



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 самостоятельной работе №2

по дисциплине

«Технология разработки программных приложений»

Выполнили:

Студенты группы ИНБО-30-23

Кук М. В.

Антошкин В. Г.

Проверил:

Преподаватель Исабекова О. А.

Москва 2025

Часть 2. Разработка проекта и системы сборки

Напишите файл README.md с общим описанием проекта, опишите зависимости проекта и команду для его запуска, рис 1.

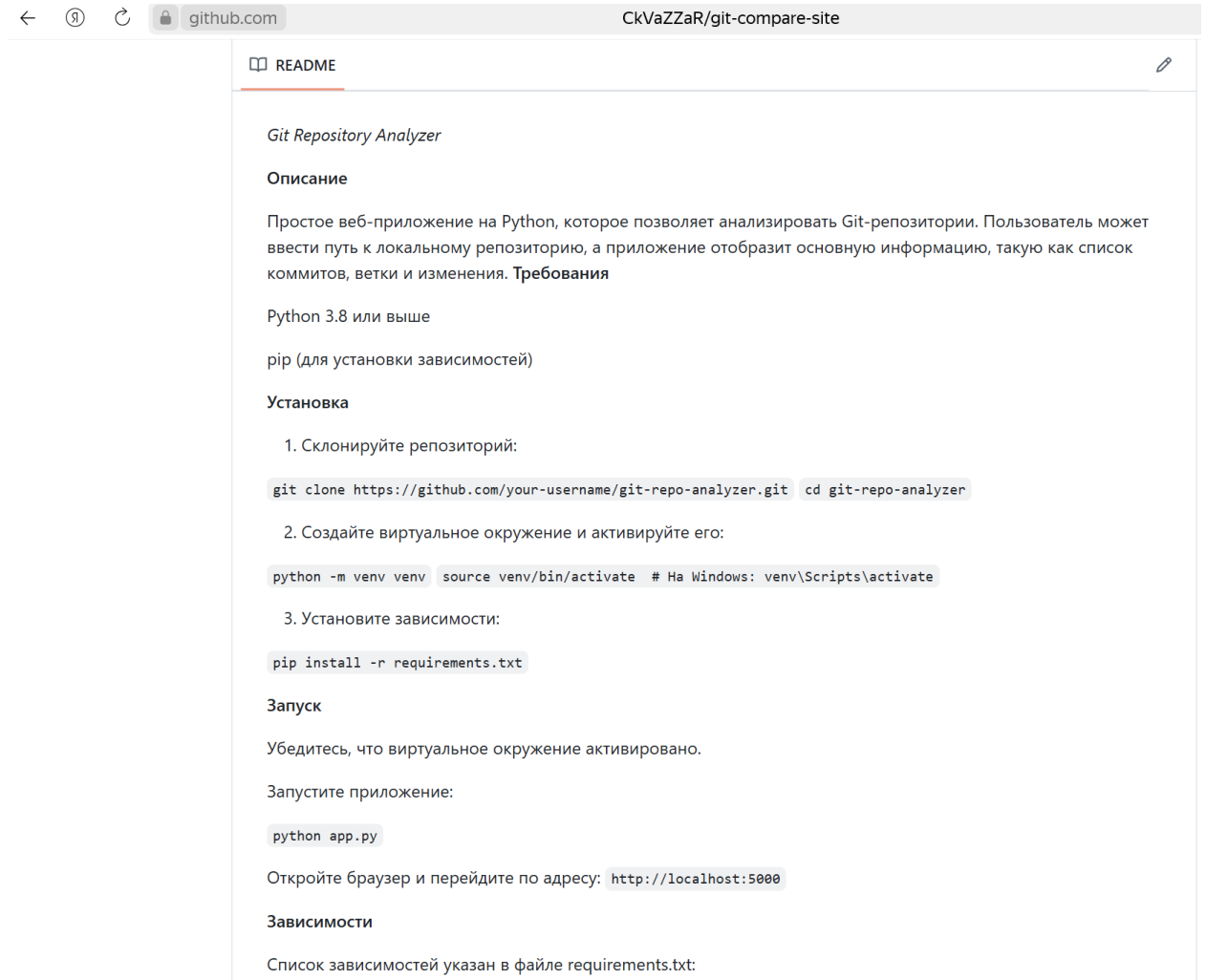


Рисунок 1 – Внешний вид репозитория с README.md

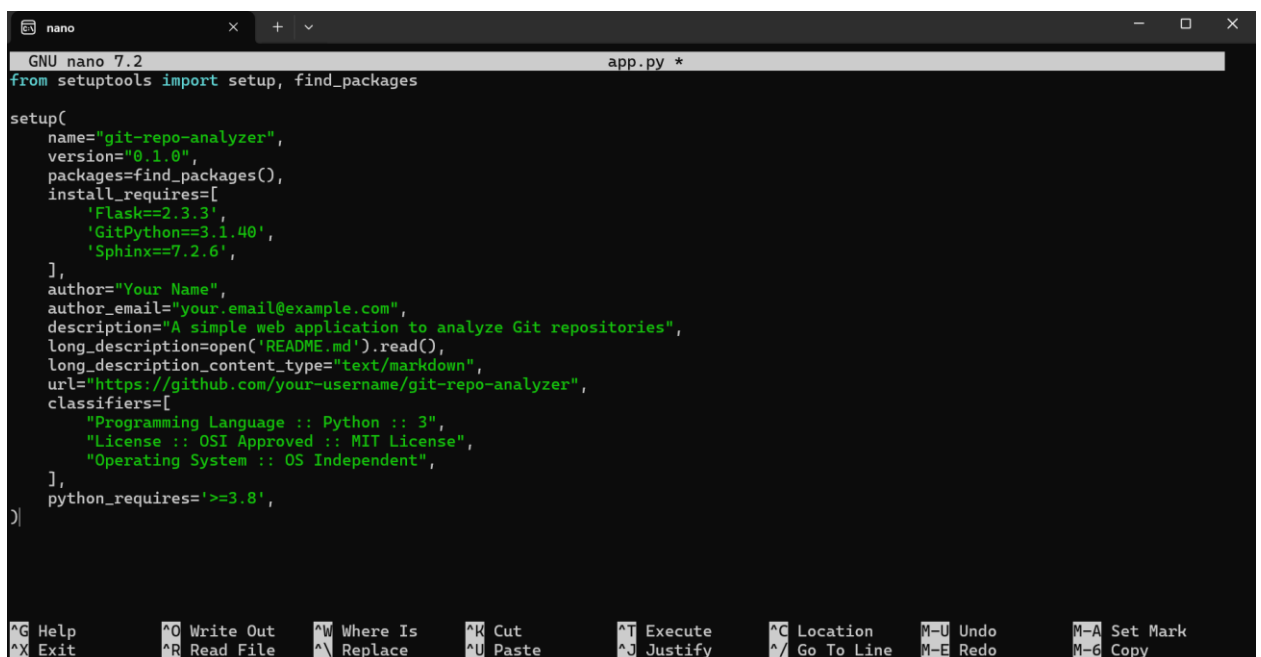
Настройте систему сборки для вашего проекта. В случае проекта на Python: надо сделать файл requirements.txt со списком зависимостей (рис. 2) и подключить setuptools для сборки пакета с проектом, рис 3.



```
GNU nano 7.2 requirements.txt *
alembic==1.16.1
blinker==1.9.0
click==8.1.7
Flask==3.1.1
Flask-Migrate==4.1.0
Flask-SQLAlchemy==3.1.0
greenlet==3.2.2
itsdangerous==2.2.0
Jinja2==3.1.6
Mako==1.3.10
MarkupSafe==3.0.2
psycopy2-binary==2.9.10
SQLAlchemy==2.0.41
typing_extensions==4.13.2
Werkzeug==3.1.3
GitPython==3.1.40
Sphinx==7.2.6
```

Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark
Exit Read File Replace Paste Justify Go To Line M-E Redo M-G Copy

Рисунок 2 – Файл requirements.txt



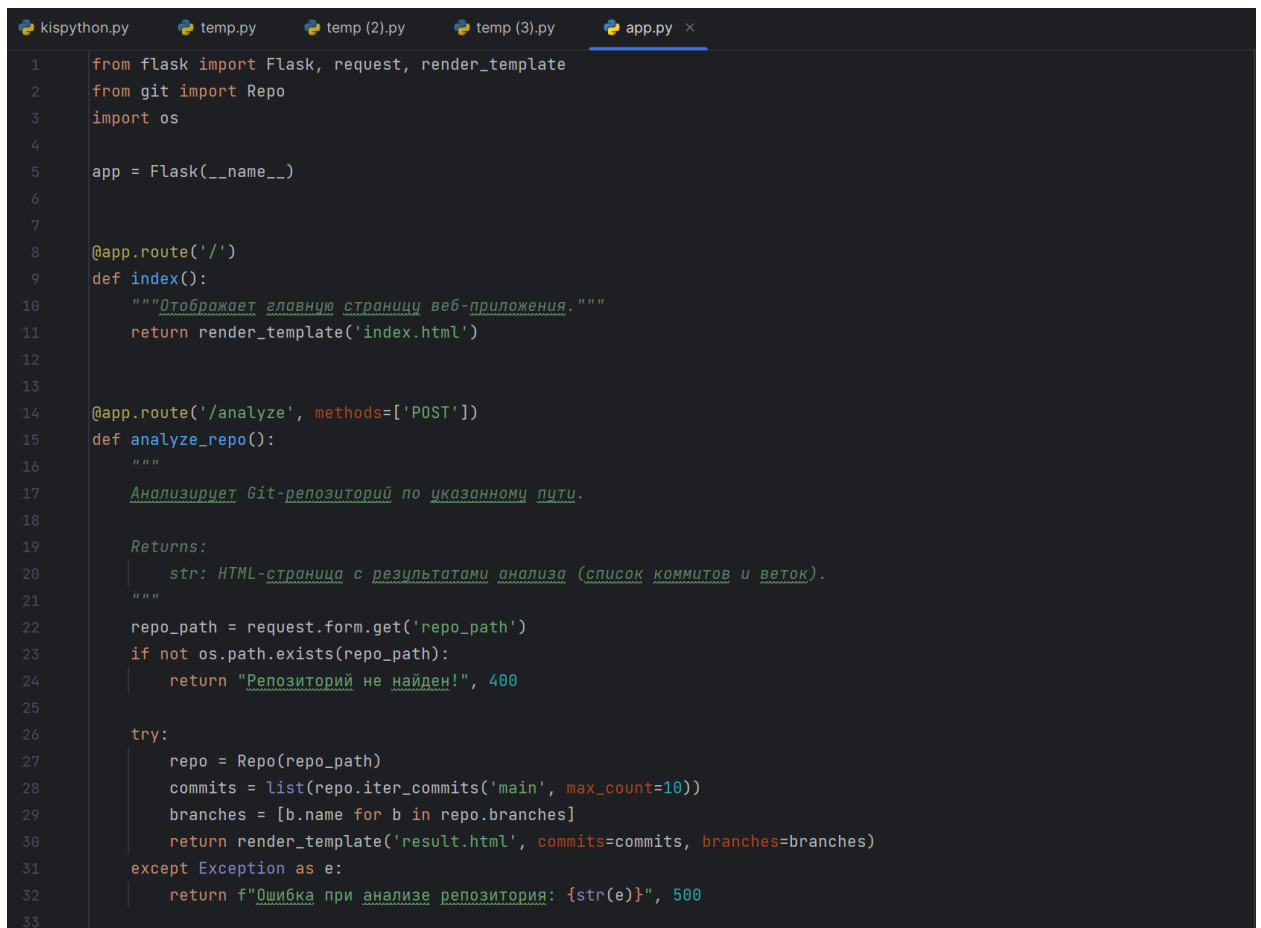
```
GNU nano 7.2 app.py *
from setuptools import setup, find_packages

setup(
    name="git-repo-analyzer",
    version="0.1.0",
    packages=find_packages(),
    install_requires=[
        'Flask==2.3.3',
        'GitPython==3.1.40',
        'Sphinx==7.2.6',
    ],
    author="Your Name",
    author_email="your.email@example.com",
    description="A simple web application to analyze Git repositories",
    long_description=open('README.md').read(),
    long_description_content_type="text/markdown",
    url="https://github.com/your-username/git-repo-analyzer",
    classifiers=[
        "Programming Language :: Python :: 3",
        "License :: OSI Approved :: MIT License",
        "Operating System :: OS Independent",
    ],
    python_requires='>=3.8',
)
```

Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark
Exit Read File Replace Paste Justify Go To Line M-E Redo M-G Copy

Рисунок 3 – Файл setup.py

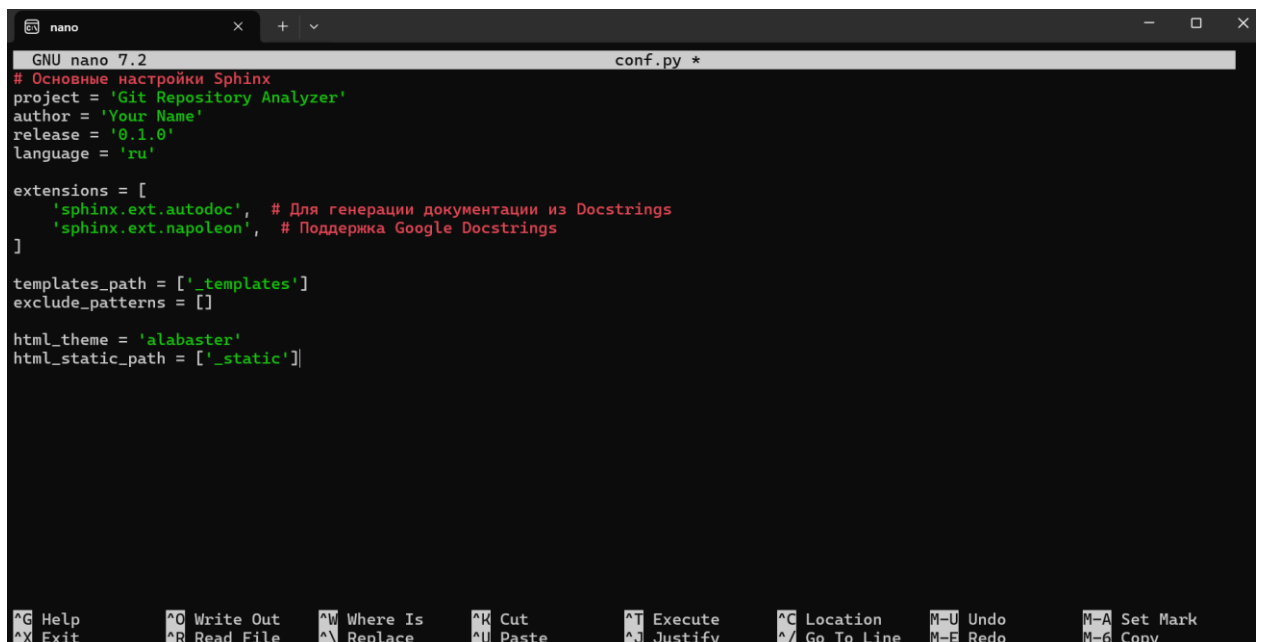
Продолжайте разработку проекта с использованием системы контроля версий. Снабжайте программный код разработанных классов, методов, функций и т.д. комментариями в формате Javadoc на Java или Docstrings на Python, рис 4.



```
1 from flask import Flask, request, render_template
2 from git import Repo
3 import os
4
5 app = Flask(__name__)
6
7
8 @app.route('/')
9 def index():
10     """Отображает главную страницу веб-приложения."""
11     return render_template('index.html')
12
13
14 @app.route('/analyze', methods=['POST'])
15 def analyze_repo():
16     """
17     Анализирует Git-репозиторий по указанному пути.
18
19     Returns:
20     str: HTML-страница с результатами анализа (список коммитов и веток).
21     """
22     repo_path = request.form.get('repo_path')
23     if not os.path.exists(repo_path):
24         return "Репозиторий не найден!", 400
25
26     try:
27         repo = Repo(repo_path)
28         commits = list(repo.iter_commits('main', max_count=10))
29         branches = [b.name for b in repo.branches]
30         return render_template('result.html', commits=commits, branches=branches)
31     except Exception as e:
32         return f"Ошибка при анализе репозитория: {str(e)}", 500
33
```

Рисунок 4 – Вид кода с Docstring на Python

Настроить систему сборки для сборки документации проекта. Для генерации документации используем Sphinx, файл conf.py, который создаёт HTML-документацию из Docstrings, рис 5.



```
GNU nano 7.2 conf.py *
# Основные настройки Sphinx
project = 'Git Repository Analyzer'
author = 'Your Name'
release = '0.1.0'
language = 'ru'

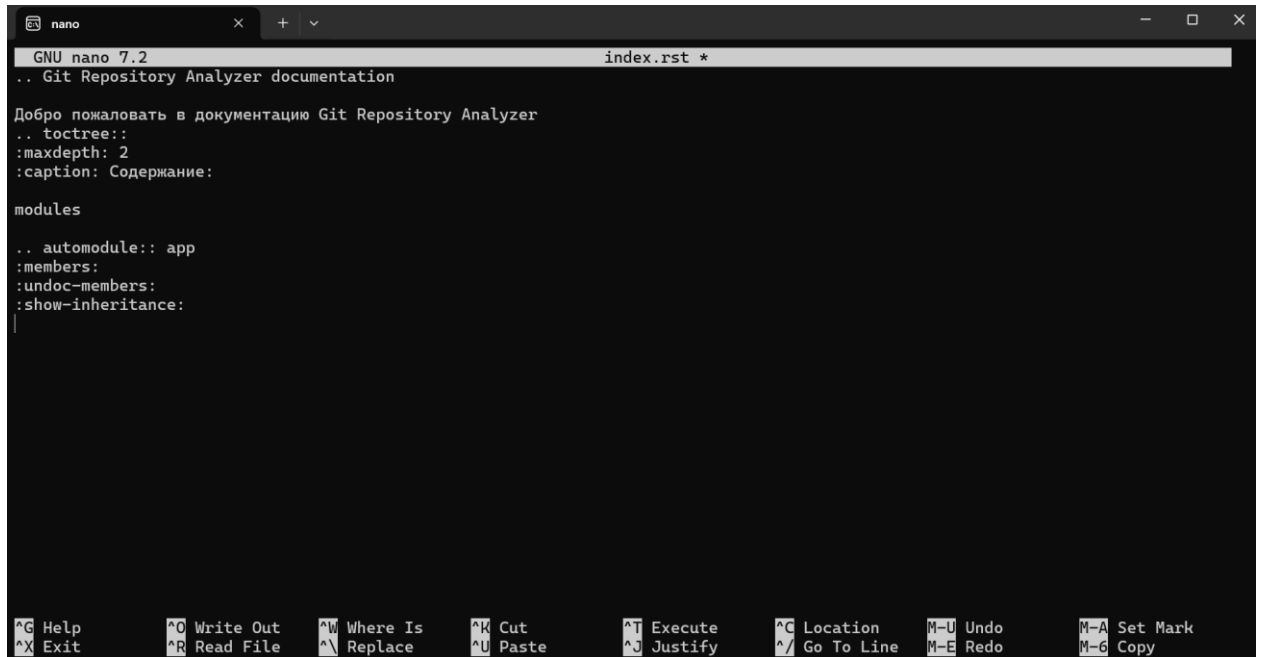
extensions = [
    'sphinx.ext.autodoc', # Для генерации документации из Docstrings
    'sphinx.ext.napoleon', # Поддержка Google Docstrings
]

templates_path = ['_templates']
exclude_patterns = []

html_theme = 'alabaster'
html_static_path = ['_static']
```

Рисунок 5 – Файл conf.py

index.rst: Главный файл документации, который включает модуль app и генерирует описание функций из Docstrings, рис 6.



```
GNU nano 7.2 index.rst *
.. Git Repository Analyzer documentation

Добро пожаловать в документацию Git Repository Analyzer
.. toctree::
:maxdepth: 2
:caption: Содержание:

modules

.. automodule:: app
:members:
:undoc-members:
:show-inheritance:
```

Рисунок 6 – Файл index.rst