Итоговый проект

Приложение для учёта расходных материалов в учебном заведении/офисе/ мастерской

Подготовил: Гамиль Диас
Группа: ПО2511
Преподаватель: Баймаканов А. Т

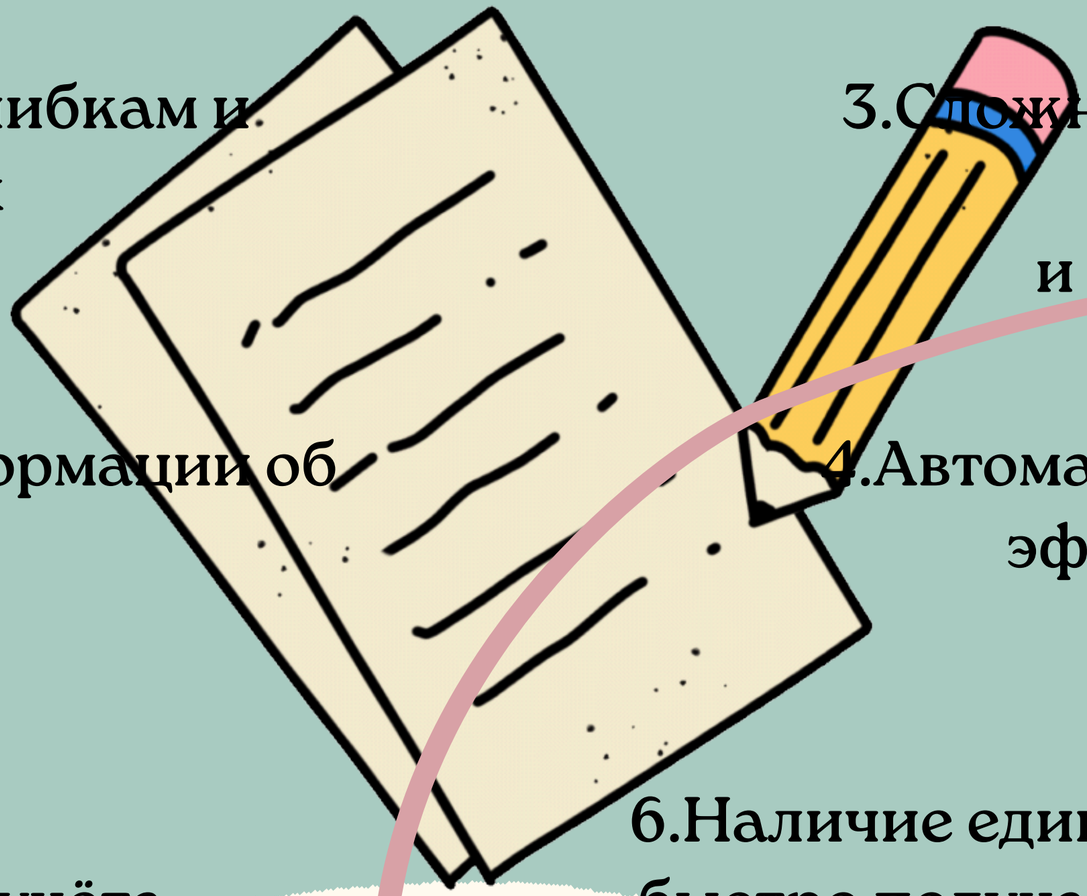
План

1.Актуальность проекта
2.Введение
3.Цель проекта
4.ЦА
5. Реализация проекта
6. Практическая часть
7.Заключение

Актуальность проекта

1.Ручной учёт ведёт к ошибкам и потерям данных

2.Нет актуальной информации об остатках



3.Сложно контролировать списание и пополнение

4.Автоматизация повышает эффективность

5.Автоматизация учёта уменьшает количество ошибок

6.Наличие единой системы позволяет быстро получать актуальные данные о материалах в любой момент

Введение

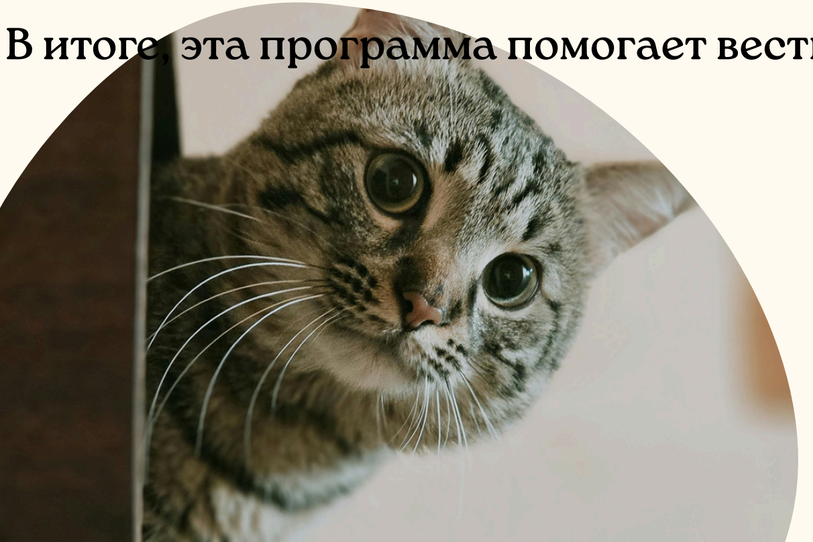
Каждый день любой компании нужны разные штуки: ручки, бумажки, всякие инструменты, и т.п. Важно следить за всем этим, чтобы всё работало как надо, тратить деньги с умом и не покупать лишнего.

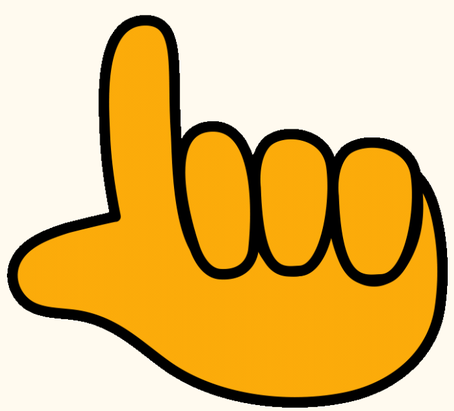
Если записывать всё в тетрадках или таблицах, легко запутаться, что-то потерять или не знать, сколько чего осталось. А если использовать компьютерную программу, то следить за материалами станет проще, ошибок будет меньше и всё будет работать быстрее.

Мы сделали приложение для компьютера, чтобы учитывать все эти расходники. В ней можно добавлять, изменять, списывать и смотреть, сколько чего осталось.

Для хранения информации мы используем PostgreSQL(pgadmin), а само приложение сделано на Python с Tkinter.

В итоге, эта программа помогает вести учёт, почти не допускать ошибок и лучше распоряжаться ресурсами компании.





ЦЕЛИ

Наша первая цель

Создать приложение для учёта расходных материалов

Наша вторая цель

хранение данных о категориях и материалах
контроль остатков

Наша третья цель

- операции списания и пополнения
- удобный интерфейс

Целевая аудитория

Целевая аудитория проекта:

- Учебные заведения (школы, колледжи, центры обучения).
- Офисы и организации, где ведётся учёт расходных материалов.
- Мастерские и лаборатории, использующие материалы и инструменты.

проблема: во многих организациях нет удобного инструмента для контроля остатков.

Решение: наш проект обеспечивает быстрый учёт и контроль материалов.

Реализация проекта

```sql
CREATE TABLE categories (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL
);
CREATE TABLE materials (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    unit VARCHAR(50) NOT NULL,
    quantity INTEGER DEFAULT 0,
    min_quantity INTEGER DEFAULT 0,
    category_id INTEGER REFERENCES categories(id) ON DELETE SET NULL
);

CREATE TABLE transactions (
    id SERIAL PRIMARY KEY,
    material_id INTEGER REFERENCES materials(id) ON DELETE CASCADE,
    type VARCHAR(10) CHECK (type IN ('приход', 'расход')),
    amount INTEGER NOT NULL,
    date DATE DEFAULT CURRENT_DATE,
    comment TEXT
);
```

```python
decrease_material(material_id, amount):
conn = get_connection()
cur = conn.cursor()

cur.execute("SELECT quantity FROM materials WHERE id = %s",
current_qty = cur.fetchone()[0]

if current_qty < amount:
    cur.close()
    conn.close()
    raise ValueError("Недостаточно материалов на складе!")

cur.execute("""
    UPDATE materials
    SET quantity = quantity - %s
    WHERE id = %s
""", (amount, material_id))

cur.execute("""
    INSERT INTO transactions (material_id, type, amount, co
    VALUES (%s, 'расход', %s, 'Операция через программу')
""", (material_id, amount))
```

Создание таблиц в pg и реализация кода в python

```python
ef refresh_table():
    """Обновление таблицы материалов"""
    for row in tree.get_children():
        tree.delete(row)

    for row in load_materials():
        tree.insert("", tk.END, values=row)


ef add_category_window():
    win = tk.Toplevel(root)
    win.title("Добавить категорию")

    tk.Label(win, text="Название категории:").pack()
    entry = tk.Entry(win)
    entry.pack()

    def save():
        add_category(entry.get())
        messagebox.showinfo("Успех", "Категория добавлена!")
        win.destroy()

    tk.Button(win, text="Сохранить", command=save).pack()


ef add_material_window():
    win = tk.Toplevel(root)
    win.title("Добавить материал")

    labels = ["Название", "Единица (шт, кг)", "Количество", "Мин. остаток", "ID
    entries = []
```

продолжение реализации проекта

```
        ):
        ение таблицы материалов"""
    w in tree.get_children():
        tree.delete(row)

for row in load_materials():
    tree.insert("", tk.END, values=row)


def add_category_window():
    win = tk.Toplevel(root)
    win.title("Добавить категорию")

    tk.Label(win, text="Название категории:").pack()
    entry = tk.Entry(win)
    entry.pack()

    def save():
        add_category(entry.get())
        messagebox.showinfo("Успех", "Категория добав
        in.destroy()

    text="Сохранить", comman
```

```
 = tk.Tk()
ot.title("Учёт расходных материалов")
ot.geometry("900x600")

ee = ttk.Treeview(
  root,
  columns=("ID","Название","Ед.","Количество","Категория"),
  show="headings"


 col in ("ID","Название","Ед.","Количество","Категория"):
  tree.heading(col, text=col)

ee.pack(fill=tk.BOTH, expand=True)

fresh_table()

ame = tk.Frame(root)
ame.pack()

.Button(frame, text="Добавить категорию", command=add_category_window).
.Button(frame, text="Добавить материал", command=add_material_window).g
.Button(frame, text="Приход", command=lambda: change_quantity_window("
.Button(frame, text="Расход", command=lambda: change_quantity_window("
.Button(frame, text="История операций", command=open_transactions_windo
.Button(frame, text="Обновить", command=refresh_table).grid(row=0, colu

t.mainloop()
```

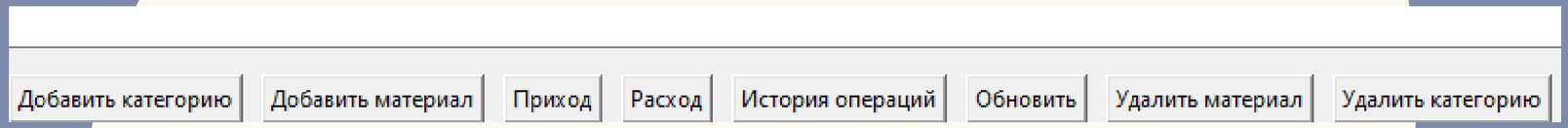
```
INTO categories (name) VALUES
 Канцелярия'),
 'Хозяйственные товары'),
('Компьютерные комплектующие'),
('Учебные материалы'),
('Расходные материалы для мастерской');

INSERT INTO materials (name, unit, quantity, min_quanti
('Бумага А4', 'пачка', 20, 5, 1),
('Ручка синяя', 'шт', 150, 30, 1),
('Маркер перманентный', 'шт', 40, 10, 1),

('Салфетки влажные', 'упаковка', 25, 5, 2),
('Мыло жидкое', 'литр', 10, 3, 2),

 'Кабель HDMI', 'шт', 12, 3, 3),
'Мышь компьютерная', 'шт', 18, 5, 3),

'Тетрадь 12 листов', 'шт', 200, 50, 4),
'Обложка для тетрадей', 'шт', 300, 100, 4),

Краска акриловая', 'банка', 35, 10, 5),
 арная', 'шт', 50, 20, 5);
```

ПРАКТИЧЕСКАЯ ЧАСТЬ

Содержимое:

- Само приложение, на скриншоте сами данные добавленные
- Скриншоты кода backend и frontend
- В нижней части приложения функцианальные кнопки

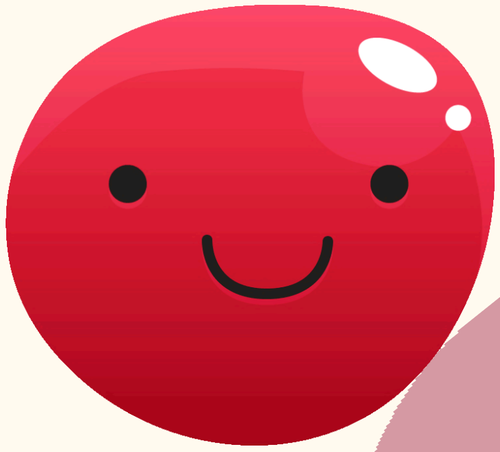

```
    tree.heading(col, text=col)
    tree.column(col, anchor='w')

tree.pack(fill=tk.BOTH, expand=True, padx=8, pady=8)

refresh_table()

frame = tk.Frame(root)
frame.pack(pady=6)

tk.Button(frame, text="Добавить категорию", command=a
tk.Button(frame, text="Добавить материал", command=ad
tk.Button(frame, text="Приход", command=lambda: chang
tk.Button(frame, text="Расход", command=lambda: chan
tk.Button(frame, text="История операций", command=op
tk.Button(frame, text="Обновить", command=refresh_ta
tk.Button(frame, text="Удалить материал", command=de
tk.Button(frame, text="Удалить категорию", command=
```




```
("Добавить материал")
("400x320")

= ["Название", "Единица (шт, кг)", "Количество", "Мин. остаток", "Категори
= []

(win, text=labels[0]).pack(anchor='w', padx=10, pady=(10,0))
= tk.Entry(win); e_name.pack(fill='x', padx=10)
.append(e_name)

(win, text=labels[1]).pack(anchor='w', padx=10, pady=(8,0))
= tk.Entry(win); e_unit.pack(fill='x', padx=10)
.append(e_unit)

(win, text=labels[2]).pack(anchor='w', padx=10, pady=(8,0))
= tk.Entry(win); e_qty.pack(fill='x', padx=10)
.append(e_qty)

(win, text=labels[3]).pack(anchor='w', padx=10, pady=(8,0))
= tk.Entry(win); e_min.pack(fill='x', padx=10)
.append(e_min)

(win, text=labels[4]).pack(anchor='w', padx=10, pady=(8,0))
obox с именами категорий
combobox_category
ox_category = ttk.Combobox(win, values=category_names, state="readonly")
ox_category.pack(fill='x', padx=10)
egory_names:
bobox_category.current(0)
```

```
f decrease_material(material_id, amount):
  conn = get_connection()
  cur = conn.cursor()

  cur.execute("SELECT quantity FROM materials WHERE id = %s
  current_qty = cur.fetchone()[0]

  if current_qty < amount:
      cur.close()
      conn.close()
      raise ValueError("Недостаточно материалов на складе!"

  cur.execute("""
      UPDATE materials
      SET quantity = quantity - %s
      WHERE id = %s
  """, (amount, material_id))

  cur.execute("""
      INSERT INTO transactions (material_id, type, amount,
      VALUES (%s, 'расход', %s, 'Операция через программу')
  """, (material_id, amount))
```

Заключение



Наш первый вывод

Создано рабочее приложение для учёта расходных материалов.

Наш второй вывод

Реализована удобная и понятная структура БД.

Наш третий вывод

Интерфейс позволяет быстро выполнять все операции.