

Microblog Task

Project requirements

You have to implement a microblog that has a simple but effective admin panel where the admin users can log in and publish/manage posts:

- Every published post, its title, content, and featured image have to be visible on the public side of the blog.
- Every post has to be manageable via the admin panel. The following should be editable: title, content, single/featured image attachment.

Technical requirements

- Back-End
- PHP ≥ 7.3
- MySQL 5.7
- Slim Framework 3.0

Front-End

- Bootstrap 4.0
- An advantage is to use React. Otherwise, you can use Angular, jQuery, or server-side rendering via Twig or something else.

Technical implementation guidelines:

Setup

- Use Composer for installing the framework and the additional packages (if there are such).
- Install the Slim 3.0 framework.

- Use NPM for the front-end unless you use jQuery that can be loaded directly from its CDN.

Back-End

- Implement an MVC layer. There should be a diversified logic/structure between models, views, and controllers. Optionally, you can use services to ease and decouple the code.
 - Create a User model & Controller, handling the users' Requests and CRUD operations.
 - Create a logic (class, service, middleware, or all of them in combination) to handle user authentication.
 - Create a Post model & Controller, handling the posts' Requests and CRUD operations.
 - Create a service responsible for the post's image upload/management.
- For the database, you can use ORM, it's an advantage to develop a custom class that handles all the DB queries.
- Implement PHPUnit tests covering the functionalities listed above.

NOTE: *The main idea of that task is showing us that you know how to build an application from scratch, creating an MVC structure that follows the best practices, etc. Therefore, keep your code DRY, with high quality, and do not copy already implemented MVC solutions for Slim or other.*

Submit the task

It is required to upload the code to a public/private GitHub repo or archive the git folder of the project by sending it to us via email.

- Git records (commits) should be grouped in the following way:
 - All subsequent commits should be logically organized, reflecting the steps you've taken developing the application.
 - Neither one large commit with all changes nor a multitude of smaller commits for every little tiny change.
 - Also, the task should have a simple setup guide. If you like, you can use the README for that purpose.