

ПРОЕКТИРОВАНИЕ ОНЛАЙН-ГЕНЕРАТОРА ЗАДАНИЙ ПО МАТЕМАТИКЕ ДЛЯ УЧАЩИХСЯ СРЕДНИХ, СРЕДНЕ-ПРОФЕССИОНАЛЬНЫХ И ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ



Глазков Никита Олегович

студент по специальности "Информатика и вычислительная техника. Веб-технологии", кафедра Инфокогнитивные технологии, Факультет информационных технологий, Московский политехнический университет



Беляева Дарья Владиславовна

студентка по специальности "Информатика и вычислительная техника. Веб-технологии", кафедра Инфокогнитивные технологии, Факультет информационных технологий, Московский политехнический университет

Аннотация: Составление вручную заданий для вариантов расчётно-графических работ по высшей математике имеет ряд заметных проблем, с которыми сталкиваются преподаватели средних, средне-профессиональных и высших учебных заведений. Наш онлайн-генератор предоставляет возможность автоматизации данного процесса, используя современные методики разработки архитектуры веб-приложений, принципы юзабилити и проектирования пользовательского опыта. Рассмотрены основные этапы разработки и масштабирования, а также особенности генерации заданий с использованием системы компьютерной вёрстки TeX и набора макрорасширения LaTeX.

Ключевые слова: генератор заданий, образовательные технологии, веб-приложение, JavaScript, Vue.js, Node.js

Abstract: Manually compiling assignments for options for calculating and graphic work in higher mathematics has a number of notable problems that teachers of secondary, secondary vocational and higher educational institutions face. Our online generator provides the ability to automate this process using modern techniques for developing web application architecture, the principles of usability and user experience design. The main stages of development and scaling, as well as the features of task generation using the TeX computer layout system and the LaTeX micro extension set, are considered.

Keywords: exercise generator, educational technology, web-application, JavaScript, Vue.js, Node.js

Введение

Всё больше и больше технических процессов внедряются в повседневную практику преподавания. Автоматизация самых различных процессов позволяет педагогическому составу меньше затрачивать сил на формализм, уделяя больше внимания обучению и улучшению учебных программ.

Одной из особенностей обучения студентов технических направлений является самостоятельное прорешивание различных вариантов заданий при рассмотрении большинства тем высшей математики. Для этого преподаватель составляет расчётно-графические работы, разбитые по определённым разделам учебной программы. В таком случае студент может обнаружить какие-либо особенности решения, которые не видны на рядовых примерах и могли быть упущены из виду педагогом на лекциях. К тому же, повторение при обучении и развитии необходимых навыков, согласно Кривой забывания

Эббингауза, позволяет качественно и уверенно достичь необходимого результата на практике [1]. Однако, даже у проверенных способов есть свои недостатки, из-за чего преподавателям необходимо идти в ногу со временем и внедрять новые методики.

С популяризацией портативных мультимедийных устройств резко повысился шанс утечки вариантов работ контрольных точек. Для повышения качества образования необходимо иметь несколько вариантов, как минимум – по одному на каждого ученика, с учётом того, что потоки студентов могут достигать нескольких сотен на факультете. Почти каждый студент имеет смартфон, способный в считанные секунды сделать фотографию заданий и переслать любому другому обучающемуся, за чем преподаватель может просто не успеть уследить.

Многие упускают из виду работу сверх нормы и её последствий для преподавателя. Перерабатыванию способствует составление

расчётно-графических работ, которое вручную может занять несколько недель. Профессиональное выгорание педагога и его составляющие (эмоциональное истощение, деформация отношений, сокращённая профессиональная реализация и прочие) может оказать пагубное воздействие на качество образования. В таком случае ценность и эффективность труда педагога резко снижается [2]. Из-за большого количества ручной работы, повышается шанс ошибки или опечатки в ходе составления упражнений.

Мы предлагаем качественно новое решение, позволяющее решить вышеперечисленные проблемы – онлайн-генератор упражнений, использующий современные стандарты разработки веб-приложений.

Опрос целевой аудитории

Полагаясь на принципы дизайн-мышления и интернет-маркетинга, мы создавали человекоориентированный информационный продукт [3], и потому начали с опроса целевой аудитории – студентов и преподавателей. Участие в анкетировании приняли студенты от 18 до 24 лет бакалавриата и магистратуры, а также преподаватели технических кафедр. Это позволило учесть некоторые особенности при проектировании приложения. Например, большинство (72,7%) сочло электронный вариант расчётно-графических работ наиболее удобным, однако весомый процент аудитории (27,3%) склонны использовать бумажные варианты. К тому же, большая часть студентов (81,8%) осознаёт необходимость самостоятельной подготовки к контрольным точкам учебной программы, но не имеет должной мотивации. Основными претензиями к современным расчётным работам являются опечатки, различие сложности в вариантах и неудобный формат.

Преподаватели выразили желание иметь возможность работать с методическими материалами в отрыве от онлайн-платформы и интернет-соединения, так как многие пространства аудиторий в вузах не оснащены персональными компьютерами, а студенты могут иметь трудности с доступом к электронным ресурсам. Таким образом, мы выявили основные акценты разработки будущего продукта.

Этапы проектирования и алгоритм генерации

Самый важных и ответственный этап разработки – проектирование будущего веб-приложения. На данном этапе формируется интерфейс, будущий пользовательский опыт, исследуются основные этапы бизнес-процессов приложения.

Наш информационный продукт делает упор на тренировке за счёт решения большого количества упражнений, а потому главная задача – это именно генерация. Огромное количество решений в Интернете, такие как Wolfram Alpha, позволяют решать сложные задачи точно и имеют гибкий API, но не имеет возможности генерировать задания или данная возможность выражена очень слабо [4]. Потому наш онлайн-генератор будет создавать набор заданий на выбранную тему, а ответы на

сгенерированные задания будут вычисляться либо с использованием стороннего API, либо библиотек. Сконцентрируемся на алгоритме генерации.

Мы будем использовать частично-шаблонную генерацию. По своей сути, каждое математическое выражение – это набор составных частей, в которых изменяются числовые коэффициенты и конфигурация частей (“шаблон”) относительно непрямых вычисляемых коэффициентов (“мета-коэффициентов”).

Исходя из выбранной темы, мы будем добавлять и формировать такие шаблоны на основе случайного выбора из списка. Однако сложности начинаются на этапе составления данного списка. Так, для отработки темы “Формула для приближённого вычисления”, как шаблон для генерации может использоваться выражение $f(A)$, где $f(x)$ – может выбираться из списка $\cos(x) / \sin(x) / \operatorname{tg}(x) / \operatorname{ctg}(x) / \operatorname{arctg}(x) / \operatorname{arctg}(x)$ при условии $A \in [1, 29] \cup [31, 44] \cup [46, 59] \cup [61, 89] \cup [91, 119] \cup [121, 134] \cup [136, 149] \cup [151, 179] \cup [181, 209] \cup [211, 224] \cup [226, 239] \cup [241, 269] \cup [271, 299] \cup [301, 314] \cup [316, 329] \cup [331, 359]$ – в данном диапазоне исключаются табличные значения, которые бессмысленны при тренировке. Однако для темы “Неопределённый интеграл” необходимо учитывать типы неопределённых интегралов и создавать для каждого шаблон. Например, если берётся функция $f(x)$, для реализации метода замены переменной, обязательно должна быть включена в упражнение и функция $f'(x)$ [5].

Дальше выбранные шаблоны в связи со случайными числовыми коэффициентами должны преобразовываться в набор TeX-команд, который будет в дальнейшем преобразовываться в HTML, SVG или MathML [6] для отображения в любом современном браузере.

Общие особенности проектирования приложения PDF-документы и печать

В связи с тем, что в приложении необходима возможность вывода сгенерированных упражнений на печать, мы включили возможность сохранения файла формата pdf с заданиями и его дальнейшей печати. В таком случае нужно учитывать особенность генерации pdf-файлов: различные способы могут создавать потери в качестве. Используемая нами библиотека jsPDF с подключёнными плагинами позволяет избежать этого. Создание документа происходит на основе сгенерированного DOM-дерева, а потому пользователь получает файл с упражнениями, которые выглядят так же, как и в отображении веб-приложения.

Адаптивность, кроссбраузерность, доступность

То, на что необходимо обратить особое внимание – это доступность и адаптивность. Пользователи должны иметь возможность пользоваться приложением вне зависимости от устройства или от функциональных возможностей человека, согласно общепринятым правилам a11y, которые подробно описаны в Web Content Accessibility Guidelines версии 2.1 (сокращённо WCAG) [7].

Адаптивность под самые различные размеры

экранов организуется встроенными инструментами медиазапросов HTML5 и CSS3, а также особенно-сти кроссбраузерной верстки. Это обычная задача для рынка веб-разработки, а потому здесь заострять внимание мы не будем [8].

Обоснование технологического стека

Клиентская часть

Для корректной работы и функционирования клиентской части необходимо соблюдать несколько простых правил. Веб-сайт должен занимать как можно меньший размер в финальной сборке проекта, но в то же время иметь возможность для масштабирования и быстрой доработки. Для уменьшения количества запросов, а значит - нагрузки на сервер, последние несколько лет используется подход SPA (single page application) - одностраничного приложения. Современные JavaScript-библиотеки и фронтенд-фреймворки, как Vue, Angular или React, предоставляют широкий спектр для разработки и поддержки такого типа веб-приложений. Использование проверенных решений и структур проекта позволяет команде любого размера оперативно включиться в процесс модернизирования приложения на любом этапе. Для разработки онлайн-генератора был использован JavaScript-фреймворк Vue.js для разработки одностраничных приложений в реактивном стиле, но нами мог быть использован любой другой - финальный продукт не изменился бы значительно.

Использование фреймворков позволяет оптимизировать время разработки и масштабирования. Для удобства и скорости разработчиками Vue.js была введена система Vue CLI. Она позволяет интерактивно создавать проект, быстро прототипировать без конфигурации, обновлять, расширять и интегрировать официальные плагины в проект через командную строку, а также предоставляет полноценный графический пользовательский интерфейс для управления проектами Vue.js. Данный инструмент обеспечивает бесперебойную работу различных инструментов сборки [9].

Поскольку Vue.js использует разбиение на компоненты, при построении архитектуры относительно крупного приложения появляется проблема отсутствия стандартизации и большого количества разрозненных хранилищ. Для решения этих особенностей, разработчиками фреймворка был создан паттерн управления состоянием и библиотеку для приложений Vuex [10]. Он организует объединённое хранилище для всех компонентов с правилами, которые обеспечивают изменение состояния только предсказуемым образом. К тому же, Vuex интегрируется с официальным расширением vue-devtools [11] для удобной отладки.

Для маршрутизации SPA (single page application), в приложения Vue.js используется официальная библиотека Vue Router. Она позволяет создавать настраиваемое поведение прокрутки страницы, автоматически добавлять активный CSS класс для ссылок, удобно контролировать навигации и анимировать

переходы [12].

Серверная часть

В случае выбора инструмента для серверной разработки, выбор не так очевиден. Наш взгляд пал на Node.js - программную платформу, основанную на движке JavaScript Chrome V8, который позволяет транслировать вызовы на языке JavaScript в машинный код [13]. С помощью Node.js можно решать проблемы различного уровня, от статической маршрутизации до построения сложного API или сложной системы асинхронных серверных вычислений.

Серверная часть зависит от механизма генерации упражнений - на сервере с дальнейшей отправкой на клиент коэффициентов или же с генерацией и распознаванием исключительно на клиенте, с отправкой коэффициентов на сервер в случае необходимости. Первый вариант делает упор на генерации и сохранении вариантов для пересылки или печати в любой момент из личного кабинета - это позволяет уменьшить нагрузку на ресурсы клиентского устройства. Второй же вариант позволяет сиюминутно генерировать варианты и уменьшить пересылаемый трафик, но он усложняет хранение, замедляет процесс записи сгенерированных коэффициентов.

Почему TeX и LaTeX?

В наши дни система компьютерной верстки TeX - золотой стандарт для академического сообщества, позволяющий создавать высококачественные документы. К тому же, с использованием библиотеки микрорасширений LaTeX, можно автоматизировать многие задачи набора текста, в том числе - форматирование математических и физических формул.

Так, например, формула:

$$f(x, y, \alpha, \beta) = \frac{\sum_{n=1}^{\infty} A_n \cos\left(\frac{2n\pi x}{\nu}\right)}{\prod \mathcal{F}g(x, y)}$$

в LaTeX будет выглядеть как:

```
\begin{equation}
f(x, y, \alpha, \beta) = \frac{\sum \limits_{n=1}^{\infty}
A_n \cos \left( \frac{2 n \pi x}{\nu} \right)}{\prod \mathcal{F} \{g(x,y)\}}
\end{equation}
```

KaTeX

До недавнего времени для преобразования математических формул на языке MathML или LaTeX в общепринятое понятное отображение использовалась библиотека MathJax.js [14], не требующая настройки и поддерживающая большинство современных браузеров и операционных систем. Однако, она требует значительное время для рендеринга.

Эту проблему решает актуальная библиотека KaTeX.js [15]. Она действует намного быстрее, так как отображает формулы синхронно и не нуждается в переформировании страницы; поддерживает высокое качество печати, рендеринг на стороне

сервера с помощью Node.js и отправки данных в виде простого HTML. К тому же библиотека достаточно автономна: она не имеет никаких зависимостей и может быть легко связана с ресурсами нашего веб-приложения.

Масштабирование и перспективы

Дополнительный функционал

Данный генератор упражнений подразумевает много возможностей для масштабирования. Например, вышеупомянутый опрос выявил, что пользователи хотели бы видеть возможность для тестирования, причём как самостоятельного, так и для проведения контрольных работ по пройденным темам с дальнейшим созданием таблицы результатов.

Помимо этого, в приложение может быть внедрена возможность выбора сложности генерируемого набора упражнений. Например, при генерации заданий на обыкновенный арифметический счёт, можно установить выбор параметров: сколько должно быть операций в выражении, какая разрядность чисел, использовать ли скобки. Аналогично для каждой темы.

Выводы

В результате мы получили готовое приложение, которое уже сейчас помогает в учебном процессе, используя современные технологии построения архитектуры веб-приложений и цифровизации образования. Оно обладает следующим функционалом:

определение тематики заданий

- выбор количества генерируемых заданий
- вывод сгенерированных заданий на печать или в pdf-файл
- создание определённых шаблонов для упражнений
- вычисление ответа на сгенерированное задание,
- а также возможности для масштабирования:
- тестирование по сгенерированному материалу
- выбор сложности сгенерированных заданий

Список литературы

1. Буймов Аркадий Георгиевич, Буймов Борис Аркадьевич "Вероятностная модель эффекта повторений в обучении" // Доклады ТУСУР. 2010. №1-2 (21).
2. Бутова Людмила Александровна, Бычкова Елена Семеновна, Лыткина Аксана Виликовна "Исследование психоэмоциональных и поведенческих компонентов эмоционального выгорания у педагогов с различным стажем работы" // Известия ВГПУ. 2019.

№3 (136).

3. Vasilieva Elena V. Design-thinking: practice of customer experience research // Современные информационные технологии и ИТ-образование. 2018. №2.
4. Клоков Александр Сергеевич, Сорокин Анатолий Никифорович "Wolfram Alpha как рабочая среда для студентов, изучающих курс теоретической механики" // Электронный научно-методический журнал Омского ГАУ. 2016. №4 (7).
5. Коновалов Я.Ю., Соболев С.К. Методические аспекты компьютерного генерирования заданий по математике // Машиностроение и компьютерные технологии. 2016
6. Справочные данные по MathML [Электронный ресурс]: MDN web docs. URL: <https://developer.mozilla.org/ru/docs/Web/MathML>
7. Web Content Accessibility Guidelines (WCAG) Overview [Электронный ресурс]: W3C. URL: <https://www.w3.org/WAI/standards-guidelines/wcag/>
8. Станкевич Александр Матвеевич, Семёнов Геннадий Николаевич Современные подходы к разработке веб-приложений на примере сайта сбалансированного питания // Успехи в химии и химической технологии. 2015. №4 (163).
9. Vue CLI overview [Электронный ресурс]: VueJS. URL: <https://cli.vuejs.org/guide/> (дата обращения: 05.02.2020.)
10. Vuex Getting Started [Электронный ресурс]: VueJS. URL: <https://vuex.vuejs.org/guide/> (дата обращения: 05.02.2020.)
11. Vue-devtools [Электронный ресурс]: GitHub. <https://github.com/vuejs/vue-devtools>
12. Vue router. Introduction [Электронный ресурс]: VueJS. URL: <https://router.vuejs.org/>
13. Долгов Антон Николаевич, Нуруллин Роман Юрьевич "Программная платформа Node. JS" // До-стижения науки и образования. 2016. №12 (13).
14. MathJax. Beautiful math in all browsers [Электронный ресурс]: MathJax. URL: <https://www.mathjax.org/>
15. KaTeX - The fastest math typesetting library for the web [Электронный ресурс]: KaTeX. URL: <https://katex.org/>

Выражаем благодарность научному руководителю С.А. Муханову за помощь в подготовке статьи.