Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий  
Кафедра «Инфокогнитивных технологий»

Направление подготовки/ специальность: Системная и программная инженерия

ОТЧЕТ

по проектной практике

Студент: Лазарева Александра Алексеевна Группа: 241-327

Студент: Николаев Алексей Владимирович Группа: 241-3211

Место прохождения практики: Московский Политех, кафедра Инфокогнитивных технологий

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

Руководитель практики: Чернова Вера Михайловна

Москва 2025

ОГЛАВЛЕНИЕ

1. Общая информация о проекте:

* Название проекта
* Цели и задачи проекта
* *Структура работы в проекте*

1. Описание задания по проектной практике

* Настройка репозитория на GitHub
* Разработка статического веб-сайта
* Разработка Телеграм бота
* Взаимодействие с компаниями-партнерами

1. Заключение
2. Список литературы
3. **Общая информация о проекте**

Проект игровой индустрии “Sarayashiki” - это многопользовательская игра в жанре Hack-and-Slash с элементами Roguelike, действие которой разворачивается в Японии периода Сэнгоку с элементами фэнтези. Игрокам предстоит сражаться с монстрами и другими игроками 1 на 1.

*Цели и задачи проекта*

Основной целью проекта является разработка и выпуск многопользовательской игры, сочетающей в себе элементы разных жанров и проработанный сеттинг.

Достижение данной цели осуществляется с помощью выполнения следующих задач:

* Разработка глубоко проработанных механик геймплея: создание боевой системы основанной на жанре Hack-and-Slash, внедрить чередующиеся PvP и PvE составляющие, создать интересную реализацию Roguelike механики.
* Проработка сеттинга и сюжета: разработка увлекательного лора, вдохновенного феодальной Японией и японской мифологией.
* Создание мультиплеерной инфраструктуры: обеспечить стабильную сетевую инфраструктуру, направленную на многопользовательский геймплей.
* Оптимизация игры: оптимизировать игру для различных устройств, обеспечивая плавный геймплей.

*Структура работы в проекте*

Проект Sarayashiki организован таким образом, чтобы обеспечить эффективное управление, контроль и выполнение задач. Структура работы основана на использовании системы контроля версий GitLab для управления репозиторием проекта и его технической реализации, а также на четком распределении ролей и обязанностей среди членов команды.

Команда разработки проекта состоит из нескольких ключевых ролей:

* *Тимлид(Team lead)*: Ответственный за техническое руководство проектом, координацию работы разработчиков, создание и планирование сроков задач.
* *Проектный менеджер (Project Manager)*: Ответственный за общее управление проектом, планирование сроков выполнения задач, распределение ресурсов и контроль за их выполнением, а также он занимается коммуникацией внутри разработчиков.
* *Разработчики*: Ответственные за написание и тестирование кода, реализацию функциональности игры и устранение багов. Разработчики работают в соответствии с задачами, назначенными им тимлидом и проектным менеджером.
* *Дизайнеры и художники*: Ответственные за создание визуального контента, включая графику и пользовательский интерфейс.
* *Сценаристы*: Ответственные за разработку сюжета и лора игры, создание интересных персонажей и событий. Сценаристы также влияют на ключевые механики и проработку геймплея.

Работа в проекте организована на недельные циклы, в течение которых тимлид и проектный менеджер составляют и раздают задания команде. Эти задания фиксируются в системе GitLab и содержатся в отдельной таблице для удобства отслеживания и контроля.

1. *Составление задач*: Тимлид и проектный менеджер определяют приоритетные задачи на предстоящую неделю, основываясь на текущем состоянии проекта и поставленных целях.
2. *Распределение задач*: Задачи распределяются среди членов команды в зависимости от их ролей, опыта и загрузки. Каждая задача описывается в репозитории GitLab, где указываются её детали, сроки выполнения и ответственные лица.
3. *Контроль и отчётность*: В течение недели команда работает над выполнением задач, регулярно обновляя статус их выполнения в GitLab. Тимлид и проектный менеджер контролируют прогресс и при необходимости вносят коррективы в план.

*Коммуникация и сотрудничество*

Для эффективной работы и взаимодействия команда использует различные инструменты и методы коммуникации:

* *Онлайн-встречи*: Еженедельные видеоконференции для обсуждения прогресса, решения возникших проблем и планирования дальнейших действий.
* *Оффлайн-встречи*: Периодические собрания команды, включающие не только обсуждение технических моментов, но и мероприятия по командообразованию. Эти встречи способствуют укреплению командного духа и улучшению взаимодействия между членами команды.
* *Чат*: Использование мессенджеров и чатов для оперативного обмена информацией и решения повседневных вопросов.

1. **Описание задания по проектной практике**

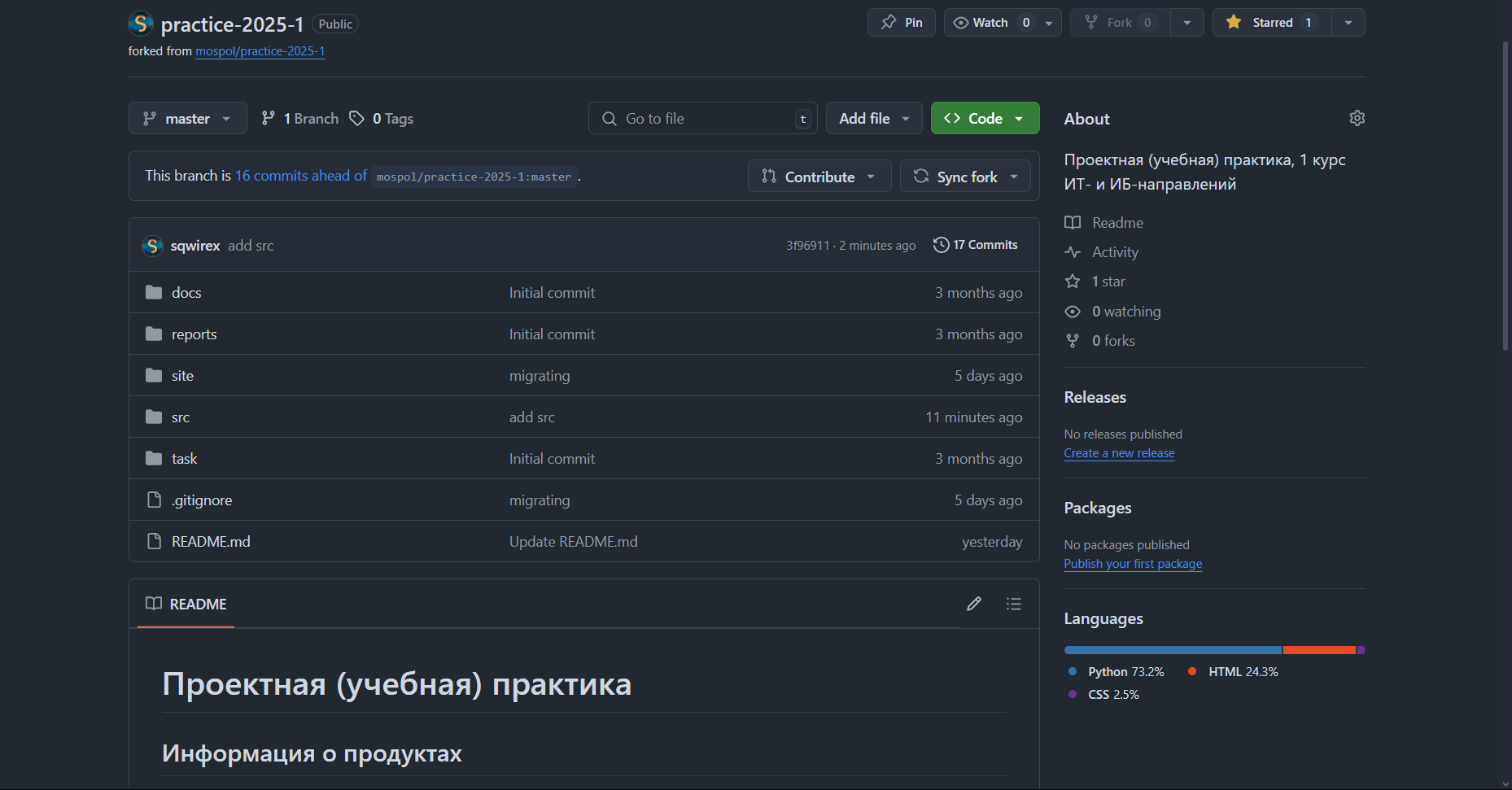
*Базовая часть*

*Настройка репозитория на GitHub*

*Структура репозитория*

Репозиторий был организован в соответствии с задачами практики и включал в себя следующие основные компоненты:

* README.md – главный файл проекта, содержащий информацию о практике, периоде проведения, списке участников и формулировке вариативного задания.
* reports/ – папка, предназначенная для хранения итоговых отчетов
* site/ – файлы статического веб-сайта, включая HTML, CSS и изображения.
* src/ – директория с исходным кодом бота Wordly, разработанного в рамках вариативного задания.
* task/ – хранилище текста задания и шаблона отчёта.



Репозиторий GitHub (Рис. 1)

*Использование системы контроля версий Git*

Для успешного ведения командной работы и управления проектом в ходе практики проводилось обучение основным принципам работы с системой контроля версий Git. Это позволило овладеть базовыми и продвинутыми навыками работы с репозиторием.

Мы ознакомились с основными принципами работы системы версионирования, индивидуально настроили рабочее окружение, научились использовать основные команды и их применение.

*Разработка статического веб-сайта*

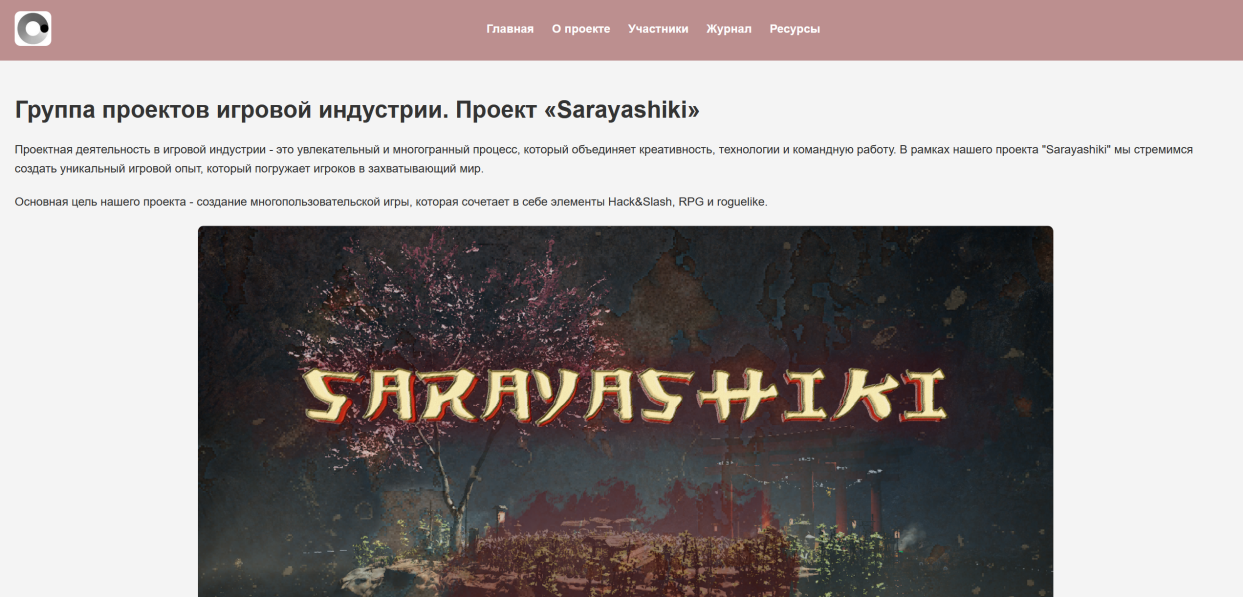
Этап 1: Изучение основ HTML и CSS

Обучение основам веб-разработки включало в себя следующие технологии:

* HTML и его семантическую структуру (<header>, <section>, <footer>).
* CSS и стилизацию (flexbox, grid, работа с шрифтами и цветами`).
* Адаптивность, позволяющую сайт корректно отображаться на разных устройствах.

Этап 2: Создание шаблона сайта

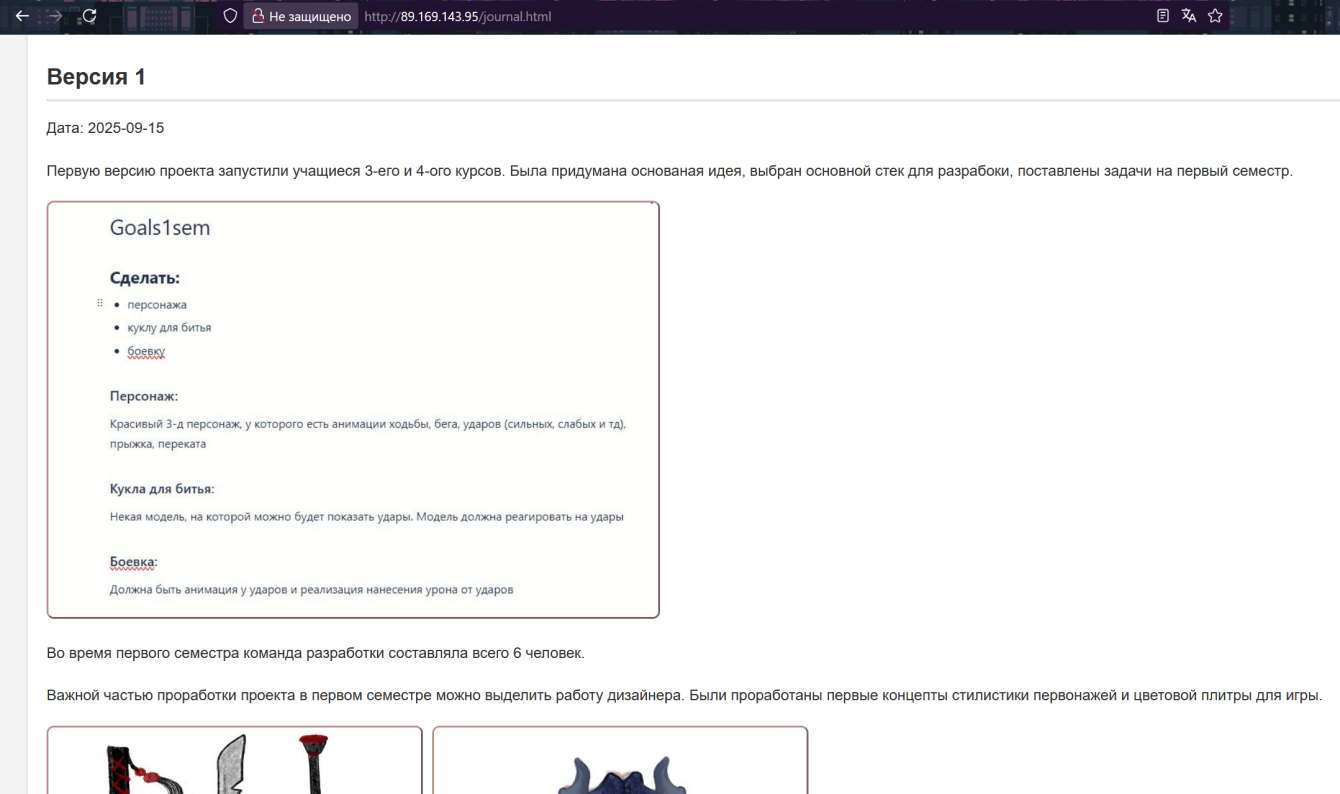
На первом этапе работы над сайтом мы сфокусировались на его структуре. Сначала мы разработали шаблон сайта, оформив его базовую HTML-разметку и стили в CSS. Это позволило создать удобный и понятный каркас, который позже наполнили контентом. Как было задано, сайт включает в себя разделы: “Домашняя страница”, “О проекте”, “Участники”, “Журнал” и “Ресурсы”.



Главная страница сайта (Рис. 2)

Этап 3: Наполнение сайта контентом

После утверждения шаблона начался процесс наполнения сайта. Мы создали текстовые описания проекта, целей и задач, поставленных для разработки, личного вклада участников и структуру работы, и заполнили «Журнал» тремя постами, а также наполнили сайт иллюстрациями и видео.



Пример наполнения сайта (Рис. 3)

Этап 4: Размещение на сервере

По мере продвижения работы мы приняли решение приобрести сервер, чтобы разместить бота на отдельной платформе, параллельно у нас появилась возможность разместить на сервере сайт. Теперь наш сайт доступен по IP-адресу, который можно найти в репозитории на GitHub.

*Вариативная часть*

*Разработка Телеграм бота*

В рамках практики нами была разработана программа — Telegram-бот Wordly, реализующий игровой процесс на основе угадывания слов.

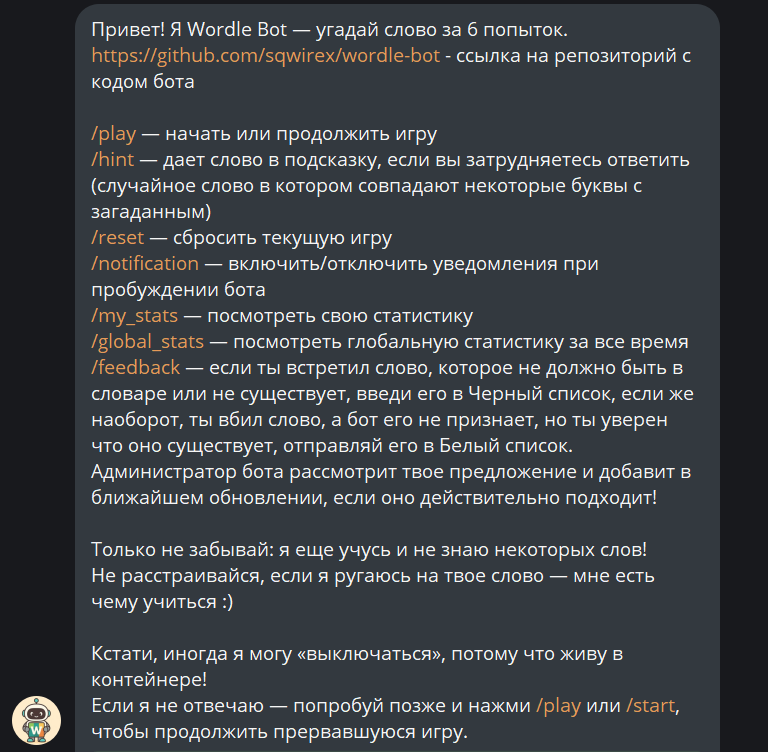
В процессе работы особое внимание уделялось организации словаря и механике обработки данных. Для этого использовалась библиотека pymorphy, позволившая фильтровать слова по частоте встречаемости, выделять существительные определенной длины и приводить их к форме именительного падежа, что обеспечило корректную обработку вводимых пользователем данных.

Техническая реализация бота базировалась на использовании хэндлеров Telegram, с помощью которых была обеспечена обработка пользовательских команд и управление игровым процессом.



Первая версия бота (Рис.4)

В ходе работы были интегрированы функциональные возможности, включая уведомления, механизмы угадывания слов, возможность сброса игры, систему проигрыша, а также инструменты для добавления новых слов. Помимо этого, был реализован механизм работы с черным и белым списками слов, что позволило фильтровать неподходящие варианты.



Приветственное слово (Рис. 5)

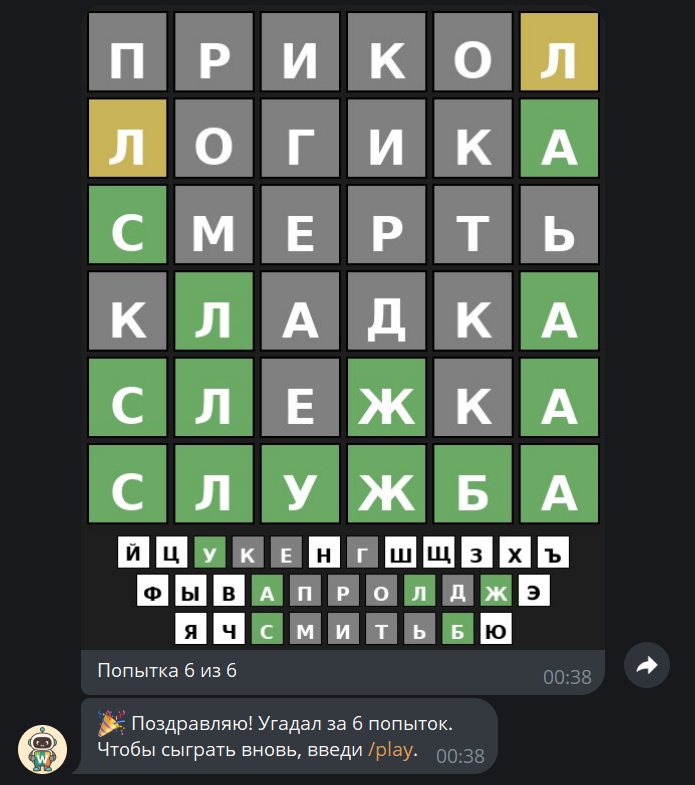
Дополнительно были разработаны глобальная и личная статистика пользователей, административные команды, такие как рассылка сообщений, блокировка и разблокировка игроков, а также система подсказок, направленная на поддержку участников в ходе игры.



Статистика бота (Рис. 6)

Важным элементом проекта стал логгер, который позволил вести детализированный учет действий игроков, анализировать игровой процесс и на основе собранных данных формировать статистику.

На этапе дальнейшей оптимизации была внедрена система визуального представления игрового процесса. В качестве инструмента для генерации изображений была выбрана библиотека Pillow, позволившая создавать графические элементы с отображением слов. Такой подход значительно повысил удобство взаимодействия с ботом и улучшил восприятие игры.



Версия игры с добавлением графики (Рис. 7)

Для развертывания проекта нами был приобретен сервер, на который были загружены репозитории бота и веб-сайта. В целях удобства управления средой выполнения был установлен Docker и настроены все необходимые зависимости. На его основе был сформирован и забилден образ с использованием Python 3.10, что обеспечило стабильную работу бота в контейнеризированной среде и позволило создать удобную инфраструктуру для развертывания приложения. Данный подход гарантировал надежность функционирования программного продукта, минимизацию возможных ошибок и удобство последующей поддержки проекта.

*Взаимодействие с компаниями-партнерами*

В рамках практики мы также уделили внимание профессиональному развитию и взаимодействию с представителями индустрии.

Первым крупным мероприятием, в котором мы приняли участие, стал Карьерный марафон Московского Политеха. На этом мероприятии нам удалось установить контакт с представителями различных компаний, узнать о ключевых критериях, по которым работодатели оценивают соискателей, и понять, какие навыки наиболее востребованы на рынке труда. Кроме того, мы получили полезные рекомендации по подготовке резюме, прохождению технических собеседований и развитию soft skills, которые играют важную роль в профессиональном росте.

Следующим значимым событием стала онлайн-конференция R-EVOlution, организованная компанией R-Vision. В ходе конференции мы ознакомились с различными направлениями технологической деятельности, включая информационную безопасность, работу с big data и особенности инфраструктуры взаимодействия заказчиков и разработчиков. Каждое из представленных направлений дало нам более глубокое понимание современных вызовов и подходов в IT-индустрии. Особенно полезными стали практические кейсы, продемонстрировавшие реальные способы решения сложных задач в сфере кибербезопасности и управления данными.

Завершающим мероприятием в рамках нашей практики стала поездка в офис Яндекса в рамках Young Con. Здесь мы смогли пообщаться с действующими разработчиками и лучше понять, как устроена корпоративная культура в крупных технологических компаниях. Нам рассказали о процессах внутри компании, принципах работы команд, подходах к разработке и важности командного взаимодействия. Поездка завершилась экскурсией по главному корпусу Яндекса в Москве, где мы получили возможность взглянуть на внутреннюю инфраструктуру компании и оценить, как организовано рабочее пространство сотрудников.

1. **Заключение**

В ходе прохождения практики нами была проделана комплексная работа, включавшая в себя изучение современных технологий, разработку программных решений и взаимодействие с представителями индустрии. Мы не только освоили инструменты контроля версий и веб-разработки, но и получили опыт проектирования и развертывания IT-продуктов в реальных условиях.

Разработка статического веб-сайта позволила нам на практике применить знания HTML и CSS, научиться адаптировать контент и организовывать его структуру. Работа с Telegram-ботом Wordly дала возможность погрузиться в принципы обработки естественного языка, создания интерактивных сценариев и организации инфраструктуры серверного окружения. Освоение Docker и управление зависимостями обеспечило нам понимание контейнеризированных сред и их роли в развертывании проектов.

Помимо технических аспектов, практика дала нам важный опыт взаимодействия с IT-сообществом и компаниями-партнёрами. Участие в мероприятиях позволило нам познакомиться с требованиями работодателей, узнать о современных трендах и расширить кругозор в сфере технологий. Мы получили представление о корпоративной культуре, процессах найма и перспективах профессионального развития.

Итогом проделанной работы стал полноценный проект, включающий веб-ресурс и Telegram-бот, полностью отвечающий заданным требованиям. Мы освоили широкий спектр инструментов, улучшили навыки командной работы и получили ценный опыт, который будет полезен в дальнейшей профессиональной деятельности.

1. **Список литературы**
   * 1. Введение в git [Электронный ресурс]. – URL: <https://git-scm.com/book/ru/v2> (дата обращения: 18.05.2025).
     2. Библиотека по работе с Telegram API [Электронный ресурс]. – URL: <https://core.telegram.org/bots/api> (дата обращения: 18.05.2025).
     3. Работа с Python и обработка данных [Электронный ресурс]. – URL: <https://docs.python.org/3/> (дата обращения: 18.05.2025).
     4. Создание веб-сайтов (HTML, CSS) [Электронный ресурс]. – URL: <https://developer.mozilla.org/> (дата обращения: 18.05.2025).
     5. Docker и развертывание серверных приложений [Электронный ресурс]. – URL: <https://docs.docker.com/> (дата обращения: 18.05.2025).
     6. Работа с NLP и обработкой текста. Официальная документация pymorphy. [Электронный ресурс]. – URL: <https://pymorphy2.readthedocs.io/> (дата обращения: 18.05.2025).
     7. Генерация изображений в Python. Официальная документация Pillow [Электронный ресурс]. – URL: <https://pillow.readthedocs.io/en/stable/> (дата обращения: 18.05.2025).