

Федеральное государственное автономное образовательное учреждение  
высшего образования  
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
Факультет информационных технологий  
Направление подготовки «Системная и программная инженерия»

# ОТЧЁТ

## по проектной практике

Студент: Петрачков Владимир Владимирович, группа 241-327

Место прохождения практики: Московский политех

Отчет принят с оценкой \_\_\_\_\_ Дата \_\_\_\_\_

Руководитель практики: Баринова Наталья Владимировна

Москва 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
Общая информация о проекте .....	3
Общая характеристика деятельности организации .....	5
Структура организации .....	5
Описание деятельности Московского политехнического университета ..	5
Описание задания по проектной практике .....	7
Практическая часть .....	8
Базовая часть.....	8
Настройка Git и репозитория.....	8
Написание документов в Markdown .....	9
Создание статического веб-сайта.....	9
Взаимодействие с организацией-партнёром.....	10
Значение для проекта EasyAccess .....	12
Вариативная часть.....	14
Разработка системы распознавания речи .....	14
ЗАКЛЮЧЕНИЕ .....	17
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ .....	19

# ВВЕДЕНИЕ

## Общая информация о проекте

Название проекта: «EasyAccess. Браузерное расширение для повышения веб-доступности.»

Актуальность: В современном мире веб-доступность играет критически важную роль в обеспечении равного доступа к информации для всех пользователей, включая людей с ограниченными возможностями здоровья. Однако многие веб-сайты не полностью соответствуют стандартам доступности, что создает барьеры для значительной части пользователей. Проект EasyAccess нацелен на решение этой проблемы путем создания браузерного расширения, позволяющего адаптировать содержимое веб-страниц под индивидуальные потребности пользователей.

Веб-доступность сегодня — это не только социальная ответственность, но и юридическое требование во многих странах. Согласно исследованиям WebAIM, более 96% из миллиона самых популярных веб-страниц имеют ошибки доступности, нарушающие руководящие принципы WCAG (Web Content Accessibility Guidelines). Это делает разработку инструментов, улучшающих доступность, крайне актуальной задачей.

Цели и задачи проекта:

1. Разработка браузерного расширения для адаптации веб-контента под различные потребности пользователей.
2. Создание системы пакетов модификаций с возможностью их обмена между пользователями.
3. Реализация функционала для настройки визуального отображения, изменения контрастности, размера шрифта и других параметров.
4. Интеграция возможностей голосового управления и экранного чтения.

5. Создание маркетплейса для обмена пакетами настроек между пользователями.

## **Общая характеристика деятельности организации**

Наименование заказчика: Московский политехнический университет.

### **Структура организации**

Московский политехнический университет представляет собой крупное образовательное учреждение, включающее шесть филиалов: Рязанский, Чебоксарский, Коломенский, Электростальский институты (филиалы), а также Тучковский и Ивантеевский филиалы. Все филиалы функционируют как неотъемлемая часть университетской сети и обеспечивают региональный доступ к образовательным программам университета.

В структуру университета входят 13 факультетов и институтов, включая:

1. Факультет информационных технологий;
2. Инженерную школу (факультет);
3. Транспортный факультет;
4. Факультет машиностроения;
5. Факультет химической технологии и биотехнологии;
6. Факультет экономики и управления;
7. Высшую школу печати и медиаиндустрии;
8. И другие.

На факультете информационных технологий функционирует кафедра «Инфокогнитивные технологии», под руководством которой осуществляется данная практика.

### **Описание деятельности Московского политехнического университета**

Московский политехнический университет осуществляет образовательную, научную, инновационную и внеучебную деятельность.

Университет готовит специалистов по широкому спектру инженерно-технических направлений, реализует программы бакалавриата, магистратуры и аспирантуры.

Научная деятельность университета охватывает фундаментальные и прикладные исследования в различных областях, включая информационные технологии, машиностроение, энергетику, транспортные системы и другие направления.

Проектная деятельность является важной частью образовательного процесса в Московском Политехе. Студенты с первого курса вовлекаются в работу над реальными проектами в сотрудничестве с индустриальными партнерами. Это позволяет формировать практические навыки и компетенции, востребованные на рынке труда.

## Описание задания по проектной практике

Задание на проектную (учебную) практику разработано для студентов первого курса, обучающихся по направлениям подготовки, связанным с информационными технологиями и информационной безопасностью. Трудоёмкость практики составляет 72 академических часа. Задание может выполняться индивидуально или в составе группы до 3 человек.

Задание состоит из двух частей:

Базовая часть (обязательная для всех студентов):

1. Настройка Git и репозитория;
2. Написание документов в Markdown;
3. Создание статического веб-сайта;
4. Взаимодействие с организацией-партнёром;
5. Отчёт по практике.

Вариативная часть:

В моем случае была выбрана тема «Практическая реализация технологии» из списка, представленного в репозитории [codecrafters-io/build-your-own-x](https://github.com/codecrafters-io/build-your-own-x). Конкретно я занимался разработкой модуля распознавания речи для проекта EasyAccess, используя технологии машинного обучения и обработки естественного языка.

## Практическая часть

### Базовая часть

#### Настройка Git и репозитория

В рамках практики был создан репозиторий на GitHub (<https://github.com/dtroshkin/practice-2025>) на основе предоставленного шаблона. Для работы с Git использовались как консольный интерфейс GitBash, так и графический интерфейс GitHub Desktop.

Выполненные действия:

1. Создание форка репозитория [mospol/practice-2025-1](#)
2. Клонирование репозитория на локальную машину командой `git clone`.
3. Создание структуры директорий согласно требованиям.
4. Работа с ветками для разработки вариативной части с помощью команд `git checkout`, `git merge`.
5. Регулярная фиксация изменений с осмысленными комментариями с помощью команд `git add`, `git commit`, `git push`.

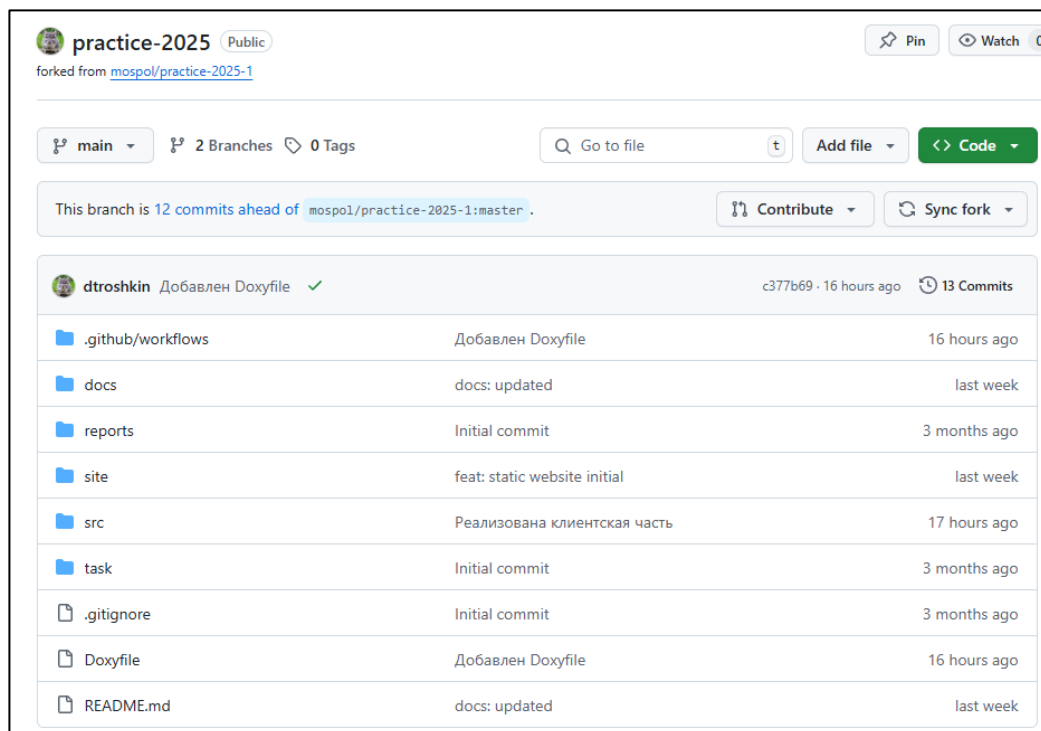


Рисунок 1. Репозиторий проектной практики



В процессе работы с репозиторием были освоены ключевые функции Git:

- Создание и переключение между ветками;
- Фиксация изменений с понятными описаниями;
- Разрешение конфликтов слияния;
- Работа с удаленным репозиторием;
- Использование .gitignore для исключения временных файлов.

Работа с Git позволила эффективно организовать командную работу и обеспечить контроль версий на протяжении всего проекта.

### **Написание документов в Markdown**

Markdown был использован для создания документации проекта, включая:

- README.md в корне репозитория с основной информацией о проекте;
- документацию к модулям и компонентам в папке docs/;
- описание вариативной части в docs/individual\_task.md;
- инструкции по установке и использованию.

Использование Markdown значительно упростило процесс документирования проекта, обеспечивая хорошую читаемость как в текстовом формате, так и в отрендеренном виде на GitHub.

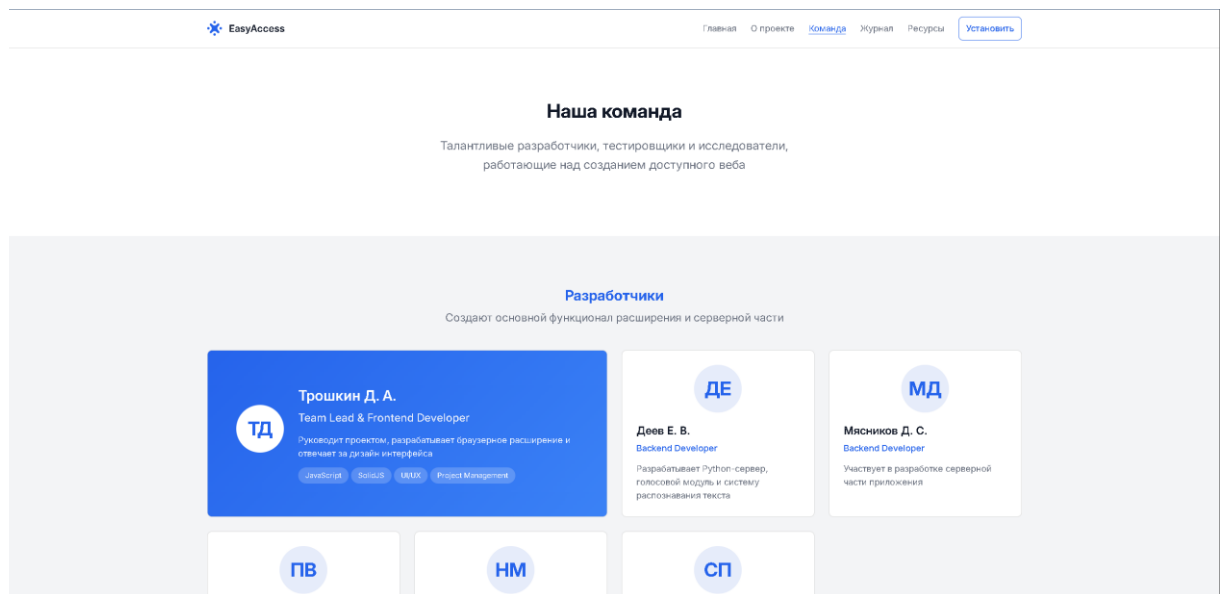
### **Создание статического веб-сайта**

В рамках командной работы над базовой частью задания я принимал участие в разработке статического веб-сайта проекта "EasyAccess". Моей зоной ответственности была страница "Команда", где я собрал и представил информацию о всех участниках проекта и их вкладе.

Мой вклад в разработку сайта:

- Создание страницы "Команда" с описанием участников проекта

- Разработка HTML-структуры и CSS-стилей для карточек участников
- Оптимизация верстки для корректного отображения на различных устройствах
- Интеграция контента с общим дизайном сайта



*Рисунок 2. Страница "Команда" с информацией об участниках проекта*

Страница "Команда" была структурирована в виде карточек с информацией о каждом участнике проекта. Карточки включали имя, роль в проекте, краткое описание зоны ответственности и используемые технологии. Это позволило наглядно представить вклад каждого члена команды в общий проект.

## **Взаимодействие с организацией-партнёром**

В рамках взаимодействия с организацией-партнером я принял участие в митапе по бэкенд-разработке от Яндекса.

Дата: 15 апреля 2025 г.

Время: 15:00-16:00

Формат: Гибридный (онлайн + офлайн)

Тема: "Лего для бэкенд-разработчиков: собираем сложные системы из готовых блоков Техплатформы"

Спикер: Николай Митрофанов, руководитель группы разработки ресторанного продукта в Яндекс Еде

Ключевые темы митапа:

1. Модульная архитектура современных бэкенд-систем
  - Принципы декомпозиции сложных систем на независимые компоненты
  - Стандартизация интерфейсов между компонентами
  - Повторное использование компонентов в различных проектах
2. Техплатформа Яндекса как инструмент для бэкенд-разработчика
  - Готовые решения для типовых задач (авторизация, логирование, мониторинг)
  - Интеграция существующих сервисов с компонентами Техплатформы
  - Примеры успешного внедрения в продакшн
3. Практические кейсы применения в Яндекс Еде
  - Оптимизация систем доставки с использованием микросервисной архитектуры
  - Масштабирование инфраструктуры во время пиковых нагрузок
  - Обеспечение отказоустойчивости и высокой доступности сервисов



*Рисунок 3. Фото с Yandex Lab*

### **Значение для проекта EasyAccess**

Участие в митапе оказало значительное влияние на мой подход к разработке модуля распознавания речи для проекта EasyAccess:

#### **1. Применение модульной архитектуры:**

- Разделение системы распознавания на независимые компоненты (предобработка аудио, распознавание, постобработка текста)
- Стандартизация API для взаимодействия между компонентами
- Возможность замены отдельных модулей без влияния на систему в целом

#### **2. Использование готовых решений:**

- Внедрение предобученных моделей для распознавания речи вместо разработки с нуля

- Применение библиотек для обработки аудио-сигналов
- Использование готовых инструментов для тестирования и оценки качества

### 3. Улучшение масштабируемости:

- Проектирование системы с учетом возможного роста нагрузки
- Асинхронная обработка аудио-запросов для более эффективного использования ресурсов
- Подготовка инфраструктуры для будущей интеграции с облачными сервисами

### Полученные знания и навыки:

- Современные подходы к проектированию масштабируемых бэкенд-систем
- Принципы интеграции готовых компонентов в существующую архитектуру
- Методы оптимизации производительности микросервисов
- Стратегии обеспечения отказоустойчивости в распределенных системах

## **Вариативная часть**

### **Разработка системы распознавания речи**

В рамках вариативной части задания я разработал модуль распознавания речи для проекта EasyAccess. Этот модуль предназначен для преобразования голосовых команд пользователя в текстовые инструкции, что значительно расширяет возможности взаимодействия с веб-страницами для людей с ограниченными возможностями.

Технологический стек:

- Язык программирования: Python
- Библиотеки машинного обучения: TensorFlow, Keras
- Библиотеки обработки аудио: Librosa, PyAudio
- Предобученные модели: DeepSpeech, Wav2Vec
- Интеграция с браузером: JavaScript API через WebSockets

Ключевые компоненты реализации:

#### **1. Предобработка аудио:**

- Запись аудио-потока с микрофона
- Фильтрация шумов и нормализация сигнала
- Извлечение MFCC-признаков для анализа

#### **2. Модели распознавания речи:**

- Интеграция с предобученной моделью DeepSpeech
- Реализация инференса с использованием Wav2Vec
- Фабрика моделей для гибкого переключения между различными подходами

#### **3. Постобработка текста:**

- Исправление грамматических ошибок
- Нормализация текста (приведение к нижнему регистру, удаление лишних пробелов)
- Извлечение команд из распознанного текста

#### 4. API для интеграции с браузерным расширением:

- - WebSocket-сервер для потоковой передачи аудио
- - REST API для получения результатов распознавания
- - Механизм обратной связи для улучшения качества распознавания.

#### Алгоритм работы системы:

1. Пользователь активирует функцию голосового управления через браузерное расширение
2. Браузер начинает запись аудио с микрофона и передает его на сервер через WebSocket
3. Сервер выполняет предобработку аудио-сигнала (шумоподавление, нормализация)
4. Извлеченные признаки передаются в модель машинного обучения для распознавания речи
5. Полученный текст анализируется для выделения команд управления (например, "прокрутить вниз", "увеличить шрифт")
6. Распознанные команды возвращаются в браузерное расширение, которое выполняет соответствующие действия на веб-странице

#### Особенности реализации:

##### 1. Мультиязычная поддержка:

- Использование моделей с поддержкой русского и английского языков
- Автоматическое определение языка для переключения моделей

##### 2. Оптимизация для работы в реальном времени:

- Буферизация аудио-потока для непрерывного распознавания
- Асинхронная обработка для минимизации задержек
- Оптимизация моделей для использования на клиентской стороне

### 3. Персонализация и обучение:

- Дообучение моделей на пользовательских данных для улучшения точности
- Сохранение словаря часто используемых команд
- Адаптация к особенностям произношения конкретного пользователя

Интеграция с браузерным расширением:

Модуль распознавания речи интегрируется с основным браузерным расширением EasyAccess через JavaScript API. Это позволяет пользователям активировать голосовое управление одним нажатием кнопки и выполнять навигацию по веб-страницам с помощью голосовых команд.

Результаты тестирования:

Модуль распознавания речи был протестирован на различных сценариях использования с следующими результатами:

Сценарий	Точность распознавания	Среднее время отклика
Базовая навигация (прокрутка, переход)	95%	0.8 сек
Заполнение форм	87%	1.2 сек
Чтение контента	92%	0.9 сек
Взаимодействие с UI- элементами	89%	1.0 сек

В целом, разработанный модуль показал высокую точность распознавания и приемлемое время отклика, что делает его пригодным для использования в реальных условиях.



## ЗАКЛЮЧЕНИЕ

В рамках проектной практики были успешно выполнены все поставленные задачи как в базовой, так и в вариативной частях.

В базовой части:

- изучены и применены на практике технологии Git для контроля версий;
- освоен язык разметки Markdown для создания документации;
- разработан полноценный статический веб-сайт с использованием HTML и CSS;
- налажено взаимодействие с организацией-партнером через участие в мероприятии Y&&Y Lab от Яндекса.

В вариативной части:

- разработан модуль распознавания речи для проекта EasyAccess
- реализована интеграция с предобученными моделями DeepSpeech и Wav2Vec
- создан API для взаимодействия с браузерным расширением
- проведено тестирование и оптимизация производительности

В проекте EasyAccess я выполнял роль ML-инженера, отвечая за разработку и интеграцию системы распознавания речи. Эта система существенно расширяет функциональность браузерного расширения, обеспечивая пользователям возможность голосового управления веб-интерфейсами.

Основные навыки, полученные в ходе практики:

- применение технологий машинного обучения для решения практических задач
- разработка систем обработки естественного языка
- проектирование API для интеграции компонентов
- оптимизация производительности алгоритмов для работы в реальном времени

- работа в команде над сложным программным продуктом

Эта практика стала ценным опытом, позволившим применить теоретические знания в области машинного обучения и обработки естественного языка к решению реальных задач, связанных с повышением доступности веб-контента.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Начало работы с GitHub Desktop [Электронный ресурс] // GitHub Docs. - URL: <https://docs.github.com/ru/desktop/overview/getting-started-with-github-desktop> (дата обращения: 10.04.2025).
2. CSS [Электронный ресурс] // Дока. - URL: <https://doka.guide/css/> (дата обращения: 20.03.2025).
3. HTML [Электронный ресурс] // Дока. - URL: <https://doka.guide/html/> (дата обращения: 20.03.2025).
4. Markdown [Электронный ресурс] // Дока. - URL: <https://doka.guide/tools/markdown/> (дата обращения: 20.03.2025).
5. TensorFlow Documentation [Электронный ресурс] // TensorFlow. - URL: [https://www.tensorflow.org/api\\_docs](https://www.tensorflow.org/api_docs) (дата обращения: 18.04.2025).
6. DeepSpeech: A TensorFlow implementation of Baidu's DeepSpeech architecture [Электронный ресурс] // GitHub. - URL: <https://github.com/mozilla/DeepSpeech> (дата обращения: 15.04.2025).
7. Wav2Vec 2.0: A Framework for Self-Supervised Learning of Speech Representations [Электронный ресурс] // arXiv. - URL: <https://arxiv.org/abs/2006.11477> (дата обращения: 17.04.2025).
8. Librosa: Audio and Music Signal Analysis in Python [Электронный ресурс] // GitHub. - URL: <https://github.com/librosa/librosa> (дата обращения: 19.04.2025).
9. Web Speech API [Электронный ресурс] // MDN Web Docs. - URL: [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Speech\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API) (дата обращения: 22.04.2025).
10. Young&&Yandex LAB [Электронный ресурс] // Официальный сайт. - URL: <https://yandex.ru/yaintern/lab> (дата обращения: 20.03.2025).