|  |  |  |
| --- | --- | --- |
|  | Изображение выглядит как зарисовка, рисунок, символ, Штриховая графика  Автоматически созданное описание |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по итоговому проекту**

по дисциплине

«**Технология разработки программных приложений**»

**Часть 2**

|  |  |
| --- | --- |
| **Выполнил:**  Студент группыИКБО-12-22 | Солобай А.П |
| **Проверил:** | ассистент Петрова А.А. |

МОСКВА 2024 г.

**Оглавление**

[Задание 3](#_Toc162221825)

[Часть 1 3](#_Toc162221826)

[Часть 2 3](#_Toc162221827)

[Часть 3 3](#_Toc162221828)

[Часть 4 3](#_Toc162221829)

[Часть 5 3](#_Toc162221830)

[Часть 6 3](#_Toc162221831)

[Выполнение заданий 4](#_Toc162221832)

[Часть 1 4](#_Toc162221833)

[Часть 2 5](#_Toc162221834)

[Часть 3 6](#_Toc162221835)

[Часть 4 7](#_Toc162221836)

[Часть 5 8](#_Toc162221837)

[Часть 6 9](#_Toc162221838)

Задание

## Часть 1

Напишите файл README.md с общим описанием проекта, опишите зависимости проекта и команду для его запуска.

## Часть 2

Опишите выбранный стек технологий для разработки проекта.

## Часть 3

Разработайте архитектуру приложения.

## Часть 4

Настройте систему сборки для вашего проекта.

## Часть 5

Продолжайте писать код.

## Часть 6

Настройте систему сборки для сборки документации проекта.

# Выполнение заданий

## Часть 1

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 1 – README.md

## Часть 2

* Операционная система сервера – Linux Ubuntu
* СУБД – PostgreSQL 16
* Язык программирования – Python 3.11.6
* Среда разработки – Microsoft Visual Studio Code
* Система непрерывной интеграции – Github CI

## Часть 3

Изображение выглядит как снимок экрана, текст, диаграмма, дизайн

Автоматически созданное описание

Рисунок 2 – архитектура приложения

## Часть 4

*Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание*

Рисунок 3 – сборка, запуск проекта

## Часть 5

Листинг 1 – файл handlers/entertainment.py

|  |
| --- |
| import asyncio  from aiogram import types  from keyboards import keyboards as kb  from main import dp, \_, bot  from middlewares.throttling\_middleware import *rate\_limit*  *@rate\_limit*(1)  *@dp*.*message\_handler*(*text*=['🎮Entertainments', '🎮Развлечения'])  async def *entertainments\_handler*(*message*: types.*Message*):      await *message*.*answer*(\_('Hi\, this is the entertainment section\!'), *reply\_markup*=kb.*return\_entertainment\_keyboard*())  *@rate\_limit*(4)  *@dp*.*message\_handler*(  *text*=["Кубик🎲", 'Dice🎲'])  async def *entertainment*(*message*: types.*Message*):      result: types.*Message* = await bot.*send\_dice*(*message*.chat.id, *emoji*=*f*'{*message*.text}',  *allow\_sending\_without\_reply*=True)      dice\_result = result.dice.value      await asyncio.*sleep*(4)      if *message*.text in ['Dice🎲', 'Кубик🎲']:          await *message*.*reply*(\_("The dice rolled \*{dice\_result}\*").*format*(*dice\_result*=dice\_result),  *parse\_mode*='MarkdownV2') |

## Часть 6

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 4 – запуск сборки документации

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 5 – документация

# Вывод

В процессе работы над проектом были достигнуты следующие этапы:

* Создание файла README.md: был написан файл с общим описанием проекта, в котором указаны зависимости и команды для запуска, обеспечивая ясность и доступность информации для потенциальных пользователей и разработчиков.
* Выбор стека технологий: Произведен выбор и описание стека технологий для разработки проекта, что позволило определить оптимальные инструменты и фреймворки для реализации поставленных задач.
* Разработка архитектуры приложения: была спроектирована и документирована архитектура приложения, что стало ключевым этапом в планировании и структурировании проекта.
* Настройка системы сборки: настроена система сборки для проекта, что упростило процесс разработки и обеспечило эффективное управление компонентами и зависимостями.
* Непрерывная разработка кода: продолжалась активная разработка кода, способствующая постепенному достижению проектных целей и реализации функционала.
* Настройка системы сборки для документации: установлена и настроена система для сборки документации проекта, обеспечивая легкость в поддержке и обновлении документации, что важно для понимания и использования проекта.

Эти шаги значительно продвинули проект вперед, укрепляя его техническую базу и облегчая последующие этапы разработки.