План

**Система курсов**

Система курсов включает в себя курс, курс включает в себя дни, дни включают в себя задания. Каждый новый день открывается ученику при условии выполнения и одобрения всех заданий предыдущего дня. Первый день в курсе открыт всегда.

Требуемые сущности:

1. Список курсов. Каждый курс включает в себя:
   1. Фотография курса
   2. Название курса
   3. Прогресс курса. Рассчитывается на основании пройденных заданий от общего количества заданий.
2. Список дней. Список дней формируется на основании выбранного курса. Каждый день включает в себя:
   1. Фотография дня. Опционально, при отсутствии используется фотография курса.
   2. Название дня
   3. Прогресс дня. Рассчитывается на основании выполненных заданий дня относительно общего количества заданий дня.
3. Список заданий. Список заданий формируется на основании выбранного дня. Каждое задание включает в себя:
   1. Фотография задания. Опционально, при отсутствии используется фотография дня, если у дня нет фото, используется фотография курса.
   2. Название задания
   3. Описание задания. Опционально.
   4. Прикрепленные файлы.
   5. Видео. Опционально.
   6. Статус задания. Не приступали, на проверке, выполнено (подтверждено куратором либо администратором), на исправлении (требуется доработка)

**Система пользователей**

Система пользователей включает в себя 3 роли:

1. Ученик. Имеет возможность проходить доступные ему курсы. Отправлять задания на проверку.
2. Куратор. Имеет возможность проверять задания у учеников в доступных куратору курсах.
3. Администратор. Имеет полный доступ ко всем курсам. Может добавлять и удалять курсы/дни/задани. Может проверять задания к учеников на любых курсах. Может добавлять/удалять в систему пользователей любых ролей.

Требуемые сущности:

1. Пользователь. Включает в себя
   1. Имя
   2. Фамилия
   3. Роль
   4. Email
   5. Пароль

Для реализации функционала требуется сделать следующее:

Со стороны дизайна:

1. Реализовать черновой макет по текущим требованиям
2. Согласовать структурную композицию сайта
3. Реализовать макет всего функционала на ПК
4. Реализовать UI kit и набор иконок
5. Реализовать макет всего функционала на мобильные устройства (планшет, квадратный монитор, смартфон, устройства типа Fold)

Со стороны интерфейса:

1. Проработать существующие решения, исправить удовлетворяющие элементы при наличии ошибок
2. Обновить интерфейс в соответствие новому макету сайта
3. Изменить систему принятия заданий
   1. Отказаться от системы чата
   2. Сделать систему проверки по списку с возможностью скачивания ответа ученика
   3. Сделать проверку в 2 кнопки принять/не принять
   4. В списке заданий ожидающих проверки реализовать фильтр
      1. По курсу
      2. По дню
      3. По ученику
4. Оптимизировать и отладить работу со списками дней, курсов, заданий.
5. Реализовать на административной панели функционал по удобному добавления курсов и наполнению их с боковой панелью для быстрого доступа.
6. Реализовать интерфейс куратора для быстрой проверки заданий.

Со стороны бэкэнда:

1. Изменить базу данных для отказа от системы чата (поле файла, ответ??)
2. Удалить API маршруты для работы с чатом (закомментировать код)
3. Протестировать и исправить все цепочки запросов курсов, дней, заданий (баги найти исправить)
4. Оптимизировать запросы для получения список курсов, дней, заданий
5. Реализовать необходимые API пути для обеспечения работы системы для куратора и обновленной системы проверки заданий
6. Реализовать систему управления ролями, ограничивать выдачу информации пользователям с разными ролями
7. Реализовать систему контроля выполнения заданий для отображения дней на курсе.
8. Отладить систему управления загруженными на сайт файлами
   1. Не ограничивать размер+ pages/api/to\_check/task\_id/answer.js
   2. Не искажать имена файлов <
   3. Контролировать удаления файлов сервера при удалении связанных с ним сущностей <
9. Реализовать систему оповещения в телеграмм при изменении статуса задания либо других событиях
   1. Создать бота со следующим функционалом
      1. При стартер бот запрашивает email
      2. При вводе email ищется пользователь в базе с таким email и его присваивается telegram id из его сообщения
      3. Если пользователь не найден сообщается об ошибке
      4. Если успешно, сообщается что теперь будут уведомления
      5. При совершении события пользователю по id пишет бот с информацией
   2. На сайте реализовать вызов api бота для отправки сообщений пользователям
10. Реализовать систему оповещения в почту при изменении статуса задания либо других событиях
    1. На сайте реализовать вызов функции отправки на почту письма с информацией при наступлении события
    2. Для корректной отправки сообщений, вероятно, понадобится использования почтового smtp сервера от хостинга

**Проведения вебинаров**

1. Поиск интегрируемых решений на платной и бесплатной основе
2. Тестирование пробных версий на 10 и более пользователях
3. Выбор решений
4. Дальнейшие действия по интеграции в зависимости от решения