Задание

Необходимо создать небольшое приложение на Yii 1,2 Framework,Laravel 5,6. Желательно оценить свое время на реализацию задачи и прислать это время ответным письмом.

Приложение для создания заявок.

На входе мы должны иметь форму с несколькими полями для ввода информации.

Необходимые поля:

* Название заявки
* Тип заявки (Сервисное обслуживание, Поддержка, Запрос технической информации)
* Приоритет заявки (Низкий, Высокий, Средний)
* Описание заявки
* Контактные данные (тел. Почта, ФИО)

Заявки и пользователи должны быть в разных таблицах и связаны в моделях.

Форма доступна без регистрации и видна на главной странице. Обязательные поля на усмотрение разработчика.

Со стороны админской панели нужно реализовать:

Обязательно

* вывод всех заявок с возможностью сортировки по статусам и типам
* возможность редактировать заявки
* возможность добавлять комментарии к заявкам при их просмотре/редактировании
* статус заявки (новая/в работе/ выполнена/отложена/решена)

На усмотрение разработчика реализовать (одну из фитч, либо все):

* Возможность учитывать время на выполнение заявки(поле для записи времени при добавлении комментария) вписывается руками по формату “ЧЧ: ММ”
* Дополнительное второе поле “время работы” с заявкой вписывается автоматически по правилу (если ведется какая-то работа в заявке, она открыта, что-то пишется в поле комментарий, то время считается автоматически и в поле вписываетсявремя, учтенное системой в автоматическом режиме)
* Если заявку открыли 2 пользователя, то первому дать права полные на редактирование, а второму и далее вывести сообщение, что с заявкой работает такой-то пользователь. И не давать редактировать заявку.

БД можно использовать SQLite.

Приложение должно быть собрано на PHP 7+, использовать composer, приложить маленькую инструкцию как его запустить локально. Код разместить на Github || Bitbucket || Gitlab и предоставить ссылку на репозиторий. Придерживаться PSR-2 при написании кода, использовать миграции для БД.

Дополнительно описать текстом:

1. Описать как сделать очередь заявок и их распределение по сотрудникам компании. При условии, что у каждого сотрудника есть загрузка (равна 100), заявки имеют вес (от 10 до 30), сотрудник может обрабатывать 10 заявок с весом в 10 или 3 весом в 30. Сотрудников 13, заявок может быть до 1000 в сутки, или более. Время на выполнение заявок по приоритетам, 10 – 5 минут, 20 –10 минут, 30 – 15 минут. Кол-во сотрудников 17 человек. Будет ли вероятность того, что за сутки 1000 заявок не будет обработано и почему?

**Ответ.**При окончании обработки заявки сотрудником вычитать ее вес из пула загрузки сотрудника, новая заявка должна не превышать или единожды превысить загрузку менеджера. Заявки выдавать в случайном порядке или по направлениям (типам заяки).  
При 17 работниках в отделе ТП и большинстве заявок с весом 20 и 30 большая доля вероятности, что будут обработаны не все заявки. Все зависит от веса заявок.

1. Как можно учитывать время на работу с заявкой в автоматическом режиме? Нужно для того, чтобы понимать реальное время, затраченное на заявку сотрудником и сравнивать со временем, которое ввел сотрудник после закрытия заявки.

**Ответ.**Нужно ввести метки времени открытия и закрытия заявки на основе ID сессии менеджера.  
Что я и продемонстрировал в web-проекте по данному ТЗ.

Проект сделан на стеке Apache 2.2, PHP 7.2, MySQL 5.6, Laravel 6.2.   
После установки проекта на локальную машину необходимо:   
1. Выполнить миграции БД- из консоли в корне приложения выполнить команду   
 php artisan migrate

2. Выполнить начальную загрузку данных в БД   
 php artisan db:seed --class=TicketPriorityTableSeeder  
 php artisan db:seed --class=TicketStatusTableSeeder  
 php artisan db:seed --class=TicketTypeTableSeeder.php  
  
Пользователи (клиенты) создаются автоматически при добавлении заявки, имена, телефоны, email генерируются случайным образом. Клиентская форма проходит валидацию. Сделана возможность оставлять комментарии к заявке, менять ее статус. После открытия заявки менеджером она блокируется для редактирования других менеджеров (но остается им видна со всей информацией). Идет фиксация времени работы с заявкой пока менеджер не нажмет на кнопку «Изменить». Перезагрузка страницы или подмена адреса URL не даст менеджеру сбросить таймер, он будет возвращен на данную заявку. По окончании редактирования вычисляется время работы с заявкой, это время плюсуется к общему времени связанному с этой заявкой. Поле ручного ввода времени не стал добавлять по соображениям автоматизации процесса.