

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных наук

Кафедра информационных технологий управления

*Разработка веб-приложения для переноса расписания занятий
факультета компьютерных наук в сервис «Google Calendar»*

Курсовая работа

09.03.03 Информационные системы и технологии

Программирование и информационные технологии

Допущено к защите в ГЭК

Зав. кафедрой _____ С.Д. Махортов, д. т. н., профессор _____.2023

Обучающийся _____ Д.А. Змаев, 3 курс, д/о

Руководитель _____ П.С. Лысачёв, ст. преподаватель

Воронеж 2023

Введение

В наше время, когда все больше процессов становятся автоматизированными и современные технологии позволяют нам оптимизировать многие аспекты нашей жизни, важно не оставаться в стороне от прогресса. В области образования и учебного процесса также есть много возможностей для автоматизации и оптимизации. Одним из примеров такой автоматизации является использование сервиса «Google Calendar» для отслеживания расписания занятий в университете.

Целью данной курсовой работы является разработка веб-приложения, которое будет автоматически переносить расписание занятий факультета компьютерных наук в сервис «Google Calendar». Такое приложение позволит студентам быстро и легко получать доступ к своему расписанию занятий, а также уведомлять их о предстоящих занятиях.

В рамках работы будет проведен анализ возможностей «Google Calendar» API, а также существующих решений для автоматического переноса расписания занятий в календарь. Будет разработана архитектура приложения, которая будет учитывать специфику расписания занятий факультета компьютерных наук. Далее, будет разработано приложение, которое будет обеспечивать автоматический перенос расписания занятий в сервис «Google Calendar». Наконец, приложение будет протестировано и документировано.

В результате выполнения данной курсовой работы студенты факультета компьютерных наук смогут получать доступ к своему расписанию занятий в календаре «Google Calendar», что значительно облегчит им организацию своего времени и позволит им более эффективно использовать свои учебные ресурсы.

1 Постановка задачи

Цель курсовой работы - Разработка веб-приложения для переноса расписания занятий факультета компьютерных наук в сервис «Google Calendar»

1.1 Цель создания системы

Система разрабатывается в следующих целях: Реализация процесса переноса расписания занятий ФКН в сервис «Google Calendar».

1.2 Требования к создаваемой системе

Разрабатываемое приложение должно иметь следующие возможности:

- Предоставление пользователю возможности выбора расписания в зависимости от его курса, группы и подгруппы
- Реализация аккаунта администратора для выгрузки расписания в случае его изменения
- Предоставление пользователю возможности выбора срока заполнения расписания

1.3 Задачи, решаемые в процессе разработки

- Спроектировать систему с учетом полученной в ходе анализа информации
- Изучить и реализовать взаимодействие с внешним ресурсом «Google Calendar API»
- Разработать пользовательский веб-интерфейс
- Описать процесс разработки и результат

2 Анализ предметной области

2.1 Терминология (гlossарий) предметной области

Google Calendar API — это интерфейс программирования приложений, который позволяет разработчикам создавать приложения, которые могут получать доступ к календарю Google, создавать, изменять и удалять события в календаре, а также выполнять другие операции. Это означает, что при помощи Google Calendar API разработчики могут создавать приложения, которые могут автоматически добавлять события в календарь Google, устанавливать напоминания и многое другое.

SQLite — это компактная, встроенная реляционная база данных, работающая на многих платформах.

Веб-приложение - программа, которая запускается в браузере и использует веб-технологии для выполнения своих функций.

Парсинг — это процесс извлечения данных из структурированного или полуструктурированного источника, такого как веб-страница или файл.

Валидация — это процесс проверки правильности и соответствия данных определенным критериям или правилам. В веб-разработке, валидация часто используется для проверки данных, введенных пользователем в форму на веб-странице, например, при регистрации аккаунта или заполнении заказа. Валидация может проверять формат, длину или тип данных, а также применять другие правила, определенные разработчиком. Цель валидации - обеспечить корректность и целостность данных, предотвратить ошибки ввода и улучшить пользовательский опыт.

2.2 Целевая аудитория

Целевая аудитория для данного веб-приложения - студенты факультета компьютерных наук, которые используют сервис «Google Calendar» для организации своего расписания и планирования учебных занятий.

3 Реализация

3.1 Средства реализации

В качестве средств реализации были использованы:

- Python - высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.
- Flask - фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2.
- SQLite - SQLite - это компактная, встраиваемая реляционная база данных, которая работает на многих платформах, включая Windows, Mac OS X, Linux, Android и iOS. Она является самодостаточной и не требует отдельного сервера баз данных, поэтому ее можно легко интегрировать в различные приложения и использовать для хранения и управления структурированными данными, такими как текст, числа, изображения и т.д.
- Google Calendar Api - это интерфейс программирования приложений, который позволяет разработчикам создавать приложения, которые могут получать доступ к календарю Google, создавать, изменять и удалять события в календаре, а также выполнять другие операции. Это означает, что при помощи «Google Calendar API» разработчики могут создавать приложения, которые могут автоматически добавлять события в календарь Google, устанавливать напоминания и многое другое.

3.2 Разработка архитектуры

- Анализ требований и выбор технологий: на этом этапе будет проведен анализ требований к приложению, определены функциональные и нефункциональные требования, а также выбраны технологии, которые будут использоваться для разработки приложения.
- Проектирование базы данных: на этом этапе будет спроектирована база данных для приложения, определены таблицы, поля, а также выбрана СУБД для хранения данных. В данном случае, можно использовать SQLite, которая является легковесной и встраиваемой СУБД.
- Разработка серверной части приложения: на этом этапе будет разработана серверная часть приложения, которая будет отвечать за обработку запросов от клиентской части, взаимодействие с базой данных и работу с API Google Calendar. Для разработки серверной части можно использовать фреймворк Flask на языке Python.
- Разработка клиентской части приложения: на этом этапе будет разработана клиентская часть приложения, которая будет отвечать за отображение данных пользователю и взаимодействие с серверной частью приложения. В данном случае в качестве клиентской части используется веб-страница, формируемая статическим шаблонизатором Jinja.

3.3 Диаграмма классов

3.4 Реализация логики

Логику приложения можно разделить на несколько этапов:

3.4.1 Получение данных

На этом этапе будут получены данные, которые будут использоваться для разбора ячеек файла с расписанием занятий расписания. В данном случае, данные могут быть получены из Excel-файла формата xls.

3.4.2 Создание класса Parser

На этом этапе будет создан класс Parser, принимающий в конструктор путь к файлу с расписанием. Класс отвечает за разбор ячеек файла с расписанием занятий. В данном случае, класс Parser будет имеет два основных метода: `parse_nominator_schedule()` и `parse_denominator_schedule()`, которые будут отвечают за числитель и знаменатель соответственно.

3.4.3 Создание событий в сервисе «Google Calendar»

На этом этапе создаются события в «Google Calendar» на основе данных о занятиях, полученных на предыдущем шаге. Для создания событий, используйте Google Calendar API и библиотека `google-api-python-client`.

3.4.4 Обработка ошибок

На этом этапе будут обработаны возможные ошибки, которые могут возникнуть в процессе разбора ячеек файла расписания или создания событий в сервисе «Google Calendar». В случае возникновения ошибок, пользователю будет выведено соответствующее сообщение.

3.5 Реализация интерфейса

При открытии веб-приложения пользователь попадает на главный экран, на котором содержится информация о приложении.

3.5.1 Интерфейс пользователя

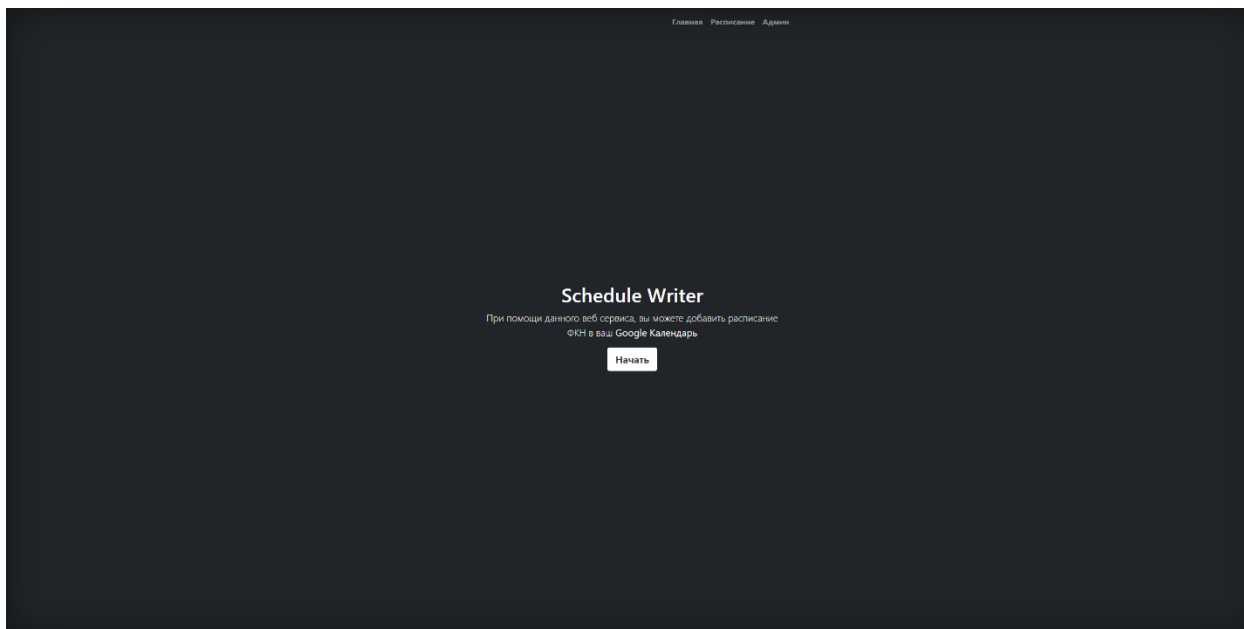


Рисунок 1 - Главная страница веб-приложения

При нажатии на кнопку «Начать» или выборе раздела «Расписание» в меню навигации пользователь попадает на страницу «Заполнение расписания»

Рисунок 2 - Страница «Заполнение расписания»

При корректном вводе всех полей пользователем, его перенаправит на страницу авторизации в Google аккаунт.

Если поля заполнены некорректно, пользователю будет отображено сообщение с просьбой заполнить незаполненное поле.

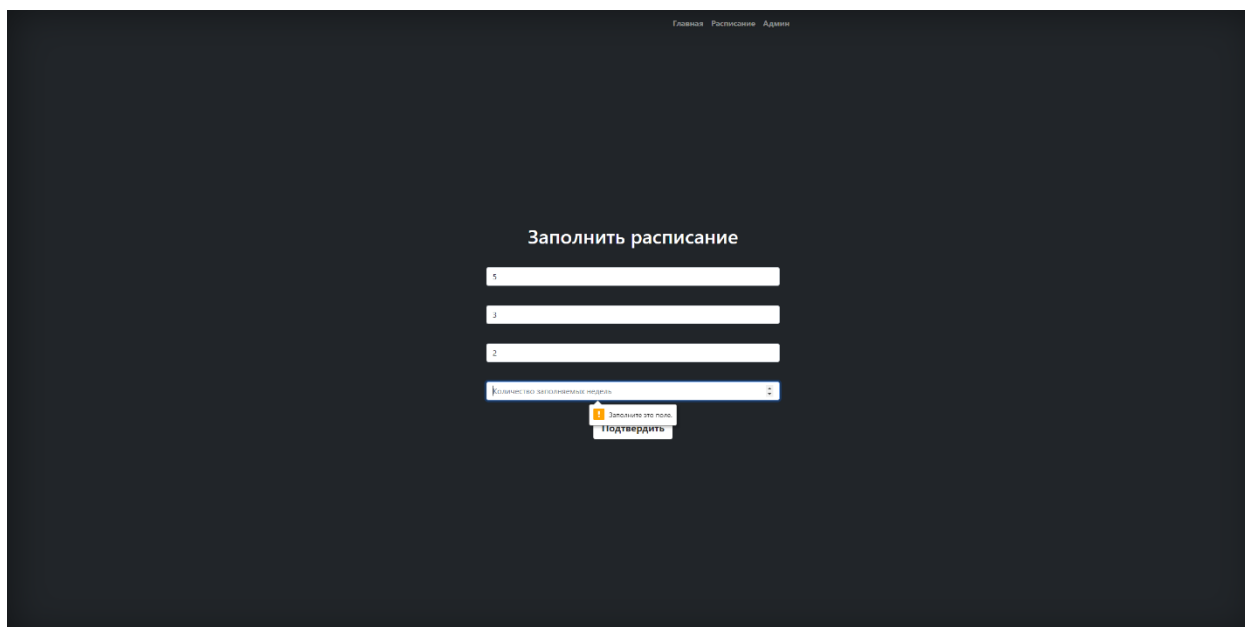


Рисунок 3 - Ошибка валидации

После завершения загрузки пользователь может открыть свой Google Calendar, привязанный к аккаунту, на который был выполнен вход и увидеть результат.

При вводе 3 курса, 5 группы и 2 подгруппы получился такой результат:

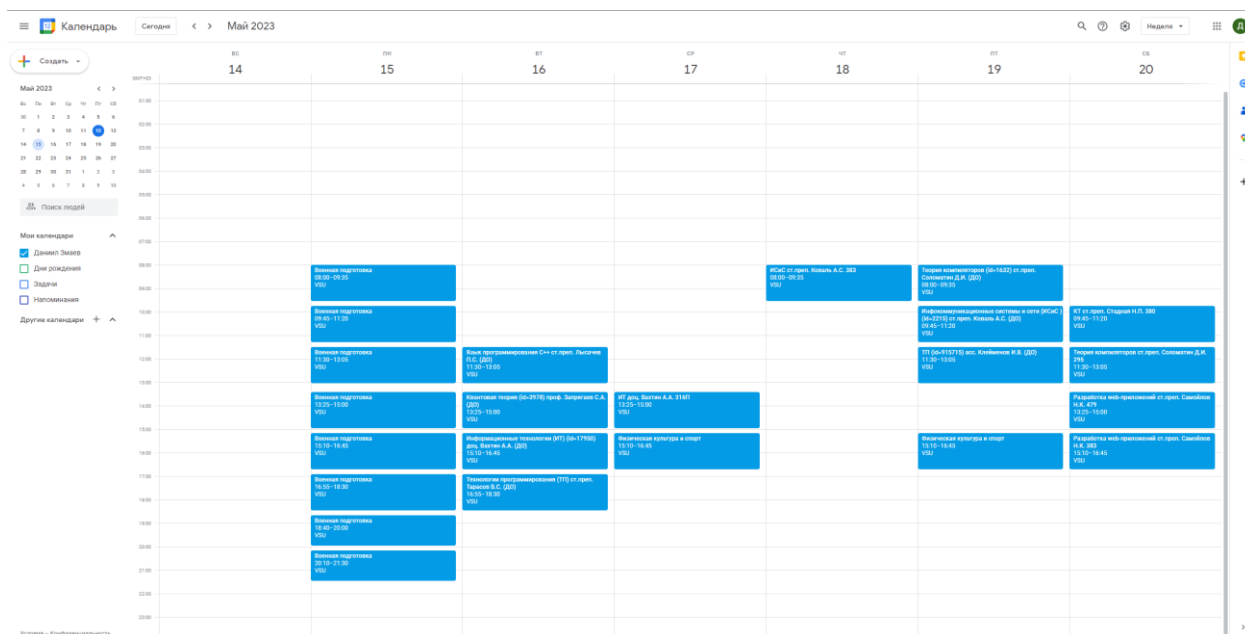


Рисунок 4 - Результат выполнения программы

3.5.2 Интерфейс администратора

При выборе пункта «Админ» на навигационной панели, открывается следующая страница:

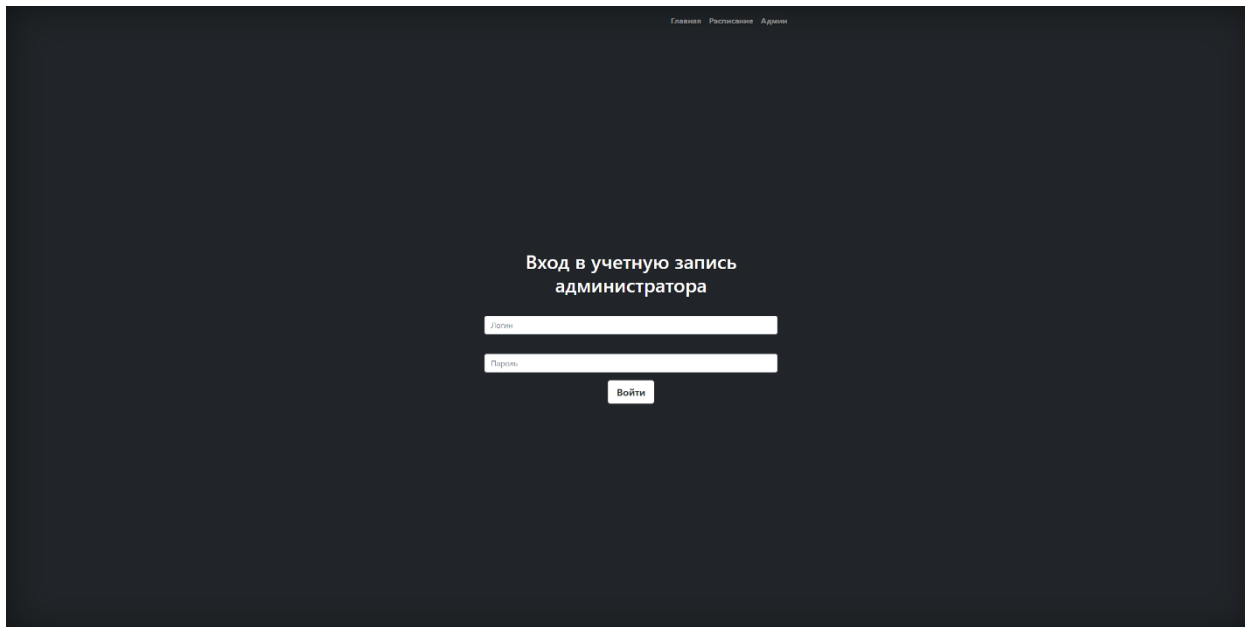


Рисунок 5 - Страница входа в учетную запись администратора

При вводе некорректных данных появится сообщение об ошибке:

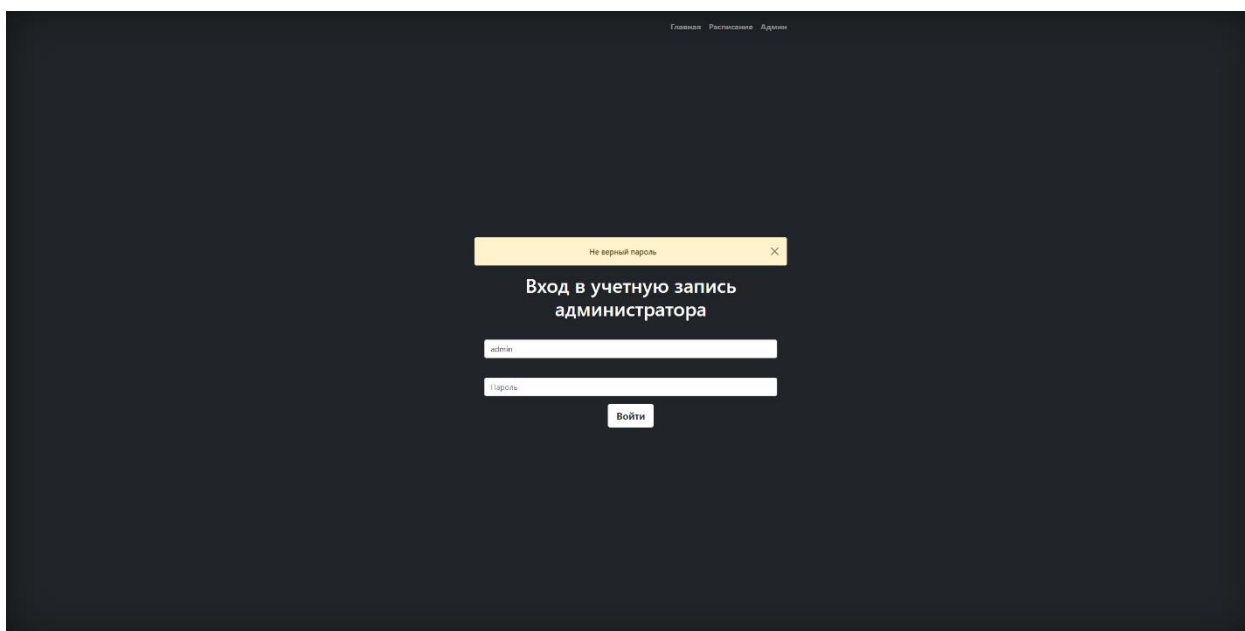


Рисунок 6 - Ошибка при валидации администратора

После того как валидация будет выполнена успешно откроется страница с выбором файла. Тут администратор может поменять расписание на новое, если, например, оно было изменено.

На этой странице, также присутствует валидация: может быть добавлен файл только формата xls, иначе появится сообщение об ошибке.

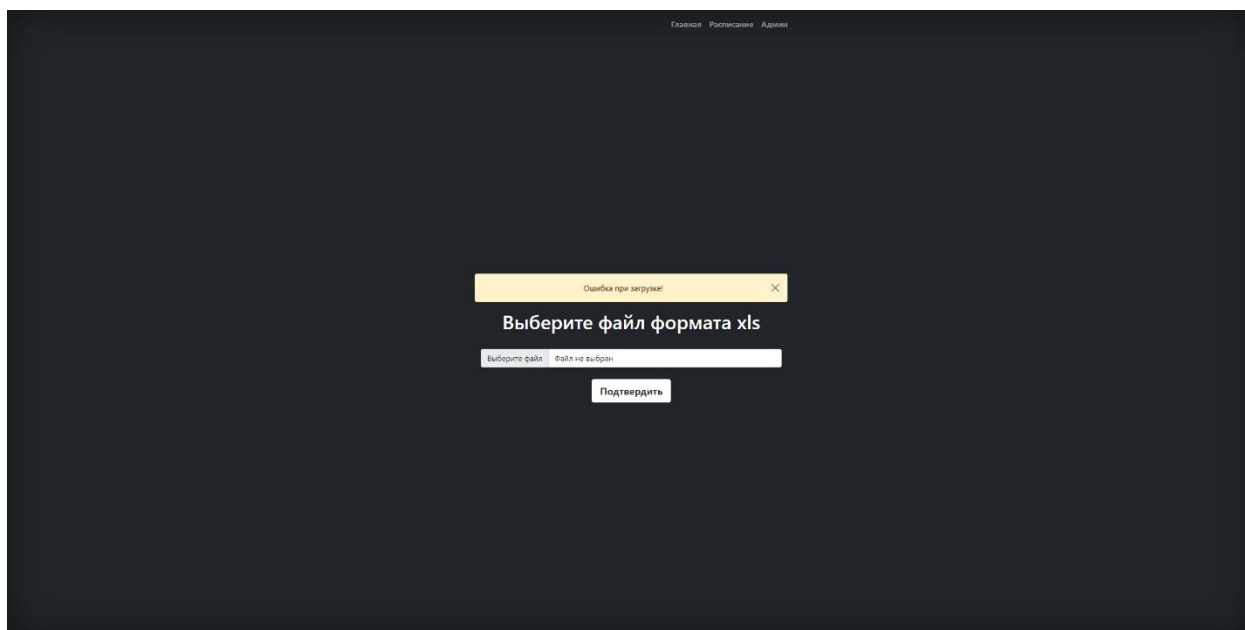


Рисунок 7 - Ошибка загрузки файла

Если все прошло успешно, то предыдущее расписание будет заменено на новое.

Стоит уточнить, что изменить расписание может только администратор, после успешного входа в систему, обычный пользователь не имеет доступа к данной странице, т.к. используется система токенов авторизации.

Заключение

Как результат выполнения данной курсовой работы, было разработано веб-приложение, которое позволяет автоматически переносить расписание занятий факультета компьютерных наук в сервис «Google Calendar». Были выполнены все поставленные перед проектом функциональные и нефункциональные требования, а именно: возможность автоматического добавления расписания занятий в Google Calendar на основе данных из источников, таких как Excel-файл, возможность настройки периода, на который будет добавлено расписание, возможность синхронизации с «Google Calendar» и уведомления студентов о предстоящих занятиях.

Проект был разработан с использованием Google Calendar API и SQLite, что позволило создать компактное, эффективное и удобное приложение. Веб-приложение может быть использовано студентами факультета компьютерных наук для доступа к их расписанию занятий.

В результате выполнения курсовой работы были получены знания и навыки, необходимые для разработки веб-приложений, а также опыт работы с «Google Calendar API» и «SQLite». Разработанное приложение может быть использовано в учебных заведениях для упрощения и оптимизации процесса обучения.