

Задание: «Интеллектуальный журнал учебных проектов»

Цель: Разработать кроссплатформенное desktop-приложение для ведения и анализа учебных проектов студентов с возможностью структурированного хранения описаний, автоматического сбора метрик активности, визуализации данных и генерации профессиональных отчётов в форматах Excel и Word.

Тема: Разработка приложения «Интеллектуальный журнал учебных проектов» с использованием Python (tkinter), PostgreSQL, pgAdmin, Markdown, openpyxl и python-docx.

1. Постановка задачи

Необходимо создать приложение, которое позволяет студентам фиксировать информацию о своих учебных проектах, включая описание, этапы выполнения, используемые технологии и достигнутые результаты. Все текстовые описания проектов должны храниться в виде Markdown-файлов на диске, а метаданные — в реляционной базе данных PostgreSQL. Приложение должно анализировать активность пользователя (создание, редактирование, просмотр проектов) и на основе этих данных формировать аналитические отчёты:

- в формате Excel с таблицами и диаграммами;
- в формате Word с оформлением в стиле профессионального документа.

Управление базой данных осуществляется через pgAdmin.

2. Функциональные требования

Часть А: Работа с PostgreSQL (модель данных)

1. Создать базу данных student_projects с помощью pgAdmin.

2. Создать следующие таблицы:

Таблица projects:

- id (SERIAL PRIMARY KEY)
- title (VARCHAR) — название проекта
- file_path (VARCHAR UNIQUE) — абсолютный путь к .md файлу
- created_at (TIMESTAMP) — дата создания записи
- updated_at (TIMESTAMP) — дата последнего изменения
- discipline (VARCHAR) — учебная дисциплина (например, «Программирование», «Электроника»)
- status (VARCHAR) — статус проекта: «В работе», «Завершён», «Приостановлен»

Таблица technologies:

- id (SERIAL PRIMARY KEY)
- name (VARCHAR UNIQUE) — например, «Python», «Tkinter», «PostgreSQL»

Таблица project_technologies (связь многие-ко-многим):

- project_id (INT, FOREIGN KEY → projects.id)
- technology_id (INT, FOREIGN KEY → technologies.id)

Таблица activity_log:

- id (SERIAL PRIMARY KEY)
- project_id (INT, FOREIGN KEY → projects.id)
- event_type (VARCHAR) — значения: CREATE, UPDATE, VIEW
- event_time (TIMESTAMP DEFAULT NOW())

3. При любом действии с проектом (создание, редактирование, просмотр) в таблицу activity_log должна добавляться соответствующая запись.

Часть В: Работа с Markdown и файловой системой

1. Все описания проектов хранятся в папке projects_md/ в формате .md.

2. При создании нового проекта:

- В БД добавляется запись в таблицу projects
- На диске создаётся пустой .md файл по пути projects_md/{title}.md
- В activity_log фиксируется событие CREATE

3. При сохранении изменений:

- Содержимое перезаписывается в тот же .md файл
- Обновляется поле updated_at в таблице projects
- В activity_log добавляется событие UPDATE

4. Реализовать кнопку «Открыть описание», которая открывает выбранный .md файл во внешнем редакторе или браузере.

Часть С: Графический интерфейс (Tkinter)

Главное окно приложения должно включать:

1. Панель управления:

- Поля ввода: «Название проекта», «Дисциплина», «Статус»
- Кнопки: «Создать», «Сохранить», «Удалить», «Открыть описание», «Экспорт в Excel», «Экспорт в Word»

2. Список проектов (Treeview):

- Отображает колонки: Название, Дисциплина, Статус, Дата создания, Дата обновления
- Поддержка сортировки по любой колонке

3. Область редактирования:

- Text-виджет для редактирования Markdown-описания проекта
- Подсказки по синтаксису: ****жирный****, ***курсив***, # Заголовок, - список

4. Панель технологий:

- Поле ввода для новой технологии и кнопка «Добавить технологию»
- Отображение текущих технологий выбранного проекта (например, через Label с переносом строк)

5. Вкладка «Аналитика и отчётность»:

- Кнопка «Сформировать отчёт»
- После нажатия собираются данные:
 - Количество проектов по дисциплинам и статусам
 - Количество действий (CREATE/UPDATE) за последние 7 и 30 дней
 - Топ-5 самых часто используемых технологий
- Эти данные экспортируются в Excel-файл reports/projects_report.xlsx с двумя листами:
 - «Статистика» — таблица с собранными метриками
 - «Графики» — автоматически построенные столбчатые диаграммы (например, распределение проектов по статусам)
- Также генерируется Word-документ reports/projects_report.docx, содержащий:
 - Титульный лист с названием отчёта и датой формирования
 - Сводную таблицу ключевых показателей
 - Встроенное изображение графика (сохранённого как PNG)
 - Список последних 5 проектов с краткой информацией (название, дисциплина, статус)

Для реализации использовать библиотеки:

- openpyxl — для работы с Excel и вставки графиков
- python-docx — для генерации Word-документов
- matplotlib — для создания графиков (сохранять как PNG и вставлять в Word и Excel)

3. Основные сценарии работы

1. Создание проекта

Пользователь вводит название, дисциплину и статус, нажимает «Создать». Создаётся запись в БД, файл на диске и запись в activity_log.

2. Редактирование

Выбор проекта → загрузка содержимого из .md → редактирование → нажатие «Сохранить» → обновление файла, поля updated_at и лога.

3. Добавление технологий

Возможность привязать к проекту одну или несколько технологий без изменения основного текста.

4. Формирование отчётов

Нажатие «Сформировать отчёт» → сбор данных → экспорт в Excel с графиками → генерация Word-документа с профессиональным оформлением.

4. Требования к оформлению кода и отчёта

- Код должен быть структурирован по модулям (БД, GUI, экспорт).
- Все SQL-запросы должны использовать параметризованные выражения.
- Приложение должно корректно обрабатывать ошибки (например, отсутствие файла, недоступность БД).
- Итоговый отчёт в Word должен соответствовать стандартам делового документа (шрифт Times New Roman, размер 12, межстрочный интервал 1.5).

Примечание: Все пути к файлам должны быть относительными и корректно работать на Windows и Linux.