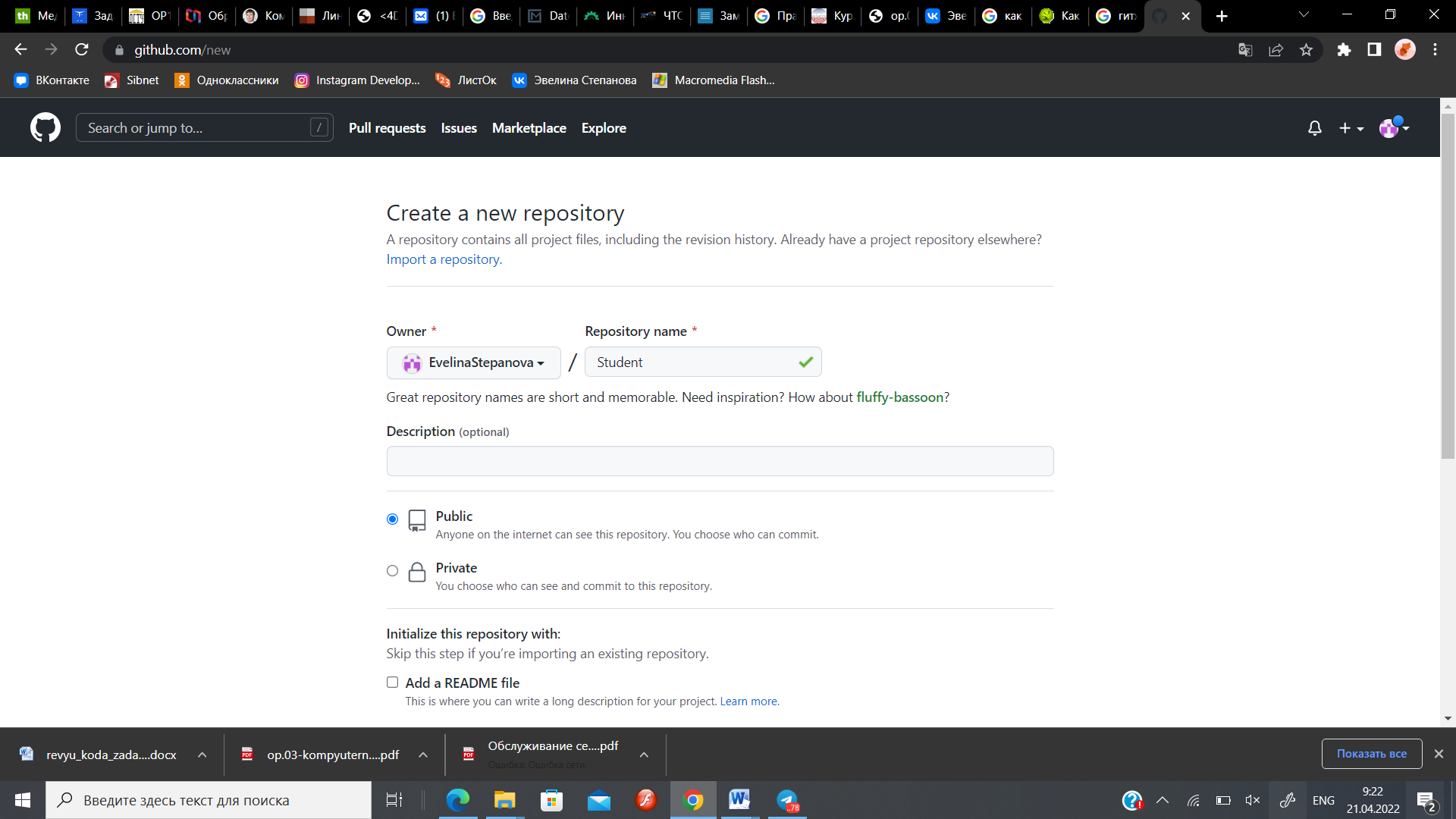
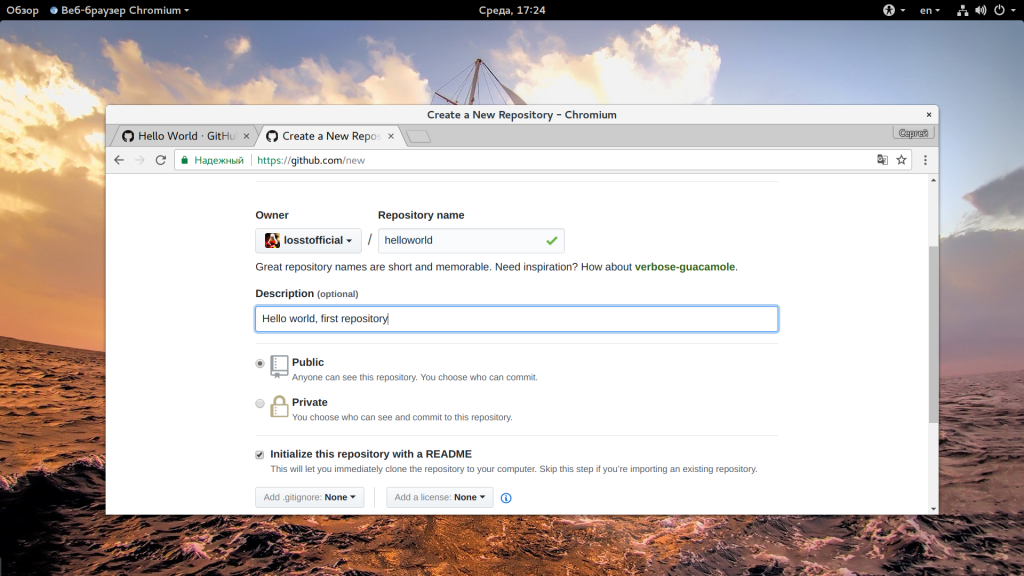
1. Зарегистрируйтесь на github.com
2. Создание репозитория

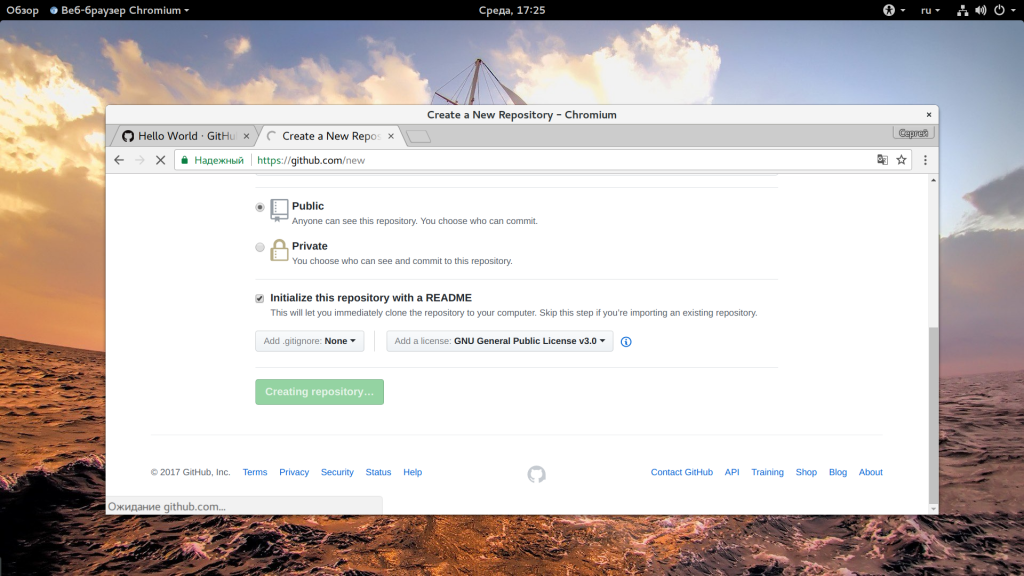
На открывшейся странице, это главная страница для авторизованных пользователей, нажмите кнопку "Start a project". Дальше введите имя и описание будущего репозитория:

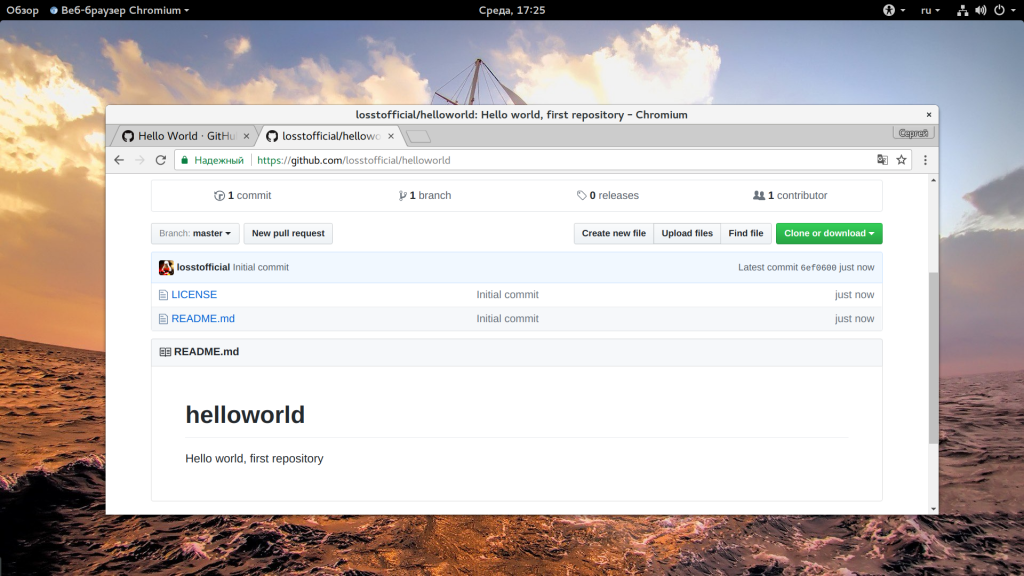


Вы можете сразу же инициализировать репозиторий, создав файл Readme, для этого нужно отметить галочку **"Initialize this repository with a README"** внизу страницы. Также можно выбрать лицензию:



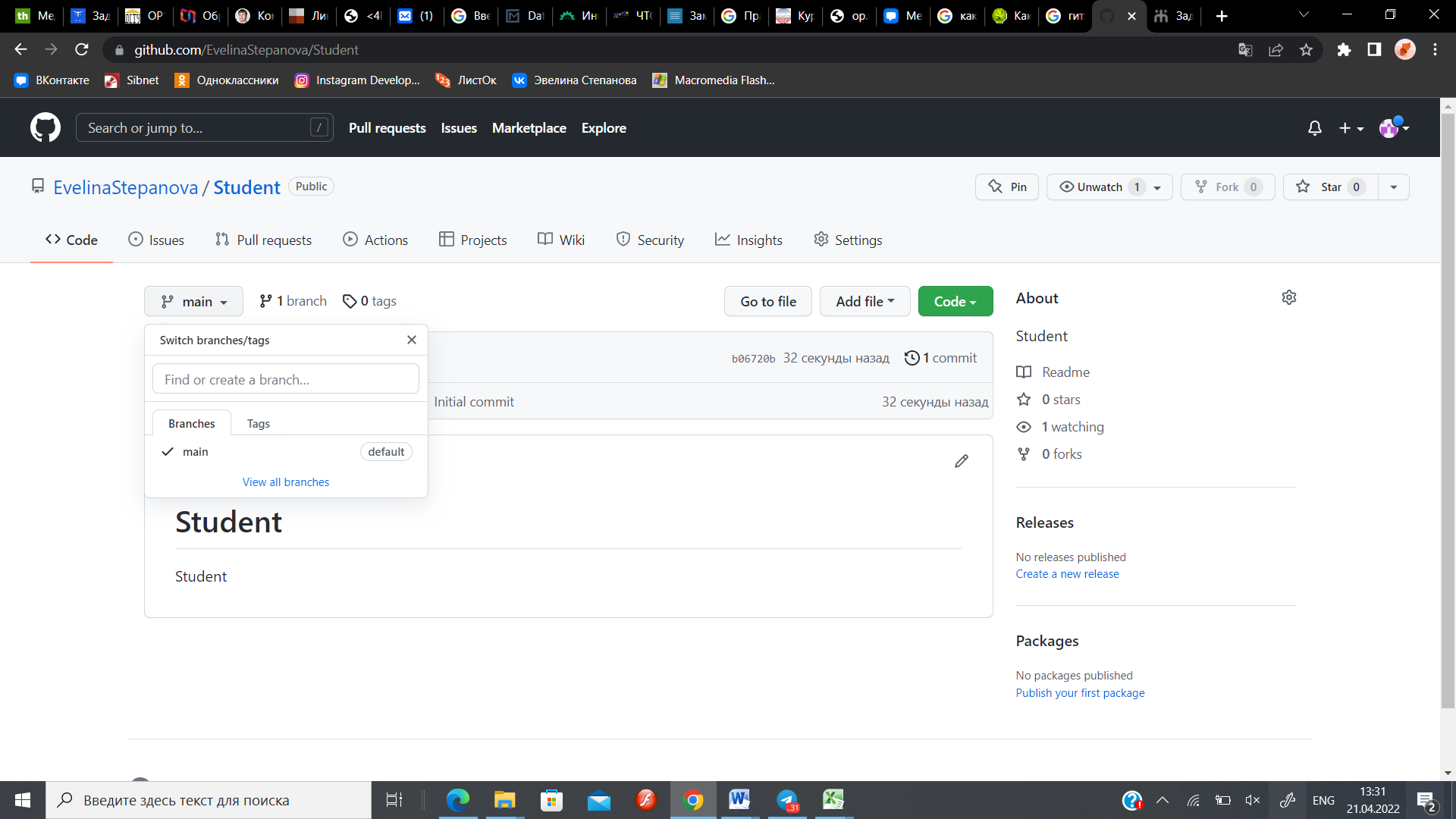
Когда все будет готово, выберите **"Create project"**, будет создан новый проект с файлом README, в котором находится описание и файлом лицензии.



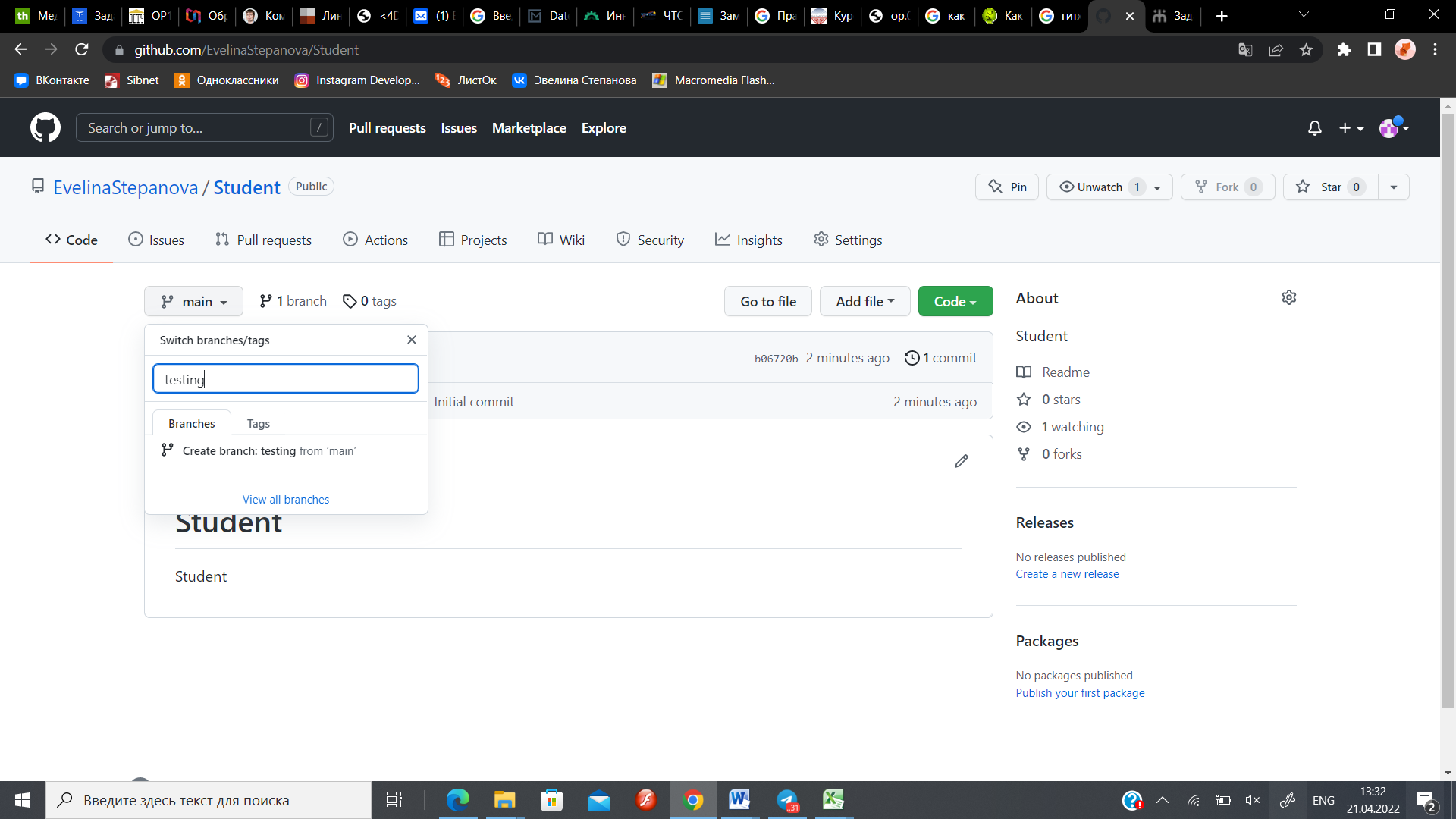


1. Добавление веток

Ветки Github позволяют работать с несколькими версиями проекта одновременно. По умолчанию при создании репозитория создается ветка master, это основная рабочая ветка. Можно создать дополнительные ветки, например, для того, чтобы тестировать программное обеспечение перед тем, как оно будет опубликовано в ветке master. Таким образом, можно одновременно разрабатывать продукт и предоставлять пользователям стабильную версию. Также можно создавать отдельные ветки для версии программы для разных систем.



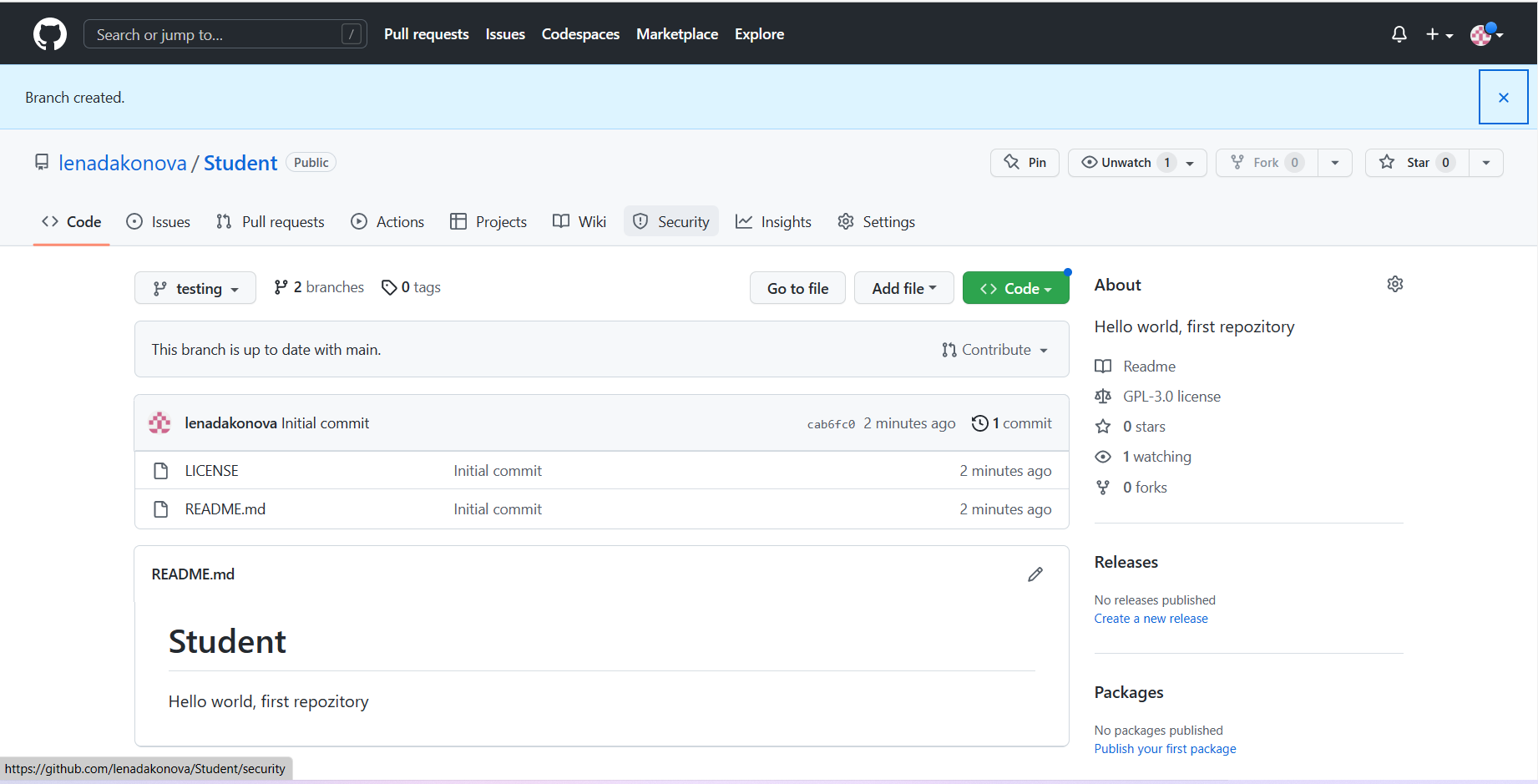
Текущая ветка обозначена в верхнем левом углу после слова "Branch". Чтобы создать новую ветку просто разверните этот список и начните набирать ее имя:



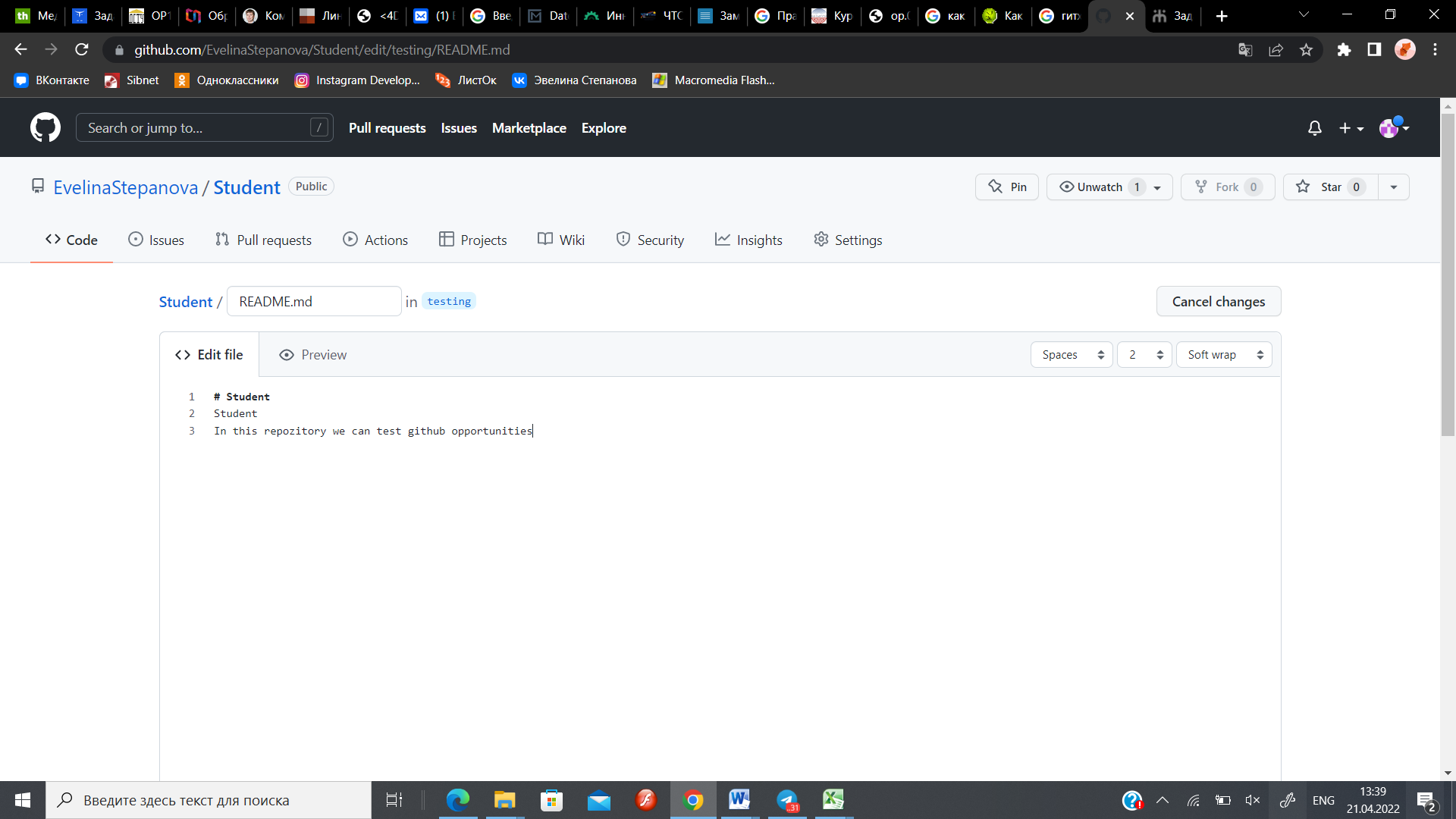
Сайт сам предложит вам создать новую ветку, выберите "Create branch".

Сразу же после создания вы будете работать с только что созданной веткой.

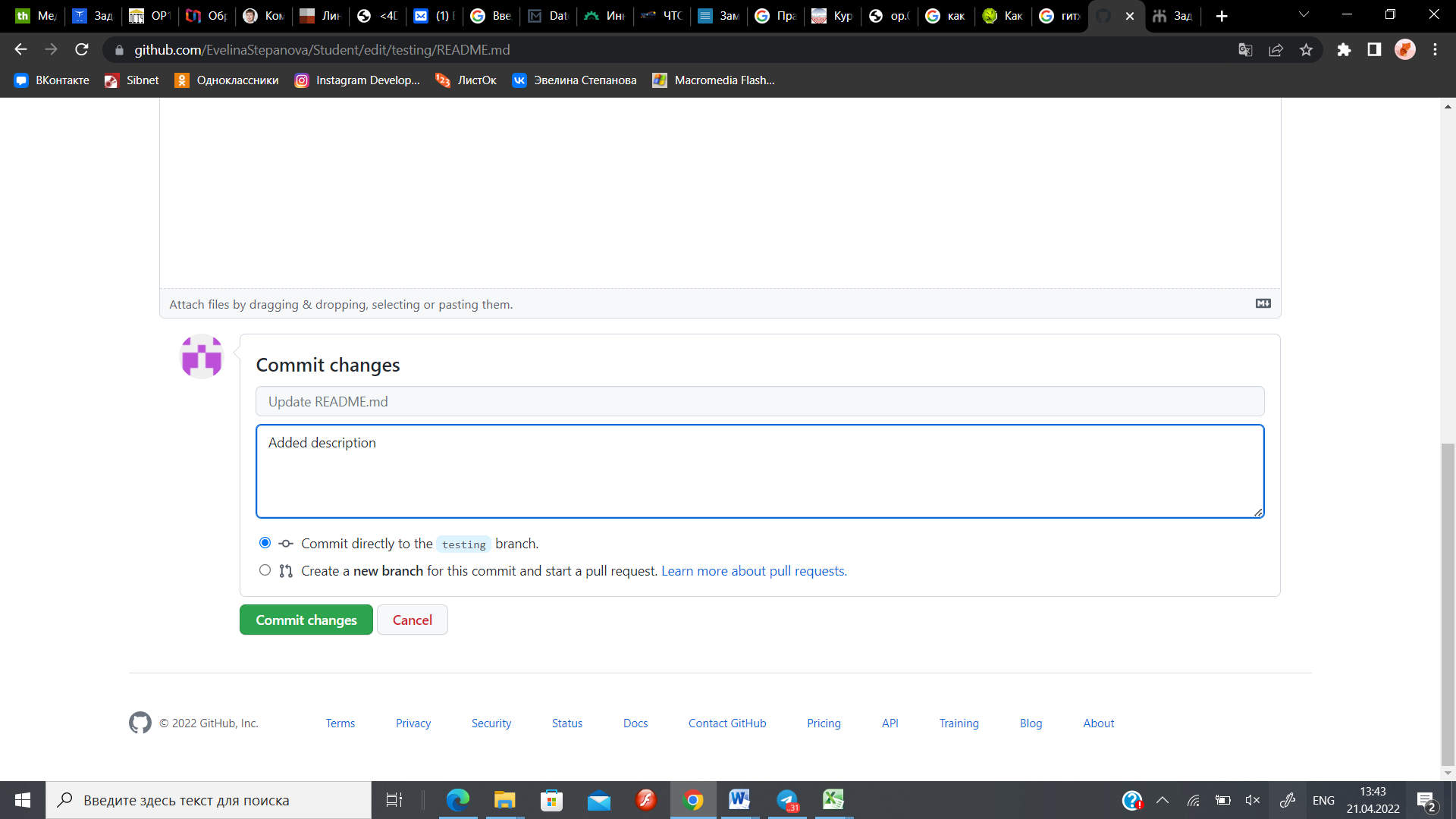
Изменение файлов и коммиты

Любые изменения файлов на Github делаются с помощью коммитов. Коммит выполняется путем внесения самих исправлений и описания этих исправлений. Это необходимо для того, чтобы вы знали что и когда вы меняли, а также позволяет легко отслеживать работу команды. Слово коммит можно перевести как "фиксировать". То есть мы можем внести изменения в несколько файлов, а затем их зафиксировать. Давайте для примера изменим файл README. Для этого найдите в правой стороне панели кнопку с кисточкой и нажмите на нее: 

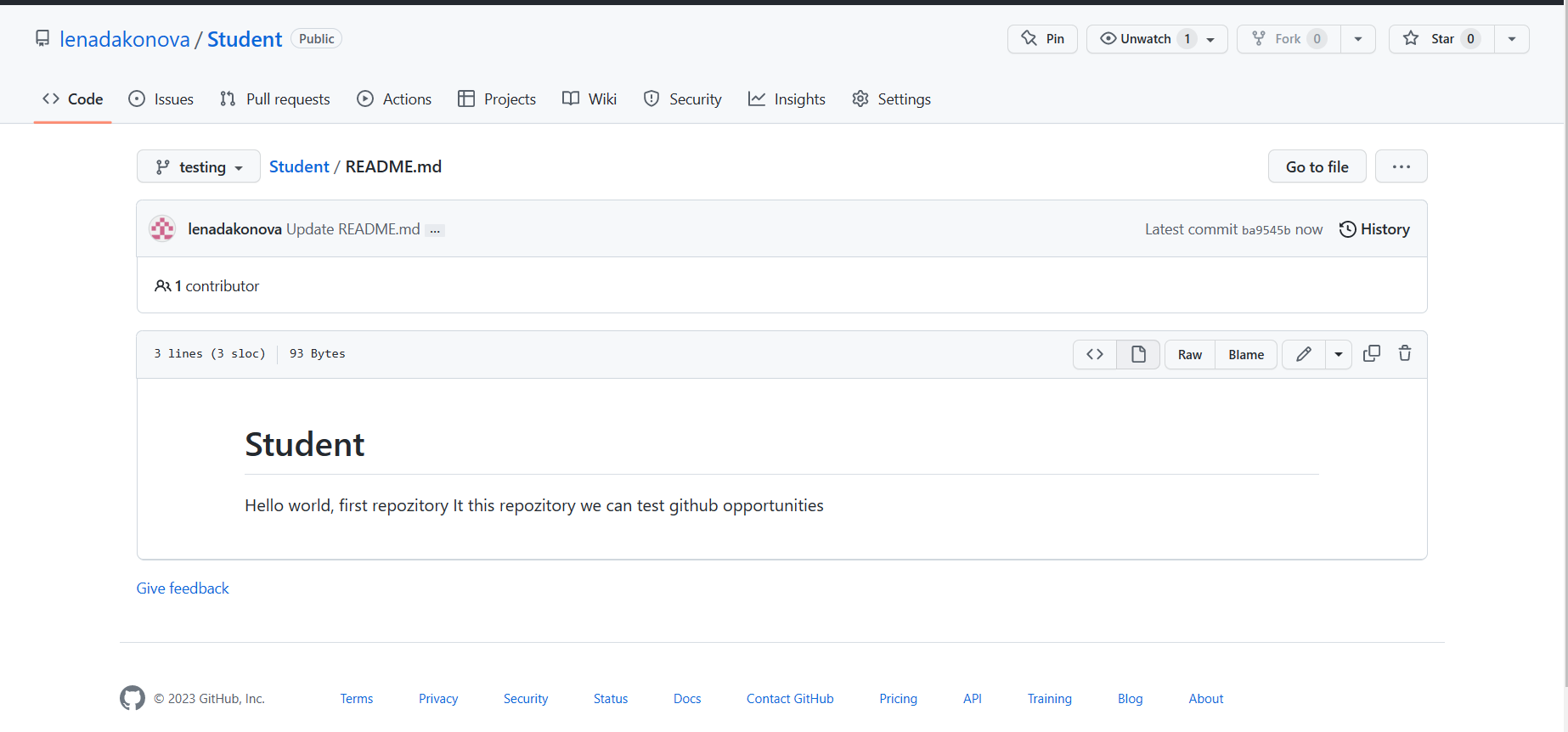
Откроется текстовый редактор, где вы можете ввести нужные вам исправления:



После того как вы сделаете все что вам нужно, необходимо заполнить поле **"Commit"** внизу страницы. Кратко опишите что было изменено, а затем нажмите кнопку **"Commit changes"**:



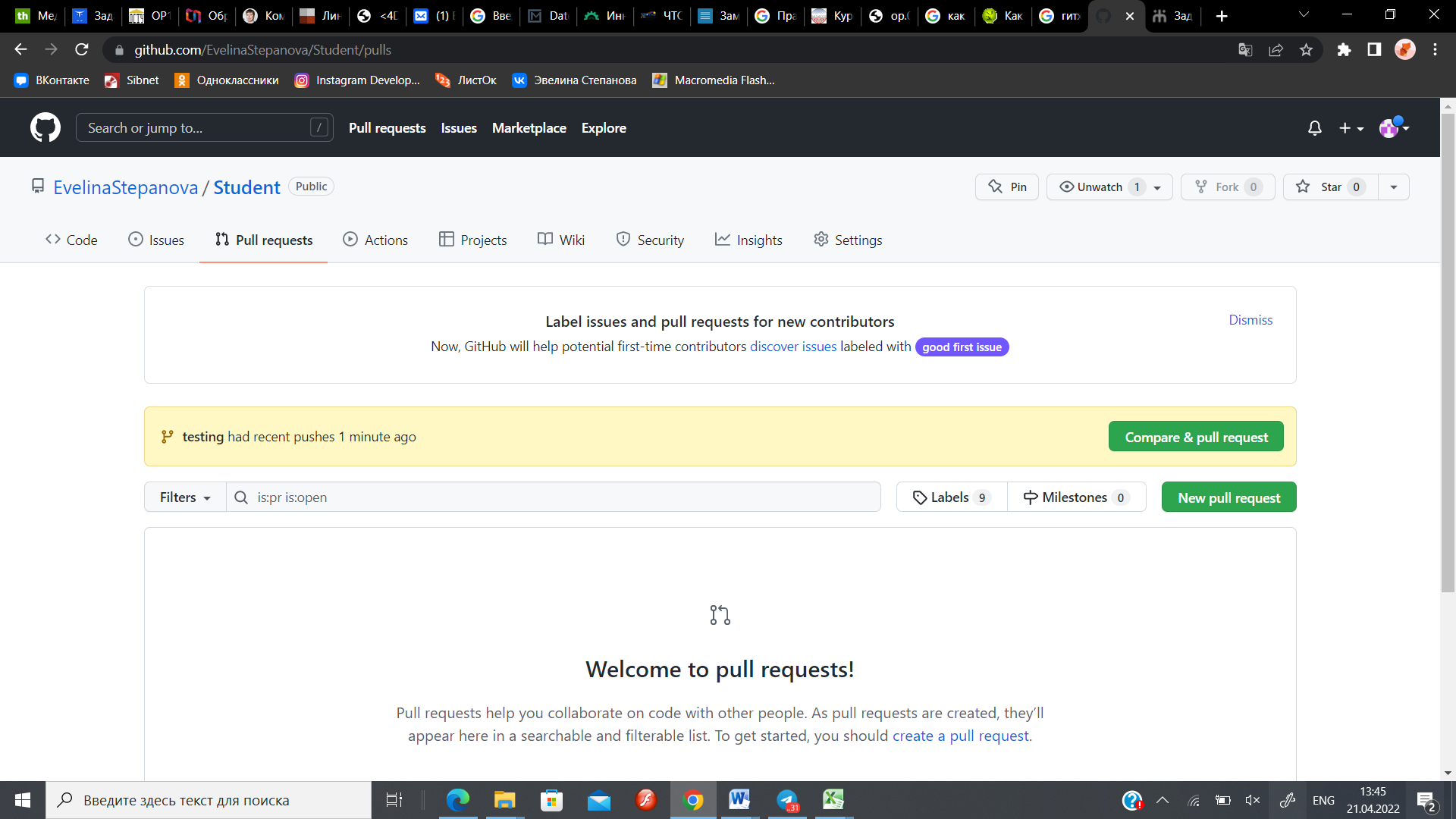
Эти изменения будут внесены в текущую ветку проекта, поскольку мы сейчас работаем с testing, то и изменения будут отправлены именно туда.



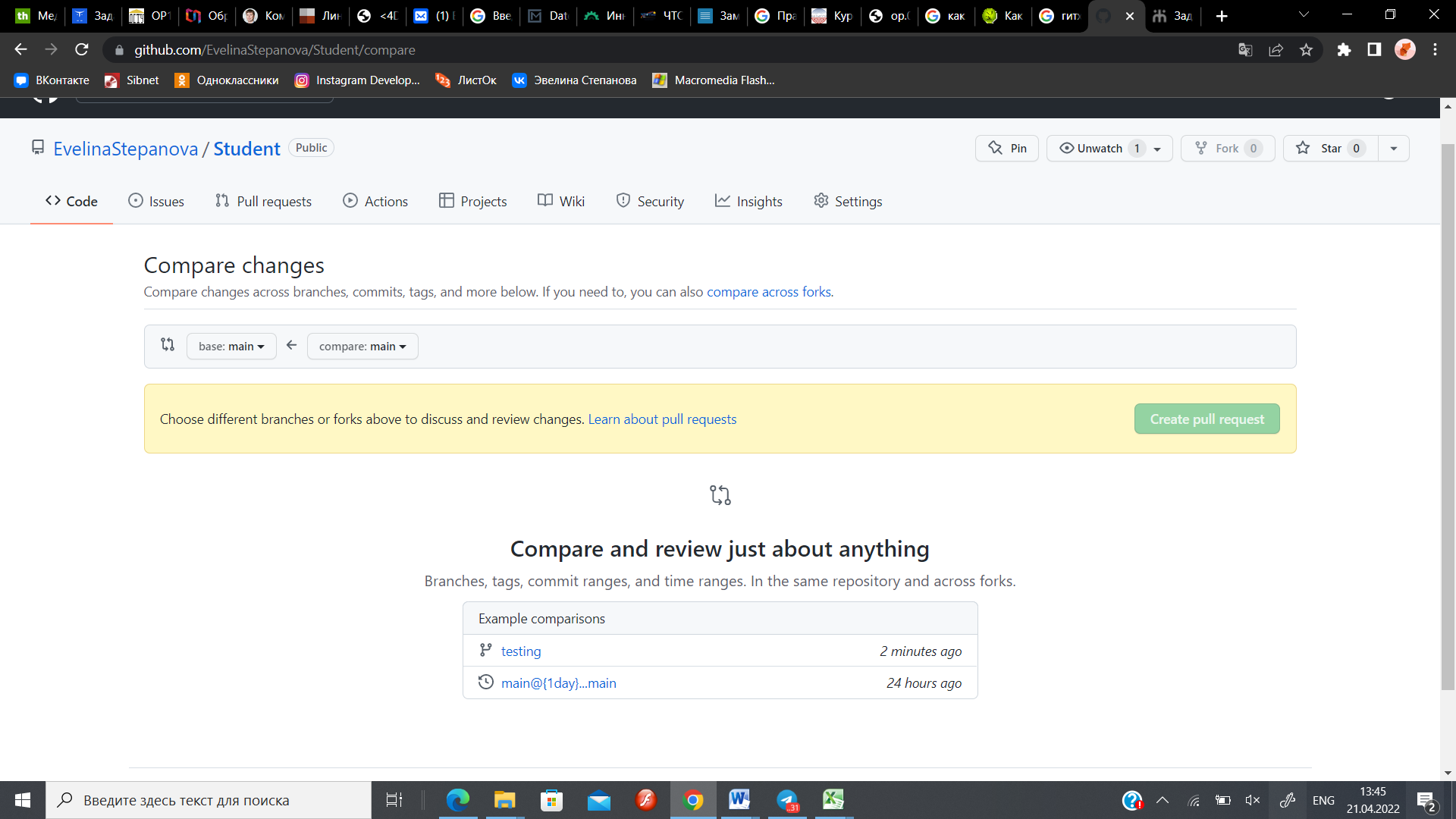
5. Создание запросов слияния (Pull Request)

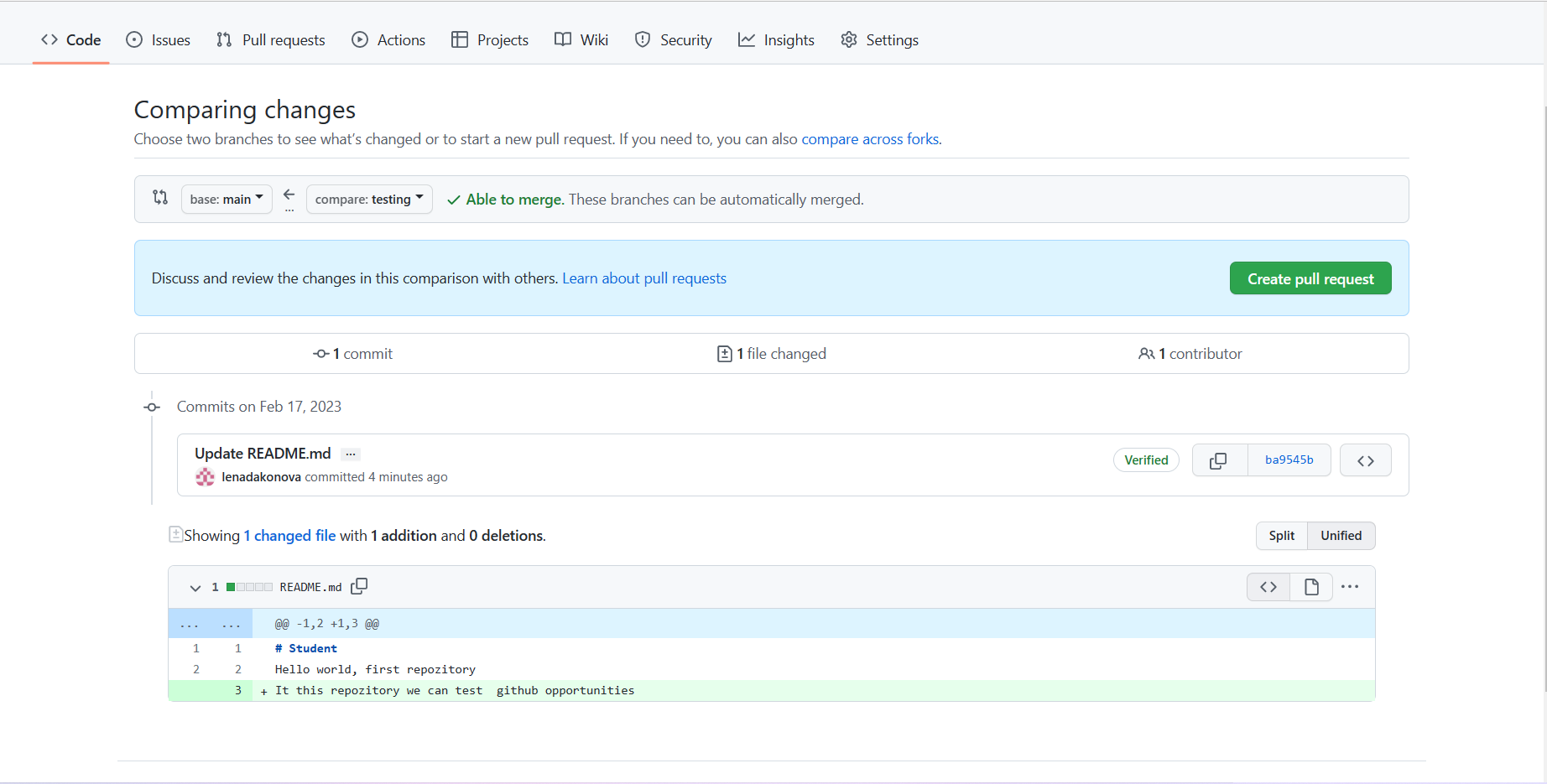
GitHub для начинающих может показаться очень сложным именно из-за таких возможностей, но это очень удобно если разобраться. Запрос слияния или Pull Request — это возможность, благодаря которой любой разработчик может попросить другого, например, создателя репозитория просмотреть его код и добавить его в основной проект или ветку. Инструмент работы с запросами слияния использует инструмент сравнения diff, поэтому вы можете увидеть все изменения, они будут подчеркнуты другим цветом. Pull Request можно создать сразу же после создания коммита. Давайте отправим Pull Request из нашей testing ветки в основную. Сначала откройте вкладку **"Pull Request".**

Здесь нажмите кнопку **"Create Pull Request"**:

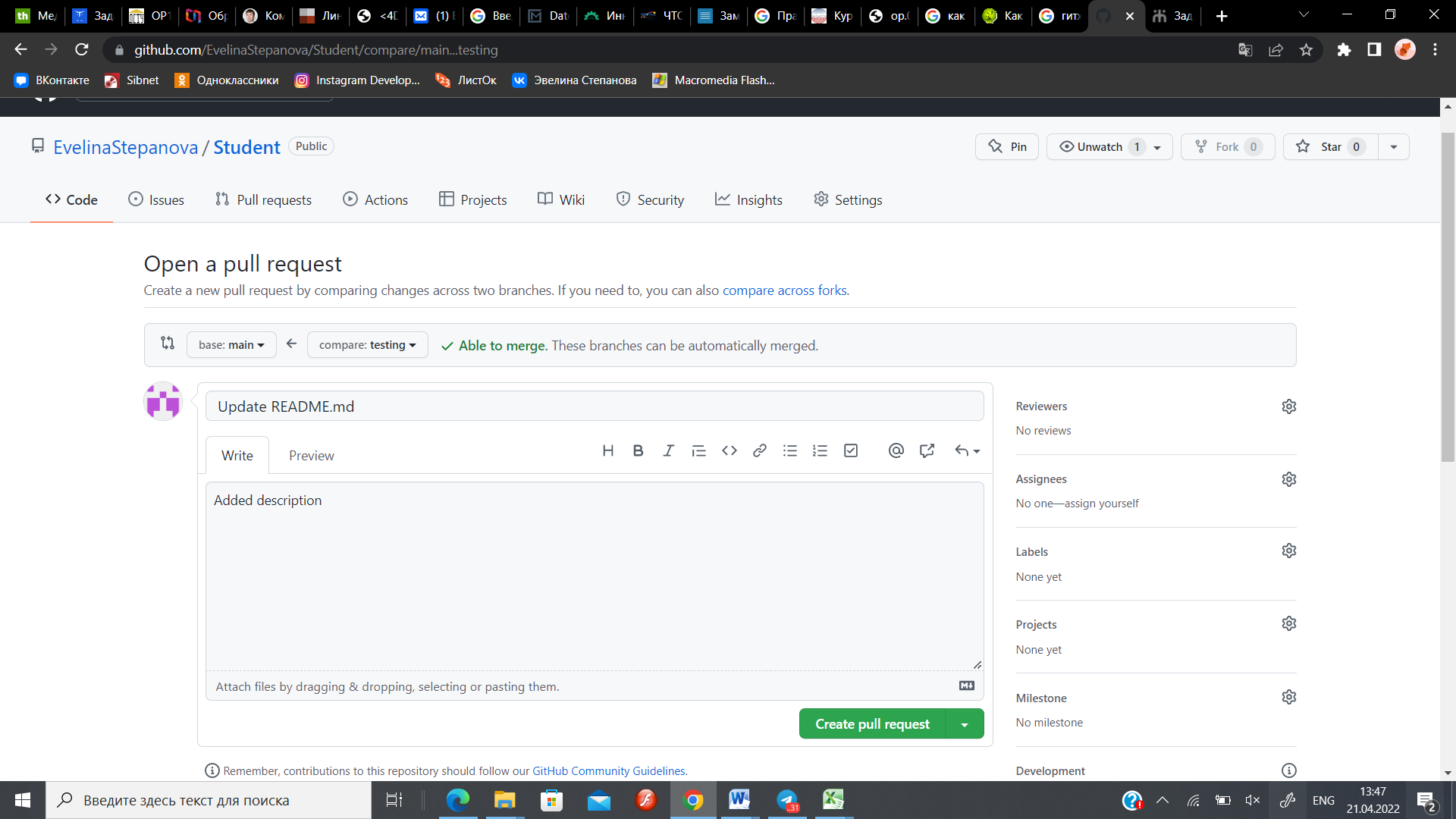


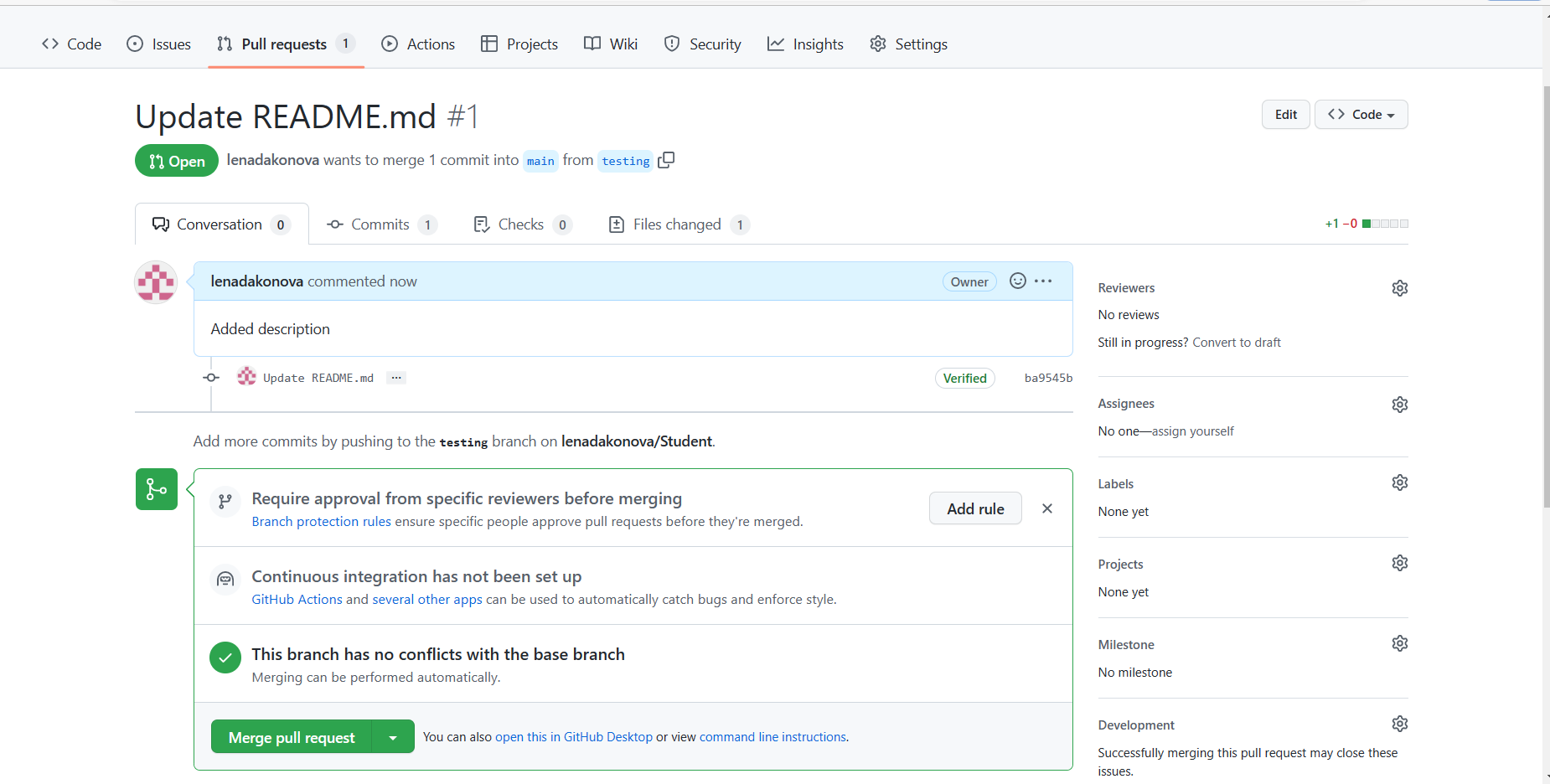
Дальше вам нужно будет выбрать ветку, которую нужно слить с основной, в нашем случае **"testing"**.



В этом окне вы можете просмотреть все изменения, сейчас мы видим, что была добавлена строчка: 

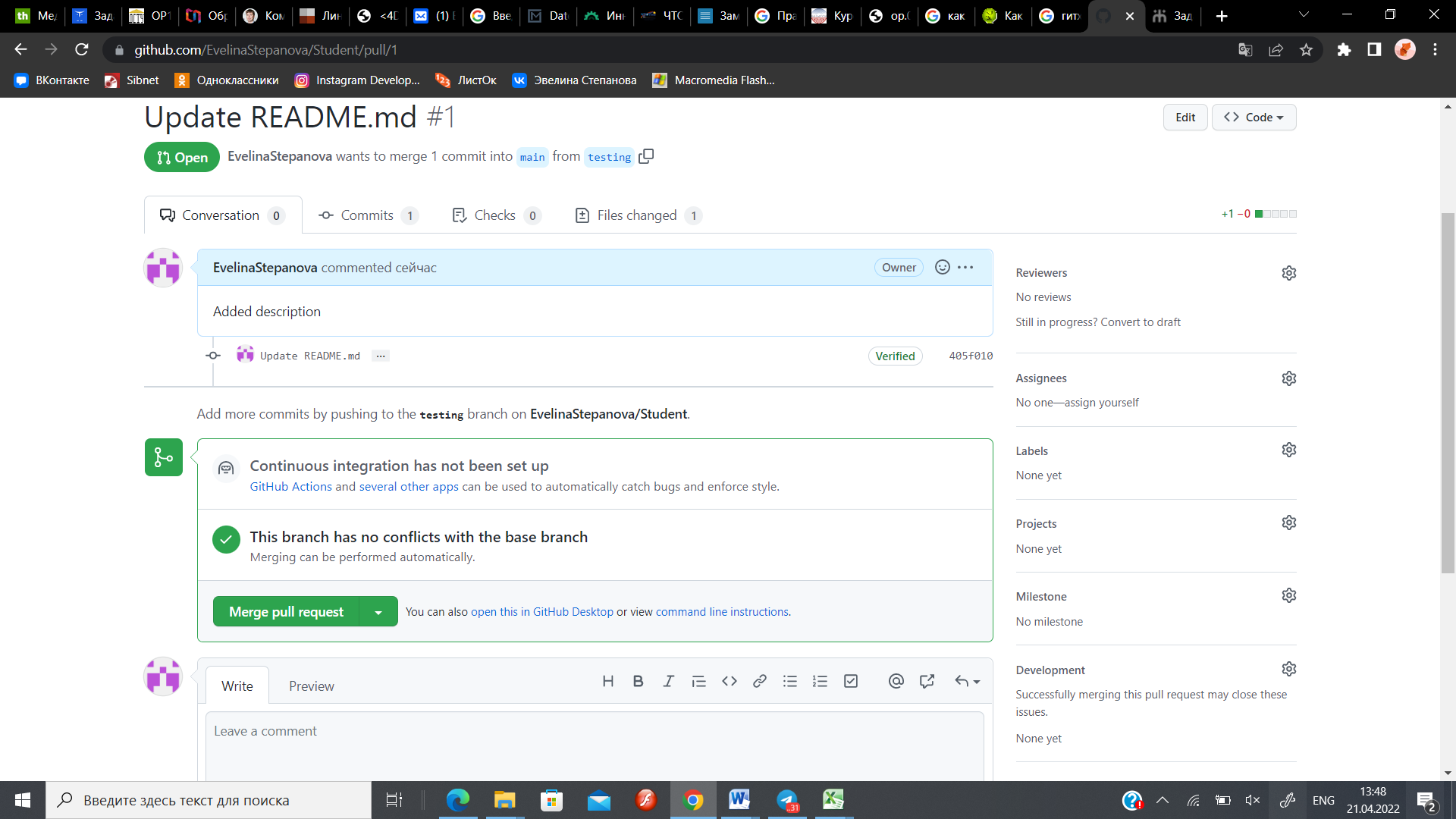
Дальше нажмите зеленую кнопку **"Create Pull Request"** и введите описание, как и для коммита:





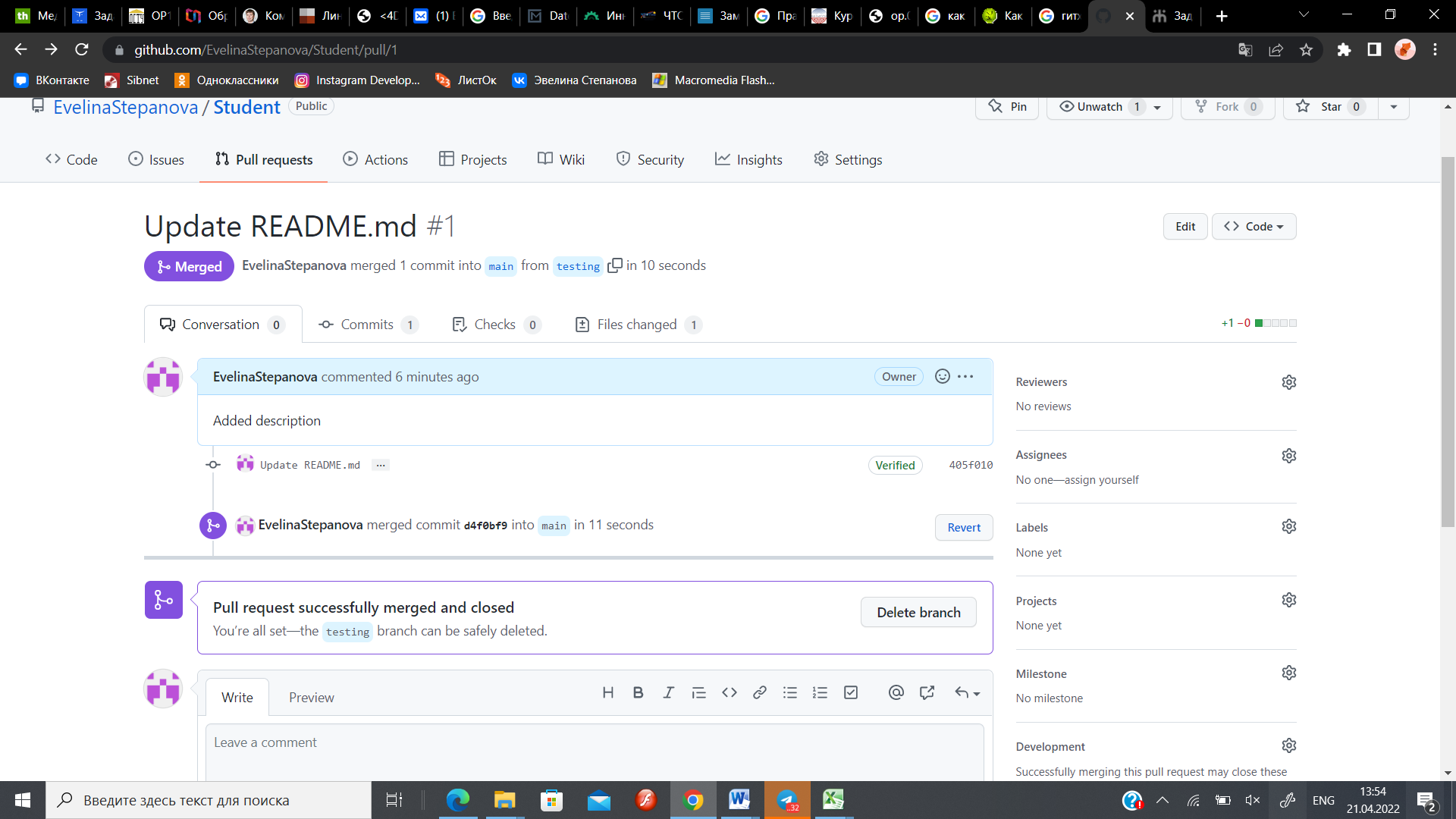
6. Просмотр и одобрение запросов на слияние

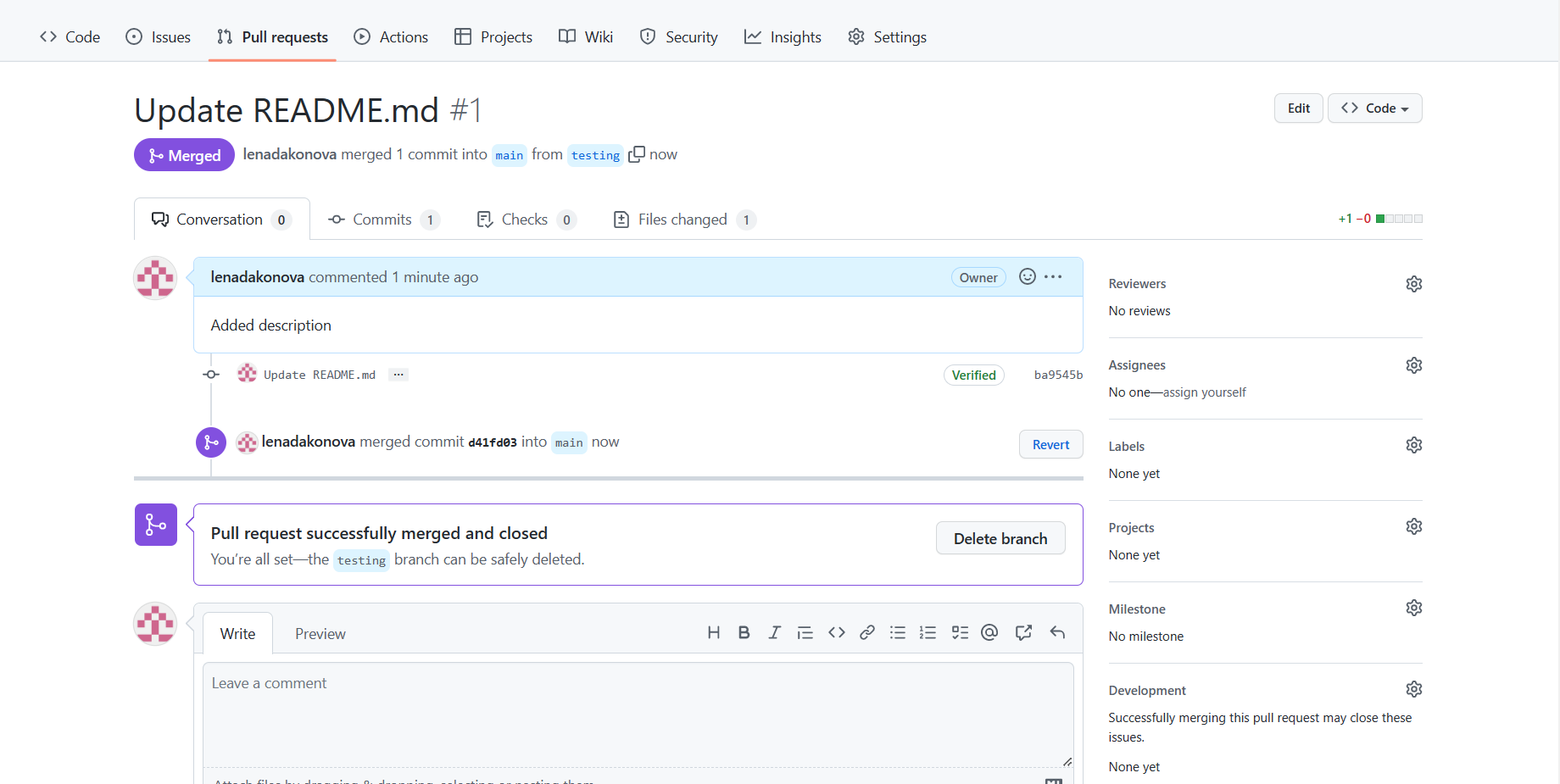
Теперь, на той же вкладке Pull Requests мы видим только что созданный запрос на слияние и нам остается только принять его нажав **"Merge Pull Request"**:



Но если этот запрос пришел от другого человека, вы должны проверить что он там изменил и нужно ли это, для этого просто нажмите на описание запроса и увидите уже знакомое окно просмотра изменений:

Дальше можно подтвердить Pull Request:

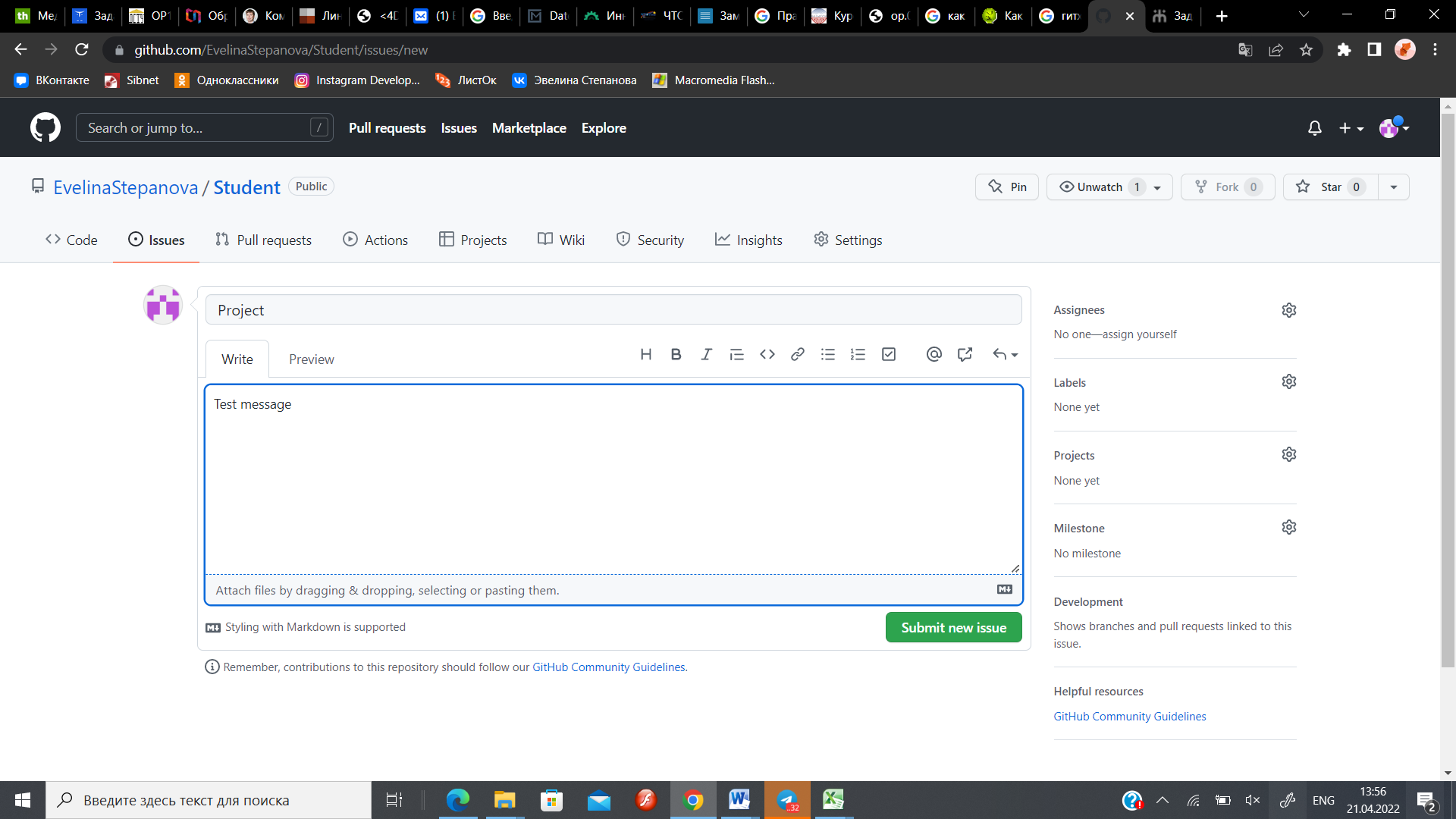




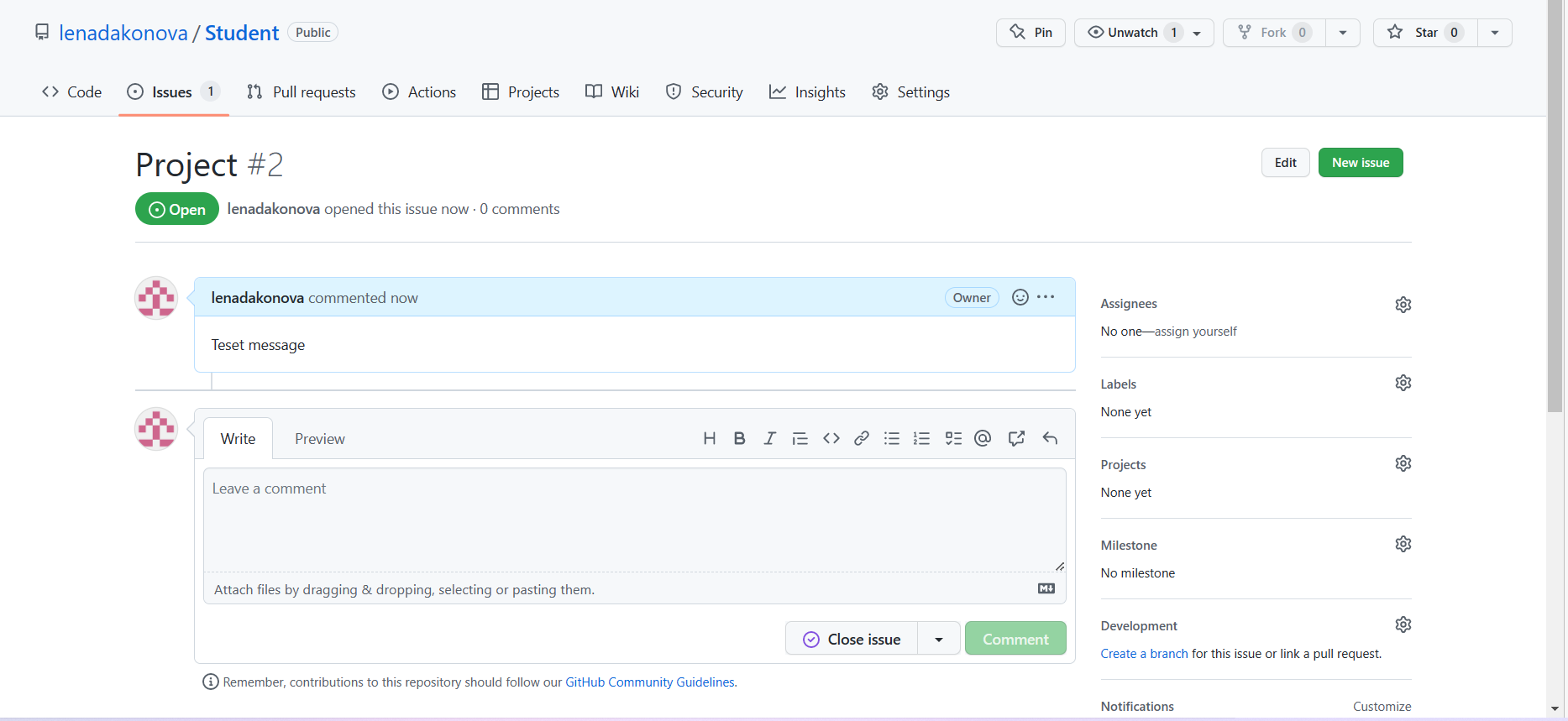
Затем код будет импортирован в основную ветку, а ветка testing может быть безопасно удалена.

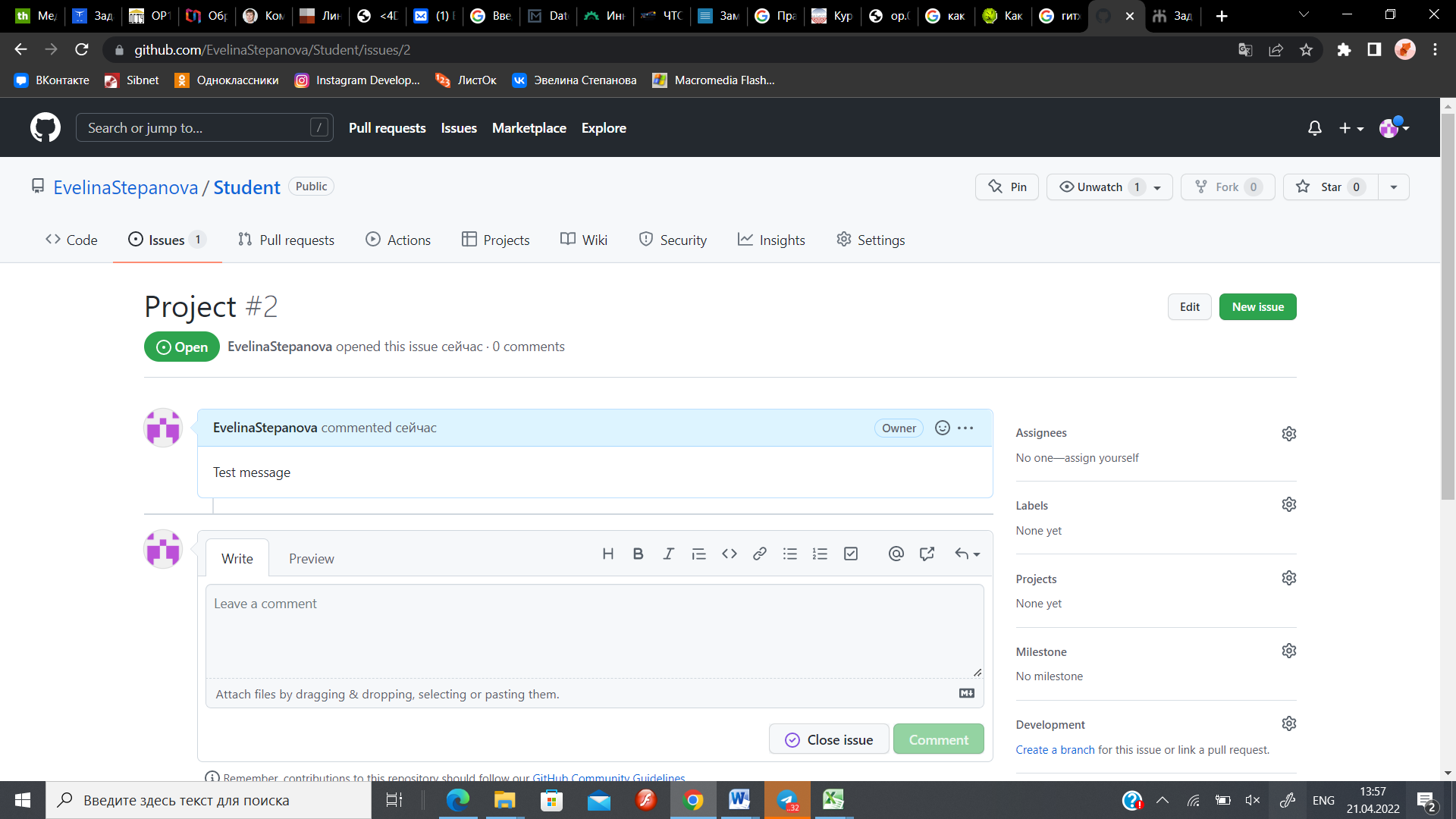
7. Отчеты об ошибках

Удобно еще то, что возможно использование GitHub не только для разработки и управления кодом, но и для обратной связи с пользователями. На вкладке **"Issue"** пользователи могут оставлять сообщения о проблемах, с которыми они столкнулись при использовании вашего продукта. Откройте вкладку **"Issues"**, и нажмите на кнопку **"New issue"**:



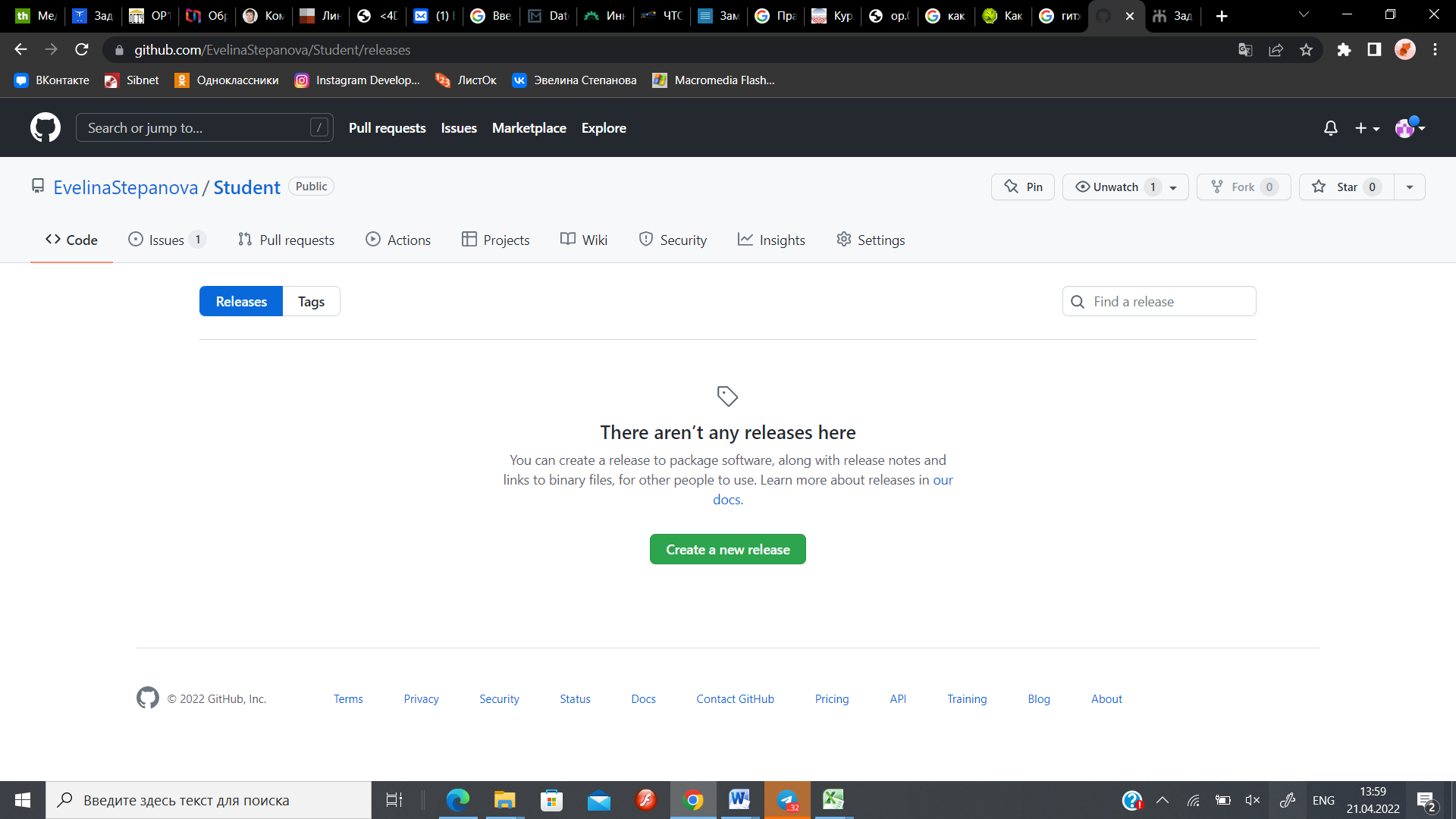
Дальше вам осталось ввести заголовок, текст и нажать **"Create new issue"**.



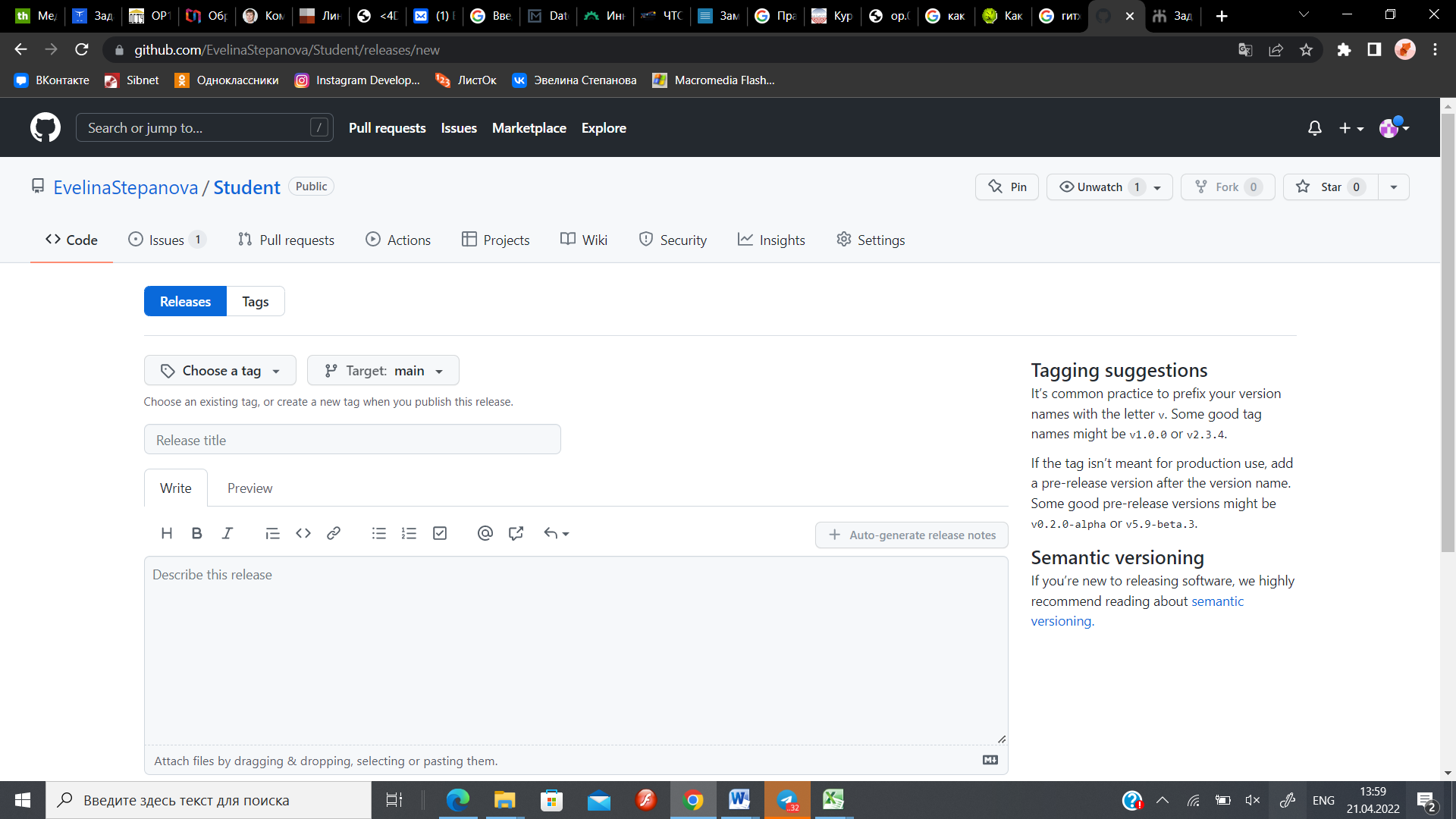


8. Релизы

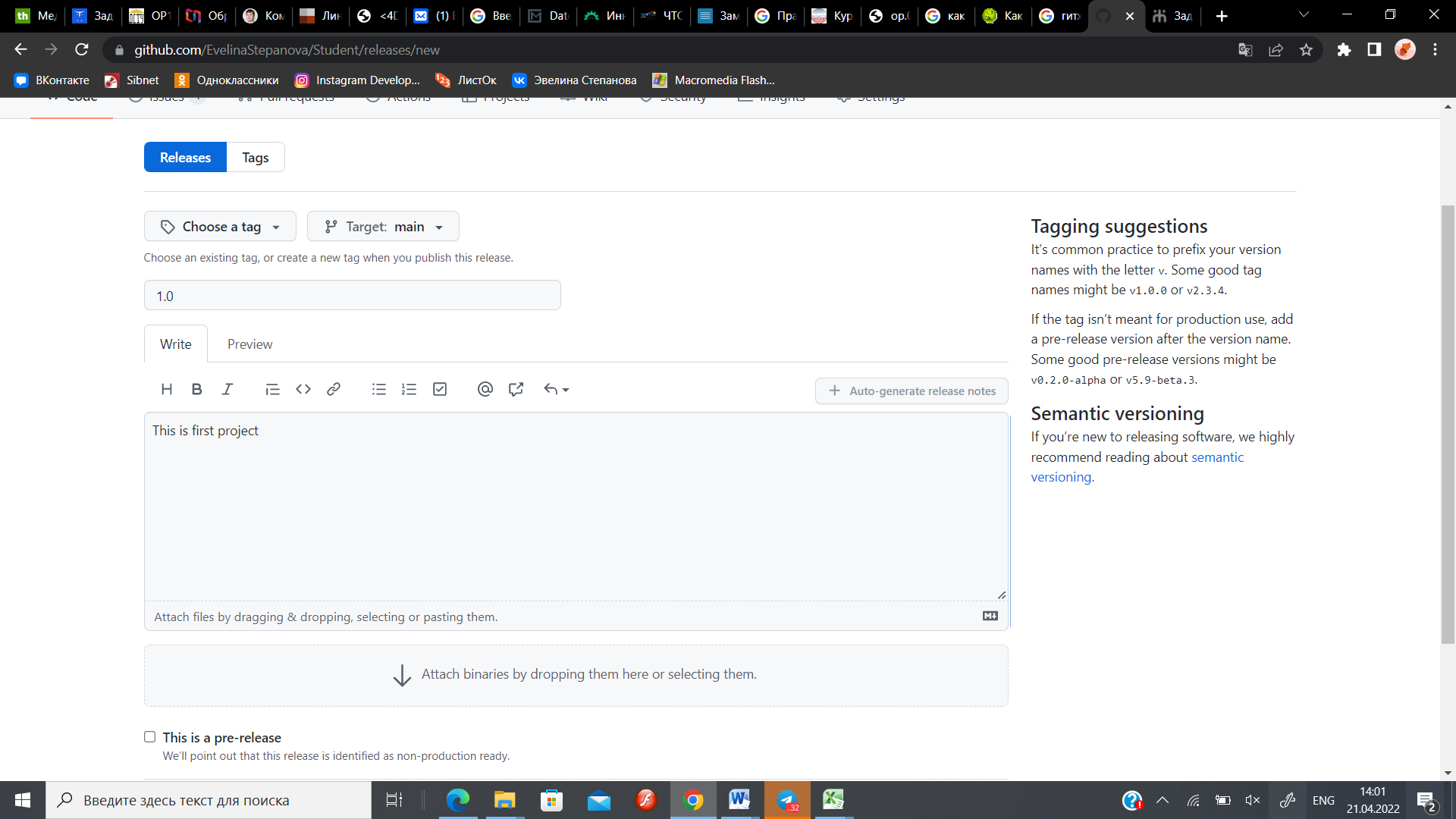
Когда продукт достиг определенной стадии можно выпустить релиз, чтобы пользователи и вы могли быть уверенны что там все стабильно и никто ничего не сломал неверным Pull Request в Master. Сначала нужно перейти на главную страницу проекта, затем на вкладку **"Releases"**:

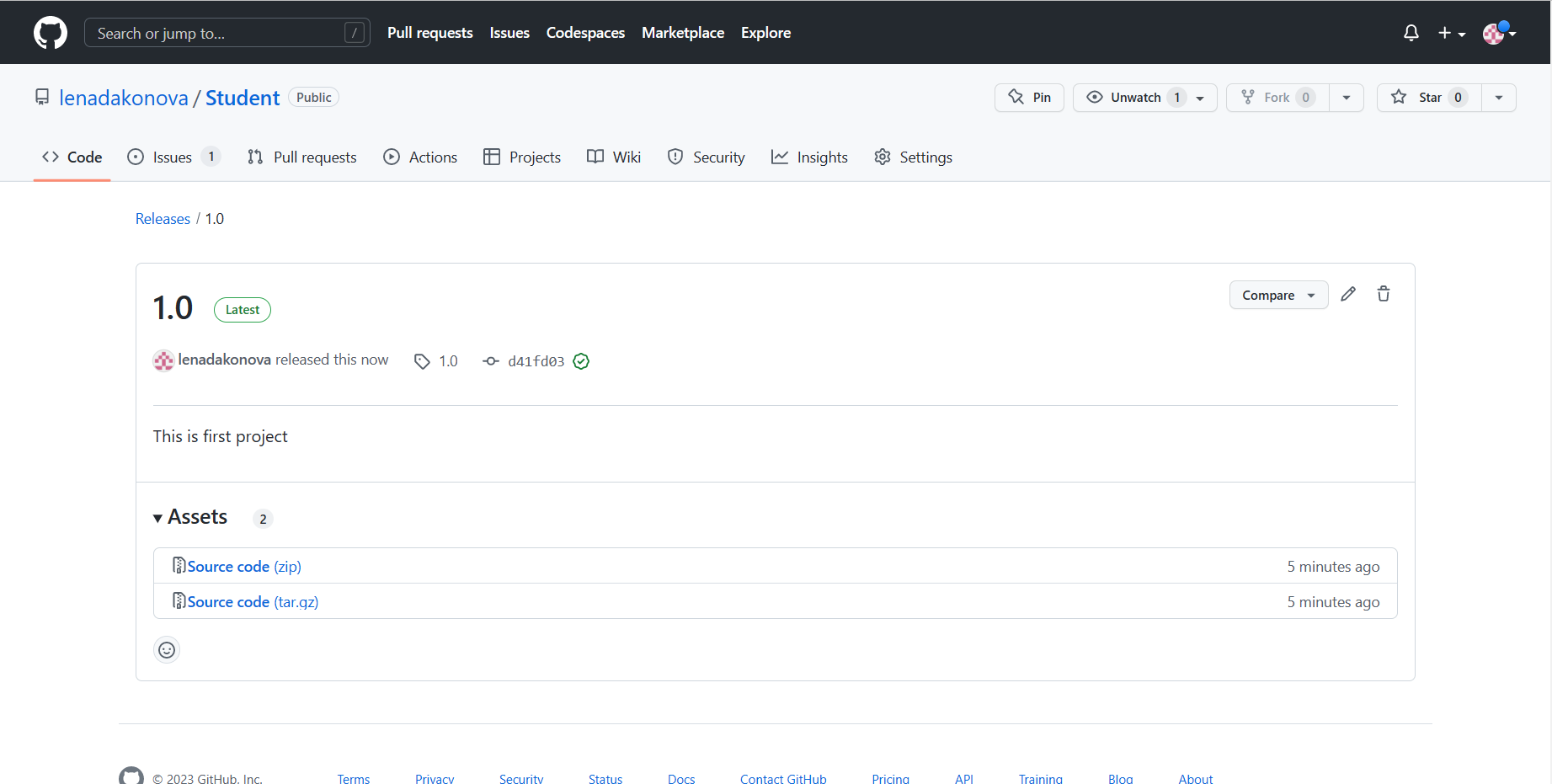


Дальше нажмите кнопку **"Create New Release"**:



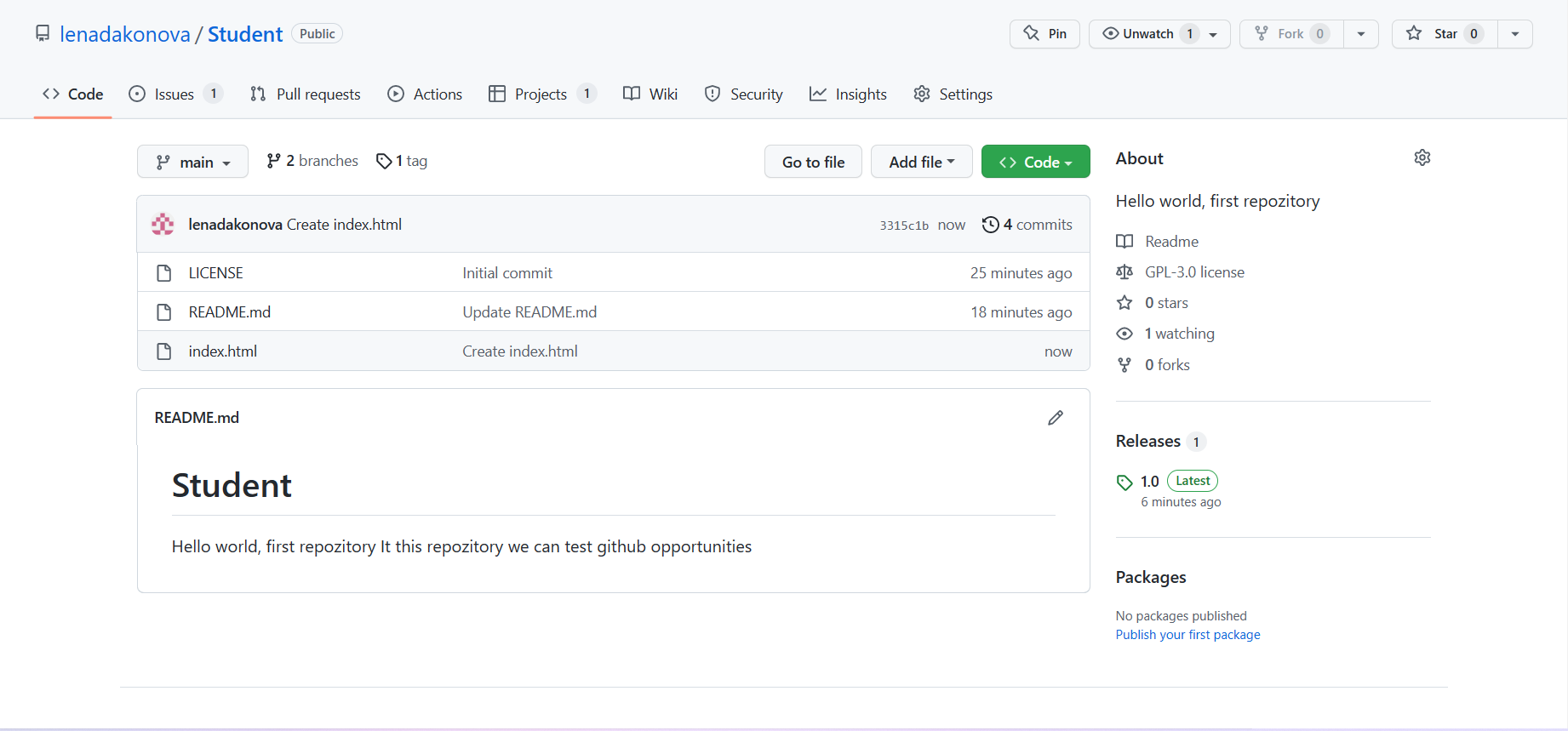
На этой странице нужно указать версию в поле **"Tag Version"**, затем имя релиза и небольшое описание. Если у вас есть скомпилированные архивы с бинарниками то их тоже нужно прикрепить сюда. Затем нажмите **"Create Release"**:

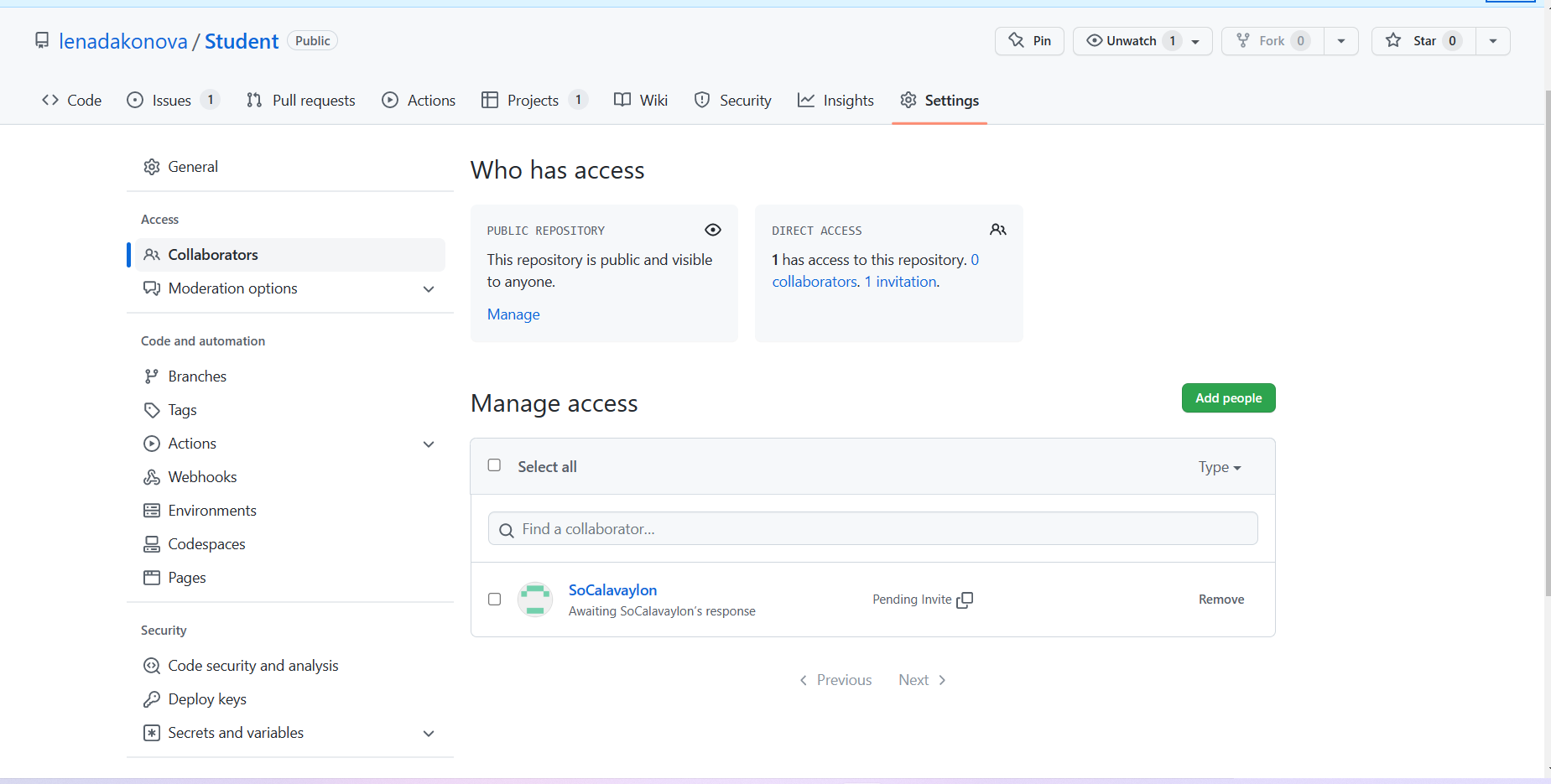




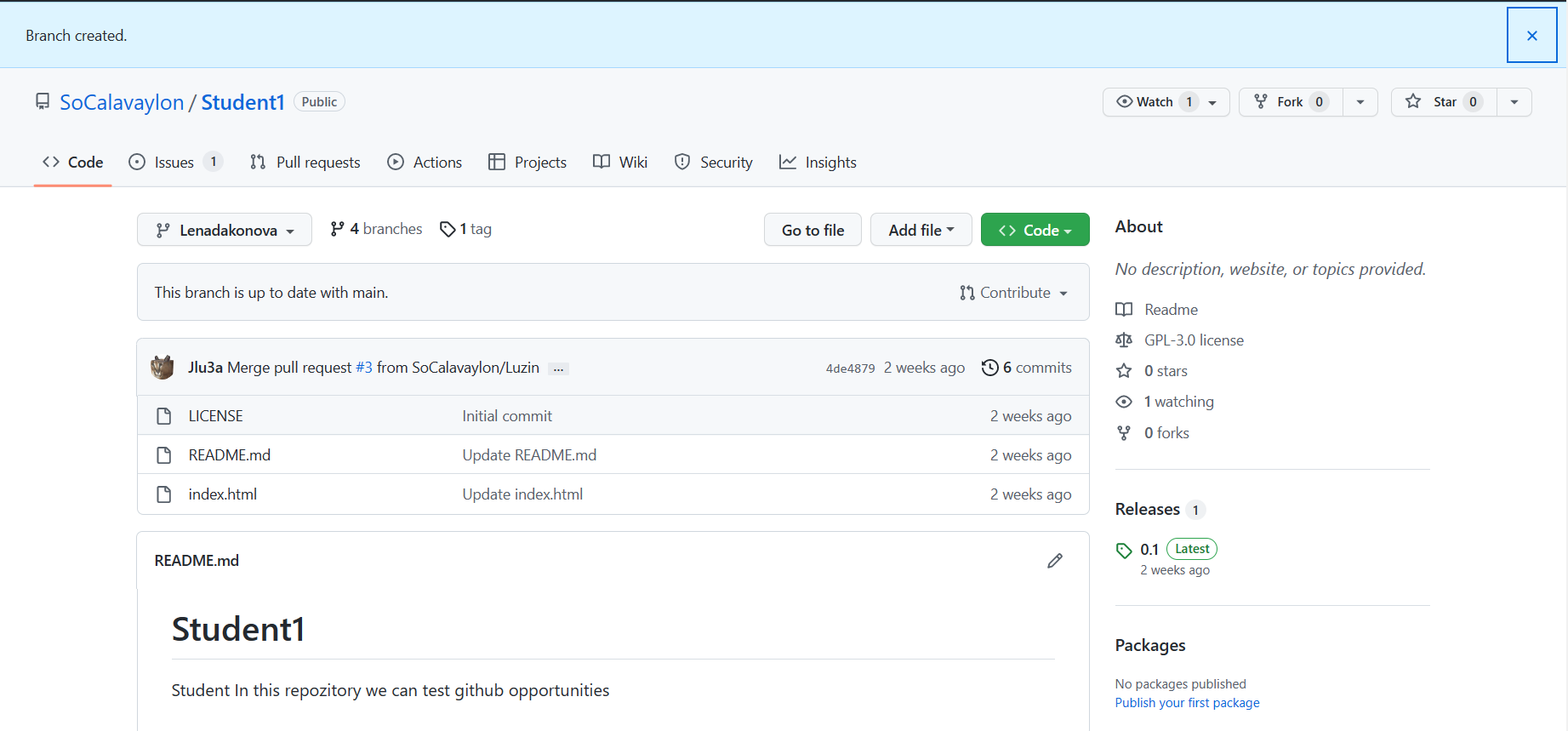
Ссылки на исходный код в tar.gz и zip будут созданы автоматически, все остальные файлы вам придется добавлять вручную.

Самостоятельно:

1. Добавить любой проект в новый репозиторий
2. 
3. Дать доступ своему одногруппнику.



1. В репозитории одногруппника создать ветку со своей фамилией.



1. Внести любые изменения в проект и сделать коммит

