

РАЗРАБОТКА ФУНКЦИОНАЛА МОДУЛЯ АСИНХРОННОГО ПРОКТОРИНГА

К.Г. Виноградов
Научный руководитель: А.С. Фадеев
Томский политехнический университет
kgv1@tpu.ru

Введение

В настоящее время дистанционное обучение занимает важную роль в системе образования. С развитием новых цифровых технологий дистанционное обучение все больше применяется не только для обучения студентов заочной формы обучения, но и при реализации смешанной модели обучения студентов очной формы. Доступность Интернета и персональных компьютеров делает дистанционное обучение доступным для многих групп населения, в том числе и для тех, кто не имеет возможности очно посещать занятия.

Однако дистанционная форма обучения уступает очному, когда речь идет о доверии к результатам обучения и выданным по этим результатам дипломам, свидетельствам и сертификатам, в связи с проблемами верификации студента и распознавания его поведения в ходе тестирований. Эти проблемы призваны решить системы прокторинга.

В классическом представлении прокторинг – это процедура верификации личности испытуемого и дистанционного сопровождения онлайн экзаменов под контролем удаленного наблюдателя, который также именуется проктором, целью которой является повышение уровня доверия к результатам обучения.

Классический прокторинг является синхронным, то есть проктор и испытуемые находятся онлайн в один момент времени и проктор наблюдает за ходом процедуры тестирования. Другой вариант прокторинга является асинхронным. Испытуемый самостоятельно проходит процедуру верификации и тестирования, но все его действия записываются на камеру, после чего проктор в любой момент времени может проверить имеющиеся записи, выявить возможные нарушения и оценить степень доверия к полученным во время тестирования результатам.

Последним словом в сфере дистанционного обучения являются частично или полностью автоматизированные системы прокторинга, в которых функции проктора переключаются на программное обеспечение. Автоматизация таких систем повышает эффективность работы наблюдателя, а в некоторых случаях может полностью заменить его.

Автоматизированный прокторинг – это реализованная программными средствами система контроля за процессом дистанционного тестирования, основными функциями которой является верификация испытуемого и отслеживания нарушений во

время прохождения экзаменов без участия наблюдателя.

Целью данной работы является разработка веб-приложения, функционал которого можно будет использовать для внедрения прокторинга на образовательные платформы.

Основной задачей в данной работе является получение видео с веб-камеры испытуемого и передача видеопотока на сервер для его последующей записи.

Обзор технологий

Для реализации передачи данных была использована технология WebRTC.

WebRTC (Web Real Time Communications) – это бесплатный проект с открытым исходным кодом, который предоставляет веб-браузерам и мобильным приложениям возможности взаимодействия в режиме реального времени (RTC) через простые интерфейсы прикладного программирования (API) [1].

Эта технология позволяет аудио и видеосвязи работать внутри веб-страниц, позволяя устанавливать прямое одноранговое соединение, устраняя необходимость установки плагин или загрузки собственных приложений [2].

Для получения видео и аудио данных с веб-камеры и микрофона испытуемого использовался входящий в состав WebRTC JavaScript API `getUserMedia()`.

Как работает WebRTC:

1. Пользователь открывает страницу, содержащую HTML5 тег `<video>`.
2. Браузер запрашивает доступ к веб-камере и микрофону пользователя.
3. JavaScript код на странице пользователя контролирует параметры соединения для обхода NAT и Firewall.
4. При получении информации о собеседнике браузер начинает согласование используемых аудио и видео кодеков.
5. Начинается процесс кодирования и передача потоковых данных между WebRTC клиентами (в нашем случае, между браузером и сервером).

Преимущества стандарта:

1. Не требуется установка ПО.
2. Очень высокое качество связи, благодаря: использованию современных видео (VP8, H.264) и аудиокодеков, автоматическому определению качества потока под условия соединения, встроенной системе эхо- и шумоподавления, а также автоматической регулировке уровня чувствительности микрофонов участников.

3. Высокий уровень безопасности: все соединения защищены и зашифрованы.
4. Возможность использования встроенного механизма захвата контента, например, рабочего стола.
5. Возможность реализации любого интерфейса управления на основе HTML5 и JavaScript.
6. Проект с открытым исходным кодом – можно внедрить в свой продукт или сервис.
7. Настоящая кросс-платформенность: одно и то же WebRTC приложение будет одинаково хорошо работать на любой операционной системе, при условии, что браузер поддерживает WebRTC. Это значительно экономит ресурсы на разработку ПО [3].

Реализация веб-приложения

В ходе выполнения работы было реализовано веб-приложение, демонстрирующее возможности записи видео с веб-камеры и загрузки файлов на сервер, интерфейс которого представлен на рисунке 1.

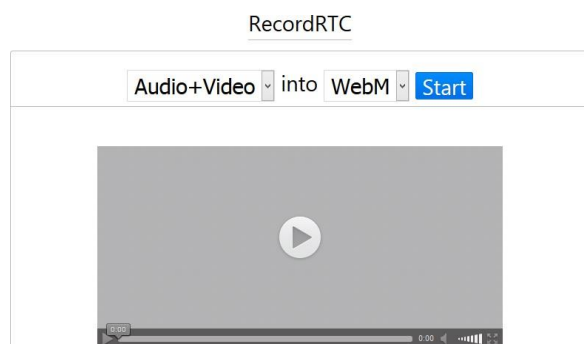


Рис. 1. Интерфейс веб-приложения

В реализованном веб-приложении пользователь может выбрать тип записи аудио и видео, а также только видео. Такой выбор доступен в связи с тем, что браузер Google Chrome не может передавать аудио и видео в одном потоке и, следовательно, их необходимо будет совмещать после записи.

После нажатия на кнопку «Start» браузер запрашивает разрешение на доступ к веб-камере и микрофону. Это реализовано с помощью метода `getUserMedia()`.

Метод `getUserMedia()` запрашивает у пользователя разрешение использовать до одного устройства ввода видео (например, камеру или общий экран) и до одного устройства ввода звука в качестве источника для `MediaStream`.

Если разрешение предоставлено, видео и звуковые дорожки, поступающие с этих устройств, доставляются на указанный обратный вызов. Если запрос на использование устройств отклонен, не существует совместимых устройств ввода или есть какого-либо другое условие вызывающее ошибку, обратный функции выполняется с `MediaStreamError` объектом, описывающим, что пошло не так [4].

После получения разрешения на использование устройств начинается запись.

Видео отображается в окне плеера. Кнопка «Start» заменяется на кнопку «Stop».

После нажатия на кнопку «Stop» происходит завершение записи видео с устройства, а также появляются кнопки «Сохранить» и «Загрузить на сервер». При нажатии на кнопку «Сохранить» файл из тега `<video>` скачивается средствами браузера в папку «Загрузки». При нажатии кнопки «Загрузить на сервер» файл сохраняется на сервере в папке «Uploads», и кнопка «Загрузить на сервер» заменяется на кнопку «Скачать с сервера», при нажатии на которую в браузере открывается новое окно с видео файлом, в адресной строке которой указан путь к файлу на сервере.

Заключение

В результате выполнения работы можно сделать вывод о том, что технология WebRTC пригодна для реализации асинхронного прокторинга, который впоследствии может быть внедрен на образовательные платформы, несмотря на то что технология создана для однорангового взаимодействия.

Реализованное веб-приложение, демонстрирующее лишь малые возможности данного стандарта, является началом большого проекта по внедрению прокторинга на сайт для дистанционного обучения Национального томского политехнического университета [5].

Функционал ляжет в основу сначала асинхронного, а потом уже и автоматизированного прокторинга, задачей которого будет не только запись студентов во время выполнения тестирований и загрузке заданий, но и автоматическое отслеживание нарушений поведения во время тестирования.

Список использованных источников

1. WebRTC. [Электронный ресурс] / WebRTC. – URL: <https://webrtc.org> (дата обращения 19.11.2018).
2. How WebRTC Is Revolutionizing Telephony [Электронный ресурс] / Trilogy-LTE. – URL: <http://blogs.trilogy-lte.com/post/77427158750/how-webrtc-is-revolutionizing-telephony> (дата обращения 19.11.2018).
3. WebRTC. Видеоконференции в браузере [Электронный ресурс] / TrueConf. – URL: <https://trueconf.ru/webrtc.html> (дата обращения 19.11.2018).
4. `Navigator.getUserMedia()` [Электронный ресурс]/MDN Web Docs. – URL: <https://developer.mozilla.org/en-US/docs/Web/API/Navigator/getUserMedia> (дата обращения 19.11.2018).
5. Платформа онлайн-обучения Томского политехнического университета. [Электронный ресурс] / ЭИОС ТПУ. – URL: <http://stud.lms.tpu.ru> (дата обращения 19.11.2018).