Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Уфимский университет науки и технологий»

Факультет информатики и робототехники

Кафедра ВМК

Отчёт по лабораторной работе № 4

По Дисциплине

«Проектирование и конструирование программного обеспечение»

На тему

«Проектирование БД. Проектирование пользовательского интерфейса»

Выполнили:

Студент группы ПРО-235Б Тюшевских Н.А., Ахмадеев Р.Р.

Проверил:

Преподаватель Ризванов Д.А.

Уфа – 2023

# Лабораторная работа №4

Цель работы:

• Ознакомление с основными методами проектирования базы данных (БД).

• Ознакомление с методами проектирования пользовательского интерфейса.

Задание:

Проектирование БД

1. Изучить дополнительный материал по теории проектирования БД (см. раздел «дополнительная информация») спроектировать БД приложения для хранения обработанных данных.

a. БД должна содержать таблицу с информацией об обработанных файлах (минимальный набор колонок: название файла, дата обработки).

b. Таблица обработанных файлов должна ограничивать основную таблицу с данными по внешнему ключу (FOREIGN KEY).

2. Написать SQL-скрипт для создания структуры спроектированной БД (см. пример SQL-скрипт в архиве pikpo4\_python.zip).

3. Изучить код примера из архива pikpo4\_python.zip. На основе данного примера реализовать необходимые CRUD (Create, Read, Update, Delete) операции для работы с БД (см. ссылку на материал по основам SQL). Загрузить код приложения на GitHub.

4. Проверить выполнение CRUD-операций на тестовой БД (только SQLite).

Проектирование пользовательского интерфейса

Во время выполнения лабораторной работы необходимо описать ожидаемое поведение разрабатываемой системы с точки зрения внешнего по отношению к ней пользователя, то есть осуществить "конструирование" внешних взаимодействий будущей ИС с пользователем без конкретизации его внутреннего устройства.

1. Определить структуру проектируемого пользовательского интерфейса (визуальное оформление, отвечающее за представление информации пользователю; функциональные возможности системы, включающие набор возможностей для эффективного выполнения профессиональной деятельности; техники взаимодействия пользователя с системой + дополнительные функциональные возможности системы) с учетом задачи (лекции, рекомендации Приложение 1, дополнительные функциональные возможности системы (пример) в Приложении 3).

2. Определить стили пользовательского интерфейса (графический (GUI, web-интерфейс (WUI), объектно-ориентированный интерфейс) с учетом задачи (лекции, рекомендации Приложение 1).

3. Определиться с размещением элементов пользовательского интерфейса (кнопки, иконки, выпадающие списки, поля для записи текста и пр.) (Лекции, рекомендации Приложение 1).

4. Написать Требования к интерфейсу пользователя (Пример в приложении 2).

5. Разработать взаимодействие разрабатываемой программы с пользователем: сценарий (можно диаграммой последовательности, диаграммой взаимодействия), экранные формы, набор подсказок (перечисление), и пр.

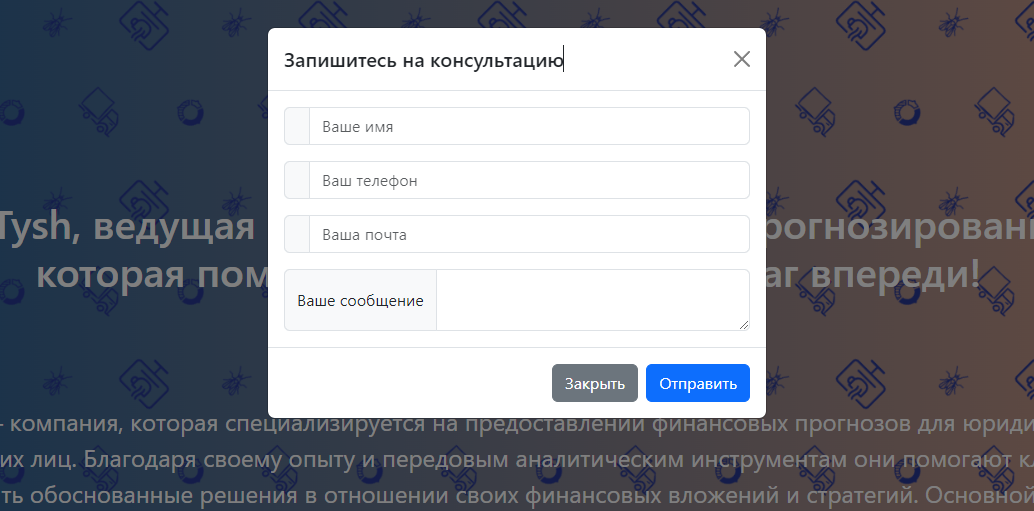
6. Ознакомиться с методическим материалом по базовой верстке веб-страниц.

7. На основе п.1-5 разработать (сверстать) основные html-страницы для вашего приложения, используя CSS-стили и HTML5-модель верстки (см. пример реализации HTML5-страниц в папке www). Загрузить код htmlстраниц на GitHub.

**Ход работы:**

**1)Взаимодействие разрабатываемой программы с пользователем**





Взаимодействие с пользователем осуществляется через сайт, посредством нажатия на кнопки «Записаться» и заполнение формы «Запишитесь на консультацию». Функционал сайта включает возможность при нажатии на кнопку, отображать форму для записи.

**2)Требования к пользовательскому интерфейсу:**

1. Простой переход от просмотра к регистрации
2. Прямой доступ к примеру продукта
3. Доступ ко всем функциям без дополнительных переходов

**3)ER- диаграмма БД**

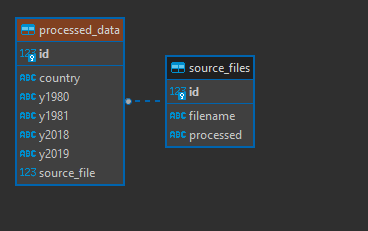
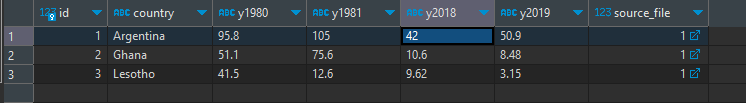
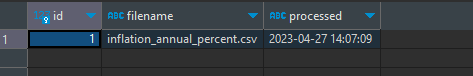


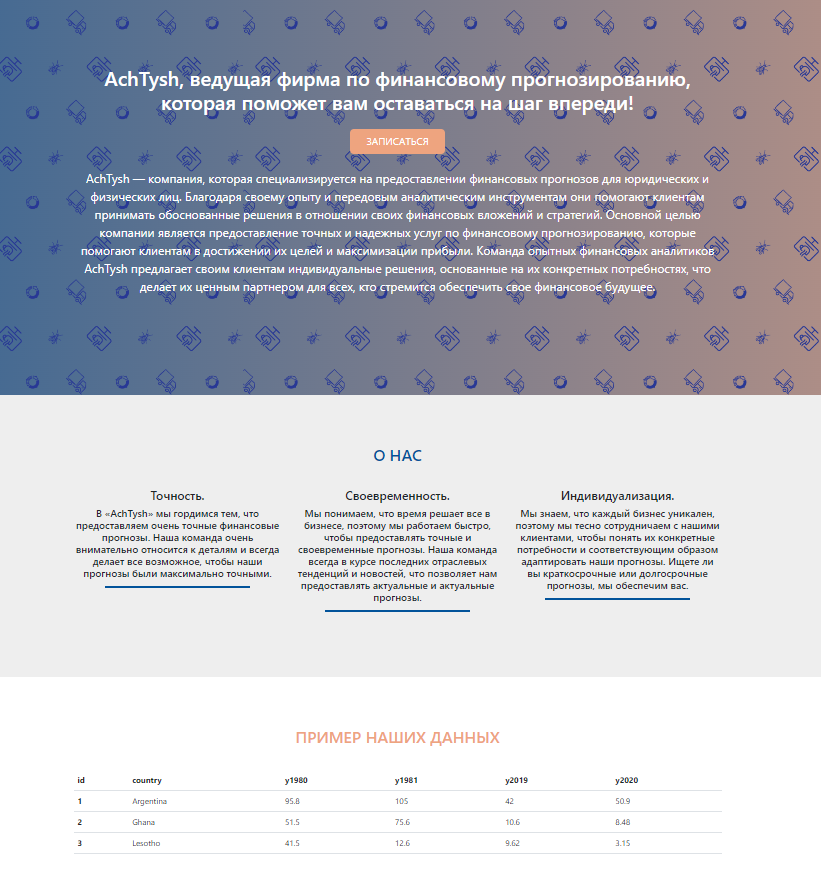
Рисунок 0.1 ER-диаграмма

Скриншоты части содержимого основной таблицы данных и связанных таблиц.





Скриншот основной html – страницы.



Тэг <head> используется для хранения метаданных документа, таких как заголовок страницы, стили, ссылки на скрипты и другую информацию, которую браузер может использовать для правильного отображения страницы.

Тэг <tr> определяет строку (ряд) таблицы. Этот тэг должен использоваться внутри тэга <table>, а строка может содержать ячейки (другие тэги по типу <th>, <td>).

Тэг <th> определяет заголовок таблицы и является ячейкой таблицы на пересечении строки заголовка и столбца. В отличие от тэга <td>, содержимое тэга <th> обычно выравнивается по центру и выделяется жирным шрифтом.

Тэг <td> определяет ячейку таблицы внутри строки. Он используется для отображения информации в таблице. Как правило, содержимое тэга <td> выравнивается по левому краю.

Тэг <body> определяет тело документа, которое отображается в окне браузера. Он содержит всю видимую информацию на странице, такую как текст, изображения, ссылки и т.д.

Листинг программы

sql- script

create table source\_files (

id integer PRIMARY KEY autoincrement,

filename varchar(255) NOT NULL,

processed datetime

);

create table processed\_data (

id integer PRIMARY KEY autoincrement,

country varchar(255),

y1980 varchar(255),

y1981 varchar(255),

y2018 varchar(255),

y2019 varchar(255),

source\_file integer NOT NULL,

CONSTRAINT fk\_source\_files

FOREIGN KEY (source\_file)

REFERENCES source\_files(id)

ON DELETE CASCADE

);

python

po4//repository/sql\_api.py

def insert\_rows\_into\_processed\_data(connector: StoreConnector, dataframe: DataFrame):

    """ Вставка строк из DataFrame в БД с привязкой данных к последнему обработанному файлу (по дате) """

    rows = dataframe.to\_dict('records')

    files\_list = select\_all\_from\_source\_files(connector)    # получаем список обработанных файлов

    # т.к. строка БД после выполнения SELECT возвращается в виде объекта tuple, например:

    # row = (1, 'seeds\_dataset.csv', '2022-11-15 22:03:16'),

    # то значение соответствующей колонки можно получить по индексу, например id = row[0]

    last\_file\_id = files\_list[0][0]  # получаем индекс последней записи из таблицы с файлами

    if len(files\_list) > 0:

        for row in rows:

            connector.execute(f"""

                INSERT INTO processed\_data (country, y1980, y1981, y2018, y2019, source\_file)

                VALUES ('{row['country']}', '{row['y1980']}', '{row['y1981']}', '{row['y2018']}', '{row['y2019']}', '{last\_file\_id}')

            """)

        print('Data was inserted successfully')

    else:

        print('File records not found. Data inserting was canceled.')

# Выводы к работе.

В ходе выполнения лабораторной работы мы ознакомились с основными методами проектирования БД и методами проектирования пользовательского интерфейса.