1. Создание удалённого репозитория

Был создан пустой репозиторий на GitHub:

https://github.com/ваш\_логин/school-management-system

Это позволило хранить проект в облаке и делиться им с другими участниками.

2. Связь локального и удалённого репозитория

git remote add origin https://github.com/ваш\_логин/school-management-system.git

Проверка связи

git remote -v

Вывод:

origin https://github.com/ваш\_логин/school-management-system.git (fetch)

origin https://github.com/ваш\_логин/school-management-system.git (push)

Таким образом, установлено соединение между локальным и удалённым репозиторием.

3. Отправка изменений на удалённый сервер

git push -u origin main

-u связывает локальную ветку main с удалённой, чтобы в будущем можно было просто использовать git push.

Все файлы были успешно отправлены на GitHub:

schedule.html

README.md

4. Клонирование проекта на другом устройстве

Чтобы продолжить работу на другом компьютере:

cd ~/Desktop

git clone https://github.com/ваш\_логин/school-management-system.git

cd school-management-system

Проект был полностью загружен с сервера, включая историю коммитов и ветки.

5. Просмотр истории после клонирования

git log --oneline

Пример вывода:

417ea33 (HEAD -> master) Добавлено расписание на среду

bad54b2 (tag: v1.0) Revert "Добавлено расписание на среду"

f639bbd Добавлено расписание на среду

4bf2d4d Добавлено расписание на вторник

d5af08f Добавлено начальное расписание

9603f95 Initial commit

Это помогло убедиться, что история сохранена при клонировании.

6. Получение новых изменений с сервера

Если кто-то добавил новые файлы на сервер:

git pull origin main

Так можно всегда быть в курсе последних правок, особенно при командной работе.

7. Работа с несколькими удаленными репозиториями (по желанию)

Можно добавить второй удалённый репозиторий, например, для совместной разработки:

git remote add shared https://github.com/другой\_пользователь/school-management-system.git Теперь можно получать изменения как от origin, так и от shared.