|  |
| --- |
| Федеральное государственное автономное образовательное учреждение высшего образования  «МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  Факультет информационных технологий  Направление подготовки «Системная и программная инженерия» |
| ОТЧЁТ  по проектной практике |
| Студент: Мясников Дмитрий Сергеевич, группа 241–327  Место прохождения практики: Московский политех  Отчет принят с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Руководитель практики: Баринова Наталья Владимировна  Москва 2025 |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc198734748)

[Общая информация о проекте 3](#_Toc198734749)

[Описание задания по проектной практике 4](#_Toc198734750)

[ПРАКТИЧЕСКАЯ ЧАСТЬ 5](#_Toc198734751)

[Базовая часть 5](#_Toc198734752)

[Написание документов в Markdown 6](#_Toc198734753)

[Создание статического веб-сайта 6](#_Toc198734754)

[Взаимодействие с организацией-партнёром 7](#_Toc198734755)

[Значение для проекта Easy Access 9](#_Toc198734756)

[Вариативная часть 10](#_Toc198734757)

[ЗАКЛЮЧЕНИЕ 14](#_Toc198734758)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 16](#_Toc198734759)

# 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.

## 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):
  + настройка Git и репозитория,
  + написание документов в Markdown,
  + создание статического веб-сайта,
  + взаимодействие с организацией-партнёром,
  + отчёт по практике;
* вариативная часть.

В моем случае была выбрана тема «Практическая реализация технологии» из списка, представленного в репозитории. Конкретно я реализовал клиентскую часть чат-приложения с использованием C++ и Qt Framework.

# ПРАКТИЧЕСКАЯ ЧАСТЬ

## Базовая часть

В рамках практики был создан репозиторий на GitHub 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 c помощью команд git checkout, git merge.
5. Регулярная фиксация изменений с осмысленными комментариями с помощью команд git add, git commit, git push.

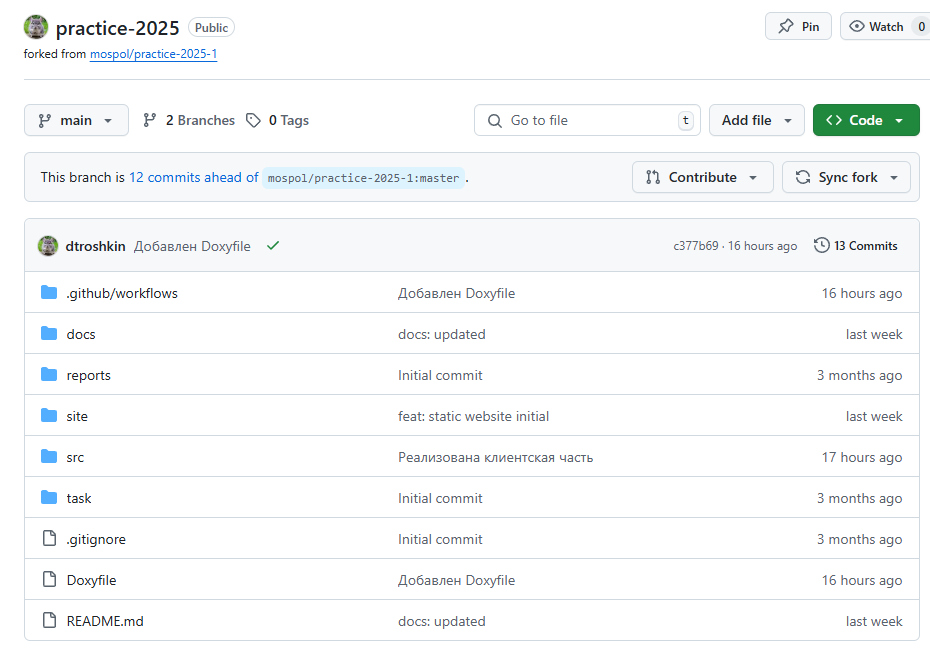


Рисунок 1. Репозиторий проектной практики

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 "Ключевая архитектура" в разделе "Журнал".

Мой вклад в разработку сайта:

* Создание страницы с описанием архитектурных решений проекта
* Интеграция разработанного контента с общим дизайном сайта
* Оптимизация CSS-стилей для обеспечения кроссбраузерной совместимости
* Тестирование адаптивности на различных устройствах

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.Рисунок 2. Страница "Журнал" с моей публикацией

Благодаря совместной работе над сайтом, удалось создать единообразный дизайн и структуру, обеспечивающие хороший пользовательский опыт и простоту навигации. Все страницы сайта имеют адаптивный дизайн, что позволяет корректно отображать содержимое на различных устройствах.

## Взаимодействие с организацией-партнёром

В рамках взаимодействия с организацией-партнером я принял участие в мероприятии Y&&Y Lab, организованном компанией Яндекс.

Дата: 20 марта 2025 г.

Место проведения: БЦ Морозов, Москва

Организатор: Яндекс (Young&&Yandex)

Формат участия: Очное присутствие



Рисунок 3. Фотография с мероприятия

Описание мероприятия:

Young&&Yandex Lab представляет собой масштабный карьерный фестиваль Яндекса, направленный на развитие молодых специалистов в IT-сфере. Мероприятие включало различные треки по направлениям:

* Бэкенд
* Фронтенд
* Мобильная разработка
* Data Science
* DevOps
* Нетехнические специальности

Ключевые активности фестиваля:

1. Презентация проектов для молодёжи

Были представлены новые возможности для студентов и начинающих специалистов, включая:

* Летние школы 2025
* IT-стажировки в Яндексе
* Business Camp для нетехнических направлений
* Тренажёр CodeRun для прокачивания навыков
* Баттл вузов 2025

2. Митап по фронтенд-разработке

В рамках митапа выступил Алексей Ершков, руководитель фронтенда i18n Плюса Фантеха, с докладом "Что может пойти не так с простым компонентом в больших проектах?".

Основные темы доклада:

* Разработка универсального переключателя языков для нескольких сервисов Яндекса
* Проблемы адаптации интерфейсов под различные языки
* Аспекты доступности мультиязычных компонентов
* Технические решения для унифицированного интерфейса

## Значение для проекта Easy Access

Участие в этом мероприятии оказало значительное влияние на мою работу над бэкендом проекта EasyAccess:

1. Понимание масштабируемой архитектуры:

Опыт Яндекса в создании унифицированных компонентов для различных сервисов помог в проектировании API и структуры бэкенда нашего проекта.

2. Применение современных подходов к разработке:

Презентация Яндекса о микросервисной архитектуре помогла лучше понять, как организовать бэкенд-инфраструктуру EasyAccess для масштабирования в будущем.

3. Улучшение взаимодействия между фронтендом и бэкендом:

Понимание проблем, с которыми сталкиваются фронтенд-разработчики, помогло мне улучшить дизайн API и документацию для упрощения интеграции.

Полученные знания и навыки:

* Современные подходы к проектированию масштабируемой архитектуры
* Техники оптимизации производительности бэкенд-систем
* Методы обеспечения безопасности и защиты данных
* Инструменты мониторинга и анализа производительности

## Вариативная часть

В рамках вариативной части задания я реализовал клиентскую часть чат-приложения на базе C++ и Qt Framework. Это приложение является частью совместного проекта команды, где каждый участник отвечал за определенную часть системы.

Технологический стек:

* Язык программирования: C++
* GUI фреймворк: Qt 6.8
* Сетевое взаимодействие: Qt Network (QTcpSocket)
* Структура данных: JSON (QJsonDocument, QJsonObject)
* Сборка проекта: CMake
* Тестирование: Qt Test Framework

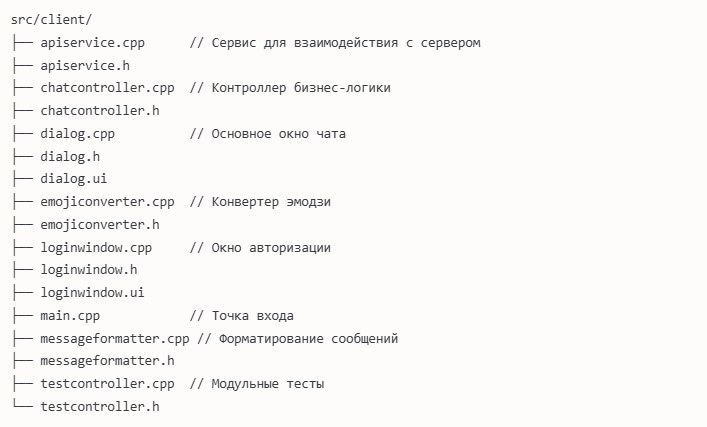


Рис.3 Структура клиентской части

Ключевые компоненты реализации:

1. ApiService - класс для взаимодействия с сервером:

* Реализован как Singleton для обеспечения единого доступа
* Обеспечивает подключение к серверу по TCP
* Отправляет и принимает JSON-сообщения
* Обрабатывает сетевые ошибки и отключения

2. ChatController - контроллер бизнес-логики:

* Обеспечивает связь между UI и сетевым слоем
* Управляет авторизацией и аутентификацией
* Обрабатывает сообщения и обновления состояния

3. Графический интерфейс - окна для взаимодействия с пользователем:

* LoginWindow - форма авторизации и регистрации
* Dialog - основное окно чата с историей сообщений и списком пользователей

4. Вспомогательные классы:

* MessageFormatter - для форматирования текста сообщений
* EmojiConverter - для конвертации текстовых эмодзи в Unicode-символы

Протокол взаимодействия с сервером:

* Клиент и сервер обмениваются сообщениями в формате JSON.

Функциональные возможности реализованного клиента:

* Подключение к серверу по IP-адресу и порту
* Регистрация новых пользователей и авторизация существующих
* Отправка и получение текстовых сообщений
* Форматирование сообщений (поддержка жирного и курсивного текста)
* Автоматическая конвертация текстовых эмодзи в Unicode-символы
* Отображение списка пользователей онлайн
* Просмотр истории сообщений
* Уведомления о входе и выходе пользователей

Особенности реализации:

1. Архитектура Model-View-Controller:

- Четкое разделение логики и представления

- Независимость UI от сетевого слоя

- Возможность переиспользования компонентов

2. Потокобезопасность:

- Использование мьютексов для защиты критических секций

- Обработка сигналов и слотов Qt для безопасной передачи данных между потоками

3. Обработка ошибок:

- Детальное логирование с использованием qDebug()

- Информативные сообщения об ошибках для пользователя

- Автоматические попытки переподключения при потере соединения

4. Модульное тестирование:

- Тесты для MessageFormatter и EmojiConverter

- Использование QTest для автоматизации тестирования

Взаимодействие с серверной частью:

* Клиентская часть разрабатывалась в тесном сотрудничестве с разработчиками серверной части. Мы совместно спроектировали протокол обмена данными и API, что позволило обеспечить бесперебойную работу всей системы.

# 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
* Реализована часть статического веб-сайта, посвященная архитектурным решениям проекта
* Налажено взаимодействие с организацией-партнером через участие в мероприятии Y&&Y Lab от Яндекса

В вариативной части:

* Реализована клиентская часть чат-приложения на C++ с использованием Qt Framework
* Создана архитектура, основанная на паттерне Model-View-Controller
* Разработан удобный пользовательский интерфейс для авторизации и обмена сообщениями
* Реализованы дополнительные функции, такие как форматирование сообщений и поддержка эмодзи

В проекте EasyAccess я выполнял роль backend-разработчика, отвечая за создание серверной части на Golang с использованием фреймворка Chi и ORM Bun. Моя работа включала:

* Проектирование архитектуры API
* Реализацию маршрутизации и обработки запросов
* Взаимодействие с базой данных
* Обеспечение безопасности и производительности системы

Основные навыки, полученные в ходе практики:

* Углубленные знания C++ и Qt Framework
* Опыт разработки сетевых приложений
* Навыки проектирования JSON API
* Опыт работы с Git в команде
* Разработка пользовательских интерфейсов
* Применение паттернов проектирования в реальных проектах

Эта практика стала ценным опытом, позволившим применить теоретические знания в реальных проектах и подготовиться к дальнейшей профессиональной деятельности в сфере разработки программного обеспечения.

# 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. Qt Documentation [Электронный ресурс] // Qt. - URL: https://doc.qt.io/ (дата обращения: 15.04.2025).

5. Markdown [Электронный ресурс] // Дока. - URL: https://doka.guide/tools/markdown/ (дата обращения: 20.03.2025).

6. Шлее М. Qt 5.10. Профессиональное программирование на C++ [Текст] / М. Шлее. – СПб.: БХВ-Петербург, 2018. – 1072 с.

7. Документация Go [Электронный ресурс] // The Go Programming Language. - URL: https://golang.org/doc/ (дата обращения: 15.04.2025).

8. Chi - легковесный маршрутизатор для Go [Электронный ресурс] // GitHub. - URL: https://github.com/go-chi/chi (дата обращения: 18.04.2025).

9. Bun - SQL ORM для Go [Электронный ресурс] // GitHub. - URL: https://github.com/uptrace/bun (дата обращения: 20.04.2025).

10. Build Your Own X [Электронный ресурс] // GitHub. - URL: https://github.com/codecrafters-io/build-your-own-x (дата обращения: 12.04.2025).

11. Созданный репозиторий [Электронный ресурс] // Github – URL: https://github.com/dtroshkin/practice-2025 (дата обращения 23.05.2025)