

АВТОМАТИЧЕСКОЕ РЕВЬЮ

Решение подготовлено командой
Буено

13.09.2025

Команда



Хомичев Денис
tg: @EverLieF

**Капитан
Разработчик**



Анпилов Кирилл
tg: @anpilove

Разработчик



Рогоза Ярослав
tg: @ne_onnn

Разработчик



Бегушева Элина
tg: @begunoooochek

**Продуктовый
менеджер**

Идея и цели

«**Автоматическое ревью**» — функция / ассистент для самопроверки проектов студентами перед отправкой на ревью.

Цели:

- снизить стресс студентов (быстрая самопроверка)
- уменьшить нагрузку ревьюеров (убрать типовые ошибки)

Преанализ

Схожие кейсы



Ключевые выводы

- Большинство платформ используют автотесты и линтеры для проверки корректности и стиля.
- ИИ-помощники (LLM) появляются как дополнительный слой — для генерации комментариев и подсказок.
- Оптимальная схема — гибридный подход: автотесты + статический анализ + AI-ассистент.
- Решения снижают нагрузку на ревьюеров, но требуют контроля (human-in-the-loop).

Проверка гипотез

1. Студенты хотят получать быстрый первичный фидбек до человеческой проверки
2. Студенты чаще всего допускают базовые ошибки (синтаксис, структура кода, автотесты)

Взгляд

Скриншоты мобильного приложения с вопросами и вариантами ответов:

- Вопрос: Обучались ли вы в онлайн (курсы, практикумы) за последний год?
Варианты: Да, Нет
Кнопка: Далее
- Вопрос: Нужна ли быстрая проверка работы до отправки преподу?
Варианты: Да, очень нужна, Иногда пригодится, Нет, не нужна
Кнопка: Далее
- Вопрос: Что чаще всего мешает вам сдать работу с первого раза?
Варианты: Работа не запускалась / не открывалась, Ошибки в оформлении, Неправильная структура, Опечатки и невнимательность, Другое
Кнопка: Далее
- Вопрос: Как относитесь к быстрой проверке (с пом. ИИ) перед преподом?
Варианты: Доверяю, это удобно, Можно, но не всегда, Не доверяю ИИ
Кнопка: Далее

Нужна ли быстрая проверка работы до отправки преподу?

Да, очень нужна Иногда пригодится Нет, не нужна



Что чаще всего мешает вам сдать работу с первого раза?

Опечатки и невнимательность Работа не запускалась / не открывалась
Ошибки в оформлении Неправильная структура Другое



Backend

Сервер на Express (Node.js)

Основные возможности:

- принимает загрузку проектов
- хранит архивы и истории проверок
- запускает автоматические проверки (линт, тесты, анализ структуры - "Статические" проверки staticCheck.ts, Pytest)
- проверки через ЛЛМ от Яндекс
- формирует отчеты и сохраняет их в виде артефактов
- отдает данные Фронтенду через REST API

Хранение в формате JSON и в паках на диске



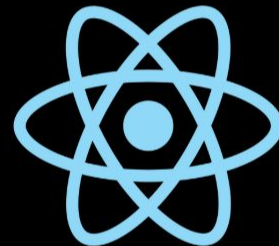
Frontend

Построен на React + TypeScript

Интерфейс:

- загрузка проекта (ZIP или ссылка на GitHub)
- просмотр файлов с подсветкой кода
- добавление и обсуждение комментариев
- отчеты : студент видит оценку и рекомендации, ревьюер - ошибки по файлам

Режим моков (MSW) - для работы без настоящего сервера



TS TypeScript

Демо

Метрики



Время обработки одного проекта ~60 секунд



Средняя стоимость 1 отчета ~ 100 рублей



Доля правильно найденных некорректных работ из тестовых данных ~80%

Дальнейшие улучшения



Добавление ХАРД-режима проверки



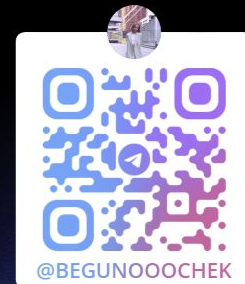
Добавление возможности дообучения модели с помощью системы “Лайк / Дизлайк” для оценки качества отчета

ССЫЛКА НА ПРОТОТИП



<https://reviewr.ru>

КОНТАКТЫ



13.09.2025