Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий  
Кафедра «Информатика и вычислительная техника»

Направление подготовки/ специальность: Системная и программная инженерия

ОТЧЕТ

по проектной практике

Студенты: Больбот Алексей Дмитриевич\_\_ Группа: \_\_\_\_241-326\_\_\_\_\_\_

Арзамазов Виктор Владимирович\_\_\_\_\_\_\_\_

Место прохождения практики: Московский Политех, кафедра \_информатики и вычислительной техники\_\_

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

Руководитель практики: \_\_\_Чернова Вера Михайловна \_\_\_\_

Москва 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

1. Общая информация о проекте: 4

* Название проекта
* Цели и задачи проекта

1. Общая характеристика деятельности организации *(заказчика проекта)* 5

* Наименование заказчика
* Организационная структура
* Описание деятельности

1. Описание задания по проектной практике 6
2. Описание достигнутых результатов по проектной практике 8

ЗАКЛЮЧЕНИЕ *(выводы о проделанной работе и оценка ценности выполненных задач для заказчика)* 9

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 10

**ВВЕДЕНИЕ**

В современном мире информационных технологий разработка программного обеспечения требует не только технических навыков, но и умения эффективно использовать инструменты для совместной работы и документирования. Данный отчет посвящен проектной практике, в рамках которой был разработан Telegram-бот для получения гороскопов с интеграцией внешнего API. Основной целью проекта стало создание удобного инструмента, предоставляющего пользователям персонализированные прогнозы на основе выбранного знака зодиака и даты. В ходе работы были освоены ключевые технологии: система контроля версий Git, язык разметки Markdown, создание статических сайтов на HTML/CSS, а также практические аспекты разработки ботов на Python с использованием библиотеки pyTelegramBotAPI.

Проектная практика позволила углубить знания в области работы с API, проектирования архитектуры приложений и документирования процессов. Исследование предметной области началось с анализа существующих решений и выбора подходящего API для получения данных о гороскопах. После оценки доступных вариантов был интегрирован сервис Horoscope-App-API, обеспечивающий актуальные прогнозы. Разработка бота включала реализацию интерактивного диалога с пользователем, обработку ошибок ввода и сетевых сбоев, а также внедрение системы кэширования для снижения нагрузки на внешний сервис.

Важным аспектом стала документация проекта. Все этапы работы фиксировались в Markdown-файлах, включая техническое руководство, описание архитектуры и инструкции по запуску. Для визуализации процессов взаимодействия бота с API были подготовлены UML-диаграммы. Дополнительно разработан статический сайт на HTML/CSS, который стал презентационной платформой проекта. Сайт включает страницы с описанием функционала, скриншотами интерфейса, списком участников и ссылками на ресурсы.

Одной из ключевых задач стало обеспечение корректной работы бота при высоких нагрузках. Для этого проведено тестирование производительности, оптимизированы запросы к API и настроена обработка исключений. Взаимодействие с куратором проекта и командой позволило своевременно корректировать требования и улучшать функционал. Результатом стал стабильно работающий бот, готовый к масштабированию.

1. **ОБЩАЯ ИНФОРМАЦИЯ О ПРОЕКТЕ**
   1. **Название проекта**

Полное наименование проекта: Telegram-бот для получения гороскопов с интеграцией внешнего API.

Краткое наименование: HoroscopeBot.

* 1. **Цели и задачи проекта**

Цель: Разработка Telegram-бота, предоставляющего пользователям персонализированные гороскопы на основе выбранного знака зодиака и даты.

Задачи:

* Исследование Telegram Bot API и выбор технологического стека.
* Интеграция с внешним API для получения данных о гороскопах.
* Реализация системы кэширования для оптимизации запросов.
* Создание интуитивного пользовательского интерфейса с обработкой ошибок.
* Документирование проекта в формате Markdown, включая UML-диаграммы.
* Разработка статического сайта для презентации проекта.

1. **ОБЩАЯ ХАРАКТЕРИСТИКА ДЕЯТЕЛЬНОСТИ ОРГАНИЗАЦИИ**
   1. **Наименование заказчика**

Федеральное государственное автономное образовательное учреждение высшего образования «Московский политехнический университет».

* 1. **Организационная структура**
* Ректорат;
* административные подразделения;
* студенческое самоуправление;
* научные и исследовательские центры;
* факультеты;
* кафедры.
  1. **Описание деятельности**

Многопрофильное высшее учебное заведение, участник программы «Приоритет 2030». Учредителем университета является Министерство науки и высшего образования Российской Федерации.

1. **ОПИСАНИЕ ЗАДАНИЯ ПО ПРОЕКТНОЙ ПРАКТИКЕ**
2. **Настройка Git и репозитория:**
   * Создайте личный или групповой репозиторий на [GitHub](https://github.com/) или [GitVerse](https://gitverse.ru/) на основе предоставленного [шаблона](https://github.com/mospol/practice-2025-1).
   * Освойте базовые команды Git: клонирование, коммит, пуш и создание веток.
   * Регулярно фиксируйте изменения с осмысленными сообщениями к коммитам.
3. **Написание документов в Markdown:**
   * Все материалы проекта (описание, журнал прогресса и др.) должны быть оформлены в формате Markdown.
   * Изучите синтаксис Markdown и подготовьте необходимые документы.
4. **Создание статического веб-сайта:**
   * Вы можете использовать **только HTML и CSS** для создания сайта, если освоение более сложных инструментов представляется трудным. Это делает задание доступным для студентов с базовым уровнем подготовки.
   * Создайте новый сайт об основном проекте по дисциплине «Проектная деятельность», выберите тему и добавьте контент. Оформление и наполнение сайта должны быть уникальными (не совпадать с работами других студентов) более, чем на 50%.
   * Сайт должен включать:
     + **Домашнюю страницу** с аннотацией проекта.
     + **Страницу «О проекте»** с описанием проекта.
     + **Страницу или раздел «Участники»** с описанием личного вклада каждого участника группы в проект по «Проектной деятельности».
     + **Страницу или раздел «Журнал»** с минимум тремя постами (новостями, блоками) о прогрессе работы.
     + **Страницу «Ресурсы»** со ссылками на полезные материалы (ссылки на организацию-партнёра, сайты и статьи, позволяющие лучше понять суть проекта).
   * Оформите страницы сайта графическими материалами (фотографиями, схемами, диаграммами, иллюстрациями) и другой медиа информацией (видео).
   * **Ожидаемое время:** изучение и настройка — 10–14 часов, дизайн и наполнение — 4–8 часов.
5. **Отчёт по практике**
   * Составьте отчёт по проектной (учебной) практике.

**4. ОПИСАНИЕ ДОСТИГНУТЫХ РЕЗУЛЬТАТОВ**

Репозиторий проекта:

Структура:

/

├── docs/ # Документация

├── site/ # Статический сайт

├── src/ # Исходный код бота

└── README.md # Описание проекта

Ссылка: https://github.com/ask3la/Project-Practice

Реализованный функционал бота:

Команды: /start, /horoscope.

Диалог для выбора знака зодиака и даты.

Интеграция с API и кэширование данных.

Обработка ошибок (некорректный ввод, сбои API).

Статический сайт:

Реализован сайт на 5 страниц. На сайте присутствуют все разделы требуемые по заданию, скриншоты работы Telegram-бота и фреймворка.

Дополнения:  
 Добавлена возможность развертки проекта с использованием Docker.

**ЗАКЛЮЧЕНИЕ**

В рамках проектной практики успешно разработан Telegram-бот для получения гороскопов, интегрированный с внешним API. Освоены ключевые навыки: работа с Git, создание документации в Markdown, разработка статических сайтов и взаимодействие с API. Реализованные модификации (кэширование, обработка ошибок) повысили надежность и удобство использования бота. Приобретенный опыт станет основой для дальнейшего развития в области разработки программного обеспечения.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ:**

1. Основы HTML. ― URL: <https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content>.
2. Основы CSS. ― URL: <https://developer.mozilla.org/ru/docs/Web/CSS>.
3. Элементы HTML. ― URL: <https://developer.mozilla.org/ru/docs/Web/HTML/Element>
4. Телеграм API ― URL: <https://core.telegram.org/>
5. Телеграм бот на python ― URL: <https://thecode.media/python-bot/>