Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий  
Кафедра «СМАРТ-технологии»

Направление подготовки/ специальность: \_09.03.01 «Информатика и вычислительная техника»

ОТЧЕТ

по проектной практике

Студент: \_Жаворонкова Дарья Дмитриевна\_\_\_\_ Группа: \_\_\_241-325\_\_\_\_\_

Студент: \_ Иванов Кирилл Анатольевич\_\_\_\_\_\_\_ Группа: \_\_\_241-325\_\_\_\_\_

Студент: \_Петрова Ксения Андреевна\_\_\_\_\_\_\_\_\_ Группа: \_\_\_241-325\_\_\_\_\_

Место прохождения практики: Московский Политех, кафедра СМАРТ-Технологий

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

Руководитель практики: \_\_Гневшев Александр Юрьевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2025

Оглавление

[**Введение** 3](#_Toc198498887)

[**1.** **Общая информация о проекте:** 4](#_Toc198498888)

[**2.** **Общая характеристика деятельности организации** 5](#_Toc198498889)

[**3.** **Описание задания по проектной практике** 6](#_Toc198498890)

[**4.** **Описание достигнутых результатов по проектной практике** 7](#_Toc198498891)

[4.1. Настройка Git и репозитория 7](#_Toc198498892)

[4.2. Написание документов в Markdown 8](#_Toc198498893)

[4.3. Создание статического веб-сайта 9](#_Toc198498894)

[4.3.1. Главная страница 9](#_Toc198498895)

[4.3.2. Страница «О проекте» 9](#_Toc198498896)

[4.3.3. Страница «Участники» 10](#_Toc198498897)

[4.3.4. Страница «Журнал» 10](#_Toc198498898)

[4.3.5. Страница «Ресурсы» 10](#_Toc198498899)

[4.4. Взаимодействие с организацией-партнёром 11](#_Toc198498900)

[4.5. Тестирование 11](#_Toc198498901)

[4.6. Вариативная часть 13](#_Toc198498902)

[**Заключение** 14](#_Toc198498903)

[**Список использованной литературы** 15](#_Toc198498904)

# **Введение**

Настоящий отчет подробно описывает процесс разработки статического веб-сайта для проекта «Киберполигон», выполненный нашей группой. В рамках данного проекта мы получили задание на создание веб-ресурса, который бы эффективно представлял информацию о «Киберполигоне», его целях, задачах и возможностях. Ключевой целью нашей работы являлась демонстрация приобретенных навыков и знаний в области веб-разработки, включая использование современных технологий, таких как:

• Git: Для контроля версий и организации совместной работы над проектом.

• Markdown: Для создания текстового контента сайта, с акцентом на простоту редактирования и читаемость.

• HTML: Для структурирования контента и создания основы веб-страниц.

• CSS: Для стилизации веб-страниц и создания привлекательного и профессионального дизайна.

В отчете представлена основная информация о проекте и заказчике, описание полученного задания, процесс разработки веб-сайта проекта "Киберполигон" и реализация вариативной части с темой на выбор. Результат нашей работы должен продемонстрировать полученные знания и навыки в области веб-разработки.

# **Общая информация о проекте:**

Название проекта: Киберполигон.

Цель проекта: запрограммировать сайт киберполигона, на котором будет продемонстрирован системный подход к обучению информационной безопасности и отражены современные потребности образования.

Задачи проекта:

* Создание пользовательского интерфейса веб-сайта, обеспечивающего удобную навигацию, привлекательный дизайн и корректное отображение контента на различных устройствах.
* Backend-разработка: Разработка серверной части веб-сайта, обеспечивающей обработку данных, взаимодействие с базами данных и интеграцию с другими системами (в дальнейшей перспективе).
* Разработать учебно-тренировочные задания и обучающие курсы: Создание практических упражнений и образовательных материалов, позволяющих пользователям приобрести навыки в области кибербезопасности.
* Разработать виртуальный макет города: Создание интерактивной модели города, имитирующей реальную инфраструктуру, которую можно использовать для проведения учений и тренировок по кибербезопасности.

# **Общая характеристика деятельности организации**

Наименование заказчика: кафедра «СМАРТ-технологии» Московского политехнического университета.

Организационная структура:

* заведующий кафедрой;
* преподаватели;
* аспиранты и докторанты;
* сотрудники и вспомогательный персонал.

Описание деятельности

Кафедра проводит научные исследования, разрабатывает образовательные программы и осуществляет подготовку специалистов в области смарт-технологий.

# **Описание задания по проектной практике**

Заданием проектной практики является разработка статического веб-сайта об основном проекте по дисциплине «Проектная деятельность» с использованием следующих технологий:

* Git: Для организации эффективной командной работы, контроля версий и обеспечения целостности проекта.
* Markdown: Для формирования отчета о проделанной работе.
* HTML: Для структурирования контента и создания скелета веб-страниц, обеспечивающего семантическую разметку и доступность
* CSS: Для стилизации внешнего вида веб-страниц, создания привлекательного дизайна и обеспечения адаптивности к различным устройствам.

Основными нашими задачами является:

* Создание репозитория Git и настройка рабочего окружения.
* Фронтенд разработка: реализовать базовую структуру HTML настроить стилизацию сайта, работа с Markdown.
* Создание основных страниц сайта.
* Провести тестирование на разных устройствах.

Также в рамках проектной практики необходимо реализовать проект на одну из предложенных тем.

# **Описание достигнутых результатов по проектной практике**

Наша команда реализовала сайт, посвященный проекту «Киберполигон», используя стек технологий, указанный в задании.

## Настройка Git и репозитория

Совместно был создан групповой репозиторий на GitHub, в соответствии с предоставленным шаблоном.

Git — это система контроля версий, которая позволяет отслеживать любые изменения в файлах, хранить их версии и оперативно возвращаться в любое сохранённое состояние.

Основными концептами Git являются:

* Репозиторий (Repository): Хранилище, содержащее все файлы проекта и историю их изменений. Может быть локальным (на компьютере разработчика) или удаленным (на сервере, например, GitHub, GitLab, Bitbucket).
* Коммит (Commit): Снимок состояния проекта в определенный момент времени. Каждый коммит имеет уникальный идентификатор (SHA-1 хеш).
* Ветка (Branch): Независимая линия разработки. Позволяет разрабатывать новые функции или исправлять ошибки, не затрагивая основную ветку (обычно main или master).
* Слияние (Merge): Процесс объединения изменений из одной ветки в другую.
* Конфликт (Conflict): Ситуация, когда Git не может автоматически разрешить изменения между двумя ветками. Требует ручного вмешательства.
* Удаленный репозиторий (Remote Repository): Репозиторий, расположенный на удаленном сервере, используемый для обмена изменениями с другими разработчиками.
* Индекс (Staging Area): Область, в которой находятся подготовленные к коммиту изменения.

В ходе работы над проектом были освоены базовые команды Git: клонирование, коммит, пуш и создание веток. Изменения регулярно фиксировались с сообщениями к коммитам, которые определяли значение изменений.

## Написание документов в Markdown

Команда изучила основной синтаксис Markdown и закрепила полученные знания в процессе оформления журнала прогресса для веб-сайта.

Markdown — это облегченный язык разметки, созданный для упрощения написания текста, который можно легко преобразовать в HTML. Он акцентирует внимание на читабельности, позволяя автору сосредоточиться на содержании, а не на сложном форматировании.

Основными элементами синтаксиса Markdown являются:

1. Заголовки: Markdown поддерживает заголовки шести уровней, которые создаются добавлением символов # в начале строки. Количество символов # определяет уровень заголовка.
2. Выделение текста:
   1. Курсив.
   2. Полужирный.
   3. Курсив и полужирный.
   4. Зачеркнутый.
3. Списки:
   1. Неупорядоченные списки.
   2. Упорядоченные списки.
   3. Вложенные списки.
4. Ссылки:
   1. Встроенные ссылки.
   2. Ссылки-сноски.

Существуют расширения Markdown, которые добавляют дополнительные возможности, но описанный здесь синтаксис является базовым и поддерживается большинством реализаций Markdown.

## Создание статического веб-сайта

Результатом нашей практической работы стал статический веб-сайт, в разработке которого участвовала вся команда.

Для реализации веб-сайта были применены следующие технологии:

* HTML (HyperText Markup Language) — это язык гипертекстовой разметки, который используется для создания и структурирования веб-страниц.
* CSS (Cascading Style Sheets) — это язык стилей для оформления веб-страниц, созданных с помощью HTML. Он отвечает за визуальное представление: цвета, шрифты, отступы, расположение элементов и даже анимацию.

В хедере каждой страницы отображено название проекта и его слоган, а также меню для перехода между страницам сайта. Все страницы сайта имеют одинаковый стиль и адаптируются под разные устройства и размеры окон браузера. Страницы сайта также были оформлены графическими материалами: иллюстрации и диаграмма.

### Главная страница

Данная страница сайта содержит аннотацию проекта. В заголовке основной части страницы идет приветствие Пользователя, далее представлены следующие компоненты:

* Краткая информация о проекте.
* Логотип проекта.
* Предложение Пользователю узнать больше о проекте посредством перехода по различным страницам сайта.

Основная задача страницы "Аннотация проекта" – заинтересовать пользователя и направить его к дальнейшему изучению сайта.

### Страница «О проекте»

Страница отображает основную информацию о проекте, которую можно отнести к следующим подразделам:

* О проекте.
  + Описание основной деятельности проекта.
* Цели.
  + Цели, к которым стремятся участники проекта.
* Анализ угроз взлома сайтов.
  + Нашей командой был проведен анализ угроза взломов сайтов и подведены итоги в виде диаграммы, представленной на странице сайта.

Для большего погружения в тематику проекта и успешного вовлечения в проект аудитории мы определили самые распространенные угрозы безопасности:

* SQL-инъекции.
* Межсайтовый скриптинг (XSS).
* Межсайтовая подделка запросов (CSRF).
* DDoS-атаки.
* Уязвимости в программном обеспечении.

В противовес угрозам мы также исследовали методы защиты от угроз:

* Валидация и фильтрация ввода.
* Использование токенов для защиты от CSRF.
* Обновление программного обеспечения.
* Использование WAF (Web Application Firewall).
* Шифрование данных.

### Страница «Участники»

На странице представлено описание личного вклада каждого участника группы в проект по «Проектной деятельности». На странице указаны Фамилия Имя и Отчество каждого члена команды и его роль в работе над заданием. На страницу также был добавлен графический элемент, который иллюстрирует членов нашей команды.

### Страница «Журнал»

Регулярно обновляемая страница, на которую выкладываются посты о прогрессе работы. Каждый пост содержит заголовок, который отображает краткую информацию о проделанной работе и дату выполнения задач, а также описание самой работы. Данные для данной страницы оформлялись с помощью синтаксиса Markdown и в дальнейшем преобразовывались в HTML.

### Страница «Ресурсы»

Данная страница предоставляет ссылки на полезные материалы. Пользователь может подробней ознакомится с проектом, благодаря ссылкам, которые направят как на официальный сайт проекта, так и на другие полезные источники информации о кибербезопасности.

## Взаимодействие с организацией-партнёром

В рамках Карьерного марафона 22.04.2025 наша команда взаимодействовали с партнерами как волонтеры.

## Тестирование

Нашей группой было проведено ручное тестирование сайта. Результаты работы представлены в формате тест-кейсов.

* Тест-кейс 1: Проверка главной страницы (index.html)

Название: Главная страница отображается корректно

Предусловие: Пользователь открывает сайт в браузере

Шаги:

1. Перейти на главную страницу сайта (index.html)
2. Оценить загрузку страницы
3. Проверить наличие логотипа и основного меню
4. Убедиться, что ссылки в меню работают

Ожидаемый результат:

1. Страница загружается без ошибок
2. Логотип отображается корректно

Фактический результат: Совпадает с ожидаемым

Все пункты меню кликабельны и ведут на соответствующие страницы

* Тест-кейс 2: Проверка страницы участников (participants.html)

Название: Отображение информации об участниках

Предусловие: Пользователь находится на странице участников

Шаги:

1. Перейти на страницу участников (participants.html)
2. Проверить наличие списка участников
3. Проверить наличие основной информации об участниках

Ожидаемый результат:

1. Список участников отображается
2. Основная информация об участниках отображается корректно

Фактический результат: Совпадает с ожидаемым

* Тест-кейс 3: Проверка загрузки изображений

Название: Изображения на сайте отображаются корректно

Предусловие: Открыта любая страница с изображением

Шаги:

1. Проверить отображение логотипа на главной странице
2. Проверить изображение диаграммы на странице с ресурсами
3. Убедиться, что изображение "we.png" отображается на странице "about.html"

Ожидаемый результат:

1. Все изображения загружаются без ошибок
2. Изображения соответствуют описанию на странице

Фактический результат: Совпадает с ожидаемым

* Тест-кейс 4: Проверка корректности ссылок на ресурсы (resources.html)

Название: Все ссылки на странице ресурсов работают

Предусловие: Пользователь находится на странице ресурсов

Шаги:

1. Перейти на страницу ресурсов (resources.html)
2. Кликнуть на каждую ссылку по очереди
3. Проверить открытие целевых страниц

Ожидаемый результат:

1. Все ссылки корректны
2. Переходы происходят без ошибок и ведут на указанные страницы

Фактический результат: Совпадает с ожидаемым

* Тест-кейс 5: Проверка кросс-браузерной совместимости

Название: Сайт отображается корректно в разных браузерах

Предусловие: Доступны современные браузеры (Chrome, Firefox, Edge, Яндекс)

Шаги:

1. Открыть сайт в каждом из указанных браузеров
2. Проверить главную страницу и несколько внутренних страниц
3. Сравнить отображение контента и функциональность

Ожидаемый результат:

1. Сайт корректно отображается во всех браузерах
2. Нет видимых расхождений в верстке и функциональности

Фактический результат: Совпадает с ожидаемым

## Вариативная часть

Наша команда выбрала для реализации тему: Python: How to Write a Lisp Interpreter in Python. Наш проект реализует Telegram-бота по отслеживанию настроения пользователей. Бот позволяет пользователям записывать свое настроение, получать статистику за указанный период и устанавливать ежедневные напоминания о необходимости обновить данные. Название Telegram-бота: MoodBuddy.

Библиотеки, используемые для реализации:

* python-telegram-bot: для взаимодействия с API Telegram.
* sqlite3: для хранения данных о настроении.
* matplotlib: для построения круговых диаграмм.
* io: для работы с графическими буферами.

Принцип работы чат-бота:

1. Пользователь запускает бота с помощью команды /start.
2. Команда /mood инициирует запрос на запись текущего настроения.
   * Если ввод корректен, настроение записывается в базу данных.
   * Если ввод некорректен, бот просит выбрать одно из предложенных настроений.
3. При вводе команды /stats бот возвращает статистику за последние 7 или 30 дней в виде круговой диаграммы.
   * Если за указанный период нет данных, бот сообщает об этом.
4. Команда /remind активирует ежедневные напоминания о записи настроения в 20:00.

Пример работы бота представлен на рисунке 1.

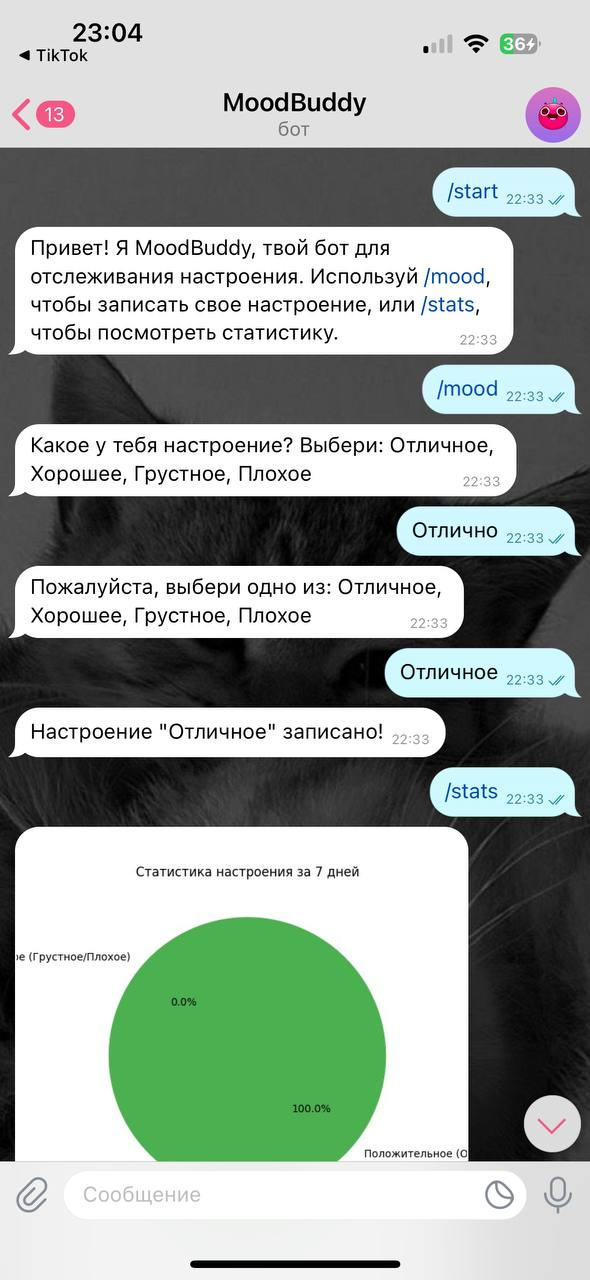


Рисунок 1

Данный Telegram-бот представляет собой эффективное средство для отслеживания настроения. Он легко интегрируется в повседневное использование благодаря простым командам и наглядной визуализации данных. Полученный опыт работы с библиотеками Telegram API, SQLite и matplotlib позволяет расширять функционал бота и применять подобные технологии в других проектах.

# **Заключение**

В ходе проектной практики наша команда успешно реализовала статический веб-сайт проекта «Киберполигон», продемонстрировав приобретенные знания и навыки в области веб-разработки. Мы использовали современные технологии, такие как Git, Markdown, HTML и CSS, что позволило создать структурированный, удобный и визуально привлекательный веб-сайт, в котором отразили основную информацию о проекте. Также в рамках командной работы был разработан Telegram-бот для отслеживания настроения пользователей. Основной функционал бота включает регистрацию настроений, визуализацию статистики в виде круговой диаграммы и установку ежедневных напоминаний.

В процессе работы были решены задачи по настройке командного репозитория на платформе GitHub, организации рабочего процесса с использованием контроля версий, а также реализации основных страниц сайта с применением адаптивного дизайна. Нашей командой было проведено тестирование, результаты подтвердили соответствие сайта заявленным требованиям. Проект реализован с использованием языка программирования Python и библиотек python-telegram-bot, sqlite3, matplotlib. Основные задачи, такие как настройка базы данных, создание команд и визуализация данных, были успешно выполнены.

Таким образом, реализованный веб-сайт соответствует целям проекта и демонстрирует уровень подготовки нашей команды в области создания веб-ресурсов. Работа над вариативной частью задания позволила закрепить навыки работы с Telegram AP. Приобретенный опыт позволит нам создавать более качественные и функциональные веб-страницы и чат-боты на Python.

# **Список использованной литературы**

1. <https://developer.mozilla.org/ru/docs/Learn_web_development/Core/CSS_layout/Introduction>
2. <https://sky.pro/wiki/javascript/osnovnye-ugrozy-bezopasnosti-veb-sajtov/>
3. <https://projects.mospolytech.ru/tproduct/413315123-945301126692-kiberpoligon>
4. <https://owasp.org/www-project-top-ten/>
5. <https://slavaver.github.io/web-course-site/useful/git/>