1.[Понятие и термины](#_30j0zll)

2. [Общие сведения](#_1fob9te)

2.1[Полное наименование системы и название веб-приложения](#_3znysh7)

2.2[Наименование исполнителя и заказчика приложения](#_2et92p0)

2.2.1 [Наименование исполнителя](#_tyjcwt)

2.2.2 [Наименование заказчика](#_3dy6vkm)

[2.3 Плановые сроки начала и окончания работы](#_4d34og8)

[2.4 Порядок оформления и предъявления заказчику результатов работ по](#_2s8eyo1) [созданию веб-приложения](#_2s8eyo1)

3. [Назначение и цели создания веб-приложения](#_17dp8vu)

3.1[Назначение веб-приложения](#_3rdcrjn)

3.2 [Цели создания веб-приложения](#_26in1rg)

3.3 [Характеристика объектов автоматизации](#_lnxbz9)

4. [Требования к веб-приложению и программному обеспечению](#_35nkun2)

4.1 [Требования к структуре и функционированию веб-приложения](#_44sinio)

4.2 [Требования к защите информации](#_2jxsxqh)

4.3 [Требования к оформлению и верстке страниц](#_z337ya)

4.4 [Требования к архитектуре](#_3j2qqm3)

4.5 [Требования к используемым технологиям](#_1y810tw)

4.6 [Требования к функциям приложения](#_4i7ojhp)

5. [Язык приложения](#_2xcytpi)

6. Навигация

7. Дизайн

8. Описание составляющих сайта

9. Возможное улучшение приложения

**Понятие и термины**

Для более точного определения понятий в данной задаче необходимо ввести следующие термины:

-Студент – человек, который обучается на факультете;

-Успеваемость – оценка знаний студента по определенным предметам за определенный период времени;

-Средний балл – среднее значение оценок по всем предметам за определенный период времени;

-Факультет – структурное подразделение университета, где проводится обучение по определенным специальностям и курсам.

-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.

-Личный кабинет, профиль – это раздел сервиса, в котором пользователь может получить доступ к персональным данным

-Сервер – это отдельный класс компьютерных устройств, предназначенных для обработки запросов от различных узлов сети.

-Frontend – это клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.

-Backend – это программно-аппаратная часть сервиса.

-HTML – это стандартизированный язык разметки документов для просмотра веб-страниц в браузере.

**Общие сведения**

**Полное наименование системы и название веб-приложения**

Полное наименование: Учет успеваемости студентов на факультете.

Название веб-приложения: Study Chek.

## Наименование исполнителя и заказчика приложения

**Наименование исполнителя**

Студент 3-го курса Семенихин Константин Сергеевич

Студент 3-го курса Парфенова Екатерина Евгеньевна

## Наименование заказчика

Старший преподаватель Тарасов Вячеслав Сергеевич

**Плановые сроки начала и окончания работы**

Плановый срок начала работ – март 2023 г. Плановый срок окончания работ – июнь 2023 г.

## 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.

## Назначение и цели создания веб-приложения

## Назначение веб-приложения

Назначением приложения является структурирование результатов учебной деятельности студентов.

## Цели создания веб-приложения

Создать программное обеспечение для учета успеваемости студентов, которое позволит автоматически генерировать отчеты по результатам сессии и формировать рейтинг на основании оценок.

## Характеристика объектов автоматизации

Объектом автоматизации в нашей работе является процесс структурирования и показа отметок пользователей (студентов) за определенный период времени и зп определенный предмет.

## Требования к веб-приложению и программному обеспечению

## 4.1 Требования к структуре и функционированию веб- приложения

При разработке клиентской части будет использоваться библиотека React, которая позволяет разрабатывать SPA при помощи инкапсулированных компонентов с собственным состоянием, а также с их дальнейшим объединением в сложные пользовательские интерфейсы.

Должна быть реализована стабильная работа приложения в следующих браузеров:

Google Chrome

Microsoft Edge

Для связи клиента и сервера следует использовать RESTful API, которое представляет из себя интерфейс, используемый двумя

компьютерными системами для безопасного обмена информацией через сеть Интернет.

В качестве основного языка программирования будет использован Java.

## 4.2Требования к защите информации

Разрабатываемое веб-приложение должно быть защищено от самых распространённых способов взлома и несанкционированного доступа, к примеру от SQL и XSS инъекций. Пароли пользователей в базе данных должны хешироваться.

## 4.3Требования к оформлению и верстке страниц

Для удобства пользователя страницы должны быть сверстаны с использованием принципов резиновой верстки, которые позволяют перестроить и адаптировать интерфейс к различным размерам экрана конечного потребителя услуг.

Также должна быть реализована кроссбраузерная поддержка веб- приложения, которая заключается в том, что разрабатываемый веб-сайт должен отображаться и функционировать во всех часто используемых 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

Microsoft Edge

## 4.4Требования к архитектуре

Серверная архитектура должна быть реализована в соответствии с подходом MVC, который подразумевает разделение данных и логики приложения на три отдельных части, а именно модель (model), вид (view) и контроллер (controller). Контроллер обрабатывает входящие запросы на сервер. Модель достаёт из базы данных информацию, нужную для выполнения конкретных запросов. Представление определяет результат запроса, который получает пользователь. Для связи клиент-сервер следует реализовать REST API.

## 4.5Требования к используемым технологиям

При разработке Frontend части приложения будут использованы следующие технологии:

Java - мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript. Именно он исполняется браузерными движками и позволяет внедрять в сайт различные функции на стороне клиента.

TypeScript - язык программирования, представленный Microsoft и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript, самый популярный сценарий использования - введение в код на этапе разработки четкую систему типов, которая позволяет избежать различных ошибок в типизации при компиляции в JS. Является обратно

совместимым с JS и компилируется в него же. Упрощает поддержку и масштабирование JS приложений со стороны разработчика.

React - библиотека с открытым исходным кодом, позволяющая упростить создание одностраничных приложений, работает как с JS, так и с TS. Поддерживает декларативный подход, который помогает сделать код более предсказуемым и упрощает дальнейшую отладку приложения.

При разработке Backend части приложения будут использованы следующие технологии:

Основной язык TypeScript исполняемый на сервере в среде Node.js в связке с фреймворком Nest.js. Nest.js — это платформа для создания эффективных, масштабируемых серверных приложений на Node.js. Сам фреймворк построен с использованием TypeScript и полностью поддерживает его (но при этом позволяет разработчикам использовать чистый JavaScript, что добавляет ему гибкости). Сочетает в себе элементы объектно-ориентированного программирования, функционального программирования и функционально-реактивного программирования. Позволяет реализовать MVC архитектуру.

Приложение будет оперировать реляционной БД, в качестве СУБД будет использоваться PostgreSQL в силу своей бесплатной модели распространения, а также широкой поддержки как со стороны серверов, так и со стороны сообщества.

В качестве инструмента развертки приложения будет использоваться Docker, который позволяет автоматизировать процесс развертывания и управления приложениями в средах с поддержкой системы контейнеризации.

Для документации и упрощения понимания особенностей

разрабатываемого REST API будет использоваться Swagger,

предоставляющий набор инструментов, который позволяет автоматически описывать API на основе его кода.

Сервисы, используемые в процессе разработки:

Git - распределенная система контроля версий, позволяющая удобно разрабатывать приложение, а также 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. Имеет широкий спектр возможности, от предоставления функционала удаленного репозитория до настройки процесса CI/CD при помощи GitHub Actions.

Miro - платформа для визуализации гипотез и моделей, предоставляющая возможности для совместной работы распределенных команд.

Trello - облачная программа для управления проектами небольших групп, поддерживает методологию управления канбан, позволяет наглядно отслеживать задачи каждого участника разработки.

Figma - онлайн-сервис для разработки и прототипирования интерфейсов с возможностью организации совместной работы в режиме реального времени.

## 4.6Требования к функциям, выполняемым веб-приложением

Авторизация

Пользователь должен иметь возможность войти в систему, используя свой уникальный логин и пароль.

- Ввод данных о студентах

Пользователь должен иметь возможность добавлять и редактировать данные студентов, включая ФИО, группу, контактную информацию.

- Ввод данных об успеваемости студентов

Пользователь должен иметь возможность добавлять и редактировать информацию об оценках студентов за каждый экзамен, зачет или курсовую работу.

-Генерация отчетов

Система должна позволять генерировать отчеты по результатам сессии, включая информацию о студентах, их оценках, рейтингах и средней оценке группы.

-Формирование рейтинга студентов

Система должна автоматически формировать рейтинг студентов на основании их оценок за сессию.

## Языковые версии веб-приложения

На данном этапе разработки предусмотрена только русскоязычная версия веб-приложения.

## Группы пользователей

В системе существуют такие пользователи: студент, преподаватель.

Для студента должна быть реализована следующая функциональность:

Возможность просмотра своих оценок за текущий семестр и за предыдущие

Возможность просмотра рейтинга

Возможность задать вопрос преподавателю

Для преподавателя:

Возможность выставления оценок

Возможность просмотра списка студентов

Возможность отправлять сообщения

## Дизайн сайта

Дизайн сайта должен быть выполнен минималистично, понятно для пользователя. Основной шрифт Montserrat. Допускается использование градиентов и теней, выдержанных в общей палитре.

## Навигация по веб-приложению

Данное приложение использует для навигации меню в верхней части экрана. Кнопка меню открывает всплывающее окно, которое позволяет перейти в следующие разделы сайта: «профиль», «главная страница»,

«статистика», «премиум-подписка».

Для пользователя, который не состоит в группе начальным экраном будет являться вкладка «личный бюджет». Для пользователя, состоящего в группе, начальная вкладка по умолчанию «общий бюджет».

## Описание составляющих сайта

**Запуск приложения**

На странице содержатся следующие элементы:

Форма входа

Кнопка «Войти в систему».

**Главная страница**

После авторизации пользователь попадает на главную страницу, которая представляет собой таблицу с оценками и названиями предметов.

## Перспективы развития, модернизации веб-приложения

Развитие демо-режима путем внедрения возможности добавлять пользовательские даты, а не только просматривать предзаписанные.

Добавление функции установки ограничения для конкретной категории трат, выбранной пользователем. Если лимит превышен система сигнализирует об этом всплывающим сообщением.

Добавление другой цветовой темы в приложение.

Переход от резиновой верстки к адаптивной, локализация на иные языке (прежде всего английский), добавление поддержки мультивалютности, интеграция с банковскими API.

## 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 серверной части приложения обеспечивается с помощью инструмента Swagger.

## Порядок контроля и приемки работы

Контроль разработки системы осуществляется путём регулярных встреч каждые две недели с представителем заказчика Зениным К.В. для обсуждения промежуточных результатов работы в рамках рубежных аттестаций. Готовая система с полной документацией будет представлена заказчику в обозначенную им дату. Заказчик определит соответствие системы всем требованиям и осуществит её приём.

Исполнитель должен предоставить следующее:

Техническое задание.

Демонстративную версию проекта со всеми ключевыми сценариями.

Курсовой проект.

Видео-презентацию работы приложения.

Исходный код системы.

**Реквизиты и подписи сторон**

СОСТАВИЛИ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование организации, предприятия | Должность исполнителя | Фамилия, имя, отчество | Подпись | Дата |
| Воронежский Государственный Университет | Студент | Пустовалов Артем Алексеевич |  | 24.03.2023 |
| Воронежский Государственный Университет | Студент | Новиков Виктор Григорьевич |  | 24.03.2023 |
| Воронежский Государственный Университет | Студент | Бордюжа Екатерина Олеговна |  | 24.03.2023 |

СОГЛАСОВАНО

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование организации, предприятия | Должность исполнителя | Фамилия, имя, отчество | Подпись | Дата |
| Воронежский Государственный Университет | Старший преподаватель | Тарасов Вячеслав Сергеевич |  | 24.03.2023 |
| Воронежский Государственный Университет | Ассистент | Зенин Кирилл Вячеславович |  | 24.03.2023 |