**Слайд 2. Актуальность**

В современном образовании учебные мероприятия проводятся не так эффективно, как могли бы. Создателям образовательных мероприятий приходится тратить много времени на проверку ответов обучающихся, составление рейтинга, анализ результатов. При наличии некоторого инструмента, который решает проблемы, перечисленные выше, организаторы могут значительно сэкономить время на работу, которую можно автоматизировать, и сосредоточится на чем-то более важном (например, на содержании мероприятия или работы, проверки наличия ошибок в работе).

**Слайд 3. Новизна**

Существуют подобные на мой проект образовательные инструменты. Самые известные из них это СДАМ ГИА — образовательный портал для подготовки к экзаменам, Kahoot — это игровая обучающая платформа [16], а также Stepik — многофункциональная и гибкая платформа для создания образовательных материалов. В каждом из этих трёх инструментов есть как плюсы, так и минусы. Например, на сайте СДАМ ГИА отсутствует возможность создавать свои уникальные задания, вместо этого организаторы могут брать задания из огромного банка заданий, а также нельзя отслеживать рейтинг в реальном времени. На сайтах Stepik, Kahoot нельзя создавать мероприятия, в которых содержатся задания, вид которых отличается от стандартных, а также нет возможности создать уникальные правила мероприятия.

Три самых важных преимущества нашего инструмента, которые отличают его от других похожих проектов – это:

1. Составление рейтинга, согласно правилам выбранного мероприятия

2. Гибкость настройки мероприятия

3. Возможность создавать мероприятия, включающие в себя задания, которые отличаются от классических (например, задания “Своей игры”, математического домино и так далее)

**Слайд 4. Цель и задачи**

Цель проекта: разработать эффективный образовательный инструмент для создания и проведения учебных мероприятий.

Задачи:

1. Составить техническое описание проекта

2. Изучить классификацию сайтов

3. Провести исследование стека технологий для написания бэкенда и фронтенда сайта

4. Реализовать проект в виде сайта

**Слайд 5. Практическое значение проекта**

Проект имеет практическое значение, так как может быть использован на занятиях в школах и других учебных заведениях, а также при проведении внеурочных мероприятий.

**Слайд 6. Функционал инструмента для пользователей**

Инструмент состоит из следующих основных функциональных блоков:

1. Регистрация, аутентификации, авторизация;
2. Функционал для организатора мероприятий;
3. Функционал для участника мероприятий;
4. Функционал для пользователей, которые в данный момент времени не являются ни организаторами, ни участниками.

Идентификация — процедура, в результате выполнения которой для субъекта идентификации выявляется его идентификатор, однозначно определяющий этого субъекта в информационной системе.

Аутентификация — процедура проверки подлинности, например проверка подлинности пользователя путем сравнения введенного им пароля с паролем, сохраненным в базе данных.

Авторизация — предоставление определенному лицу или группе лиц прав на выполнение определенных действий.

**Слайд 7. Виды мероприятий**

Тест

Тест обладает следующей особенностью – к вопросу предложен список ответов, из которых надо выбрать один или несколько верных.

Тест предполагает следующие функционал:

● Количество ответов может быть не более 1000;

● Проверка ответов происходит автоматически.

Контрольная работа

Контрольная работа обладает следующей особенностью – ниже вопроса есть поля для ввода ответа, в который нужно ввести точный ответ или прикрепить файл.

Контрольная работа предполагает следующие функциональные блоки:

● Если вопрос требует ответа, состоящего из слов, то проверка проходит автоматически.

● Если вопрос требует развернутого ответа, то нужно прикрепить файл. В данном случае проверка происходит вручную организатором мероприятия.

Смешанный тип мероприятия

Смешанный тип мероприятия (тест + контрольная) – это мероприятие, включающее в себя совокупность всех особенностей вышеперечисленных мероприятий. Для каждого задания в отдельности предложено выбрать формат его представления.

**Слайд 10. Дизайн**

Минималистичный, интуитивно понятный. Белый фон, с элементами синего цвета на странице. В рамках проекта разработан логотип инструмента, который присутствует в «шапке» сайта. Сайт адаптивный и динамический.

**Слайд 12. Что такое сайт?**

Сайт — это совокупность связанных между собой веб-страниц, которые доступны в интернете через протоколы HTTP/HTTPS. Совокупность всех общедоступных веб-сайтов называют всемирной паутиной или сетью. (протокол передачи гипертекста)

Сайт — совокупность электронных документов (файлов) частного лица или организации в компьютерные сети, объединённых под одним адресом (доменным именем или IP-адрес). (Internet Protocol)

Все сайты в совокупности составляют Всемирную паутину, где коммуникация (паутина) объединяет сегменты информации мирового сообщества в единое целое — базу данных и коммуникации планетарного масштаба. Для прямого доступа клиентов к сайтам на серверах был специально разработан протокол HTTP.

**Слайд 13. Классификация инструмента по видам сайта**

Мой проект относится к web-приложению по функциональности.

По отзывчивости дизайна мой проект относится к адаптивному макету (В адаптивной разработке дизайна, макет определяет ширину экрана и с применением медиа-запросов будет корректироваться соответственно).

По способу формирования контента мой проект относится к динамическому виду сайта (Динамические сайты позволяют отображать различный контент при каждом посещении).

По типу контенту мой проект относится к образовательным ресурсам.

**Слайд 14. Что такое Django?**

Django — мощный веб-фреймворк на языке Python, предоставляющий обширные возможности для создания сложных сайтов и веб-приложений. Он состоит из набора компонентов, которые решают распространенные задачи веб-разработки, и эти компоненты слабо сцеплены между собой, что позволяет управлять ими независимо и разделять обязанности разных слоев веб-фреймворка.

**Слайд 15. Обоснование выбора Django**

При выборе инструмента для написания бэкенда сайта были рассмотрены следующие фреймворки на языке Python: Django, Flask, FastAPI.

Django является одним из самых популярных и проверенных временем фреймворков для разработки веб-приложений на Python. В отличие от Flask, который имеет меньший размер кодовой базы и больше подходит для небольших проектов, Django обеспечивает более полный и структурированный подход к разработке сложных веб-приложений с акцентом на управлении базами данных.

FastAPI, будучи самым современным и передовым фреймворком, предоставляет высокую производительность и поддержку асинхронных запросов. Однако, выбор Django более предпочтителен для нашего проекта, так как стабильность, многолетний опыт использования и обширное сообщество разработчиков имеют большее значение.

В целом, выбор фреймворка Django обоснован его проверенным качеством, богатым функционалом и поддержкой сложных веб-приложений с акцентом на управлении базами данных.

**Слайд 16. Проекты и приложения**

Один из основных принципов фреймворка — DRY (don't repeat yourself). Веб-системы на Django строятся из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из заметных архитектурных отличий этого фреймворка от некоторых других.

В Django проектом считается установленный веб-фреймворк Django с несколькими настроечными параметрами.

Приложение — это группа моделей, шаблонов, URL-адресов и представлений. Приложения взаимодействуют с веб-фреймворком с целью обеспечения определенных функциональностей, и их можно реиспользовать в разных проектах. (Uniform Resource Locator, Унифицированный указатель ресурса)

Проект можно трактовать как свой собственный веб-сайт, содержащий несколько приложений, которые другие проекты Django тоже могут использовать.

**Слайд 17. Компоненты веб-фреймворка**

● Модель – определяет логическую структуру данных и является обработчиком данных между базой данных и их представлением;

● Шаблон – это слой представления. В Django используется система текстовых шаблонов, в которой хранится все, что браузер прорисовывает на страницах;

● Представление – взаимодействует с базой данных через модель и передает данные в шаблон для их прорисовки и просмотра.

Сам веб-фреймворк выступает в качестве контроллера. Он отправляет запрос в надлежащее представление в соответствии с конфигурацией URL-адреса.

**Слайд 18. Архитектура**

На рисунке демонстрируется процесс обработки запросов Django включая различные ключевые компоненты системы, такие как модели, шаблоны, URL-адреса и представления, которые управляют циклом запроса/ответа.

Вот как Django оперирует HTTP-запросами и генерирует ответы:

1. Веб-браузер запрашивает страницу по ее URL-адресу, и веб-сервер передает HTTP-запрос веб-фреймворку Django.

2. Django просматривает свои сконфигурированные шаблоны URL-адресов и останавливается на первом, который совпадает с запрошенным URL-адресом.

3. Django исполняет представление, соответствующее совпавшему шаблону URL-адреса.

4. Представление потенциально использует модели данных, чтобы извлекать информацию из базы данных.

5. Модели данных обеспечивают определение данных и их поведение. Они используются для запроса к базе данных.

6. Представление прорисовывает шаблон (обычно с использованием HTML), чтобы отображать данные на странице, и возвращает их вместе с HTTP-ответом.

**Слайд 19. Что такое Bootstrap**

Bootstrap — это набор инструментов для разработки веб-сайтов и веб-приложений. Инструментарий включает в себя HTML, CSS и JavaScript шаблоны и компоненты, такие как меню, формы, кнопки и т.д. Bootstrap разработан, чтобы ускорить процесс создания веб-сайтов и дать возможность разработчикам создавать современные и отзывчивые веб-сайты. Bootstrap также предоставляет возможность создания адаптивных веб-сайтов, которые оптимально отображаются на разных мобильных устройствах, планшетах и настольных компьютерах.

**Слайд 20. Обоснование выбора Bootstrap**

Плюсы использования Bootstrap для фронтенд-разработки сайтов включают следующие аспекты:

• Быстрота создания качественной адаптивной верстки благодаря готовым классам и компонентам, разработанным профессионалами, что упрощает работу даже для начинающих веб-разработчиков.

• Кроссбраузерность и кроссплатформенность, обеспечивающие корректное отображение и функционирование сайта в разных браузерах и операционных системах.

• Большое количество готовых, хорошо продуманных компонентов, которые прошли проверку большим сообществом веб-разработчиков на разных устройствах.

• Возможность настройки фреймворка под свой проект с помощью изменения SCSS переменных и использования миксинов, что позволяет менять количество колонок, цвета, радиусы скругления, отступы и многое другое.

• Низкий порог вхождения, так как для работы с Bootstrap не нужны глубокие знания HTML, CSS и JavaScript - достаточно знать основы этих технологий.

• Единый стиль дизайна и согласованность между различными компонентами, что обеспечивает гармоничный внешний вид.

• Огромное количество сообществ и учебных материалов, которые помогут не только разобраться с фреймворком, но и найти ответы на возникающие вопросы.

Однако, Bootstrap не подходит для всех ситуаций. Например, его нецелесообразно использовать для:

● Создания уникального дизайна фронтенда проектов.

● Верстки проектов, если у вас достаточно времени и знаний в HTML, CSS и JavaScript для самостоятельной разработки.

Bootstrap, как и большинство подобных фреймворков, имеет недостатки. Среди них можно отметить следующие:

● Больший размер итоговых CSS и JS-файлов проекта по сравнению с самостоятельной разработкой, так как стили фреймворка и его JS-код содержат универсальный код для множества ситуаций, из которых в конкретном проекте может потребоваться только часть.

● Сложность использования Bootstrap для верстки сайтов с уникальным дизайном, так как в этом случае придется значительно переписывать код, и простая настройка переменных Bootstrap не будет достаточной.

Bootstrap был выбран в качестве инструмента для разработки фронтенда проекта, поскольку, несмотря на некоторые недостатки, этот фреймворк предлагает ряд значительных преимуществ. Основные причины выбора Bootstrap включают адаптивность, широкий набор компонентов, возможность стилизации и кастомизации тем, а также обширную документацию и активное сообщество разработчиков. Благодаря этим преимуществам, можно быстро и эффективно разрабатывать высококачественные веб-страницы и современные пользовательские интерфейсы, что играет ключевую роль в разработке нашего проекта.

**Информация на всякий случай**

Bootstrap включает в себя следующие компоненты:

● Инструменты для создания макетов, такие как обёрточные контейнеры, мощная система сеток, гибкие медиа-объекты и адаптивные утилитные классы.

● Классы для стилизации основного контента, включая текст, изображения, код, таблицы и элементы figure.

● Готовые компоненты, такие как кнопки, формы, горизонтальные и вертикальные навигационные панели, слайдеры, выпадающие списки, аккордеоны, модальные окна и всплывающие подсказки.

● Утилитные классы для решения традиционных задач, которые часто возникают перед веб-разработчиками, такие как выравнивание текста, отображение и скрытие элементов, задание цвета, фона, margin и padding отступов и т.д.

Некотрые возможности Django:

● ORM, API доступа к БД с поддержкой транзакций: Django предоставляет высокоуровневый ORM, который позволяет работать с базами данных, используя Python-код, без необходимости писать SQL-запросы. Он поддерживает транзакции для обеспечения целостности данных.

● Встроенный интерфейс администратора, с уже имеющимися переводами на многие языки: Django включает в себя готовый к использованию интерфейс администратора, который позволяет управлять данными вашего приложения прямо из браузера. Интерфейс администратора имеет переводы на разные языки и может быть легко настроен.

● Диспетчер URL на основе регулярных выражений: Django предоставляет мощный механизм маршрутизации URL, который позволяет определить пути к вашим представлениям на основе регулярных выражений. Это упрощает процесс создания SEO-оптимизированных URL-адресов.

● Расширяемая Инструмент шаблонов с тегами и наследованием: Django использует собственный язык шаблонов, который позволяет упростить процесс создания HTML-страниц. Язык шаблонов предоставляет теги и фильтры для обработки данных и поддерживает наследование шаблонов, что облегчает повторное использование кода.

● Инструмент кэширования: Django предлагает различные способы кэширования для улучшения производительности вашего веб-приложения. Можно кэшировать отдельные представления, части шаблона или результаты запросов к базе данных.

● Интернационализация: Django поддерживает многоязычность и позволяет легко создавать интернационализированные веб-приложения, используя встроенные инструменты для перевода текстов и форматирования даты и времени.

● Подключаемая архитектура приложений, которые можно устанавливать на любые Django-сайты: Django позволяет создавать модульные приложения, которые можно легко интегрировать в другие проекты на Django. Это облегчает повторное использование кода и расширение функциональности веб-приложения.

● «generic views» — шаблоны функций контроллеров: Django предоставляет набор предопределенных представлений, которые облегчают создание стандартных функций контроллеров, таких как отображение списка объектов или обработка форм.

● Авторизация и аутентификация. Django имеет встроенную систему аутентификации и авторизации, которая позволяет управлять пользователями и их правами доступа. Вы также можете подключить сторонние модули аутентификации, такие как LDAP или OpenID.

● Инструмент фильтров («middleware») в Django позволяет создавать дополнительные обработчики запросов. Встроенные фильтры включают кэширование, сжатие, нормализацию URL и поддержку анонимных сессий. Middleware — это система "плагинов" для глобального изменения входных и выходных данных Django. Каждый компонент middleware отвечает за выполнение определенной функции, например, аутентификацию пользователей через сессии.

● Библиотека для работы с формами в Django облегчает создание и наследование форм на основе существующих моделей базы данных. Это позволяет ускорить процесс разработки, так как не требуется создавать формы с нуля.

● В Django имеется встроенная автоматическая документация по тегам шаблонов и моделям данных, доступная через административное приложение. Это удобный инструмент для быстрого доступа к информации о структуре и возможностях Django.