



# Project specification for CoinTracker

## General

CoinTracker is an application for tracking your monthly income as well as tracking your monthly expenses with the possibility of setting budget limits.

CoinTracker should be accessible from mobile devices, for that reason, mobile responsiveness is required, and the desktop version is nice to have, but not required.

The low budget of this project does not allow designers to be hired, so we decided to go with Material UI as a library for all components. This should fast forward the development, which will lead to faster MVP on the market.

## Resources

Icons: <https://material-ui.com/components/material-icons/>

MaterialUI\_Components: <https://material-ui.com/>

Date Picker: <https://material-ui-pickers.dev/getting-started/installation> (datefns - recommended)

Autocomplete: <https://material-ui.com/components/autocomplete/>

Charts: <https://github.com/reactchartjs/react-chartjs-2>

Multi axis: <https://reactchartjs.github.io/react-chartjs-2/#/multi-axis-line>

Random User API: <https://randomuser.me/api>

Design: <https://xd.adobe.com/view/80132099-0eba-434f-9efb-92e1fb688e10-d540/>

## Pages

- Sign in
- Sign Up
- Welcome wizard
- Overview
- New Entry modal
- Categories
- Statistics
- Entries (income / expense)

## Sign In

Username

Input type text

Password

Input type password / text

Eye Icon - Initial, as user types, "\*" should appear for each character, but if user clicks on "eye" icon, should be able to view the actual password characters

Sign up now, it is free

Should link to sign up page

Sign in button

Ajax request is sent to this API (<https://randomuser.me/api/>) which will retrieve users information and upon success, redirect to Overview page should happen. From this response, the image should be placed into top-right placeholder in the header

## Sign Up

Username

must be valid email

Password

To be able to view the password if clicked on (eye icon)

Validation	Message
Min Length 8	Password must be at least 8 characters
Max Length 32	Password must be less than 32 characters
Special Char	Password must contain at least one of this characters (!@#\$%^&*)

Sign in please

Should link to sign in page

Sign up button,

Ajax request is sent to this API (<https://randomuser.me/api/>) which will retrieve users information and upon success, redirect to the Welcome Wizard page

should happen. From this response, the image should be placed into top-right placeholder in the header

## Welcome wizard

(only first time when user signs up)

Stepper with 3 steps

1. How much money you have now
2. Choose Categories
3. Set budgets

Depending on this choice, these categories will be available for further use. (should be saved globally)

### Step 1

Input for current amount of money

Validation, amount has to be greater than 0

"Add" button disabled when amount is not entered, clicking if enabled leads to next step

### Step 2

List of categories

Create JSON with initial categories, you should list from those

Proposal Properties for each category: Type(income/expense),  
Name(string), Budget(number), Icon(string - for material-icon-name),  
isEnabled(Boolean)

User should be able to choose from list of categories (checkbox) (this should enable/disable)

"Done" button is disabled if not single category is checked

Categories that are chosen will be used for further use (consider saving them into some global variable/context)

### Step 3

List of categories (the selected from the previous screen or only the enabled) should be displayed in the list with number input on the right. Input will be for budgeting the category (update the budget property into corresponding categories). This is optional (if not entered 0 will be taken into consideration)

"Complete" button should be available all the time (no disable state). After clicking, the user is redirected to the Overview Page.

# Overview

Here you can see the current status of the application, contained in 3 different boxes

## 1. Income

1. Shows cumulative amounts by income categories In current month / planned amount
2. If budget(planned) is greater than 0, progress bar appears at the bottom of each category which should visualize the completed status

## 2. Expenses

1. Shows cumulative amounts by expance categories In current month / budget amount
2. If budget is greater than 0, progress bar appears at the bottom of each category which should visualize the completed status
  1. Progress bar, Icon, amount and name of category goes "red" if the accumulative value exceeds the budget defined

## 3. Entries

1. List of entries that contain the category, the amount and the date when money are spent/get
2. If category is from income, than number should be green with "+" sign in front
3. If category is from expenses, than number should be red with "-" sign in front
4. Left clicking on entry opens the Update Entry Modal with prefilled inputs from the entry that is clicked
5. Right clicking on entry opens context menu with 2 options
  1. Duplicate - opens New Entry modal with prepopulated all the same values for all fields
  2. Create new - opens New Entry modal with empty fields
  3. Delete - will delete the entry ONLY with confirmation dialog(modal)

## 4. "Add new" (FEB)

1. Should set white overlay with 75% white
2. And display 2 buttons
  1. Add expense
  2. Add income
3. Each should open same modal (New/Update Entry), but with preselected dropdown value for type of entry

## New/Update Entry

This modal should be available to open on any screen/view

### NEW

- Type of entry (required)
  - Income (default)
  - Expense

When changing this select, the options in next select should be changed according to type of category selected. (example if "Income" is chosen, than in categories only categories of type income should be displayed)
- Category should be autocomplete of all available categories (required)
  - Autocomplete means to be able to search before you select
- Amount (required)
  - Input field for number
- Date (required)
  - Today's date is preselected
  - Date picker opens
- Description (optional)
  - Text input

Cancel - dismiss modal and happen nothing

Add - add the new entry in the corresponding list (Entries box in overview), and close the modal

### UPDATE

If this modal is opened when updating Entry, the same rules apply but all inputs are prefilled with values from the entry that is clicked.

Text of the button is "Update" instead of "Add" and corresponding entry should be updated instead of creating new.

## Categories

List of all categories is displayed.

For each category (Icon, Name, Budget if any)

Color of the Icon, name and budget depends on the type of category, for income is green, for expense is red.

Left clicking on any category opens Edit Category modal (same modal for creating but with prefilled inputs from the clicked category)

At the top of the list, new category button is present always  
Clicking on it opens Create New Category Modal

## New/Update Category

### NEW

Opens modal for creating new category

- Type (required)
  - Select with 2 options
    - Income (default option)
    - Expense
- Name (required)
  - Input text
- Icon (optional)
  - Custom component that will allow users to select(pick) icons from predefined list of icons
- Budget
  - Input number (optional)
- Enabled
  - Checkbox (checked by default)

### UPDATE

Same modal, but with prefilled inputs from the category that is clicked

## Statistics

This page visualizes some of the data

- Income (Horizontal bar chart)
  - Accumulative per income category
    - X-axis amount
    - Y-axis category
- Expense (Horizontal bar chart)
  - Accumulative per income category
    - X-axis amount
    - Y-axis category
- Income & Expense (Multi Axis Line Chart)
  - X-Axis = Dates, starting (1st of [current month]), until today
  - Y-Axis1 = Green Line = Accumulative per day for all income categories
  - Y-Axis2 = Red Line = Accumulative per day for all expense categories

## Scoring

All the application should be dynamic and data driven, not single value should be hard coded (except the initial JSON for categories)

Think as you have back-end APIs,

example: create separate functions for:

- new entry,
- update entry,
- get all the entries,
- Delete entry

This way will be much easier to achieve the bonus 20 points and later to add the back-end communication.

Sign In/Sign Up	10
Welcome Wizard	10
Overview	10
New/Update Entry	5
Date picker	5
Autocomplete	5
Categories	5
New/Update Category	5
Custom Icon Picker	10
Statistics	5
Income/Expense	5
Income & Expense	5
Desktop Design	20
BONUS: Local Storage support, for the categories and data entries	20