

----- Анкета -----

1) Укажите полезные источники по laravel и другим темам

<https://laravel.ru>

<https://laravel-news.com/your-first-laravel-application> - базовый tutorial

<https://laravel.su/docs/5.2/structure> - структура проекта

<https://habr.com/ru/post/334776/> - экосистема ларавель

habr.com/ru/post/354036/ - 20 Eloquent ORM трюков

laravel-news.com/firstornew-firstorcreate-firstor-updateorcreate - разница этих методов

tutsmake.com/laravel-many-to-many-polymorphic-relationship-example/ - Как создать полиморфный many to many

<https://webdevetc.com/blog/laravel-features-you-may-not-know-about>

<https://laracasts.com/series/learn-vue-2-step-by-step> - Vue + Laravel

<https://scotch.io/tutorials/build-a-to-do-app-with-vue-js-2> - базовый tutorial по Vue

2) Управление проектом/задачами/исполнителями: что получилось / что не получилось / что изменить ?

Управление в целом получилось, но из минусов можно отметить позднее включение участников в проект. Управление исполнителями затруднялось отсутствием практических навыков во фрейворке.

Участники долго не могли развернуть проект, поэтому хотелось бы добавить урок по развёртке локального окружения.

3) Сбор, анализ и формализация требований: что получилось / что не получилось / что изменить ?

Сбор требований забрал много времени, хотелось бы общий документ с перечисленным списком базовых требований.

4) Прототипирование: что получилось / что не получилось / что изменить ?

Проблем не возникло

5) Моделирование структуры данных: что получилось / что не получилось / что изменить ?

Проблем не возникло, кроме организации хранения истории.

6) Макетирование интерфейса: что получилось / что не получилось / что изменить ?

Проблем не возникло

7) Моделирование бизнес-процессов: что получилось / что не получилось / что изменить ?

У участников, которые делали их, возникли проблемы с пониманием что вообще должно быть на диаграммах состояния.

8) Моделирование состояния ключевых сущностей: что получилось / что не получилось / что изменить ?

У участников, которые делали их, возникли проблемы с пониманием что вообще должно быть на диаграммах состояния.

9) Архитектурные решения: что получилось / что не получилось / что изменить ?

На мой взгляд получился хороший MVC проект, потому что: модель, а не контроллер отвечает за изменение своих состояний, директория со вьюхами логически организована, в моделях организована связь с другими моделями не длинными джоинами, а встроенными средствами ORM.

10) Конструирование программы: что получилось / что не получилось / что изменить ?

Все получилось без костылей, без перегруженного кода на бэке, так как имеется опыт в конструировании laravel приложений и знание некоторых фиш ларавеля, которые помогают значительно уменьшить код.

11) Совместное владение кодом: что получилось / что не получилось / что изменить ?

Проблем с ведением репозитория не было. Однако, из-за незнания как организован гитхаб, один из участников скачал архив проекта и инициализировал свой локальный репозиторий, не мог понять где удаленные ветки. Хотелось бы, чтобы для людей без опыта пояснялось, что репозиторий надо клонировать.

12) Тестирование: что получилось / что не получилось / что изменить ?

Все получилось

13) Отладка и исправление ошибок: что получилось / что не получилось / что изменить ?

Все получилось, но полезной для других была бы установка <https://github.com/barryvdh/laravel-debugbar>

14) Развертывание конечного приложения: что получилось / что не получилось / что изменить ?

Все получилось на удивление легко

15) Время - как ресурс: что получилось / что не получилось / что изменить ?

Было сложно смотивировать участников на работу, потому что на других предметах гоняли больше.

N) Разное: что получилось / что не получилось / что изменить ?

Я считаю, что Laravel достаточно простой для понимания фреймворк, поэтому он хорошо подходит людям без опыта. Наличие ORM очень сильно ускорило разработку. Возможно, было бы лучше, если провели какой-то гайд по структуре проектов на ларавеле, потому что участники не особо понимали как, например, роутинг связывает контроллеры и вьюхи.