

# CRUD Application - Laravel and React

---

## Some guidance

1. Use Laravel and React to solve this challenge. Code must be working.
  2. A proper combination of SOLID principles and design patterns with appropriate use can help you resolve the problems faster.
  3. Data doesn't must be saved in DB, you can save it memory on the server.
  4. When you're done, send us a zip file with the code and DB creation scripts (if done). Don't need to send all packages.
  5. Comment your code so that when you look back at it in a year, you'll remember what you were doing.
  6. There are many ways to approach and solve the problems presented in this exercise.
  7. Have fun!
- 

## The Task

A factory has employees in different departments,

We need to build a system to help manage employees, so we need screens that allow:

- ✓ A manager to add / update / delete employees.
- ✓ See a list of employees.
- ✓ Calculate their salary as part of the list of employees.

Employee details are:

#	Field name	Type	Size	Required	Note
---	------------	------	------	----------	------

1	First name	Text	70	Yes	
2	Last Name	Text	70	Yes	
3	Phone	Text	15	No	
4	Email	Text	50	No	
5	Work hours	Float		Yes	in the last month
6	salary type	Number		Yes	
7	salary	Number		Yes	Meaning depends on the type of salary
8	Department	Number		Yes	

- ✓ How to calculate the salary on display:
  - Type 1 - by hourly wage.
  - Type 2 - fixed each month (value of field salary).
  - Type 3 - If more than 100 hours then full pay otherwise 0.75% of pay per hour.
- ✓ Each employee can be in only one department.
- ✓ The data type in the feed must be verified and an appropriate error displayed if the feed is incorrect.
- ✓ There is no need for a UI to update departments and pay types.