Under Budget

Cameron Rutledge

Overview

Under Budget will be a web application for consumers of variety of ages to track their expenses and receive monthly analysis of their expenses. It will provide the tools for users to visualize how much money they are estimated to save and breakdown the distribution of their expenses into various categories.

Personas

My application is targeted towards two personas. The main persona I am targeting are teenagers/young adults that are entering the workforce, giving them the tools to start saving early.

The second persona I am targeting are older users who are looking for a tool to help them manage their day-to-day expenses quickly and easily.

Benefits/Value

The base features of my application are intended to grow the userbase, future updates will add more features to provide more detailed breakdowns of expenditures and expand the capabilities of the application to connect to other services to automate expense tracking. These features will be only available to users who purchase the “premium” Under Budget application.

Interaction

Through a user’s interaction with my application, they will have a better grasp of how much they spend each month. They will also be given the tools available to analyze their expenses to identify categories of expenses they may be spending too much on.

Minimum Viable Product (MVP)

Overview

The main purpose of the application is to provide the end user the tools to track and analyze their expenses. The user will need to be able to input and store their expenses in addition to other relevant savings information. To this end my development will focus on creation of a monthly savings goal and the tracking of expenses of a monthly period within a database.

Features

1. User logon functionality
2. Ability for user to create and update their savings goal
3. Ability to create and delete expense entries
4. Ability for user to view and receive data breakdown of expense entries

System Architecture

I will be utilizing the python-based web framework Flask for my user backend API and Bootstrap as my frontend framework. For data storage I will be using SQLite. Bootstrap will render the data on the webpage and allow for user interaction while Flask will be used to communicate with my SQLite database to store the user’s savings goal and expense data.

Data Structure

Since I am using SQLite I am utilizing a relational database. I will have 3 tables: Users, Months, and Expenses. There is a one-to-many relationship between the Users and Months table, and in addition a one-to-many relationship between the Months and Expenses table. For every user there will be many month expense sheets and for every month expense sheet there will be many individual expenses. This relationship and the contents of the tables are visualized below.

Graphical user interface, application

Description automatically generated

Table users as U {

id int [pk, increment] // auto-increment

username varchar

password varchar

full\_name varchar

created\_at timestamp

savings\_goal int

}

Ref: U.id < M.user\_id

Table months as M {

id int [pk, increment]

month date

user\_id int

}

Ref: M.id < E.month\_id

Table expenses as E {

id int [pk, increment]

expense\_type varchar

cost int

month\_id int

}