**Принцип работы проекта**

Это решение позволяет в реальном времени распознавать движение человека используя технологию websocket и openpose. Наша основная цель использовать проект для прокторинга. Когда учащийся студент сдает экзамен наша система запуститься и будет следить за движением студента. Например, при взгляде в разные стороны (направо, налево), то наша система распознает это движение и выдаст сообщение TurnHead, если же студент находится вне видимости камеры, то сообщение будет “No People”.

Изначально, мы создали WebUI для клиента, где устанавливается соединение через вебсокеты с сервером и передает ему поток данных (изображения), где в свою очередь питоновский файл запускает сокет и получает поток с клиента JS. Мы опубликовали WebUI в IIS сервере.

**Технологии при использовании приложении и IIS сервера:**

1. Используемая версию .NET 4.7.2 MVC. Оно является последней версией данной технологии.

2. Версия Web API контроллера 2.1, благодаря которому мы взаимодействуем с клиентской частью и запускаем питон файл.

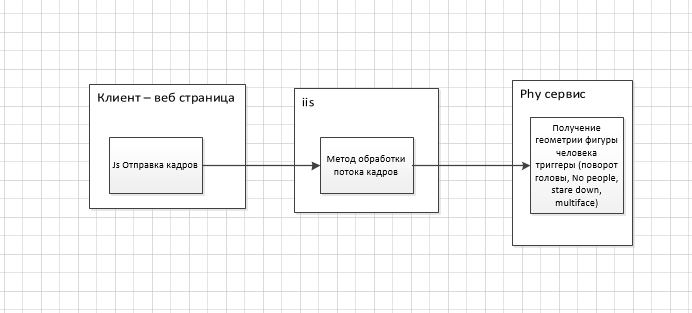
3. Версия операционной системы Windows 2016 – для публикации приложении

4. Linux Ubuntu 18.04 – для использование python службы

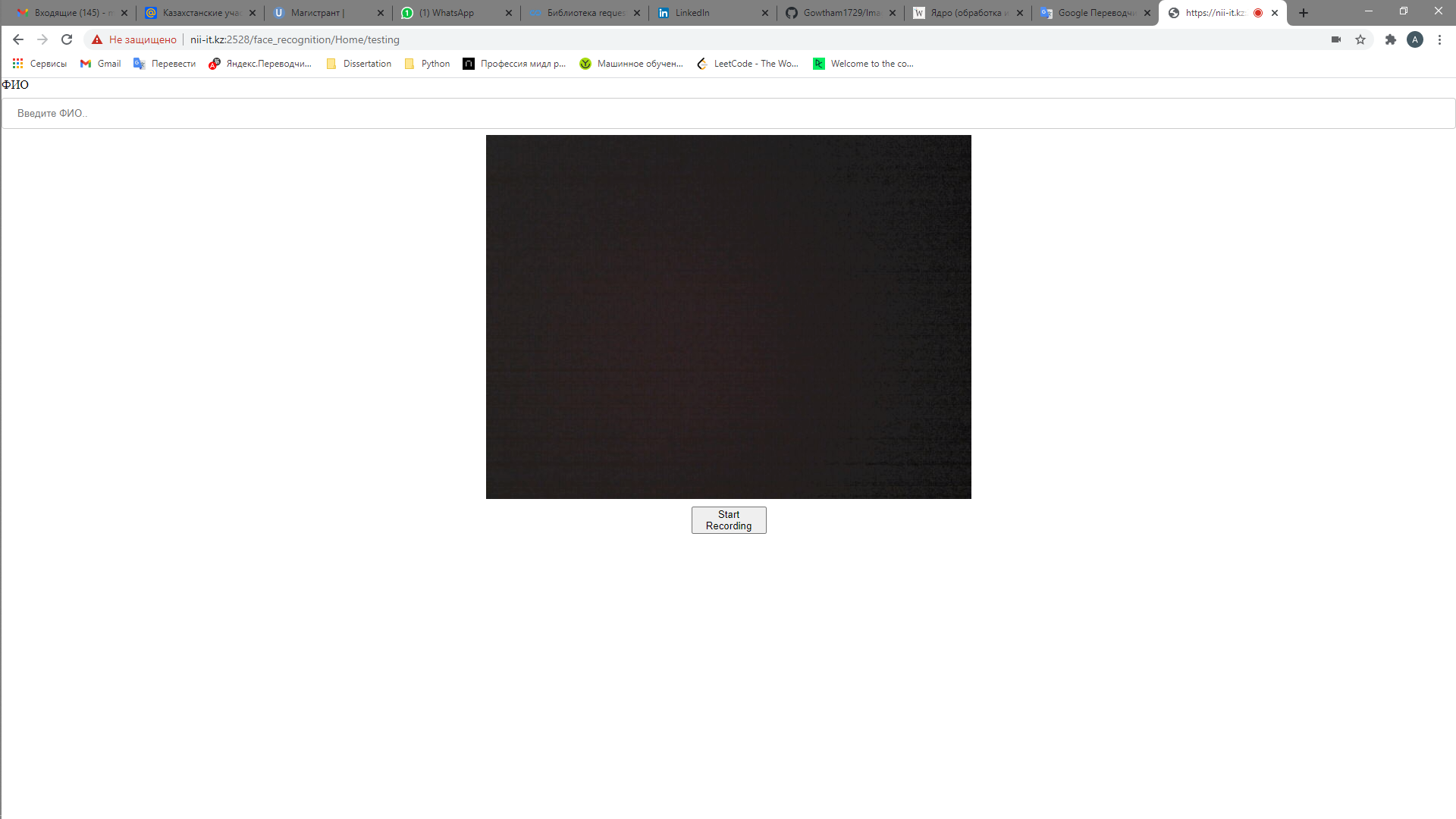
5.Версия Internet Information Service 8.5, где опубликовано наше приложение.

6. Доменное имя и ssl сертификаты также установлены на IIS сервере и привязаны к домену nii-it.kz

Архитектура

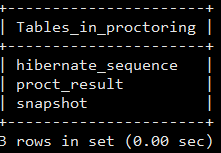


Веб приложение или WebUI выглядит следующим образом.

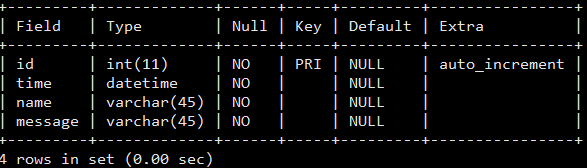


Результаты распознавания поз выглядеть следующим образом: если человек поворачивает головой, то выдают сообщение – Turn Head, если перед камерой нет человека, то – No People

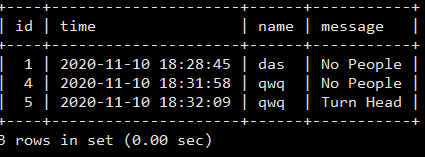
В Linux Ubuntu 18.04 мы создали таблицу в базе данных MySQL. Таблица называется proct\_result.



В этой таблице есть 3 колонки. Это: время, имя, сообщение.



Когда учащийся студент сдает экзамен наша система запуститься и будет следить за движением студента. Например, при взгляде в разные стороны (направо, налево), то наша система распознает это движение и выдаст сообщение TurnHead, если же студент находится вне видимости камеры, то сообщение будет “No People”. И эти сообщение сразу записываются в базу данных. Кроме этого, базу записываются Имя студента, время и результат сообщение о движении студента.



Описание python службы:

С помощью технологий Websocket происходит соединение между собой клиент и сервер, python файл и сервер. Клиент отправляет на сервер свое ФИО в виде строки и фото в виде массив байтов.

Сервер получив эти данные отправляет их на python файл. Питон файл получает эти данные, затем он преобразует массив байтов в кадры. Преобразование массив байтов в кадры происходит через этот код:

**# bufer – это массив байтов полученный от сервера**

**bufer = np.fromstring(bufer)**

**image = cv2.imdecode(bufer, cv2.IMREAD\_COLOR)**

**# image – это получившийся кадр из массив байтов**

Для того чтобы получить скелет человека проводим этот кадр через OpenPose (OpenPose представляет собой первую многопользовательскую систему в реальном времени для совместного обнаружения ключевых точек человеческого тела, рук, лица и стопы на отдельных изображениях). Результаты OpenPose записываются в базу данных MySQL в виде строки “Turn head” и “No people”. Вместе с этим в базу данных записывается ФИО клиента и время.

Рython файл подключается в базу данных локально. Подключение происходит через этот код с помощью библиотеки mysql.connector:

**mydb = mysql.connector.connect(**

**host="localhost", # локальный хост**

**user="proctoring", # имя пользователя**

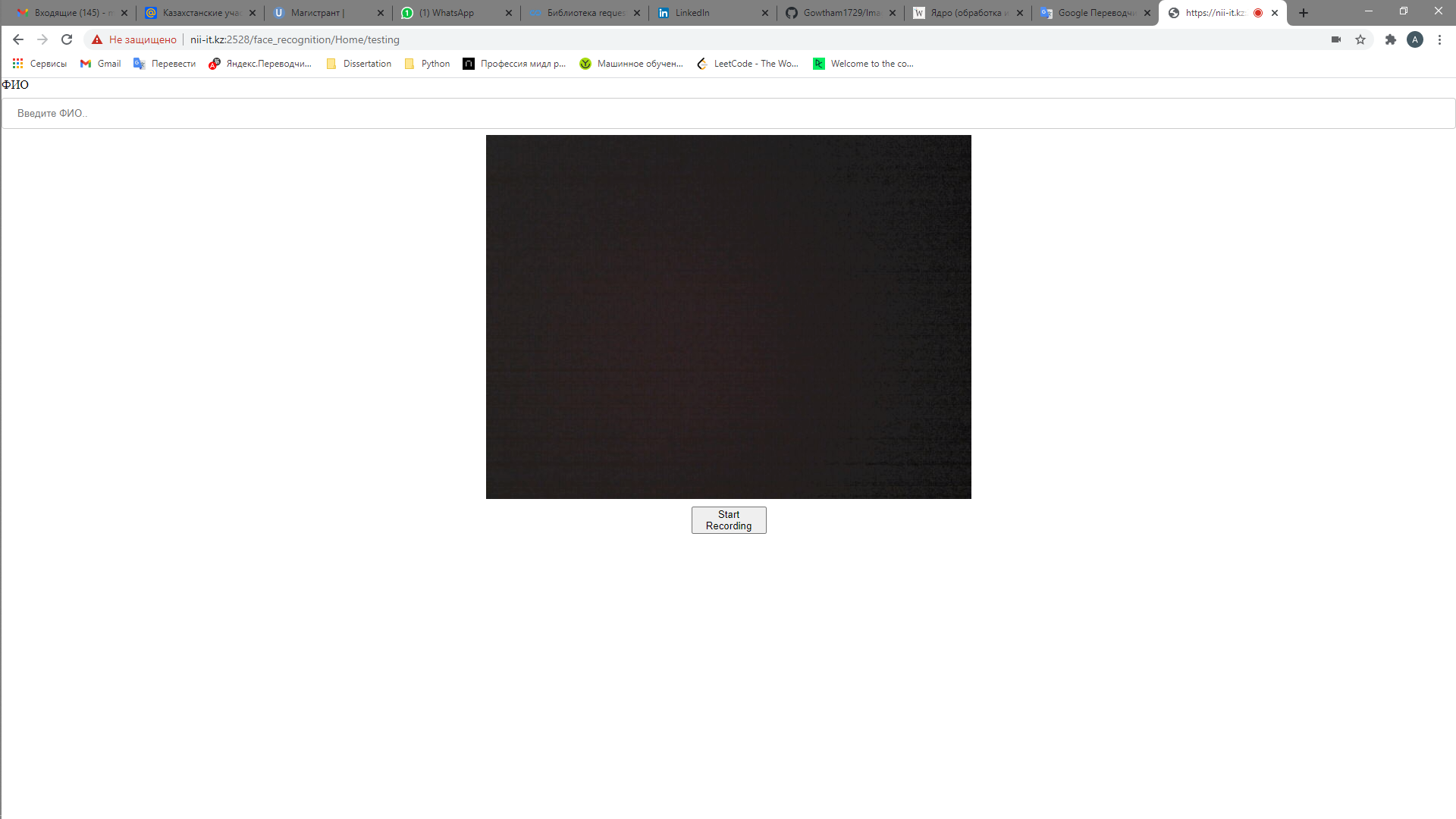
**passwd="123", # пароль**

**database="proctoring" # имя базы**

**)**

**Инструкция для использования.**

После входа на линуксовский сервер, мы запустили файл Posture.py который устанавливает связь с помощью вебсокета и принимает приходящий поток. По такому адресу мы заходим на наше web UI : <https://nii-it.kz:2528/face_recognition/Home/testing> и у вас появится такая страничка :



Нажав на Start Recording и поворачивая голову, либо уйдя с камеры мы получаем соотвествующее сообщение. Все полученные сообщения мы можем увидеть на питон службе. Конечный вывод:

