МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Техническое задание

на разработку учебного проекта

«Система учета успеваемости студентов»

Группа 231-3212

Студент Васильев Владимир

Дата 09.09.2025

2024

Оглавление

[1. ОБЩИЕ СВЕДЕНИЯ 3](#_Toc17)

[1.1. Полное наименование системы: 3](#_Toc18)

[1.2. Условное обозначение: 3](#_Toc19)

[1.3. Заказчик: 3](#_Toc20)

[1.4. Разработчик: 3](#_Toc21)

[1.5. Основание для разработки: 3](#_Toc22)

[2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ 3](#_Toc23)

[2.1. Назначение Системы 3](#_Toc24)

[2.2. Цели создания Системы 3](#_Toc25)

[3. ХАРАКТЕРИСТИКА ОБЪЕКТА АВТОМАТИЗАЦИИ 3](#_Toc26)

[4. ТРЕБОВАНИЯ К СИСТЕМЕ 4](#_Toc27)

[4.1. Требования к Системе в целом 4](#_Toc28)

[4.2. Функциональные требования 4](#_Toc29)

[4.3. Не функциональные требования 5](#_Toc30)

[5. СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ 6](#_Toc31)

[5.2. Требования к программному обеспечению 6](#_Toc32)

[5.3. Требования к лингвистическому обеспечению 7](#_Toc33)

[6. СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ 7](#_Toc34)

[7. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ СИСТЕМЫ 7](#_Toc35)

[8. ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ 8](#_Toc36)

# 1. ОБЩИЕ СВЕДЕНИЯ

## 1.1. Полное наименование системы:

Система учета успеваемости студентов.

## 1.2. Условное обозначение:

Система.

## 1.3. Заказчик:

Преподаватель.

## 1.4. Разработчик:

Студент Васильев Владимир , группа 231-3212.

## 1.5. Основание для разработки:

Учебный план дисциплины Автоматизация процессов жизненного цикла программных средств».

# 2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

## 2.1. Назначение Системы

Система предназначена для автоматизации процессов учета контингента студентов, учебных планов и журнала успеваемости в рамках образовательного учреждения.

## 2.2. Цели создания Системы

Основными целями создания Системы являются:

Практическая цель: Создание программного продукта, реализующего функционал по сбору, хранению, обработке и предоставлению данных об успеваемости студентов.

Учебная цель: Приобретение и закрепление практических навыков проектирования, разработки, тестирования и документирования программного обеспечения с использованием современных технологий (Python, Django, SQLite).

# 3. ХАРАКТЕРИСТИКА ОБЪЕКТА АВТОМАТИЗАЦИИ

Объектом автоматизации являются процессы, связанные с:

Ведением списка студентов (контингента).

Управлением учебными планами и перечнем дисциплин.

Регистрацией и хранением оценок студентов по различным дисциплинам.

# 4. ТРЕБОВАНИЯ К СИСТЕМЕ

## 4.1. Требования к Системе в целом

4.1.1. Система должна быть реализована в виде монолитного веб-приложения с клиент-серверной архитектурой.

4.1.2. Система должна обеспечивать работу в многопользовательском режиме, однако в рамках данного ТЗ предполагается использование одной основной роли пользователя – «Администратор», который обладает полными правами на доступ и изменение всех данных в Системе.

4.1.3. Доступ к Системе должен осуществляться через стандартные веб-браузеры.

## 4.2. Функциональные требования

Система должна обеспечивать выполнение следующих функций:

4.2.1. Подсистема управления данными (CRUD-операции)

Данная подсистема должна предоставлять интерфейс для выполнения операций создания, чтения, обновления и удаления (CRUD) для всех ключевых сущностей Системы.

4.2.1.1. Управление сущностью «Студент»:

Создание новой записи о студенте с указанием всех атрибутов.

Просмотр списка всех студентов с возможностью поиска и фильтрации.

Редактирование атрибутов существующей записи о студенте.

Удаление записи о студенте из Системы.

4.2.1.2. Управление сущностью «Дисциплина»:

Создание новой записи о дисциплине.

Просмотр списка всех дисциплин.

Редактирование атрибутов существующей записи о дисциплине.

Удаление записи о дисциплине.

4.2.1.3. Управление сущностью «Оценка» (Журнал успеваемости):

Создание новой записи об оценке, устанавливающей связь между студентом и дисциплиной.

Просмотр списка всех оценок с возможностью поиска и фильтрации по студенту или дисциплине.

Редактирование атрибутов существующей записи об оценке.

Удаление записи об оценке.

4.2.2. Подсистема информационно-аналитических запросов

Данная подсистема должна предоставлять пользователю инструменты для получения агрегированной информации.

4.2.2.1. Функция «Расчет количества студентов»:

Пользователю предоставляется возможность выбрать форму обучения из предопределенного списка (дневная, вечерняя, заочная).

После выбора Система выполняет запрос к базе данных и отображает на странице итоговое количество студентов, обучающихся по выбранной форме.

4.2.2.2. Функция «Получение информации о дисциплине»:

Пользователю предоставляется текстовое поле для ввода названия искомой дисциплины.

После ввода и отправки запроса Система выполняет поиск по базе данных.

В случае нахождения дисциплины на странице отображается информация о количестве часов и форме отчетности по ней. В противном случае выводится сообщение об отсутствии данных.

## 4.3. Не функциональные требования

4.3.1. Требования к производительности:

Время отклика пользовательского интерфейса при выполнении стандартных операций (открытие страниц, сохранение форм) не должно превышать 3 секунд при работе в локальной сети.

4.3.2. Требования к надежности:

Система должна обеспечивать целостность данных за счет использования механизмов реляционной СУБД (внешние ключи, ограничения).

Система должна корректно обрабатывать неверный ввод данных пользователем (например, попытка сохранить форму с незаполненными обязательными полями), выводя соответствующие уведомления об ошибках.

4.3.3. Требования к безопасности:

Доступ к функциям управления данными должен быть защищен и предоставляться только аутентифицированным и авторизованным пользователям (через механизм аутентификации Django Admin).

# 5. СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ

5.1. Требования к информационному обеспечению

5.1.1. Логическая структура базы данных должна быть основана на реляционной модели и включать следующие сущности и их атрибуты:

|  |  |  |  |
| --- | --- | --- | --- |
| **Сущность** | **Атрибут** | **Тип данных** | **Описание** |
| **Student** | last\_name | Строка (100) | Фамилия, обязательное поле |
|  | first\_name | Строка (100) | Имя, обязательное поле |
|  | middle\_name | Строка (100) | Отчество, необязательное поле |
|  | admission\_year | Целое число | Год поступления |
|  | form\_of\_study | Строка (20) | Форма обучения (выбор из списка) |
|  | group\_name | Строка (50) | Номер/название группы |
| **Discipline** | name | Строка (200) | Название дисциплины, обязательное поле |
|  | specialty | Строка (200) | Название специальности |
|  | semester | Целое число | Номер семестра |
|  | hours | Целое число | Количество часов |
|  | report\_form | Строка (20) | Форма отчетности (выбор из списка) |
| **Grade** | student | Внешний ключ | Ссылка на запись в таблице Student |
|  | discipline | Внешний ключ | Ссылка на запись в таблице Discipline |
|  | year | Целое число | Год сдачи |
|  | semester | Целое число | Семестр сдачи |
|  | grade | Строка (50) | Оценка (включая "зачтено"/"не зачтено") |

5.1.2. Физически данные должны храниться в файле СУБД SQLite.

## 5.2. Требования к программному обеспечению

5.2.1. Общесистемное программное обеспечение:

Операционная система: Windows 10/11, дистрибутивы Linux (напр., Ubuntu 20.04+), macOS.

Система управления базами данных: SQLite версии 3.

5.2.2. Прикладное программное обеспечение:

Среда исполнения: Python версии 3.9 или выше.

Основной фреймворк: Django версии 4.0 или выше.

## 5.3. Требования к лингвистическому обеспечению

5.3.1. Все элементы пользовательского интерфейса (надписи, метки полей, кнопки, сообщения) должны быть реализованы на русском языке.

5.3.2. Используемая в Системе терминология должна быть унифицирована и соответствовать предметной области.

# 6. СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ

Разработка Системы выполняется поэтапно в соответствии с таблицей ниже.

|  |  |  |  |
| --- | --- | --- | --- |
| **№ Этапа** | **Наименование этапа** | **Содержание работ** | **Результат** |
| 1 | **Проектирование и настройка окружения** | Анализ требований, создание структуры проекта Django, настройка виртуального окружения, инициализация системы контроля версий Git. | Готовая к разработке структура проекта. |
| 2 | **Реализация модели данных** | Описание структуры базы данных в виде моделей Django (models.py), создание и применение миграций. | Создана структура БД в файле db.sqlite3. |
| 3 | **Реализация интерфейса управления данными** | Настройка административной панели Django (admin.py) для управления всеми сущностями. Создание суперпользователя. | Работоспособный интерфейс для CRUD-операций. |
| 4 | **Реализация пользовательских функций** | Создание представлений (views.py), HTML-шаблонов и URL-маршрутов (urls.py) для реализации информационно-аналитических запросов. Кастомизация админ-панели. | Полнофункциональный прототип Системы. |

# 7. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ СИСТЕМЫ

7.1. Виды испытаний: Приемка Системы осуществляется путем проведения приемо-сдаточных испытаний.

7.2. Объем и методы испытаний: Испытания проводятся в форме демонстрации функционала Системы Заказчику. Проверяется соответствие реализованных функций требованиям, изложенным в разделе 4 настоящего ТЗ.

7.3. Программа испытаний: В ходе демонстрации должны быть последовательно выполнены следующие тестовые сценарии:

Аутентификация в административной панели.

Создание, редактирование и удаление записи о студенте.

Создание, редактирование и удаление записи о дисциплине.

Создание и удаление записи об оценке, проверка работы поиска по связанным сущностям.

Проверка работы функции подсчета студентов для каждой из форм обучения.

Проверка работы функции поиска информации о существующей и несуществующей дисциплине.

Проверка работоспособности навигационных ссылок и автоматической переадресации.

7.4. Критерии приемки: Система считается принятой, если все тестовые сценарии выполнены успешно и реализованный функционал соответствует требованиям настоящего ТЗ. Результатом является положительная оценка Заказчика.

# 8. ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

CRUD – Create, Read, Update, Delete (Создание, Чтение, Обновление, Удаление) – базовые операции для работы с данными.

Python – Высокоуровневый язык программирования.

Django – Веб-фреймворк на языке Python.

СУБД – Система управления базами данных.

SQLite – Компактная встраиваемая реляционная СУБД.

URL – Uniform Resource Locator (Единый указатель ресурса).

HTML – HyperText Markup Language (Язык гипертекстовой разметки).