Malina 64 v.2.00

1. Описание

* Описание проекта
* Цель проекта
* Системные требования
* Используемые библиотеки и фреймворки
* Установка

1. Использование

* Описание функциональности
* Как подключиться к базе данных

1. Схема базы данных

* Описание структуры таблиц
* Связи между таблицами

1. Структура проекта, файлы, классы и функции

* Структура проекта, описание файлов
* Описание классов и функций
* Их параметры
* Их возвращаемые значения
* Примеры использования

1. Тесты

* Описание тестов
* Их параметры
* Ожидаемые результаты
* Примеры использования

1. Руководство по обслуживанию

* Проблемы, которые могут возникнуть
* Решения для часто возникающих проблем
* Общие рекомендации по сопровождению

# **ОПИСАНИЕ**

## Описание проекта

Приложение Malina64 создается как сборник программ для решения задач возникающих у сотрудников при выполнении их функциональных обязанностей.

## Цель проекта

Создать полнофункциональное приложение, на языке Python, с записью и получением необходимых данных в/из БД PostgreSQL.

## Системные требования

Windows 7 или выше. Наличие пакета Microsoft Excel версии 10 или выше. Наличие интернет-соединения.

## Используемые библиотеки и фреймворки

PyQt6 – для создания интерфейсов (UI)

Passlib – для шифрования паролей

Psycopg2 – для работы с БД PostgreSQL

## Установка

Клонирование из репозитория GitHub

<https://github.com/PavelZagumenkin/Malina_Orders_v2.git>

[git@github.com:PavelZagumenkin/Malina\_Orders\_v2.git](mailto:git@github.com:PavelZagumenkin/Malina_Orders_v2.git)

gh repo clone PavelZagumenkin/Malina\_Orders\_v2

# **ИСПОЛЬЗОВАНИЕ**

## Описание функциональности

* 1. **Регистрация пользователя (только через администратора)**
  2. **Авторизация пользователя**
  3. **Выбор раздела для работы в зависимости от прав пользователя**
  4. **Раздел Автозаказы**
* Подраздел Выпечка
* Подраздел Пирожные
* Подраздел Торты
* Подраздел Начинки
* Будущие подразделы

Создание заказов на производство выбранных позиций, анализ торговли на данных из продающей системы iiko (таблицы Excel), прогнозирование торговли, сохранение необходимых промежуточных значений в БД, редактирование промежуточных или конечных значений в БД, удаление данных из БД, получение данных из БД, верификация действий пользователя, логирование ключевых действий пользователя, выгрузка результата работы программы в Excel, печать результата работы программы на принтер.

* 1. **Раздел Зарплата**

Создание графика работы сотрудников. Фактическое табелирование. Возможность перераспределение % от выручки между сотрудниками (бывшая мотивация). Ежедневная связь отработанного времени с выручкой, с отображение заработка почти в реальном времени. Возможность редактирования «на лету» ключевых показателей влияющих на зарплату (процент от выручки). Отображение необходимы сводных, средних показателей. Выгрузка результата работы в удобном формате. Печать результата работы в удобном формате на притер. Сохранение, редактирование, удаление, просмотр всех промежуточных и конечных результатов в БД. Верификация и логирование всех ключевых действий пользователей.

* 1. **Раздел История**

Доступ ко всех логам пользователей, просмотр сравнения внесенных изменений и изначальных данных, откат внесенных изменений, доступ к аналитическим данным.

* 1. **Раздел Управление**

Административный раздел разработчика. Доступ к настройкам подключения к БД. Регистрация новых пользователей. Изменение прав пользователей. Удаление пользователей. Доступ ко всем логам. Отладка приложения. Доступ к консоли разработчика. Индивидуальные запросы к БД. Выпуск обновлений программы. Сброс паролей пользователей.

## Как подключится к базе данных

Имя базы данных – malina64

Пароль - спросите у разработчика.

# **СХЕМА БАЗЫ ДАННЫХ**

## Описание структуры таблиц

* 1. Таблица **version**

|  |  |
| --- | --- |
| [PK]**id** (integer) | **version** (numeric) |
| 1 | 2.02 |

* 1. Таблица **users**

|  |  |  |
| --- | --- | --- |
| [PK]**id** (integer) | **username** (character varying (50)) | **password** (character varying (100)) |
| 1 | logist | HEX |
| 2 | supervisor | HEX |

* 1. Таблица **users\_role**

|  |  |  |
| --- | --- | --- |
| [PK]**id** (integer) | **username** (character varying (50)) | **role** (character varying (50)) |
| 1 | logist | logist |
| 2 | supervisor | supervisor |

* 1. Таблица **konditerskie**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [PK]**id** (integer) | **konditerskie\_name** (character varying (50)) | **kond\_bakery** (integer) | **kond\_pie** (integer) | **kond\_cake** (integer) | **sklad\_bakery**  (numeric) |
| 1 | Магазин | 1 | 1 | 1 | 2.5 |
| 2 | Чехова | 0 | 1 | 1 | 0 |

* 1. Таблица **foods**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [PK] **id** (integer) | **food\_kod** (character varying (50)) | **food\_name** (character varying (50)) | **food\_kod\_pf** (character varying (50)) | **food\_name\_pf** (character varying (50)) | **bakery\_layout** (integer) | **bakery\_kf** (numeric) | **bakery\_otgruz** (numeric) | **bakery\_zames** (numeric) | **pie\_kvant** (integer) | **pie\_zames** (integer) | **cake\_massa** (numeric) |
| 1 | 00004 | Курник | 00005 | Курник п/ф | 4 | 1.2 | 10 | 50 |  |  |  |

* 1. Таблица **bakery\_prognoz**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [PK] **id** (integer) | **data** (data) | **day\_week**  (character varying (50)) | **index**  **(**character varying (50)) | **food\_kod** (character varying (50)) | **food\_name** (character varying (50)) | **kf\_food**  (numeric) | kond\_  name1 | kond\_  name1 |
| 1 | 27.03.2023 |  | kf\_food | 00004 | Курник | 1.2 | 256 | 128 |
| 2 | 27.03.2023 | Понедельник | kf\_day\_week |  |  |  | 1.1 | 1.0 |
| 3 | 27.03.2023 |  | kf\_kondi |  |  |  | 1.2 | 1.1 |

* 1. Таблица **pie\_prognoz**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [PK] **id** (integer) | **data** (data) | **day\_week**  (character varying (50)) | **index**  **(**character varying (50)) | **food\_kod** (character varying (50)) | **food\_name** (character varying (50)) | **kf\_food**  (numeric) | kond\_  name1 | kond\_  name1 |
| 1 | 27.03.2023 |  | kf\_food | 02546 | Трайфл Малина | 1.2 | 256 | 128 |
| 2 | 27.03.2023 | Понедельник | kf\_day\_week |  |  |  | 1.1 | 1.0 |
| 3 | 27.03.2023 |  | kf\_kondi |  |  |  | 1.2 | 1.1 |

* 1. Таблица **cake\_prognoz**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [PK] **id** (integer) | **data** (data) | **day\_week**  (character varying (50)) | **index**  **(**character varying (50)) | **food\_kod** (character varying (50)) | **food\_name** (character varying (50)) | **kf\_food**  (numeric) | kond\_  name1 | kond\_  name1 |
| 1 | 27.03.2023 |  | kf\_food | 03569 | Торт Наполеон | 1.2 | 256 | 128 |
| 2 | 27.03.2023 | Понедельник | kf\_day\_week |  |  |  | 1.1 | 1.0 |
| 3 | 27.03.2023 |  | kf\_kondi |  |  |  | 1.2 | 1.1 |

* 1. Таблица **logs**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [PK] **id** (integer) | **data\_logs**  (data) | **username**  (character varying (50)) | **role** (character varying (50)) | **table**  (character varying (50)) | id | whay\_is\_done | **archive\_table**  (character varying (50)) | id\_archive | column | old | new |
| 1 | 28.03.2023 09.03.55 | logist | logist | bakery\_prognoz | 1 | create |  |  |  |  |  |
| 2 | 28.03.2023 09.03.51 | logist | logist | cake\_prognoz | 1 | update |  |  | Беговая | 1.1 | 1.2 |
| 3 | 28.03.2023 09.05.54 | logist | logist | pie\_prognoz | 2 | delete | archive\_cake\_prognoz | 1 |  |  |  |

* 1. Таблица **archive\_bakery\_prognoz**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [PK] **id** (integer) | **data** (data) | **day\_week**  (character varying (50)) | **index**  **(**character varying (50)) | **food\_kod** (character varying (50)) | **food\_name** (character varying (50)) | **kf\_food**  (numeric) | kond\_  name1 | kond\_  name1 |
| 1 | 27.03.2023 |  | kf\_food | 00004 | Курник | 1.2 | 256 | 128 |
| 2 | 27.03.2023 | Понедельник | kf\_day\_week |  |  |  | 1.1 | 1.0 |
| 3 | 27.03.2023 |  | kf\_kondi |  |  |  | 1.2 | 1.1 |

* 1. Таблица **archive\_pie\_prognoz**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [PK] **id** (integer) | **data** (data) | **day\_week**  (character varying (50)) | **index**  **(**character varying (50)) | **food\_kod** (character varying (50)) | **food\_name** (character varying (50)) | **kf\_food**  (numeric) | kond\_  name1 | kond\_  name1 |
| 1 | 27.03.2023 |  | kf\_food | 02546 | Трайфл Малина | 1.2 | 256 | 128 |
| 2 | 27.03.2023 | Понедельник | kf\_day\_week |  |  |  | 1.1 | 1.0 |
| 3 | 27.03.2023 |  | kf\_kondi |  |  |  | 1.2 | 1.1 |

* 1. Таблица **archive\_cake\_prognoz**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [PK] **id** (integer) | **data** (data) | **day\_week**  (character varying (50)) | **index**  **(**character varying (50)) | **food\_kod** (character varying (50)) | **food\_name** (character varying (50)) | **kf\_food**  (numeric) | kond\_  name1 | kond\_  name1 |
| 1 | 27.03.2023 |  | kf\_food | 03569 | Торт Наполеон | 1.2 | 256 | 128 |
| 2 | 27.03.2023 | Понедельник | kf\_day\_week |  |  |  | 1.1 | 1.0 |
| 3 | 27.03.2023 |  | kf\_kondi |  |  |  | 1.2 | 1.1 |

## Связи между таблицами

* Таблицы **users** и **users\_role** связаны по столбцу **username**
* Таблица **foods** и таблицы **bakery\_prognoz, pie\_prognoz**, **cake\_prognoz** связаны по столбцам **food\_kod** и **food\_name**

# **СТРУКТУРА ПРОЕКТА, ФАЙЛЫ, КЛАССЫ И ФУНКЦИИ**

## Структура проекта, описание файлов

* **Основная директория**
* **data**
* **images**
* **Файлы изображений, иконка приложения**
* **requests**
* **Файлы хранящие запросы к БД**
* **ui**
* **Файлы созданные в Qt Designer, скомпилированные в .py файлы**
* **windows**
* **Файлы хранящие классы окон**
* **active\_session.py – файл хранящий активную сессию пользователя**
* **signals.py – файл хранящий все созданные сигналы от UI**
* **config.py – файл конфигурации подключения к БД**
* **malina64.py – исполняемый файл**
* **.gitignore – файл описывающий игнорируемые директории и файлы**
* **README.md – описание проекта для GitHub**
* **requirements.txt – файл со списком используемых библиотек**
* **license.txt – файл с текстовой лицензией**

## Описание классов и функций

### Файл windows.windows\_authorization.py

Класс WindowAuthorization(QtWidgets.QMainWindow) – экземпляр класса Ui\_WindowAuthorization(). Хранит в себе UI отображения окна авторизации. Принимает от пользователя логин и пароль. После нажатия кнопки «Начать работу» вызывает функцию self.login.

Функция self.login. Создает локальные переменные username и password в которые сохраняет введённые пользователем данные. Создает локальную переменную login\_result в которую помещает результат выполнения функции self.database.login(username, password). Если вернувшийся результат типа кортеж(tuple) и он хранит 2 значения (len(login\_result) == 2), тогда создаем 2 переменных. Переменная result принимает в себя первое значение вернувшегося кортежа, а переменная role принимает второе значение вернувшегося кортежа. Если в переменной result пришло сообщение «Авторизация успешна», тогда в файл **active\_session.py** в класс session сохраняем переменную role(хранящую права пользователя) через функцию self.session.set\_role(role).