**Техническое задание для проекта "Электронный документооборот"**

**1. Общая информация**

**Название проекта**: Электронный документооборот (ЭДО)  
**Цель проекта**: Разработка веб-приложения для управления электронными документами в рамках учебного проекта.  
**Технологический стек**:

* Backend: Java, Spring Boot, Spring Data JPA, Spring Security
* Frontend: HTML, CSS, JavaScript, React
* База данных: PostgreSQL
* Формат обмена данными: REST API, JSON
* Аутентификация: Логин/пароль

**2. Функциональные требования**

**2.1. Управление пользователями**

* Регистрация и авторизация пользователей с двумя ролями: администратор и сотрудник.
* Администратор может просматривать список пользователей и блокировать их.
* Аутентификация через логин и пароль.

**2.2. Управление документами**

* Создание документов с полями: название, описание, дата создания, автор.
* Просмотр, редактирование и удаление документов.
* Прикрепление файлов к документам.
* Поиск документов по названию или автору.

**2.3. Бизнес-процессы**

* Простая система статусов документа: "Черновик", "На рассмотрении", "Утвержден".
* Возможность изменения статуса документа (только администратор может утверждать).
* Уведомления о смене статуса в веб-интерфейсе.

**2.4. Отчеты**

* Отображение списка документов с фильтрацией по статусу или автору.
* Экспорт списка документов в CSV.

**3. Нефункциональные требования**

* **Интерфейс**: Простой и интуитивно понятный, без сложных анимаций.
* **Производительность**: Система должна работать с 10–20 пользователями без задержек.
* **Безопасность**:
  + Хранение паролей в хэшированном виде (BCrypt).
  + Защита от базовых уязвимостей (например, SQL-инъекции).
* **Логирование**: Запись действий пользователей (создание/изменение документов).

**4. Архитектура системы**

* **Монолитная архитектура**: Единое приложение с модульной структурой.
* **Слои приложения**:
  + Контроллеры (REST API).
  + Сервисный слой (бизнес-логика).
  + Репозитории (доступ к данным).
  + Модель данных (JPA-сущности).

**5. Технические требования к реализации**

* **Backend**:
  + Spring Boot 3.x.
  + ORM: Hibernate (Spring Data JPA).
  + Тестирование: JUnit (базовые тесты).
* **Frontend**:
  + HTML-страница с JavaScript.
  + Bootstrap для стилизации.
  + React.
* **API**:
  + Документация через комментарии или Swagger (опционально).
* **База данных**:
  + H2 для локальной разработки или PostgreSQL.
  + Простая схема: таблицы для пользователей, документов и файлов.