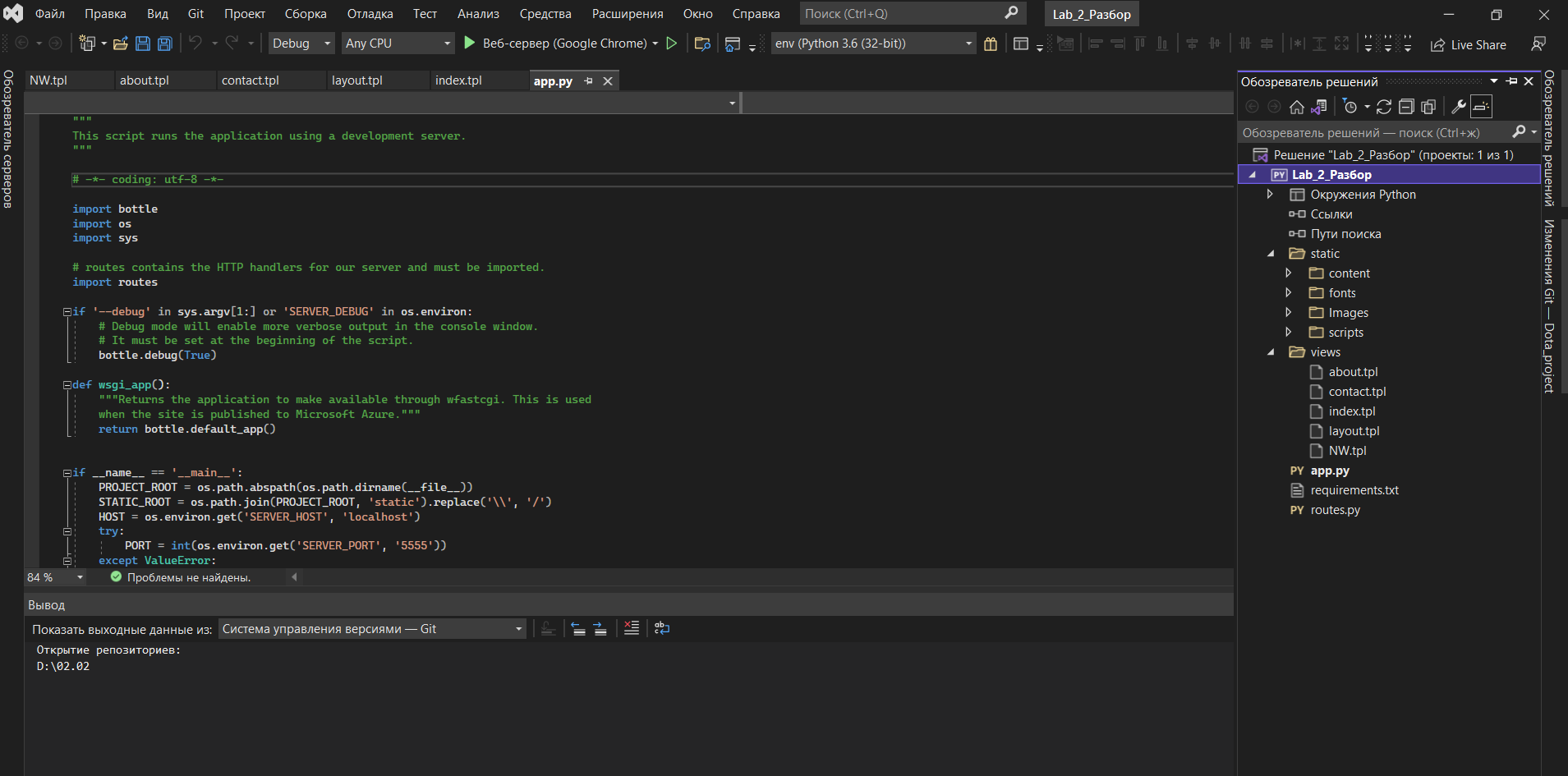
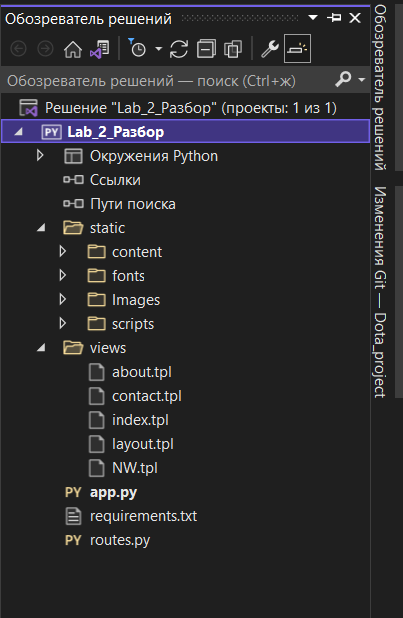
**Лабораторная работа №2-4**

**Цель**: создание модульной структуры проекта в виде диаграммы, разработка перечня артефактов (основных файлов) и протоколов, настройка Git.

1. Создание файла



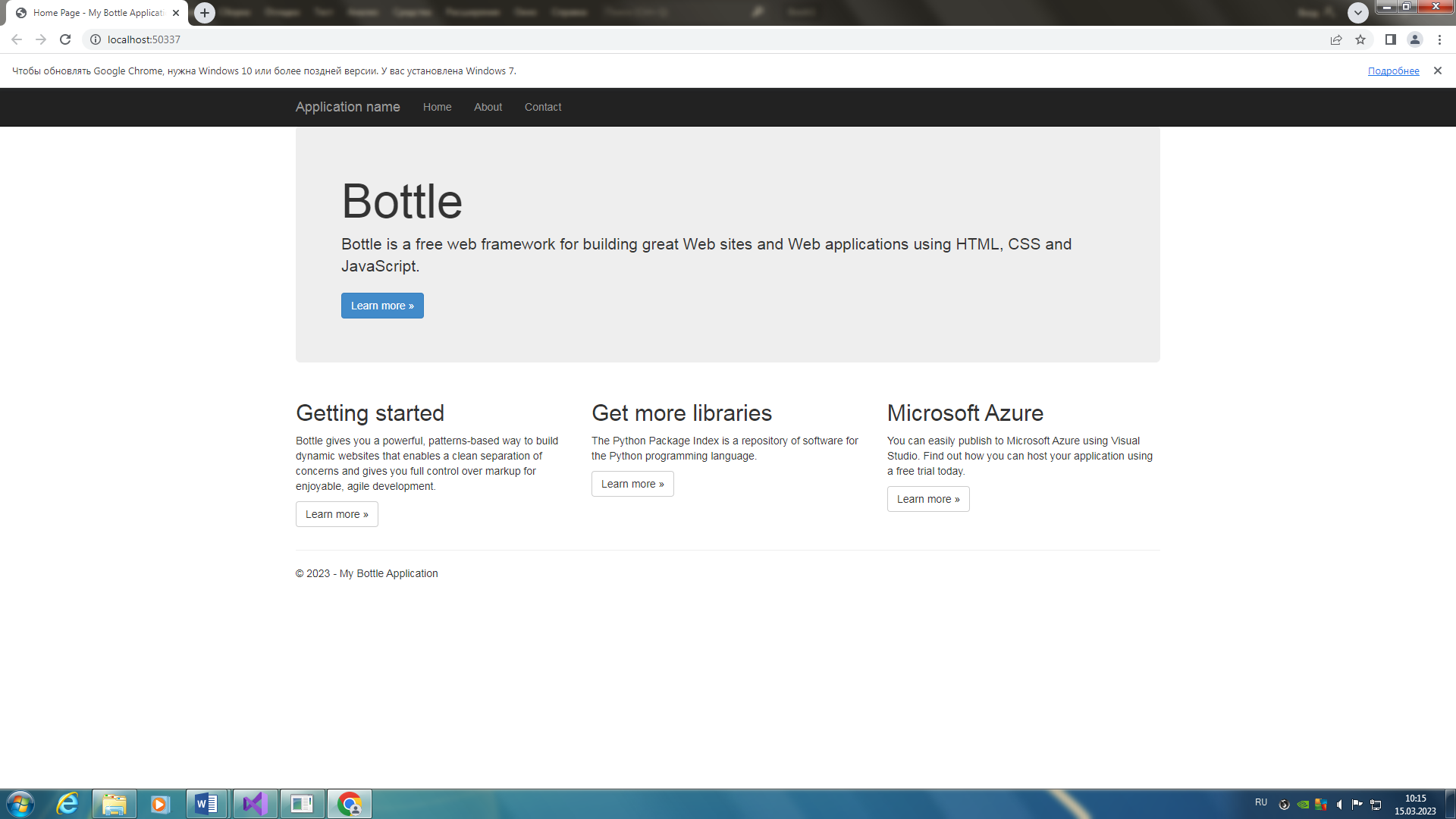
1. Обозреватель решений



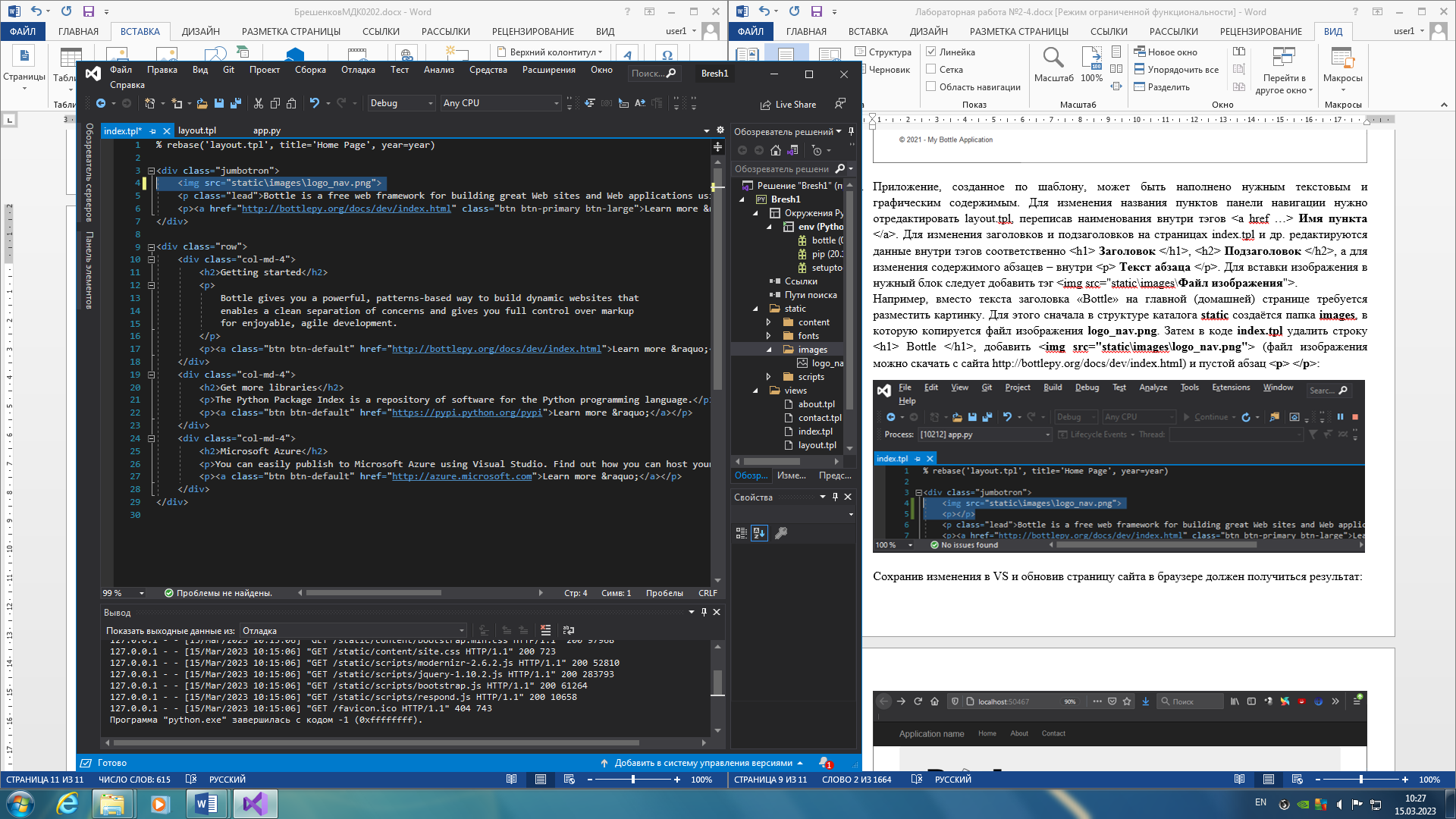
В папке **static** содержатся стилевые css-файлы для оформления внешнего вида веб-страниц, шрифтовые наборы и js-скрипты.

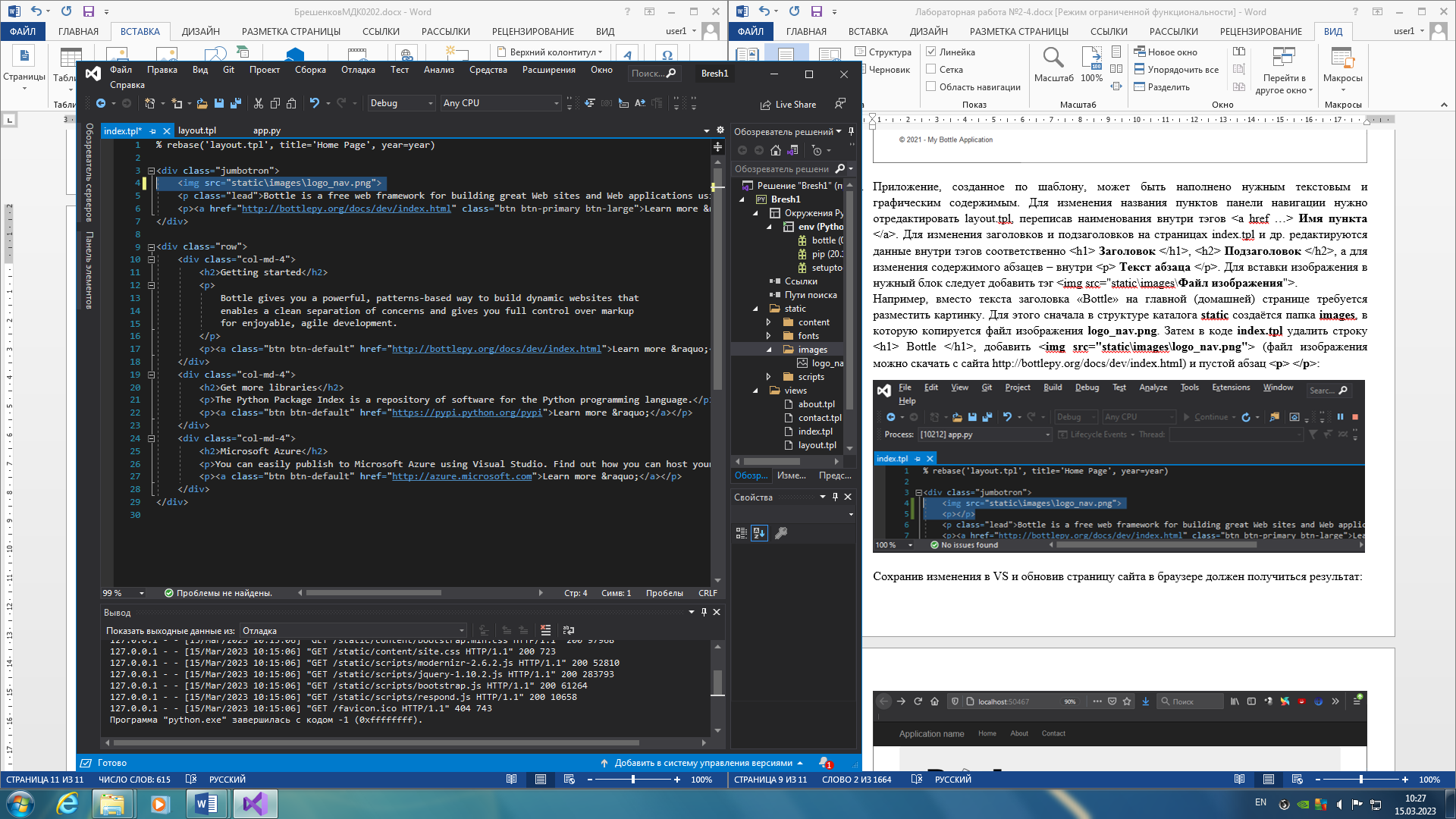
Выбрав для просмотра файлы из **папки** views, можно убедиться, что они представляют собой макеты веб-страниц: **главной** (index.tpl), **о нас** (about.tpl), **контакты** (contact.tpl), а файл layout.tpl – не что иное, как шаблон обёртки с панелью навигации, содержащей четыре гиперссылки: Application name, Home, About, Contact в «шапке» сайта и небольшим «подвалом».

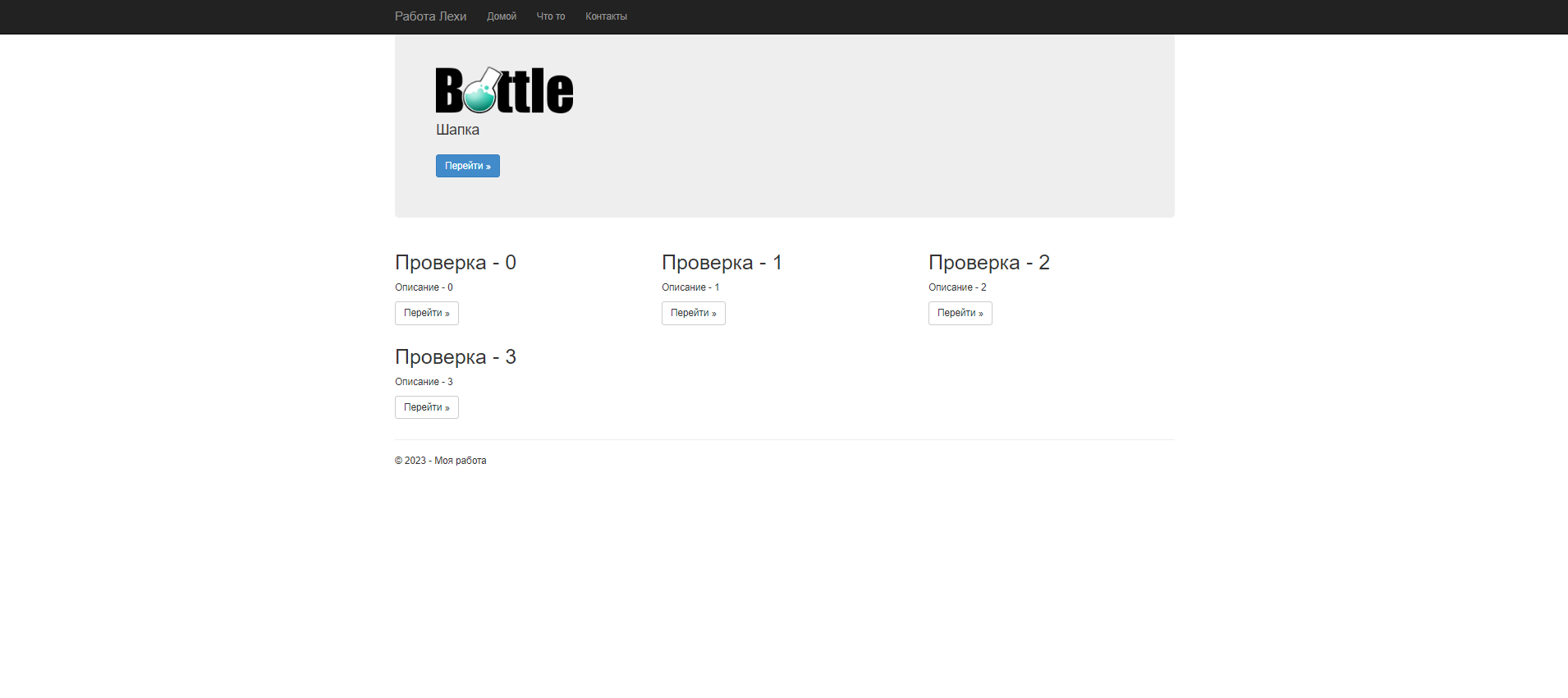
1. Выполнить запуск проекта



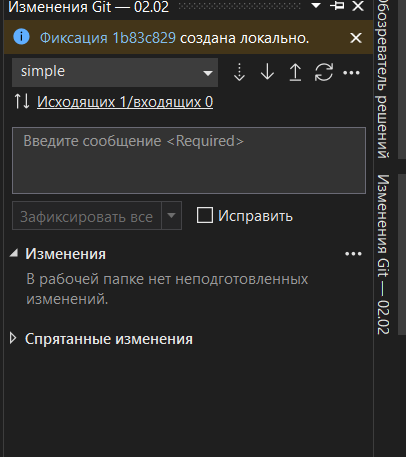
1. Добавление картинки

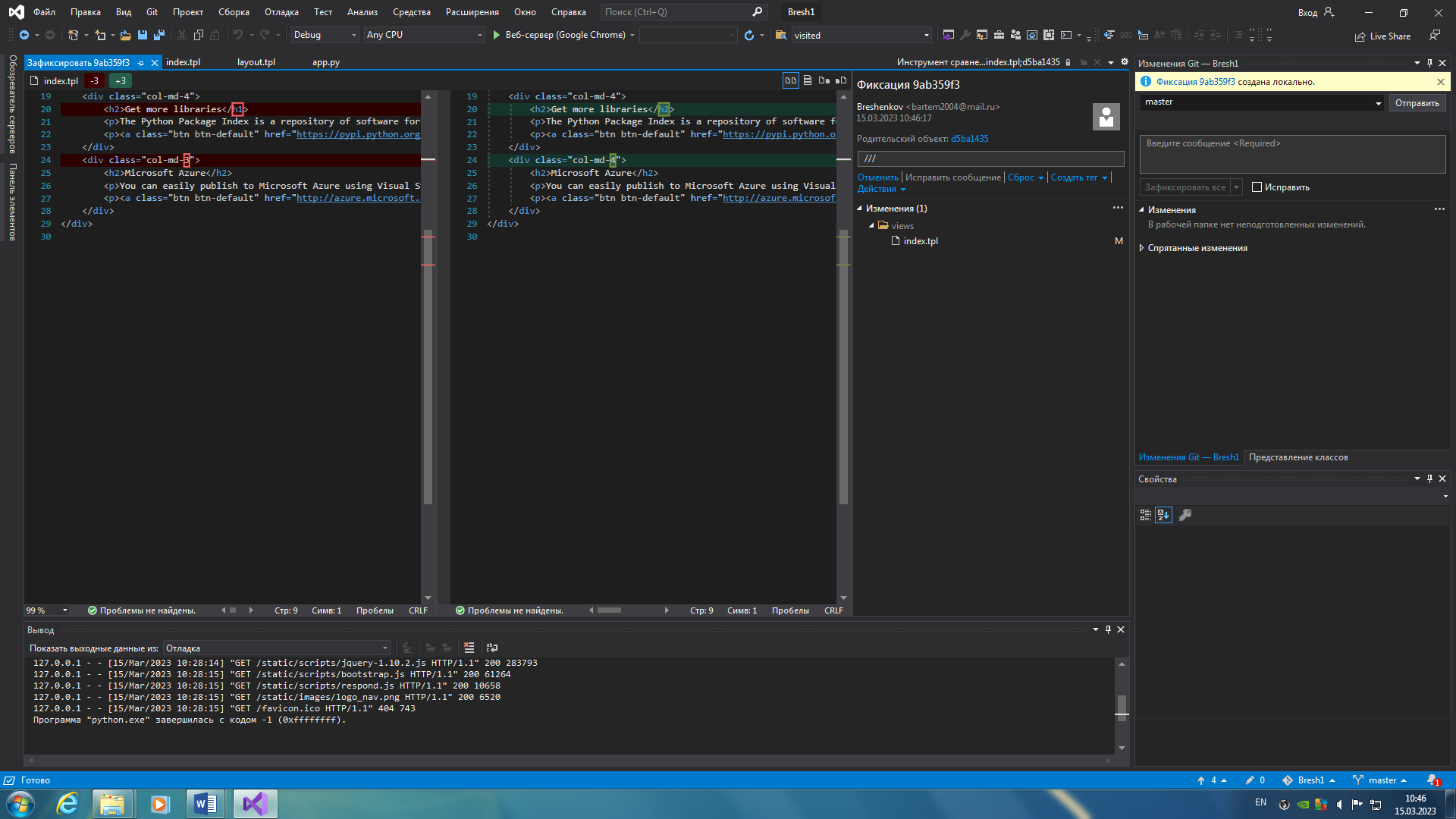


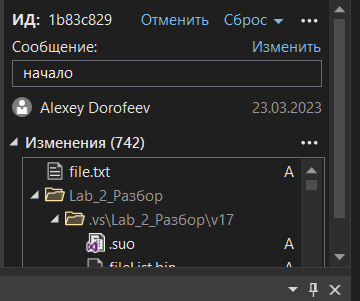




1. Git настройка



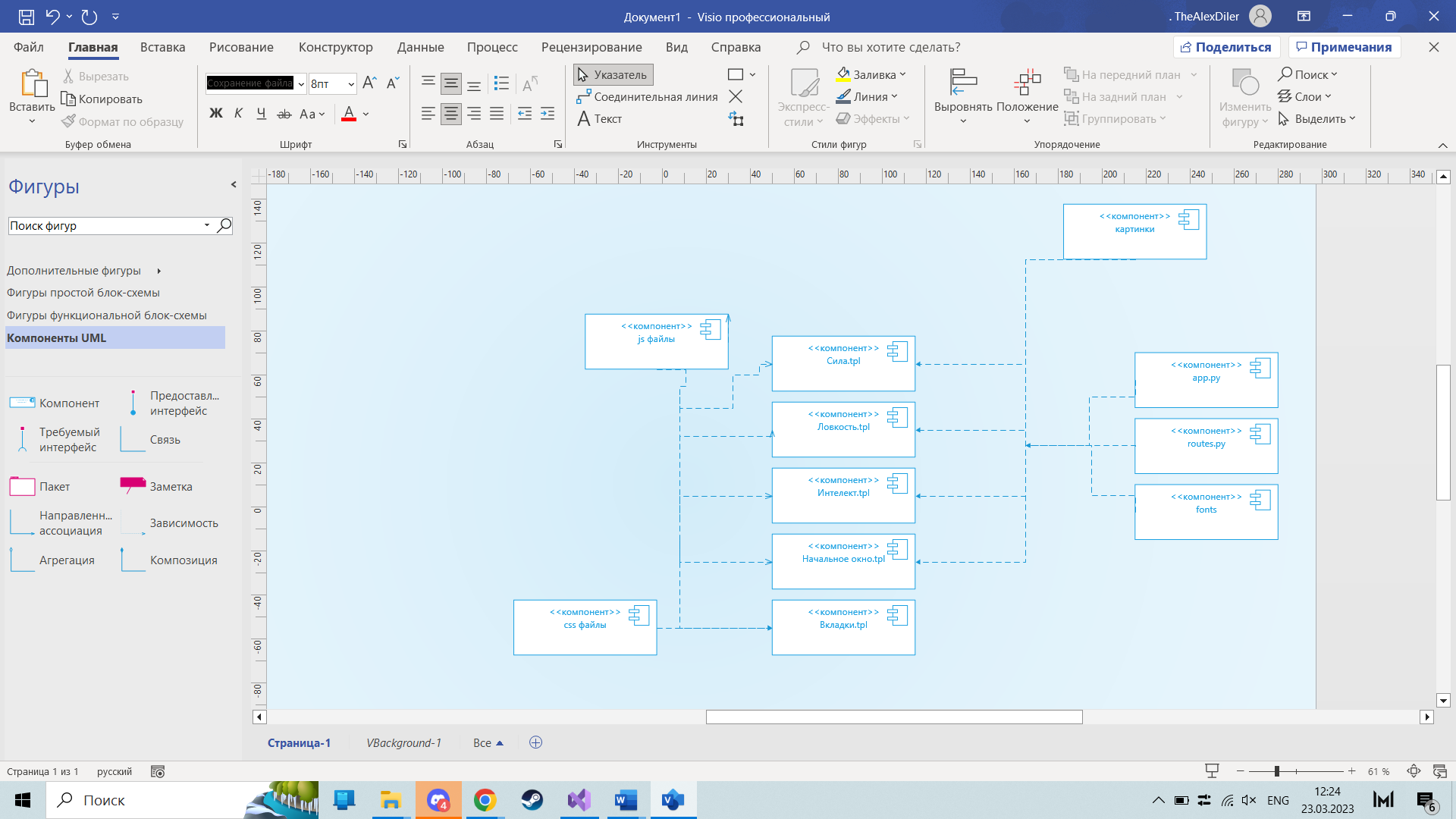




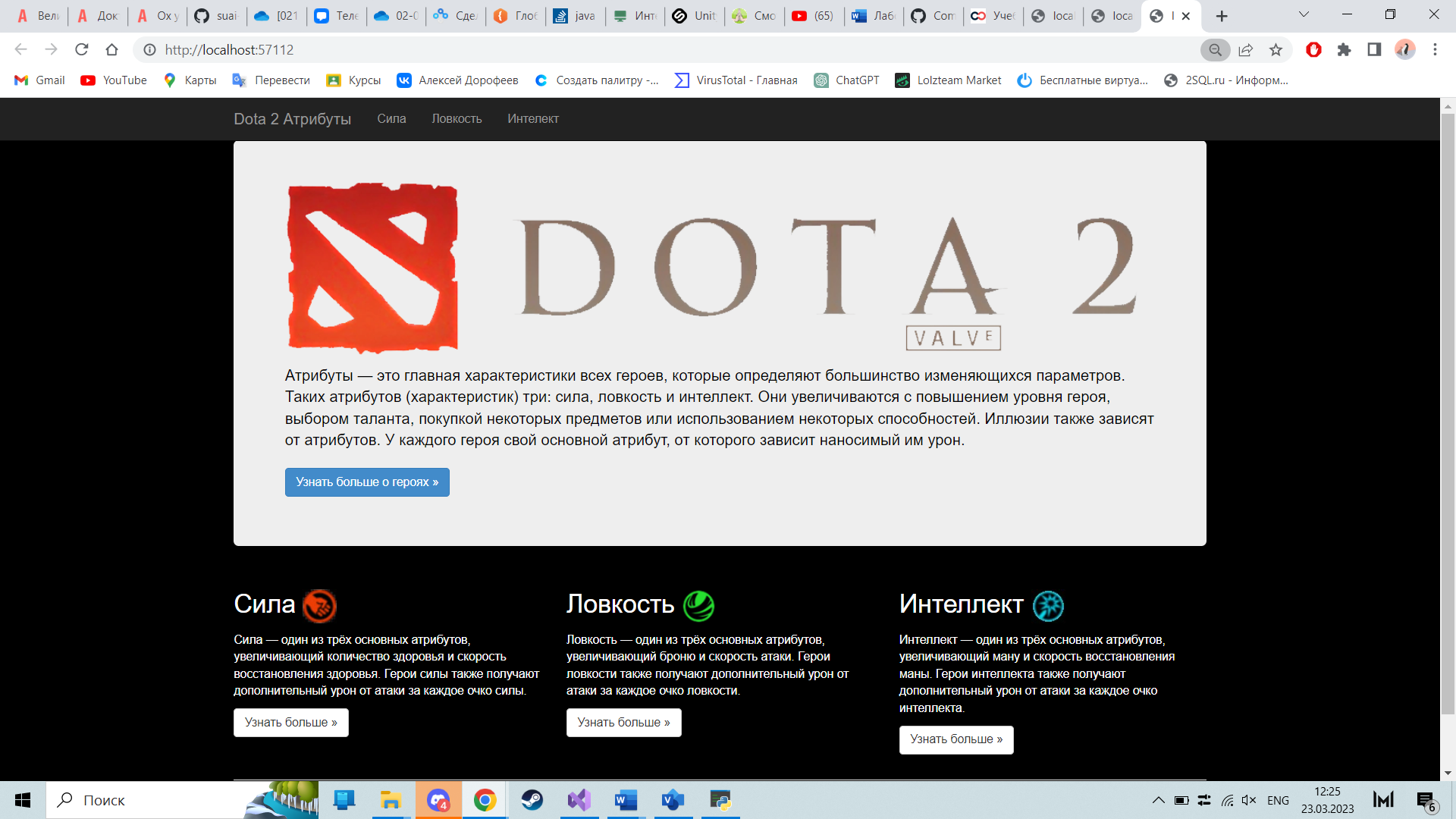
1. Требуется создать веб-приложение (тематический сайт) на основе шаблона фреймворка Bottle.

Тематика приложения: Dota 2 Атрибуты

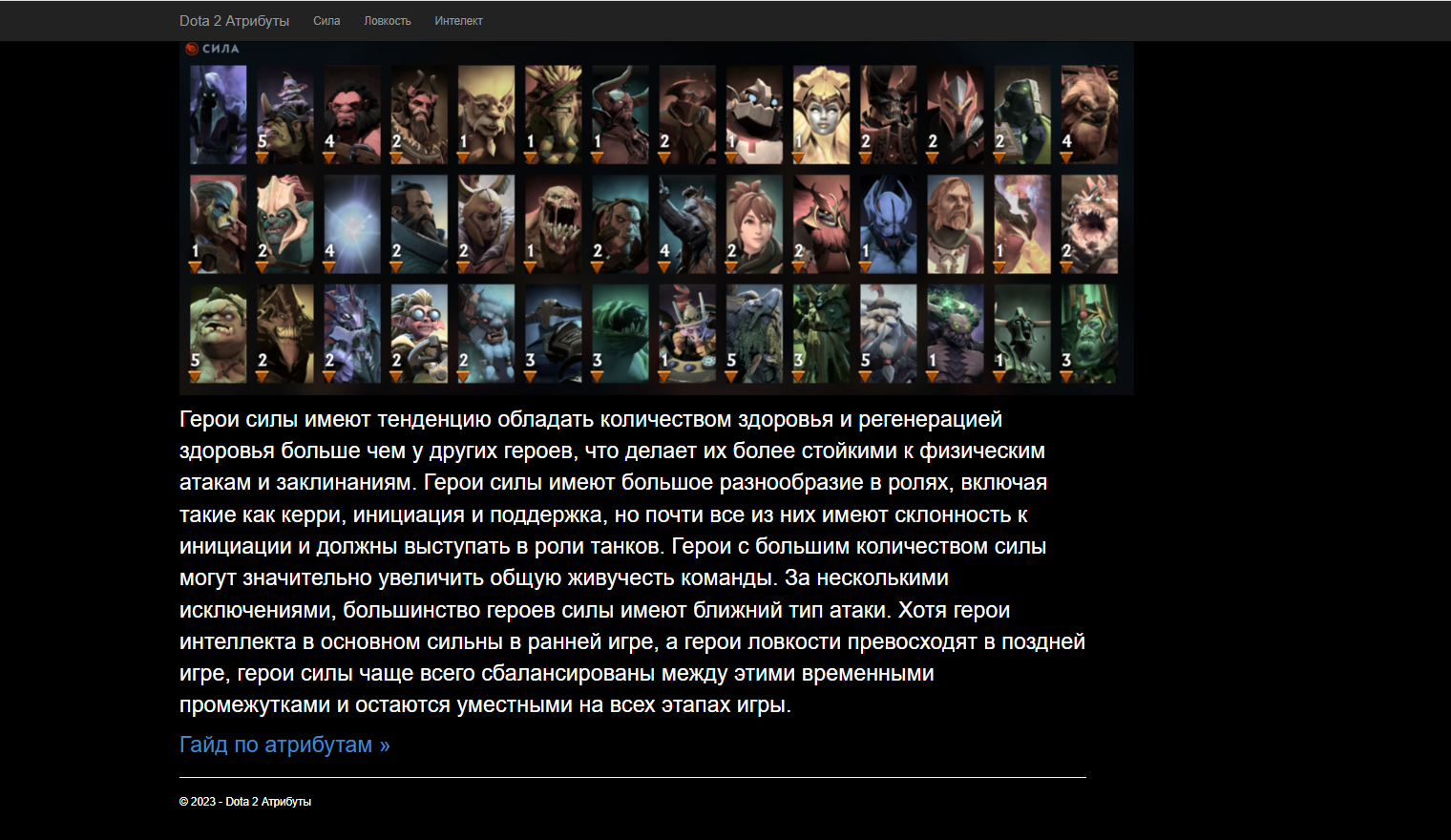
UML-диаграмма компонентов приложения:



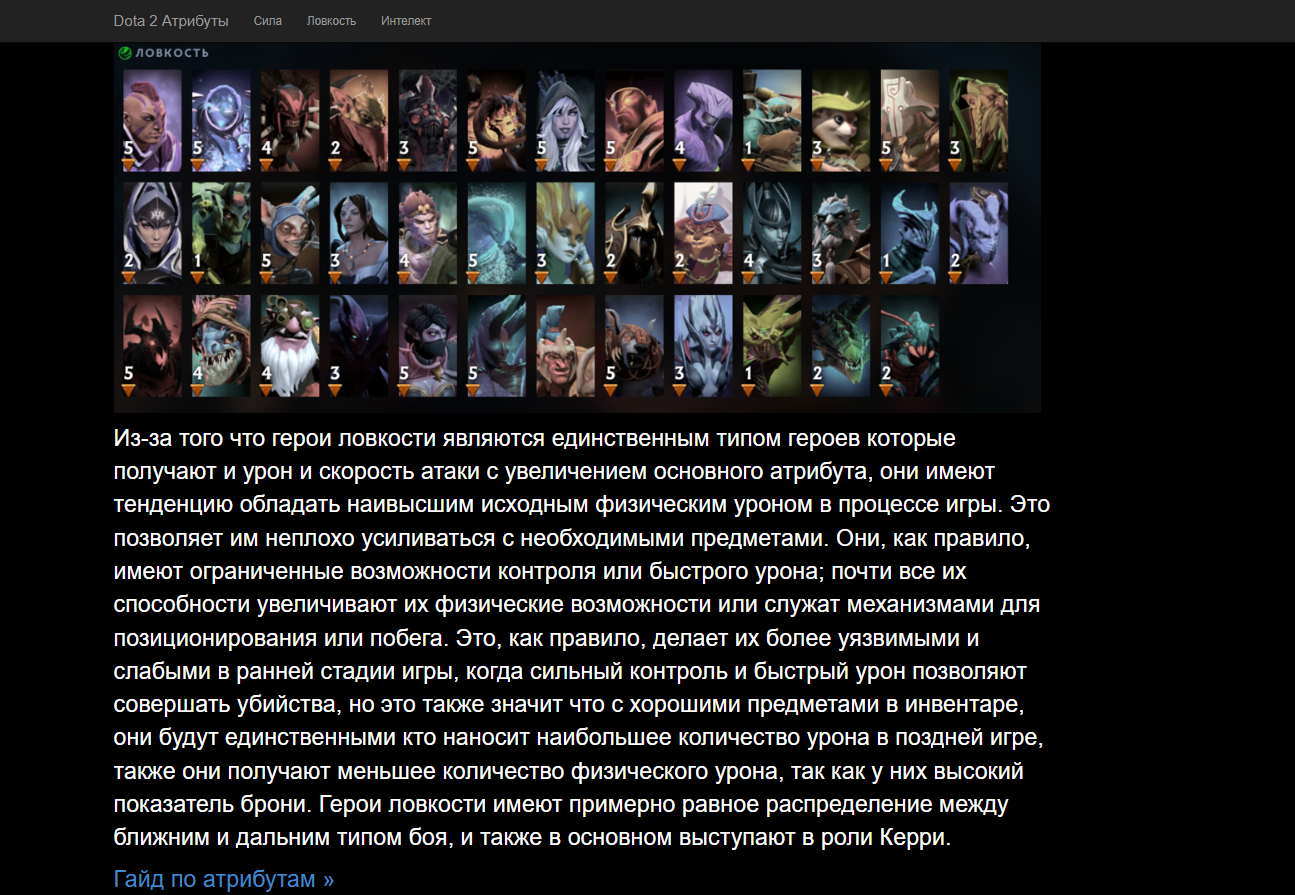
Внешний вид приложения:



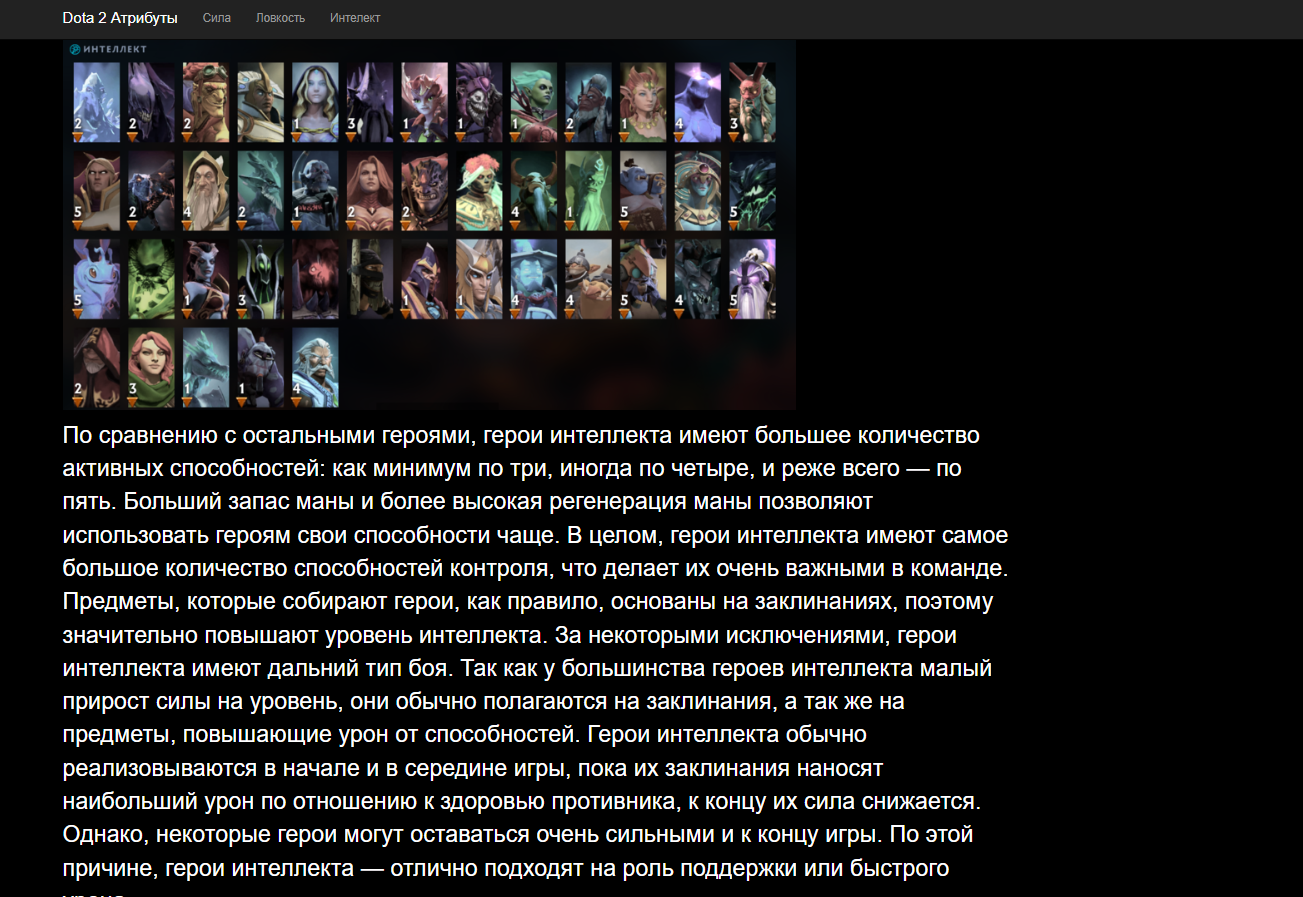
Начальная страница



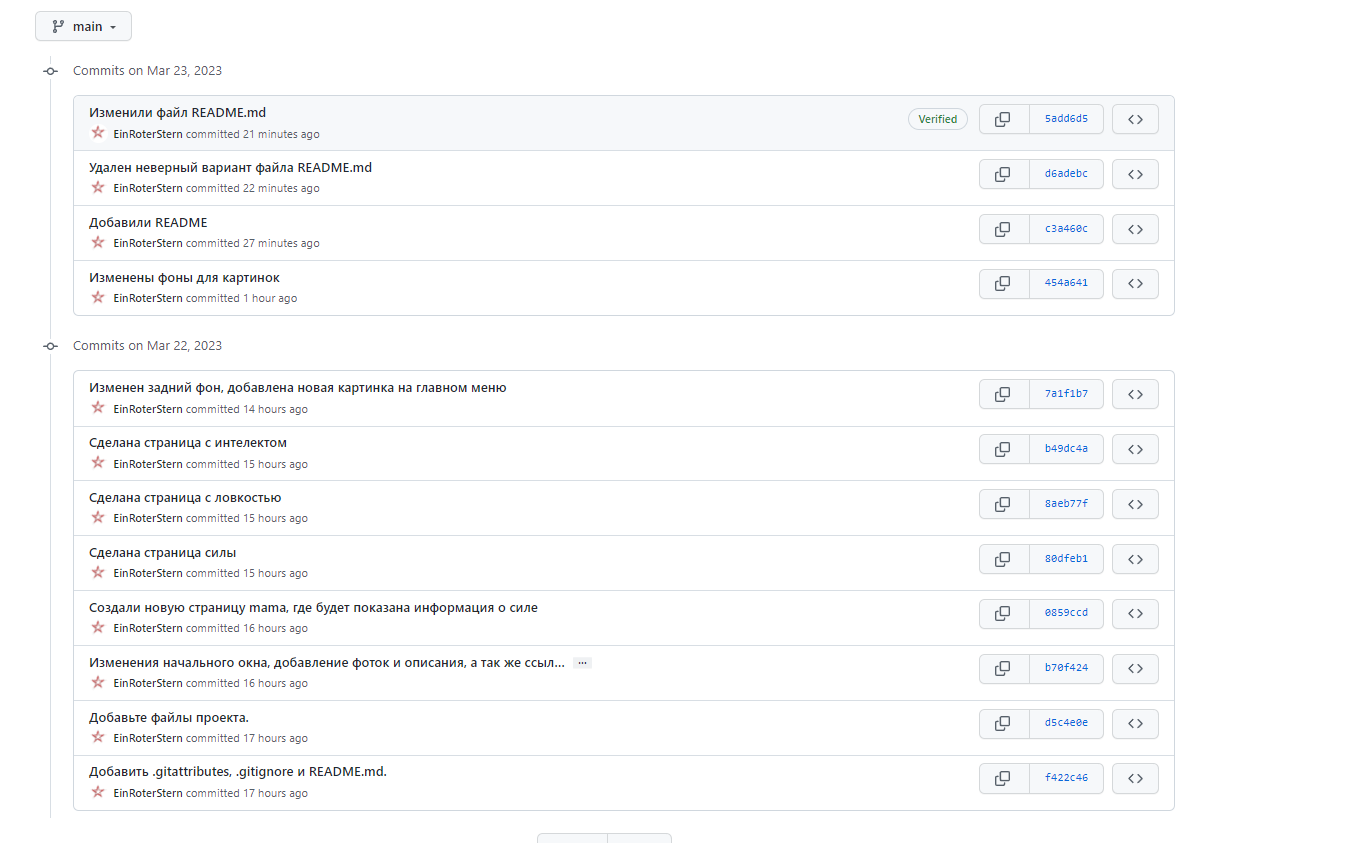
Страница Сила



Страница Ловкость

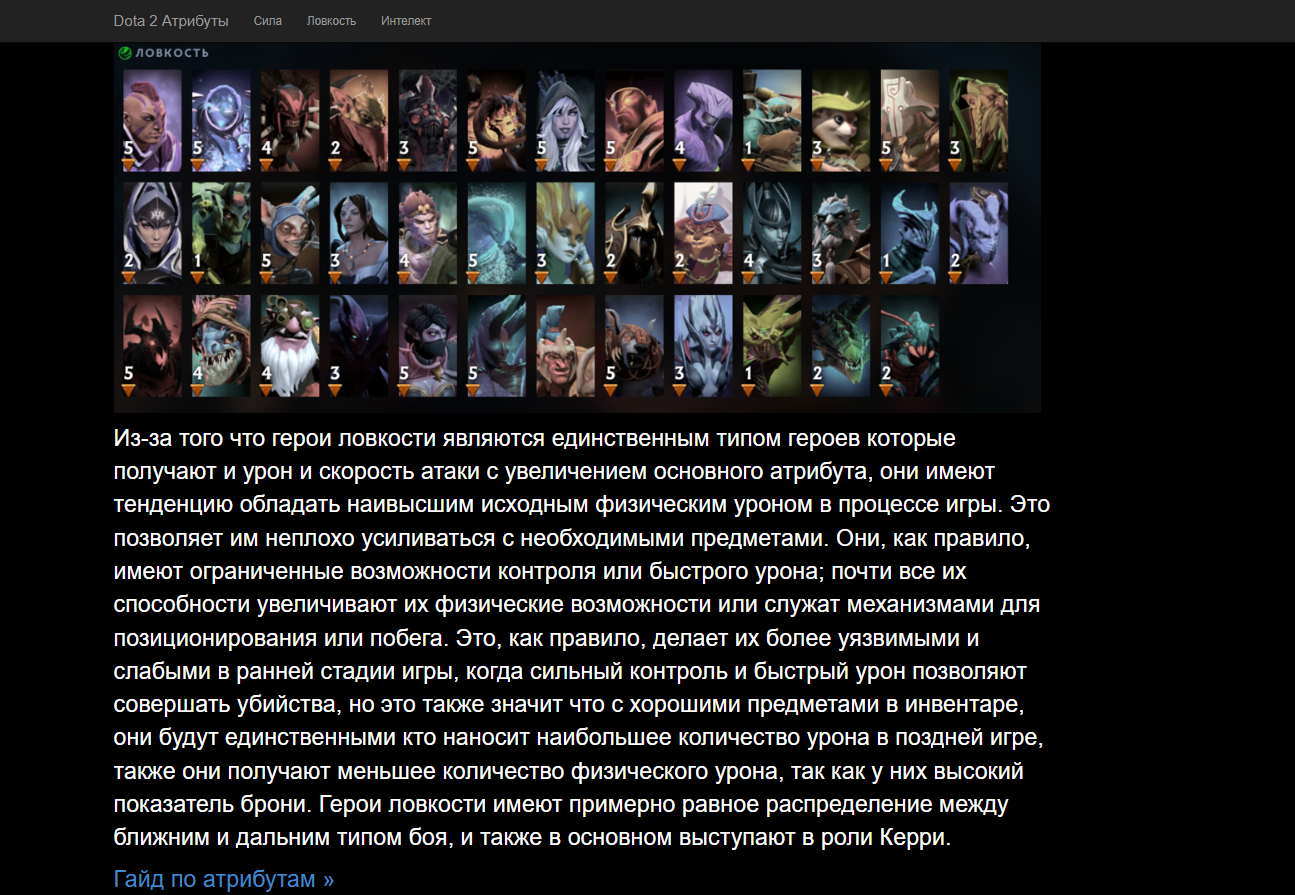


Страница Интеллект



Коммиты

Что я делал:



% rebase('layout.tpl', year=year)

<div class="Sila">

<img src="static\images\lovkost\_heroes.png">

<p></p>

<p class="siloviki">Из-за того что герои ловкости являются единственным типом героев которые получают и урон и скорость атаки с увеличением основного атрибута,

они имеют тенденцию обладать наивысшим исходным физическим уроном в процессе игры. Это позволяет им неплохо усиливаться с необходимыми предметами.

Они, как правило, имеют ограниченные возможности контроля или быстрого урона; почти все их способности увеличивают их физические возможности

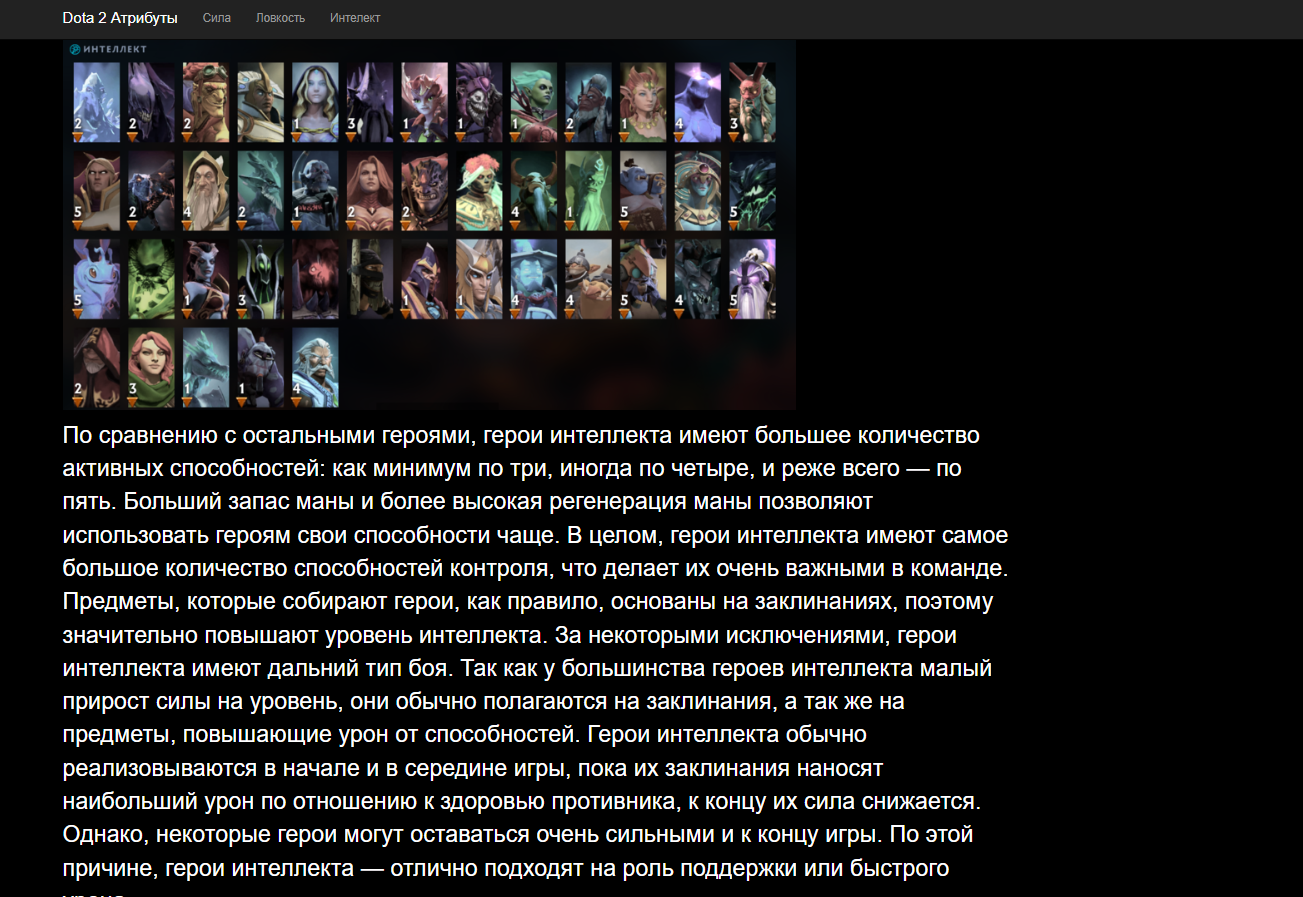
или служат механизмами для позиционирования или побега. Это, как правило, делает их более уязвимыми и слабыми в ранней стадии игры, когда сильный контроль

и быстрый урон позволяют совершать убийства, но это также значит что с хорошими предметами в инвентаре, они будут единственными кто наносит наибольшее количество урона

в поздней игре, также они получают меньшее количество физического урона, так как у них высокий показатель брони. Герои ловкости имеют примерно равное распределение

между ближним и дальним типом боя, и также в основном выступают в роли Керри.</p>

<p><a href="https://www.youtube.com/watch?v=i3YEstGdV9Y&ab\_channel=XENO" class="button\_video">Гайд по атрибутам &raquo;</a></p>

</div>

* % rebase('layout.tpl', year=year)
* <div class="Sila">
* <img src="static\images\intelect\_heroes.png">
* <p></p>
* <p class="siloviki">По сравнению с остальными героями, герои интеллекта имеют большее количество активных способностей: как минимум по три, иногда по четыре,
* и реже всего — по пять. Больший запас маны и более высокая регенерация маны позволяют использовать героям свои способности чаще. В целом, герои интеллекта
* имеют самое большое количество способностей контроля, что делает их очень важными в команде. Предметы, которые собирают герои, как правило, основаны на заклинаниях,
* поэтому значительно повышают уровень интеллекта. За некоторыми исключениями, герои интеллекта имеют дальний тип боя. Так как у большинства героев интеллекта
* малый прирост силы на уровень, они обычно полагаются на заклинания, а так же на предметы, повышающие урон от способностей. Герои интеллекта обычно реализовываются
* в начале и в середине игры, пока их заклинания наносят наибольший урон по отношению к здоровью противника, к концу их сила снижается. Однако, некоторые герои могут
* оставаться очень сильными и к концу игры. По этой причине, герои интеллекта — отлично подходят на роль поддержки или быстрого урона.</p>
* <p><a href="https://www.youtube.com/watch?v=i3YEstGdV9Y&ab\_channel=XENO" class="button\_video">Гайд по атрибутам &raquo;</a></p>
* </div>
* Контрольные вопросы:  
    
  1. Каково основное назначение UML-диаграмм и их роль в разработке приложений?  
  Основное назначение UML-диаграмм - визуализация и описание различных аспектов системы, которую нужно разработать или модифицировать. UML-диаграммы помогают увидеть общую картину и понять структуру системы, ее функциональность, взаимодействие между компонентами и объектами, а также общий процесс взаимодействия системы с внешним миром. UML-диаграммы являются важным инструментом для коммуникации между разработчиками и заинтересованными сторонами проекта.  
    
  2. Можно ли считать артефактами веб-страницы сайта? Обоснуйте ответ.  
  Да, веб-страницы сайта можно считать артефактами. Веб-страницы являются результатом работы веб-разработчика, и они представляют собой конечный продукт разработки. Кроме того, веб-страницы могут быть использованы в качестве артефактов для оценки качества разработки, например, при тестировании.  
    
  3. Предпочтительнее ли протокол HTTPS протоколу HTTP? Почему (да / нет)?  
  Да, предпочтительнее использовать протокол HTTPS вместо HTTP. Это связано с тем, что HTTPS обеспечивает более высокий уровень защиты данных, потому что он использует шифрование для защиты данных, передаваемых между сервером и клиентом. HTTPS также обеспечивает подтверждение подлинности сервера, что предотвращает атаки типа "man-in-the-middle".  
    
  4. Что значит код ответа сервера «200»? Какие ещё группы и коды ответов вы знаете (привести по 1-2 примера из каждой группы)?  
  Код ответа сервера "200" означает успешный ответ на запрос клиента. Еще группы и коды ответов, которые могут быть получены в ответ на запрос клиента, включают:  
  Коды ответов 300-399 - перенаправление, например, "301 Moved Permanently" или "302 Found".  
  Коды ответов 400-499 - ошибка на стороне клиента, например, "404 Not Found" или "403 Forbidden".  
  Коды ответов 500-599 - ошибка на стороне сервера, например, "500 Internal Server Error" или "503 Service Unavailable".  
    
  5. Какую систему защиты и сертификации данных использует Git по умолчанию?  
  Git использует систему защиты и сертификации данных под названием GPG (GNU Privacy Guard) по умолчанию. GPG используется для подписи и проверки целостности коммитов в репозитории, что помогает предотвратить внесение нежелательных изменений в репозиторий.  
    
  6. Что значит «клонировать» репозиторий в терминах Git?  
  "Клонирование" репозитория в терминах Git означает создание локальной копии удаленного репозитория Git на вашем компьютере. Клонирование позволяет получить полную копию удаленного репозитория на локальном устройстве, включая все истории коммитов, ветки и теги.

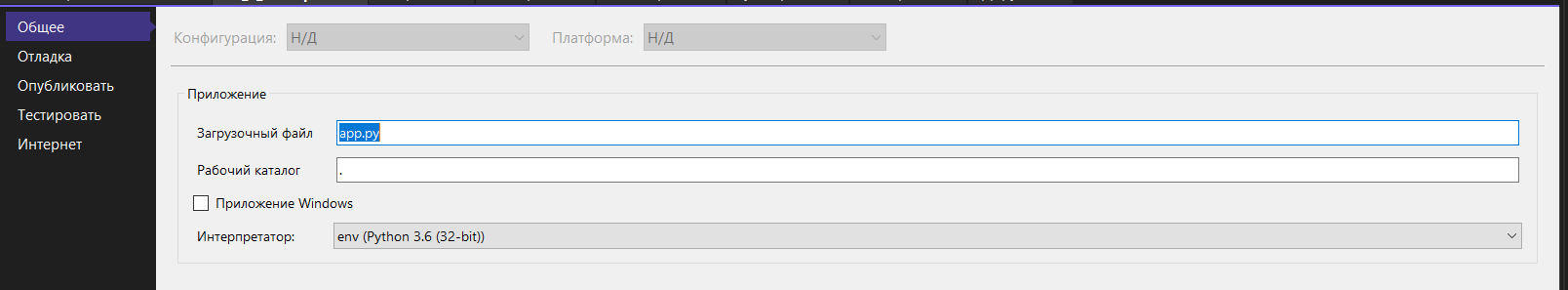
**Лабораторная работа №5, 6**

**Разработка и интеграция модулей проекта. Отладка отдельных модулей ПП.**

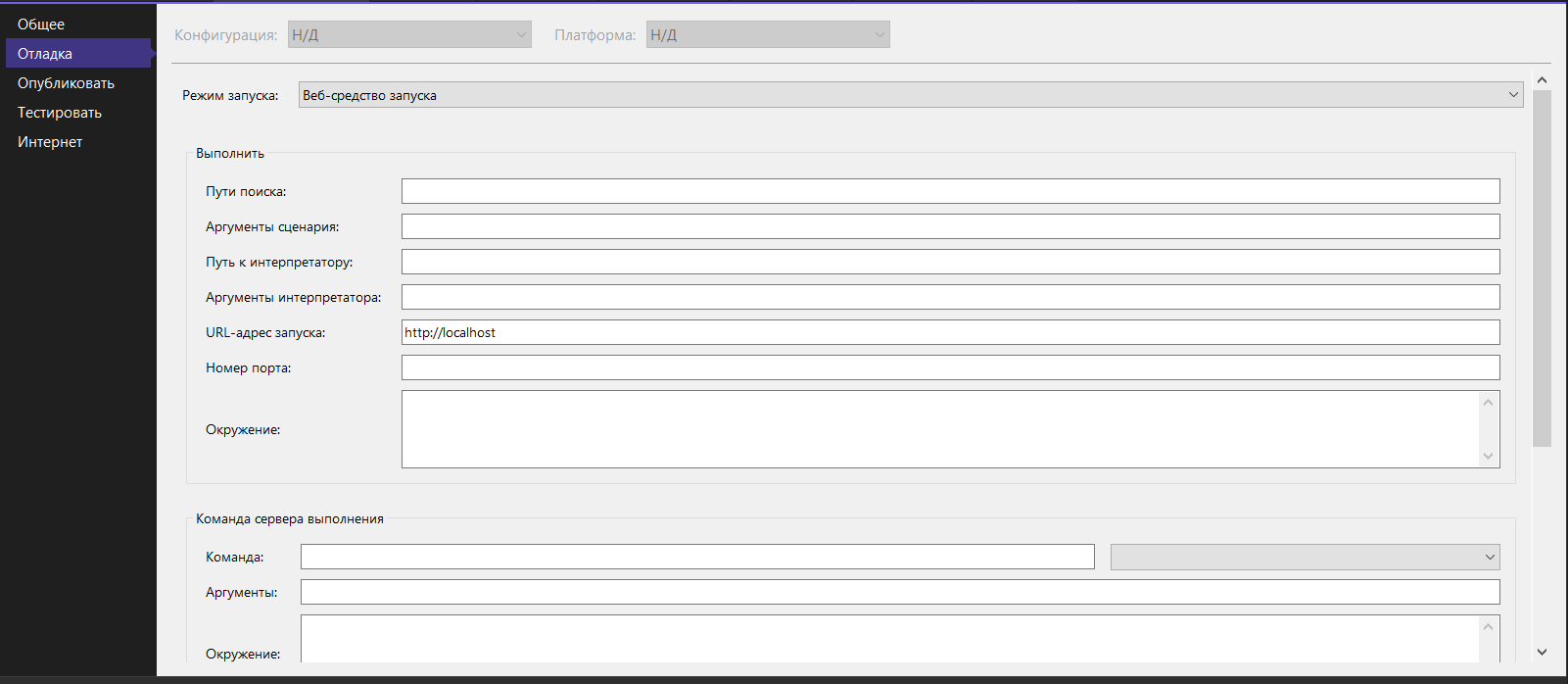
**Организация обработки исключений.**

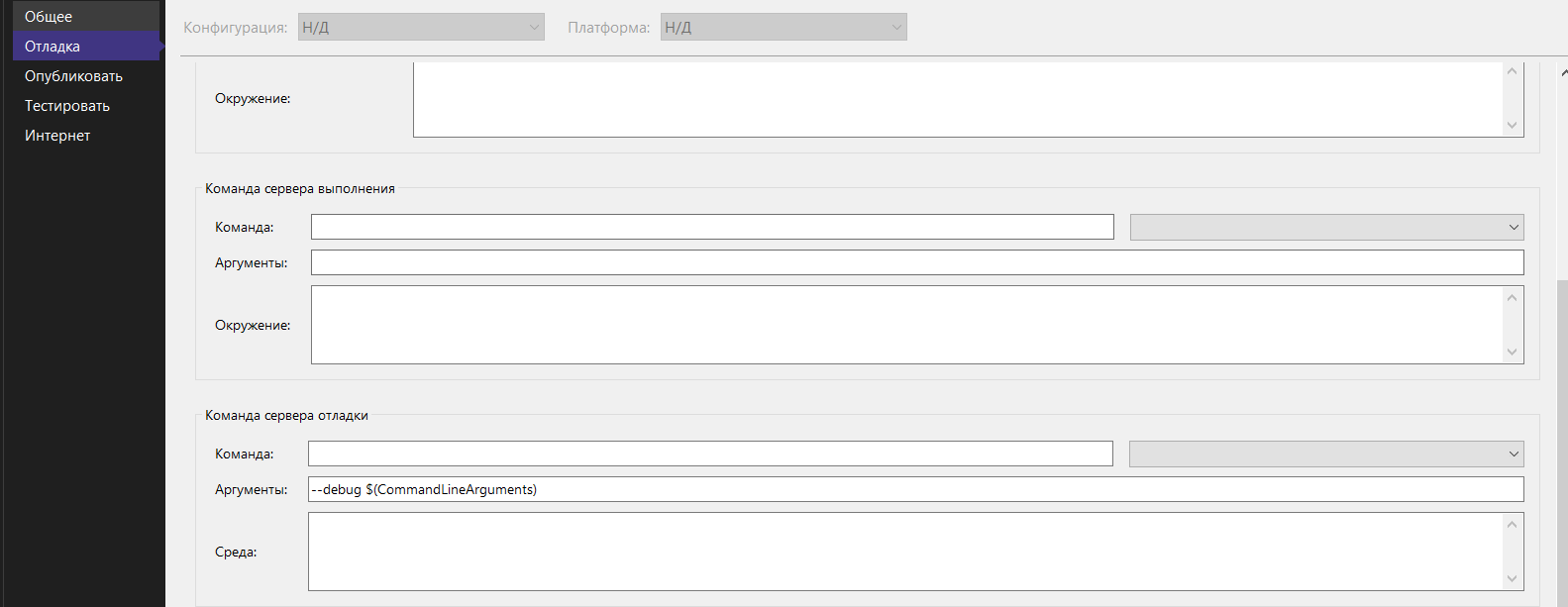
Цель: создание дополнительного модуля проекта, его интеграция в систему, выполнение отладки и обработки исключительных ситуаций.

1. В открывшемся окне свойств отладки с помощью всплывающих подсказок изучить описания параметров и те их значения, которые заданы по умолчанию. Результаты внести в отчёт.



общее



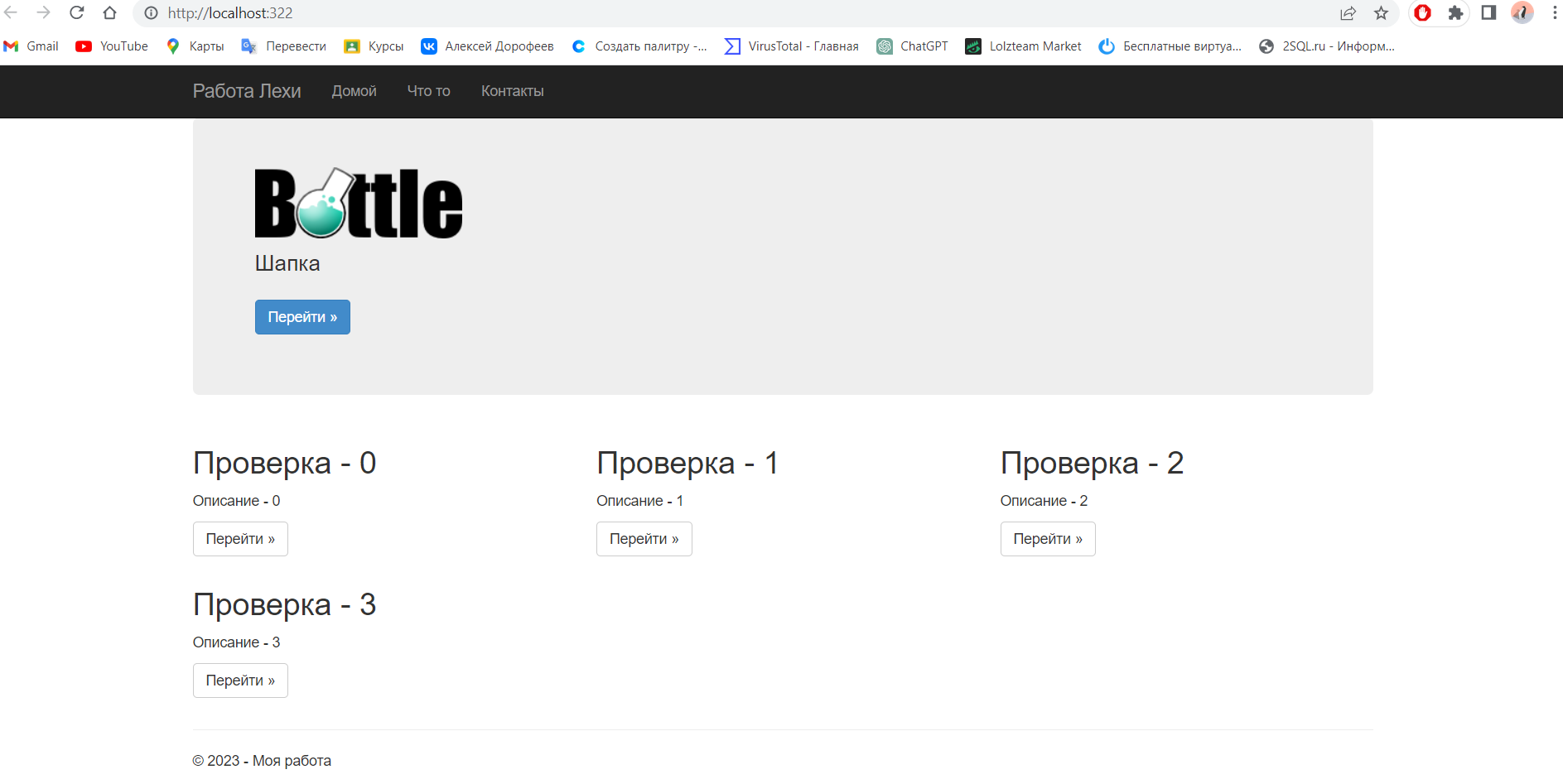


Откладка

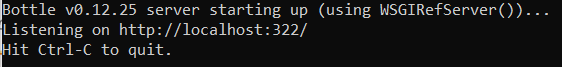
1. Установить фиксированный номер порта, например, 322:



Установка порта



Порт на сайте





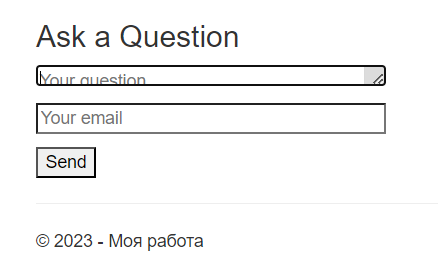
Удалили порт



Порт на сайте после удаления

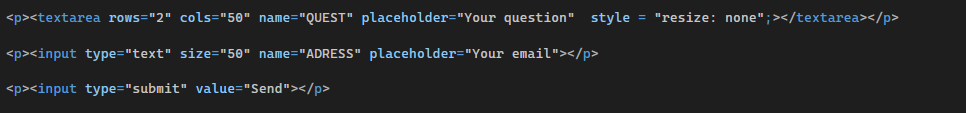


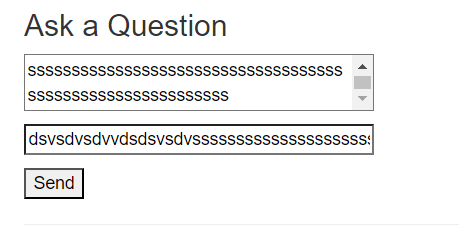
1. Добавить на главную страницу сайта простейшую форму обратной связи, состоящую из текстовой области, текстового поля и кнопки.



Выяснить, что означают атрибуты HTML-тэгов, сделать размер textarea неизменяемым.

Style = “resize: none”;



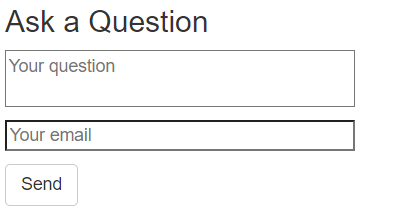


После фиксации

Т.к. кнопка отличается по стилю, то следует в соответствующий её тэг добавить

class="btn btn-default". Обновить страницу, проверить результат.





После изменения кнопки

1. Теперь необходимо добавить в папку проекта файл-обработчик для формы. Для этого сначала остановить отладку, затем добавить новый объект, выполнив Проект > Добавить … (Add New Item…). В открывшемся списке выбрать Пустой файл Python, ввести имя файла myform.py

Код в файле myfrom.py:

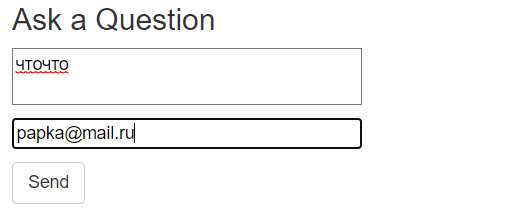
from bottle import post, request

@post('/home', method='post')

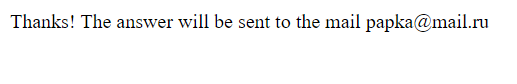
def my\_form():

mail = request.forms.get('ADRESS')

return "Thanks! The answer will be sent to the mail %s" % mail



Заполнение полей



Результат после нажатия на кнопку