**Название:** D-lab admin

**Проблемное поле:** Сфера зуботехники является относительно молодой сферой. С каждым днем открываются все больше зуботехнических лабораторий, но владельцы лабораторий сталкиваются с проблемой - учетом заказов. Большинство владельцев ведут учет вручную бумажным способом, что не очень рационально, pтак как в конце отчетного периода очень сложно посчитать общую сумму выполненных работ, определить сумму для клиник, посчитать зарплату сотрудников. В процессе расчетов, велика вероятность человеческой ошибки, что ведет к денежным убыткам. Некоторые владельцы ведут учет с помощью специализированного ПО ("ПО ЗТЛ"), но оно имеет ряд недостатков, в числе которых:

1. Доступно только для ОС Windows
2. Высокая стоимость продукта
3. Отсутствие push-уведомлений о сроках заказа
4. Отсутствие отслеживания заказов врачами - заказчиками
5. Отсутствие возможности составления плана работы для курьера лаборатории
6. Отсутствие отслеживания доставок курьера лаборатории
7. Устаревший интерфейс

Помимо "ПО ЗТЛ" программ для учета найти в открытых источниках не удалось

Этот проект я решил реализовать, так как летом работал в зуботехнической лаборатории, в которой учет велся бумажным способом. Возникали такие проблемы, как:

1. Оформление счетов в конце месяца: в конце месяца работы, сделанные за месяц, отправляются бухгалтеру, который в свою очередь выставляет счета клиникам. Чтобы собрать все эти работы, приходилось весь месяц записывать их в одном месяце, потом каждый раз сверять что было сделано в каждой из них, считать стоимость общую для каждой клиники или врача. Это требовало большого количества времени и при исправлении каких – то ошибок в записи работ приходилось все пересчитывать (вручную на калькуляторе). Очевидно, если таких работ больше 100 – ни о какой точности и систематизации в учете речи быть не может.
2. Расчет зарплат в конце месяца: зарплаты в зуботехнической лаборатории не фиксированные, а рассчитываются в зависимости от количества работ, выполненных зуботехником. Ситуация с зарплатами была ровно такая же, как с счетами – отсутствие системы учета вело к большому количеству ошибок в счете, которые невозможно отследить.

Как я уже упоминал раннее, есть программа для учета в зуботехнической лаборатории, но пользоваться ей я не стал по ряду причин, перечисленных выше.

**История работы над ИВР** – Вначале я заявил проект ЕСР – Единая Система Регистрации (на олимпиады). Проектом занимался в момент, когда нужно было сдать что – либо к дедлайну. Будучи незаинтересованным в нем, я не начинал его полноценную разработку. Но занявшись учетом в лаборатории передо мной встала вполне реальная задача, решив которую и создав веб-сервис d-lab, я смог бы его использовать в реальной компании. Далее я выделил несколько аспектов, которые должны присутствовать в проекте, максимум упростить учет в лаборатории. Затем по ним я начал саму разработку.

**Заказчик или целевая аудитория** – виза заказчика на последней странице.

**Описание продукта** – На данный момент проект в проекте реализовано ядро, необходимое для учета в лаборатории. То есть можно добавлять различные справочные материалы (врачей, клиники, материалы, работы, операции, техников, прайслисты работ и операций), добавлять заказы (используя справочники), можно вести склад, получать два отчета (по сделанным работам и зарплатам техников). Все заявленные юзер – кейсы реализованы.

**Рефлексия** – В процессе работы я научился работать с Django Rest Api Framework. Благодаря этой технологии можно реализовать полностью независмый бекенд и использовать его как для веб-приложений, так и для мобильных. Не удалось реализовать пуш – уведомления. Причиной тому является, что пока я собираюсь использовать проект локально, то есть запускать на localhost, а запуская проект таким образом нельзя использовать пуш-уведомления (push – уведомления работают только с https серверами). Пока не реализована функция просмотра заказа врачами, так как интегрировать в самом конце разные типы аккаунтов оказалось затруднительным.

**Как можно дальше развивать проект** – Далее я вижу развитие проекта таким:

1. Весь интерфейс, готовый к защите, тестируется на факт багов, и уязвимостей.
2. Далее добавляются доп. функционал:
   1. Push-уведомления (или sms-оповещения).
   2. Отслеживание заказов врачами из вне.
   3. Автоматические списание материалов в заказе.
   4. Добавление Авансов (в платежах).
   5. Экспорт отчетов (например, в excel).
   6. Улучшение api (уменьшение запросов при добавлении заказ-наряда, добавлении операций к работам и т.д.)
   7. Выпуск в Production

