МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационные системы и технологии

Специальность 1–40 01 01 Программное обеспечение информационных технологий

Специализация 1–40 01 01 Программное обеспечение информационных технологий (программирование интернет – изданий)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«Реализация базы данных опросника с использованием файрвола, прослушивание конкретных адресов, безопасность на уровне строк, аудит»

Выполнил студент Дащинский Максим Леонидович

(Ф.И.О.)

Руководитель проекта ст. преп, Блинова Е.А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой к.т.н., доц. Смелов В.В.

(учен. степень, звание, должность, Ф.И.О., подпись)

Консультант:  ст. преп, Блинова Е.А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Нормоконтролер: ст. преп, Блинова Е.А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой

Минск 2022

Реферат

Пояснительная записка курсового проекта содержит 30 страниц пояснительной записки, 9 иллюстраций, 5 источников литературы, 6 приложений.

PYTHON, POSTGRESQL 14, TKINTER, FIREWALL, ROW-LEVEL SECURITY, AUDIT, XML IMPORT, XML EXPORT.

Основная цель курсового проекта: проектирование базы данных для опросника с использованием файрвола, прослушивание конкретных адресов, безопасность на уровне строк, аудит.

Первая глава описывает похожие сервисы, а также их недостатки, которые необходимо устранить в приложении.

Вторая глава описывает процесс разработки и настройки базы данных, с которой будет происходить взаимодействие.

В третьей главе приведены тестирование и способы оптимизации запросов.

Четвёртая глава описывает изученные и использованные в процессе разработки технологии.

В пятой главе приведено руководство пользователя по взаимодействию с приложением для каждой из ролей.

В заключении приведены результаты проделанной работы.

Abstract

The course project explanatory work consists of 30 pages of explanatory note, 9 illustrations, 5 literature sources, 6 appendixes.

PYTHON, POSTGRESQL 14, TKINTER, FIREWALL, ROW-LEVEL SECURITY, AUDIT, XML IMPORT, XML EXPORT.

The main goal of the course project: designing a database for a questionnaire using a firewall, listening to specific addresses, row-level security, auditing.

The first chapter describes similar services and their shortcomings that need to be removed in the application.

The second chapter describes the process of developing and configuring the database with which the interaction will take place.

The third chapter provides testing and ways to optimize queries.

The fourth chapter describes the technologies, that studied and used in the development process.

The fifth chapter provides a user guide for interacting with the application for each of the roles.

In conclusion, the results of the work are presented.

**Содержание**

[Введение 5](#_Toc122044409)

[1 Аналитический обзор литературы 6](#_Toc122044410)

[1.1 Аналитический обзор источников 6](#_Toc122044411)

[1.2 Обзор аналогов 6](#_Toc122044412)

[2 Разработка программной системы 8](#_Toc122044413)

[2.1 Таблицы базы данных 8](#_Toc122044414)

[2.2 Роли и пользователи базы данных 9](#_Toc122044415)

[2.3 Процедуры и функции для поставленных задач 10](#_Toc122044416)

[2.3.1 Схема для пользователя 10](#_Toc122044417)

[2.3.2 Схема для администратора 11](#_Toc122044418)

[2.3.3 Схема для программиста 14](#_Toc122044419)

[2.4 Триггеры базы данных 14](#_Toc122044420)

[3 Тестирование и улучшение производительности 15](#_Toc122044421)

[4 Описание технологии 16](#_Toc122044422)

[4.1 Настройка файрвола 16](#_Toc122044423)

[4.2 Прослушивание конкретных адресов 16](#_Toc122044424)

[4.3 Безопасность на уровне строк 16](#_Toc122044425)

[4.4 Аудит 17](#_Toc122044426)

[5 Руководство пользователя 18](#_Toc122044427)

[Заключение 21](#_Toc122044428)

[Список литературы 22](#_Toc122044429)

[Приложение А 23](#_Toc122044430)

[Приложение Б 24](#_Toc122044431)

[Приложение В 25](#_Toc122044432)

[Приложение Г 26](#_Toc122044433)

[Приложение Д 29](#_Toc122044434)

[Приложение Е 30](#_Toc122044435)

Введение

Задача данной работы является разработка приложения с реляционной базой данных, предназначенная для пользования социологов и маркетологов, которая позволяет им создавать опросы и получать результаты опроса. База данных обязана быть составлена с возможностью пользования пользователями и администраторов с разным уровнем доступа к встроенным функциям.

База данных – организованная структура, предназначенная для хранения информации, систематизированная таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины. Реляционная база данных означает, что она основана на реляционной модели данных и между данными ними имеется сильная связь. В качестве СУБД для базы данных выбрана PostgreSQL из-за высокой надежности и производительности.

Также необходимо разработать приложение, для демонстрации работы базы данных. Приложение написано на языке программирования Python с кроссплатформенной библиотеки Tkinter для разработки графического интерфейса. Tkinter – это пакет для Python, предназначенный для работы с библиотекой Tk. Библиотека Tk содержит компоненты графического интерфейса пользователя. Psycopg – это библиотека для PostgreSQL в языке программирования Python.

Для защиты данных пользователей приложение в проекте будет использоваться хеширование паролей от аккаунта перед записью в базу данных.

Требования к приложению:

* хранение информации;
* реализация трех ролей: пользователя, администратора и программиста;
* анализ результатов опроса;
* редактирование учетных данных пользователем;
* взаимодействие с базой данных с помощью хранимых процедур;
* защита базы данных от несанкционированного доступа.

Содержание данной пояснительной записки отражает этапы выполнения курсового проекта.

1 Аналитический обзор литературы

1.1 Аналитический обзор источников

В ходе подготовки пояснительной записки была изучена специальная техническая, учебно-методическая и справочная литература, статьи и материалы, опубликованные в сети интернет.

Основная информация о работе с кроссплатформенной библиотекой Tkinter была получена из официальной документации и сайта metanit.com в разделе руководство по Tkinter.

Принципы работы с SQL были получены из статьи «Подключение к базе данных». В статье было рассмотрено подключение необходимых библиотек, работа с SQL.

Обеспечение защиты базы данных PostgreSQL была получена из официальной документации и из сайта habr.com при помощи статьи «Обеспечение безопасности базы данных PostgreSQL».

1.2 Обзор аналогов

Опросы – очень популярный и удобный способ получения обратной связи. Позволяют получать отзывы от пользователей и анализировать насколько то иное действие было верным.

Для создания опросника в виде программного продукта проанализируем существующие аналоги.

На результате анализов существующих аналогов сформируем требования к проектируемому программному средству. В качестве исследуемых аналогов выбраны программные продукты, которые создают онлайн опросы в сети Интернет.

Один из популярных сервисов по составлению опросов является сервис survio, основная страница которого представлена на рисунке 1.1.

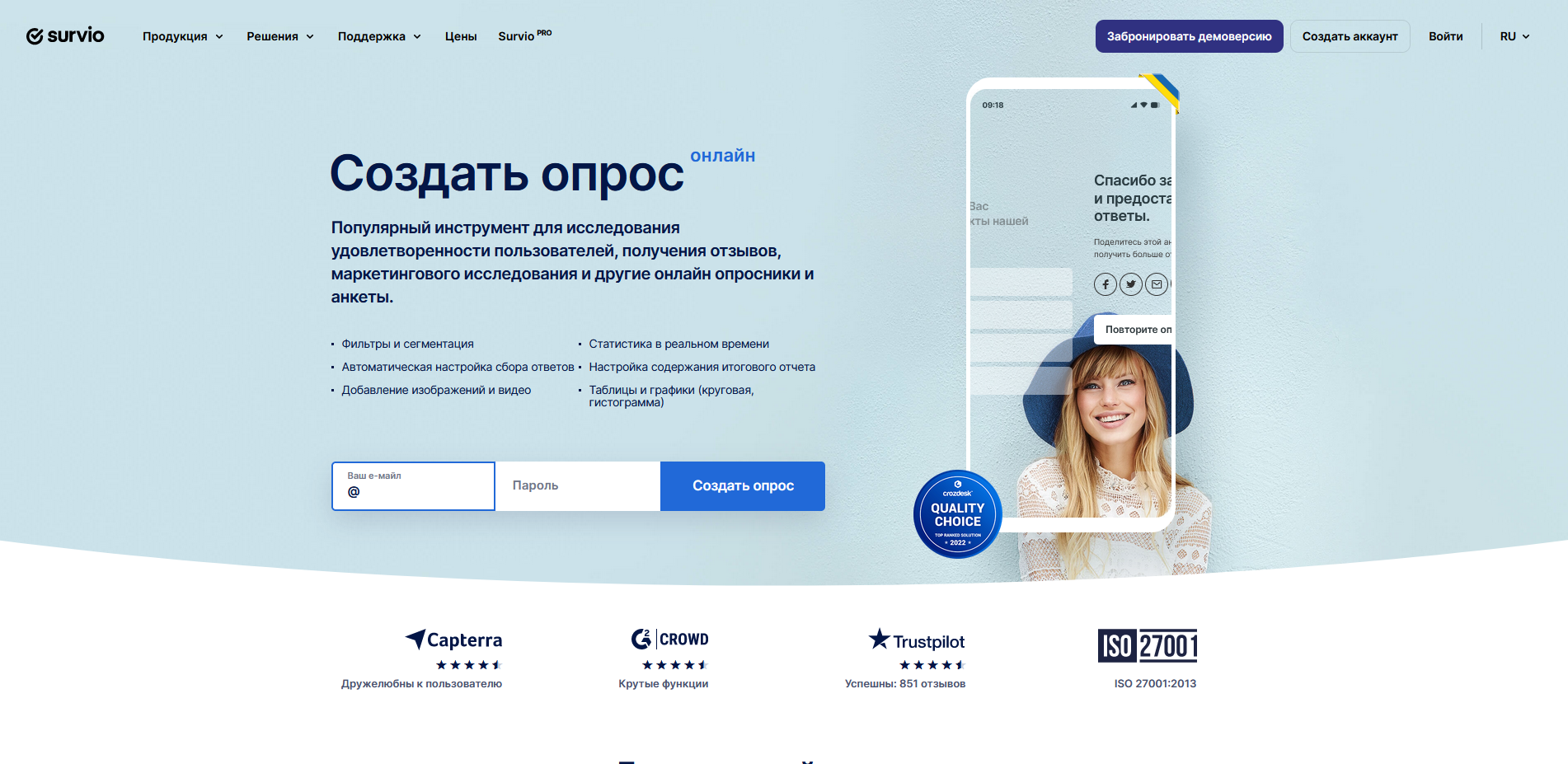


Рисунок 1.1 – Домашняя страница survio

С помощью этого сервиса пользователь может создавать в своем личном кабинете опросы и посылать их другим людям. Затем после окончания опроса, сервис анализирует анализ опросов и предоставит анализ результатов владельцу опроса.

Из минусов это условно-бесплатность – можно взять демоверсию, но в будущем потребуют плату.

Самым популярным в мире и простым в использовании является Google Forms, домашняя страница которого представлена на рисунке 1.2.

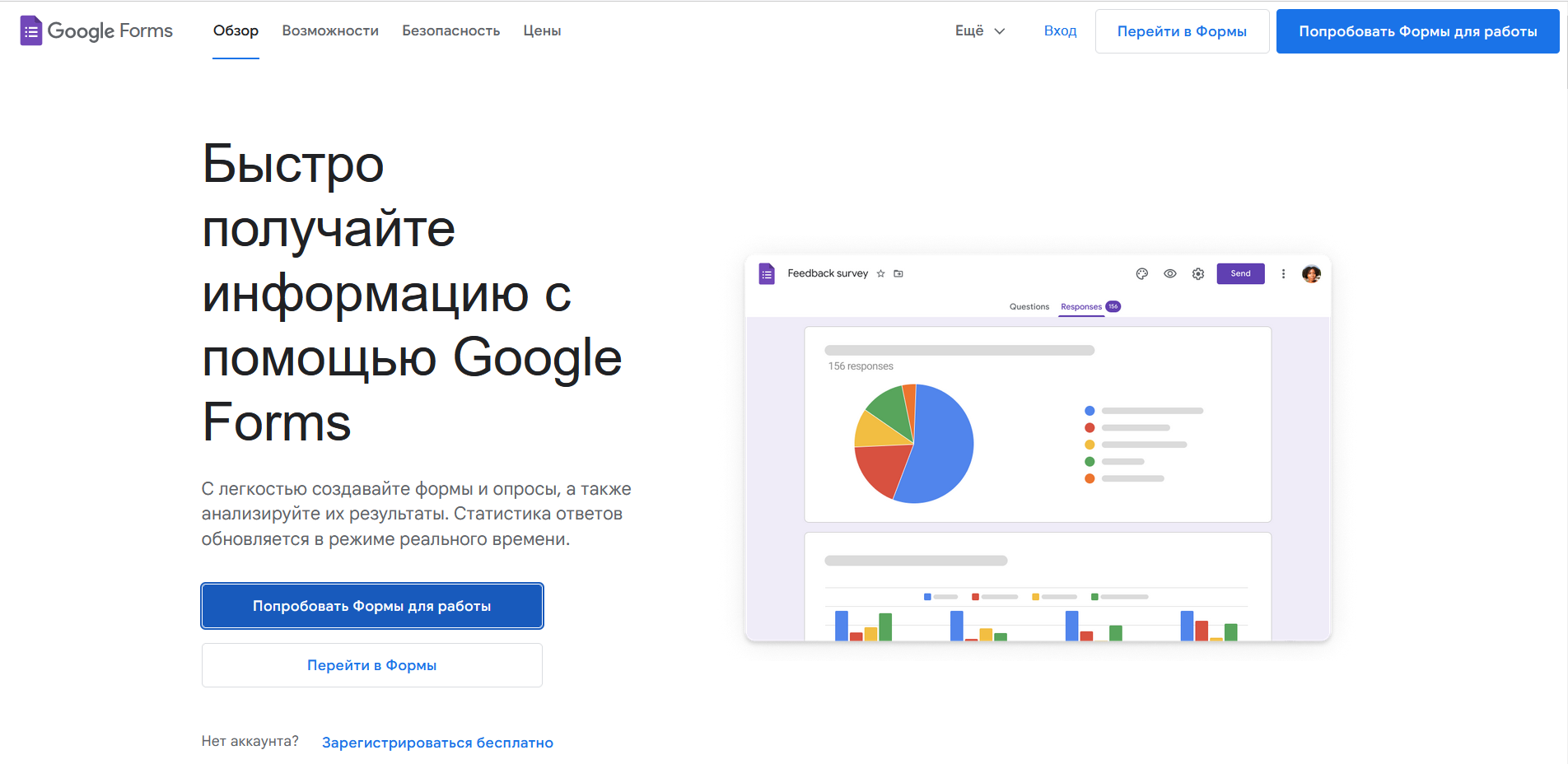


Рисунок 1.2 – Домашняя страница Google Forms

Данный сервис позволяет создавать не только опросы, но и формы. Пример готового теста приведен на рисунке 1.3.

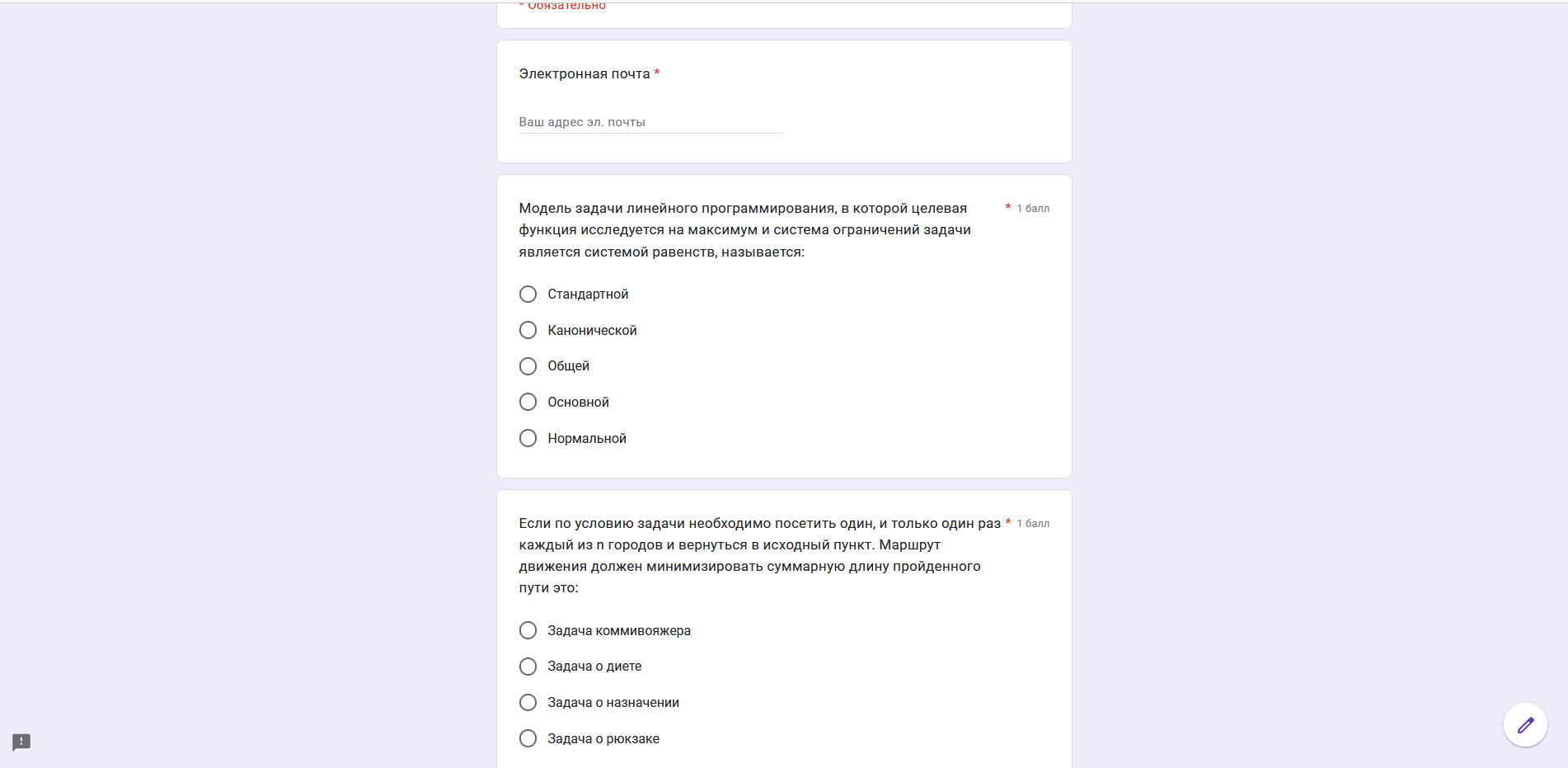


Рисунок 1.3 – Пример готового теста

Из недостатков можно отметить то, что у этого сайта местами неочевидный интерфейс.

2 Разработка программной системы

При разработке приложения для курсового проекта была использована база данных Postgresql 14.

2.1 Таблицы базы данных

Для реализации базы данных опросника было разработано 7 таблиц, диаграмма со структурой связей которых представлена в приложении А.

В структуру базы данных входят следующие таблицы: authorization, user, category, poll, question, option, answer. Далее представлено описание каждой из них.

Таблица authorization используется для авторизации пользователей, а именно их логина, хэше пароля и является ли пользователь администратором. Её структура представлена в таблице 2.1.

Таблица 2.1 – Структура таблицы authorization

|  |  |  |
| --- | --- | --- |
| Столбец | Тип данных | Ограничение целостности |
| auth\_id | integer | Primary key |
| login | varchar(50) | Not null, unique |
| password | text | Not null |
| is\_admin | boolean |  |

Таблица user используется для хранения информации о пользователях, а именно имени и фамилии. Её структура представлена в таблице 2.2.

Таблица 2.2 – Структура таблицы user

|  |  |  |
| --- | --- | --- |
| Столбец | Тип данных | Ограничение целостности |
| user\_id | integer | Primary key |
| first\_name | varchar(50) | Not null |
| last\_name | varchar(50) | Not null |
| auth\_id\_fk | integer | Foreign key |

Таблица category используется для хранения категорий опросов. Её структура представлена в таблице 2.3.

Таблица 2.3 – Структура таблицы category

|  |  |  |
| --- | --- | --- |
| Столбец | Тип данных | Ограничение целостности |
| category\_id | integer | Primary key |
| content | varchar(100) | Not null |

Таблица poll является хранилищем опросов, которое содержит имя опроса, описание опроса, дата создания опроса, дата закрытия опроса, открыт ли опрос, автор опроса и категория опроса. Её структура представлена в таблице 2.4.

Таблица 2.4 – Структура таблицы poll

|  |  |  |
| --- | --- | --- |
| Столбец | Тип данных | Ограничение целостности |
| poll\_id | integer | Primary key |
| name | varchar(100) | Not null |
| description | varchar(250) |  |
| date\_created | timestamp | Not null |
| date\_closed | timestamp |  |
| is\_open | boolean |  |
| user\_id\_fk | integer | Foreign key |
| Category\_id\_fk | integer | Foreign key |

Таблица question используется для хранения вопросов для опроса, которое содержит текст вопроса и к какому опросу принадлежит вопрос. Её структура представлена в таблице 2.5.

Таблица 2.5 – Структура таблицы question

|  |  |  |
| --- | --- | --- |
| Столбец | Тип данных | Ограничение целостности |
| question\_id | integer | Primary key |
| text | varchar(250) | Not null |
| poll\_id\_fk | integer | Foreign key, unique |

Таблица option используется для хранения ответов для вопроса, которое содержит текст ответа, количество проголосовавших за этот ответ и к какому вопросу принадлежит ответ. Её структура представлена в таблице 2.6.

Таблица 2.6 – Структура таблицы option

|  |  |  |
| --- | --- | --- |
| Столбец | Тип данных | Ограничение целостности |
| option\_id | integer | Primary key |
| text | varchar(250) | Not null |
| quantity | integer |  |
| question\_id\_fk | integer | Foreign key |

Таблица answer используется для хранения ответа от пользователя, а именно дата ответа, какой ответ был выбран и какому пользователю принадлежит ответ. Её структура представлена в таблице 2.7.

Таблица 2.7 – Структура таблицы answer

|  |  |  |
| --- | --- | --- |
| Столбец | Тип данных | Ограничение целостности |
| answer\_id | integer | Primary key |
| date\_answer | timestamp | Not null |
| user\_id\_fk | integer | Foreign key |
| option\_id\_fk | integer | Foreign key |

2.2 Роли и пользователи базы данных

База данных содержит 3 основных пользователя: poll\_programmer, poll\_admin и poll\_user. Пользователь poll\_programmer используется для создания инфраструктуры и объектов на базе данных. Остальные пользователи могут взаимодействовать с базой данных благодаря полномочий на вызов процедур и функций из указанных пакетов, которые описаны в главе 2.3.

Каждому пользователю надо дать схему для выполнения разрешенных процедур и функций для этого пользователя, но сначала нужно создать пользователя и забрать у него привилегии, которые создаются при создании пользователя. Создание пользователя poll\_user приведено в листинге 2.1.

|  |
| --- |
| create user poll\_user with password 'poll\_user';  revoke all privileges on database poll from poll\_user; |

Листинг 2.1 – Создание пользователя poll\_user

Для каждого пользователя создается его собственная cхема. Схема – поименованный набор функций и процедур. Схема для пользователя poll\_user представлен в листинге 2.2.

|  |
| --- |
| create schema if not exists poll\_user;  revoke all privileges on schema poll\_user from poll\_user;  grant usage on schema poll\_user to poll\_user;  grant execute on all routines in schema poll\_user to poll\_user; |

Листинг 2.2 – Создание схемы для пользователя poll\_user

Остальные пользователи базы данных создаются по тому же принципу, но разным схемам присваивается привилегия на другие вызовы процедур и функций.

2.3 Процедуры и функции для поставленных задач

Для управления данными, пользователи вызывают процедуры и функции, которые располагаются в схемах. Схема Postgres PL/SQL – это объект БД, который группирует логически связанные типы, элементы и подпрограммы.

В базе данных используется три пакета: poll\_programmer, poll\_admin и poll\_user, описание которых представлено ниже, графическое описание функций приведено в приложении Б.

2.3.1 Схема для пользователя

Схема poll\_user содержит 13 функций, предназначенных для взаимодействия с базой данных на уровне пользователя, описание этой схемы приведено в приложении В.

Функция find\_poll\_id позволяет пользователю найти идентификатор опроса. Она принимает два параметра: курсор и название вопроса.

Функция find\_question\_id позволяет пользователю найти идентификатор вопроса. Эта функция принимает два параметра: курсор и текст вопроса.

Функция get\_auth\_info позволяет пользователю получить о себе данные на авторизацию. Она принимает два параметра: курсор и идентификатор аутентификации.

Функция get\_option\_text позволяет пользователю получить текст ответа на вопрос. Она принимает два параметра: курсор и идентификатор вопроса.

Функция get\_poll\_status позволяет пользователю получить данные на опрос. Она принимает два параметра: курсор и идентификатор опроса.

Функция get\_polls\_name позволяет пользователю получить имя опроса. Она принимает один параметр:курсор.

Функция get\_questions\_text позволяет пользователю получить текст на вопрос. Она принимает один параметра: курсор.

Функция get\_user\_info позволяет пользователю получить о себе данные. Она принимает два параметра: курсор и идентификатор аутентификации.

Процедура insert\_data\_answer используется пользователем для добавления ответа. Она принимает параметром идентификатор пользователя и идентификатор ответа.

Процедура insert\_data\_authorization позволяет пользователю добавить авторизации для пользователя. Она принимает параметром идентификатор логин и пароль.

Процедура insert\_data\_user позволяет пользователя добавить пользователя. Она принимает параметром имя, фамилия и идентификатор аутентификации.

Процедура update\_login\_authorization используется пользователем для обновления логина. Она принимает параметры новый логин и идентификатор аутентификации.

Процедура update\_password\_authorization используется пользователем для обновления пароля. Она принимает параметры новый пароль и идентификатор аутентификации.

2.3.2 Схема для администратора

Пакет poll\_admin содержит 43 процедур и функций, предназначенных для взаимодействия с базой данных на уровне администратора, описание этой схемы приведено в приложении Г.

Процедура delete\_selected\_category используется администратором для удаления категории. Она принимает параметром название категории.

Процедура delete\_selected\_poll используется администратором для удаления опроса. Она принимает параметром название опроса. При удалении опроса вместе с ним удаляется вопросы, ответы и ответы от пользователей для выбранного опроса.

Функция find\_auth\_id принимает параметры такие, как курсор и логин пользователя. Требуется если известен только логин пользователя.

Функция find\_category\_id принимает параметры такие, как курсор и название категории. Требуется если известно название категории.

Функция find\_option\_id принимает параметры такие, как курсор и текст ответа. Требуется если известно текст ответа.

Функция find\_option\_id\_by\_question\_id принимает параметры такие, как курсор и идентификатор вопроса.

Функция find\_poll\_id принимает параметры такие, как курсор и название опроса. Требуется если известно название опроса.

Функция find\_question\_id принимает параметры такие, как курсор и название вопроса. Требуется если известно название вопроса.

Функция find\_question\_id\_by\_poll\_id принимает параметры такие, как курсор и идентификатор опроса.

Функция find\_user\_id принимает параметры такие, как курсор и идентификатор аутентификации.

Функция get\_auth\_info позволяет администратору получить о себе данные на авторизацию. Она принимает два параметра: курсор и идентификатор аутентификации.

Функция get\_option\_text позволяет администратору получить текст ответа на вопрос. Она принимает два параметра: курсор и идентификатор вопроса.

Функция get\_poll\_status позволяет администратору получить данные на опрос. Она принимает два параметра: курсор и идентификатор опроса.

Функция get\_polls\_name позволяет администратору получить имя опроса. Она принимает один параметр:курсор.

Функция get\_questions\_text позволяет администратору получить текст на вопрос. Она принимает один параметра: курсор.

Функция get\_user\_info позволяет администратору получить о себе данные. Она принимает два параметра: курсор и идентификатор аутентификации.

Функция find\_poll\_id принимает параметры такие, как курсор и название опроса. Требуется если известно название опроса.

Функция find\_poll\_id принимает параметры такие, как курсор и название опроса. Требуется если известно название опроса.

Для получения название категории, администратору потребуется воспользоваться функцией get\_categories\_content. Принимаемые параметры только курсор.

При большом количестве категорий, администратору потребуется воспользоваться функцией get\_categories\_content\_pagination. Принимаемые параметры курсор, номер страницы, количество записей на странице.

Функция get\_option\_id\_from\_answer\_by\_user\_id принимает параметры такие, как курсор и идентификатор пользователя. Принимаемые параметры курсор и идентификатор пользователя.

Функция get\_option\_texts\_by\_question\_id позволяет администратору получить тексты ответа по идентификатору вопроса. Принимаемые параметры курсор и идентификатор вопроса.

Функция get\_questions\_texts\_by\_poll \_id позволяет администратору получить текст вопроса по идентификатору опроса. Принимаемые параметры курсор и идентификатор опроса.

Функция get\_text\_and\_quantity\_by\_question\_id позволяет администратору получить текст и количество ответов по идентификатору вопроса. Принимаемые параметры курсор и идентификатор вопроса.

Процедура insert\_data\_answer используется администратором для добавления ответа пользователя администратором. Она принимает параметры идентификатор пользователя и идентификатор ответа.

Процедура insert\_data\_authorization используется администратором для добавления авторизации для пользователя. Она принимает параметры идентификатор логин и пароль.

Процедура insert\_data\_user используется администратором для добавления ответа пользователя. Она принимает параметры имя, фамилия и идентификатор аутентификации.

Процедура insert\_data\_option используется администратором для добваления ответа на вопрос. Она принимает параметры текст ответа и идентификатор вопроса.

Процедура insert\_data\_сategory используется администратором для добавления категории. Она принимает параметром название категории.

Процедура insert\_data\_poll используется администратором для добавления опроса. Она принимает параметры название, описание статус опроса, идентификатор пользователя и идентификатор категории.

Процедура insert\_data\_question используется администратором для добавления вопроса. Она принимает параметры текст вопроса и идентификатор опроса.

Функция select\_answers позволяет администратору получить данные с таблицы ответов. Принимаемые параметр курсор.

Функция select\_authorizations позволяет администратору получить данные с таблицы авторизации. Принимаемые параметр курсор.

Функция select\_categories позволяет администратору получить данные с таблицы категорий. Принимаемые параметр курсор.

Функция select\_options позволяет администратору получить данные с таблицы ответов. Принимаемые параметр курсор.

Функция select\_polls позволяет администратору получить данные с таблицы опросов. Принимаемые параметр курсор.

Функция select\_questions позволяет администратору получить данные с таблицы вопросов. Принимаемые параметр курсор.

Функция select\_users позволяет администратору получить данные с таблицы пользователей. Принимаемые параметр курсор.

Процедура update\_login\_authorization используется администратором для обновления логина. Она принимает параметры новый логин и идентификатор аутентификации.

Процедура update\_password\_authorization используется администратором для обновления пароля. Она принимает параметры новый пароль и идентификатор аутентификации.

Процедура update\_content\_category используется администратором для обновления категории. Она принимает параметры новое название категории и идентификатор категории.

Процедура update\_poll\_date\_closed используется администратором для обновления даты закрытия опроса. Она принимает параметры новую дату закрытия и идентификатор опроса.

Процедура update\_poll\_status используется администратором для обновления статуса опроса. Она принимает параметры новый статус опроса и идентификатор опроса.

Процедура update\_quantity\_option используется администратором для обновления счетчика ответов. Она принимает параметр новый идентификатор ответа.

Процедура update\_text\_question используется администратором для обновления текста вопроса. Она принимает параметры новый текст опроса и идентификатор вопроса.

2.3.3 Схема для программиста

Схема poll\_programmer содержит 5 процедур и функций и одного триггера, предназначенных для взаимодействия с базой данных на уровне программиста, описание этой схемы приведено в приложении Д. Особенность программиста, что ему принадлежат все процедуры и функции, если их нет даже в схеме.

Процедура insert\_100\_000\_data\_category не принимает параметров и служит для вставки 100000 строк в таблицу категорий.

Процедура delete\_100\_000\_data\_category не принимает параметров и служит для удаления 100000 строк в таблице категорий.

Процедура export\_category\_by\_id служит для экспортирования данных из таблицы категорий с диапазоном по идентификатору. Принимает параметры с какого идентификатора экспортировать данные, до какого идентификатора экспортировать данные и путь, куда сохранить экспортированные данные.

Процедура import\_category\_data служит для импортирования с файла в формате xml. Принимает параметр путь до файла, откуда будет производиться импортирование данных.

Функция alert\_is\_admin\_change нужна для триггера. Создает файл при срабатывании триггера. Не принимает параметры.

2.4 Триггеры базы данных

Для решения задачи об оповещении программиста о действиях в базе данных были использованы триггеры. Триггер – именованный программный блок, выполняемый в ответ на происходящие в базе данных события.

Для реализации системы оповещений был создан триггер: update\_is\_admin. Реализация этого триггера приведена в приложении Е.

Триггер update\_is\_admin используется для информирования программиста, что было изменение столбца is\_admin в таблице authorization. Он срабатывает при изменении на таблице authorization. В функции связанной с триггером создается файл с названием alert.txt, что было изменение статуса пользователя.

.

3 Тестирование и улучшение производительности

Для тестирования номинальности триггеров, было выполнено изменение поля is\_admin в таблицу authorization случайного из пользователей. После изменения значение в столбце был создан файл alert.txt.

Для улучшения производительности была создана специальная функция с пагинацией, которая уменьшила обращение к таблице category с содержимым 100000 строк и более. Прежняя стоимость соствляла 1456, а с улучшением стало 150. Раньше таблица могла грузить данные по 5 секунд и более, а с улучшением примерно 100 милисекунд.

Также была проведена проверка на отказоустойчивость приложения. Например, на рисунке 3.1 представлено, что будет если попытаться вставить более одного вопроса в опрос.

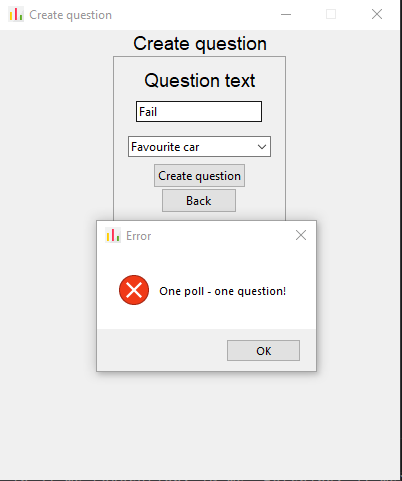


Рисунок 3.1 – Результат запроса на получение городов

4 Описание технологии

В качестве основной технологии данного курсового проекта была выбрана настройка системы безопасности сервера СУБД, такие как настройка файрвола, прослушивание конкретных адресов, безопасность на уровне строк, аудит. Эта технология включает в себя защиту сервера СУБД от несанкционированного доступа.

Защита баз данных необходимо от хищение данных, несанкционированного доступа, перегрузок, снижение производительности с последующим ограничением работы авторизованных пользователей, поэтому было принято решение сделать защиту как в приложении, так и в базе данных

4.1 Настройка файрвола

Для настройки файрвола, требовалось запустить командную строку от имени администратора и ввести команду: netsh advfirewall firewall add rule name="Postgre Port" dir=in action=allow protocol=TCP localport=5432. В этой команде была создано правило для файрвола с именем Postgre Port и с разрешающим портом 5432

4.2 Прослушивание конкретных адресов

Для настройки прослушивание конкретных адресов, требовалось зайти в установленную папку PostgreSQL и в файле postgresql.conf в разделе CONNECTIONS AND AUTHENTICATION параметр установить в значение listen\_addresses в localhost в одинарных кавычках.

4.3 Безопасность на уровне строк

Одной из наиболее продвинутых функций системы привилегий PostgreSQL является безопасность на уровне строк, которая позволяет предоставлять привилегии подмножеству строк в таблице.

Чтобы начать использовать безопасность на уровне строк, вам нужны две вещи: включить ее для таблицы и определить политику, которая будет контролировать доступ на уровне строк.

Включение безопасности на уровне строк для всех таблиц приведено в листинге 4.3.1.

|  |
| --- |
| alter table answer enable row level security;  alter table "option" enable row level security;  alter table poll enable row level security;  alter table question enable row level security;  alter table "user" enable row level security;  alter table "authorization" enable row level security;  alter table "category" enable row level security; |

Листинг 4.3.1 – Включение безопасности на уровне строк для всех таблиц

После включения безопасности на уровне строк, создадим политику. Пример политики на удаление приведен в листинге 4.3.2, которая позволяет удалять в таблице category строки если пользователь poll\_programmer или poll\_admin.

|  |
| --- |
| create policy delete\_category  on "category" for delete  using (current\_user = 'poll\_programmer' or current\_user = 'poll\_admin'); |

Листинг 4.3.2 – Политика на удаление

4.4 Аудит

Аудит базы данных является безопасность базы данных и её контроль, который также может включать в себя некоторые другие виды аспектов мониторинга. Аудит включает себя также введение журналов, что происходило в базе данных. Для включения введения журнала требуется зайти в установленную папку PostgreSQL и в файле postgresql.conf найти параметры log\_connections, log\_disconnections и log\_statement. В двух первых прописать on, а в последнем all. Теперь в папке log там, где и находится файл postgresql.conf будет теперь записываться кто присоединился, кто отсоединился и все выполненные операторы в базе данных.

5 Руководство пользователя

Разработанное в рамках курсового проекта приложение поддерживает две роли: пользователь и администратор. Запуская приложение пользователю предоставляется окно для авторизации, а также кнопка для открытия окна регистрации пользователя. Главная страница приложения представлена на рисунке 5.1.

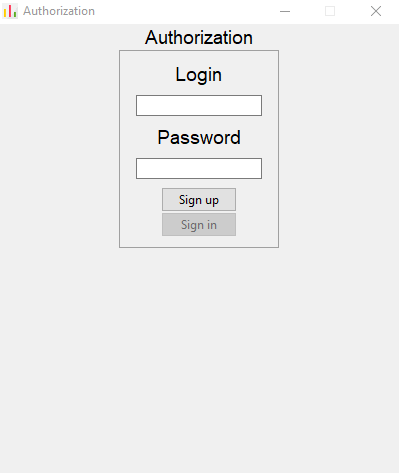


Рисунок 5.1 – Главная страница приложения

После успешной авторизации пользователь попадает в окно, в котором он может выбрать из выпадающего списка за какой опрос проголосовать или просмотреть информацию о себе.

При нажатии кнопке информацию о себе, пользователь модет просмотреть свою личную информацию и имеет возможность сменить пароль и логин. Основное окно пользователя представлено на рисунке 5.2.

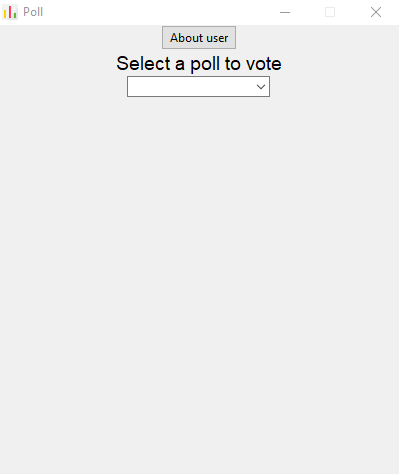


Рисунок 5.2 – Основная страница пользователя

Если авторизация была выполнена под администратором, то приложение откроет окно для администратора. В окне для администратора администратор имеет кнопки для создания или удаления опросов, создания категории или удаления их, создание вопросов, создание ответов, статус опроса и промежуточные результаты опроса.

При удалении все категории будут выведены постранично, по 1000 категорий на страницу. Переключение между страницами осуществляется с помощью кнопок под списком.

При создании опроса администратор должен заполнить все поля и выбрать все необходимое, иначе кнопка создания будет неактивна.

При создании вопроса администратор должен заполнить все поля и выбрать все необходимое, иначе кнопка создания будет неактивна. На каждый опрос – один вопрос. При попытке нарушить это правило будет выдана ошибка

При создании категории администратор должен заполнить все поля и выбрать все необходимое, иначе кнопка создания будет неактивна.

При создании ответов администратор должен заполнить все поля и выбрать все необходимое, иначе кнопка создания будет неактивна.

При удалении опросов администратор должен понимать, что все связанные данные с опросом будут удалены.

Пример интерфейса для окна администратора приведен на рисунке 5.3.

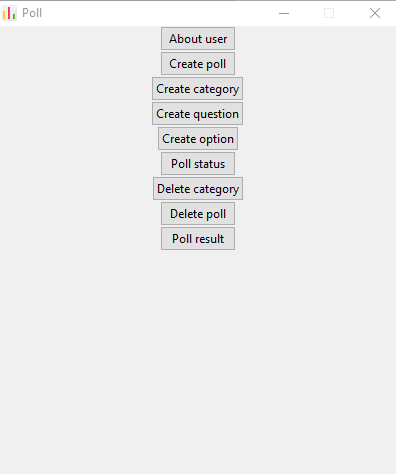


Рисунок 5.3 – Главная страница администратора

Заключение

Результатом выполнения курсового проекта стало десктопное приложение с использованием графического интерфейса Tkinter для клиентской части и предоставления соединения с базой данных при помощи Psycopg. В ходе курсового проекта была использована база данных PostgreSQL 14, которая спроектирована для опросника.

Были реализованы основные требования, указанные в листе задания вместе со следующими пунктами:

* регистрация пользователей;
* авторизация пользователей;
* пароли при регистрации шифруются;
* возможность изменения личной информации пользователю и администратору;
* просмотр промежуточных результатов опроса администратором;
* уведомление программиста об изменение статуса пользователя;
* логирование всех действий в базе данных;

Данный курсовой проект является примером разработки базы данных и приложения для работы с ней. Протестировав все компоненты приложения, можно прийти к заключению, что все основные требования выполнены и приложение работает исправно.

Список литературы

1. PostgreSQL Documentation [Электронный ресурс] / Режим доступа: https://www.postgresql.org/docs/ – Дата доступа: 15.10.2022.

2. Обеспечение безопасности базы данных PostgreSQL [Электронный ресурс] / Режим доступа: https://habr.com/ru/post/550882/ – Дата доступа: 01.11.2022.

3. Установка и базовая настройка PostgreSQL в Windows 10 [Электронный ресурс] / Режим доступа: https://winitpro.ru/index.php/2019/10/25/ustanovka-nastrojka-postgresql-v-windows/ – Дата доступа: 13.12.2022

4.  PostgreSQL Antipatterns: накручиваем себе проблемы [Электронный ресурс] / Режим доступа: https://habr.com/ru/company/tensor/blog/507688/ – Дата доступа: 26.11.2022

5. Коллекция готовых SQL запросов для PostgreSQL [Электронный ресурс] / Режим доступа: https://github.com/rin-nas/postgresql-patterns-library – Дата доступа: 7.12.2022.

Приложение А

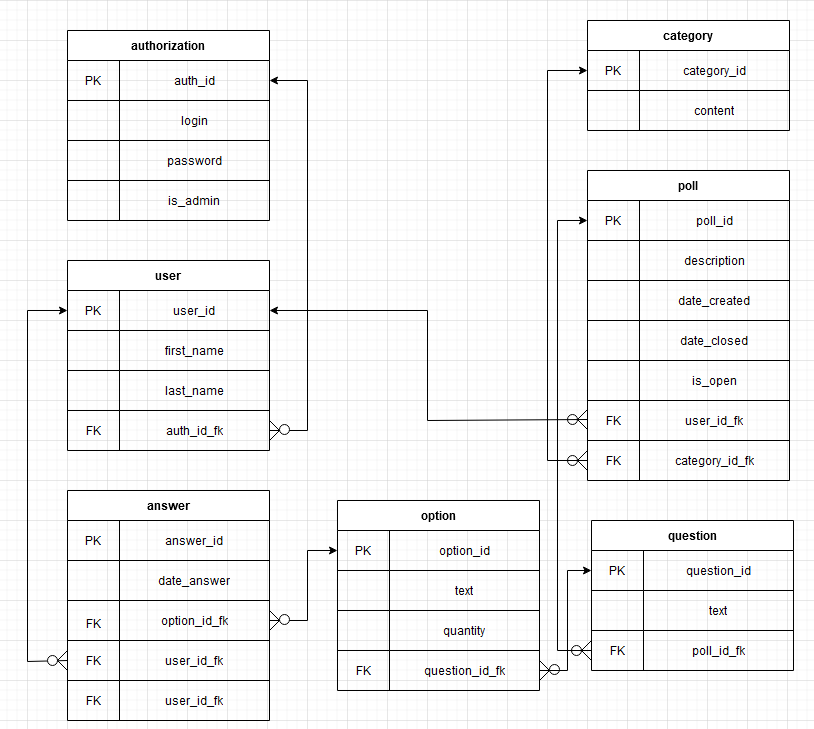


Схема базы данных

Приложение Б

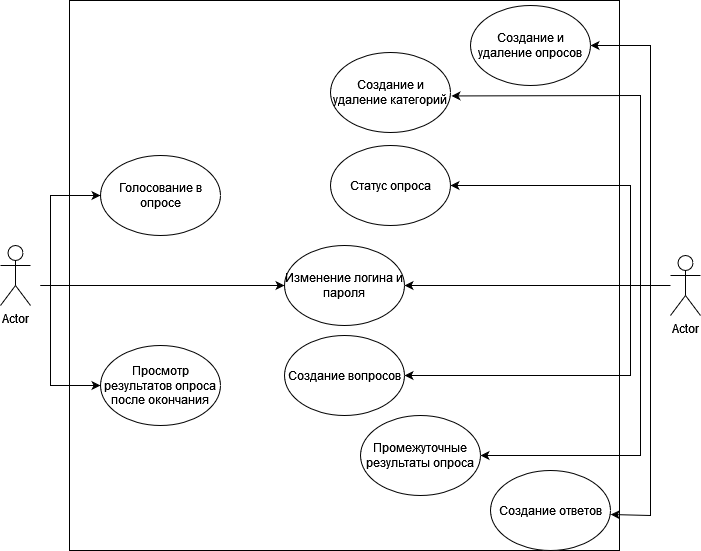


Диаграмма использования

Приложение В

Описание процедур и функций схемы«poll\_user»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя функции/процедуры | Название параметра | Тип данных | Тип параметра |
| find\_poll\_id | - | refcursor | out |
| find\_poll\_id | ref | refcursor | in |
| find\_poll\_id | poll\_name | varchar | in |
| find\_question\_id | - | refcursor | out |
| find\_question\_id | ref | refcursor | in |
| find\_question\_id | question\_text | varchar | in |
| get\_auth\_info | - | refcursor | out |
| get\_auth\_info | ref | refcursor | in |
| get\_auth\_info | auth\_id\_p | integer | in |
| get\_option\_text | - | refcursor | out |
| get\_option\_text | ref | refcursor | in |
| get\_option\_text | question\_id\_p | integer | in |
| get\_poll\_status | - | refcursor | out |
| get\_poll\_status | ref | refcursor | in |
| get\_poll\_status | poll\_id\_p | integer | in |
| get\_polls\_name | - | refcursor | out |
| get\_polls\_name | ref | refcursor | in |
| get\_questions\_text | - | refcursor | out |
| get\_questions\_text | ref | refcursor | in |
| get\_user\_info | - | refcursor | out |
| get\_user\_info | ref | refcursor | in |
| get\_user\_info | auth\_id\_p | integer | in |
| insert\_data\_answer | user\_id\_fk | integer | in |
| insert\_data\_answer | option\_id\_fk | integer | in |
| insert\_data\_authorization | login | varchar | in |
| insert\_data\_authorization | password | text | in |
| insert\_data\_user | first\_name | varchar | in |
| insert\_data\_user | last\_name | varchar | in |
| update\_login\_authorization | new\_login | varchar | in |
| update\_login\_authorization | auth\_id\_p | integer | in |
| update\_password\_authorization | new\_password | text | in |
| update\_password\_authorization | auth\_id\_p | integer | in |

Приложение Г

Описание процедур и функций схемы «poll\_admin»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя функции/процедуры | Название параметра | Тип данных | Тип параметра |
| delete\_selected\_category | Content\_p | varchar | in |
| delete\_selected\_poll | name\_p | varchar | in |
| find\_auth\_id | - | refcursor | out |
| find\_auth\_id | ref | refcursor | in |
| find\_auth\_id | login\_p | varchar | in |
| find\_category\_id | - | refcursor | out |
| find\_category\_id | ref | refcursor | in |
| find\_category\_id | content\_p | varchar | in |
| find\_option\_id | - | refcursor | out |
| find\_option\_id | ref | refcursor | in |
| find\_option\_id | option\_text | varchar | in |
| find\_option\_id\_by\_question\_id | - | refcursor | out |
| find\_option\_id\_by\_question\_id | ref | refcursor | in |
| find\_option\_id\_by\_question\_id | question\_id\_p | varchar | in |
| find\_poll\_id | - | refcursor | out |
| find\_poll\_id | ref | refcursor | in |
| find\_poll\_id | poll\_name | varchar | in |
| find\_question\_id | - | refcursor | out |
| find\_question\_id | ref | refcursor | in |
| find\_question\_id | question\_text | varchar | in |
| find\_question\_id\_by\_poll\_id | - | refcursor | out |
| find\_question\_id\_by\_poll\_id | ref | refcursor | in |
| find\_question\_id\_by\_poll\_id | poll\_id\_p | varchar | in |
| find\_user\_id | - | refcursor | out |
| find\_user\_id | ref | refcursor | in |
| find\_user\_id | auth\_id\_p | varchar | in |
| get\_auth\_info | - | refcursor | out |
| get\_auth\_info | ref | refcursor | in |
| get\_auth\_info | auth\_id\_p | varchar | in |
| get\_categories\_content | - | refcursor | out |
| get\_categories\_content | ref | refcursor | in |
| get\_categories\_content\_pagination | - | refcursor | out |
| get\_categories\_content\_pagination | ref | refcursor | in |
| get\_categories\_content\_pagination | page | integer | in |
| get\_categories\_content\_pagination | limit\_p | integer | in |
| get\_option\_id\_from\_answer\_by\_user\_id | - | refcursor | out |
| get\_option\_id\_from\_answer\_by\_user\_id | ref | refcursor | in |
| get\_option\_id\_from\_answer\_by\_user\_id | user\_id\_p | integer | in |

Продолжение процедур и функций схемы «poll\_admin»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя функции/процедуры | Название параметра | Тип данных | Тип параметра |
| get\_option\_text | - | refcursor | out |
| get\_option\_text | ref | refcursor | in |
| get\_option\_text | question\_id\_p | integer | in |
| get\_option\_texts\_by\_question\_id | - | refcursor | out |
| get\_option\_texts\_by\_question\_id | ref | refcursor | in |
| get\_option\_texts\_by\_question\_id | question\_id\_p | integer | in |
| get\_poll\_status | - | refcursor | out |
| get\_poll\_status | ref | refcursor | in |
| get\_poll\_status | poll\_id\_p | integer | in |
| get\_polls\_name | - | refcursor | out |
| get\_polls\_name | ref | refcursor | in |
| get\_question\_texts\_by\_poll\_id | - | refcursor | out |
| get\_question\_texts\_by\_poll\_id | ref | refcursor | in |
| get\_question\_texts\_by\_poll\_id | poll\_id\_p | integer | in |
| get\_questions\_text | - | refcursor | out |
| get\_questions\_text | ref | refcursor | in |
| get\_text\_and\_quantity\_by\_question\_id | - | refcursor | out |
| get\_text\_and\_quantity\_by\_question\_id | ref | refcursor | in |
| get\_text\_and\_quantity\_by\_question\_id | question\_id\_p | integer | in |
| get\_user\_info | - | refcursor | out |
| get\_user\_info | ref | refcursor | in |
| get\_user\_info | auth\_id\_p | integer | in |
| insert\_data\_answer | user\_id\_p | integer | in |
| insert\_data\_answer | option\_id\_fk | integer | in |
| insert\_data\_authorization | login | varchar | in |
| insert\_data\_authorization | password | text | in |
| insert\_data\_category | content | varchar | in |
| insert\_data\_option | text\_p | varchar | in |
| insert\_data\_option | question\_id\_p | integer | in |
| insert\_data\_poll\_create | name | varchar | in |
| insert\_data\_poll\_create | description | varchar | in |
| insert\_data\_poll\_create | is\_open | boolean | in |
| insert\_data\_poll\_create | user\_id\_fk | integer | in |
| insert\_data\_poll\_create | category\_id\_fk | integer | in |
| insert\_data\_question | text\_p | varchar | in |
| insert\_data\_question | poll\_id\_p | integer | in |
| insert\_data\_user | first\_name | varchar | in |
| insert\_data\_user | last\_name | varchar | in |
| insert\_data\_user | auth\_id\_fk | integer | in |

Продолжение процедур и функций схемы «poll\_admin»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя функции/процедуры | Название параметра | Тип данных | Тип параметра |
| poll\_already\_have\_question | - | refcursor | out |
| poll\_already\_have\_question | ref | refcursor | in |
| poll\_already\_have\_question | poll\_id\_p | integer | in |
| select\_answers | - | refcursor | out |
| select\_answers | ref | refcursor | in |
| select\_authorizations | - | refcursor | out |
| select\_authorizations | ref | refcursor | in |
| select\_categories | - | refcursor | out |
| select\_categories | ref | refcursor | in |
| select\_options | - | refcursor | out |
| select\_options | ref | refcursor | in |
| select\_polls | - | refcursor | out |
| select\_polls | ref | refcursor | in |
| select\_questions | - | refcursor | out |
| select\_questions | ref | refcursor | in |
| select\_users | - | refcursor | out |
| select\_users | ref | refcursor | in |
| update\_content\_category | new\_content | varchar | in |
| update\_content\_category | category\_id\_p | integer | in |
| update\_login\_authorization | new\_login | varchar | in |
| update\_login\_authorization | auth\_id\_p | integer | in |
| update\_password\_authorization | new\_password | text | in |
| update\_password\_authorization | auth\_id\_p | integer | in |
| update\_poll\_date\_closed | date\_closed\_p | timestamp | in |
| update\_poll\_date\_closed | poll\_id\_p | integer | in |
| update\_poll\_status | new\_status | boolean | in |
| update\_poll\_status | poll\_id\_p | integer | in |
| update\_quantity\_option | option\_id\_p | integer | in |
| update\_text\_question | new\_text | varchar | in |
| update\_text\_question | question\_id\_p | integer | in |

Приложение Д

Описание процедур и функций схемы «poll\_programmer»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя функции/процедуры | Название параметра | Тип данных | Тип параметра |
| delete\_100\_000\_data\_category | - | - | - |
| insert\_100\_000\_data\_category | - | - | - |
| export\_category\_by\_id | category\_id\_p\_from | integer | in |
| export\_category\_by\_id | category\_id\_p\_to | integer | in |
| export\_category\_by\_id | filepath | text | in |
| import\_category\_data | filepath | text | in |
| alert\_is\_admin\_change | - | - | - |

Приложение Е

|  |
| --- |
| create trigger update\_is\_admin  after update on "authorization" for each row  execute procedure poll\_programmer.alert\_is\_admin\_change(); |

Листинг – Создание триггеров