Senior PHP : Laravel
Round: Programming

Challenge:
A development team is working on a software application milestone which have following tasks.

TASK_ID	TASK_NAME	START_DATE	END_DATE	DURATION	DEPENDANCY	(Description)
1	SRS_CREATION	2019-01-01	?	12 DAYS	null	Task can start without any lag
2	WIREFRAMING	?	?	12 DAYS	TASK 1 Start Date + 4 Days	TASK can only start 4 days after the dependant task is started
3	UX_DESIGNS	?	?	12 DAYS	TASK 1 End Date + 0 Days	TASK can only start 0 days after the dependant task is finished
4	UI_HTML	?	?	12 DAYS	TASK 3 Start Date + 4 Days	TASK can only start 4 days after the dependant task is started
5	ALGORITHM design	?	?	12 DAYS	TASK 3 End Date + 0 Days	TASK can only start 0 days after the dependant task is finished
6	CONCEPT_SIGNOFF	?	?	3 DAYS	TASK 1 End Date, Task 2 End Date, Task3 End Date, Task 4 End Date, Task 5 End Date + 4 Days	TASK can only start after all the dependant task are finished considering lag aftre finish if any
7	DATABASE_SCHEMA_DESIGN	?	?	12 Days	TASK 1 End Date + 0 Days	TASK can only start 0 days after the dependant task is finished
8	SOFTWARE DESIGN PATTERN FINALIZATION			5 DAYS	NULL	Task can start without any lag
9	CREATING MODEL REPERSENTATIONS IN DATA MAPPER PATTERN	?	?	7 DAYS	TASK 7 Start Date + 4 Days	TASK can only start 4 days after the dependant task is started
10	ROUTING DEFINITIONS	?	?	12 DAYS	TASK 8 End Date + 0 Days	TASK can only start 0 days after the

						dependant task is finished
12	MODULE DEVELOPMENT	?	?	35 DAYS	TASK 10 Start Date + 4 Days	TASK can only start 4 days after the dependant task is started
13	UNIT TEST DEVELOPMENT	?	?	35 DAYS	TASK 12 Start Date + 0 Days	TASK can only start 0 days after the dependant task is started
14	BUILD CREATION	?	?	2 DAYS	TASK 6 End Date, TASK 13 End Date	TASK can only start after all the dependant task are finished considering lag aftre finished

Help the development team to assess / plan the milestone schedule.

Solutions Needed:

1. Develop a re-usable scheduling library / class which can calculate the dates based on supplied task and dependancy array.
2. Develop an unit test case which asserts the programm.
3. Create a boilerplate / basic frontend interface which displays

 3. (a) A blank schedule as provided aboove in the task table and a button to generate schedule / calculate schedule.
 3. (b) An updated schedule based on the dates calculated using the scheduling library
 3. (c) An option to edit the start date of TASK 1, and on submit re-calculate the updated schedule
 3. (d) Maintain the versions/revisions of the complete schedule each time it is updated

Technologies to be used

1. Programming Language - PHP (Laravel or Lumen Framework preferrbally)
2. Database MySQL / PGSQL / MongoDB (whichever suites best or a combination of a relational / nosql as per requirement)
3. Frontend - Bootstrap (HTML , CSS, JS/Jquery) - which comes as a boilerplate with default laravel.

Tips

- Focus on an effective and optimized program.
- Do not put more efforts on frontend / css.

Hints (Skills to be tested)

- Use of OOPS (Interfaces / traits), appropriate data types, data structures and algorithm used to solve the problem.
- Unit Testable Code
- Error / Exception handling
- MySQL Knwoledge