**Приложение 2**

**PyTPretator**

**Для чего?**

Сайт разработан для изучения основ языка программирования Python, а также практики. Обучающий контент подобран с учетом основных пользователей – школьников. Справочные материалы содержат информацию по самым основным темам, необходимым ученику для решения простейших задач формата ОГЭ, ЕГЭ и старта в программировании.

**Какие технологии используются?**

Язык программирования – Python 3.10.2

Backend:

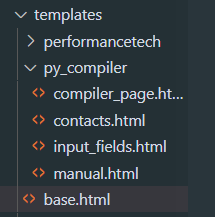
Django 3.2 - <https://www.djangoproject.com/>

Frontend:

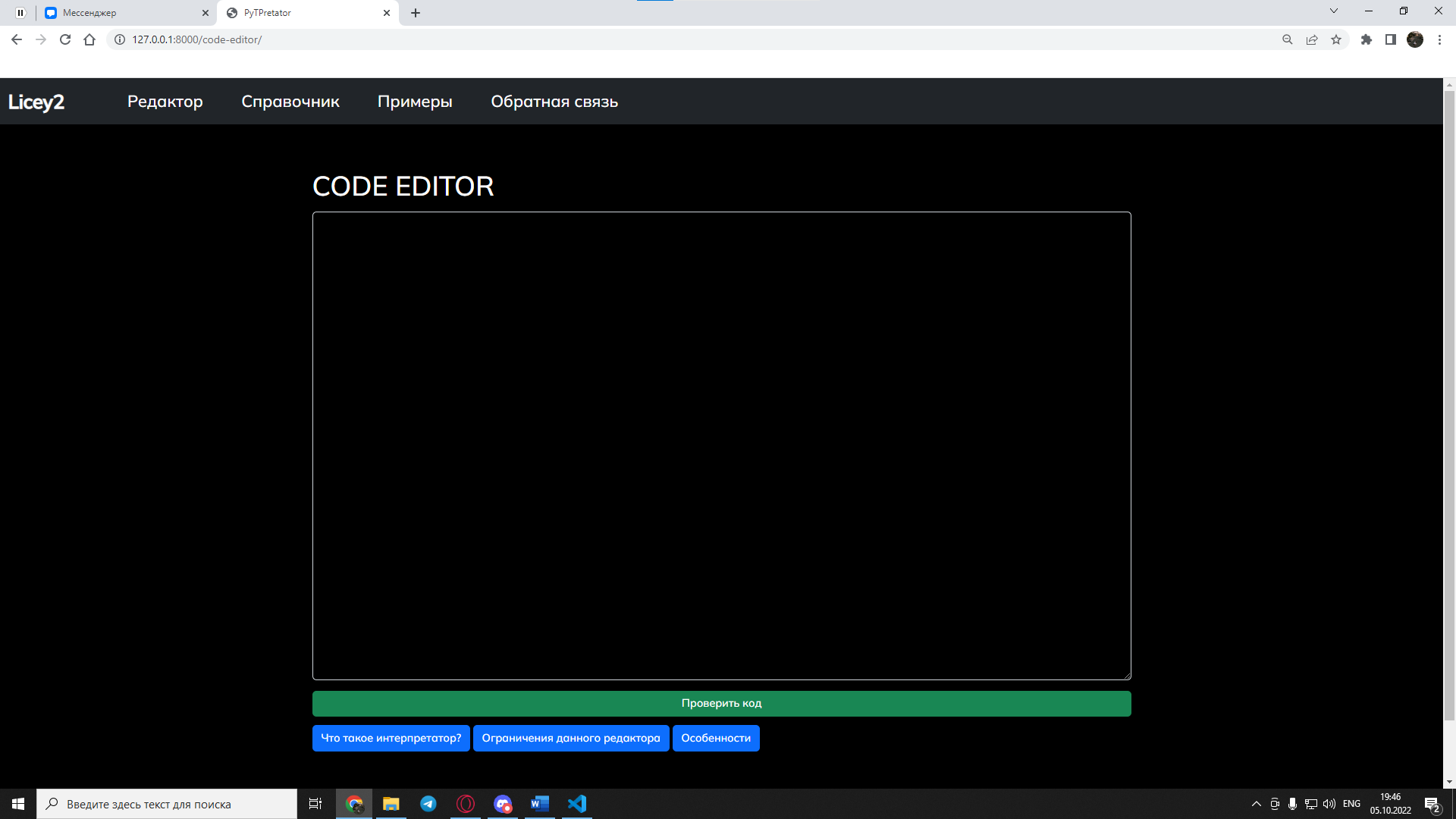
Bootstrap 5 (+ jQuery) - <https://getbootstrap.com/>

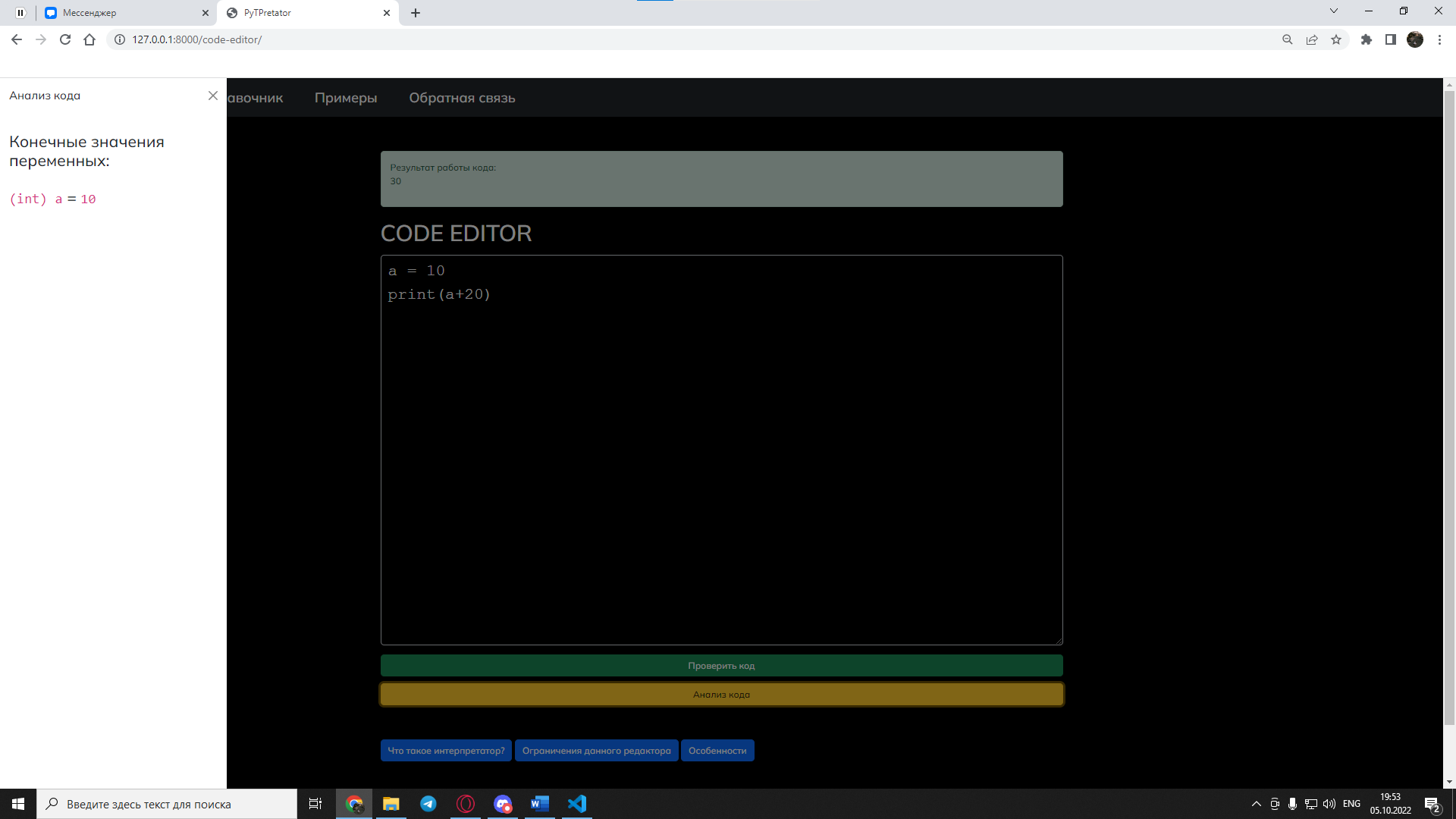
**Ход разработки**

1. Был создан каркас Django-проекта с помощью инструментов, предоставляемых фреймворком.
2. Создано приложение py\_compiler. Изначально оно имело именно такое название, а не PyTPretator. Затем, в связи с неуместным использованием слова «compiler» для интерпретируемого языка, было принято решение переименовать приложение, однако в папке проекта оно по-прежнему имеет старое название.
3. Для приложения был сверстан главный шаблон, от которого наследуются остальные шаблоны страниц. Для верстки применялись технологии Bootstrap, которые позволяют создавать приложения в строгом, но стильном виде.

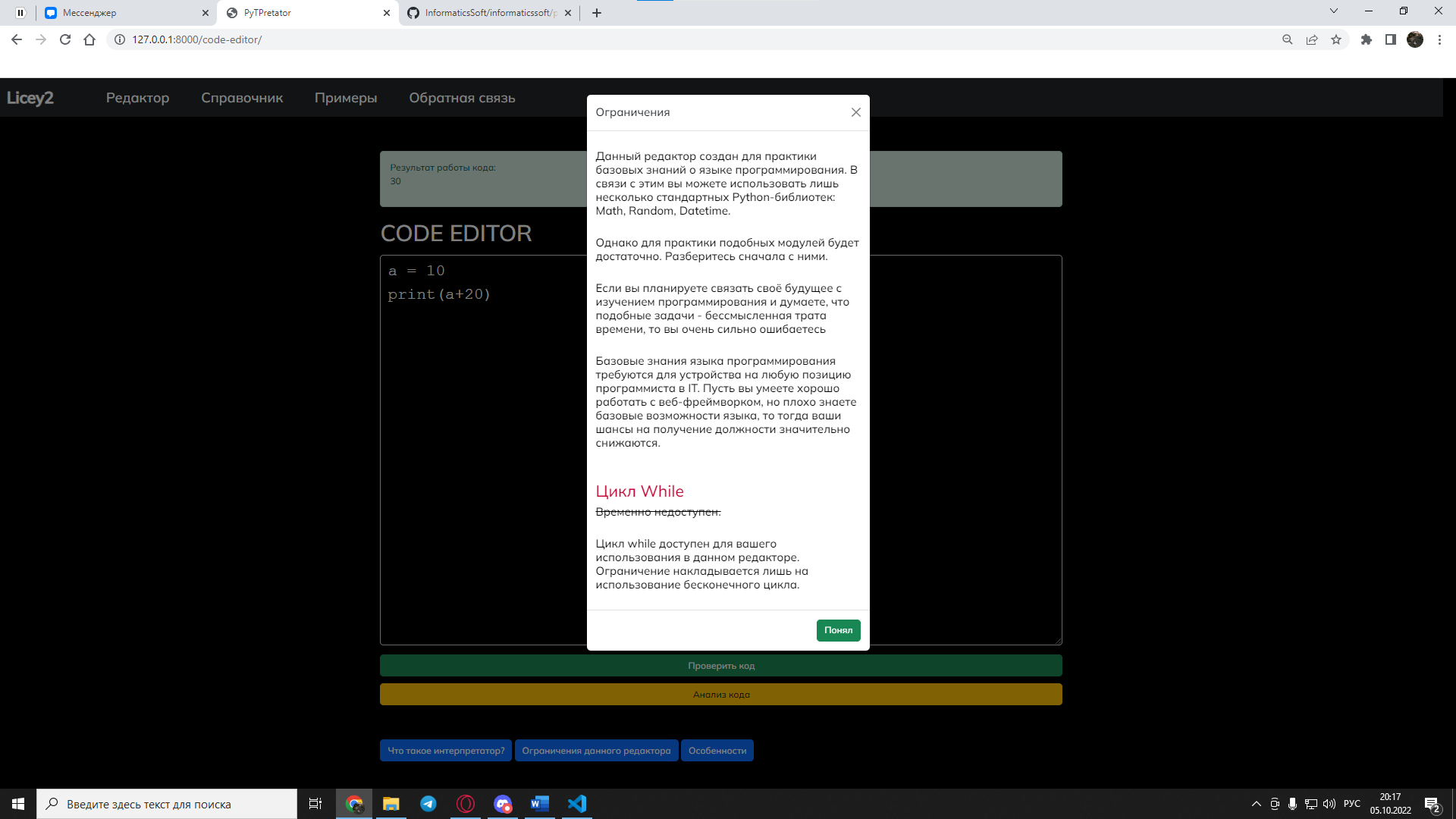


1. Первой была создана главная страница с редактором кода. Пользователь может писать код на Python, затем запускать его и получать выводимый результат. Кроме того, ему доступен анализатор кода – инструмент, с помощью которого пользователь может увидеть все созданные им переменные, их конечные значения и типы данных.



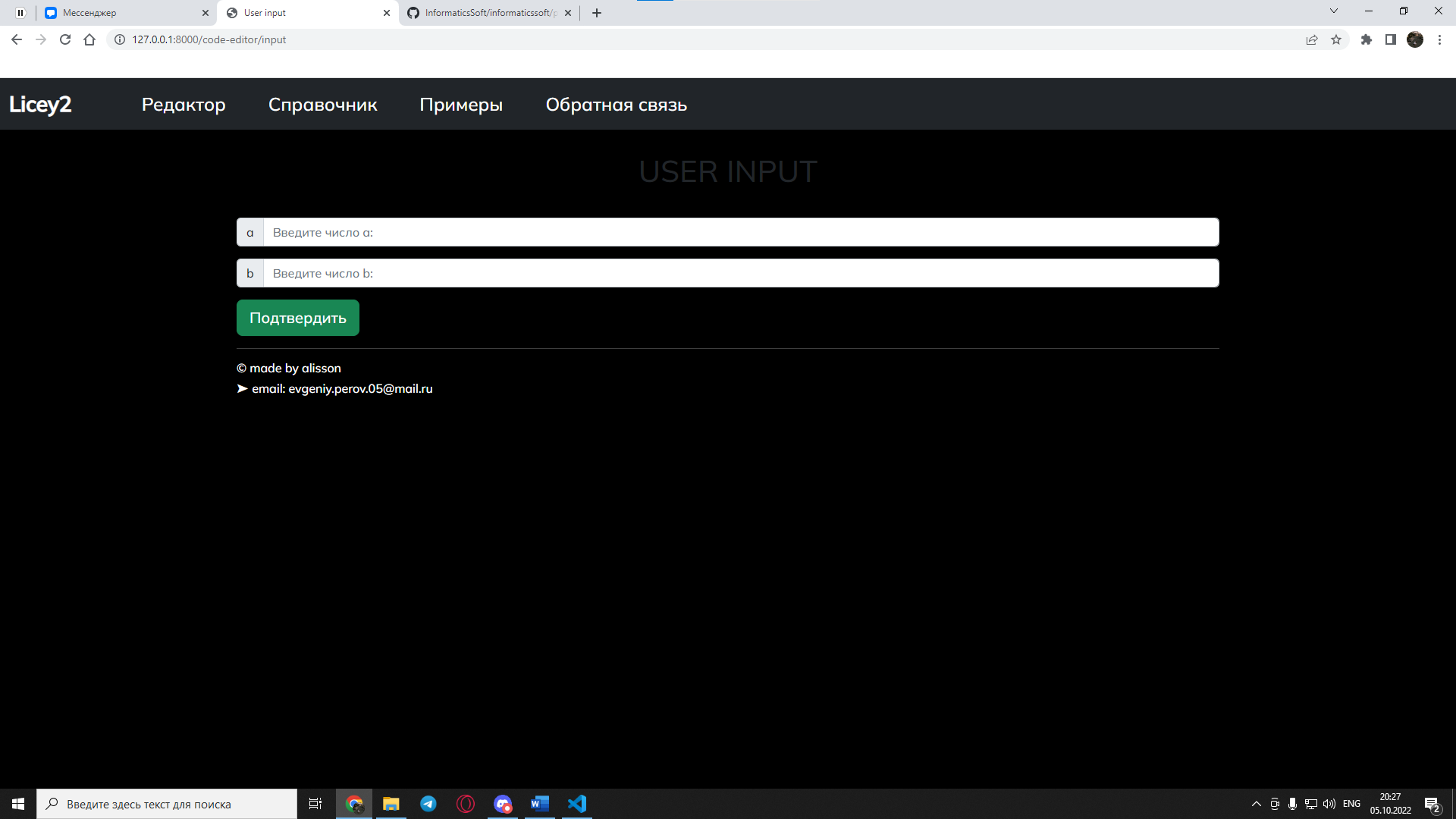


1. Код, написанный пользователем, исполняется на устройстве, на котором запущен проект (на сервере). Поэтому, было необходимо предусмотреть некоторые ограничения по возможностям редактора. Так, доступ к библиотекам ограничен до трёх: Math, Random, Datetime. Однако этого достаточно для учебного программирования. Использование бесконечных циклов запрещено. Функции write() и read() для работы с файлами запрещены.

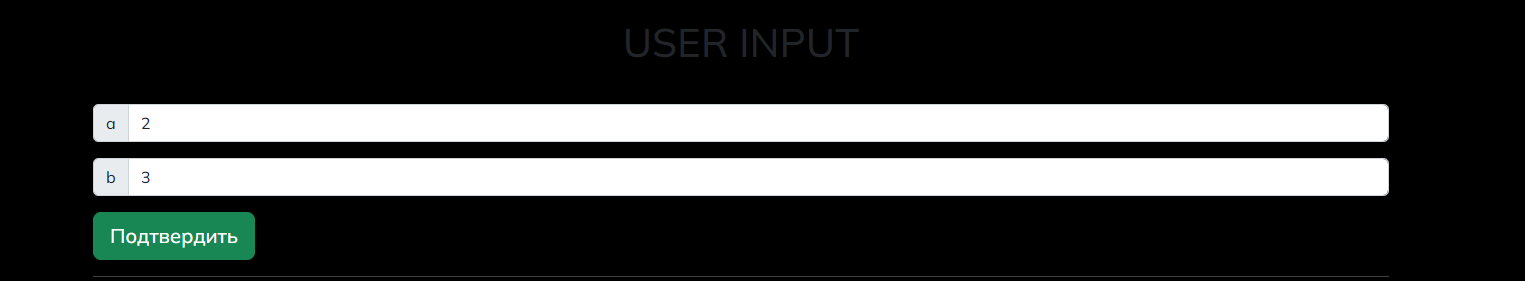


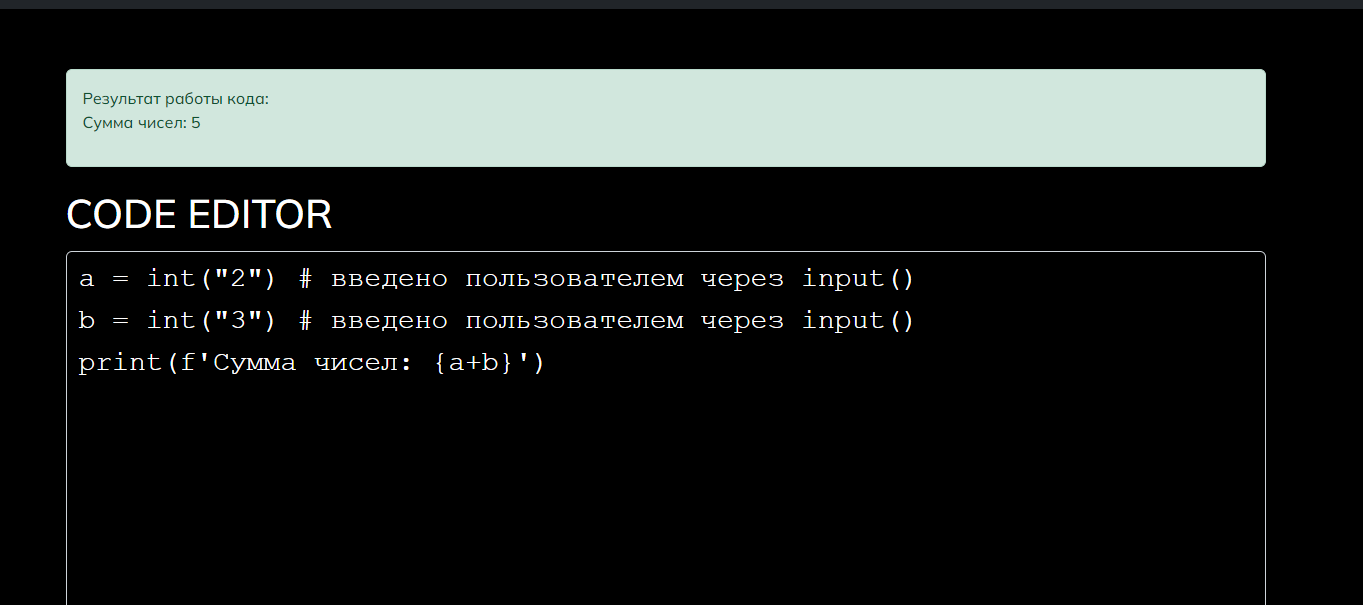
1. Поскольку input() приостанавливает работу кода до ввода значения, допускать использование этой функции напрямую пользователям не нужно, иначе это будет иметь такой же эффект, что и при бесконечном цикле: код, отвечающий за отрисовку страницы не будет исполняться до получения значения от пользователя. Однако, данная функция является одной из самых важных, поэтому было придумано альтернативное решение: После нажатия на кнопку «Проверить код», при наличии input() в коде, пользователя перенесет на отдельную страницу с полями ввода, выглядит это следующим образом:



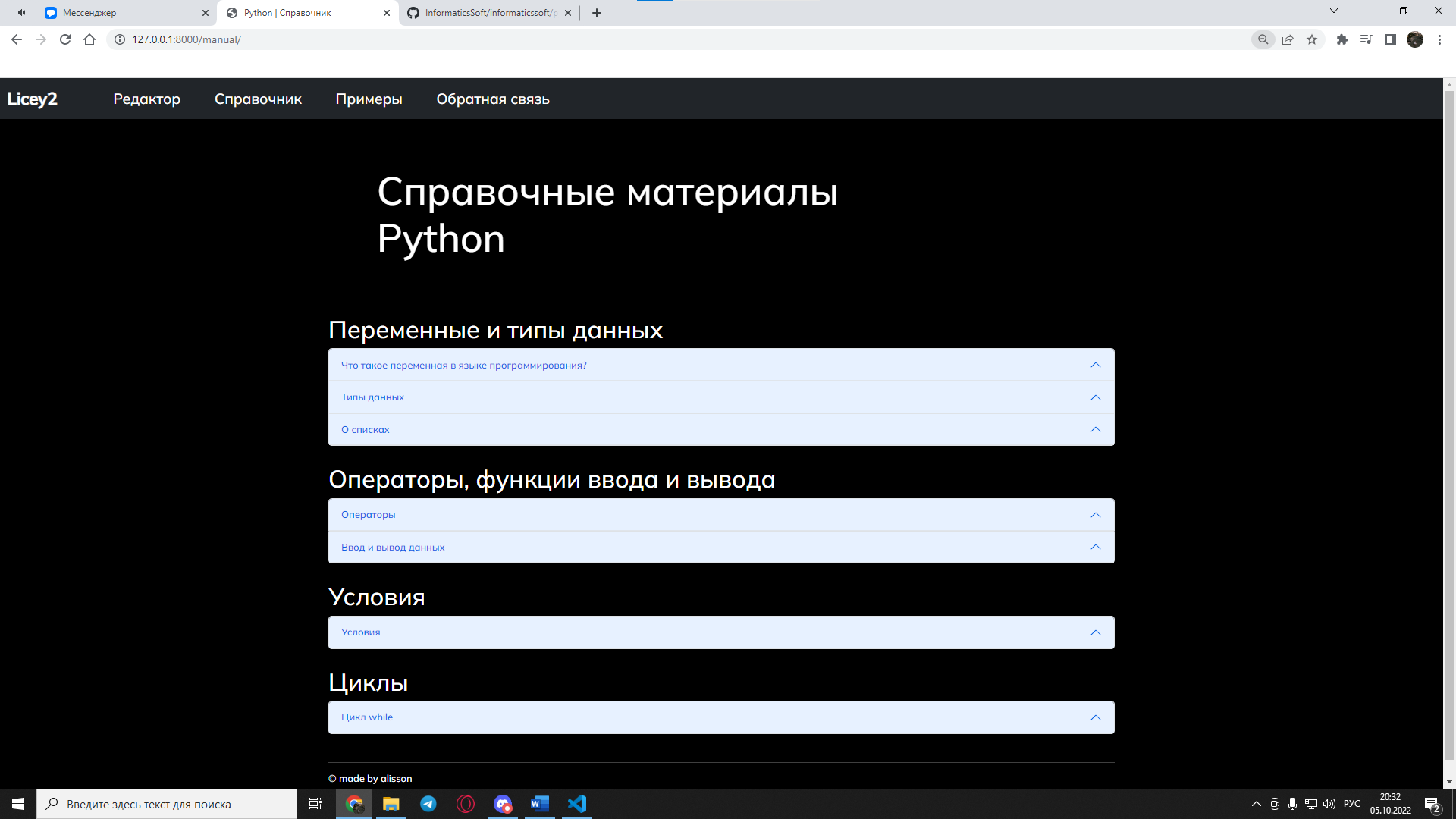


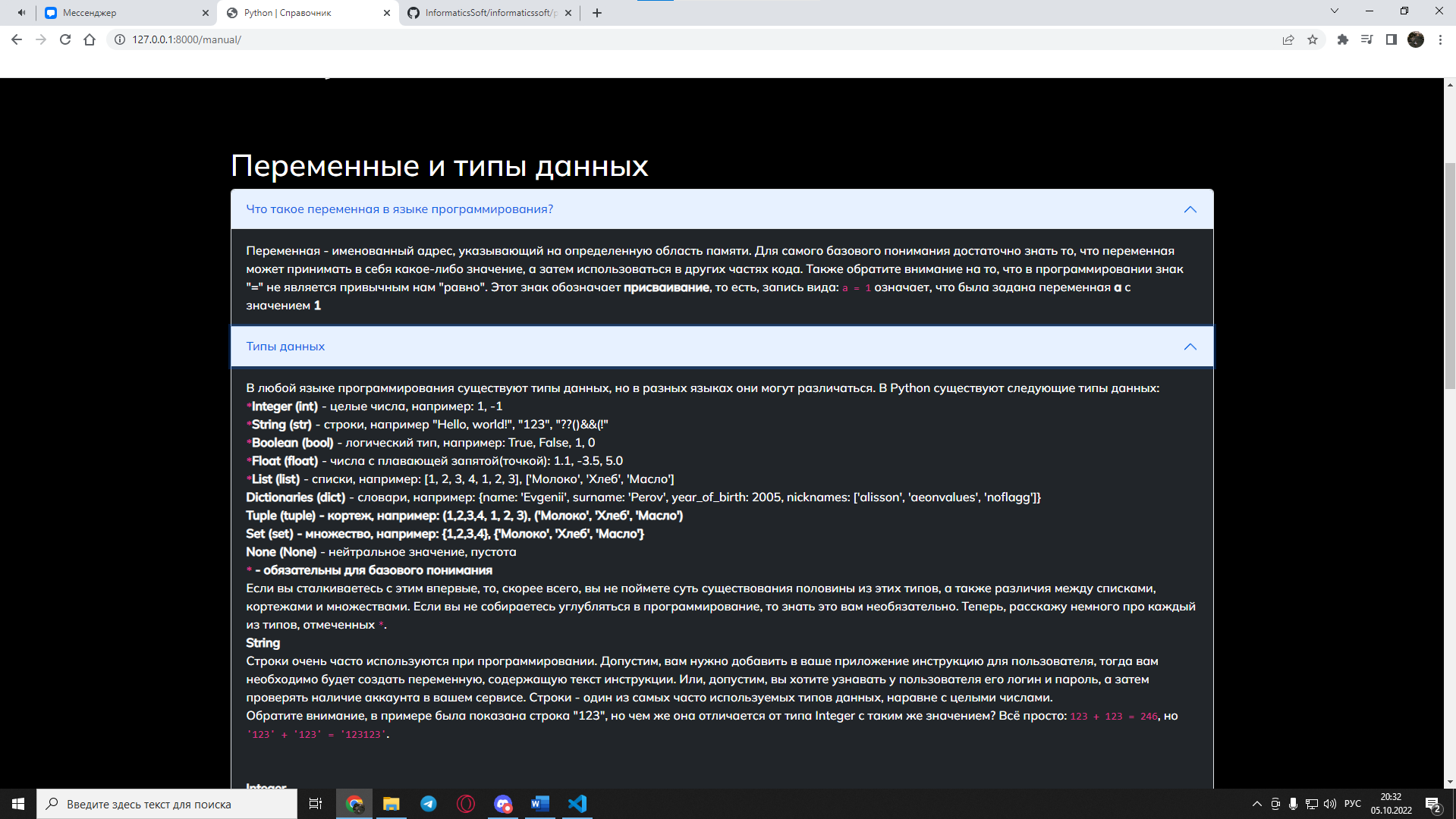
После нажатия на кнопку «Подтвердить», пользователь вернется к коду, в котором все input() будут заменены на введенные значения.



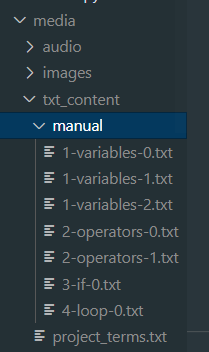


1. Затем был реализован раздел «Справочник». В нём пользователь может ознакомиться с материалами по языку программирования.





1. Контент подставляется в шаблон из текстовых файлов, расположенных в папке «media» на одном уровне с папкой приложения.



1. В разделе «Примеры» загружены видеоролики с моими объяснениями тем.

Ознакомиться с кодом PyTPretator вы можете на GITHUB: <https://github.com/aeonva1ues/InformaticsSoft/tree/master/informaticssoft/py_compiler>

Также, на YouTube загружен обзор проекта: <https://disk.yandex.ru/i/Nbue6fvtCiY2wQ>