Проблемы

**База данных**

Избыточность таблиц в базе данных (хранение статичных данных типа options для select, всякие характеристики и т.д) (нужно хранить что-то динамическое)

Хранение в базе неатомарных данных (массивы, json и т.д.)

Хранение в базе html/css/php кода (слишком много мест генерации кода, поэтому не понятно куда смотреть чтобы что-то изменить)

Отсутствие (нативных) связей в базе (нарушается целостность данных, возникает дублирование данных, есть вероятность удаления нужных данных)

**PHP**

Паттерн MVC реализован криво

Правила паттерна не соблюдаются (куча логики во view)

Все делается через один контроллер (все остальные контроллеры вообще по всей видимости не контроллеры)

Дублирование кода

Плохой уровень безопасности (как минимум sql запросы не безопасны)

Не вложена система ролей

В целом архитектура странная (куча кода в разных местах)

Очень много обращений к базе в процессе генерации страницы

Непонятно зачем админка если все уже реализовано во фронте зачем копировать то же самое.

**HTML/CSS/JS**

CSS и JS часто внедрен прямо в html код (Все должно находиться раздельно чтобы код был читабельным и его легко было рефакторить и чтобы не было дублирования кода)

По какой-то причине поля для форм генерируются на лету, хотя можно было сделать их статическими как обычно делают (это же не конструктор сайтов все-таки)

В итоге если у нас есть 100 полей то есть и сто папок для этих полей плюс сложная система генерации

Каждый раз когда возникает необходимость формы лучше каждый раз писать статичный код со статичными полями

В результате если у нас 10 форм с 10 полями в каждой у нас получается 10 файлов в которых если надо можно что-то легко отредактировать

Плюс генерация каждого поля занимает много времени т.к. для генерации каждого поля выполняется запрос к базе данных

**ДИЗАЙН**

Не интуитивный, нелогичный

**В ОБЩЕМ**

Все выше описанное пагубно влияет на:

* Производительность
* Целостность данных
* Безопасность
* Удобство конечного пользователя
* Возникновение багов
* Изменение существующего функционала (долго, трудозатратно)
* Разработку нового функционала (долго, трудозатратно)

**Решение**

**База данных**

Я спроектирую качественную базу со всеми связями, тем самым избавлюсь от вышеперечисленных проблем

Знания имеются (из интернета, из универа)

**PHP**

Использую фреймворк

Вот почему:

* Разработка на фреймворке (в отличие от самописных решений) позволяет добиться простоты сопровождаемости проекта.
* Система стандартизирована и понятна
* Быстрее, безопаснее
* Есть готовые решения

Фреймворк предоставляет готовую удобную проверенную временем архитектуру приложения

В основе архитектуры лежит паттерн MVC

Так же он предоставляет готовый функционал в виде регистрации, аутентификации, авторизации (так же на основе ролей), пагинации, навигации, ActiveRecord (удобнейшая абстракция для работы с базой данных), быструю настройку rest full api и т.д.

Решено разделить приложение на бэк и фронт (Это точно решит проблему сложности системы и читабельности кода)

Не будет месива из php/html/css/js в одном файле и даже в одном проекте

Это очень удобно поддерживать, изменять, дополнять

На php yii2 сделаю только бэкенд т.е. только API

На фронте будет работать vueJs

API так же позволит в перспективе работать с базой данных из любого другого приложения (хоть айос хоть андроид хоть любой другой проект)

**HTML/CSS/JS**

Vue js решит все вышеперечисленные проблемы

Компонентный подход не даст запутаться в коде и избавит от дублирования кода

Во Vue пишем только представление (только фронтовая логика)

Все будет работать без перезагрузки страницы

Vue быстрый js фреймворк так что все будет работать быстро

**Дизайн**

Сделаю интуитивно понятным и логичным, чтобы было с первого взгляда понятно куда нажимать и за что эта кнопка отвечает

**В общем**



**О чем я не знаю**

Как реализована генерация ПДФ

Что происходит с циан и всем таким

Какие еще сервисы интегрированы

**Минусы**

Время: Около 8 месяцев

Стоимость: Около 1 миллиона

А еще жалко функционал который делался два года