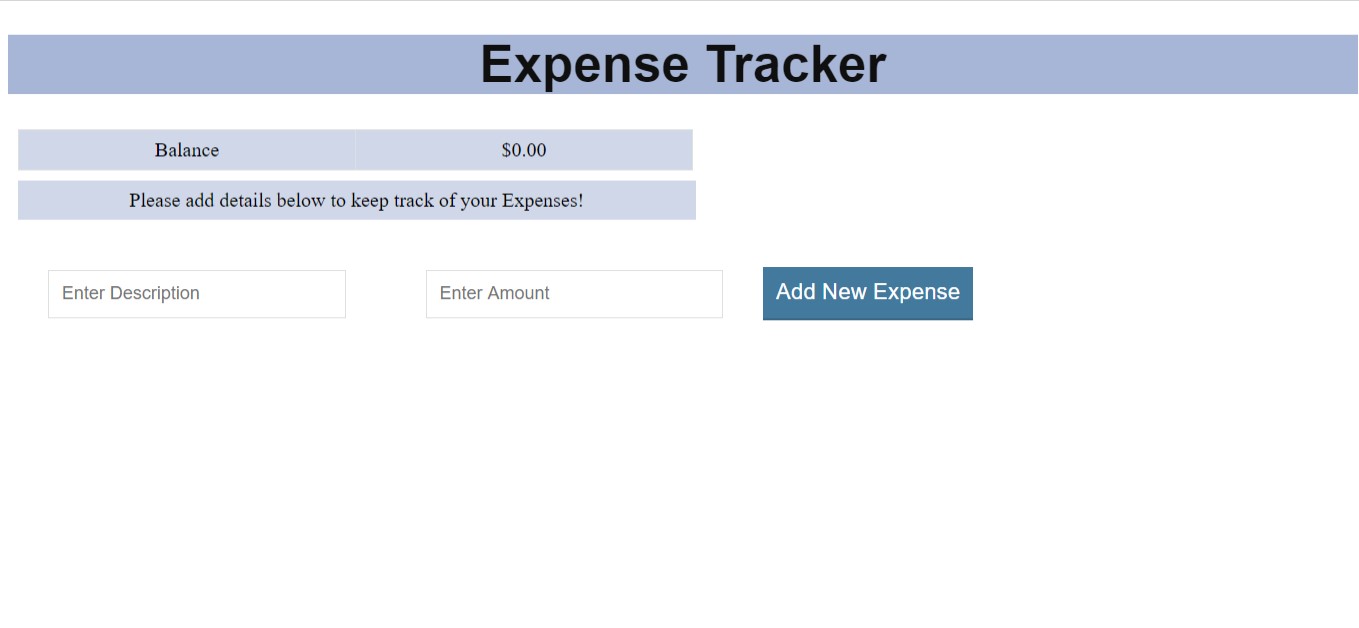
1. Open the windows terminal and navigate to the Expense Tracker folder.(Path where all the files are stored.)
2. Install live-server and babel on your local machine with the following command:

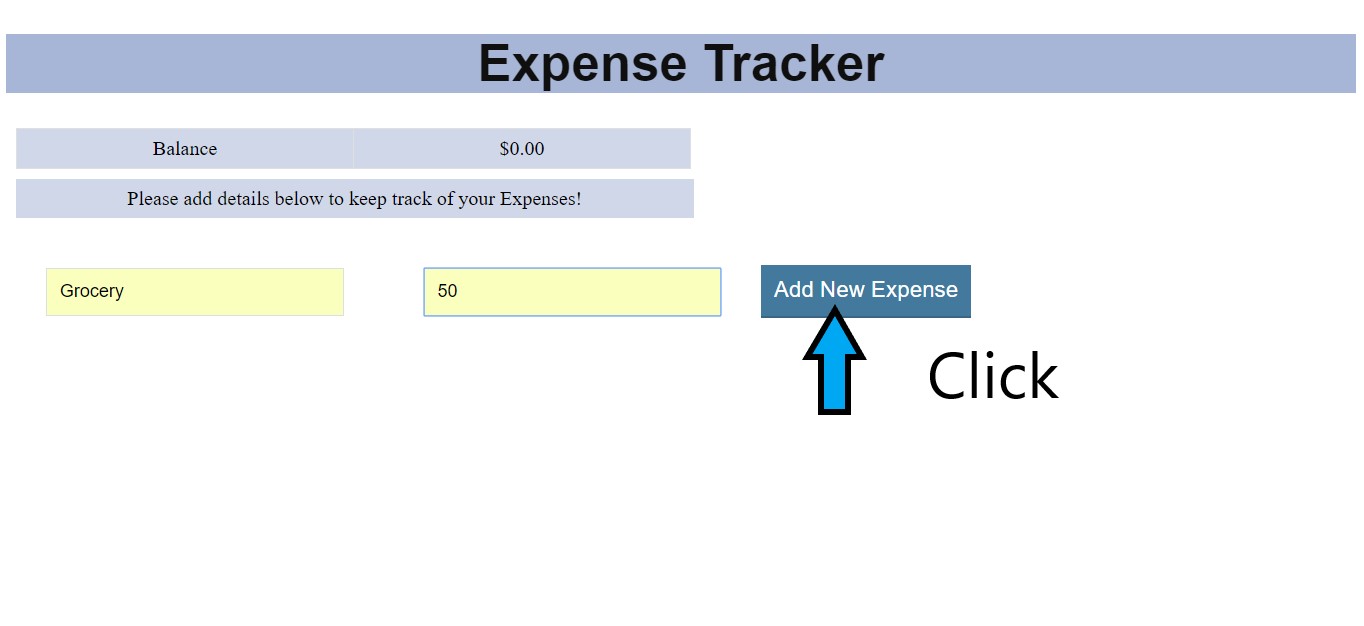
“yarn add live-server [babel-cli@6.24.1](mailto:babel-cli@6.24.1)”

1. After installation is a success, give the following command in the terminal:

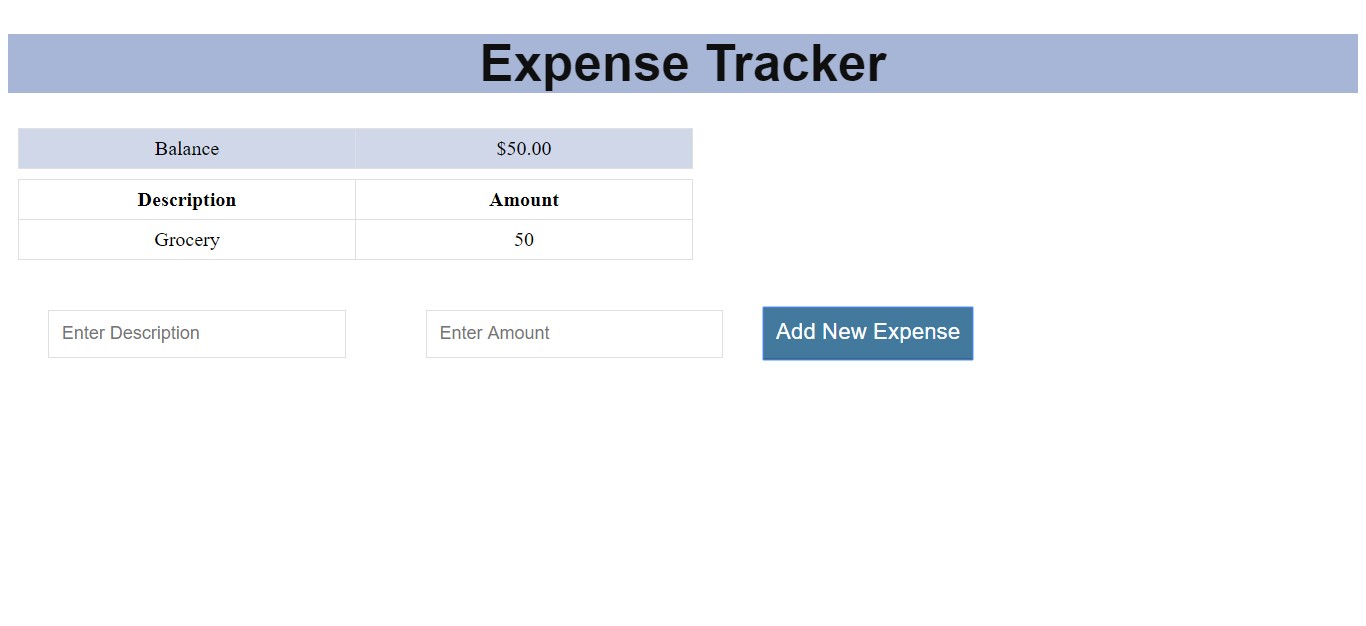
“yarn run serve”



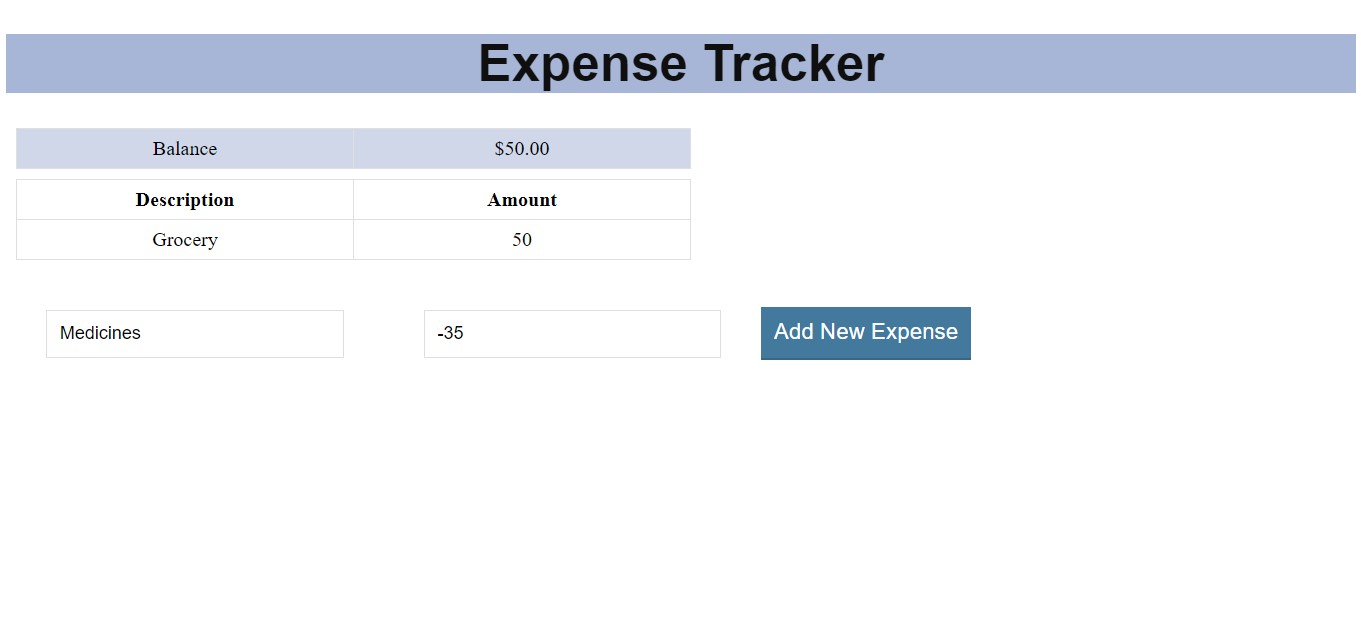
1. The browser opens automatically with a local port. The following screen below shows up
2. The user enters the Description and Amount and hits the ‘Add New Expense’ button.



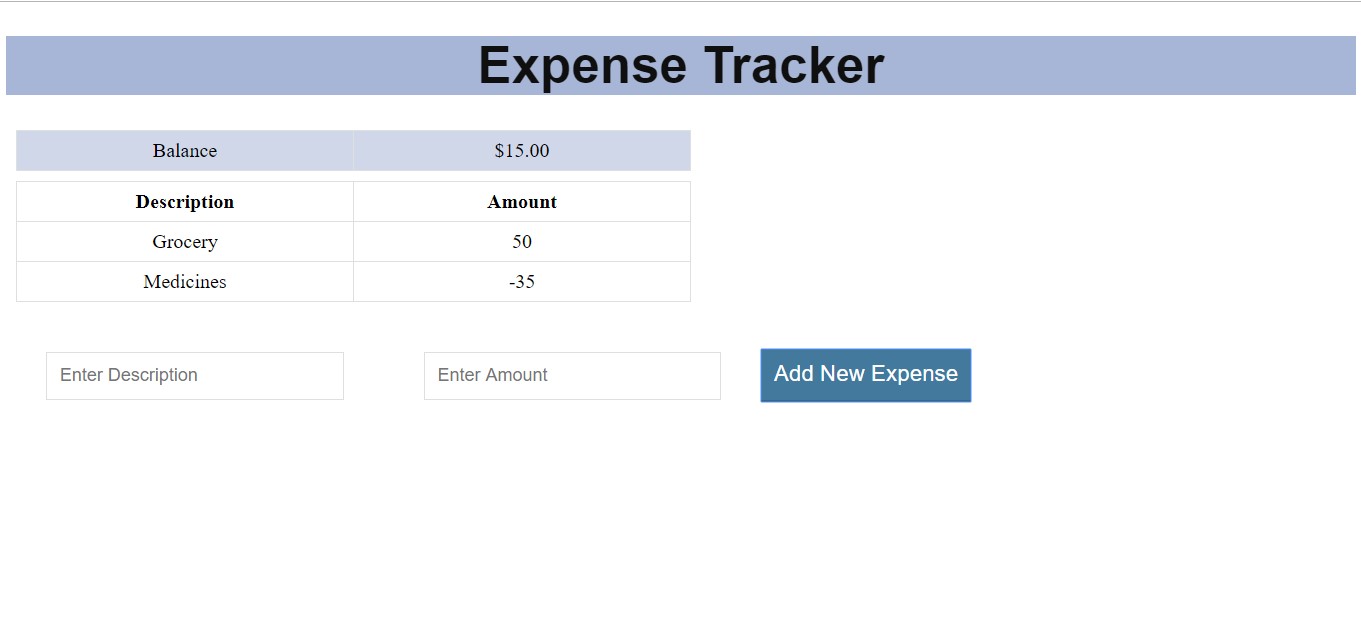
1. The Expense is added and shows up in a table like shown below:



1. Negative values can also be added.



1. The expense is added to the list of expenses. The Balance is also updated automatically.



**FEATURES:**

* Allows to add one expense at a time.
* Validations are applied for Description and Amount fields.
* The records are displayed in a table, in row-column format for easy visibility. Each row corresponds to one expense.
* UI is simple for easy access.
* The values entered are stored in the backend. When the session ends,(i.e.),when the browser is refreshed, the values are not lost.

**DESIGN ELEMENTS:**

|  |  |  |
| --- | --- | --- |
| **NAME OF DESIGN ELEMENT** | **NAME/NUMBER** | **COMMENTS** |
| Language | HTML,CSS,React.js |  |
| <div> Element | 1 | Displays all React Components. |
| Number of Input Fields | 2 | Description  Amount |
| Number of buttons | 3 | Add Expense |
| CSS | Internal CSS | Within the HTML file in <style> tag in <head> section |
| Script Functions | 6 functions | 1) React components and stateless functional components are written in app.js file under src folder.  2) This file is converted into a file, app.js in public-et folder with browser compatible code using Babel.  3) This is added within the HTML file in <script> tag in <body> section |

**ENHANCEMENTS THAT CAN BE DONE (2.0):**

1. **Edit** Button to edit information against each expense and save changes.
2. **Remove** button against each expense to delete**.**
3. Incorporate **Date** Field when adding a new Expense.
4. **Sort by** Date and Category.