**Пояснительная записка**

Для подготовки к ЕГЭ по информатике необходимо решать множество задач, для отработки тех или иных навыков. Но не все сборники задач имеют необходимое количество задач, либо не имеют ответы к имеющимся задачам. Поэтому создание генератора, который сможет создавать большое количество задач, способно решить эти проблемы.

Генератор должен позволять пользователю выбрать нужные типы и необходимое количество этих задач. После чего он должен вывести на экран пользователя сгенерированные условия и ответы к ним (опционально).

В качестве заданий, которые будет создавать генератор, были выбраны комбинаторные задачи. *Почему*

Процесс создания генератора потребовал решения множество задач. Первая из задач - выбор платформы на которой будет реализована задумка. Для этого были выбраны веб-средства, поскольку благодаря веб-средствам генератором смогут воспользоваться все пользователи, которые имеют доступ к интернету.

Следующая задача, заключалась в том, что нужно было продумать, как будет выглядеть веб-приложение, как оно будет реализовано. *Что было сделано*

Далее нужно было найти и проанализировать информацию о комбинаторных задачах: поиск и отбор задач, которые будут в последующем будут генерироваться, выявление алгоритмов решения этих задач, структурировать полученную информацию. *Все это было выполнено благодаря материалам с сайта Константина Полякова.*

После этого нужно было выбрать и изучить средства, позволяющие создать веб-приложение. Для этих целей были выбраны такие средства: язык разметки HTML, каскадная таблица стилей CSS, язык программирования JS.

HTML – язык разметки, который позволяет понимать браузеру, как правильно отображать блоки на веб-странице. При загрузке веб-страницы браузер интерпретирует код HTML и выводит на экран пользователя контент. HTML имеет блочную структуру. С помощью HTML на веб-сайте получилось реализовать подобную структуру: шапка веб-страницы, блок главного экрана, блок выбора типов задач, блок сгенерированных задач. *пример сайта без цсс*

Но только с помощью HTML сложно создать полноценную веб-страницу, так как у нее будет отсутствовать стилизация. Поэтому на веб-странице используется каскадная таблица стилей CSS. В современной веб-разработке почти каждый сайт использует связку HTML и CSS. CSS позволяет указать браузеру какой элемент нужно стилизовать. Для создания генератора задач потребовалось задать стили каждому блоку из HTML разметки.

Также веб-сайт должен быть адаптивным, то есть нормально функционировать на устройствах с разными размерами экранов. Для упрощения адаптации веб-сайта был использован фреймворк (средство, которое объединяет в себе различные технологии, позволяющее упростить процесс разработки продукта) Materialize, основанный на популярном направлении, которое было разработано Google, - material design. Фреймворк предоставляет 12 колоночную разметку, которая позволяет удобно располагать элементы веб-страницы относительно друг друга, в зависимости от ширины экрана на котором просматривается содержимое страницы.

Чтобы добавить обратную реакцию на действия пользователя и логику потребовалось использовать язык программирования JavaScript. Это интерпритируемый язык высокого уровня. Код интерпритируемого языка выполняется построчно интерпретатором. Часто JavaScript применяют для создание сценариев и скриптов веб-страницы. Скрипты представляют из себя набор инструкций, которые выполняются при наступлении того или иного события. Код JavaScript интерпретируется браузером во время загрузки страницы. *интерпретатор*

На веб-странице JavaScript используется для решения многих задач. Одна из них – это переходы между блоками. Например, при нажатии кнопки «сгенерировать» блок главного экрана сменяется блоком выбора типов задач. Также с помощью JavaSctipt была реализована генерация текстов задач, которые выводятся в блок сгенерированных задач, и ответов к ним. С помощью фреймворка Materialize на веб-страницу удалось добавить вкладки для каждого типа задач. Это позволило структурировать содержимое веб-страницы и сделать более понятный интерфейс. *Подробнее о генерации*

В итоге выполнения всех поставленных подзадач и задач получилось реализовать полностью функционирующий генератор комбинаторных задач, который предоставляет пользователю выбор типов интересующих его заданий и вывод множества вариантов этих заданий с ответами.

Данный генератор имеет востребованность среди учащихся, которые заинтересованы в улучшении навыка решения комбинаторных задач, которые встречаются в ЕГЭ по информатике, выявлении слабых сторон при решении и уменьшении времени, затрачиваемого на решение.

Создание продукта проектной деятельности позволило достичь таких положительных эффектов: возможножность формирования интереса у обучающихся к решению, повышение процента решаемости, получение базовых и углубленных знаний в области решения комбинаторных задач.

В процессе реализации генератора комбинаторных задач возникло множество сложностей. Одной из сложностью было создание веб-сайта, который можно будет запустить на любом типе устройства, то есть добавление адаптивности на веб-страницу. *Добавить решение этой проблемы*

Также возникла такая сложность: нужно было проанализировать множество комбинаторных задач для выявления различных типов и последующего выявления алгоритмов решения этих задач.

Другая сложность заключалась в том, что полученные алгоритмы решения задач нужно было реализовать с помощью JavaScript. Основная трудность при реализации алгоритмов была в написании функций, которые генерировали ответы и выводили их в конце каждого блока с уже сгенерированными задачами. Для правильного функционирования этих функций нужно генерировать последовательности слов, состоящих из случайно выбранных букв, что также вызвало трудности в реализации. *Переписать короче*

Помимо этого другая сложность была в том, чтобы разделить код JavaScript по разным файлам для более легко понимания кода, а также упрощения нахождения ошибок в коде и исправлений их. *Переписать короче*

Проект