Levelset Coding Exercise Challenge

The code challenge is a small project for you to demonstrate your ability to understand requirements and implement a solution using PHP or similar scripting language in a fixed timeframe of two days.

For this task, we're asking you to import a file of some data, create objects based on each row, save the results, do some calculations on the objects to calculate some numbers, and output reports.

Project Goals

- Import a CSV file of project data.
- Create projects and orders from import.
- Calculate deadlines for when an order needs to be executed.
- Output information about upcoming orders and their deadlines based on parameters passed in during import.

Exercise Evaluation Criteria

In addition to reviewing your code, your output will be evaluated using the following criteria:

- Properly understanding the project requirements and implementing the correct solution.
- Using OOP & software engineering concepts to implement an object oriented solution.
- Placing different pieces of functionality in the correct layers of the system.
- Solving problems and understanding code without relying too much on others.
- Selecting the right data structures.
- Solution robustness and covering corner cases.
- Code quality, for example code readability and simplicity.
- Ability to finish usable features on time.
- Choosing a sufficient amount of inline or external comments for elaboration

High Level Requirements

The basic idea is to build a simple feature over an existing MVC web application with minimalistic interface complying with proper software engineering and design concepts. If you do not decide to use PHP, you may pick a similar scripting language and MVC framework or library, which may take more time than using the sample code we provide.

The code of the existing application resides here.
https://drive.google.com/file/d/1vM7UX2ApuWptNX3Pjs1FRiHec6hrxnpy/

- Create a new controller in which the user can create an array of objects by importing data from files.
- Design and implement required classes to import a CSV file and parse its data (**no 3rd party libraries**) with proper error handling and properly placed in the architecture to be modular and reusable.
- Design and implement the proper models to make the system work with minimal interface and views.
- Output information about upcoming orders and their deadlines based on parameters passed in during import.
- Do some logic on the models using an algorithm based on some predefined rules described below.

- Bonus: Use SQLite DB to save the orders data.

Requirements

- Projects
  - Are uniquely identified by their Address, City, State, and Zip Code.
  - Only one unique project may exist in the system at one time.
- Import
  - The projects CSV file may have multiple entries for the same project.
  - The non-unique components of a project must be set to the last instance of that project in the import file.
  - If Project Commencement Date is missing or invalid, use today's date.
  - If a required field is missing, do not create the project nor the order.
- Project Orders
  - There are 2 Order Types applicable to projects: "Notice" and "Lien".
  - Each order has a deadline by which it has to be executed.
- Order Deadline Calculation
  - The deadline for Order Type "Notice" is 60 days after the last day of the Project Commencement Date's month.
  - The deadline for Order Type "Lien" is 90 days after the last day of the Project Start Date's month.
  - The deadline for Order Type "Notice" in Texas is the next 15th day of the month after the Project Commencement Date.
  - Deadlines must not be on a weekend (Saturday or Sunday) or a mailing holiday (see table below in the references). If the deadline falls on a weekend or holiday, deadline is the first non-weekend or holiday day before that day.
- Orders with Deadlines in the past must not be created or displayed in the output report.
- Output Report
  - List of all projects from the import with calculated order deadlines. Output the following fields:
    - Customer Name
    - Project Name
    - Order Type
    - Deadline Date
  - Bonus: List of customer outstanding debt:
    - Customer Name
    - Total Outstanding Debt
    - Total Number of Orders
    - Total Number of Projects
  - The user can specify what report they want to see after the import.
- Bonus: Save data
  - Save the imported data in an SQLite Database and create a report that can list saved data.

Deliverables

- The modified PHP code (and SQLite database if available) or code to import CSVs file described and create projects and orders that meet the requirements above; you may modify the provided classes if needed.
- The final output should have the following web pages:

    - A page for the user to upload the CSV file
    - A page to display reports
    - Bonus: A page to display saved data
    - You don't need to spend any time making the interface look pretty. Use basic web forms with no styling.
- CSV file including list of test cases used to evaluate solutions.
- Documentation on how to setup environment to test

Considerations

- Please timebox your work on this challenge and make sure that you deliver a usable output by the end.
- Please consider the goals and the evaluation criteria of this challenge when designing and implementing code to make sure that you are on the right track.
- Please consider that testers will try to break your implementation logic, so make sure special cases are covered.
- Please send any questions you have about the requirements to: coding-exercise@zlien.com
- The code provided is portable and has a built-in application server with PHP engine that runs on linux. So no packages should be required other than the ones described in the requirements below.

References

CSV file fields

Name	Data Type	Required
Customer Name	String	Y
Project Name	String	Y
Project Address	String	Y
Project City	String	Y
Project State	2 character string	Y
Project Zip	5 digit integer	Y
Project Start Date	String, format: YYYY-MM-DD	Y
Project Outstanding Debt	Float, 2 decimal places	N
Project Commencement Date	String, format: YYYY-MM-DD	N

Deadline Holidays

Name	Date
New Year's Eve	12-31

National Doughnut Day	06-02
Thanksgiving	11-24
Second Friday in December	(example: 2016-12-09)

System Requirements

- PHP 7.*
- Composer
- `php-cli`
- `php-xml`
- SQLite
- `php-sqlite3`

Installation and Usage

- Using the terminal, navigate to the project directory and install dependencies using composer:

  ```
  composer install
  ```

- To run the application server run the command below from the directory of the provided code which will run a full built-in application server as a daemon:

  ```
  php bin/console server:start
  ```

- By default, it will use the port 8000, and the built-in application server will serve requests at http://localhost:8000/ , and you can check a hello world page included at http://localhost:8000/hello/.

- To stop the application server, run:

  ```
  php bin/console server:stop
  ```