

Geekbrains

**Разработка back-end части аналитического вэб-приложения «Дашборд сети розничных магазинов»**

Программа:

Python-разработчик. Специалист

Боргуль Андрей Владимирович

Екатеринбург

2024

# Содержание

[Содержание 1](#_Toc169731030)

[Введение 3](#_Toc169731031)

[Теоретическая часть 5](#_Toc169731032)

[Глава 1. Теоретические основы разработки дашбордов 5](#_Toc169731033)

[1.1. Понятие и назначение дашбордов 5](#_Toc169731034)

[1.2. Исследование рынка 6](#_Toc169731035)

[1.3. Виды и типология дашбордов 7](#_Toc169731036)

[1.4. Обзор существующих дашбордов: плюсы и минусы 9](#_Toc169731037)

[1.5. Обзор наиболее популярных конструкторов дашбордов 12](#_Toc169731038)

[Практическая часть 18](#_Toc169731039)

[Глава 2. Поиск данных и функциональная модель дашборда. 18](#_Toc169731040)

[2.1. Разработка функциональной рабочей модели дашборда для проверки гипотез 18](#_Toc169731043)

[2.2. Контур программного обеспечения. 23](#_Toc169731044)

[2.3. Компоновка данных, схема-макет дашборда. 24](#_Toc169731045)

[2.4. Цветовая дифференциация. 25](#_Toc169731046)

[Глава 3. Применение цветовой дифференциации к показателям 26](#_Toc169731047)

[3.1. Основные показатели 26](#_Toc169731049)

[3.2. Показатели по спец. Задачам 31](#_Toc169731050)

[3.3. Показатели по трафику 32](#_Toc169731051)

[3.4. Показатели по чекам 33](#_Toc169731052)

[3.5. Показатели по конверсии 34](#_Toc169731053)

[3.6. Компоновка показателей сети магазинов в едином блоке дашборда 36](#_Toc169731054)

[3.7. Визуализация ключевых показателей в нижней части дашборда в виде диаграмм 41](#_Toc169731055)

[Глава 4. Разработка структуры и код приложения 46](#_Toc169731056)

[4.1. Структура кода 46](#_Toc169731058)

[4.2. Модели (Models) 46](#_Toc169731059)

[4.3. Представления (Views) 50](#_Toc169731060)

[4.4. Шаблоны (Html) 50](#_Toc169731061)

[4.5. Пути (Urls) 51](#_Toc169731062)

[Заключение 53](#_Toc169731063)

[Список используемой литературы 54](#_Toc169731064)

[Приложения 55](#_Toc169731065)

[Приложение №1 Контур программного обеспечения 56](#_Toc169731066)

[Приложение №2 Матрица цветовой индикации основных показателей дашборд: 57](#_Toc169731067)

[Приложение №3 Функциональный макет-схема дашборда. Свернутые показатели. 59](#_Toc169731068)

[Приложение №4 Функциональный макет-схема дашборда. Развернутые показатели. 60](#_Toc169731069)

[Приложение №5 Модели (Код). 61](#_Toc169731070)

[Приложение №6 Представления (Код). 63](#_Toc169731071)

[Приложение №7 Шаблоны и экраны (Код). 69](#_Toc169731072)

# Введение

"Успех в бизнесе часто зависит от того, научились ли вы правильно интерпретировать метрики."[[1]](#footnote-1)

**Тема проекта:** Разработка back-end части интерактивного дашборда для веб-приложения сети розничных магазинов является актуальной и востребованной темой. Дашборд представляет собой визуальное представление ключевых показателей эффективности бизнеса, которое позволяет руководству компании получать исчерпывающую информацию о состоянии дел в режиме реального времени. Создание такого инструмента аналитики является важным компонентом цифровой трансформации компании в современных рыночных условиях.

**Цель проекта:** Основной целью проекта является разработка концепции эффективного и функционального дашборда, который обеспечит руководство сети розничных магазинов всей необходимой информацией для принятия взвешенных и обоснованных управленческих решений, а также back-end части web-приложения «Дашборд показателей». Дашборд должен стать ключевым инструментом, позволяющим оперативно отслеживать ключевые показатели, анализировать тенденции и выявлять области для улучшения. Доступен в любое время, в любой разумной точке мира. Достижение этой цели позволит компании повысить эффективность операционной деятельности и укрепить свои позиции на рынке.

**Какую проблему решает:** Реализация данного проекта направлена на решение важной задачи по повышению эффективности управления сетью розничных магазинов. Дашборд будет включать в себя визуализацию данных о продажах, запасах, доходности, а также другие важные метрики, такие как трафик, конверсия, не доступные на текущий момент в одном целом, т.к. находятся на разных информационных ресурсах и не доступны для формирования единой картины необходимой для принятия обоснованных управленческих решений. Визуальная часть дашборда представлена графическим рисунком, скомпонованным в удобную для восприятия форму.

**Задачи проекта:**

Провести анализ существующих решений и лучших практик в области разработки дашбордов для ритейл-бизнеса.

Изучить и проанализировать специфику деятельности сети розничных магазинов, ключевые показатели эффективности и информационные потребности руководства.

Определить основные функциональные требования и технические характеристики разрабатываемого дашборда.

Разработать концептуальный дизайн дашборда, включая структуру, визуальные компоненты и способы визуализации данных.

Создать работоспособную концептуальную модель дашборда.

Выбрать оптимальные технологии и инструменты для реализации дашборда, включая платформы визуализации данных, фреймворки и библиотеки.

Спроектировать архитектуру системы, определить источники данных и способы их интеграции.

Разработать алгоритмы сбора, обработки и визуализации данных в соответствии с требованиями.

Реализовать back-end прототип дашборда и протестировать его на соответствие функциональным требованиям.

Разработать рекомендации по дальнейшему развитию и масштабированию дашборда в соответствии с потребностями бизнеса.

Разработка дизайна, front-end web-приложения и настройка API-функции не включены в данный проект.

Разработка будет вестись на основании таблиц с загруженными данными, которые будут обрабатываться кодом написанном на языке программирования python, с запуском тестового сервера внутри локальной сети.

Используемые инструменты: MS Excel, PyCharm, Python 3.12, Django

В рамках работы над данным проектом, участвовал в командном проекте в роли разработчик в Excel (обработка массивов информации макросами VBA), разработчик на Python (подготовка данных и передача их в back-end web-приложения), тестировщик (проверка полноты получаемой информации), также в проекте будут в последствии задействованы дизайнер, программист 1С и front-end разработчик.

# Теоретическая часть

## Глава 1. Теоретические основы разработки дашбордов

### Понятие и назначение дашбордов

"Визуализация данных — это не самоцель. Она должна помогать людям принимать лучшие решения на основе имеющейся информации"[[2]](#footnote-2)

Дашборд (от англ. "dashboard" - приборная панель) представляет собой визуальное представление ключевых показателей эффективности (KPI) бизнеса или отдельных его процессов. Дашборды служат для обобщения, анализа и наглядной визуализации важной информации, необходимой для принятия обоснованных управленческих решений.

Основное назначение дашборда - обеспечить руководство и аналитиков компании актуальной, структурированной и легко воспринимаемой информацией о текущем состоянии бизнеса. Это достигается за счет грамотного подбора и визуализации ключевых метрик, отражающих наиболее значимые аспекты деятельности.

Дашборды позволяют:

- Отслеживать ключевые показатели эффективности в режиме реального времени

- Выявлять тренды и закономерности в развитии бизнеса

- Оперативно реагировать на возникающие проблемы

- Принимать обоснованные управленческие решения

- Визуально представлять сложные данные в понятном виде

Таким образом, дашборд является важным инструментом для информационно-аналитического обеспечения процесса управления современной организацией.

### Исследование рынка

Исследование, проведенное компанией PwC "Dashboards and Visualization Report 2022"[[3]](#footnote-3) позволило структурировать их по уровню детализации предоставляемой информации - от высокоуровневых обзорных панелей до глубоких аналитических экранов. Выявленные паттерны использования интерактивных элементов позволили выделить различные категории дашбордов, ориентированных на решение конкретных управленческих задач. Полученные данные о предпочтениях пользователей в визуальном оформлении и способах навигации легли в основу рекомендаций по структурированию дашбордов для повышения их эффективности.

Основные выводы исследования:

* + 1. Дашборды делятся по типу назначения на:

- Стратегические дашборды (24% компаний)

- Тактические дашборды (43% компаний)

- Операционные дашборды (33% компаний)

* + 1. Наиболее распространенные виды визуализации:

- График/диаграмма (89% дашбордов)

- Таблица (71% дашбордов)

- Карта (32% дашбордов)

* + 1. По источнику данных для дашбордов:

- Интегрированные из нескольких систем (67% дашбордов)

- Специализированные на основе одной системы (23% дашбордов)

- Комбинированные (10% дашбордов)

* + 1. Также были выделены и обобщены ключевые преимущества эффективных дашбордов:

- Улучшение принятия управленческих решений (85% компаний)

- Повышение прозрачности бизнес-процессов (79% компаний)

- Ускорение реакции на изменения (73% компаний)

* + 1. Основные барьеры внедрения дашбордов:

- Недостаток навыков визуализации данных (51% компаний)

- Сложность интеграции данных из разных систем (47% компаний)

- Недостаточное финансирование ИТ-проектов (37% компаний)

Данное исследование дает подробную картину текущего состояния рынка дашбордов, основные тренды и ключевые факторы, влияющие на их эффективность в бизнесе.

### Виды и типология дашбордов

Существует множество различных видов и типов дашбордов, которые классифицируются по ряду признаков:

#### По охвату бизнеса:

- Корпоративные (общеорганизационные) дашборды

- Функциональные (по отдельным направлениям/подразделениям)

- Индивидуальные (персональные для конкретного руководителя)

#### По степени детализации:

- Стратегические (обобщенные, высокоуровневые KPI)

- Тактические (ключевые показатели для оперативного управления)

- Оперативные (детальные метрики для анализа и контроля)

#### По источникам данных:

- Интегрированные (из различных информационных систем)

- Специализированные (на базе одной системы)

- Комбинированные (сочетание нескольких источников)

#### По способу визуализации:

- Табличные

- Графические (диаграммы, графики, гистограммы)

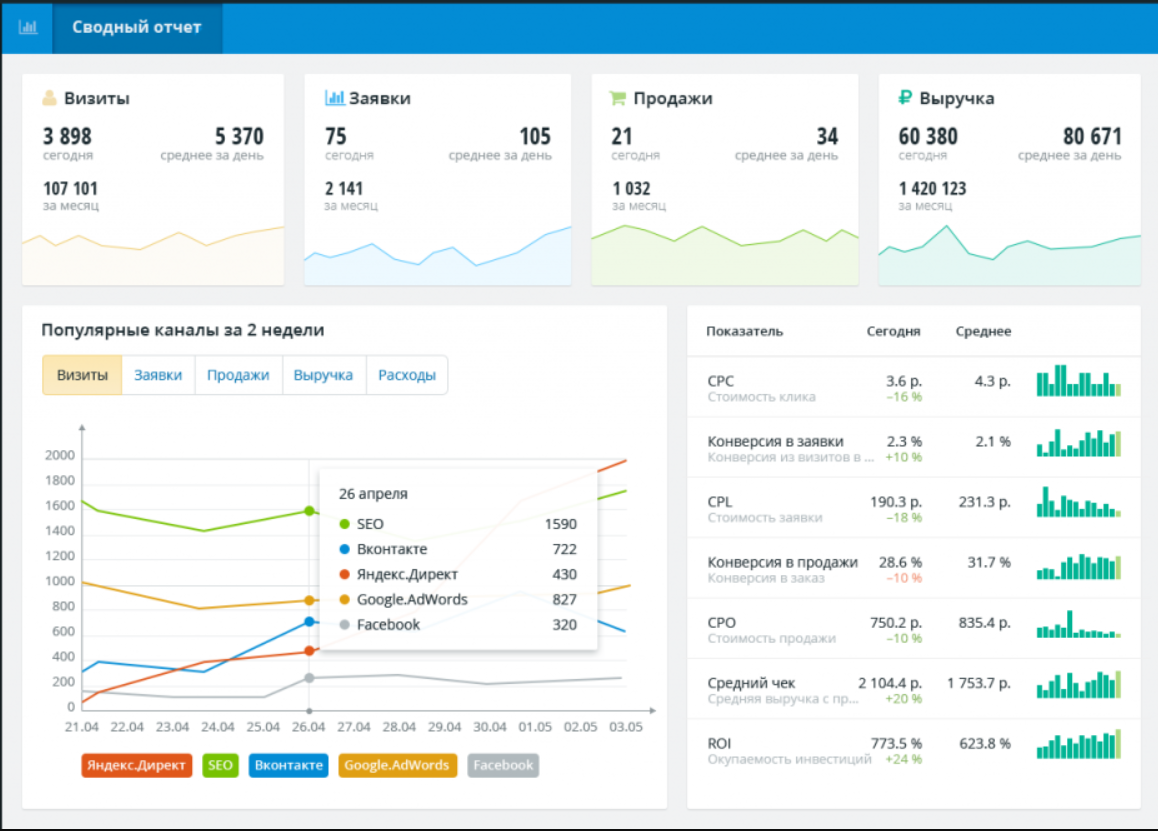
- Комбинированные (сочетание таблиц и графики)

Выбор типа дашборда определяется информационными потребностями конкретного пользователя, его уровнем управления, а также спецификой бизнес-процессов организации.

### Обзор существующих дашбордов: плюсы и минусы

Несмотря на многообразие возможных форматов дашбордов, их можно условно сгруппировать в несколько наиболее распространенных видов. Одной из ключевых классификаций является деление на обзорные, аналитические и оперативные дашборды. Обзорные панели предоставляют высокоуровневые сводки ключевых показателей, аналитические экраны позволяют глубоко погружаться в детали, а оперативные дашборды отражают текущее состояние и динамику процессов. Кроме того, дашборды могут различаться по степени интерактивности, уровню персонализации, используемым визуальным элементам и другим параметрам. Рассмотрим несколько примеров популярных дашбордов, применяемых в различных сферах бизнеса, и выделим их основные преимущества и недостатки.

Пример 1. Дашборд для интернет-магазина



Плюсы:

- Наглядная визуализация ключевых показателей продаж

- Возможность быстрого анализа динамики и трендов

- Удобный мониторинг конверсии на всех этапах воронки продаж

Минусы:

- Отсутствие данных о закупках, остатках и рентабельности

- Ограниченность функционала для принятия стратегических решений

- Недостаточная кастомизация под специфику бизнеса

Пример 2. Дашборд для финансового отдела



Плюсы:

- Всесторонний анализ финансовых потоков и показателей

- Возможность построения прогнозных моделей

- Интеграция с учетными системами

Минусы:

- Сложность в понимании и интерпретации финансовых метрик

- Перегруженность данными, затрудняющая быстрый анализ

- Недостаточная адаптивность под потребности пользователей

Пример 3. Дашборд для отдела продаж



Плюсы:

- Наглядное отображение воронки продаж и эффективности менеджеров

- Возможность детализации по клиентам, сделкам и источникам

- Удобный мониторинг KPI отдела в режиме реального времени

Минусы:

- Узкая направленность, без учета других аспектов бизнеса

- Отсутствие взаимосвязи с другими подразделениями

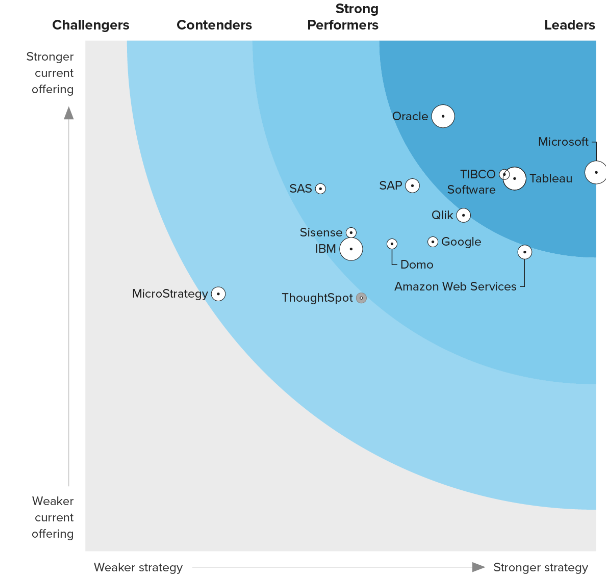
- Ограниченные возможности кастомизации и настройки

Данные примеры демонстрируют, что современный рынок предлагает действительно широкий выбор готовых дашбордных решений, покрывающих самые разные сферы бизнеса и управления. Производители аналитических панелей предлагают как универсальные шаблоны, так и отраслевые специализированные варианты, учитывающие специфику определенных предметных областей. Это позволяет организациям быстро внедрять готовые инструменты визуализации данных без необходимости разработки их с нуля.

Тем не менее, практическое применение даже самых проработанных дашбордных решений требует тщательного анализа информационных потребностей конкретной организации. Важно понять, какие именно метрики, показатели и аналитические срезы являются ключевыми для принятия управленческих решений, какая детализация данных необходима, а также оценить возможности по интеграции дашбордов с существующими информационными системами предприятия. Только на основе такого всестороннего анализа можно адаптировать готовые решения или создать максимально эффективные пользовательские панели управления.

### Обзор наиболее популярных конструкторов дашбордов

Почему бы, вместо того чтобы «изобретать велосипед» не воспользоваться готовыми решениями в виде конструкторов дашбордов, на рынке представлен широкий выбор программ-конструкторов дашбордов, которые позволяют создавать интерактивные, настраиваемые и визуально привлекательные панели мониторинга. Рассмотрим наиболее популярные решения на международном рынке согласно проведенного исследования, проведенного компанией Forrester[[4]](#footnote-4) , к наиболее популярным производителям аналитических конструкторов относятся компании: Amazon Web Services, Domo, Google, IBM, Microsoft, MicroStrategy, Oracle. По мнению Компании Forrester, рынок аналитических услуг дашбордов показан на рисунке ниже:



Рассмотрим наиболее привычные для россиян.

#### Power BI производитель Microsoft

Power BI — это набор инструментов бизнес-аналитики от Microsoft (<https://powerbi.microsoft.com/ru-ru/>), который позволяет создавать интерактивные дашборды и отчеты. Он предлагает широкий спектр визуальных элементов, возможность подключения к различным источникам данных и мощные функции аналитики. Это один из наиболее популярных и функциональных конструкторов дашбордов на рынке.



#### Tableau (https://www.tableau.com/)

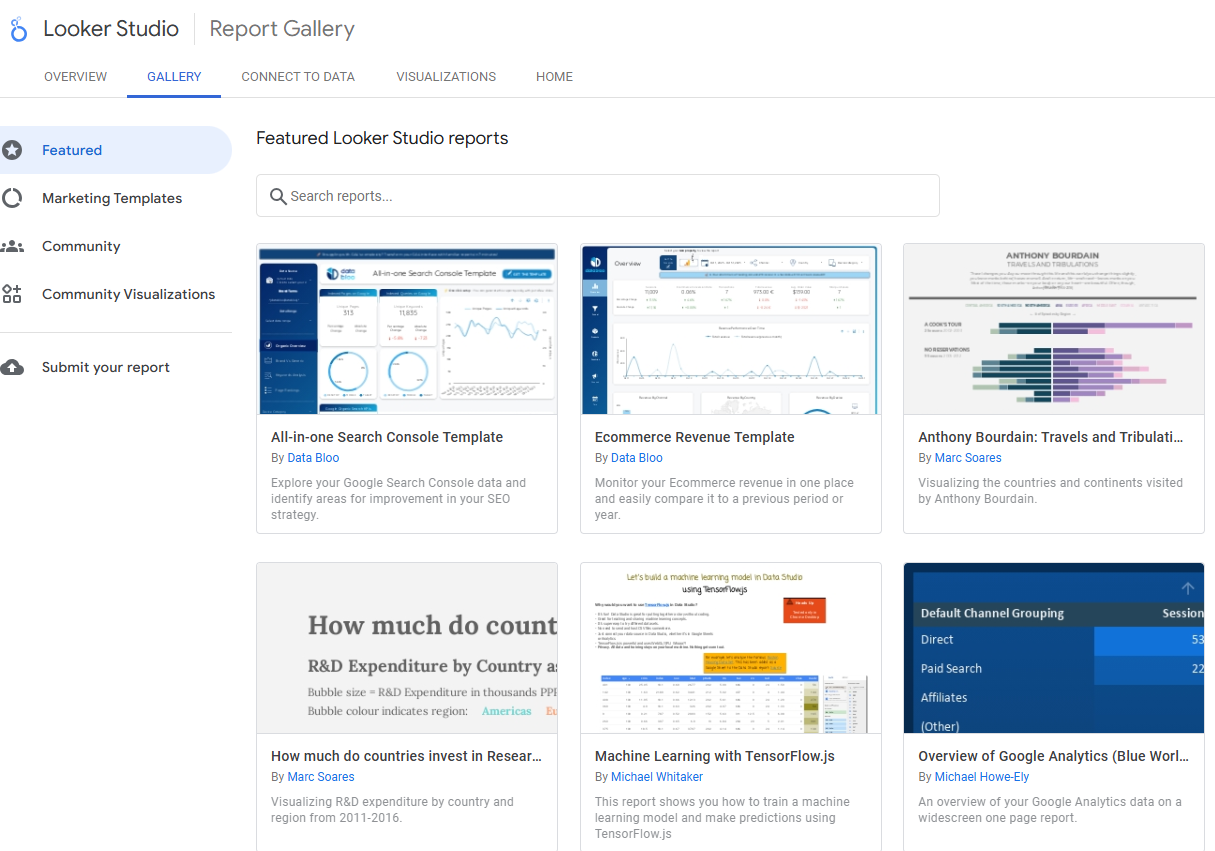
Tableau — это мощная платформа визуализации данных, которая позволяет создавать высококачественные дашборды. Она отличается интуитивным интерфейсом, большим набором визуальных элементов и возможностью подключения к множеству источников данных. Tableau известен своими передовыми возможностями анализа и визуализации. К сожалению, ввиду санкций не доступен для россиян.

#### Qlik Sense

Qlik Sense (<https://www.qlik.com/ru-ru>) — это платформа бизнес-аналитики, которая позволяет создавать интерактивные дашборды. Она отличается мощными функциями анализа данных, гибкостью и визуальной привлекательностью. Qlik Sense известен своими возможностями по обработке больших объемов данных. Также не доступен ввиду санкций.

#### Google Data Studio

Google Data Studio (<https://datastudio.google.com/overview>) - это бесплатный инструмент для создания дашбордов и отчетов, который интегрируется с другими продуктами Google, такими как Google Analytics. Он предлагает простой в использовании интерфейс и широкий выбор визуальных элементов, что делает его доступным для пользователей любого уровня.



Если коротко – отличный продукт, очень универсальный и информативный.

#### Looker от Google.

Looker (https://looker.com/)- это платформа бизнес-аналитики, которая позволяет создавать интерактивные дашборды. Разработчиком является компания Google. Она отличается продвинутыми возможностями по анализу данных, гибкостью и масштабируемостью. Looker известен своими передовыми функциями моделирования данных и интеграцией с различными источниками. Содержит мощный аналитический инструментарий на основе Искусственного Интеллекта.

Эти программы-конструкторы дашбордов представляют собой лишь несколько примеров из широкого спектра решений, доступных на рынке. Каждое из них имеет свои сильные стороны и особенности, поэтому выбор наиболее подходящего инструмента зависит от конкретных потребностей и задач вашего бизнеса. Но все они не являются бесплатными для бизнеса, кроме того, не все согласны, что данные по коммерческой деятельности предприятия, хранятся на иностранных облачных серверах. Ввиду ввода санкций из-за международной ситуации, гарантировать работоспособность подобных сервисов не представляется возможным.

Крупные предприятия, как правило, имеют сложные и уникальные требования к визуализации данных, и им часто бывает сложно найти готовое решение, которое полностью соответствует их нуждам. В таких случаях они часто предпочитают создавать собственные, кастомизированные дашборды, которые идеально подходят под их бизнес-процессы и потребности.

Основные преимущества разработки собственного дашборда для крупных предприятий:

1. Полное соответствие требованиям бизнеса

Когда вы создаете дашборд самостоятельно, вы можете точно настроить его под ваши уникальные бизнес-цели, метрики и KPI. Вы не будете ограничены возможностями готовых решений, а сможете реализовать именно то, что вам необходимо.

2. Высокая степень кастомизации

Создавая дашборд с нуля, вы получаете полный контроль над его внешним видом, структурой, функциональностью и интеграцией с вашими системами. Это позволяет добиться максимальной эффективности и удобства использования.

3. Масштабируемость и гибкость

Собственный дашборд легче адаптировать под растущие потребности бизнеса. Вы можете постепенно расширять его функциональность и интегрировать с новыми источниками данных по мере необходимости.

4. Безопасность и конфиденциальность

Разработка собственного дашборда позволяет обеспечить надежную защиту ваших критически важных данных и строго контролировать доступ к ним.

Конечно, создание собственного дашборда требует больших временных и финансовых затрат, а также наличия опытной команды разработчиков. Тем не менее, для крупных предприятий с уникальными потребностями это может быть наилучшим решением, обеспечивающим максимальную эффективность и отдачу от визуализации данных.

"Единственная непреложная истина в аналитике — это то, что все может измениться завтра."[[5]](#footnote-5)

# Практическая часть

## Глава 2. Поиск данных и функциональная модель дашборда.



### Разработка функциональной рабочей модели дашборда для проверки гипотез

"Создание эффективного дашборда — это не просто размещение множества диаграмм и сводных таблиц на одном листе. Это искусство представления ключевой информации в понятном и привлекательном формате"[[6]](#footnote-6)

При анализе рабочей группой было рассмотрены примеры дашбордов, с точки зрения визуального восприятия и сложившихся привычек восприятия информации, соответственно принята следующая структура создания дашборда, как наиболее отвечающая требованиям бизнеса:

* + - * тип дашборда – тактический
      * вид визуализации дашборда – таблица с элементами диаграмм
      * тип источника данных - комбинированный

Перед формированием типологии и дизайна дашборда, была проведена разработка функциональной рабочей модели дашборда, которая является ключевым этапом в создании аналитического веб-приложения для сети розничных магазинов.

Данная модель позволяет проверять различные гипотезы и сценарии на основе моделирования в Microsoft Excel по ключевым показателям, полученным из анализа финансовых и управленческих требований, предъявленных заказчиком к дашборду.

Для целей формирования дашборда, были выбраны следующие показатели:

#### Продажи (выручка) как план-факт, источник данных - учетная система 1С Предприятие

Анализ товарооборота в разрезе плановых и фактических показателей дает возможность оценить эффективность деятельности сети розничных магазинов, выявить причины отклонений от плана и своевременно принимать корректирующие меры. Под план-фактом подразумевается отношение прогноза выполнения плана по товарообороту текущего месяца к установленному плану по товарообороту на этот месяц. Измеряется в процентах.

#### Наценка как план-факт, источник данных - учетная система 1С Предприятие

Мониторинг уровня наценки, в том числе в сравнении с плановыми значениями, позволяет оценивать ценовую политику сети, эффективность управления ассортиментом и формирования торговой наценки. Под план-фактом подразумевается отношение прогноза выполнения плана по наценке текущего месяца к установленному плану по наценке на этот месяц. Измеряется в процентах.

#### Оборачиваемость как план-факт, источник данных - учетная система 1С Предприятие

Анализ показателей оборачиваемости товаров является важным для оптимизации складских запасов, повышения эффективности использования оборотных средств и управления ассортиментной матрицей. Оборачиваемость рассчитывается как отношение прогноза реализации за текущий месяц к среднему товарному остатку за этот же месяц, посчитанный в себестоимости. Под план-фактом подразумевается отношение оборачиваемости текущего месяца к установленному плану по оборачиваемости на этот месяц. Измеряется в процентах.

#### Остатки как план-факт, источник данных - учетная система 1С Предприятие

Мониторинг остатков товаров в разрезе плановых и фактических значений дает возможность выявлять "застойные" позиции, определять необходимость корректировки закупочной политики и управлять товарными запасами. Под план-фактом подразумевается отношение товарного остатка на текущую дату к установленному плану по товарному остатку на конец месяца. Измеряется в процентах.

Данные показатели выводятся по 13 магазинам розничной сети в разрезе товарных направлений, что позволяет проводить углубленный анализ и сравнение ключевых метрик на уровне каждой торговой точки. Дает возможность, при одновременном выводе в дашборд, не только проводить сравнение показателей работы магазинов между собой, для поиска и оценки проблем в конкретном магазине, но и дает косвенное представление об управляющем персонале каждого магазина.

Управление через товарные направления, позволяет разделить весь ассортимент товара на логические блоки, сформированные категорийными менеджерами.

#### Следующий блок показателей – показатели по специальным задачам (спец. задачам) поставленные коммерческим отделом компании.

Всего спец. задач – три, источник данных - учетная система 1С Предприятие, показатели выводятся показатели как план-факт. Под план-фактом подразумевается отношение прогноза продаж на текущий месяц, рассчитанного по накопленным продажам с начала месяца и до текущей даты к установленному плану по продажам по каждой спец. задаче. Измеряется в процентах.

#### Блок: Трафик

Трафик представляет собой количество вошедших в магазин клиентов. Данные получают через считывание датчиками на входе в магазин при входе/выходе посетителя. Данные передаются посредством сети интернет в облачную систему DATA Traffic Cloud. Анализ трафика позволяет оценивать эффективность маркетинговых акций, сезонные колебания спроса, а также выявлять проблемы с привлечением и удержанием клиентов.

Важными показателями в этом блоке являются:

##### Количество вошедших клиентов. Для дашборда используется абсолютное значение, полученное из облачной системы DATA Traffic Cloud.

##### Отношение к количеству вошедших клиентов в аналогичный период прошлого года (АППГ). Источник данных DATA Traffic Cloud, измеряется в процентах.

##### План-факт по трафику источник данных - облачная система DATA Traffic Cloud

Под план-фактом подразумевается отношение накопленного количества вошедших посетителей с начала месяца и до текущей даты к расчетному значению планового количества посетителей на исходя из планового количества посетителей пропорционально прошедшему периоду. Измеряется в процентах.

#### Блок: Чеки

Средний чек — это условная величина, рассчитываемая как общая сумма продаж, накопленная за период, поделенная на количество чеков, накопленное за тот же период. Анализ среднего чека дает возможность оценивать эффективность продаж, выявлять изменения в покупательском поведении и корректировать ассортимент и ценообразование.

Ключевые показатели в этом блоке:

##### Сумма среднего чека источник данных - учетная система 1С Предприятие. Измеряется в абсолютных цифрах. Рассчитывается как среднее значение с начала месяца.

##### Отношение к количеству чеков за аналогичный период прошлого года (АППГ), расчетный показатель. Измеряется в процентах.

##### Количество чеков РТС – общее количество чеков (свершившихся покупок) по всей розничной сети (РТС), источник данных - учетная система 1С Предприятие. Рассчитывается накопительным итогом с начала месяца и до текущей даты. Измеряется в абсолютных числах.

##### Количество чеков ИМ – общее количество чеков (свершившихся покупок), оформленных по заказам совершенных через сайт компании, источник данных - учетная система 1С Предприятие. Рассчитывается накопительным итогом с начала месяца и до текущей даты. Измеряется в абсолютных числах.

##### Доля РТС – отношение количества чеков РТС к общему количеству чеков (суммарно совершенных в магазинах и оформленных через сайт).

Этот показатель (доля РТС) позволяет отслеживать динамику изменения соотношения количества покупок, совершаемых через офлайн-магазины и онлайн-канал компании. Анализ этого соотношения дает важную информацию о том, как потребители предпочитают взаимодействовать с брендом - отдают ли они предпочтение традиционным офлайн-продажам или активно переходят в онлайн, что может указывать на необходимость корректировки маркетинговой стратегии и оптимизации распределения ресурсов между физическими и цифровыми точками продаж. Отслеживание динамики этого показателя позволит компании своевременно реагировать на изменение покупательских предпочтений. Вывод его в блок итоговых значений отнесен на следующий этап формирования дашборда.

#### Блок: Конверсия

Конверсия представляет собой отношение количества чеков, к общему числу посетителей за один и тот-же период.

Показатели в этом блоке:

##### Текущее значение конверсии, абсолютное значение, расчетный показатель

##### Отношение к конверсии в аналогичный период прошлого года (АППГ), расчётный показатель, измеряется в процентах.

##### План-факт по конверсии, расчетный показатель. Под план-фактом подразумевается отношение текущего показателя конверсии к аналогичному показателю месяца прошлого года. Измеряется в процентах.

Анализ конверсии позволяет всесторонне оценивать эффективность работы торгового персонала, оптимальность выкладки товаров, наличие необходимого покупателю ассортимента и другие важные факторы, влияющие на вероятность завершения покупки. Этот показатель дает возможность выявить "слабые места" в процессе продаж - будь то недостаточная клиентоориентированность сотрудников, неудобная планировка торгового зала или нехватка популярных позиций. Изучение динамики конверсии позволяет находить точки роста, вносить корректировки в операционные процессы и тем самым повышать общую эффективность работы точек продаж. Регулярный мониторинг этого показателя становится важным инструментом оптимизации бизнес-процессов и улучшения покупательского опыта.

### Контур программного обеспечения.

Контур программного обеспечения — это базовая архитектурная структура, которая определяет основные компоненты и их взаимосвязи в программной системе. Он служит фундаментом, на котором строится вся программа, задавая её общую логику и организацию.

Основная цель контура - обеспечить целостность, масштабируемость и управляемость программного продукта. Он разделяет систему на логические модули, каждый из которых отвечает за свою часть функционала. Это упрощает разработку, тестирование и сопровождение кода, позволяя командам работать параллельно над разными компонентами.

Контур также определяет границы между различными слоями приложения - представлением, бизнес-логикой и данными. Это помогает избежать тесной связанности компонентов и обеспечивает гибкость системы. Изменения в одном модуле не будут затрагивать другие, что сокращает риски при доработках и масштабировании.

Так как чётко определённый контур ПО является ключевым элементом архитектуры любой сложной информационной системы, в рамках данного проекта после проведенных встреч совместно с инженерами отдела ИТ был сформирован обобщенный контур программного обеспечения, наглядно показывающий взаимосвязь необходимых данных и требуемые дополнительные ресурсы по настройке путей предоставления данных, например необходимость настроить получение данных от стороннего сервиса DATA Traffic Cloud через API

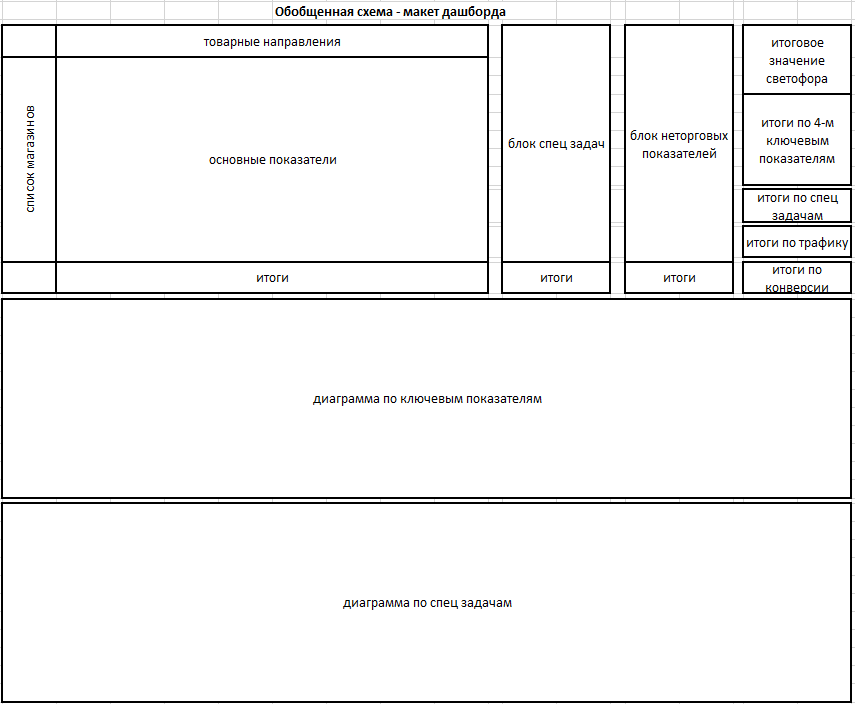
Общая схема Контура программного обеспечения представлена в [Приложении №1](#_Приложение_№1_Контур)

### Компоновка данных, схема-макет дашборда.

После определения ключевых данных для дашборда и способов их получения, стало очевидно, как лучше скомпоновать их на экране, чтобы информация была наглядной и легко воспринималась. Детально обсудив различные варианты визуального представления, команда остановилась на наиболее оптимальной форме, которая позволит пользователям быстро и эффективно анализировать ключевые показатели. Продуманное расположение метрик, наглядная визуализация трендов и взаимосвязей между данными – эти аспекты легли в основу финальной компоновки дашборда.

Итогом стала схема-макет размещения информации на экране Рисунок №1:

Рисунок №1



### Цветовая дифференциация.

В рамках проекта была принята цветовая дифференциация каждого показателей в виде его цветовой заливке по классической схеме «красный - желтый – зеленый» т.к. цветовая индикация «красный-желтый-зеленый» интуитивно понятна пользователям, так как ассоциируется с общепринятой системой сигнальных цветов. Это позволяет быстро оценить состояние показателя и предпринять соответствующие действия. Данная цветовая схема обеспечивает эффективную визуализацию информации, упрощая её восприятие и анализ.

## Глава 3. Применение цветовой дифференциации к показателям

"Успех в бизнесе часто зависит от того, научились ли вы правильно интерпретировать метрики."[[7]](#footnote-7)



### Основные показатели

#### Цветовая дифференциация основных показателей

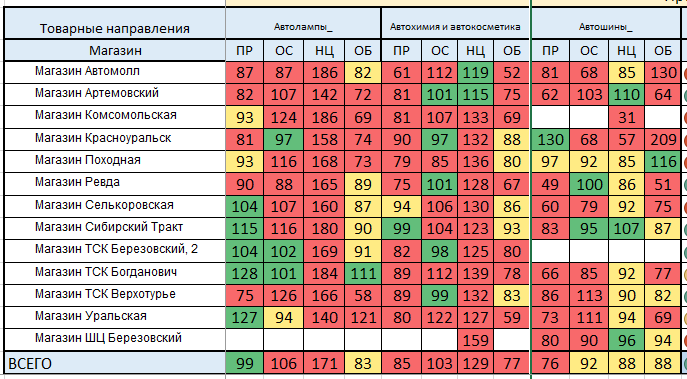
Для каждого показателя, было определено отдельное значение показателя, при котором происходит смена его цветовой заливки (см. таблицу №1):

Таблица №1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Показатель / Цвет заливки | Краткое обозначение в дашборде | красный | желтый | зеленый |
| План-факт по продажам | ПР | <93 | 93-98 | >98 |
| План-факт по остаткам | ОС | <90>102 | 90-95 | > =95-102<= |
| План-факт по наценке | НЦ | <80> =120 | 80-95 | 95-120 |
| План-факт по оборачиваемости | ОБ | <80> =120 | 80-95 | 95-120 |

В итоге получили табличную часть дашборда с цветовой заливкой показателей (рисунок №2)

Рисунок №2.



Чтобы не перегружать дашборд обилием информации и тем самым уменьшить эффект «рассеивания» внимания, но оставить возможность просмотра итога каждого показателя, было принято решение добавить функционал «свертывания колонок» таким образом, чтобы оставался один ключевой показатель - План-факт по продажам.

Свернутые показатели для улучшения визуализации, оставляющие только главный – основной показатель план-факт по продажам изображены на рисунке №3:

Рисунок №3



Для еще большего повышения наглядности и удобства восприятия было решено сгруппировать четыре ключевых показателя в компактный блок с цветовым кодированием по принципу "светофора". Это позволяет быстро оценить общее состояние по данным метрикам, акцентируя внимание на критических областях, обозначенных красным цветом. Такая визуализация позволяет легко отслеживать динамику и вовремя принимать корректирующие меры по улучшению ситуации.

При этом в обсуждении проектной командой, было обнаружено, что разные показатели имеют разную первоочередность по влиянию общий ход продаж, и требуют разной скорости принятия управленческих воздействий, отсюда возникла необходимость «взвесить» значимость (критичность) показателей в итоговом выводе цвета.

#### Разработка модели весов параметров для вывода в "светофор" дашборда итогового показателя по каждому магазину

Каждый показатель: план-факт по продажам, план-факт по остаткам, план-факт по наценке, план-факт по оборачиваемости, был исследован на его значимость (первоочерёдность для управленческих решений), и каждому был присвоен ВЕС. (см. таблицу №2)

Таблица№2

|  |  |  |
| --- | --- | --- |
| Показатель | Значимость для принятия оперативного управленческого решения | ВЕС показателя |
| План-факт по продажам | важный/срочный | 3 |
| План-факт по остаткам | не важный/срочный | 1 |
| План-факт по наценке | важный/не срочный | 2 |
| План-факт по оборачиваемости | не важный/не срочный | 0,5 |

Для перевода цветовой индикации в математическое описание, каждому цвету показателя было присвоено цифровое обозначение:

* Зеленый = 1
* Желтый = 2
* Красный = 3

Таким образом была получена матрица значений итогового цвета «светофора» (цветовой заливки обобщающей ячейки).

Формула расчета параметров с учетом их веса выглядит следующим образом:

ОБОБ = (ПР\*ВЕСпр + ОС\*ВЕСос + НЦ\*ВЕСнц + ОБ\*ВЕСоб)/4, где:

ОБОБ – значение обобщенного показателя, которое соответствует одному из цветов: зеленый, желтый, красный, черный.

ПР – значение показателя план-факт по продажам,

ВЕСпр – значение веса показателя ПР

ОС – значение показателя план-факт по остатку

ВЕСос - значение веса показателя ОС

НЦ – значение показателя план-факт по наценке,

ВЕСнц – значение веса показателя НЦ

ОБ – значение показателя план-факт по оборачиваемости

ВЕСоб - значение веса показателя ОБ

На основании рассчитанного веса параметра выводится цветовая индикация, приведенная в таблице №3:

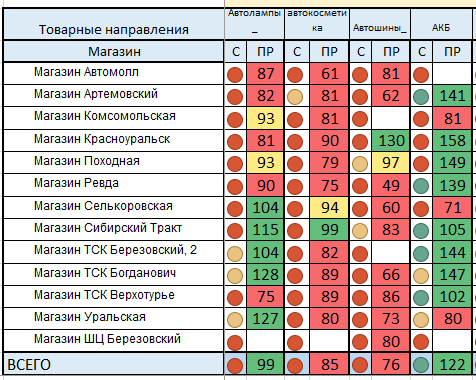
Таблица №3

|  |  |  |
| --- | --- | --- |
| Значение ОБОБ | Цвет «Светофора» в дашборде (С) | Смысловое значение |
| ОБОБ <= 2 | зеленый | Все хорошо, вмешательства не требуется |
| 2> ОБОБ <= 3 | желтый | Не все показатели хороши, но еще в пределах нормы |
| 3> ОБОБ <= 4 | красный | Плохое состояние в показателях, требуется анализ проблемы и срочное управленческое воздействие |
| ОБОБ > 4 | черный | Требуется немедленное вмешательство, один или несколько показателей имеют критическое значение для бизнеса. |

Матрица значений обобщенного показателя для светофора (ОБОБ) приведена в [Приложении №2.](#_Приложение_№2_Матрица)

В итоге информация в дашборд должна выводится в виде, показанном на рисунке №4

Рисунок №4



### Показатели по спец. Задачам

Как было описано выше, специальные задачи, поставленные коммерческим отделом, могут включать продвижение новых товарных категорий, увеличение доли продаж премиальных брендов или расширение присутствия в определенных каналах сбыта. Эти нестандартные цели требуют особого подхода и координации усилий всей команды продаж. Достижение таких задач зачастую связано с необходимостью перестройки бизнес-процессов, изменения мотивации персонала и внедрения эффективных маркетинговых инструментов. Именно поэтому показатели по спец. задачам вынесены в дашборд.

Всего спец. задач – три (их условные наименования: LEMARC2, ТОСОЛ-СИНТЕЗ, RELIT), имеют одинаковый вес. Смена цвета показателя приведена в таблице №4

Таблица №4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Спец.задачи |  | | | |
| цвет заливки | красный | желтый | зеленый | вес |
| код цвета | 3 | 2 | 1 |  |
| LEMARC2 | <90 | 90-99,99 | >100 | 1 |
| ТОСОЛ-СИНТЕЗ | <90 | 90-99,99 | >100 | 1 |
| RELIT | <90 | 90-99,99 | >100 | 1 |

Вывод в дашборд отдельного поля со «светофором» не требуется, планируемый макет представлен на рисунке № 5

Рисунок №5



### Показатели по трафику

Для блока по трафику предусмотрены цветовые заливки только для показателя АППГ и ПФ, абсолютное значение – Количество, не требует цветовой дифференциации. Данные по зависимости цвета заливки от самого показателя представлены в таблице №5

Таблица №5.

|  |  |  |  |
| --- | --- | --- | --- |
| трафик |  | | |
| цвет заливки | красный | желтый | зеленый |
| код цвета | 3 | 2 | 1 |
| АППГ | <98 | 98-99,99 | >100 |
| Количество | нет цветовой дифференциации | | |
| ПФ | <98 | 98-99,99 | >100 |

Вывод в дашборд отдельного поля со «светофором» не требуется, планируемый макет представлен на рисунке № 6

Рисунок №6.



Итоговые значения АППГ и ПФ также имеют цветовую дифференциации согласно итоговому значению. Смена цветовой схемы происходит аналогично указанной в таблице №5.

### Показатели по чекам

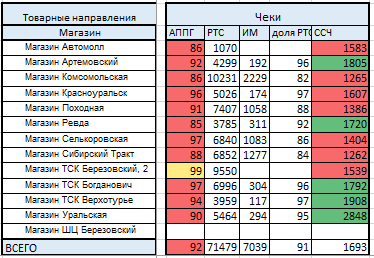
Для блока по чекам предусмотрены цветовые заливки только для показателя АППГ и ССЧ (Средняя Сумма Чека), абсолютное значение – РТС, ИМ, Доля РТС, не требует цветовой дифференциации. Данные по зависимости цвета заливки от самого показателя представлены в таблице №6

Таблица №6

|  |  |  |  |
| --- | --- | --- | --- |
| чеки |  | | |
| цвет заливки | красный | желтый | зеленый |
| код цвета | 3 | 2 | 1 |
| АППГ | <98 | 98-99,99 | >100 |
| РТС | нет цветовой дифференциации | | |
| ИМ | нет цветовой дифференциации | | |
| Доля РТС | нет цветовой дифференциации | | |
| ССЧ | < СреднееССЧ \* 0,95 | > СреднееССЧ \* 0,95 и  < СреднееССЧ | >= СреднееССЧ |

Вывод в дашборд отдельного поля со «светофором» по этому показателю не требуется, планируемый макет представлен на рисунке № 7

Рисунок №7.



Итоговое значение АППГ также имеет цветовую дифференциацию согласно его итоговому значению. Смена цветовой схемы происходит аналогично указанной в таблице №6.

Цветовая маркировка итогового параметра ССЧ не требуется, т.к. этот параметр является средним значением параметров ССЧ по магазинам, и цветовая дифференциация по столбцу строится относительно его.

### Показатели по конверсии

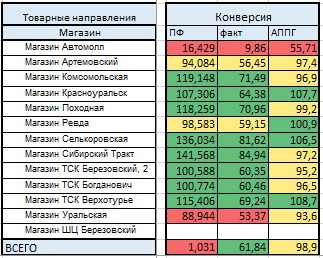
Для блока по конверсии предусмотрены цветовые заливки для всех показателей ПФ, факт и АППГ. Данные по зависимости цвета заливки от самого показателя представлены в таблице №7

Таблица №7.

|  |  |  |  |
| --- | --- | --- | --- |
| конверсия |  | | |
| цвет заливки | красный | желтый | зеленый |
| код цвета | 3 | 2 | 1 |
| ПФ | <90 | 90-99,99 | >=100 |
| факт | <55 | 55-59,99 | >=60 |
| АППГ | <90 | 90-99,99 | >100 |

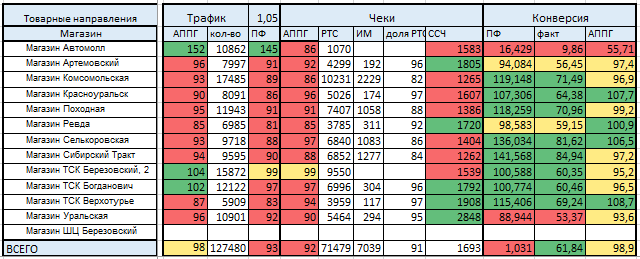
Вывод в дашборд отдельного поля со «светофором» по этому показателю не требуется, планируемый макет представлен на рисунке № 8

Рисунок №8



Итоговый макет вывода по всем не торговым показателям представлен на рисунке №9

Рисунок №9



Для того, чтобы не перегружать дашборд обилием данных и не рассеивать внимание пользователя, было принято решение добавить возможность сворачивать отдельные колонки. Таким образом, на основном экране будет отображаться только ключевой показатель в каждом блоке информации. Это позволит сохранить доступ ко всем важным метрикам, но при этом избежать чрезмерной загруженности интерфейса. Такое решение обеспечивает лаконичное и наглядное представление данных, фокусируя внимание пользователя на наиболее значимых индикаторах. Макет со свернутыми колонками представлен на рисунке №10

Рисунок №10



### Компоновка показателей сети магазинов в едином блоке дашборда

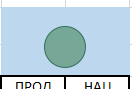
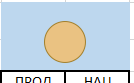
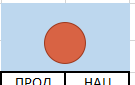
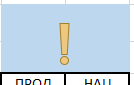
В правой верхней четверти дашборда расположен итоговый блок (см. Рисунок №1 в п. 2.3), который компактно отображает все ключевые показатели сети розничных магазинов в виде цветовой дифференциации. Данный блок позволяет руководству быстро оценить общее состояние бизнеса и вовремя среагировать на возникающие проблемы. Цветовая индикация позволяет мгновенно выявить проблемные области, требующие пристального внимания, что значительно повышает эффективность управления розничной сетью.

В эту часть выведены следующие показатели:

#### Итоговый обобщённый индикатор по всем основным показателям.

Оформлен в виде светофора классического вида, в него сведена вся обобщенная информация по основным показателям: продажи, наценка, оборачиваемость, остатки. По факту, цветовая индикация совпадает с итоговым «светофором» из основного блока, но больше по размеру и соответственно более заметен. В дашборде сайта индикация в виде светофора, может быть реализована в виде спидометра автомобиля, но этот вариант отнесен по дополнительные доработки (бантики), после запуска дашборда в формате MVP (минимально жизнеспособный продукт). Индикация в виде светофора представлена на рисунке №11

Рисунок №11

Обратите внимание, что в итоговом обобщенном показателе предусмотрена отдельная индикация в виде восклицательного знака, предусмотренная под отдельные, критические значения любого из параметров дашборда. При наступлении такого «критического» события, данные о нем передаются напрямую в итоговый светофор, без промежуточного расчета относительно других показателей.

Подобным критическим событием могут быть:

* Отрицательная наценка по одному из товарных направлений или магазинов
* Критически низкое значение выполнения плана продаж по магазину, или товарному направлению
* Критически низкий товарный остаток, и так далее

Формула расчета итогового параметра с учетом весов итоговых параметров выглядит следующим образом:

ОБОБитог = (ПРитог\*ВЕСпр + ОСитог\*ВЕСос + НЦитог\*ВЕСнц + ОБитог\*ВЕСоб)/4, где:

ОБОБитог – значение итогового обобщенного показателя, которое соответствует одному из цветов: зеленый, желтый, красный.

ПРитог – значение итогового по всем магазинам показателя план-факт по продажам,

ВЕСпр – значение веса показателя ПР

ОСитог – значение итогового по всем магазинам показателя план-факт по остатку

ВЕСос - значение веса показателя ОС

НЦитог – значение итогового по всем магазинам показателя план-факт по наценке,

ВЕСнц – значение веса показателя НЦ

ОБитог – значение итогового по всем магазинам показателя план-факт по оборачиваемости

ВЕСоб - значение веса показателя ОБ

Параметры смены цвета этого индикатора указаны в таблице №8

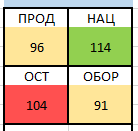
Таблица №8

|  |  |  |
| --- | --- | --- |
| Значение ОБОБитог | Цвет в дашборде | Смысловое значение |
| ОБОБитог < = 2 | зеленый | Все хорошо, вмешательства не требуется |
| 2 > ОБОБитог < = 3 | желтый | Не все показатели хороши, но еще в пределах нормы |
| 3 > ОБОБитог < 4 | красный | Плохое состояние в показателях, требуется анализ проблемы и срочное управленческое воздействие |
| 4 | Восклицательный знак | Появление «критического» значения одного из показателей. Требуется анализ проблемы и оперативное упразднение. |

#### Сводный блок итоговых индикаторов по основным показателям.

Сводный блок представлен в виде квадрата, содержащего внутри итоговые значения ключевых показателей: план-факт по продажам итоговый (ПФ ПРОДитог), план-факт по наценке итоговый (ПФ НАЦитог), план-факт по остаткам итоговый (ПФ ОСТитог), план-факт по оборачиваемости итоговый (ПФ ОБОРитог). Каждый сегмент блока имеет свою цветовую дифференциацию, соответствующую аналогичной цветовой дифференциации в основном блоке. Это позволяет руководству мгновенно выявить зоны риска и отклонения от плановых показателей, принимая своевременные управленческие решения для корректировки ситуации. Сочетание цвета сегмента с абсолютным значением отображаемого показателя дает дополнительную визуальную информацию, помогая быстро оценить критичность ситуации и принять необходимые меры. Общий вид этого блока представлен на рисунке № 12.

Рисунок №12



Переходы цвета в этом блоке построены по тому же принципу, что и цвета значений основных показателей, только за основу теперь берутся итоговые значения основных показателей, смена цветов представлена в таблице №9.

Таблица №9

|  |  |  |  |
| --- | --- | --- | --- |
| код | 3 | 2 | 1 |
| цвет | красный | желтый | зеленый |
| ПФ ПРОДитог | <93 | >= 93 и 98 <= | > 98 |
| ПФ ОСТитог | <90 или >102 | >= 90 и 95 <= | >= 95 и 102 <= |
| ПФ НАЦитог | <80 или >=120 | >= 80 и 95 <= | >= 95 и 102 <= |
| ПФ ОБОРитог | <80 или >=120 | >= 80 и 95 <= | >= 95 и 102 <= |

Абсолютные значения итоговых показателей совпадают со значениями этих же показателей в итоговой части основного блока показателей дашборда (часть «ВСЕГО»).

#### Сводный блок итоговых индикаторов по неторговым показателям.

Данный блок выводит обобщённую информацию по следующим показателям: специальные задачи (СПЕЦ), АППГ по трафику (анализ показателя к прошлому году (АППГ), ТРАФИК), АППГ по количеству чеков (ЧЕКИ), план-факт по конверсии (КОНВЕРСИЯ). Условно блок состоит из четырех индикаторов типа «светофор» расположенных вертикально друг над другом. Схематичный вывод блока представлен на рисунке №13.

Рисунок №13



Рассмотрим показатели более подробно:

##### Показатель специальные задачи (СПЕЦ)

СПЕЦ– является обобщенным для трех итоговых показателей по трем спец. задачам. В «светофор» выводиться показатель СРЕД(СПЕЦ1, СПЕЦ2, СПЕЦ3) который является усредненным значением итоговых показателей по всем спец.задачам. Цветовая дифференциация представлены в таблице №10

Таблица №10

|  |  |  |  |
| --- | --- | --- | --- |
| итоговый светофор СПЕЦ | | | |
| цвет | красный | желтый | зеленый |
| Значение  СРЕД(СПЕЦ1, СПЕЦ2, СПЕЦ3) | <98 | 98-99,99 | >100 |

##### Показатель АППГ по трафику (ТРАФИК)

ТРАФИК является дублирующим цветовым дифферентом итогового значения показателя АППГ трафик описанного ранее в п. 3.3 [Показатели по трафику](#_Показатели_по_трафику). В данном случае добавление его в итоговый блок, сделано только для улучшения его визуализации и повышения скорости считывания итоговой информации.

##### Показатель АППГ по чекам (ЧЕКИ)

ЧЕКИ так же как и ТРАФИК является дублирующим цветовым дифферентом итогового значения показателя АППГ ЧЕКИ описанного ранее в п. 3.4 Показатели по чекам.

##### Показатель ПФ по конверсии (КОНВЕРСИЯ)

Показатель ПФ, так же как и ЧЕКИ и ТРАФИК является дублирующим цветовым дифферентом итогового значения показателя ПФ Конверсия описанного ранее в п.3.5 Показатели по конверсии. В этом случае мы добавляем элемент в итоговый блок, чтобы сделать его визуально более привлекательным и облегчить чтение итоговой информации. Это улучшает восприятие и скорость понимания результата.

### Визуализация ключевых показателей в нижней части дашборда в виде диаграмм

"Диаграммы — это инструмент, который помогает познать реальность. Они позволяют нам видеть закономерности и связи, которые ускользают от простого перечисления цифр"[[8]](#footnote-8)

В нижней части дашборда размещены диаграммы (см. Рисунок №1 в п. 2.3), которые позволяют пользователям детально анализировать текущие значения ключевых показателей в разрезе дат. Данная визуализация представлена в виде наглядных гистограмм, что дает возможность сравнивать плановые и фактические значения, отслеживать динамику изменений и выявлять важные тенденции. Такой формат обеспечивает простоту восприятия и облегчает принятие обоснованных управленческих решений.

На данном этапе проекта планируется вывод двух блоков диаграмм сходных по стилю и оформлению. Диаграмма по выручке и диаграмма по спец.задачам. Обе диаграммы должны в лаконичной форме выводить текущую информацию по показателям: дневной план, план-факт выполнения продаж (выручки) и абсолютный показатель выручки.

При этом, абсолютный показатель выручки выводится в виде гистограммы (столбиками), а показатель план-факт по продажам и дневной план в виде графика (по аналогии с маркированным графиком MS Excel).

Рассмотрим более подробно каждую из диаграмм.

#### Диаграмма по выручке

Конструктивный макет блока диаграммы представлен на схеме рисунка №14

Рисунок №14

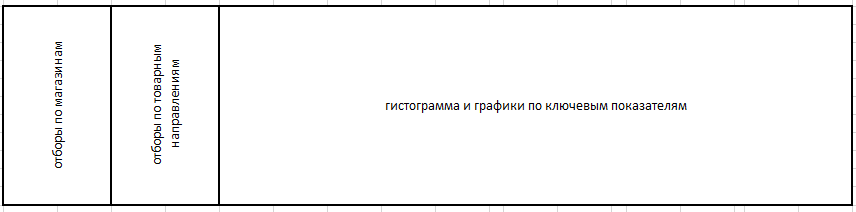


Диаграмма условно разделена на три блока:

##### Блок управления отборами магазинов.

Данный блок позволяет производить отбор как одного магазина, так и нескольких, с точки зрения исполнения на дашборде рекомендуется выполнить в виде списка магазинов с чекбоксами, соответственно выставляя необходимое количество «галочек» в чекбоксы, можно получать данные в диаграмме по выбранным объектам торговли.

##### Блок управления отборами товарных направлений.

Данный блок позволяет производить отбор как одного товарного направления, так и нескольких, с точки зрения исполнения на дашборде так же рекомендуется выполнить в виде списка товарных направлений с чекбоксами, соответственно выставляя необходимое количество «галочек» в чекбоксы, можно получать данные в диаграмме по выбранным товарным направлениям.

##### Блок диаграмма.

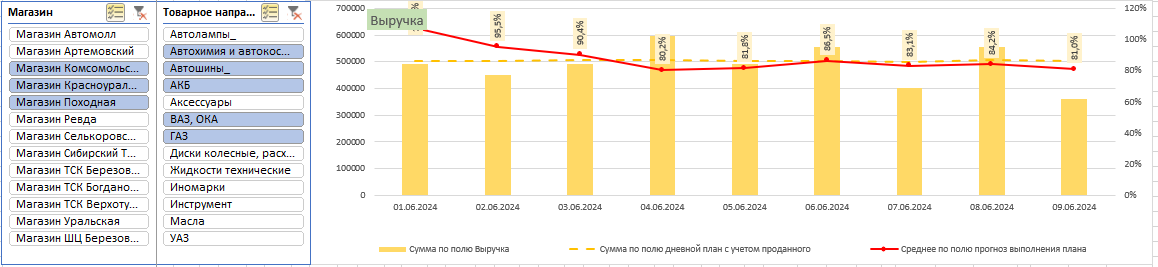
Данный блок представляет из себя сочетание:

* гистограммы по показателю: «выручка» (абсолютное значение),
* графика по показателю «ежедневный план» (абсолютное значение),
* графика по показателю «план-факт по выручке» (относительное значение).

все показателю выводятся в дашборд по каждому дню текущего месяца.

Функциональный макет блока «Диаграмма по выручке» представлен на рисунке №15

Рисунок №15



#### Диаграмма по спец.задачам

Конструктивный макет блока диаграммы представлен на схеме рисунка №16

Рисунок №16

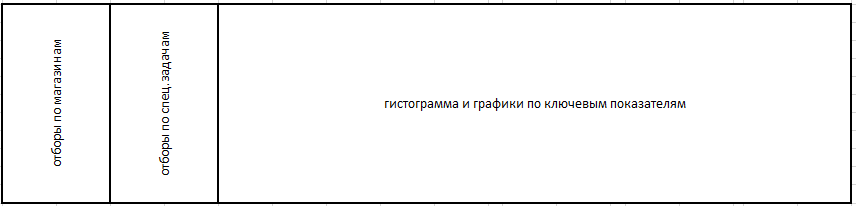


Диаграмма так же условно разделена на три блока:

##### Блок управления отборами магазинов.

Данный блок позволяет производить отбор как одного магазина, так и нескольких, с точки зрения исполнения на дашборде рекомендуется выполнить в виде списка магазинов с чекбоксами, соответственно выставляя необходимое количество «галочек» в чекбоксы, можно получать данные в диаграмме по выбранным объектам торговли.

##### Блок управления отборами спец. задач.

Данный блок позволяет производить отбор как одной спец. задачи, так и нескольких, с точки зрения исполнения на дашборде так же рекомендуется выполнить в виде списка товарных направлений с чекбоксами.

##### Блок диаграмма.

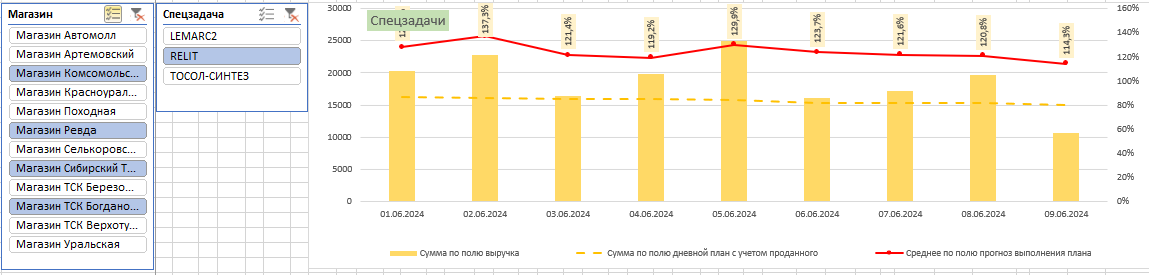
Данный блок аналогичен блоку по выручке, выполняется в той-же стилистике и представляет из себя сочетание:

* гистограммы по показателю: «выручка по спец. задаче» (абсолютное значение),
* графика по показателю «ежедневный план по спец. задаче» (абсолютное значение),
* графика по показателю «план-факт по выручке спец. задаче» (относительное значение).

все показателю выводятся в дашборд по каждому дню текущего месяца.

Функциональный макет блока «Диаграмма по спец. задачам» представлен на рисунке №17

Рисунок №17



## Глава 4. Разработка структуры и код приложения



### Структура кода

Разработка сложных программных систем, реализующих веб-приложения, требует применения комплексных инструментальных средств, обеспечивающих высокую производительность и качество процесса программирования. В этом контексте интегрированная среда разработки PyCharm и веб-фреймворк Django представляют собой эффективное сочетание для создания надежных и масштабируемых веб-ориентированных программных решений.

Среда PyCharm предоставляет широкий спектр возможностей, таких как автоматическое завершение кода, интеллектуальная подсветка синтаксиса и интеграция с системами контроля версий, которые значительно повышают скорость и качество разработки. В свою очередь, Django, как высокоуровневый фреймворк с открытым исходным кодом, значительно упрощает реализацию типовых функций веб-приложений, таких как обработка HTTP-запросов, управление базами данных и создание административной панели, позволяя разработчикам сосредоточиться на уникальной бизнес-логике.

Эффективное взаимодействие и интеграция PyCharm и Django способствуют повышению продуктивности разработчиков за счет автоматизации рутинных задач, предоставления контекстно-зависимых подсказок и ускорения итерационного процесса разработки.

Именно поэтому разработка велась с использованием инструментального комплекса Django.

### Модели (Models)

Модели отвечают за хранение информации, на основе моделей формируются таблицы в базе данных. Код написания моделей на языке Python моделей представлен в [Приложении №5](#_Приложение_№5_Модели). Для реализации проекта были созданы следующие модели:

#### class Shops

Модель содержащая наименования магазинов. Используется для связи между остальными моделями.

#### class ProductLine

Основная модель, содержащая данные по продажам в следующих полях:

Магазин – Товарное направление – показатели.

* Магазин
* Основной цвет (обобщенный показатель состояния показателей ОБОБ расчет указан в [п. 3.1.2](#_Цветовая_дифференциация_основных))
* Наименование
* План - факт по продажам
* План - факт по наценке
* План - факт по остаткам
* План - факт по оборачиваемости

Показатели описаны в [п.3.1](#_Основные_показатели)

#### class SpecialTask

Модель, содержащая данные по итоговым показателям по всем спецзадачам в следующих полях:

* Магазин
* Спец. задача
* План-факт

Показатели описаны в [п.3.2](#_Показатели_по_спец.)

#### class Traffic

Модель, содержащая данные по трафику в следующих полях:

* Магазин
* АППГ
* Количество
* План-Факт

Показатели описаны в [п.3.3](#_Показатели_по_трафику)

#### class Checks

Модель, содержащая данные по чекам в следующих полях:

* Магазин
* АППГ
* Кол-во чеков розницы
* Количество чеков с сайта
* Доля чеков розницы

Показатели описаны в [п.3.4](#_Показатели_по_чекам)

#### class Conversion

Модель, содержащая данные по конверсии в следующих полях:

* Магазин
* План-факт
* Фактическая конверсия
* АППГ

Показатели описаны в [п.3.5](#_Показатели_по_конверсии)

#### class ChartSales

Модель, содержащая данные по продажам в следующих полях:

* Магазин
* Дата
* Наименование товарного направления
* Фактические продажи
* План-факт
* План дневной

Показатели описаны в [п.3.7.1](#_Диаграмма_по_выручке)

#### class ChartSpecialTask

Модель, содержащая данные по продажам спец. задач в следующих полях:

* Магазин
* Дата
* Наименование спец. задачи
* Фактические продажи
* План-факт
* План дневной

Показатели описаны в [п.3.7.2](#_Диаграмма_по_спец.задачам)

#### class MainIndicator

Модель, содержащая данные для основного блока «светофорных» показателей дашборда в следующих полях:

* Основной цвет (описан в [п.3.6.1](#_Итоговый_обобщённый_индикатор))
* Цвет Продажи (описан в [п.3.6.2](#_Сводный_блок_итоговых))
* План-факт продажи (описан в [п.3.6.2](#_Сводный_блок_итоговых))
* Цвет Наценки (описан в [п.3.6.2](#_Сводный_блок_итоговых))
* План-факт наценки (описан в [п.3.6.2](#_Сводный_блок_итоговых))
* Цвет Остатки (описан в [п.3.6.2](#_Сводный_блок_итоговых))
* План-факт остатков (описан в [п.3.6.2](#_Сводный_блок_итоговых))
* Цвет Спец. задачи (описан в [п.3.6.3.1](#_Показатель_специальные_задачи))
* Цвет Траффик (описан в [п.3.6.3.2](#_Показатель_АППГ_по))
* Цвет Чеки (описан в [п.3.6.3.3](#_Показатель_АППГ_по_1))
* Цвет Конверсия (описан в [п.3.6.3.4](#_Показатель_ПФ_по))

### Представления (Views)

Представления отвечают за обработку информации, извлекаемую из базы данных. Код написания представления на языке Python представлен в [Приложении №6](#_Приложение_№6_Представления). Созданы следующие представления:

#### class SaleProductLine(View) – обработка данных по основным показателям

#### class SpecialTaskShops(View) – обработка данных для дашборда по спец. задачам

#### class TrafficShops(View) - обработка данных для дашборда по трафику

#### class ChecksShops(View) - обработка данных для дашборда по чекам

#### class ConversionShops(View) - обработка данных для дашборда по конверсии

#### class ChartSalesShops(View) - обработка данных для диаграммы по продажам

#### class ChartSpecialTaskShops(View) - обработка данных для диаграммы по спец. задачам

#### class MainIndicatorShops(View) - обработка данных для основного «светофора» дашборда.

### Шаблоны (Html)

Шаблоны отвечают за вид вывода информации на экран сайта, в данном проекте шаблоны отвечают за вывод информации в back – end web-приложения, в дальнейшем, эта информация будет преобразована в красивый интерфейс дашборда с помощью стилей, с использованием языка Java-Script. Код написания шаблонов на языке HTML и визуализация экрана back – end представлены в [Приложении №7](#_Приложение_№6_Представления). Созданы следующие шаблоны:

#### sale.html - вывод в back – end данных по основным показателям

#### stask.html - вывод в back – end данных для дашборда по спец. задачам

#### traffic.html) - вывод в back – end данных для дашборда по трафику

#### check.html- вывод в back – end данных для дашборда по чекам

#### conversion.html- вывод в back – end данных для дашборда по конверсии

#### mainindicator.html- вывод в back – end данных для основного «светофора» дашборда.

#### chartsalesshops.html- вывод в back – end данных для диаграммы по продажам

#### chartspecsalesshops.html- вывод в back – end данных для диаграммы по спец. задачам

### Пути (Urls)

Для управления выводом информации по маршрутам, настроены следующие пути:

#### 4.5.1 urls.py – проекта

Предусмотрена «админка» и установлен debug\_toolbar:

from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('backend/', include('backendapp.urls')),  
 path('\_\_debug\_\_/', include("debug\_toolbar.urls")),  
  
]

#### 4.5.1 urls.py – приложения

Настроены пути вывода информации через URL-строку:

from django.urls import path  
from . import views  
  
from .views import (SaleProductLine, SpecialTaskShops, TrafficShops, ChecksShops,  
 MainIndicatorShops, ConversionShops, ChartSalesShops, ChartSpecialTaskShops)  
  
  
  
urlpatterns = [  
  
 path('sale/', SaleProductLine.as\_view(), name='sale'),  
 path('stask/', SpecialTaskShops.as\_view(), name='stask'),  
 path('traffic/', TrafficShops.as\_view(), name='traffic'),  
 path('check/', ChecksShops.as\_view(), name='check'),  
 path('conf/', ConversionShops.as\_view(), name='conf'),  
 path('chsale/', ChartSalesShops.as\_view(), name='conf'),  
 path('chspecsale/', ChartSpecialTaskShops.as\_view(), name='conf'),  
 path('mi/', MainIndicatorShops.as\_view(), name='mi'),

# Заключение

"Не ждите, пока у вас будет идеальная аналитика. Начинайте с того, что имеете, и совершенствуйтесь по ходу."[[9]](#footnote-9)

В ходе работы по разработке back-end приложения - дашборда были получены следующие теоретические и практические выводы:

Рассмотрены различные варианты дашбордов, изучены их типовые элементы, методы визуализации и способы взаимодействия с пользователем. Проанализированы наиболее подходящие решения для представления ключевых бизнес-показателей.

Отобраны наиболее релевантные метрики и показатели, характеризующие эффективность работы организации. Разработана комплексная модель в Microsoft Excel для проверки гипотез и оценки влияния различных факторов.

Проведенное исследование показало высокую практическую значимость разработанного дашборда. Он позволяет руководству компании оперативно анализировать ключевые бизнес-показатели, выявлять проблемные зоны и принимать обоснованные управленческие решения. Рекомендуется регулярно актуализировать набор метрик и показателей, совершенствовать визуальное представление данных с учетом обратной связи от пользователей.

В целом, поставленная цель достигнута, все задачи выполнены, гипотеза о повышении эффективности управления бизнесом с помощью разработанного дашборда подтверждена.

Предложено продолжить исследования в направлении интеграции дашборда с другими информационными системами, а также его адаптации под мобильные устройства. Улучшить визуальную составляющую выводимых метрик, добавить более современные стилистические элементы.

Достигнут практический результат – на тестовом сервере развернут back-end web-приложения для дальнейшей обработки html-запросов в полноценное красивое приложение. Этот функционал будет реализован позднее с помощью стилизации на языке Java Script.

# Список используемой литературы

##### Книга: Alister Croll & Ben Yoskowitz (Алистер Крол и Бен Йосковиц). "Lean Analytics". 398 стр. Издательство: O'Reilly Media. США, 2013год.

##### Книга: Джин Желязны. «Говори на языке диаграмм». 210 стр. Издательством PRC Publishing Ltd Великобритания. 1981 год.

##### Книга: Michael Alexander & John Walkenbach (Майкл Александер и Джон Волкенбах). «Excel Dashboards & Reports». 336 стр. Издательство Wiley в Hoboken, New Jersey, США. 2010 год.

##### Иследование: "Dashboards and Visualization Report 2022" (Источник: <https://www.pwc.com/>) 2022г.

##### Отчет компании: The Forrester Wave™: Augmented Business Intelligence Platforms, Q2 2023 (<https://reprints2.forrester.com/#/assets/2/108/RES178493/report>) 2023г.

# Приложения

**Приложение №1** Контур программного обеспечения

**Приложение №2** Матрица цветовой индикации основных показателей дашборд

**Приложение №3** Функциональный макет-схема дашборда. Свернутые показатели.

**Приложение №4** Функциональный макет-схема дашборда. Развернутые показатели.

[**Приложение №5**](#_Приложение_№5_Модели) Модели(код)

[**Приложение №6**](#_Приложение_№6_Представления) Представления(код)

[**Приложение №7**](#_Приложение_№7_Шаблоны) Шаблоны(код) и экраны

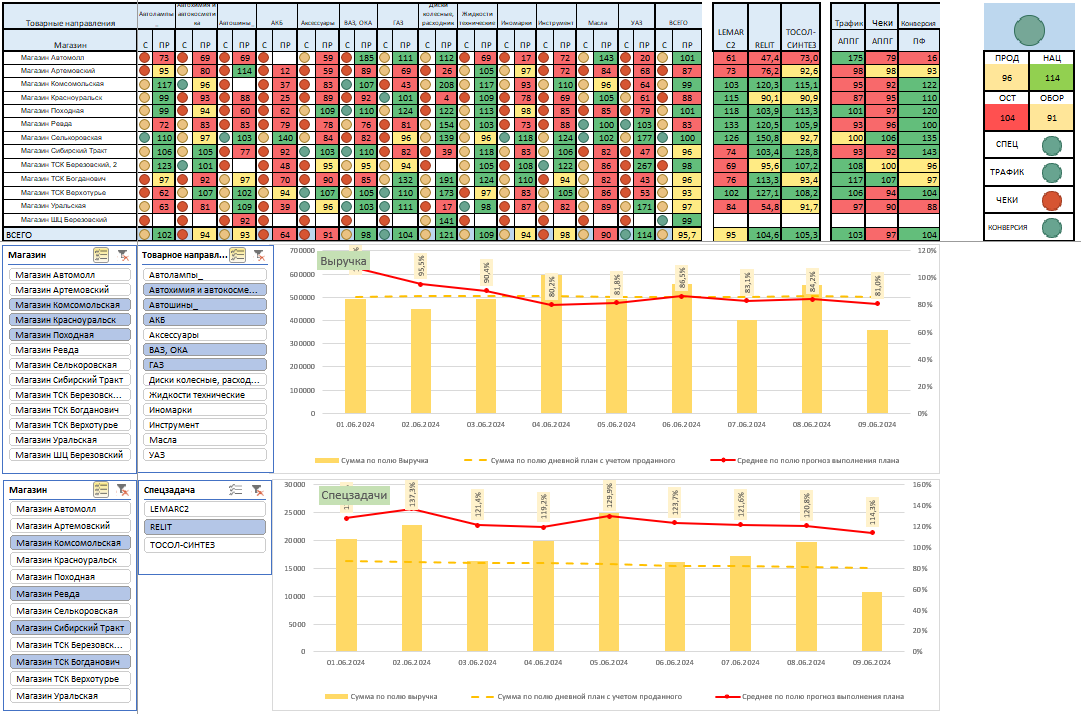
## Приложение №1 Контур программного обеспечения

## Приложение №2 Матрица цветовой индикации основных показателей дашборд:

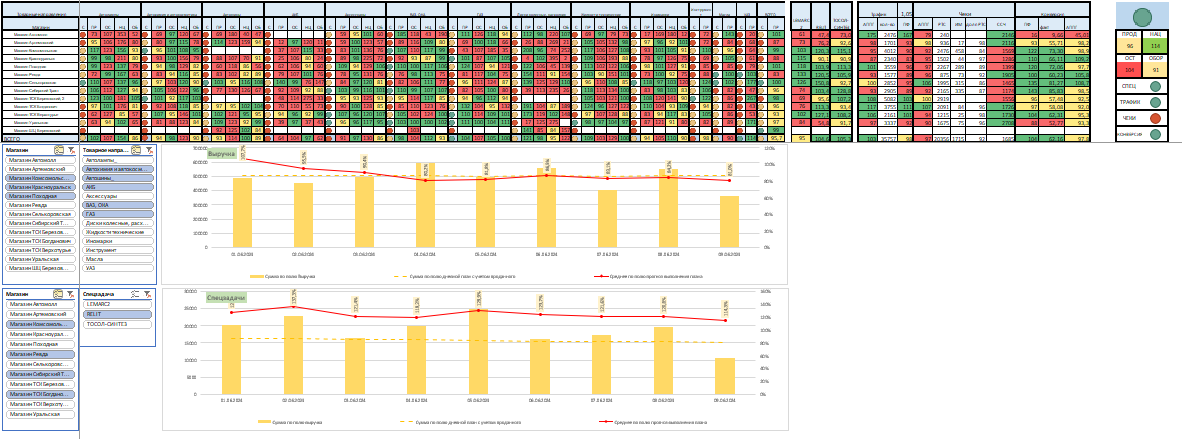




## Приложение №3 Функциональный макет-схема дашборда. Свернутые показатели.



## Приложение №4 Функциональный макет-схема дашборда. Развернутые показатели.



## 

## Приложение №5 Модели (Код).

from django.db import models  
from datetime import date  
  
class Shops(models.Model):  
 name = models.CharField(max\_length=30)  
  
  
class ProductLine(models.Model):  
 name = models.CharField(max\_length=30)  
 shops = models.ForeignKey(Shops, on\_delete=models.CASCADE)  
 color = models.IntegerField()  
 colorName = models.CharField(max\_length=1, default='3')  
 planFact = models.DecimalField(decimal\_places=2, max\_digits=5)  
 planFactName = models.CharField(max\_length=2, default='ПР')  
 margin = models.DecimalField(decimal\_places=2, max\_digits=5)  
 marginName = models.CharField(max\_length=2, default='ОС')  
 balances = models.DecimalField(decimal\_places=2, max\_digits=5)  
 balancesName = models.CharField(max\_length=2, default='НЦ')  
 turnover = models.DecimalField(decimal\_places=2, max\_digits=5)  
 turnoverName = models.CharField(max\_length=2, default='ОБ')  
  
  
class SpecialTask(models.Model):  
 name = models.CharField(max\_length=30)  
 planFact = models.DecimalField(decimal\_places=2, max\_digits=5)  
 shops = models.ForeignKey(Shops, on\_delete=models.CASCADE)  
  
  
class Traffic(models.Model):  
 shops = models.ForeignKey(Shops, on\_delete=models.CASCADE)  
 APPG = models.DecimalField(decimal\_places=2, max\_digits=5)  
 count = models.IntegerField()  
 planFact = models.DecimalField(decimal\_places=2, max\_digits=5)  
  
  
class Checks(models.Model):  
 shops = models.ForeignKey(Shops, on\_delete=models.CASCADE)  
 APPG = models.DecimalField(decimal\_places=2, max\_digits=5)  
 checksRoz = models.IntegerField()  
 checksEcom = models.IntegerField()  
 partCheckRoz = models.DecimalField(decimal\_places=2, max\_digits=5)  
  
  
class Conversion(models.Model):  
 shops = models.ForeignKey(Shops, on\_delete=models.CASCADE)  
 planFact = models.DecimalField(decimal\_places=2, max\_digits=5)  
 conversionFact = models.DecimalField(decimal\_places=2, max\_digits=5)  
 APPG = models.DecimalField(decimal\_places=2, max\_digits=5)  
  
  
  
class ChartSales(models.Model):  
 shops = models.ForeignKey(Shops, on\_delete=models.CASCADE)  
 date = models.DateField(default=date.today)  
 sales = models.DecimalField(decimal\_places=2, max\_digits=10)  
 productLine = models.CharField(max\_length=100,default="Проверка")  
 planFact = models.DecimalField(decimal\_places=4, max\_digits=10)  
 planDay = models.DecimalField(decimal\_places=2, max\_digits=10)  
  
  
class ChartSpecialTask(models.Model):  
 shops = models.ForeignKey(Shops, on\_delete=models.CASCADE)  
 date = models.DateField(default=date.today)  
 specialTask = models.CharField(max\_length=30)  
 sales = models.DecimalField(decimal\_places=2, max\_digits=10)  
 planFact = models.DecimalField(decimal\_places=4, max\_digits=10)  
 planDay = models.DecimalField(decimal\_places=2, max\_digits=10)  
  
  
class MainIndicator(models.Model):  
 color\_value = models.IntegerField()  
 colorPROD = models.IntegerField()  
 planFactMI = models.DecimalField(decimal\_places=2, max\_digits=5)  
 colorNAC = models.IntegerField()  
 marginMI = models.DecimalField(decimal\_places=2, max\_digits=5)  
 colorOST = models.IntegerField()  
 balancesMI = models.DecimalField(decimal\_places=2, max\_digits=5)  
 colorOBOR = models.IntegerField()  
 turnoverMI = models.DecimalField(decimal\_places=2, max\_digits=5)  
 colorSPEC = models.IntegerField()  
 colorTRAFF = models.IntegerField()  
 colorCHECK = models.IntegerField()  
 colorCONF = models.IntegerField()

## Приложение №6 Представления (Код).

import logging  
  
  
from django.http import JsonResponse  
from .models import ProductLine, Checks, ChartSales, ChartSpecialTask  
from .models import SpecialTask, Shops, Traffic, Conversion, MainIndicator  
from django.views import View  
from django.shortcuts import render  
  
  
logger = logging.getLogger(\_\_name\_\_)  
  
  
**class SaleProductLine(View):**  
 def get(self, request):  
 try:  
 product\_lines = ProductLine.objects.all()  
 except ProductLine.DoesNotExist:  
 return JsonResponse({'error': 'Данные не найдены'}, status=404)  
  
 data = []  
  
 for product\_line in product\_lines:  
 row = [  
 product\_line.shops.name,  
 product\_line.name,  
 product\_line.color,  
 product\_line.planFact,  
 product\_line.margin,  
 product\_line.balances,  
 product\_line.turnover  
 ]  
 data.append(row)  
  
 column\_names = ['Магазин', 'Продукт', 'Светофор', 'План/Факт', 'Наценка', 'Остатки', 'Оборачиваемость']  
  
 return render(request, 'sale.html', {'data': data, 'column\_names': column\_names})

**class SpecialTaskShops(View):**  
 def get(self, request):  
 try:  
 special\_tasks = SpecialTask.objects.all()  
 except SpecialTask.DoesNotExist:  
 return JsonResponse({'error': 'Данные не найдены'}, status=404)  
  
 data = []  
  
 for special\_task in special\_tasks:  
 row = [  
 special\_task.shops.name,  
 special\_task.name,  
 special\_task.planFact,  
  
 ]  
 data.append(row)  
  
 column\_names = ['Магазин', 'Спецзадача', 'План/Факт']  
  
 return render(request, 'stask.html', {'data': data, 'column\_names': column\_names})

**class TrafficShops(View):**  
 def get(self, request):  
 try:  
 traffics = Traffic.objects.all()  
 except Traffic.DoesNotExist:  
 return JsonResponse({'error': 'Данные не найдены'}, status=404)  
  
 data = []  
  
 for traffic in traffics:  
 row = [  
 traffic.shops.name,  
 traffic.APPG,  
 traffic.count,  
 traffic.planFact,  
 ]  
 data.append(row)  
  
 column\_names = ['Магазин', 'к АППГ', 'Фактическое количество', 'План/Факт']  
  
 return render(request, 'traffic.html', {'data': data, 'column\_names': column\_names})

**class ChecksShops(View):**  
 def get(self, request):  
 try:  
 checks = Checks.objects.all()  
 except Checks.DoesNotExist:  
 return JsonResponse({'error': 'Данные не найдены'}, status=404)  
  
 data = []  
  
 for check in checks:  
 row = [  
 check.shops.name,  
 check.APPG,  
 check.checksRoz,  
 check.checksEcom,  
 check.partCheckRoz,  
 ]  
 data.append(row)  
  
 column\_names = ['Магазин', 'к АППГ', 'Всего чеков розницы', 'Всего чеков с сайта', 'Доля чеков розницы']  
  
 return render(request, 'check.html', {'data': data, 'column\_names': column\_names})

**class ConversionShops(View):**  
 def get(self, request):  
 try:  
 conversions = Conversion.objects.all()  
 except Conversion.DoesNotExist:  
 return JsonResponse({'error': 'Данные не найдены'}, status=404)  
  
 data = []  
  
 for conversion in conversions:  
 row = [  
 conversion.shops.name,  
 conversion.planFact,  
 conversion.conversionFact,  
 conversion.APPG,  
 ]  
 data.append(row)  
  
 column\_names = ['Магазин', 'План/Факт', 'Конверсия', 'к АППГ']  
  
 return render(request, 'conversion.html', {'data': data, 'column\_names': column\_names})

**class ChartSalesShops(View):**  
 def get(self, request):  
 try:  
 chartsales = ChartSales.objects.all()  
 except ChartSales.DoesNotExist:  
 return JsonResponse({'error': 'Данные не найдены'}, status=404)  
  
 data = []  
  
 for chartsale in chartsales:  
 row = [  
 chartsale.shops.name,  
 chartsale.date,  
 chartsale.sales,  
 chartsale.productLine,  
 chartsale.planFact,  
 chartsale.planDay,  
 ]  
 data.append(row)  
  
 column\_names = ['Магазин', 'Дата', 'Выручка','Товарное направление', 'План/Факт', 'Дневной план']  
  
 return render(request, 'chartsalesshops.html', {'data': data, 'column\_names': column\_names})

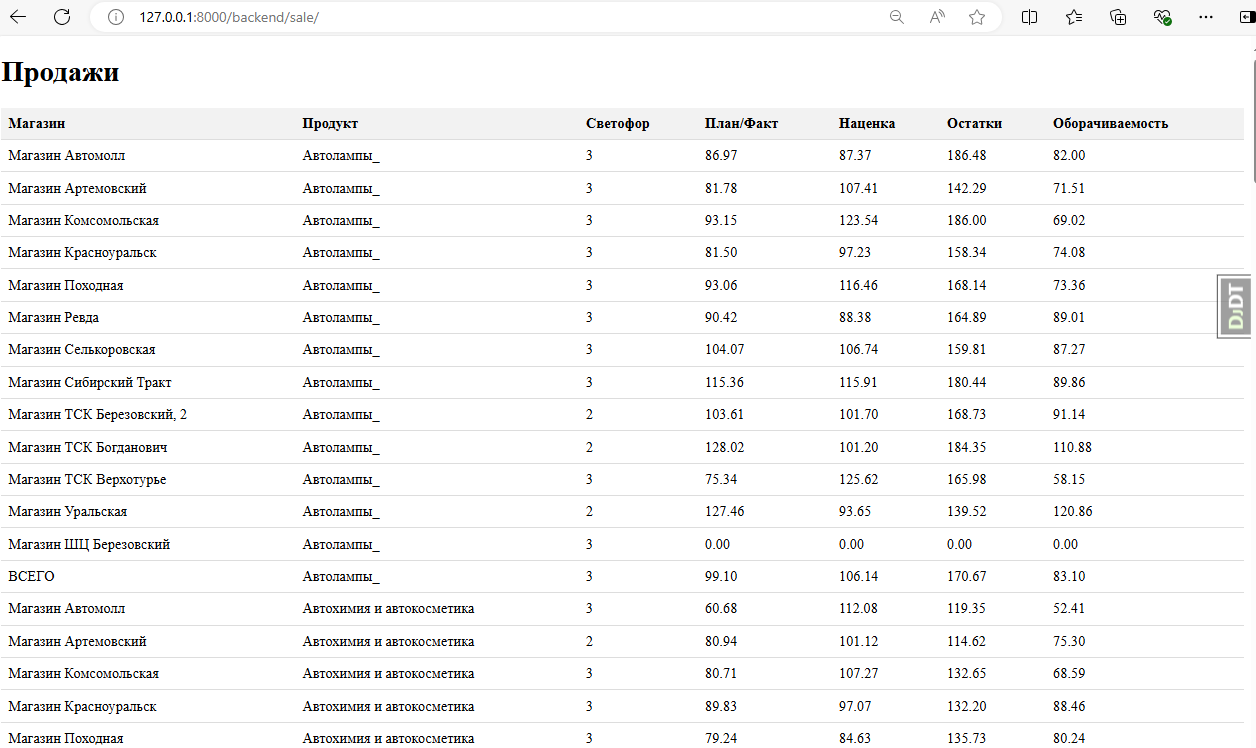
**class ChartSpecialTaskShops(View):**  
 def get(self, request):  
 try:  
 chartspecialsales = ChartSpecialTask.objects.all()  
 except ChartSpecialTask.DoesNotExist:  
 return JsonResponse({'error': 'Данные не найдены'}, status=404)  
  
 data = []  
  
 for chartspecialsale in chartspecialsales:  
 row = [  
 chartspecialsale.shops.name,  
 chartspecialsale.date,  
 chartspecialsale.sales,  
 chartspecialsale.specialTask,  
 chartspecialsale.planFact,  
 chartspecialsale.planDay,  
 ]  
 data.append(row)  
  
 column\_names = ['Магазин', 'Дата', 'Выручка','Спец.задача', 'План/Факт', 'Дневной план']  
  
 return render(request, 'chartspecsalesshops.html', {'data': data, 'column\_names': column\_names})

**class MainIndicatorShops(View):**  
 def get(self, request):  
 try:  
 mainindicators = MainIndicator.objects.all()  
 except MainIndicator.DoesNotExist:  
 return JsonResponse({'error': 'Данные не найдены'}, status=404)  
  
 data = []  
  
 for mainindicator in mainindicators:  
 row = [  
 mainindicator.color\_value,  
 mainindicator.colorPROD,  
 mainindicator.planFactMI,  
 mainindicator.colorNAC,  
 mainindicator.marginMI,  
 mainindicator.colorOST,  
 mainindicator.balancesMI,  
 mainindicator.colorOBOR,  
 mainindicator.turnoverMI,  
 mainindicator.colorSPEC,  
 mainindicator.colorTRAFF,  
 mainindicator.colorCHECK,  
 mainindicator.colorCONF,  
 ]  
 data.append(row)  
  
 column\_names = ['Итоговый светофор', 'Цвет ПРОД', 'ПРОД','Цвет НАЦ', 'НАЦ', 'Цвет ОСТ', 'ОСТ',  
 'Цвет ОБОР', 'ОБОР', 'Цвет СПЕЦ', 'Цвет ТРАФИК', 'Цвет Чеки', 'Цвет КОНВ']  
  
 return render(request, 'mainindicator.html', {'data': data, 'column\_names': column\_names})

## Приложение №7 Шаблоны и экраны (Код).

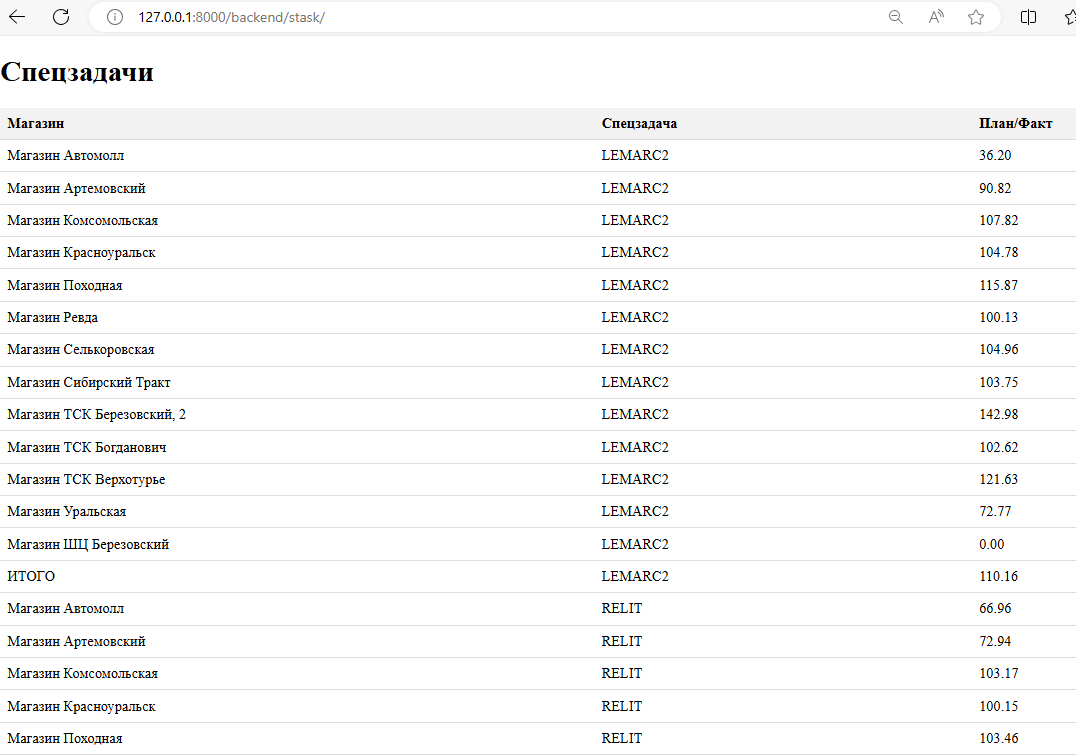
#### 7.1. sale.html

<!DOCTYPE html>  
<html>  
<head>  
 <title>Продажи</title>  
 <style>  
 table {  
 width: 100%;  
 border-collapse: collapse;  
 }  
 th, td {  
 padding: 8px;  
 text-align: left;  
 border-bottom: 1px solid #ddd;  
 }  
 th {  
 background-color: #f2f2f2;  
 }  
 </style>  
</head>  
<body>  
 <h1>Продажи</h1>  
 <table>  
 <thead>  
 <tr>  
 {% for column in column\_names %}  
 <th>{{ column }}</th>  
 {% endfor %}  
 </tr>  
 </thead>  
 <tbody>  
 {% for row in data %}  
 <tr>  
 {% for cell in row %}  
 <td>{{ cell }}</td>  
 {% endfor %}  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
</body>  
</html>



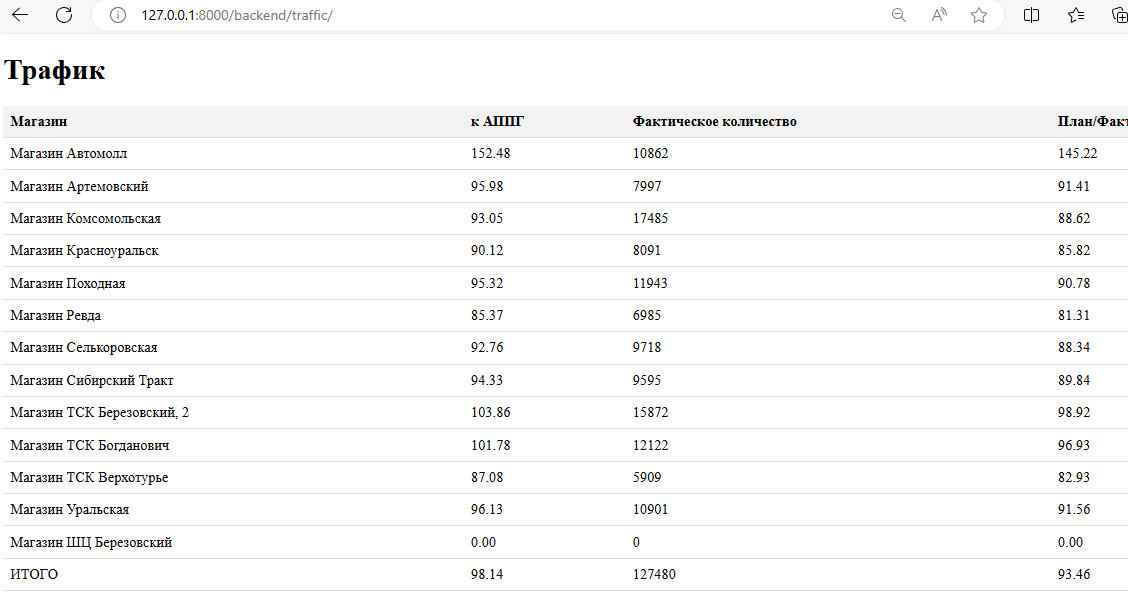
#### 7.2. stask.html

<!DOCTYPE html>  
<html>  
<head>  
 <title>Спецзадачи</title>  
 <style>  
 table {  
 width: 100%;  
 border-collapse: collapse;  
 }  
 th, td {  
 padding: 8px;  
 text-align: left;  
 border-bottom: 1px solid #ddd;  
 }  
 th {  
 background-color: #f2f2f2;  
 }  
 </style>  
</head>  
<body>  
 <h1>Спецзадачи</h1>  
 <table>  
 <thead>  
 <tr>  
 {% for column in column\_names %}  
 <th>{{ column }}</th>  
 {% endfor %}  
 </tr>  
 </thead>  
 <tbody>  
 {% for row in data %}  
 <tr>  
 {% for cell in row %}  
 <td>{{ cell }}</td>  
 {% endfor %}  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
</body>  
</html>



#### 7.3. traffic.html

<!DOCTYPE html>  
<html>  
<head>  
 <title>Трафик</title>  
 <style>  
 table {  
 width: 100%;  
 border-collapse: collapse;  
 }  
 th, td {  
 padding: 8px;  
 text-align: left;  
 border-bottom: 1px solid #ddd;  
 }  
 th {  
 background-color: #f2f2f2;  
 }  
 </style>  
</head>  
<body>  
 <h1>Трафик</h1>  
 <table>  
 <thead>  
 <tr>  
 {% for column in column\_names %}  
 <th>{{ column }}</th>  
 {% endfor %}  
 </tr>  
 </thead>  
 <tbody>  
 {% for row in data %}  
 <tr>  
 {% for cell in row %}  
 <td>{{ cell }}</td>  
 {% endfor %}  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
</body>  
</html>



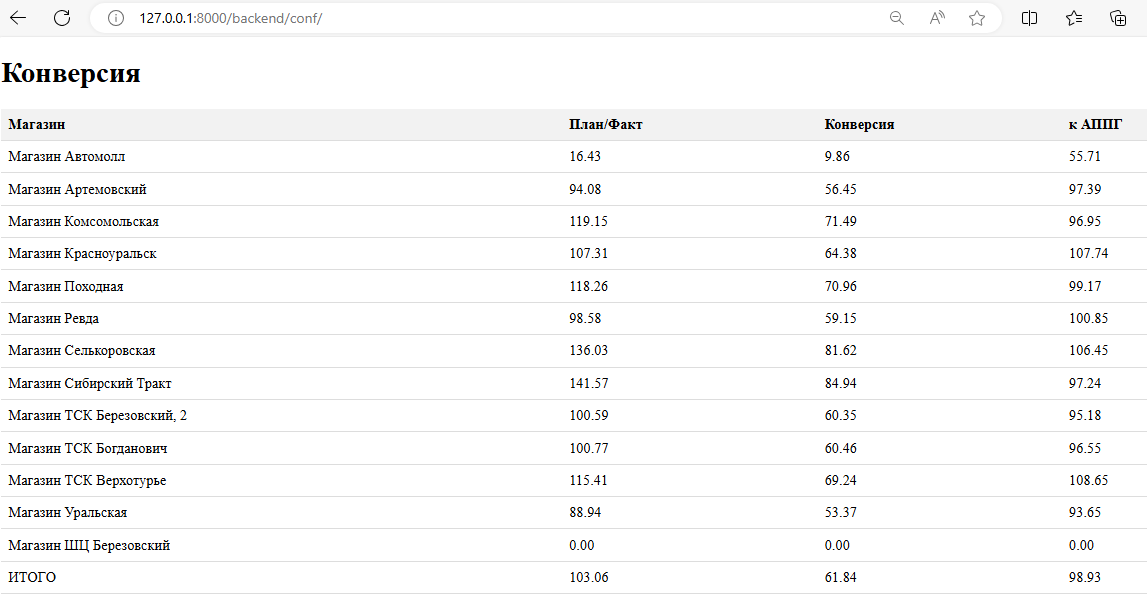
#### 7.4. check.html

<!DOCTYPE html>  
<html>  
<head>  
 <title>Чеки</title>  
 <style>  
 table {  
 width: 100%;  
 border-collapse: collapse;  
 }  
 th, td {  
 padding: 8px;  
 text-align: left;  
 border-bottom: 1px solid #ddd;  
 }  
 th {  
 background-color: #f2f2f2;  
 }  
 </style>  
</head>  
<body>  
 <h1>Чеки</h1>  
 <table>  
 <thead>  
 <tr>  
 {% for column in column\_names %}  
 <th>{{ column }}</th>  
 {% endfor %}  
 </tr>  
 </thead>  
 <tbody>  
 {% for row in data %}  
 <tr>  
 {% for cell in row %}  
 <td>{{ cell }}</td>  
 {% endfor %}  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
</body>  
</html>



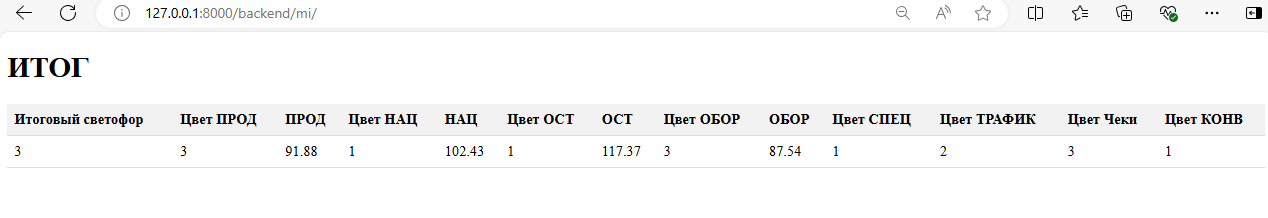
#### 7.5. conversion.html

<!DOCTYPE html>  
<html>  
<head>  
 <title>Конверсия</title>  
 <style>  
 table {  
 width: 100%;  
 border-collapse: collapse;  
 }  
 th, td {  
 padding: 8px;  
 text-align: left;  
 border-bottom: 1px solid #ddd;  
 }  
 th {  
 background-color: #f2f2f2;  
 }  
 </style>  
</head>  
<body>  
 <h1>Конверсия</h1>  
 <table>  
 <thead>  
 <tr>  
 {% for column in column\_names %}  
 <th>{{ column }}</th>  
 {% endfor %}  
 </tr>  
 </thead>  
 <tbody>  
 {% for row in data %}  
 <tr>  
 {% for cell in row %}  
 <td>{{ cell }}</td>  
 {% endfor %}  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
</body>  
</html>



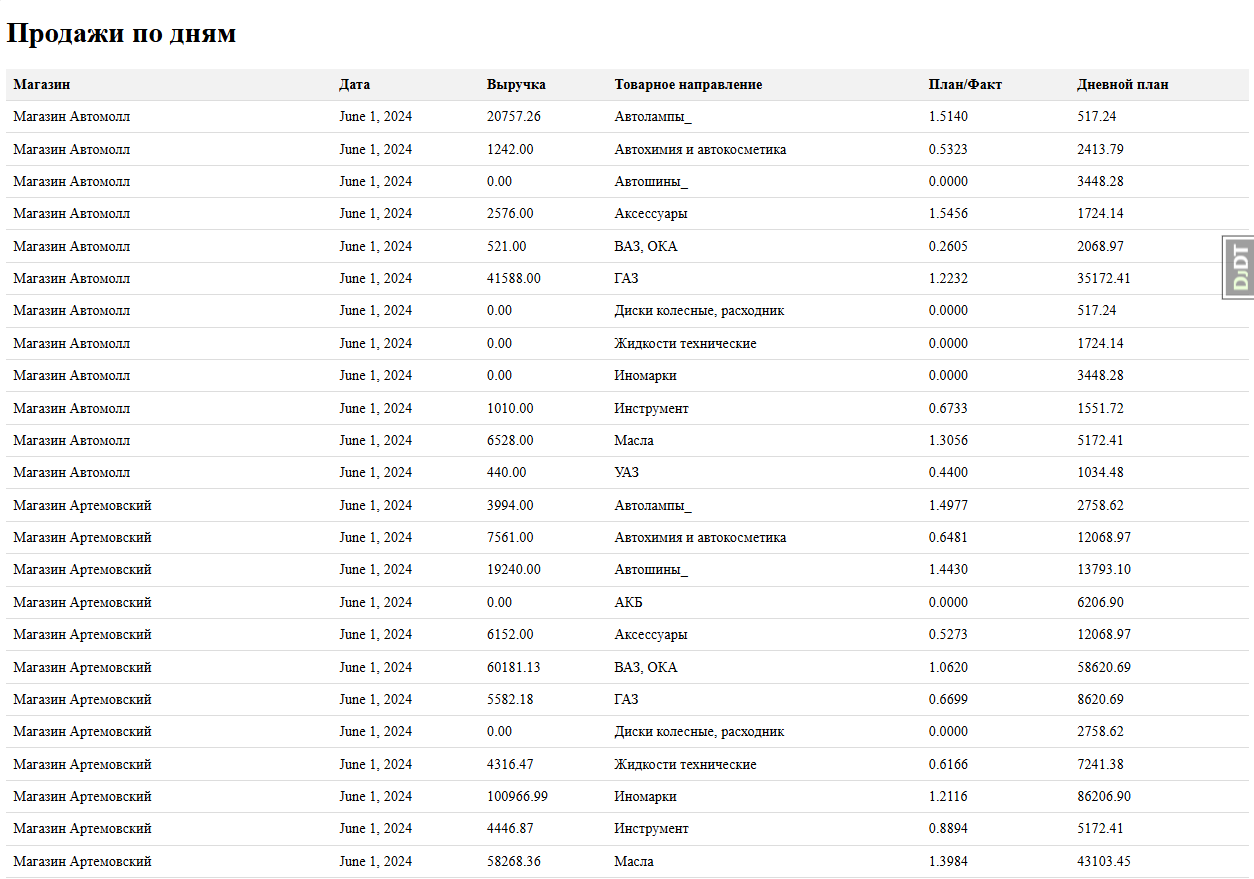
#### 7.6. mainindicator.html

<!DOCTYPE html>  
<html>  
<head>  
 <title>ИТОГ</title>  
 <style>  
 table {  
 width: 100%;  
 border-collapse: collapse;  
 }  
 th, td {  
 padding: 8px;  
 text-align: left;  
 border-bottom: 1px solid #ddd;  
 }  
 th {  
 background-color: #f2f2f2;  
 }  
 </style>  
</head>  
<body>  
 <h1>ИТОГ</h1>  
 <table>  
 <thead>  
 <tr>  
 {% for column in column\_names %}  
 <th>{{ column }}</th>  
 {% endfor %}  
 </tr>  
 </thead>  
 <tbody>  
 {% for row in data %}  
 <tr>  
 {% for cell in row %}  
 <td>{{ cell }}</td>  
 {% endfor %}  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
</body>  
</html>



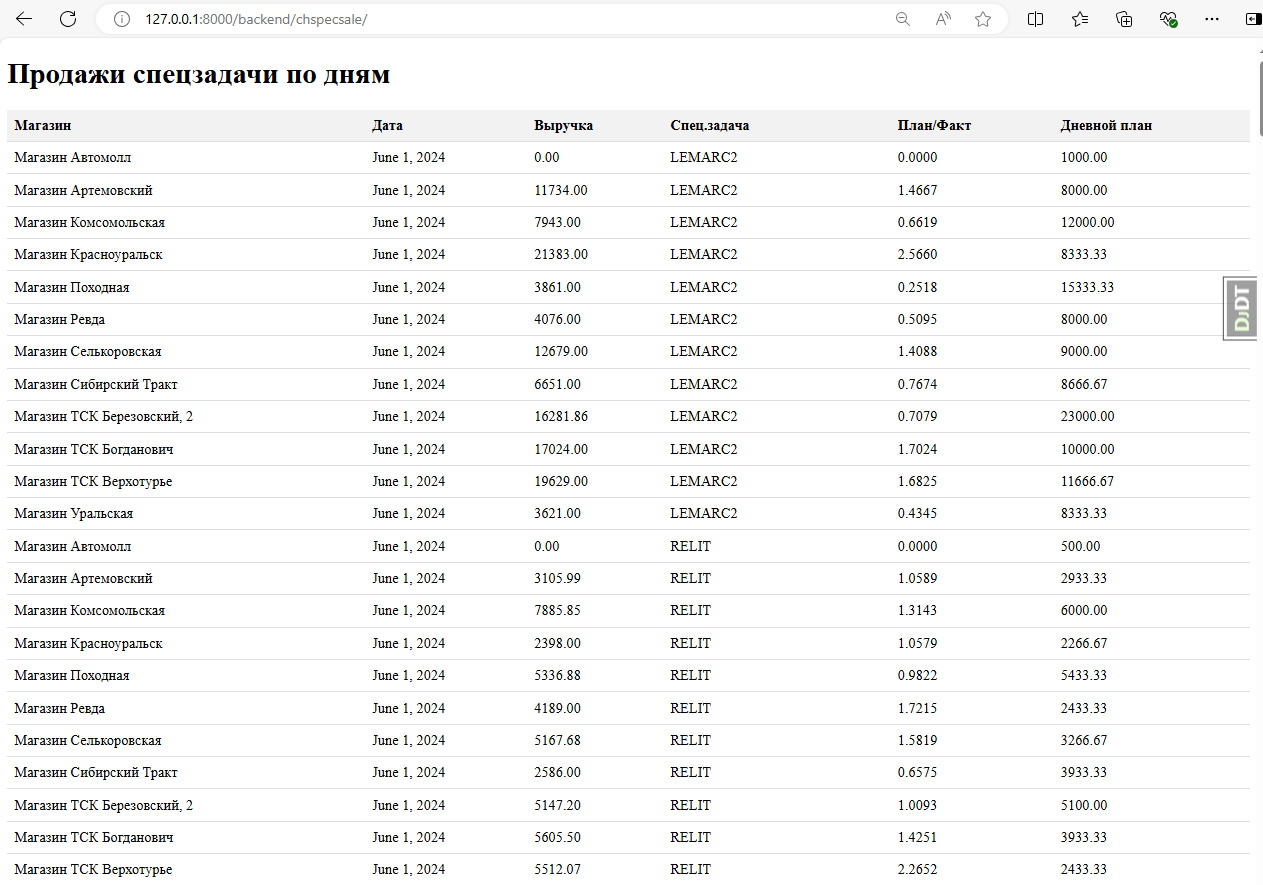
#### 7.7. chartsalesshops.html

<!DOCTYPE html>  
<html>  
<head>  
 <title>Продажи по дням</title>  
 <style>  
 table {  
 width: 100%;  
 border-collapse: collapse;  
 }  
 th, td {  
 padding: 8px;  
 text-align: left;  
 border-bottom: 1px solid #ddd;  
 }  
 th {  
 background-color: #f2f2f2;  
 }  
 </style>  
</head>  
<body>  
 <h1>Продажи по дням</h1>  
 <table>  
 <thead>  
 <tr>  
 {% for column in column\_names %}  
 <th>{{ column }}</th>  
 {% endfor %}  
 </tr>  
 </thead>  
 <tbody>  
 {% for row in data %}  
 <tr>  
 {% for cell in row %}  
 <td>{{ cell }}</td>  
 {% endfor %}  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
</body>  
</html>



#### 7.8. chartspecsalesshops.html

<!DOCTYPE html>  
<html>  
<head>  
 <title>Продажи спецзадачи по дням</title>  
 <style>  
 table {  
 width: 100%;  
 border-collapse: collapse;  
 }  
 th, td {  
 padding: 8px;  
 text-align: left;  
 border-bottom: 1px solid #ddd;  
 }  
 th {  
 background-color: #f2f2f2;  
 }  
 </style>  
</head>  
<body>  
 <h1>Продажи спецзадачи по дням</h1>  
 <table>  
 <thead>  
 <tr>  
 {% for column in column\_names %}  
 <th>{{ column }}</th>  
 {% endfor %}  
 </tr>  
 </thead>  
 <tbody>  
 {% for row in data %}  
 <tr>  
 {% for cell in row %}  
 <td>{{ cell }}</td>  
 {% endfor %}  
 </tr>  
 {% endfor %}  
 </tbody>  
 </table>  
</body>  
</html>



1. [Alister Croll & Ben Yoskowitz. "Lean Analytics"](#_Отчет_компании_The) [↑](#footnote-ref-1)
2. [Michael Alexander & John Walkenbach «Excel Dashboards & Reports»](#_Книга:_Michael_Alexander) [↑](#footnote-ref-2)
3. [Исследование "Dashboards and Visualization Report 2022"](#_Исследование:_\"Dashboards_and) [↑](#footnote-ref-3)
4. [Отчет Forrester: платформы расширенной бизнес-аналитики, 2 квартал 2023 г.](#_Отчет_компании_The_1) [↑](#footnote-ref-4)
5. [Alister Croll & Ben Yoskowitz. "Lean Analytics"](#_Отчет_компании_The) [↑](#footnote-ref-5)
6. [Michael Alexander & John Walkenbach «Excel Dashboards & Reports»](#_Книга:_Michael_Alexander) [↑](#footnote-ref-6)
7. [Alister Croll & Ben Yoskowitz. "Lean Analytics"](#_Отчет_компании_The) [↑](#footnote-ref-7)
8. [Джин Желязны «Говори на языке диаграмм»](#_Книга:_Джин_Желязны._1) [↑](#footnote-ref-8)
9. [Alister Croll & Ben Yoskowitz. "Lean Analytics"](#_Отчет_компании_The) [↑](#footnote-ref-9)