

# UrBudget: User Documentation

## Introduction

UrBudget is a simple Ruby on Rails Web Application to allow users to track how they are spending their monthly disposable income. Using the Chartkick gem to display graphs of user spending, the app lets a user see a neat summary of their current financial status.

This documentation will be divided into two sections:

- Submitted files
- Online Production Copy (<https://shielded-retreat-9618.herokuapp.com>)
- Questions you may have?

## Submitted Files

Installation: Mac

(steps taken largely from <http://railsapps.github.io/installrubyonrails-mac.html>)

- Install the latest version of Xcode Command Line Tools
- Download RVM ([rvm.io](http://rvm.io)), a Ruby Version Manager
- To install RVM and Ruby: type the following code in the terminal

```
$ \curl -L https://get.rvm.io | bash -s stable --ruby
```
- Update the version of RubyGems (already on your computer) using ‘gem update –system’ in terminal
- Type ‘gem update’ into terminal to update all gems
- Type ‘gem install rails’ into terminal to install latest version of rails
- Now cd into downloaded directory containing submitted files.
- Type ‘rails server’ into the terminal and navigate to ‘localhost:3000’ in your browser to load the application

Installation: Windows

- Please try not to use Windows to run Rails apps. It is very painful. If you must, please head over to <http://railsapps.github.io/installing-rails.html> and navigate down to “Windows” heading.

Installation: Linux

(steps taken largely from <http://railsapps.github.io/installrubyonrails-ubuntu.html>)

- Update package manager (“sudo apt-get update”)
  - Install Curl (“sudo apt-get install curl”)
  - Install RVM (same process as for Mac)
  - Install node.js (“sudo apt-get install nodejs”)
  - Update RubyGems (same as Mac step)
  - Update Gems (same as Mac step)
  - Install Nokogiri Gem (“gem install nokogiri”)
-

- Install Rails (same as Mac steps)
- Now cd into downloaded directory containing submitted files.
- Type 'rails server' into the terminal and navigate to 'localhost:3000' in your browser to load the application

#### Usage:

1. Click "Sign Up"
2. Enter your user information and submit form. Feel free to test out invalid emails, unmatched passwords, or passwords shorter than 6 letters.
3. Enter your monthly disposable income in budget form. You may try to navigate away from this page. You should not, but it should also not let you proceed to other pages (except static pages) without you finishing the budget form. Feel free to test out non-numerical, negative, or ridiculously high values.
4. After you submit budget form, you will be redirected to a homepage, with an html form allowing you to submit expenses on the left and a blank feed of expenses on the right.
5. Enter details about the expense in the form and submit it to see an expense pop up in your newsfeed, and the values of money remaining and used update.
6. Click the "View Graphically" to see a snazzy graph added by the Chartkick gem, which connects the Google Charts API into Rails, depicting spending by category. \*\*SEE NOTE\*\*
7. Feel free to log out, log in, update your profile (click "Settings" on dropdown), and update budget (click 'upgrade urBudget') on dropdown as needed.

## Online Production Copy

Installation: Go to the linked URL at (<https://shielded-retreat-9618.herokuapp.com>)

Usage: Same

## Questions You May Have? / Next Steps

- Why can I spend more than my disposable income?
    - We live in America, the home of debt and easy lines of credit! You can absolutely exceed your monthly budget.
  - Does the budget database actually reset its values (remaining and used) at the start of each month?
    - Sadly not yet, but the feature is definitely fairly high on the priority list moving forward.
  - Are categories case sensitive?
    - Try and find out!
  - How sweet is the Chartkick Gem?
-

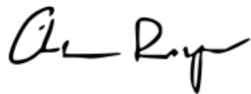
- Sweeeeeeeeeettttt!
- Next Steps:
  - Implement monthly reset of budget values so user can start fresh at start of each month.
  - Allow user to break down their budget according to categories (ie X % to Food)
  - Implement more Graphing Capabilities (trends over time)
  - Ability to export reports.

## NOTE

The differences in the type of database: PostgreSQL for the production database and Mysqlite3 for the development database result in the graph not working in the production database. I tried to make them work across environments, but unfortunately could not (groupdate gem doesn't work in Mysqlite3 and querying method for graph (expenses have no primary key) doesn't work with PostgreSQL).

Known Error: Sometimes Graphs will display loading... rather than graph. Refresh the page for it to appear, this is a known error in Chartkick

Thank you for reading,



Arun Rangarajan