

Задание: «Электронный портфолио студента-исследователя»

Цель:

Разработать кроссплатформенное desktop-приложение для ведения персонального портфолио студента-исследователя, включающего научные публикации, участие в конференциях, гранты, преподавательскую деятельность и достижения. Приложение должно обеспечивать структурированное хранение данных, автоматический учёт активности, визуализацию карьерного роста и генерацию профессиональных отчётов в форматах Excel и Word.

Тема:

Разработка приложения «Электронный портфолио студента-исследователя» с использованием Python (tkinter), PostgreSQL, pgAdmin, Markdown, openpyxl и python-docx.

1. Постановка задачи

Необходимо создать приложение, позволяющее студенту фиксировать и систематизировать свою научно-исследовательскую и академическую деятельность. Каждая запись (например, публикация или участие в конференции) описывается в виде файла формата Markdown на диске, а метаданные хранятся в реляционной базе данных PostgreSQL. Приложение должно отслеживать действия пользователя (добавление, редактирование, просмотр записей) и на основе этих данных формировать аналитические отчёты:

- в формате Excel — с таблицами и диаграммами (например, динамика публикаций по годам);
- в формате Word — оформленные как официальное портфолио или резюме исследователя.

Управление базой данных осуществляется через pgAdmin.

2. Функциональные требования

Часть А: Работа с PostgreSQL (модель данных)

1. Создать базу данных research_portfolio с помощью pgAdmin.

2. Создать следующие таблицы:

Таблица entries (записи в портфолио):

id — целое число, автоинкремент, первичный ключ
title — строка, название записи (например, «Доклад на конференции ИТ-2025»)
entry_type — строка, тип записи: «Публикация», «Конференция», «Грант», «Преподавание», «Достижение»
file_path — строка, уникальный путь к файлу описания в формате .md
year — целое число, год события
created_at — дата и время создания записи
updated_at — дата и время последнего изменения

Таблица coauthors (соавторы):

id — целое число, автоинкремент, первичный ключ
name — строка, ФИО соавтора

Таблица entry_coauthors (связь многие-ко-многим):

entry_id — целое число, внешний ключ к entries.id
coauthor_id — целое число, внешний ключ к coauthors.id

Таблица activity_log:

id — целое число, автоинкремент, первичный ключ
entry_id — целое число, внешний ключ к entries.id
event_type — строка, возможные значения: CREATE, UPDATE, VIEW
event_time — дата и время события, по умолчанию текущее время

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

Часть В: Работа с Markdown и файловой системой

1. Все описания записей хранятся в папке portfolio_md в формате .md.

2. При создании новой записи:

- В базе данных добавляется запись в таблицу entries
- На диске создаётся пустой файл portfolio_md/название.md (название очищено от недопустимых символов)
- В таблицу activity_log добавляется событие CREATE

3. При сохранении изменений:

- Содержимое перезаписывается в тот же .md файл
- Обновляется поле updated_at в таблице entries
- 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», «Тип записи», «Год»
- Кнопки: «Создать», «Сохранить», «Удалить», «Открыть описание», «Экспорт в Excel», «Экспорт в Word»

2. Список записей (Treeview):

- Отображает колонки: Название, Тип, Год, Дата создания
- Поддерживает сортировку по любой колонке

3. Область редактирования:

- Text-виджет для редактирования Markdown-описания
- Подсказки по синтаксису: цитаты (> текст), код (три обратные кавычки), ссылки ([текст](url))

4. Панель соавторов:

- Поле ввода для имени соавтора и кнопка «Добавить соавтора»
- Отображение текущих соавторов выбранной записи (например, в виде текстовой метки с переносом строк)

5. Вкладка «Аналитика и отчётность»:

- Кнопка «Сформировать отчёт»
- После нажатия собираются данные:
 - Распределение записей по типам и годам
 - Количество уникальных соавторов
 - Активность за последние 12 месяцев (по месяцам)
- Эти данные экспортируются в Excel-файл reports/portfolio_report.xlsx с двумя листами:
 - «Статистика» — таблица с собранными метриками
 - «Графики» — автоматически построенные столбчатые диаграммы (например, количество публикаций по годам)
- Также генерируется Word-документ reports/portfolio_report.docx, содержащий:
 - Титульный лист с названием отчёта и датой формирования
 - Сводную таблицу ключевых показателей
 - Встроенное изображение графика (сохранённого как PNG)
 - Список последних 5 записей с краткой информацией (название, тип, год)

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 записи

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

2. Редактирование

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

3. Добавление соавторов

Возможность привязать к записи одного или нескольких соавторов 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 операционных системах Windows и Linux.