

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

Студент: Крипак Ксения Романовна

Группа: 241-326

Место прохождения практики: Московский Политех, кафедра информатики и
вычислительной техники

Отчет принят с оценкой _____ Дата _____

Руководитель практики: Чернова Вера Михайловна

Оглавление

ВВЕДЕНИЕ	3
1 Общая информация о проекте.....	4
1.1 Название проекта.....	4
1.2 Цели и задачи проекта	4
2. Общая характеристика деятельности организации (заказчика проекта).....	5
2.1 Наименование заказчика	5
2.2 Организационная структура.....	5
2.3 Описание деятельности	5
3. Описание задания по проектной практике	6
3.1 Базовая часть задания	6
3.2 Вариативная часть: практическая реализация технологии.....	7
4 Описание достигнутых результатов	8
4.1 Разработан полнофункциональный Telegram-бот для проведения викторины	8
4.2 Реализована структура проекта и базовая архитектура	9
4.3 Создано пошаговое руководство по созданию Telegram-бота.....	9
4.4 Интеграция с базой данных и создание веб-сервиса.....	10
4.5 Подготовлен и оформлен сопутствующий сайт проекта	10
4.6 Получен практический опыт взаимодействия с ИТ-сообществом	10
5 Выводы и анализ результатов	11
Список использованной литературы.....	13

ВВЕДЕНИЕ

Проектная практика в текущем семестре была посвящена участию в разработке и развитии проекта "Киберполигон". Работа над проектом велась в составе команды, деятельность которой была разделена на несколько ключевых направлений, каждое из которых внесло существенный вклад в общий результат. Основной акцент был сделан на практическое применение знаний в сфере кибербезопасности, а также на командную работу, наставничество и развитие проекта как полноценной образовательной и технической платформы.

Также проектная практика включала вариативную часть, отдельную от основного проекта.

1 Общая информация о проекте

1.1 Название проекта

Я участвую в проекте с названием «Киберполигон».

1.2 Цели и задачи проекта

Целью проекта является создание Киберполигона — специализированной платформы для обучения и тренировки в сфере кибербезопасности. Основные задачи проекта включают:

1. Сокращение затрат на подготовку киберучений за счёт создания цифровой среды для тренировок;
2. Обеспечение гибкости моделирования различных технологических процессов и инцидентов;
3. Предоставление возможностей для практического обучения и развития навыков в области кибербезопасности, включая обучение с нуля для людей без предварительной подготовки;
4. Создание реальных учебных заданий (тасков), выполняемых участниками проекта;
5. Разработка концептуального и технического макета проекта с целью обоснования финансирования;
6. Формирование системы наставничества, направленной на включение первокурсников в проектную деятельность и развитие их профессиональных компетенций.

2. Общая характеристика деятельности организации (заказчика проекта)

2.1 Наименование заказчика

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

2.2 Организационная структура

Московский Политех — один из крупнейших многопрофильных вузов России, образованный в 2016 году путем слияния Университета машиностроения (МАМИ) и Московского государственного университета печати имени Ивана Федорова (МГУП). Университет включает в себя девять факультетов и два института, охватывающих широкий спектр направлений подготовки: от информационных технологий и машиностроения до издательского дела и дизайна.

Организационная структура университета предполагает активное вовлечение студентов в проектную деятельность с первого курса обучения. В рамках проекта «Киберполигон» руководство осуществляет Гневшев Александр Юрьевич, который координирует работу студентов и направляет развитие проекта.

2.3 Описание деятельности

Московский Политех реализует более 275 образовательных программ и активно развивает научную деятельность в различных сферах: от инженерии и биотехнологий до книгоиздательского дела и дизайна. Особое внимание уделяется проектному обучению, что позволяет студентам совмещать теоретические знания с практическими навыками, работая над реальными задачами и кейсами потенциальных работодателей.

Университет активно сотрудничает с более чем 750 индустриальными партнерами, что способствует актуализации образовательных программ и повышению конкурентоспособности выпускников на рынке труда.

3. Описание задания по проектной практике

В рамках проектной практики студентам первого курса, обучающимся по направлениям информационных технологий и информационной безопасности, было предложено выполнить практическое задание объёмом 72 академических часа. Целью практики является формирование у студентов навыков командной работы, освоение инструментов совместной разработки, документирования и презентации результатов, а также углубление знаний в области разработки ИТ-проектов.

3.1 Базовая часть задания

Задание включало несколько обязательных этапов:

- **Настройка среды разработки и Git-репозитория.** Необходимо было создать репозиторий GitHub или GitVerse, освоить базовые команды Git и вести версионирование кода и документов.
- **Создание документации в формате Markdown.** Все отчётные и сопровождающие документы проекта оформлялись в формате Markdown, что обеспечивало единообразие представления материалов.
- **Разработка статического сайта.** В рамках практики был создан статический сайт, описывающий цели, задачи и прогресс проекта. Использовались базовые технологии HTML и CSS, структура сайта включала страницы: «О проекте», «Участники», «Журнал», «Ресурсы» и др.
- **Взаимодействие с организацией-партнёром.** Было организовано участие в ИТ-чемпионате «Цифровая Эра Транспорта», что стало значимым опытом проектного взаимодействия с профессиональной средой.
- **Подготовка итогового отчёта.** Итоговый отчёт был оформлен в форматах DOCX и PDF, размещён в репозитории и загружен в СДО.

3.2 Вариативная часть: практическая реализация технологии

В качестве вариативной части задания была выбрана **реализация телеграм-бота на Python**, предназначенного для проведения тестирования на определение уровня владения английским языком. Проект разрабатывался индивидуально и предусматривал использование современных технологий и архитектурных решений.

4 Описание достигнутых результатов

В рамках проектной практики были достигнуты следующие ключевые результаты:

4.1 Разработан полнофункциональный Telegram-бот для проведения викторины

Создан Telegram-бот, реализующий интерактивную викторину для определения уровня владения английским языком. Основные особенности реализации:

1. Использование асинхронного фреймворка **Aiogram 3**;
2. Управление диалогом с пользователем с помощью **машины состояний (FSM)**;
3. Применение **инлайн-клавиатур** для организации удобного и наглядного интерфейса;
4. Подсчёт правильных ответов и определение уровня пользователя по шкале от A0 до B2;
5. Защита от повторного прохождения викторины.

Бот прошёл полное тестирование и может быть использован как готовый сервис для быстрой оценки уровня английского языка.

4.2 Реализована структура проекта и базовая архитектура

Проект организован в виде модульного Python-приложения с чётким разделением логики по папкам:

```
quiz_bot/
├── app/
│   ├── data/
│   │   └── questions.py      # Список вопросов и вариантов ответов
│   ├── db/
│   │   ├── crud.py          # Взаимодействие с базой данных
│   │   ├── db.py            # Подключение к базе данных
│   │   └── crud.py          # Модели SQLAlchemy, соответствующие таблицам в базе данных
│   ├── handlers/
│   │   ├── start.py         # Приветствие и начало диалога
│   │   └── quiz.py          # Основная логика викторины
│   ├── keyboards/
│   │   └── quiz_kb.py       # Генерация инлайн-клавиатур
│   ├── states/
│   │   └── quiz_states.py    # Определение состояний FSM
│   ├── create_bot.py        # Настройки бота
│   ├── main.py              # Точка входа в приложение
│   └── webserver.py          # Простой вебсервер на FastAPI, с эндпоинтом для бд.
├── initdb/
│   └── init.sql              # Точка входа (запуск бота)
│                               # Скрипт для создания таблиц
├── docker-compose.yml        # Файл для поднятия БД в докер-контейнере
└── requirements.txt          # Зависимости проекта
```

Рисунок 1 – структура созданного бота

Такая структура обеспечила удобную навигацию по коду и возможность масштабирования.

4.3 Создано пошаговое руководство по созданию Telegram-бота

Разработано учебное методическое пособие в виде документа «Пошаговое руководство: как создать телеграм-бота», включающее:

- анализ предметной области и требований;
- описание среды разработки и используемых библиотек;
- практические примеры кода (включая FSM, инлайн-клавиатуры, обработку callback-запросов);
- разбор архитектурных решений;
- рекомендации по тестированию и масштабированию проекта.

Этот документ может использоваться как учебное пособие для студентов и начинающих разработчиков Telegram-ботов.

4.4 Интеграция с базой данных и создание веб-сервиса

Добавлен дополнительный функционал:

- результаты прохождения викторины автоматически сохраняются в базу данных (PostgreSQL);
- разработан веб-сервер на FastAPI, предоставляющий JSON-эндпоинт (/database) для получения списка всех результатов;
- реализована схема базы данных, обеспечивающая хранение ID пользователя, количества набранных баллов и определённого уровня.

Таким образом, проект приобрёл серверную компоненту и может использоваться в качестве полноценного учебного или статистического инструмента.

4.5 Подготовлен и оформлен сопутствующий сайт проекта

Создана веб-страница проекта с подробной информацией о:

- целях и задачах проекта;
- команде разработчиков;
- ход выполнения работы по датам;
- ссылках на репозиторий и Telegram-бота;
- документации и примерах кода.

Сайт выполнен в статическом виде и служит публичной презентацией проекта.

4.6 Получен практический опыт взаимодействия с ИТ-сообществом

В период практики велось взаимодействие с организацией-партнёром: участие в **ИТ-чемпионате «Цифровая Эра Транспорта»**, где проект получил отклик и демонстрировался в рамках конкурсной программы. Это дало дополнительную мотивацию и позволило применить проект в более широком контексте.

5 Выводы и анализ результатов

В рамках проектной практики был реализован функциональный Telegram-бот для проведения интерактивной викторины, определяющей уровень владения английским языком. Работа охватывала все этапы жизненного цикла программного продукта: от постановки задачи до публикации и тестирования. Кроме того, в проект был включён полноценный веб-сайт и организовано взаимодействие с партнёрской организацией.

Были достигнуты следующие цели:

1. **Разработка Telegram-бота:** реализован на базе библиотеки Aiogram 3, с поддержкой асинхронной FSM, инлайн-кнопок, хранения пользовательских данных и учёта результатов викторины.
2. **Создание пошагового руководства:** оформлена подробная инструкция, описывающая процесс создания Telegram-бота, включая настройку окружения, структуру проекта и примеры кода.
3. **Интеграция с базой данных и веб-интерфейсом:** реализован REST-сервис с помощью FastAPI, отображающий результаты пользователей в формате JSON.
4. **Разработка веб-сайта проекта:** создан сайт, презентующий цели, функциональность и доступ к Telegram-боту. Это повысило доступность и удобство взаимодействия с проектом для конечных пользователей.
5. **Взаимодействие с организацией-партнёром:** осуществлялась коммуникация с представителями образовательного центра, обсуждались потенциальные сценарии использования бота, получены рекомендации по улучшению контента и структуры вопросов викторины.

Были получены следующие компетенции:

1. Разработка программных продуктов с использованием **Python**, **Telegram Bot API**, **Aiogram 3** и **FastAPI**;

2. Управление пользовательскими состояниями с помощью **FSMContext**;
3. Работа с **PostgreSQL** в асинхронной среде (через **asyncpg**);
4. Создание и оформление **многофайловых проектов**, структурированных по принципам модульности и расширяемости;
5. Разработка и публикация **одностраничного веб-сайта** (landing page);
6. Практика деловой переписки и **проектного взаимодействия с внешними партнёрами**;
7. Тестирование продукта, сбор обратной связи и его итерационное улучшение.

Все материалы проектной практики размещены в github-репозитории и доступны по ссылке: <https://github.com/xseniaKri/practice-2025-1-kripak>

Список использованной литературы

1. Telegram Bot API. [Электронный ресурс] – Режим доступа: <https://core.telegram.org/bots/api>
2. Aiogram 3 Documentation. [Электронный ресурс] – Режим доступа: <https://docs.aiogram.dev/>
3. FastAPI Documentation. [Электронный ресурс] – Режим доступа: <https://fastapi.tiangolo.com/>
4. PostgreSQL Documentation. [Электронный ресурс] – Режим доступа: <https://www.postgresql.org/docs/>
5. Python AsyncIO Documentation. [Электронный ресурс] – Режим доступа: <https://docs.python.org/3/library/asyncio.html>
6. Григорьев С.В. Создание телеграм-ботов на Python. — СПб.: Питер, 2022.
7. Бовкун Д. Telegram-боты на Python: от простого к сложному. — М.: БХВ-Петербург, 2021.
8. Пархоменко В.В. Асинхронное программирование на Python. — М.: ДМК Пресс, 2020.
9. Маршрутизация и диспетчеризация в Aiogram 3.0 // Хабр. [Электронный ресурс] – Режим доступа: <https://habr.com/ru/articles/697654/>
10. Разработка Telegram-бота с использованием FSM и FastAPI // Medium. [Электронный ресурс] – Режим доступа: <https://medium.com/>