# ****Техническое задание для админ-панели****

## ****1. Общие требования****

* Админ-панель должна быть интуитивно понятной, удобной для работы операторов.
* Все действия должны быть логически организованы, с минимальным количеством кликов для выполнения задач.
* Интерфейс должен поддерживать адаптивность для работы на разных устройствах (ПК, планшеты).
* Цветовая гамма: использовать спокойные, профессиональные тона (например, серый, белый, синий). Кнопки действий выделять цветом для быстрого восприятия (зеленый — успех, красный — удаление/отклонение, желтый — предупреждение).
* Важные уведомления и кнопки должны быть заметны.

## ****2. Функциональные требования****

### ****2.1. Общий пул заявок****

#### ****2.1.1. Заявки в работе****

* Отображение двух типов заявок:
  + **Крипта → Рубли** .
  + **Рубли → Крипта** .
* Для каждой заявки показывать:
  + ID заявки.
  + Статус (в работе, оплачена, заморожена и т.д.).
  + Дата создания.
  + Сумма перевода.
  + Выданные реквизиты (если есть).

#### ****2.1.2. Выданные реквизиты****

* Показывать список реквизитов с указанием:
  + К какой заявке они привязаны.
  + Откуда выданы (источник).
* Возможности:
  + **Смена реквизитов** для заявок типа "Крипта → Рубли".
  + **Удаление заявки** для типа "Рубли → Крипта" (при отказе клиента).

#### ****2.1.3. Поиск заявки по ID и фильтры****

* Поле поиска в верхней части интерфейса.
* При вводе ID отображается карточка заявки с подробной информацией.
* Предусмотреть поля фильтрации по типу заявки, по дате, по сумме(от и до), по статусу

#### ****2.1.4. Реквизиты в статусе "Заморозить"****

* Отдельный раздел для просмотра всех замороженных реквизитов.
* Возможность выбора дальнейших действий (одобрить, отклонить).

### ****2.2. Уведомления и действия****

#### ****2.2.1. Уведомление о выдаче реквизитов****

* После выдачи реквизитов отправка уведомление всем операторам.
* В уведомлении указывается:
  + ID заявки.
  + Куда и кому выданы реквизиты.

#### ****2.2.2. Получение чеков****

* После загрузки чеков появляется уведомление с кнопкой **"Проверить"** .
* Если реквизитов несколько:
  + Для каждого реквизита открывается отдельный блок с кнопками:
    - **Отклонить** .
    - **Заморозить** .
    - **Одобрить** .

#### ****2.2.3. Действия после проверки чеков****

* **Одобрение** : Оператор нажимает **"Оплаченная заявка"** .
* **Отклонение** :
  + Появляется выбор:
    - **Заменить** реквизиты.
    - **Пересчитать заявку** .
    - **Удалить** заявку.
* **Заморозка** :
  + Если клиент сообщает, что перевод попал в обработку, или это видно на чеке, оператор может нажать **"Заморозить"** .
  + После решения проблемы с обработкой:
    - **Отклонить** (если клиент запросил новые реквизиты или отказался от оплаты).
    - **Одобрить** (если деньги ушли получателю).

#### ****2.2.4. Возврат на предыдущий этап****

* На каждом этапе должна быть возможность вернуться на предыдущий шаг (например, если оператор ошибся при выборе действия).

### ****2.3. Работа с заявками "Рубли → Крипта"****

* **Продление времени** : Если клиент не успевает оплатить в указанный срок, оператор может нажать **"Продлить время"** .
* **Удаление заявки** : Если клиент отказался от оплаты, оператор может удалить заявку.
* **Ошибочная заявка** : Если возникли проблемы с клиентом, оператор может отметить заявку как **"Ошибочная"** .

## ****3. Визуальные требования****

### ****3.1. Общий дизайн****

* Использовать современный минималистичный дизайн.
* Шрифты: Sans-serif (например, Roboto, Open Sans).
* Размер шрифта: основной текст — 14px, заголовки — 16-18px.
* Цвета:
  + Фон: светлый (белый или светло-серый).
  + Текст: черный или темно-серый.
  + Кнопки:
    - Успех: зеленый (#4CAF50).
    - Предупреждение: желтый (#FFC107).
    - Ошибка: красный (#F44336).
    - Нейтральные действия: синий (#2196F3).

### ****3.2. Элементы интерфейса****

* **Таблицы** : Для отображения пула заявок использовать таблицы с возможностью сортировки по столбцам (ID, дата, статус и т.д.).
* **Карточки заявок** : При нажатии на заявку открывается детальная информация в виде карточки.
* **Кнопки действий** : Размещать рядом с каждой заявкой или в нижней части карточки.
* **Уведомления** : Всплывающие окна в правом верхнем углу экрана.

### ****3.3. Адаптивность****

* Интерфейс должен корректно отображаться на экранах разного размера.
* На мобильных устройствах таблицы преобразуются в карточки с вертикальным расположением данных.

## ****4. Технические требования****

* Верстка должна быть выполнена с использованием HTML5, CSS3, JavaScript.
* Поддержка современных браузеров (Chrome, Firefox, Safari, Edge).
* Использование фреймворков: Bootstrap, Tailwind CSS.
* Интеграция с бэкендом через API (формат данных — JSON).

## ****5. Пример пользовательского сценария****

1. Оператор заходит в админ-панель.
2. Видит общий пул заявок.
3. Выбирает заявку с ID 12345.
4. Проверяет статус и реквизиты.
5. Если чеки загружены, нажимает **"Проверить"** .
6. Для каждого реквизита выбирает действие: **Одобрить** , **Заморозить** или **Отклонить** .
7. После успешной проверки отмечает заявку как **"Оплаченная"** .

## ****6. Дополнительные пожелания****

* Добавить возможность экспорта данных в Excel (например, список заявок за определенный период).
* Реализовать фильтры для быстрого поиска заявок по статусу, дате, типу.