***Техническое задание MY WEB(развёрнутое):***

1. **Фронт**:

◆ Главная(просто лого сайта и какая то инфа о разрабах, о проекте с тремя

кнопками - входа, регистрации ОУ, регистрация Работодателя, регистрация Студента

◆ Страницы логина(почта и пароль который хэшируется через Werkzeug)

◆ Страница регистрации, с выбором кто регистрируется: если работодатель

- то доступны поля:

* ИНН(проверяется через сторонний API) -> по нему подтягиваются:
* Наименование рабочего места(к примеру Кинотеатр Маяк)
* Адрес
* /1) имя 2) фамилия 3) отчество/ руководителя
* Номер телефона
* Сфера деятельности
* Почта(email)
* Пароль(в форме сам пароль, в БД заносится хэшированный)

Если образовательное учреждение - то:

* ИНН(проверяется через сторонний API) -> по нему подтягиваются:
* Наименование этого учреждения
* Адрес
* /1) имя 2) фамилия 3) отчество/ директора
* Тип ОУ(универ, гимназия, лицей, школа, онлайн-курс и тп.) - дроп-лист
* Почта(email)
* Пароль(в форме сам пароль, в БД заносится хэшированный)

Если студент - то:

* ФИО студента
* Дата рождения(дд.мм.гггг datetime)
* Номер телефона
* О себе

◆ Страницы пользования ОУ, работодателя и студента описаны(изображены) на картинке. Студент может смотреть и скачивать свои серты.

1. **БЭК:**

* Таблицы БД:
* Users:
* Id
* Email
* Hashed\_password
* Role(university, employer, student)
* Universities:
* Id
* User\_id (Foreign key to Users)
* Title
* Adress
* Boss’s NSP
* Type
* Employers:
* Id
* User\_id (Foreign key to Users)
* Title
* Adress
* Boss’s NSP
* Phone\_num
* Scope
* Students:
* Id
* User\_id (Foreign key to Users)
* Students’s NSP
* Born
* Phone\_num
* About
* Employer\_id (STRING, ids of employers, айди работодателей, которые лайкнули этого студента)
* Achievements:
* Id
* Token(уникальный хэш номер каждого достижения, по нему идёт проверка на странице пользования работодателя)
* File\_path (local)
* Description
* Start\_date (datetime)
* End\_date (пусто или datetime)
* Student\_id (Foreign key to Students)
* Uneversity\_id (Foreign key to Universities)
* Использование стороннего API: при регистрации ОУ, подтягивание названия

образовательного учреждения по .. используя ...

* Логирование в файл
* Обработчик формы авторизации:
* при входе:
* Проверяется емаил и пароль(переводится в хэшированный) и сверяется по базе данных(находится строка с введённым емаилом и проверяется пароль на совпадение). Если все удачно, то смотрится роль и исходя из неё выполняется поиск по соответствующей таблице filter(user\_id==айди аккаунта который мы нашли в таблице Users). После всего этого идёт redirect на нужную страницу: 1) в html передаётся название(от кого вошли) 2) запоминается роль и айди в соответствующей таблице для дальнейшего пользования страницей.
* При регистрации:
* Создаётся аккаунт в Users(email+hashed\_pass+role) и все необходимые поля записываются в соответствующую таблицу, привязывается аккаунт из Users, после чего по тому же алгоритму что и при входе идёт редирект на страницу пользования.
* Часть из логики:
* при зачислении на курс какого то студента в таблице Achievements создаётся строка, где заполняется description, start\_date, student\_id и uners\_id
* В форме выпуска с курса в дроп-листе доступны только те названия курсов(discription), которые являются открытыми, то есть не имеют end\_date по типу datetime(такое означает что курс не завершён)
* При выпуске с курса после Submit в соответствующую строчку таблицы Achievements проставляется hash\_id(генерируется рандомно), end\_date и с помощью кода из моего первого проекта(см. Его, там создавался pdf файл) - создаётся сертификат, состоящий из:
* Названия курса(description->Achievements)
* Наименования ОУ(Title->Universities(смотрится по univers\_id из Achievements))
* ФИО студента(Student’s NSP->Students(смотрится по student\_id из Achievements))
* Выдан <ФИО директора>(Boss’s NSP так же как и наименование ОУ)
* дата начала и завершения(start\_date, end\_date->Achievements)

После чего он сохраняется в нужную папку в проекте и его локальный путь записывается в Achievements в file\_path.

* При проверке сертификата на главной странице поиск идёт только по завершённым курсам.
* Работодателю высвечивается табличка завершённых курсов студента с введённым айди: идёт поиск по таблице Achievements по колонке student\_id, в найденных выбираются те, у которых есть дата окончания(завершённые) и считываются только названия курса и наименование ОУ. Эти две данные и посылаются в html в тэг <table>.
* Поиск у ОУ похожий, но смотрятся только те, у кого НЕТ даты окончания(открытые курса) и считываются название курса, дата начала и оценка(просто на рандом выставляется).
* Использование Rest-API(моя часть):
* Restapi для …