1. Создание новой ветки

Создана ветка feature/schedule для добавления расписания на среду:

git checkout -b feature/schedule

Это позволило работать над модулем расписания отдельно от основной ветки main.



2. Внесение изменений в файлы

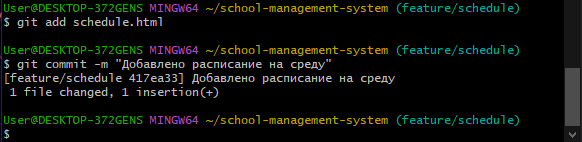
В файл schedule.html добавлено расписание на среду:

<p>Среда: Химия, Английский</p>

Коммит:

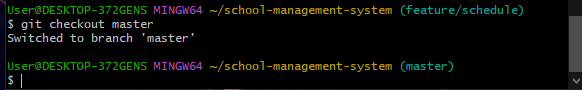
git add schedule.html

git commit -m "Добавлено расписание на среду"



3. Переключение обратно на master

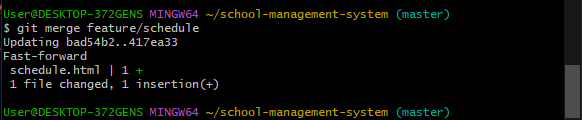
git checkout master



4. Слияние веток

После завершения работы в feature/schedule, слияние в main:

git merge feature/schedule



Если были конфликты — они разрешены (см. пункт ниже)

5. Разрешение конфликтов

Если два пользователя одновременно редактировали один и тот же файл (например, schedule.html), Git обнаруживает конфликт:

<<<<<<< HEAD

<p>Среда: История, География</p>

=======

<p>Среда: Химия, Английский</p>

>>>>>>> feature/schedule

Как было решено:

Файл открыт в текстовом редакторе

Удалены маркеры конфликта (<<<<<<<, =======, >>>>>>>)

Оставлены актуальные данные:

<p>Среда: Химия, Английский</p>

Затем:

git add views/schedule.html

git commit -m "Конфликт разрешён: объединено расписание"

Такая практика позволяет сохранить целостность проекта при командной работе

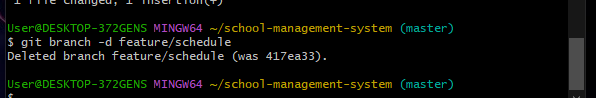
6. Удаление временных веток после слияния

После успешного слияния ветка feature/schedule больше не нужна:

git branch -d feature/schedule

Если была отправлена на сервер:

git push origin :feature/schedule



Это помогает поддерживать чистый репозиторий