

[Слайд 1 – Титульный слайд]

Добрый день!

Меня зовут Богомоллов Юрий, это мой коллега Бодрухин Александр. Мы представляем проект "Облачная информационная система обучения студентов".

[Слайд 2 - Существующие проблемы]

На сегодняшний день в мире существует множество систем управления обучением (далее я буду называть их LMS - от Learning Management System). Эти LMS обычно создаются в рамках одной организации или вуза, и позволяют собственным студентам и работникам обучаться и преподавать. Но далеко не все учреждения практикуют подобное, особенно редко это встречается в школах.

Иногда такие организации как CISCO, Huawei и прочие позволяют использовать свои курсы другим организациям и вузам. Тем не менее, если необходимо использовать курсы нескольких различных LMS, это может вызвать неудобства. Например, приходится посещать множество различных ресурсов, на каждом из которых необходимо регистрироваться и вводить личные данные. Кроме того, достижения студентов на этих ресурсах остаются внутри этих ресурсов, что затрудняет составление портфолио.

Кроме того, если организация решит поделиться своими учебными материалами с другими организациями, придётся решать проблему размещения контента и предоставления к нему доступа - например, придётся разрабатывать для этого сайт.

[Слайд 3 - Функции системы - организация]

Таким образом мы пришли к идее, что необходима централизованная система предоставления электронных курсов. Функции такой системы можно рассмотреть с нескольких точек зрения.

С точки зрения организации:

- 1) позволять размещать и контролировать собственные курсы;
- 2) позволять предоставлять доступ к курсам, в том числе на платной основе;
- 3) предоставлять серверные мощности для требующих того курсов;

[Слайд 4 – Функции системы - студент и преподаватель]

С точки зрения студента:

- 1) предоставлять доступ к электронным курсам, на которые зарегистрирован студент;
- 2) хранить информацию о достижениях;

С точки зрения преподавателя:

- 1) позволять выдавать студентам индивидуальные и групповые задания;

[Слайд 5 - Структура системы]

Общий вид такой системы можно отобразить с помощью следующей иллюстрации. Как можно заметить, здесь имеется несколько важных частей системы.

Веб-сайт. Он обеспечивает взаимодействие пользователей и системы.

Учебные материалы. Они будут составляться преподавателями, изучаться и решаться студентами. И разрабатывать, и решать их можно будет с помощью приложений.

Приложения. Во-первых, они могут объединяться в целые системы. Во-вторых, они могут иметь серверную часть и клиентскую. Клиентская часть – просто веб-странички, которые могут использоваться для отображения материала, отправки студентом решений и т.д. Серверная же часть – какие-то программы, исполняющие какую-либо обработку данных.

Серверное программное обеспечение. Является необходимым для функционирования системы. Например, оно может обеспечивать запуск и взаимодействие различных приложений.

Кроме того, такая структура способна обеспечить поддержку SCORM, xAPI, CMI-5 и любых других стандартов реализации электронных курсов.

У вас наверняка возник вопрос: "А какие приложения и материалы могут разрабатывать организации?" Приведём следующие примеры.

[Слайд 6 - Лекционное приложение]

1) Лекционное приложение. Имеет клиентскую и серверную части. Клиентская часть отображает в браузере студента материалы, а также регистрирует пользовательскую активность и отправляет её на сервер. Серверная же часть собирает таким образом статистические данные, которые могут пригодиться преподавателю для улучшения материала.

[Слайд 7 - Задача вида "Дано/Решение"]

2) Задачи вида "Дано/Решение": по химии, физике, математике. На сервере по шаблону генерируется условие задачи и отправляется в браузер пользователя. Тот решает задачу с помощью редактора формул или с помощью тетради и ручки, затем нажимает кнопку "Отправить". Если был отправлен скан или фотография, то на сервере запускается приложение для преобразования рукописного текста в электронный. Затем решение проходит проверку с помощью чекера на сервере.

[Слайд 8 - Лабораторная работа по программированию]

3) Лабораторная работа по программированию. Клиентская часть отображает студенту условие задачи, а студент в свою очередь загружает исходный код программы. Решение компилируется и запускается на тестовых примерах.

Как видите, приложения могут иметь входные и выходные данные любого типа. Таким образом достигается гибкость системы.

[Слайд 9 - Текущая разработка]

В данный момент система находится в активной разработке: пишется сайт, детально прорабатывается концепция серверной части системы. Разработка серверной части ведётся на языке Python 3. Интерфейс веб-сайта, естественно, на HTML + CSS + JavaScript. Используются фреймворк Django, библиотека Bootstrap, СУБД PostgreSQL и ПО для "контейнеризации" приложений Docker.

[Слайд 10 - Финальный слайд]

На этом всё, спасибо за внимание. Ожидаем ваши вопросы.

[Подсказка: другие примеры приложений]

- Лекционное приложение: терминологический модуль
- Тренажёры разборов в лингвистике: морфологический разбор слова, синтаксический разбор предложения и прочие
- Балансировка химических уравнений
- Задачи вида "Дано/Решение": по биологии (генетические), по физике, по химии, по математике
- Тесты по географии (с использованием карты)
- Тренажёры в информатике и программировании: по системам счисления, по обратной польской нотации, по задачам динамического программирования, по созданию квайнов