

[Слайд 1 – Титульный слайд]

Добрый день!

Меня зовут Богомоллов Юрий, и я представляю проект «Облачная информационная система обучения студентов».

Информационные технологии окружают нас в повседневной жизни: можно оплачивать покупки телефоном; заказывать еду, не выходя из дома; заводить знакомства через Интернет. Но в учебный процесс внедрение информационных технологий почему-то происходит гораздо медленнее. В лучшем случае используются технологии, которым полтора десятка лет, в худшем – обычные бумага и ручка. Нет, я против этих вещей ничего не имею, порой без них не обойтись. Но ведь есть возможность внедрять новые технологии в учебный процесс.

[Слайд 2 – Цели системы]

Поэтому возникла идея разработки системы, которая позволила бы модернизировать процесс обучения. Целей у такой системы несколько. Во-первых, способствовать обучению студентов с помощью интересного и интерактивного учебного материала. Во-вторых, позволять преподавателю разрабатывать подобный материал. И в-третьих – снизить нагрузку на преподавателей за счёт автоматизации процесса проверки решений студентов.

[Слайд 3 – Концепция: структура системы]

Разрабатываемая система, будет состоять из следующих частей [указываю рукой на слайд]. Далее я чуть более подробно опишу каждую из этих частей.

[Слайд 4 – Концепция: веб-сайт, базы данных и серверное ПО]

Первое – это веб-сайт. Он обеспечивает взаимодействие пользователей и системы.

Второе – базы данных. Мы планируем два их типа в системе: одну глобальную БД и множество локальных. Глобальная используется самой системой для её корректного функционирования. Локальные же могут использоваться приложениями в своих целях.

Третье – различное серверное программное обеспечение, необходимое для функционирования системы. Например, оно может обеспечивать взаимодействие различных приложений или предоставлять пользователям доступ к системам контроля версий.

[Слайд 5 – Концепция: учебные материалы]

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

[Слайд 6 – Концепция: приложения]

Пятое – приложения. Во-первых, они могут объединяться в целые системы. Во-вторых, они могут быть серверными и клиентскими. Клиентские – просто веб-странички, которые могут использоваться для отображения материала, отправки студентом решений и т.д. Серверные же приложения – какие-то программы, исполняющие какую-либо обработку данных.

[Слайд 7 – Приложения: примеры]

Давайте чуть подробнее остановимся на приложениях. Их примеры вы сейчас можете видеть на слайде [указываю рукой на слайд]. Компиляторы, интерпретаторы, какие-либо скрипты и даже нейросети. Давайте рассмотрим, как приложения можно связать в системы.

[Слайд 8 – Приложения: лекционный материал]

Первая такая система – для разработки и изучения лекционного материала. Пусть преподаватель разрабатывает какие-то лекционные материалы. Они попадают в банк лекций. Затем студенты после пары, например, изучают эти материалы. Там могут быть какие-то контрольные вопросы, интерактивные изображения, формы с пошаговой интерпретацией исходного кода, представляющего алгоритм. И какие-нибудь такие статистические данные вроде времени, проведённого студентом в разных частях лекции, или взаимодействие студента с формами/картинками, попадают к преподавателю, чтобы он мог анализировать активность учащихся и эффективность тех или иных частей материала. Кроме того, студенты могут указать преподавателю на какие-то непонятные моменты с помощью формы обратной связи.

[Слайд 9 – Приложения: тест]

Второй вариант системы приложений – тестирование студентов. Работа скучная и механическая, легко поддаётся автоматизации. Например, таким образом. Преподаватель создаёт вопросы по теме, формирует тест из них и выдаёт студенту. Тот прорешивает его, а система затем проверяет правильность ответов.

[Слайд 10 – Приложения: задача спортивного программирования]

Ещё одна система приложений – задачи спортивного программирования, т.е. олимпиады. Преподаватель составляет задачи, ограничения, тесты к ним – и выдаёт студентам. Те пишут код и отправляют его на сервер. Там, конечно же, происходит проверка на наличие небезопасных системных вызовов, а затем компиляция или интерпретация программы с последующим прогоном по тестам. По такому принципу работают многие площадки по спортивному программированию. Например, популярный в СНГ сайт [Codeforces.com](https://codeforces.com), сайт Новосибирского Государственного Университета [Olympic.nsu.ru](https://olympic.nsu.ru) и наш Томский сайт [Lerna.pro](https://lerna.pro), созданный на основе аналогичного сайта ТУСУРа.

[Слайд 11 – Приложения: задачи по математике]

Теперь представьте работу преподавателя по математике. А если конкретнее, то ту её часть, где нужно проверить контрольные работы сотни студентов по интегралам. И в каждой работе с десятком интегралов. У нашего преподавателя, например, уходила неделя или даже две на это. А всё потому, что он проверял каждый интеграл вручную. Но процесс-то механический. Его можно автоматизировать с помощью следующей системы приложений.

Есть генератор задач, в котором задаются параметры генерации. Сгенерированные задачи попадают к студенту. Тот решает всё в тетрадке или в редакторе формул. В первом случае появляется дополнительный шаг с распознаванием текста, но ведь это не невозможно. Затем решение проходит проверку чекером. И преподавателю остаётся только проанализировать вердикты и какие-то сомнительные решения.

[Слайд 12 – Приложения: эссе по литературе]

Ну и последний пример. Школа. Кабинет русского языка и литературы. Учитель даёт детям задание дома написать сочинение. И какова же будет вероятность того, что школьники не захотят схитрить и просто списать сочинение с сайта с готовыми домашними заданиями?

Схема здесь очень похожа на предыдущую. В основном из-за того, что здесь так же есть распознавание рукописного текста. Но здесь также добавляется архив уже сданных сочинений, или сочинений с сайтов с готовыми домашними заданиями. Такая система позволит учителям узнать, насколько добросовестные ученики у них.

[Слайд 13 – Текущая разработка]

В заключении хотелось бы сказать, что идея постоянно эволюционирует в сторону универсальности и большей функциональности.

Кроме того, система уже находится в активной разработке. На данный момент мы с коллегой разрабатываем веб-сайт и базу данных. Веб-сайт разрабатывается с использованием фреймворка Django для языка программирования Python 3, а также инструмента для разработки дизайна Bootstrap 4.

База данных на данный момент насчитывает восемнадцать сущностей, не считая нераскрытые связи многие-ко-многим. В качестве СУБД при этом используется PostgreSQL.

[Слайд 14 – Финальный слайд]

На этом всё. Спасибо за внимание, ожидаю ваши вопросы.