Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Кафедра «Информатики и информационных технологий»

Направление подготовки/ специальность: \_ИСИТ/АСОИиУ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ОТЧЕТ

по проектной практике

Студент: \_*Алиханов Богдан Тагирович*\_\_\_\_\_\_\_ Группа: 241-337\_\_\_\_\_\_\_\_\_

Место прохождения практики: Московский Политех, кафедра ИиИт\_\_\_\_\_\_\_

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

Руководитель практики: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2025

# **Отчёт по проектной практике**

# **ВВЕДЕНИЕ**

## Общая информация о проекте:

Название проекта: АИС для транспортной компании — Telegram-бот «Дневник в Телеграмме»  
  
Проект направлен на разработку Telegram-бота для фиксации задач и учёта времени сотрудников, с возможностью генерации аналитических отчётов. Работа выполнялась в составе проектной группы по дисциплине «Проектная деятельность» под руководством Володиной О.В.

## Цели и задачи проекта

Цель: создание Telegram-бота «Дневник в Телеграмме» для учёта пользовательской активности и автоматического формирования отчётов о затраченном времени.  
  
Задачи:  
- Анализ требований  
- Проектирование командной системы  
- Разработка структуры базы данных  
- Реализация логики FSM  
- Генерация отчётов  
- Проведение тестирования  
- Документирование

**Общая характеристика деятельности организации (заказчика проекта)**

Наименование заказчика: Общество с ограниченной ответственностью «А+А Эксист-Инфо»

Сфера деятельности: Разработка корпоративных информационных систем (ИС) для логистики, складского хранения и электронной коммерции.

ООО «А+А Эксист-Инфо» более 25 лет специализируется на создании распределённых сетевых систем и комплексных программных решений. Компания разрабатывает ИС масштаба предприятия, которые применяются:

в логистике перевозок и складского хранения для транспортных компаний;

в сопровождении онлайн-магазинов с каталогами свыше 100 млн позиций.

**Ключевые продукты:**

ТРАНСНЭТ — система автоматизации перевозок, снабжения и складов на базе распределённой БД NETDBS, обеспечивающая устойчивость, масштабируемость и высокую производительность;

Exist.ru — комплекс программ для управления заказами, каталогами, клиентами, складскими и финансовыми процессами в сфере торговли автозапчастями (реализован по модели SaaS).

### ****Описание задания по проектной практике****

В рамках проектной практики выполнялось комплексное задание, включающее обязательную и вариативную части.

**Базовая часть включала:**

* Настройку репозитория и работу с системой контроля версий Git;
* Оформление документации в формате Markdown;
* Создание статического сайта проекта на HTML и CSS с разделами: главная, описание проекта, участники, журнал разработки и ресурсы.

**Вариативная часть заключалась** в индивидуальной разработке консольного приложения на C# — «Мониторинг системных ресурсов».  
Программа реализует отслеживание загрузки процессора, памяти, процессов, логирование и базовый анализ активности. Проект разработан с использованием системных API Windows и архитектуры MVVM, с поддержкой асинхронной и многопоточной обработки данных.

Обе части проекта задокументированы, оформлены и размещены в едином репозитории с сопровождающим веб-сайтом.

## Описание достигнутых результатов по проектной практике

**В рамках проектной практики мной были выполнены следующие работы:**

#### ****1. Разработка статического сайта проекта****

Был создан статический веб-сайт с использованием HTML и CSS, отражающий структуру и содержание командного проекта. Сайт включает в себя следующие разделы:

* **Главная страница** с аннотацией проекта;
* **О проекте** — описание целей, задач и сути разработки;
* **Участники** — вклад каждого члена команды;
* **Журнал** — хронология выполнения этапов проекта (не менее трёх записей);
* **Ресурсы** — полезные ссылки и материалы, включая сведения о заказчике.

Для сайта была реализована:

* **Адаптивная вёрстка** с использованием Flexbox и медиазапросов;
* Единая **система навигации**, доступная на всех страницах;
* Визуальное оформление с использованием шрифтов, цветовых акцентов и стилей;

#### ****2. Разработка C#-приложения «Мониторинг системных ресурсов»****

Проект представляет собой утилиту для мониторинга ресурсов ПК с интерфейсом на базе библиотеки Spectre.Console. Программа отображает в терминале данные о загрузке CPU, памяти, процессов, а также содержит функции анализа и логирования. Применены системные API Windows и .NET. Разработка велась по архитектуре MVVM с асинхронной обработкой.

**Хронология:**  
- 9–13 мая: базовая структура, сбор данных, интерфейс  
- 15–18 мая: добавление фильтрации, анализа, визуальных таблиц  
- 19 мая: тестирование и оптимизация

**Функциональные возможности приложения:**

* Получение и отображение в реальном времени информации о загрузке **центрального процессора (CPU)**;
* Отслеживание использования **оперативной памяти (RAM)**;
* Получение списка **запущенных процессов** с указанием идентификаторов, имени, занимаемой памяти и времени запуска;
* Выделение процессов с высокой нагрузкой для анализа аномалий;
* Ведение **лога событий** (журналирование) для последующего анализа.

**Технические особенности реализации:**

* Приложение построено с использованием **.NET System.Diagnostics** и **ManagementObjectSearcher (WMI)** для доступа к системной информации;
* Вся логика разделена на **отдельные модули**: сбор данных, анализ, отображение, логирование;
* Использована **асинхронная обработка данных** для обновления информации без блокировки пользовательского интерфейса;
* Вывод информации реализован в виде **структурированных таблиц**, отображаемых в консоли.

**Ход разработки:**

* На первых этапах была создана архитектура приложения и реализован базовый функционал по сбору системных данных;
* Далее добавлены модули логирования, фильтрации и анализа;
* Финальный этап включал оптимизацию производительности и тестирование приложения в различных условиях нагрузки.

Приложение успешно выполняет задачи системного мониторинга и может быть использовано для базового анализа производительности ПК и обнаружения ресурсоёмких процессов.

### ****ЗАКЛЮЧЕНИЕ****

**В процессе прохождения проектной практики были успешно выполнены:**

* командный проект Telegram-бота для учёта задач (в рамках проектной деятельности);
* индивидуальный проект на C# для мониторинга системных ресурсов;
* создание и оформление веб-сайта проекта.

Разработка Telegram-бота позволила углубить навыки работы с Python, библиотекой Aiogram, базами данных SQLite и средствами построения пользовательского взаимодействия через FSM. Бот был реализован в рамках учебной дисциплины «Проектная деятельность» и интегрирован в общую структуру практики.

Индивидуальная часть работы была направлена на освоение языка C# и системных API Windows. В результате была создана консольная программа для мониторинга системных ресурсов, реализованная на принципах модульного и многопоточного программирования с использованием архитектуры MVVM и асинхронной обработки данных.

Также был разработан и оформлен статический веб-сайт проекта на HTML и CSS, включающий основные разделы и адаптивный интерфейс. Это позволило дополнительно закрепить навыки вёрстки, навигации и организации проектной презентации.

Все результаты были протестированы, документированы и размещены в итоговом репозитории и на сайте проекта. Практика способствовала развитию технических и организационных навыков, а также формированию целостного подхода к реализации IT-проектов.

# **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

*1. Telegram Bot API:* [*https://core.telegram.org/bots/api*](https://core.telegram.org/bots/api) *2. Aiogram Framework:* [*https://docs.aiogram.dev*](https://docs.aiogram.dev) *3. SQLite3:* [*https://docs.python.org/3/library/sqlite3.html*](https://docs.python.org/3/library/sqlite3.html) *4. Spectre.Console:* [*https://spectreconsole.net*](https://spectreconsole.net) *5. System.Diagnostics:* [*https://learn.microsoft.com/en-us/dotnet/api/system.diagnostics*](https://learn.microsoft.com/en-us/dotnet/api/system.diagnostics) *6. WPF Framework:* [*https://learn.microsoft.com/en-us/dotnet/desktop/wpf/*](https://learn.microsoft.com/en-us/dotnet/desktop/wpf/) *7. HTML и CSS:* [*https://developer.mozilla.org/ru/docs/Web*](https://developer.mozilla.org/ru/docs/Web)

**ПРИЛОЖЕНИЯ**  
Ссылка на видео: [*https://disk.yandex.ru/i/Lhg8IhM\_3ZMhLQ*](https://disk.yandex.ru/i/Lhg8IhM_3ZMhLQ)Ссылка на git репозиторий:[*https://github.com/Evenmurmur/Practice-2025-1*](https://github.com/Evenmurmur/Practice-2025-1)