В \*\*Кабинет поставщика\*\* можно добавить несколько функциональных блоков и улучшений, чтобы сделать интерфейс более удобным и информативным. Вот предложения:

### \*\*1. Детализация поставок\*\*

- \*\*Просмотр содержимого поставки\*\* (по клику на ID поставки):

- Список товаров в формате таблицы (из JSON в `Deliveries.item\_list`).

- Кнопка \*\*«Отменить поставку»\*\* (если статус `pending`).

- \*\*Фильтры и сортировка\*\*:

- По статусу (`pending`, `received`, `cancelled`).

- По дате (за месяц, квартал, год).

---

### \*\*2. Статистика и аналитика\*\*

- \*\*Графики/карточки\*\*:

- Количество поставок за период.

- Среднее время обработки (от `pending` до `received`).

- Топ-5 товаров по частоте поставок.

- \*\*Уведомления\*\*:

- Если поставка в статусе `pending` дольше N дней.

---

### \*\*3. Управление профилем поставщика\*\*

- \*\*Редактирование контактных данных\*\*:

- Название компании, телефон, email (из таблицы `Suppliers`).

- \*\*Смена пароля\*\* (через форму с проверкой старого пароля).

---

### \*\*4. Дополнительные функции для формы добавления поставки\*\*

- \*\*Шаблоны товаров\*\*:

- Возможность сохранять часто используемые товары для быстрого выбора.

- \*\*Валидация данных\*\*:

- Проверка формата даты, обязательных полей (цена > 0, вес > 0).

- \*\*Предпросмотр JSON\*\* перед отправкой.

---

### \*\*5. История действий\*\*

- \*\*Лог операций\*\* (из `Inventory\_Transactions`):

- Например: \*«23.05.2025: Создана поставка #4 (5 товаров)»\*.

---

### \*\*6. Интеграция с документами\*\*

- \*\*Генерация PDF-отчета\*\* по поставке:

- С печатью списка товаров, даты, статуса.

- \*\*Экспорт данных\*\* в Excel/CSV.

---

### \*\*7. Визуальные улучшения\*\*

- \*\*Цветовая маркировка статусов\*\*:

- `pending` — желтый, `received` — зеленый, `cancelled` — красный.

- \*\*Хлебные крошки\*\* для навигации (например: \*Главная / Поставки / Новая поставка\*).

---

### \*\*Пример реализации в коде (Flask/Jinja2)\*\*

Для раздела \*\*«Детализация поставки»\*\* можно добавить маршрут:

```python

@app.route('/supplier/delivery/<int:delivery\_id>')

def delivery\_details(delivery\_id):

if session.get('role') != 'supplier':

return redirect('/login')

conn = get\_db\_connection()

cursor = conn.cursor()

cursor.execute("""

SELECT D.delivery\_id, D.delivery\_date, D.status, D.item\_list, S.supplier\_name

FROM Deliveries D

JOIN Suppliers S ON D.supplier\_id = S.supplier\_id

WHERE D.delivery\_id = ? AND S.user\_id = ?

""", (delivery\_id, session['user\_id']))

delivery = cursor.fetchone()

conn.close()

if not delivery:

return "Поставка не найдена", 404

items = json.loads(delivery.item\_list) # Парсим JSON

return render\_template('delivery\_details.html', delivery=delivery, items=items)

```

---

\*\*Что выбрать?\*\*

Зависит от задач диплома. Если нужна глубокая проработка — добавьте \*\*аналитику\*\* и \*\*логирование\*\*. Для простоты — хватит \*\*детализации поставок\*\* и \*\*шаблонов товаров\*\*.

Хочешь, помогу реализовать любой из этих пунктов в коде или описать их в теоретической части диплома? 😊