

## 1. Expense manager

Personal finance tracker. You can record all your spendings and income to understand where does your money actually go.

Basic functions

- Authentication
- Transaction list (Add / Edit / Remove transaction functions)
  - Two types of transactions: income / expense
  - Each transaction should contain following data: amount, date, place (can be empty), comment (can be empty), image (can be empty).
- Categories for transactions (Add / Edit / Remove category functions)
- Statistics for user per period (month, year, week)
  - Total spendings (and per category)
  - Total income (and per category)

Examples: [ZenMoney](#), [Spendee](#)

## 1. UI design

### Acceptance criteria

Milestone's acceptance criteria: a semi-finalized set of pages with production-ready design (can be updated on later stages of development) for both screen sizes.

Modern CSS best-practices should be used (no `float` block positioning, `!important` overrides or `<table>` layouts are allowed).

Disclaimer: as we do not study UI design in this course, there are no strict requirements to your project interface appearance. However, a more or less meaningful UI is required to pass this milestone, because it will enforce you to use a good amount of CSS. Consider yourself advised to look through at least [Material Design reference](#) to have a basic idea of how a good UI might look like. You can make a quick search over the internet to find a good reference your project may follow. Get ready for additional tasks during your project showcase on this milestone.

### Result grading

- [4-6] basic CSS usage. Customized `input`, `button` and other components, `:hover` and `:disabled` states for interactive elements, `display: flex` block positioning.
- [7-8] `display: flex` or `display: grid` block positioning, `:focus` state for interactive elements (different from `:hover`), [BEM](#)-like CSS class naming.
- [9-10] dark theme support (via media query and CSS variables), animations and rich visual style.

## 2. Front-end basics

### Acceptance criteria

At this point you must provide a working skeleton of the website: basic navigation, routing, authentication (if required), state-management, networking, etc. A check-point to see if everything is going well before the final stage of development.

### Result grading

- [4-6] basic JS. Functions, variables (no `var` is allowed), cycles, arrays.
- [7-8] classes, AJAX requests (via [Fetch API](#)), promises.
- [9-10] SPA implementation (via browser's History API), [web components](#), `async / await`, local storage.