МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра информационных технологий управления

Техническое задание к проекту «Ведение домашнего бюджета»

в соответствии с ГОСТ 34.602-89

Исполнитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.С. Семенихин, 3 курс, д/о

Исполнитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Еремеев, 3 курс, д/о

Исполнитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.Е. Парфенова, 3 курс, д/о

Заказчик \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.С. Тарасов, старший преподаватель

Воронеж 2023

Содержание

[1 Понятие и термины 4](#_Toc146651830)

[2 Общие сведения 6](#_Toc146651831)

[2.1 Полное наименование системы и название веб-приложения 6](#_Toc146651832)

[2.2 Наименование исполнителя и заказчика приложения 6](#_Toc146651833)

[2.2.1 Наименование исполнителя 6](#_Toc146651834)

[2.2.2 Наименование заказчика 6](#_Toc146651835)

[2.3 Перечень документов, на основании которых создается сайт 6](#_Toc146651836)

[2.4 Плановые сроки начала и окончания работ 6](#_Toc146651837)

[2.5 Порядок оформления и предъявления заказчику результатов работ по созданию веб-приложения 6](#_Toc146651838)

[3 Назначение и цели создания веб-приложения 8](#_Toc146651839)

[3.1 Назначение веб-приложения 8](#_Toc146651840)

[3.2 Цели создания веб-приложения 8](#_Toc146651841)

[3.3 Характеристика объектов автоматизации 8](#_Toc146651842)

[4 Требования к веб-приложению и программному обеспечению 9](#_Toc146651843)

[4.1 Требования к веб-приложению в целом 9](#_Toc146651844)

[4.1.1 Требования к структуре и функционированию веб-приложения 9](#_Toc146651845)

[4.1.2 Требования к оформлению и верстке страниц 9](#_Toc146651846)

[4.1.3 Требования к используемым технологиям 10](#_Toc146651847)

[4.2 Требования к функциям, выполняемым веб-приложением 11](#_Toc146651848)

[5 Языковые версии веб-приложения 12](#_Toc146651849)

[6 Группы пользователей 13](#_Toc146651850)

[7 Дизайн сайта 14](#_Toc146651851)

[8 Навигация по веб-приложению 15](#_Toc146651852)

[9 Описание страниц сайта 16](#_Toc146651853)

[9.1 Запуск приложения 16](#_Toc146651854)

[9.2 Страница «Домой» 16](#_Toc146651855)

[9.3 Страница «Оценки студентов» 16](#_Toc146651856)

[10 Перспективы развития, модернизации веб-приложения 17](#_Toc146651857)

[11 Требования к документированию 18](#_Toc146651858)

[12 Порядок контроля и приемки работы 19](#_Toc146651859)

[Приложение 20](#_Toc146651860)

1. Понятие и термины

* Авторизация – это процесс проверки прав пользователя на осуществление определенных действий на сайте;
* База данных – это упорядоченный набор структурированной информации или данных, которые хранятся в электронном виде в компьютерной системе;
* Веб-приложений, веб-сервис, интернет-сервис, онлайн-сервис, проект – это программное обеспечение, которое размещено на удаленном сервере и доступно через браузеры в интернете;
* Клиент – это объект, запрашивающий информацию по сети;
* Личный кабинет, профиль – это раздел сервиса, в котором пользователь может получить доступ к персональным данным;
* Развертывание – это все действия, которые делают систему готовой к использованию;
* Регистрация – это способ сообщить сервису данные о себе и в обмен получить доступ к дополнительным ресурсам на сайте, которые недоступны гостям;
* Сервер – это отдельный класс компьютерных устройств, предназначенных для обработки запросов от различных узлов сети;
* СУБД – это система управления базы данных;
* Фреймворк – это программная среда, облегчающая разработку и объединение разных компонентов большого программного проекта;
* Frontend – это клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса;
* Backend – это программно-аппаратная часть сервиса;
* REST API – это стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов;
* React – это JavaScript библиотека для создания пользовательских интерфейсов;
* SPA – это одностраничное веб-приложение, которое работает на одной HTML-странице, обновляя данные на ней;
* GitHub – это крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки;
* HTTP, HTTPS – это широко распространённый протокол передачи данных, предназначенный для передачи документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам;
* HTML – это стандартизированный язык разметки документов для просмотра веб-страниц в браузере;
* Header – это визуальный элемент, расположенный в верхней части страницы;
* MVC архитектура – это способ организации кода, который предполагает выделение блоков, отвечающих за решение разных задач;
* SQL-запросы – это наборы команд для работы с реляционными (табличными) базами данных;

1. Общие сведения
   1. Полное наименование системы и название веб-приложения

Полное наименование: приложение для учета успеваемости студентов.

Название веб-приложения: Academic Performance.

* 1. Наименование исполнителя и заказчика приложения
     1. Наименование исполнителя

Студент 3-го курса Семенихин Константин Сергеевич, кафедра информационных систем и технологий управления предприятием.

Студент 3-го курса Еремеев Александр Алексеевич, кафедра информационных систем и технологий управления предприятием.

Студент 3-го курса Парфенова Екатерина Евгеньевна, кафедра информационных систем и технологий управления предприятием.

* + 1. Наименование заказчика

Старший преподаватель Тарасов Вячеслав Сергеевич, кафедра программирования и информационных технологий.

* 1. Перечень документов, на основании которых создается сайт

На основании Технического Задания, документа «Стратегия повышения финансовой грамотности в Российской Федерации на 2017-2023 годы» и закона «Об информации, информационных технологиях и о защите информации» от 27.07.2006 №149-ФЗ создается данное веб-приложение.

* 1. Плановые сроки начала и окончания работ

Плановый срок начала работ – март 2023 г.

Плановый срок окончания работ – сентябрь 2023 г.

* 1. Порядок оформления и предъявления заказчику результатов работ по созданию веб-приложения

Предъявление результатов работ заказчику осуществляется в следующем виде:

* Работающее веб-приложение согласно Техническому Заданию;
* Документация к веб-приложению;
* Презентация и видео с демонстрацией функциональности веб-приложения;
* Исходный код веб-приложения;
* Защита проекта.

1. Назначение и цели создания веб-приложения
   1. Назначение веб-приложения

Назначением приложения является учет успеваемости студентов.

* 1. Цели создания веб-приложения

Упростить процесс обучения и преподавания.

* 1. Характеристика объектов автоматизации

Объектом автоматизации является процесс мониторинга и выставления баллов, а также управление списком студентов от лица преподавателей.

1. Требования к веб-приложению и программному обеспечению
   1. Требования к веб-приложению в целом

Приложение должно удовлетворять следующим требованиям:

* Обеспечить работу в Google Chrome, Firefox и Microsoft Edge;
* Предоставить возможность регистрации и авторизации;
* Предоставить панель администратора для преподавателей;
* Предоставить удобный интерфейс для отслеживания баллов.
  + 1. Требования к структуре и функционированию веб-приложения

При разработке клиентской части будет использоваться библиотека React, которая позволяет разрабатывать SPA, а также предлагает компонентный подход, при котором интерфейсы собираются из отдельных элементов (компонентов).

Приложение должно запускаться в следующих браузерах:

* Google Chrome (версия 90.0.4430.93 или новее);
* Firefox (версия 90.0.2 или новее);
* Microsoft Edge (версия 90.0.818.39 или новее).

Для связи клиента и сервера следует использовать REST API, которое представляет собой интерфейс, используемый двумя компьютерными системами для безопасного обмена информацией через сеть Интернет.

В качестве основного языка программирования будет использован JavaScript.

* + 1. Требования к оформлению и верстке страниц

Страницы должны быть сверстаны с использованием принципов резиновой верстки, которые позволяют перестроить и адаптировать интерфейс к различным размерам экрана.

Также должна быть реализована кросс-браузерная поддержка веб-приложения, которая заключается в том, что разрабатываемый веб-сайт должен отображаться и функционировать во всех перечисленных ниже браузерах идентично. Под идентичностью функционирования подразумевается: отсутствие некорректной работы, отсутствие ошибок в верстке и способность отображать материал с одинаковой степенью читабельности.

Обеспечить поддержку следующих браузеров:

* Google Chrome (версия 90.0.4430.93 или новее);
* Firefox (версия 90.0.2 или новее);
* Microsoft Edge (версия 90.0.818.39 или новее).

В качестве СУБД будет использоваться реляционная СУБД PostgreSQL.

* + 1. Требования к используемым технологиям

При разработке Frontend части приложения будут использованы следующие технологии:

* JavaScript (JS) – язык программирования, который выполняется внутри браузера и позволяет внедрять в сайт различные функции на стороне клиента;
* Приложение будет оперировать реляционной БД, в качестве СУБД будет использоваться СУБД с открытым исходным кодом PostgreSQL;
* Для документации разрабатываемого REST API будет использоваться Swagger, предоставляющий набор инструментов, который позволяет автоматически описывать API на основе его кода;
* React – библиотека с открытым исходным кодом, позволяющая создавать одностраничные приложения, работает как с JS, так и с TS.

При разработке Backend части приложения будут использованы следующие технологии:

* Основной язык – Python, framework Django;
* Приложение будет оперировать реляционной БД, в качестве СУБД будет использоваться СУБД с открытым исходным кодом SQLite;
* В качестве инструмента развертки приложения будет использоваться Docker, который позволяет автоматизировать процесс развертывания и управления приложениями.

Сервисы, используемые в процессе разработки:

* Git – распределенная система контроля версий, позволяющая отслеживать, вносить и удалять изменения;
* GitHub – крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git;
* Miro – платформа для визуализации гипотез и моделей, предоставляющая возможности для совместной работы распределенных команд;
* Trello – облачная программа для управления проектами небольших групп, позволяет наглядно отслеживать задачи каждого участника разработки;
* Figma – онлайн-сервис для разработки и прототипирования интерфейсов с возможностью организации совместной работы в режиме реального времени.
  1. Требования к функциям, выполняемым веб-приложением

Разрабатываемое веб-приложение должно обладать следующими функциональными возможностями:

* Возможность регистрации и авторизации для пользователей;
* Возможность преподавателя зайти в приложение в качестве администратора;
* Возможность просмотра студентом своих баллов по разным предметам.

1. Языковые версии веб-приложения

На данном этапе разработки предусмотрена только русскоязычная версия приложения.

1. Группы пользователей

В системе существуют такие группы пользователей как: студент, преподаватель, администратор.

Пользователь «студент» – посетитель веб-сайта, зарегестрированный в базе данных университета. Для данного пользователя должна быть реализована следующая функциональность:

* Регистрация в веб-приложении;
* Просмотр своей успеваемости;
* Просмотр информации о курсе, старосте и кураторе;
* Возможность выхода из аккаунта.

Пользователь «преподаватель» – пользователь веб-сайта, работающий преподавателем в университете. Для данного пользователя должна быть реализована следующая функциональность:

* Просмотр баллов студентов и их рейтинга;
* Выставление баллов по своему предмету;
* Возможность выхода из аккаунта.

Пользователь «администратор» – пользователь веб-сайта, авторизовавшийся в системе под учетными данными администратора. Для данного пользователя должна быть реализована следующая функциональность:

* Возможность добавления новых пользователей;
* Возможность добавления новых направлений;
* Возможность добавления новых дисциплин;
* Возможность добавления новых групп студентов;

Возможность редактирования баллов.

1. Дизайн сайта

Дизайн сайта должен быть выполнен в классическом стиле. Основные цвета: базовые (#FFFEFF, #E2E3E6), второстепенные (#F8493F, #FEEEC2, #C1DDE0, #C1DDC0).

1. Навигация по веб-приложению

Для навигации приложение использует меню в верхней части экрана, в котором содержатся кнопки «Домой», «Оценки студента» (только у преподавателей), «Выход».

Для пользователей «Студент» и «Преподаватель» начальным экраном по умолчанию является вкладка «Домой».

1. Описание страниц сайта
   1. Запуск приложения

При первом запуске приложения перед неавторизованным пользователем появляются следующие элементы:

* Форма входа;
* Поля для ввода информации, необходимой для регистрации пользователя в системе;
* Кнопка «Войти».

Если введенные данные оказались верны, то после нажатия на кнопку «Войти» происходит переход на страницу «Домой». Если при входе возникла какая-то ошибка, пользователь остается на той же странице и получает сообщение об ошибке.

* 1. Страница «Домой»

После авторизации пользователь попадает на страницу «Домой». Если пользователем является студент, то отрывшаяся страница будет представлять таблицу, в которой содержится вся информация об успеваемости. Если пользователем является преподаватель, то отрывшаяся страница будет представлять таблицу с занятиями, внутри которых можно проставлять студентам оценки.

* 1. Страница «Оценки студентов»

Данная страница доступна только преподавателям. На данной странице содержится список студентов. После выбора нужного студента откроется таблица с его успеваемостью.

1. Перспективы развития, модернизации веб-приложения

* Добавления расширенной статистики об успеваемости;
* Добавление темной темы;
* Взаимодействие между пользователями;
* Рейтинговая система;
* Локализация на иные языки (прежде всего английский).

1. Требования к документированию

Документирование системы ведется в рамках Технического Задания в соответствие с ГОСТ 34.602-89.

Предоставление курсового проекта осуществляется на основе данного Технического задания.

Отслеживание рабочего процесса и управлением задачами участников проекта производится при помощи инструмента Trello.

Описание основных сценариев работы приложения осуществляется в сервисе Miro.

Документирование API обеспечивается с помощью инструмента Swagger.

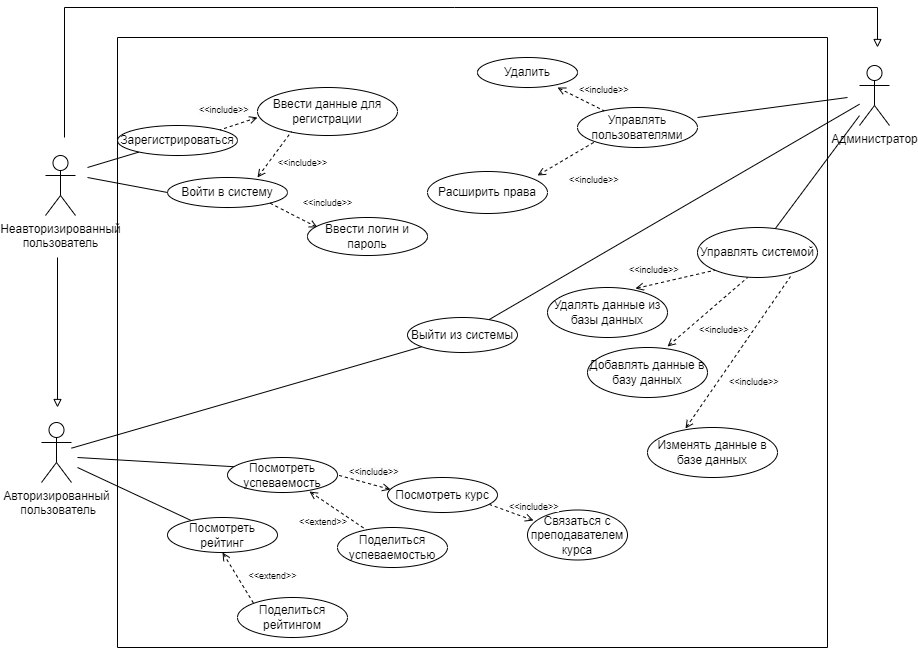
1. Порядок контроля и приемки работы

Исполнитель должен предоставить следующее:

* Техническое задание;
* Демонстрационную версию проекта со всеми ключевыми сценариями;
* Курсовой проект;
* Видео-презентацию работы приложения;
* Исходный код системы.

Приложение

Диаграмма прецедентов:



Диаграммы состояний:

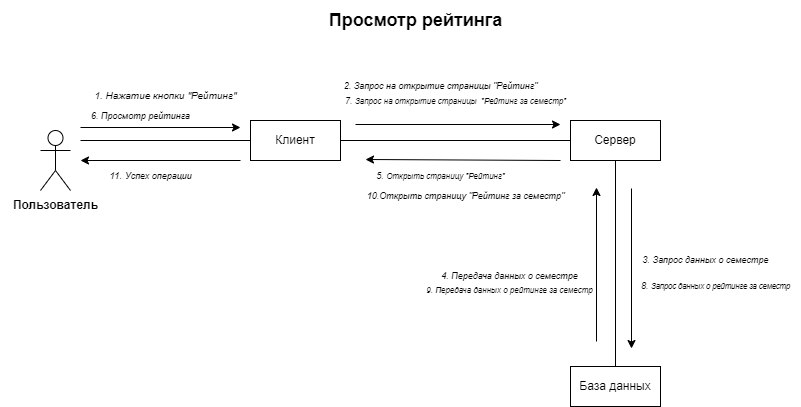




Диаграмма активностей:

