

1. Тип прачечной

- Радиокнопки: Своя / Удалённая.
 - Если "Своя" → отобразить:
 - Стоимость электроэнергии (₽/кВт·ч).
 - Стоимость воды (₽/литр).
 - Стоимость химии (₽/цикл).
 - Амортизация оборудования (₽/цикл).
 - Температура стирки для хлопка (°C).
 - Температура стирки для гибрида (°C).
 - % экономии энергии (автозаполнение на основе разницы температур, но можно редактировать).
 - % экономии химии.
 - Если "Удалённая" → отобразить:
 - Стоимость стирки (₽/кг).
-

2. Параметры отеля

- Количество номеров (число).
 - Заполняемость (% в месяц/год).
 - Интервал стирки:
 - Выпадающий список:
 - Ежедневно.
 - Каждые 2 дня.
 - Каждые 3 дня.
 - Каждые 4 дня.
-

3. Параметры полотенец

Параметр	Хлопок	Гибрид
Стоимость комплекта (₽)	<input>	<input>
Вес комплекта (кг)	<input>	<input>
Стирок до износа	<input>	<input>

Формулы для расчётов

1. Срок службы полотенец

Срок службы (лет)=Стирок в годСтирок до износа

Стирок в год:

Стирок/год=Интервал стиркиНомеров×Заполняемость×365

2. Затраты на стирку

- **Для своей прачечной:** $\text{Стоимость цикла} = \text{Энергия} + \text{Вода} + \text{Химия} + \text{Амортизация}$
 - Энергия для гибрида: $\text{Энергия} = \text{Стоимость кВт}\cdot\text{ч} \times (1 - 100\% \text{ экономии})$
 - **Для удалённой прачечной:** $\text{Стоимость стирки} = \text{Вес комплекта} \times \text{Р/кг}$
-

3. Окупаемость в днях

$\text{Окупаемость} = (\text{Годовая экономия} - \text{Доп. затраты}) \times 365$

- **Доп. затраты:** $(\text{Стоимость гибрида} - \text{Стоимость хлопка}) \times \text{Количество номеров}$
 - **Годовая экономия:** $\text{Срок службы гибрида} - \text{Суммарная экономия за срок службы гибрида}$
-

Пример реализации на Python

```
def calculate_payback(inputs):
    # Рассчитать стирок в год
    cycles_per_year = (inputs["rooms"] * inputs["occupancy"] * 365) /
inputs["wash_interval"]

    # Срок службы
    cotton_lifespan = inputs["cotton_cycles"] / cycles_per_year
    hybrid_lifespan = inputs["hybrid_cycles"] / cycles_per_year

    # Затраты на стирку
    if inputs["laundry_type"] == "own":
        # Для своей прачечной
        cotton_energy = inputs["energy_cost"] * (inputs["cotton_temp"] / 100) #
Пример: 90°C → 90% затрат
        hybrid_energy = inputs["energy_cost"] * (1 -
inputs["energy_saving"]/100)
        # ... аналогично для воды и химии
    else:
        # Для удалённой
        cotton_cost = inputs["remote_cost_kg"] * inputs["cotton_weight"]
        hybrid_cost = inputs["remote_cost_kg"] * inputs["hybrid_weight"]

    # Окупаемость
    additional_cost = (inputs["hybrid_cost"] - inputs["cotton_cost"]) *
inputs["rooms"]
    annual_saving = (total_cotton - total_hybrid) / hybrid_lifespan
    payback_days = (additional_cost / annual_saving) * 365

    return payback_days
```

Выводы для программирования

1. Интерфейс:

- Используйте `conditional rendering` для отображения полей в зависимости от типа прачечной.
- Добавьте подсказки:

- "Срок службы вычисляется автоматически на основе износостойкости и стирок в год".
- "Процент экономии энергии рассчитывается как разница между температурами стирки".

2. Валидация:

- Проверка, что стирок до износа > 0 .
- Интервал стирки ≥ 1 .

3. UI-фичи:

- График сравнения затрат.
- Кнопка "Пример расчёта" с предзаполненными данными.

Итоговый результат

Пользователь получит:

- Срок окупаемости в днях.
- Сравнение затрат за срок службы гибрида.
- Рекомендацию:

Если окупаемость < 30 дней \rightarrow "Гибрид окупится менее чем за месяц. Срочно внедряйте!"

На выходе должно быть:

Учёт экономии для своей и удалённой прачечной

В калькуляторе реализовано сравнение гибрида с 100% хлопком по трём ключевым параметрам:

1. Для своей прачечной

Параметр	Хлопок (90°C)	Гибрид (50°C)	Экономия
Энергия/цикл	15 ₽	10 ₽	5 ₽ (-33%)
Химия/цикл	12 ₽	8.4 ₽	3.6 ₽ (-30%)
Замена полотенец	3 раза за 4.44 г.	1 раз за 4.44 г.	180,000 ₽

2. Для удалённой прачечной

Параметр	Хлопок	Гибрид	Экономия
Стоимость стирки/кг	50 ₽/кг	50 ₽/кг	0 ₽
Замена полотенец	3 раза за 4.44 г.	1 раз за 4.44 г.	180,000 ₽

Формула суммарной экономии:

$$\text{annual_saving_remote} = (\text{cotton_cost_per_kg} - \text{hybrid_cost_per_kg}) * \text{total_kg} + \text{towel_replacement_saving}$$

Как это работает в коде

```
# Для своей прачечной
if laundry_type == "own":
    # Экономия за цикл
    energy_saving = cotton_energy_cost - hybrid_energy_cost
    chemical_saving = cotton_chemical_cost - hybrid_chemical_cost
    # Экономия на замене полотенец
    towel_saving = (cotton_replacements - hybrid_replacements) * towel_cost
    # Итог
    annual_saving = (energy_saving + chemical_saving) * cycles_per_year +
towel_saving

# Для удалённой прачечной
else:
    # Экономия только на замене полотенец
    annual_saving = towel_saving
```

Реализуем пошаговую визуализацию экономии с промежуточными итогами. Пример реализации:

Динамическая визуализация экономии (UI/UX)

```
# Пример данных после расчётов
savings_steps = [
    {"name": "Экономия на энергии", "value": 150000, "desc": "Снижение температуры стирки с 90°C до 50°C"},
    {"name": "Экономия на химии", "value": 108000, "desc": "-30% расход химикатов"},
    {"name": "Сокращение замен", "value": 432000, "desc": "В 3 раза дольше срок службы"}
]

total_saving = 0
for step in savings_steps:
    total_saving += step["value"]
    # Вывод в интерфейс:
    print(f'''
<div class="saving-step">
    <div class="progress-bar" style="width: {(step["value"]/690000)*100}%">
        <span>+{step["value"]:,} ₺</span>
    </div>
    <p>{step["name"]}: {step["desc"]}</p>
    <div class="current-total">
        Итого накоплено: {total_saving:,} ₺
    </div>
</div>
''')
```

Как это будет выглядеть для пользователя

1. Шаг 1: Экономия на энергии

![[=====]] +150 000 ₽

Итого накоплено: 150 000 ₽

2. Шаг 2: Экономия на химии

![[=====]] +108 000 ₽

Итого накоплено: 258 000 ₽

3. Шаг 3: Сокращение замен

![[=====]] +432 000 ₽

Итого накоплено: 690 000 ₽

Формулы для каждого этапа

1. Энергия (годовая):

$$\text{energy_saving} = (\text{base_energy_cost} - \text{hybrid_energy_cost}) * \text{cycles_per_year}$$

2. Химия (годовая):

$$\text{chemical_saving} = (\text{base_chemical_cost} * (\text{chemical_saving_percent}/100)) * \text{cycles_per_year}$$

3. Замена полотенец (за срок службы гибрида):

$$\text{replacement_saving} = (\text{cotton_sets_needed} - \text{hybrid_sets_needed}) * \text{cotton_set_cost}$$

Адаптивный вывод для разных сценариев

Своя прачечная

```
{
  "savings_breakdown": [
    {"type": "energy", "annual": 150000, "cumulative": 150000},
    {"type": "chemical", "annual": 108000, "cumulative": 258000},
    {"type": "replacement", "total": 432000, "cumulative": 690000}
  ]
}
```

Удалённая прачечная

```
{
  "savings_breakdown": [
    {"type": "replacement", "total": 180000, "cumulative": 180000}
  ]
}
```

Рекомендации по анимации

1. Прогресс-бар:

- Заполнение с анимацией 0% → 100% за 1 сек для каждого шага
- Цвета:
 - Энергия: оранжевый → #FF6B00
 - Химия: синий → #0066CC
 - Замены: зелёный → #00C853

2. Текст:

- Постепенное появление с эффектом fade-in
- Числа с анимацией счётчика (0 → N)

3. Итоговая строка:

```
<div class="final-summary" style="border-left: 4px solid #00C853; padding-left: 1rem;">
  Общая экономия за {years} лет: <strong>690,000 ₪</strong>
</div>
```

Логика для фронтенда

```
// Пример на React
function SavingsVisualizer({ steps }) {
  const [currentStep, setCurrentStep] = useState(0);
  const total = steps.reduce((sum, step) => sum + step.value, 0);

  useEffect(() => {
    const timer = setInterval(() => {
      if (currentStep < steps.length - 1) {
        setCurrentStep(prev => prev + 1);
      }
    }, 1500);
    return () => clearInterval(timer);
  }, [currentStep]);

  return (
    <div>
      {steps.map((step, index) => (
        <div key={step.name} className={`step ${index <= currentStep ? 'visible' : ''}`}>
          <ProgressBar
            width={(step.value / total) * 100}
            label={`+${formatNumber(step.value)} ₪`}
          />
          <p>{step.name}: {step.desc}</p>
          {index === currentStep && (
            <div className="current-sum">
              Накоплено: {formatNumber(steps.slice(0, index+1).reduce((sum, s)
=> sum + s.value, 0))} ₪
            </div>
          )}
        </div>
      )}
    </div>
  );
}
```

Итог

Пользователь увидит, как экономия «накапливается» поэтапно:

1. Сначала быстрый рост за счёт энергии/химии
2. Затем резкий скачок от экономии на заменах
3. Итоговая сумма с визуальным акцентом

Это делает расчёты прозрачными и усилит доверие к рекомендациям калькулятора.