Название программы: ДСК

Название команды: X-NET

Team lead (ФИО, tg): Михайловская Мария Александровна @mshlik

Ментор (ФИО, tg): Тихонов Федор Андреевич @fedoska\_t

Паспорт проекта

*Нейросеть для оформления текстов работ по ГОСТу*

1. Общая информация

* **Краткое описание проекта**:  
  Разработать нейросеть (в формате веб-приложения для удобства пользователя), которая будет способна оформить текстовые работы (выпускные квалификационные, курсовые, научно-исследовательские) в соответствии с требованиями ГОСТа
* **Команда**:  
  Михайловская Мария Александровна - Teamlead/Frontend-разработчик

Бокарев Никита Александрович - DevOps

Ефременко Кирилл Андреевич - DevOps

Жданович Елизавета Таймуразовна - UX/UI-дизайнер

Мамонтов Никита Андреевич – Backend- разработчик/ Тестировщик

Маркин Роман Владимирович - Frontend-разработчик

Нургалиев Даниэль Ильдарович- Backend-разработчик/ Архитектор

Соловьева Надежда Сергеевна - ML-инженер

Федорова Екатерина Васильевна - ML-инженер/ Технический писатель

Явметдинов Максим Русланович - Backend-разработчик/Архитектор

2. Цель проекта

* **Цель проекта**:  
  Разработать нейросеть (в формате веб-приложения для удобства пользователя), которая будет способна оформить текстовые работы (выпускные квалификационные, курсовые, научно-исследовательские) в соответствии с требованиями ГОСТа
* **Ожидаемые результаты**:  
  Сайт, который будет способен оформить текстовые работы в соответствии с требованиями ГОСТа

3. Задачи и процесс работы

*Для данного пункта подготовьте к показу свой трекер задач. В трекере должны быть зарегистрированы все участники команды и распределены роли. У задач должны быть статусы и дедлайны выполнения*

* **К работе (только цифрами)**:
  + 5.
  + 3.
  + 2.

4. Прогресс и результаты

* **Текущий статус проекта**:  
  *Проект продвигается с установленными дедлайнами, зарегистрировались на кейс ВК*
* **Достигнутые результаты по задачам**:  
  *Выполнены первые прототипы дизайнов, странницы входа, бдшка, архитектура, нормальное возвращаемый файл от нашего сервиса при состоянии скачать файл*
* **Риски и препятствия**:  
  *Технические: несоответствие стандартам, высокая серверная нагрузка.*

*Организационные: нехватка времени, ошибок или компетенций команды*

*Правовые: изменения ГОСТ, авторские права.*5. Ресурсы и материалы проекта

* **Используемые инструменты и технологии**:  
  *VS Code, Docker, Java, Python, JSX, TSX, React, Next.js, серверы, библиотеки (TensorFlow, Python-docx), тестовые данные.*
* **Ссылки на внешние ресурсы**:  
  [*https://github.com/xyle-net/VK\_DPP*](https://github.com/xyle-net/VK_DPP)

<https://github.com/users/xyle-net/projects/2>

https://docs.google.com/document/d/1RsHiSfvIojDpB4eFSukiCTiaAGRE0zRfL72Mqq9n\_wY/edit?tab=t.0

* **Данные**

Пока отсутствуют, в процессе создания

6. Комментарии и мысли команды

* **Комментарии**:  
  *Все идет с установленными дедлайнами, к сожалению, выяснили, что вк датасет не предоставляет, поэтому приходится создавать его самим*

**Исследование и планирование (сейчас)**

1. **Сбор требований:**
   * Изучение ГОСТов, связанных с оформлением текстовых работ
   * Определение ключевых требований к форматированию (шрифты, отступы, заголовки, списки, таблицы, ссылки и т.д.)
   * Составление технического задания (ТЗ)
   * Поиск похожих сервисов в интернете
   * Изучение отзывов на форумах про проверку ГОСТов
   * Опрос студентов о главных проблемах с форматированием
   * Создание списка самых частых ошибок в оформлении
   * Составление чек-листа для проверки документов
2. **Проектирование архитектуры:**

**Java Backend:**

* + Изучение Spring Boot и его модулей (Security, JPA, Web)
  + Настройка микросервисной архитектуры:
    - Auth Service (JWT, роли пользователей)
    - Document Service (загрузка/выгрузка файлов)
    - Admin Service (мониторинг и управление)
  + Подключение PostgreSQL для хранения данных
  + Настройка RabbitMQ для общения с Python

**Python Backend:**

* + Выбор между FastAPI/Django для ML-сервиса
  + Планирование ML-пайплайна:
    - Парсинг документов
    - Анализ форматирования
    - Генерация отчетов
  + Интеграция с Java через очереди

**Frontend:**

* + Планирование основных компонентов:
    - Страница загрузки файлов
    - Личный кабинет
    - Просмотр результатов

**DevOps:**

* + Настройка Docker для локальной разработки
  + Планирование CI/CD пайплайна
  + Выбор сервиса для хостинга (возможно Heroku)

1. **Сбор данных для ML:**

**Источники данных:**

* + Поиск готовых курсовых/дипломов в университетских репозиториях
  + Сбор методичек и примеров оформления от кафедр
  + Скачивание работ с профильных сайтов
  + Запрос у знакомых студентов их работ

**Создание датасета:**

* + Разделение документов на категории:
    - Курсовые работы
    - Дипломные работы
    - Отчеты по практике
    - Рефераты
  + Создание примеров типичных ошибок:
    - Неправильные отступы
    - Разные шрифты
    - Кривые списки/таблицы
    - Неверная нумерация

**Разметка данных:**

* + Создание простой системы тегов для ошибок
  + Маркировка проблемных мест в документах
  + Составление эталонных примеров оформления
  + Ведение статистики частых нарушений

1. **Дизайн:**

**Анализ популярных сервисов:**

* + Google Docs (система комментариев и совместной работы)
  + Microsoft Office Online (интерфейс загрузки и просмотра)
  + Overleaf (работа с LaTeX документами)
  + Grammarly (подсветка ошибок и интерфейс исправлений)
  + PDF24 (конвертация и работа с PDF)

**Разработка нашего дизайна:**

* + Создание минималистичного интерфейса в Figma:
    - Страница авторизации с формой входа/регистрации
    - Главная страница с drag-and-drop загрузкой
    - История запросов и результатов проверки

**Особенности дизайна:**

* + Использование нейроморфизма в UI элементах
  + Светлая цветовая схема с акцентами
  + Волнистые декоративные элементы
  + Простая и понятная навигация
  + Адаптивность под все устройства

**Разработка MVP**

1. **Frontend:** [2 человека]
   * Базовые компоненты:
     + Авторизация/регистрация
     + Загрузка документов
     + Просмотр результатов
   * Интеграция с API Gateway
2. **Design:** [1 человек]
   * Прототипы интерфейса
   * Дизайн компонентов
   * Адаптивная верстка
3. **Java Core Services:** [1 человек]
   * Auth Service:
     + JWT авторизация
     + Управление пользователями
   * Doc Service:
     + Загрузка файлов
     + Валидация документов
     + Очередь на обработку
   * Admin Service:
     + Сбор метрик
     + Логирование операций
     + API для аналитики
4. **Python Backend:** [4 человека]
   * Doc Validation Service:
     + Проверка форматирования
     + Генерация отчетов
   * ML Queue Service:
     + Управление очередью задач
     + Распределение нагрузки
   * ML ГОСТ Service:
     + Базовая модель проверки
     + Обработка документов
   * Analytics Service:
     + Сбор данных с сервисов
     + Построение графиков
     + Анализ метрик
   * Test Service:
     + Интеграционные тесты
     + Нагрузочное тестирование
     + Автоматизация тестирования
5. **DevOps:** [2 человека]
   * Контейнеризация:
     + Docker compose для сервисов
     + Настройка volumes
     + Сеть контейнеров
   * Мониторинг:
     + Prometheus + Grafana
     + ELK Stack для логов
     + Алерты и дашборды
   * CI/CD:
     + GitHub Actions
     + Автоматический деплой
   * Безопасность:
     + SSL/TLS сертификаты
     + Настройка Nginx
     + Бэкапы данных

**Развитие проекта**

1. **Оптимизация производительности:**
   * Фронтенд:
     + Оптимизация загрузки страниц
     + Работа с большими файлами
     + Предпросмотр документов
   * Бэкенд:
     + Кэширование результатов
     + Оптимизация БД (индексы)
     + Параллельная обработка
   * ML:
     + Ускорение обработки документов
     + Оптимизация моделей
2. **Улучшение функционала:**
   * Интерфейс:
     + Умные подсказки
     + Детальные сообщения об ошибках
     + Поддержка LaTeX и Markdown
   * ML модели:
     + Обработка сложных элементов
     + Дообучение на новых примерах
     + Оценка качества форматирования
   * Новые фичи:
     + Проверка на плагиат
     + Автоматическое оглавление
3. **Стабильность системы:**
   * Расширенное тестирование
   * Обработка ошибок
   * Система бэкапов
   * Мониторинг и алерты
4. **Продвижение проекта:**
   * Презентация на кафедре
   * Демо для других факультетов
   * Публикация в студенческих группах
   * Сбор отзывов от преподавателей
   * Участие в конкурсе проектов