Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

**Отчёт**

по предмету «Проектирование интернет систем»

Лабораторная работа №1

«Платформа для обучения и контроля знаний по математике»

Студент: Фамилия И. О.

ФИТ 4 курс 6 группа

Преподаватель: Якубенко К. Д.

Минск 2024

**Введение**

Проект «Платформа для обучения и контроля знаний по математике» направлен на создание удобной и гибкой системы для организации учебного процесса в цифровом формате.

Современные образовательные системы активно переходят в онлайн-пространство, что особенно заметно в условиях пандемии и глобальной цифровизации. Математика, как одна из ключевых дисциплин, требует особого подхода к обучению, где важны как теоретические знания, так и их практическое применение. Использование цифровых технологий позволяет преподавателям создавать разнообразные форматы занятий, тестов и интерактивных задач, что значительно повышает интерес учеников и их мотивацию к обучению.

Цели проекта:

* перевести учебные курсы, материалы и тесты по математике в онлайн-формат, обеспечив доступ к ним в любое время;
* обеспечить возможность проведения автоматических тестов с мгновенной оценкой, что облегчит преподавателям контроль за успеваемостью учеников;
* предоставить преподавателям и ученикам возможность отслеживать результаты и прогресс в обучении;
* обеспечивать возможность создания интерактивных математических моделей, что позволит делать процесс обучения более наглядным и интересным;
* упростить планирование занятий и тестов через интеграцию с системами календарей.

К целевой аудитории проекта можно отнести репетиторов по математике желающих организовать свои занятия в цифровом формате. Платформа предоставляет им все необходимые инструменты для создания курсов, тестов, а также управления расписанием занятий и отслеживания прогресса учеников. Кроме того, приложение направлено на учеников, которые предпочитают изучать математику онлайн. Платформа позволяет им получать учебные материалы, решать тесты и следить за своим прогрессом, получая регулярные отчеты об успеваемости.

Приложение объединяет преподавателей и студентов в едином цифровом пространстве, где можно создавать, распространять и анализировать учебные материалы. Преподаватели получают возможность разрабатывать интерактивные математические курсы, тесты и задачи, которые студенты могут проходить онлайн, отслеживая свои результаты и прогресс.

Система предоставляет удобные инструменты для контроля успеваемости, автоматического формирования отчетов и ведения статистики. Она адаптирована под потребности как индивидуальных репетиторов, так и образовательных учреждений, создавая гибкую и персонализированную среду для каждого пользователя.

1. **Назначение разработки**

Проект «Платформа для обучения и контроля знаний по математике» создается с целью формирования инновационного образовательного пространства, которое объединяет классические методики преподавания с передовыми цифровыми инструментами. Это приложение призвано облегчить организацию учебного процесса для преподавателей и репетиторов, оптимизировать их взаимодействие с учениками и предоставить современные средства для оценки и контроля знаний.

Основное назначение платформы заключается в том, чтобы автоматизировать рутинные задачи, такие как создание учебных материалов, формирование тестов и проверка ответов, тем самым снижая нагрузку на преподавателей и позволяя им сосредоточиться на непосредственном обучении. Для учеников система открывает доступ к разнообразным интерактивным материалам и тестам, что делает процесс обучения не только удобным, но и более интересным и понятным. Использование таких материалов способствует глубокому пониманию сложных математических концепций и обеспечивает возможность практического применения полученных знаний.

Приложение соединяет преподавателей и студентов в единой цифровой среде, где можно создавать, делиться и анализировать учебные материалы. Преподаватели могут разрабатывать интерактивные математические курсы, тесты и задачи, доступные для прохождения онлайн, в то время как студенты могут отслеживать свои результаты и прогресс обучения в реальном времени.

Целевая аудитория проекта включает репетиторов по математике, стремящихся организовать свои занятия в цифровом формате. Платформа предоставляет им полный набор инструментов для создания курсов, тестов, а также управления расписанием и отслеживания успеваемости учеников. Кроме того, приложение ориентировано на учеников, которые предпочитают изучать математику онлайн. Оно позволяет им получать учебные материалы, проходить тесты и отслеживать свой прогресс, получая регулярные отчеты об успеваемости.

Кроме того, платформа предоставляет мощные аналитические инструменты для мониторинга успеваемости учеников, что помогает преподавателям следить за прогрессом каждого учащегося и оперативно реагировать на трудности, с которыми они могут столкнуться. Интерактивные математические модели позволяют сделать процесс обучения более визуальным и доступным, что повышает вовлеченность учеников и эффективность занятий.

1. **Описание информационной и вербальной модели**

Объекты информационной вербальной модели для разрабатываемого проекта приведены в таблице 3.1.

Таблица 3.1 – Объекты информационной вербальной модели

|  |  |  |  |
| --- | --- | --- | --- |
| Объект | Описание | Атрибуты | Связи |
| Репетитор | Пользователь платформы с ролью преподавателя, создающий курсы, учебные материалы, тесты и отслеживающий прогресс учеников | Имя, фамилия, адрес электронной почты, список созданных курсов, список созданных тестов | Связан с курсами (репетитор создаёт и управляет курсами), учениками (отслеживает их успеваемость), тестами (создаёт и проверяет тесты) |
| Ученик | Пользователь платформы с ролью ученика, изучающий учебные материалы и проходящий тесты | Имя, фамилия, адрес электронной почты, прогресс по курсам | Связан с курсами (ученик записывается на курсы), тестами (ученик проходит тесты), результатами (получает оценки за тесты), прогрессом (отслеживает свой прогресс в обучении) |
| Курс | Образовательный модуль, включающий учебные материалы, тесты и интерактивные модели, созданные репетитором | Название курса, дата создания, дата последнего обновления, ссылка на страницу курса | Связан с репетитором (создаёт курс) и учениками (изучают курс и проходят тесты) |
| Тест | Проверка знаний ученика в рамках курса, созданная репетитором | Название теста, вопросы, дата создания | Связан с курсами (каждый тест принадлежит курсу), учениками (выполняют тест) и результатами (оценка за тест) |
| Вопрос теста | Задание, на которое ученик должен ответить в процессе прохождения теста | Текст вопроса, тип вопроса (множественный выбор, короткий ответ, правда/ложь) | Связан с тестами (вопрос принадлежит конкретному тесту) и ответами (возможные варианты ответа) |

Продолжение таблицы 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| Ответ на вопрос | Возможный вариант ответа на вопрос теста, который выбирает ученик | Текст ответа, правильность (правильный/неправильный) | Связан с вопросом теста (ответы соответствуют конкретному вопросу) |
| Результат | Оценка знаний ученика на основе выполнения тест | Количество баллов, дата завершения теста | Связан с учениками (результат за тест) и тестами (оценка за тест) |

Представленная информационная вербальная модель обеспечивает структурированное представление всех ключевых объектов системы, их атрибутов и взаимосвязей, что способствует эффективной реализации функционала платформы для обучения и контроля знаний по математике.

Построение информационной и вербальной модели является важным этапом разработки любого программного обеспечения, особенно в образовательных проектах, таких как «Платформа для обучения и контроля знаний по математике». Этот процесс помогает разработчикам и заказчикам четко понять, как будет функционировать система, какие данные она будет обрабатывать и как будут взаимодействовать пользователи.

Информационная модель описывает основные сущности системы и связи между ними. В случае образовательной платформы это могут быть такие объекты, как пользователи (репетиторы и ученики), курсы, тесты, вопросы и результаты тестов. Каждый объект наделен определенными атрибутами, например, у репетитора это имя, email, список курсов, у ученика — прогресс и успеваемость, а у тестов — вопросы и правильные ответы. Правильно выстроенная информационная модель позволяет эффективно организовать хранение данных, их обработку и взаимодействие между элементами системы.

Вербальная модель представляет собой текстовое описание функционала и взаимодействия элементов системы. Это важно для того, чтобы команды разработки, аналитики и заказчики одинаково понимали, что представляет собой каждый объект и как он должен работать в рамках проекта. Вербальная модель помогает наладить взаимопонимание и устранить двусмысленности, которые могут возникнуть при проектировании.

Построение этих моделей помогает обеспечить логичное и согласованное функционирование системы, повысить эффективность разработки, сократить время на доработки и уменьшить количество ошибок в процессе создания продукта.

1. **Требования к программному средству**
   1. Требования к клиентской части приложения

Клиентская часть представляет собой интерфейс, через который пользователи (ученики и репетиторы) взаимодействуют с системой. Основные функции включают регистрацию и авторизацию пользователей, управление учебными материалами, тестами, отслеживание прогресса и просмотр результатов.

Требования к пользовательскому интерфейсу:

* поддержка различных устройств (адаптивная вёрстка для мобильных телефонов, планшетов и компьютеров);
* интерфейс для создания курсов, материалов и тестов с удобными инструментами для форматирования;
* возможность просмотра учебных материалов и прохождения тестов учениками;
* личный кабинет ученика с отображением прогресса по курсам и тестам.

Таким образом, были сформированы основные требования к клиентской части приложения.

* 1. Требования к серверной части приложения

Серверная часть отвечает за обработку данных, выполнение бизнес-логики и обеспечение надежного хранения информации. Включает функционал для управления пользователями, обработку запросов на создание и редактирование учебных материалов, выполнение тестов, хранение результатов и обеспечение безопасности данных.

Авторизация и аутентификация:

* возможность входа через Google OAuth для учеников;
* поддержка ролей: ученик и репетитор;
* возможность для репетитора создания новых учеников с отправкой им ссылки для активации через Google OAuth.

Календарь и планирование:

* интеграция с Google Calendar для отслеживания занятий и тестов;
* репетитор имеет возможность создавать, редактировать и удалять занятия, которые отображаются у учеников в календаре.

Интерактивные модели:

* возможность создания интерактивных математических моделей с динамическим отображением графиков и формул.

Далее представлены требования к серверной части приложения.

Обработка данных:

* поддержка хранения и обработки данных о пользователях, курсах, тестах, результатах и прогрессе учеников;
* хранение статических страниц курсов, сгенерированных на сервере, с возможностью интеграции с React.

API:

* RESTful API для взаимодействия клиентской части с сервером, поддерживающее операции CRUD для всех сущностей (курсы, тесты, ученики, репетиторы);
* API для взаимодействия с внешними сервисами, такими как Google OAuth и Google Calendar.

Серверная часть должна быть надежной и масштабируемой, обеспечивая стабильную работу, обработку запросов, а также хранение и управление данными. Соблюдение всех требований обеспечит высокую производительность системы, её безопасность и удобство использования, что позволит успешно достигнуть целей проекта и удовлетворить потребности целевой аудитории.

Таким образом, были сформированы требования к серверной части приложения.

* 1. Требования к безопасности приложения

Основные требования к безопасности включают:

* валидация данных на серверной стороне;
* защита от атак типа SQL-инъекций и XSS;
* реализация ролевой системы доступа с разделением прав для учеников и репетиторов.

Требования к безопасности описывают меры и практики, которые будут внедрены для защиты данных и обеспечения безопасного функционирования веб-приложения. Это включает в себя защиту личных данных пользователей, предотвращение несанкционированного доступа и обеспечение защиты от потенциальных угроз и уязвимостей.

В результате анализа требований к программному средству были определены ключевые функциональные и технические характеристики, необходимые для эффективного функционирования системы. Программное обеспечение должно обеспечить удобство использования для всех категорий пользователей, от преподавателей до студентов, а также поддерживать основные функции, такие как регистрация, создание курсов, тестирование и отслеживание прогресса. Важно, чтобы клиентская часть системы предоставляла интуитивно понятный интерфейс, позволяя пользователям легко взаимодействовать с платформой.

1. **Требования к программной документации**

Требования к программной документации описывает обязательные элементы и стандарты документации, которые должны быть предоставлены для обеспечения полного понимания, поддержки и дальнейшего развития веб-приложения.

Техническое описание:

* детализированное описание архитектуры веб-приложения, включая структуру серверной и клиентской частей, базы данных и взаимодействие между компонентами;
* подробное описание доступных API-интерфейсов, методов, параметров и форматов запросов и ответов.

Руководство пользователя:

* пошаговые инструкции по установке, конфигурации и запуску приложения;
* описание пользовательского интерфейса, включая назначение элементов управления, формы и страницы.

Данный раздел важность наличия детализированной и структурированной документации на всех этапах разработки и эксплуатации программного обеспечения. Документация должна включать в себя полное описание архитектуры системы, технические спецификации, инструкции по установке, настройке и эксплуатации, а также материалы для конечных пользователей. Это позволит обеспечить прозрачность всех процессов разработки, поддержку и развитие программного продукта в будущем, а также упростит его использование для преподавателей и учеников.

Качественная документация способствует упрощению дальнейших доработок системы, облегчает её сопровождение и масштабирование, что критически важно для долгосрочного успеха проекта. Тщательная проработка документационных требований позволит избежать ошибок при внедрении, минимизирует время на обучение пользователей и поддержку системы. В результате, это не только повысит удовлетворенность конечных пользователей, но и ускорит процесс внедрения и использования платформы.

Таким образом, документация является критически важной для обеспечения корректной эксплуатации приложения, его дальнейшего развития и поддержки.

1. **Стадии и этапы разработки**

Стадии и этапы разработки описывает ключевые фазы процесса разработки веб-приложения, от начальной концепции до завершения и поддержки. Каждая стадия включает в себя конкретные этапы, задачи и цели, необходимые для успешного завершения проекта.

* 1. Предварительное планирование

Планирование включает в себя несколько этапов.

Первым этапом является анализ требований. Он включает в себя:

* сбор и анализ требований от заказчиков и пользователей;
* определение целей и задач проекта;
* создание первоначального документа с требованиями и спецификациями.

Следующим этапом является проектирование. Основные этапы проектирования веб-приложения:

* разработка архитектуры системы, включая выбор технологий и инструментов;
* создание схем баз данных и проектирование интерфейсов.

Завершающим этапом планирования является планирование ресурсов;

* определение необходимых ресурсов, включая разработчиков, дизайнеров, тестировщиков и другие ключевые роли;
* разработка графика проекта и оценка сроков выполнения.

Раздел "Предварительное планирование" играет ключевую роль в успешной реализации проекта, поскольку закладывает основу для всех последующих этапов разработки. Четкое определение целей, задач и необходимых ресурсов на этом этапе позволяет минимизировать риски, связанные с реализацией, и обеспечить выполнение всех требований в заданные сроки. Предварительное планирование включает в себя оценку технических и функциональных требований, распределение ролей в команде и определение ключевых контрольных точек.

Эффективное планирование помогает избежать непредвиденных задержек и отклонений от поставленных целей, что существенно повышает шансы на успешное завершение проекта. Четко продуманный план также способствует улучшению коммуникации внутри команды и гарантирует согласованность между всеми участниками разработки, обеспечивая прозрачность и последовательность действий на каждом этапе.

* 1. Разработка

Основные этапы разработки включают:

* создание прототипов пользовательского интерфейса и макетов;
* проведение пользовательских тестирований для получения обратной связи и доработки дизайна;
* реализация серверной логики, API-интерфейсов и баз данных;
* интеграция с внешними сервисами и библиотеками;
* реализация пользовательского интерфейса и взаимодействие с серверной частью;
* обеспечение кросс-браузерной совместимости и адаптивности интерфейса;
* интеграция всех компонентов системы и проведение интеграционных тестов;
* выполнение юнит-тестов, функциональных тестов и пользовательских тестов.

Этап разработки требует четкого соблюдения методологий, сроков и стандартов качества, что позволяет минимизировать ошибки и обеспечить надежную работу программного обеспечения. Грамотное управление процессом разработки гарантирует, что конечный продукт будет соответствовать ожиданиям пользователей и заказчиков, а также готов к дальнейшему использованию и расширению в будущем.

* 1. Завершение и развертывание

Этап завершения и развертывания включает следующие шаги:

* проведение окончательной проверки системы на соответствие требованиям и спецификациям;
* исправление выявленных ошибок и недочетов;
* подготовка и развертывание приложения на производственном сервере;
* настройка окружения и обеспечение бесперебойной работы системы;
* обучение пользователей и администраторов системы;
* подготовка и передача всей необходимой документации.

На этапе завершения и развертывания приложение подвергается финальному тестированию и подготовке к запуску, обеспечивая его готовность к полноценному функционированию в рабочей среде.

* 1. Поддержка и развитие

Поддержка и развитие проекта включают:

* оказание технической поддержки и решение возникающих проблем;
* регулярное обновление и поддержка системы;
* сбор отзывов пользователей и анализ эффективности приложения;
* реализация улучшений и новых функций на основе полученных данных.

Поддержка и развитие приложения включают в себя регулярное обновление, устранение возникающих проблем и добавление новых функций, что обеспечивает его долгосрочную актуальность и эффективность.

Эти стадии и этапы обеспечивают систематический подход к разработке, позволяя эффективно управлять проектом, контролировать качество и достигать поставленных целей.

**Заключение**

Проект нацелен на создание современного и эффективного инструмента, который сочетает традиционные педагогические методы с передовыми цифровыми технологиями. Эта платформа стремится объединить функционал преподавания и обучения в едином цифровом пространстве, предоставляя преподавателям возможности для более организованного планирования и проведения занятий, а ученикам — для более эффективного и интерактивного освоения материала. Система разработана с учетом потребностей образовательных учреждений и частных репетиторов, обеспечивая им мощные инструменты для управления курсами, тестами и оценкой успеваемости учеников.

Назначение разработки акцентирует внимание на ключевых целях и задачах системы, таких как автоматизация рутинных процессов, упрощение создания и управления учебными материалами, а также мониторинг прогресса учеников. Введение в информационную и вербальную модель помогает понять структуру и взаимосвязи ключевых объектов системы, таких как пользователи, курсы и тесты, что способствует более глубокому пониманию функциональных возможностей и архитектуры проекта. Требования к программному средству детализируют как клиентские, так и серверные аспекты, обеспечивая четкие и конкретные руководства для разработки, внедрения и тестирования системы.

Этапы разработки системы охватывают весь процесс создания — от предварительного планирования до окончательного развертывания и последующей поддержки. Предварительное планирование включает в себя анализ требований и определение архитектуры, разработка охватывает создание функционала и интерфейсов, завершение и развертывание направлены на тестирование и внедрение системы в рабочую среду, а поддержка и развитие обеспечивают её актуальность и эффективность в течение всего жизненного цикла. Каждый из этих этапов критически важен для успешного запуска и эксплуатации платформы, а также для её постоянного улучшения в ответ на изменяющиеся потребности пользователей.

Таким образом, данная документация представляет собой детализированный и структурированный план для успешной реализации проекта. Она предоставляет всестороннее руководство по созданию и внедрению платформы для обучения и контроля знаний по математике, обеспечивая ясное понимание целей, требований и этапов разработки, что является ключевым для достижения успешного результата.