Zero CLI

A command line interface application based on [Zero](https://www.zerofasting.com/)

## **Introduction**

Zero is a time tracking application for intermittent fasting, which helps the user to track the time when he/she has an active fast (meaning the user is not eating or drinking anything for a specific time period).

## **Technical Project Details and information**

The application should use native PHP without usage of any frameworks or packages

The code should placed on GitHub (it can be public or private)

Use Git branches and create pull requests when a feature is implemented or a bug is fixed

Use proper git commit messages, branch names, pull request titles etc.

Add reviewers (davorminchorov and slobodansgithubusernamegoeshere, you may need to add us as collaborators to your repository for this action)

Write detailed instructions on how to install and run the CLI application in the README.md file.

Everything should be written in english

Use object oriented programming concepts as much as possible.

Also, make sure to format the code based on the [PHP PSR 12 rules](https://www.php-fig.org/psr/psr-12/)

You can start with any of the features, they don’t have to be done in order.

If you have any questions don’t hesitate to ask

## **Requirements**

The CLI application should have the following features:

1. Check the fast status
2. Start a fast ( available only if there is not an active fast)
3. End an active fast (available only if there is an active fast)
4. Update an active fast (available only if there is an active fast)
5. List all fasts

When the application is started we should have an ordered list menu, so the user can select some of the above options.

### **1. Check Fast Status**

#### **Description**

As a user, I would like to check my current fasting status so that I know how long I have been fasting for.

#### **Acceptance Criteria**

* When the “Check the fast status” options is selected, display the current status of a fast which will print the following details:
  + Status (Active / Inactive)
  + Started Fasting (the start date when the fast started, example April 1, 12:00)
  + Fast Ending (the expected date and time, example April 2, 20:00)
  + Elapsed Time (the current elapsed time for the current fast, example 1:30:45)
  + Fast Type (The type of fast, example: 16 hours)
* If the fast is inactive, it should just display a message that lets the user know that he is currently not fasting.
* The information should be displayed from the JSON file.

## 2. Start a Fast

#### **Description**

As a user, I would like to start a fast so that I can track my fasting. This option should be available only when there is not an active fast.

#### **Acceptance Criteria**

* When we select the “Start a fast” option the user will be asked for input for the following details:
  + Start Date (When the fast started)
  + Fast Type (the type of fast, example 16 hours)
* The end date for the fast should be calculated based on the start date and the fast type.
* The details should be saved in a JSON file.
* If there’s an active fast already, display the information about the current running fast instead of starting a new fast.
* There should be validation for the input values from the user and display any error messages if there’s an error in the user input.

## **3. End an Active Fast**

#### **Description**

As a user, I would like to end a fast so that I can save the current fast time. This option should be available only when there is an active fast.

#### **Acceptance Criteria**

* When the “End an active fast” option is selected.
  + The action should change the fast status to inactive.
  + The details should be updated in the JSON file.
  + If there’s no active fast, print a message to inform the user.
  + There should be validation for the input values from the user and error messages displayed if validation fails.

## **4. Update an Active Fast**

#### **Description**

As a user, I would like to change fast so that I can change the starting time and type of an active fast.

#### **Acceptance Criteria**

* There should be a command that will update the active fast, which will ask the user for input for the following details:
  + Start Date (when the fast started)
  + Fast Type (one of the possible hour values: 16, 18, 20, 36)
* If the fast type is changed, it will automatically recalculate the end date on the fly.
* The details should be updated in the JSON file.
* After each change, print a successful message to the user that the fast was updated.

## 5. List all fasts

#### **Description**

As a user, I would like to see a list of fasts so that I know when I fasted previously.

#### **Acceptance Criteria**

* There should be a command that will list of all fasts which will display the following details for each one of them:
  + Status (Active / Inactive)
  + Started Fasting (the start date when the fast started, example April 1, 12:00)
  + Fast Ending (the expected date and time, example April 2, 20:00)
  + Elapsed Time (the current elapsed time for the current fast, example 1:30:45)
  + Fast Type (The type of fast, example: 16 hours)
* The information should be displayed from the JSON file.
* If there are no fasts at the moment, display a message about it.