МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Донецкий национальный технический университет»

Факультет ИСП

Кафедра ПИ им Л.П.Фельдмана

Лабораторная работа № 2

на тему: «Работа с ветками»

по курсу: «Профессиональная практика программной инженерии»

Проверил:

асс. каф. ПИ им. Л.П.Фельдмана Филипишин Д.А.

Выполнил:

ст. гр. ПИ-21б

Голосиенко А.И.

Донецк-2024

Цель работы – познакомиться с основами использования веток в системе контроля версий Git.

Вариант 4. Форум (с ветками-обсуждениями с обязательной реализацией групповых ролей: гость, посетитель, модератор, администратор);

1. Регистрация на GitHub (рис. 1). Аккаунт: [youngtosha](https://github.com/youngtosha). Репозиторий: [DiscussionBoard](https://github.com/youngtosha/DiscussionBoard).

1. Распределим модули по веткам и укажем модули, принадлежащие к каждой ветке:

\* Финансовые операции и прогнозирование — для модулей, связанных с финансовыми расчетами.

\* Управление задачами и командная работа — для всех функций, связанных с задачами, проектами и командами.

\* Инновации и интеграции — для модулей, которые включают новые технологии и интеграции, такие как голосовые команды, VR, чат-боты и другие.

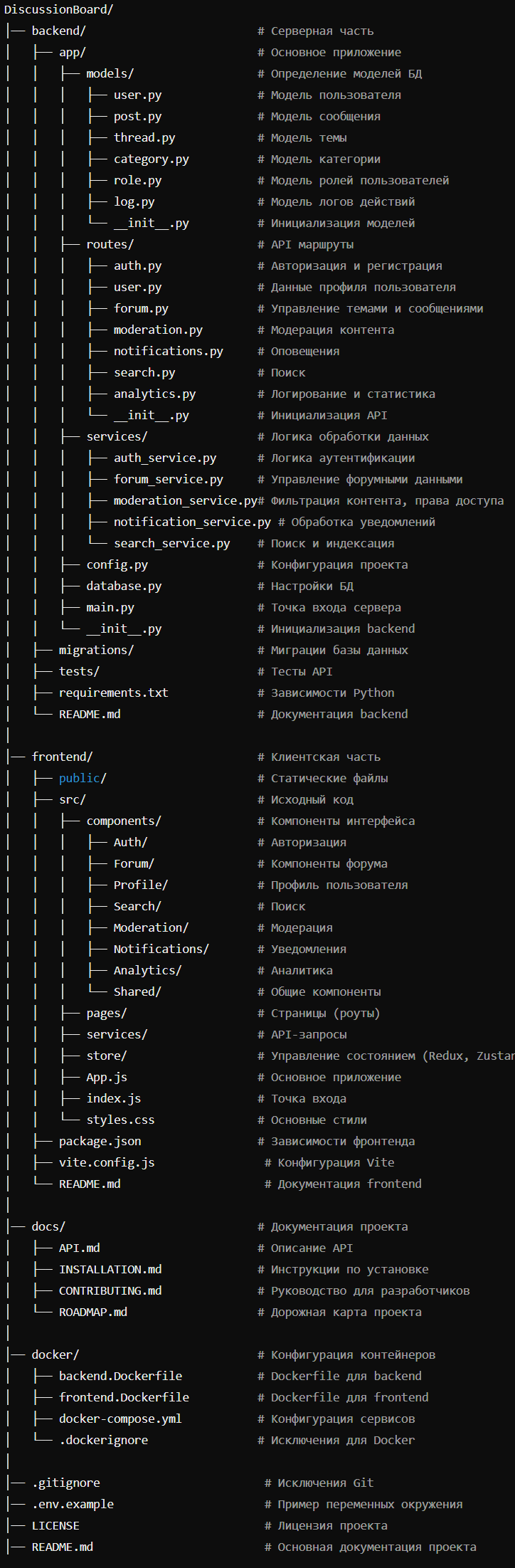


Рисунок 1 – Структура проекта

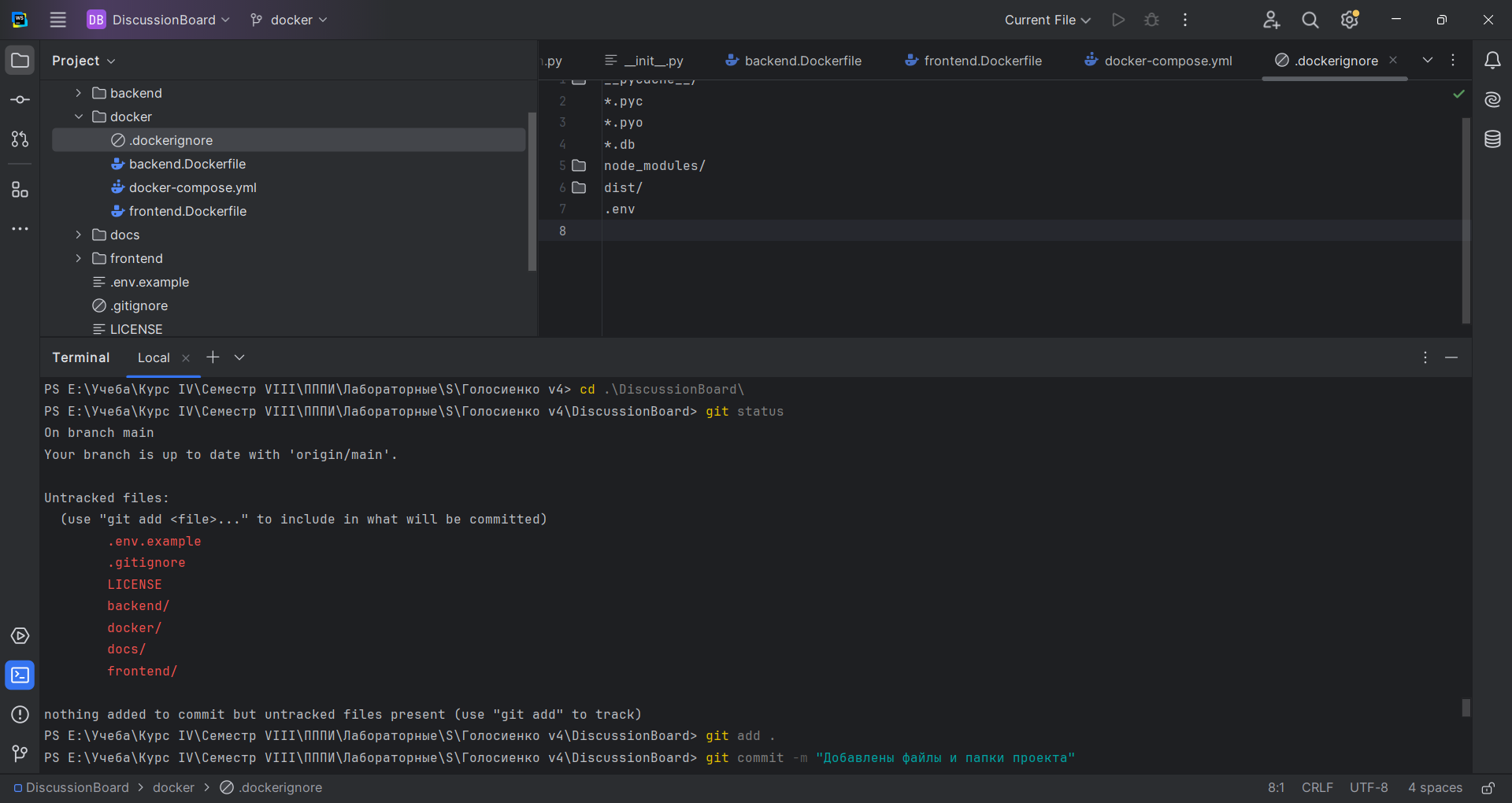


Рисунок 2 – Добавление файлов и папок в проект

2. Создадим новых веток (рис. 3-5).

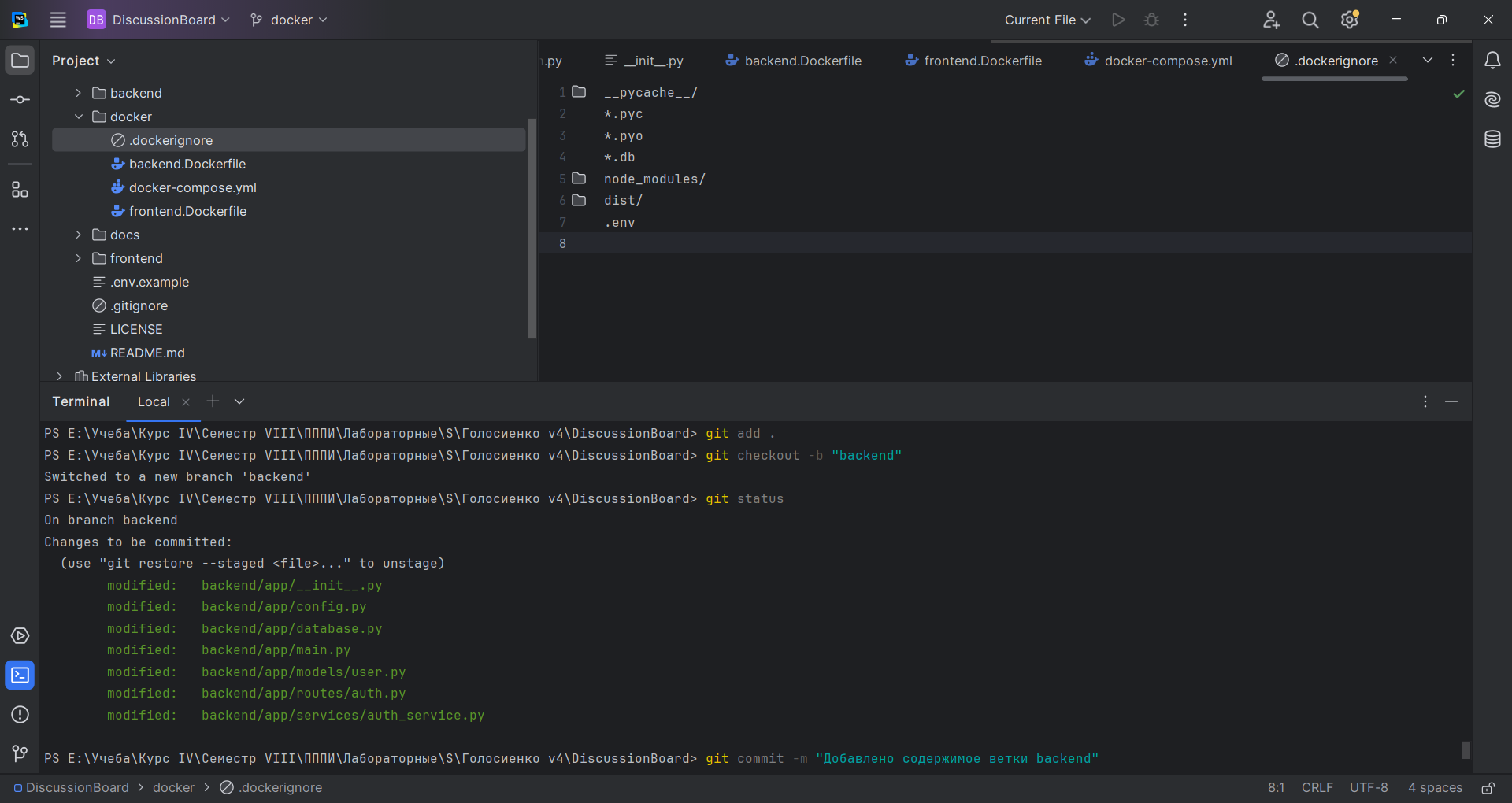


Рисунок 3 – Добавление ветки backend

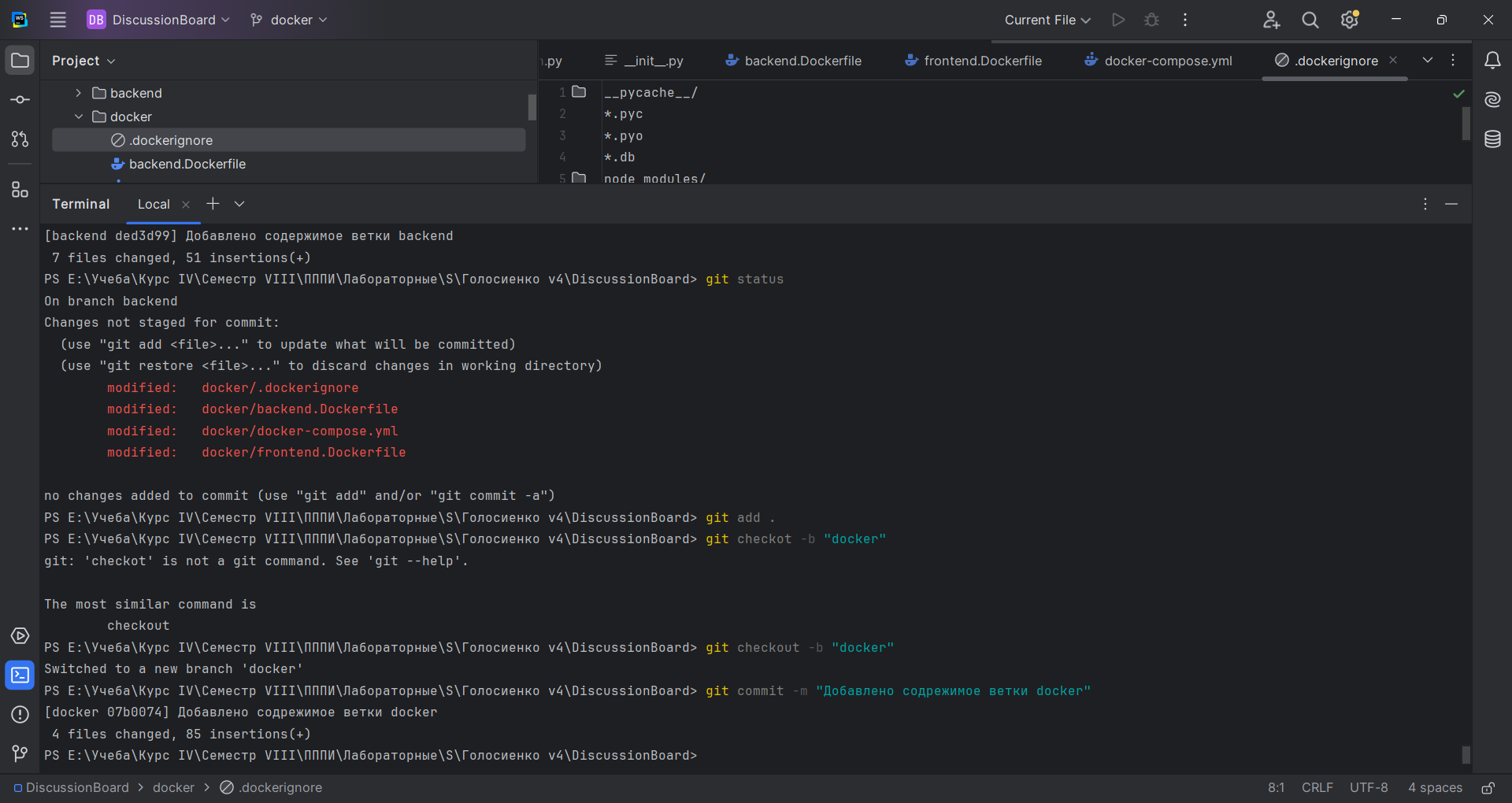


Рисунок 4 – Добавление ветки docker

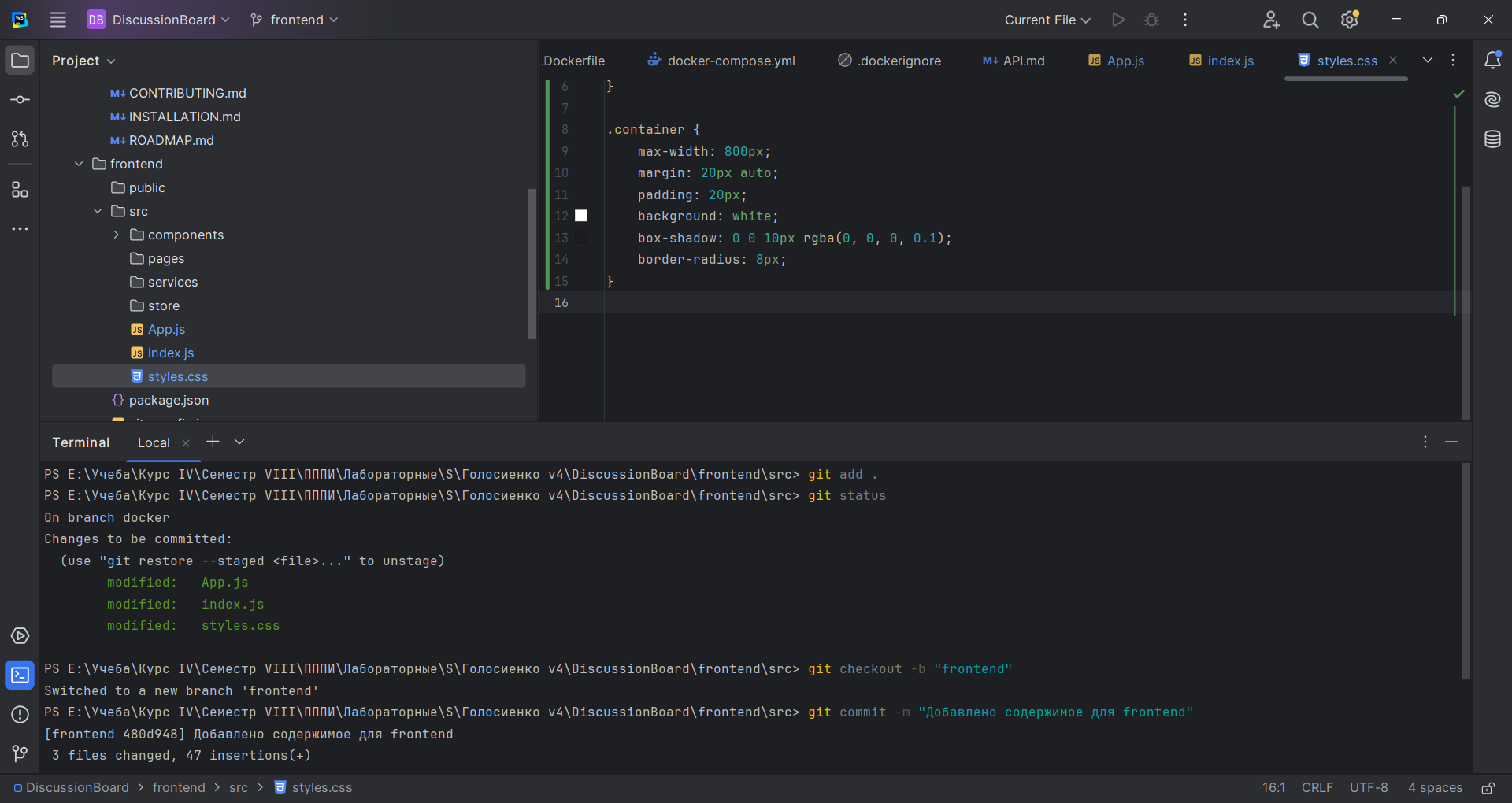


Рисунок 5 – Добавление ветки frontend

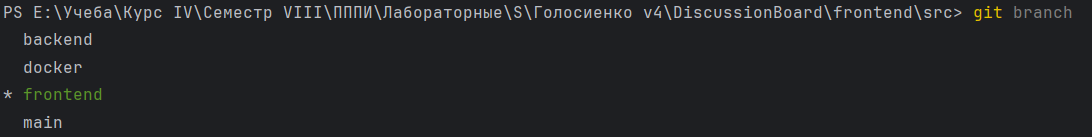


Рисунок 6 – Пример работы git branch

3. Внесем изменения в файл user.py со в обеих ветках main и beckend

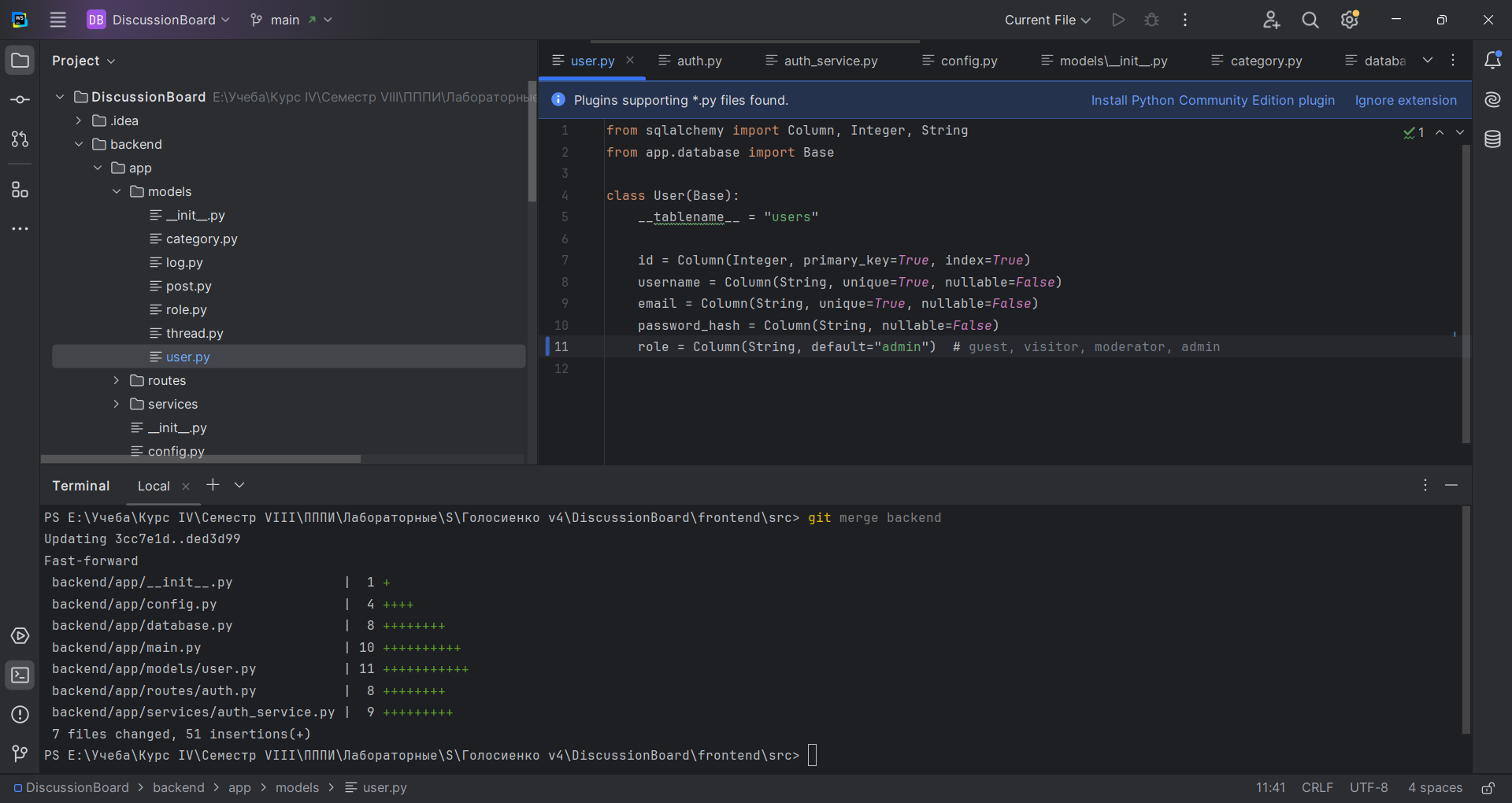


Рисунок 7 – Выполняем git merge для удобства создания конфликта

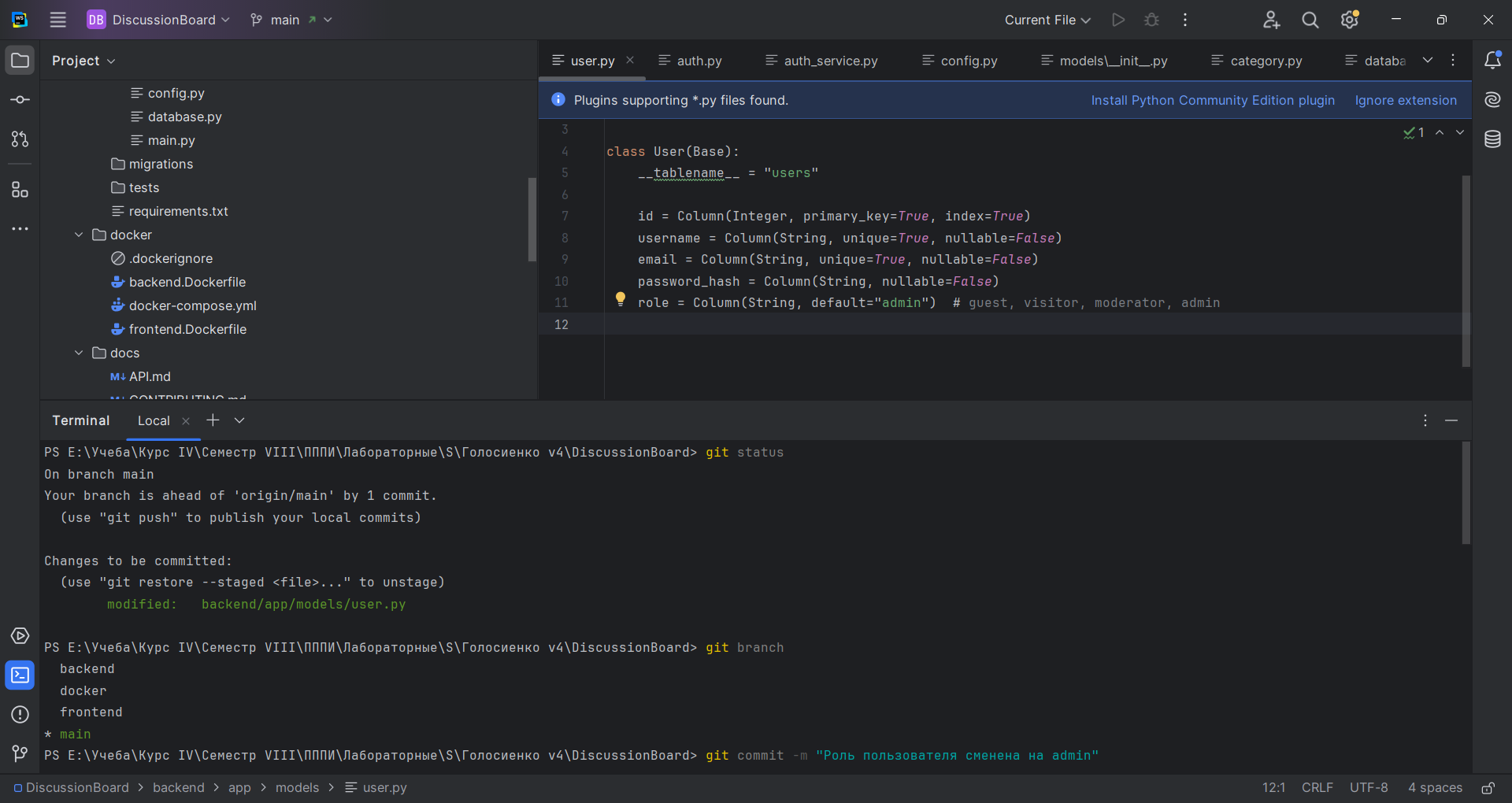


Рисунок 8 - Изменяем user.py со стороны main

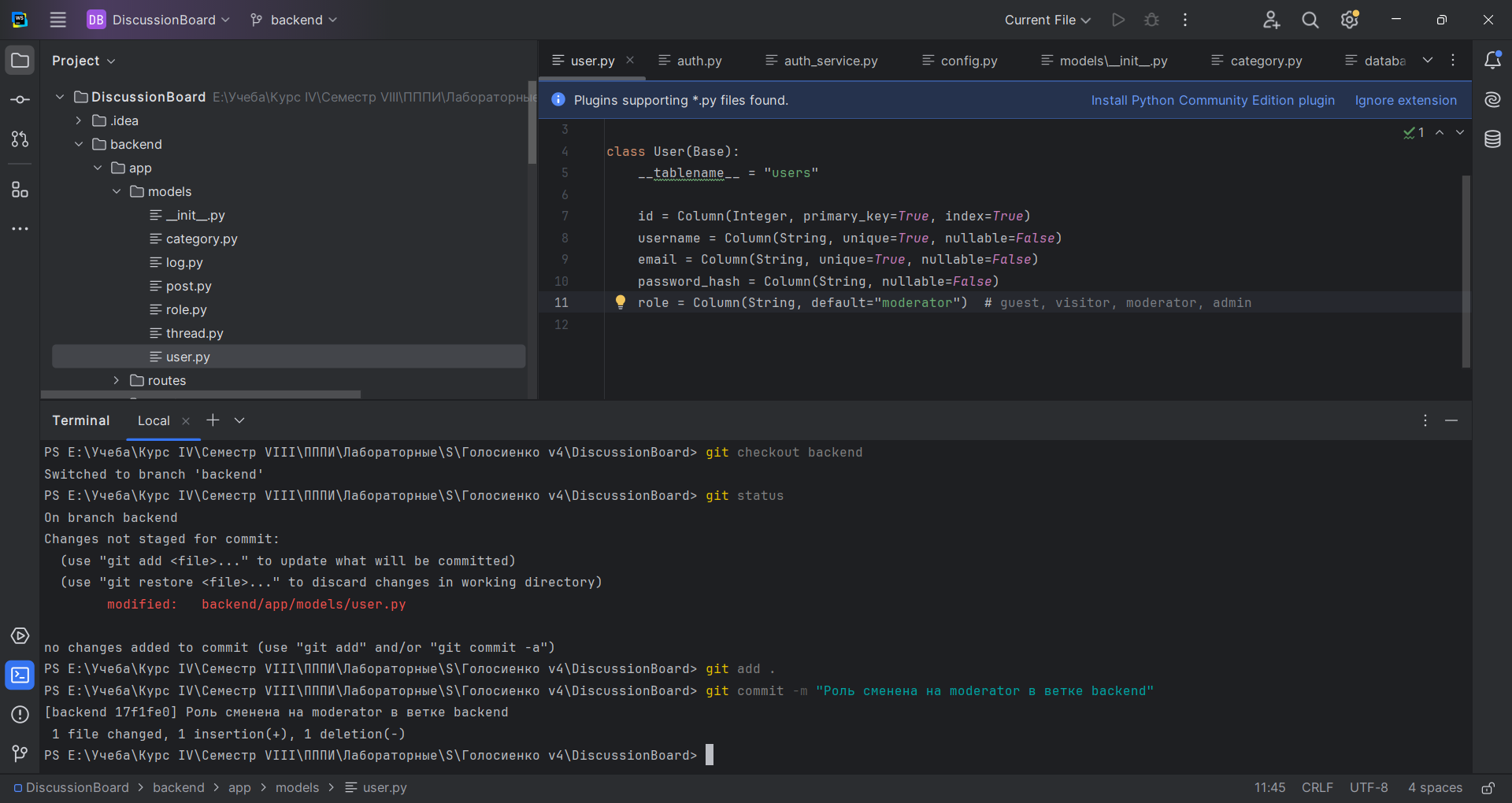


Рисунок 8 - Изменяем user.py со стороны backend

4. Выполняем git merge

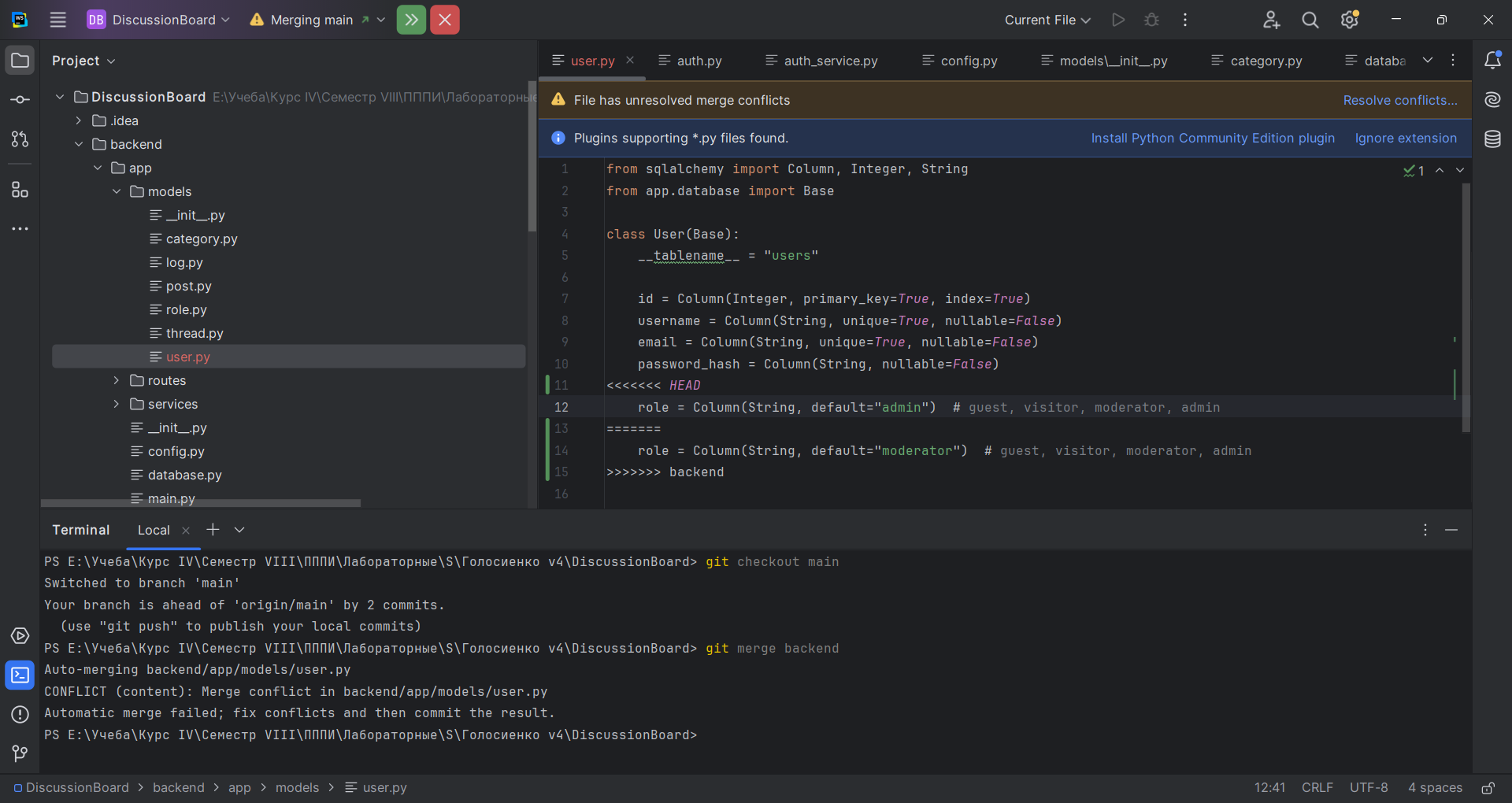


Рисунок 9 – Выполнение повторного git merge с веткой backend

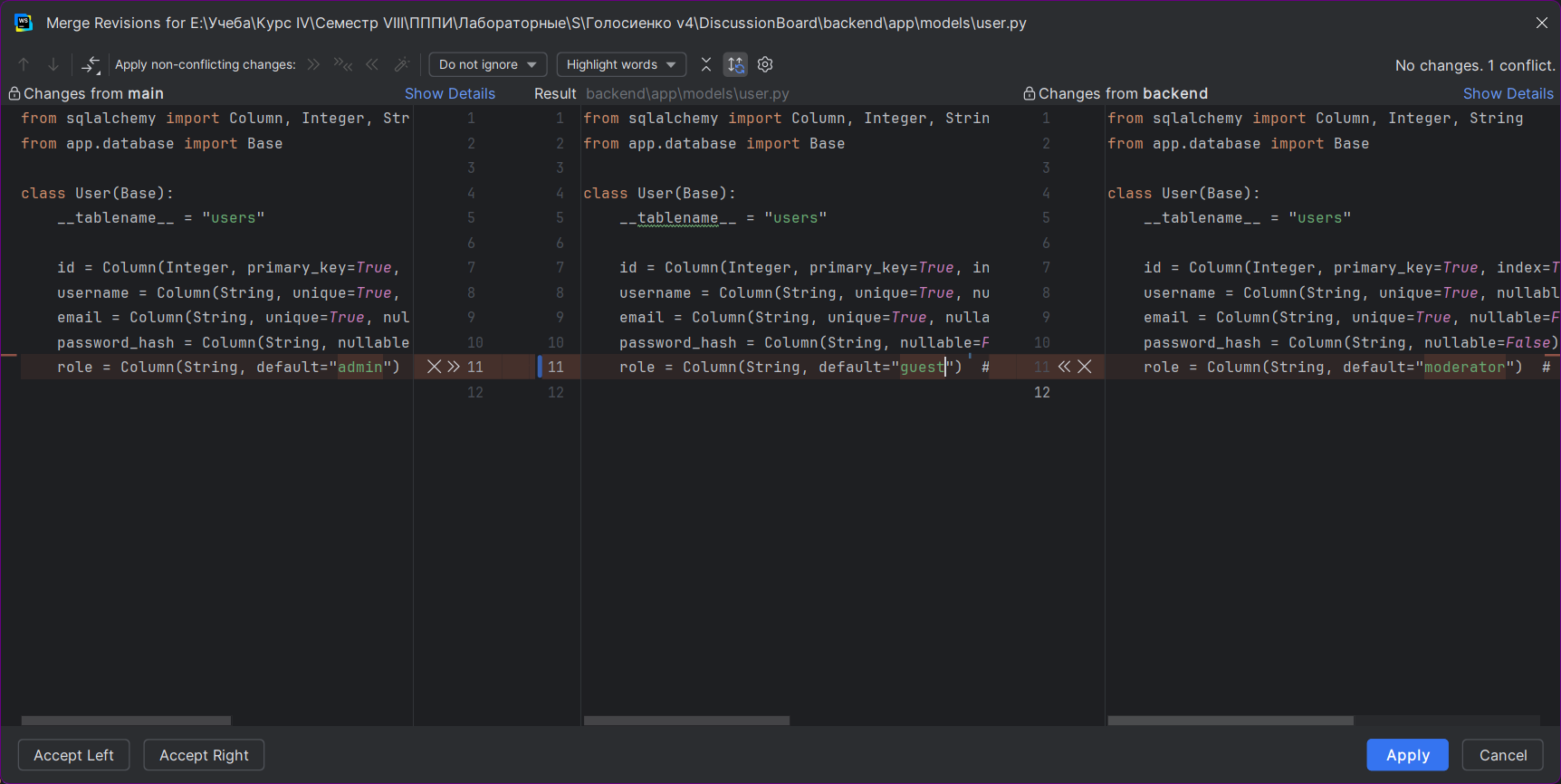


Рисунок 10 – Пример редактора конфликтов и его исправления

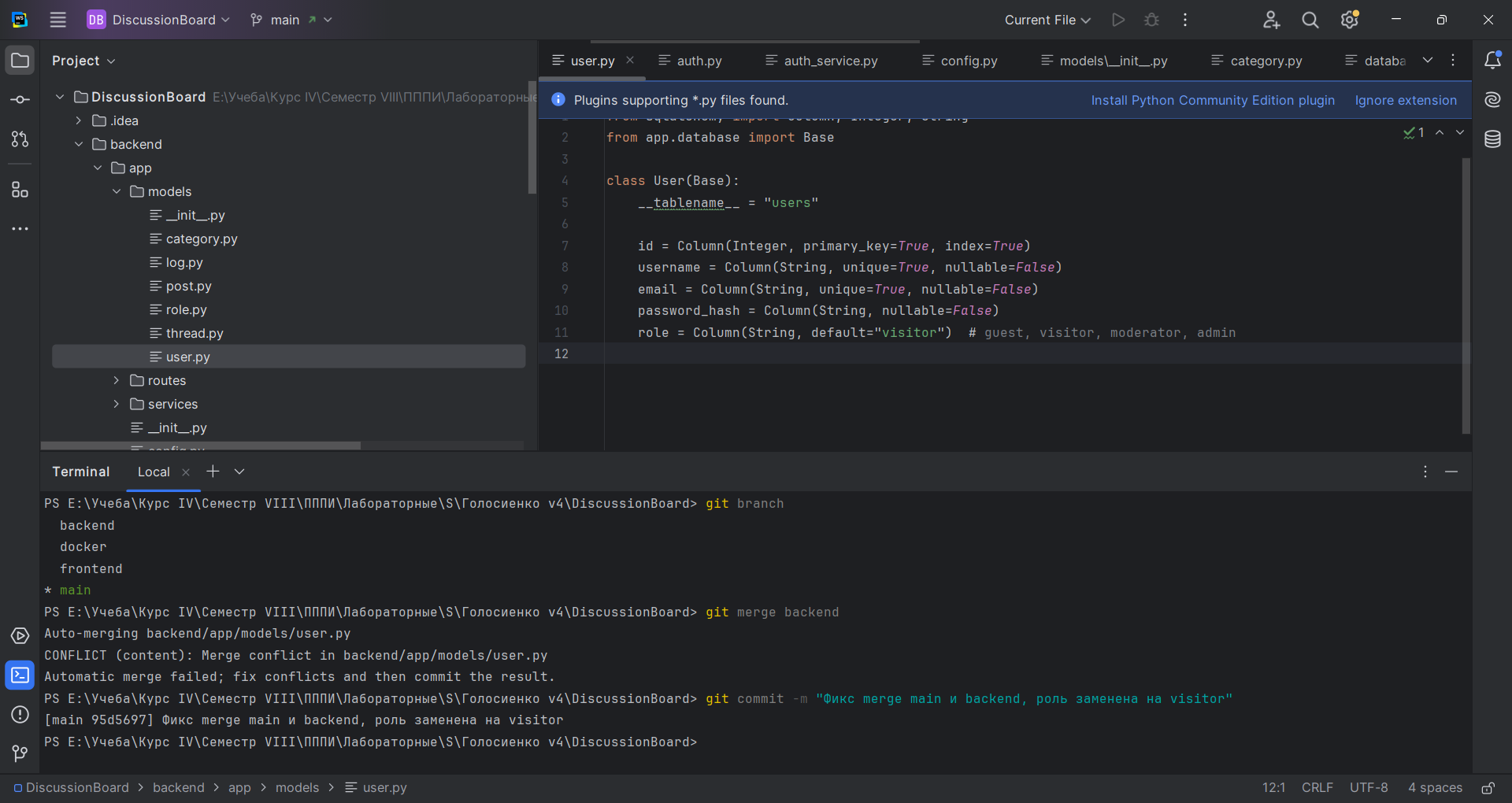


Рисунок 11 – Пример фиксации изменений после исправления конфликта

5. Результаты работы

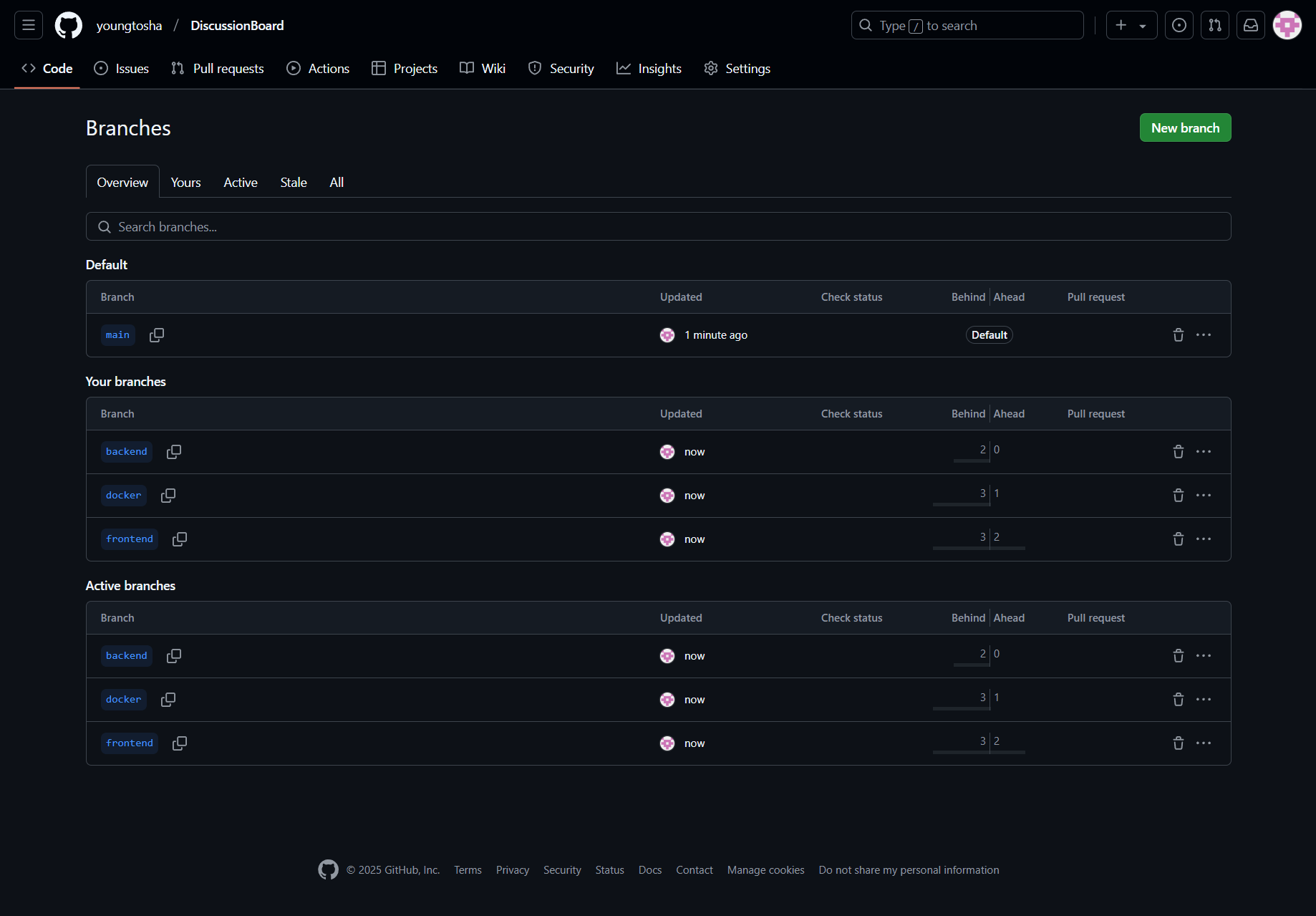


Рисунок 12 – Скриншот веток в GitHub

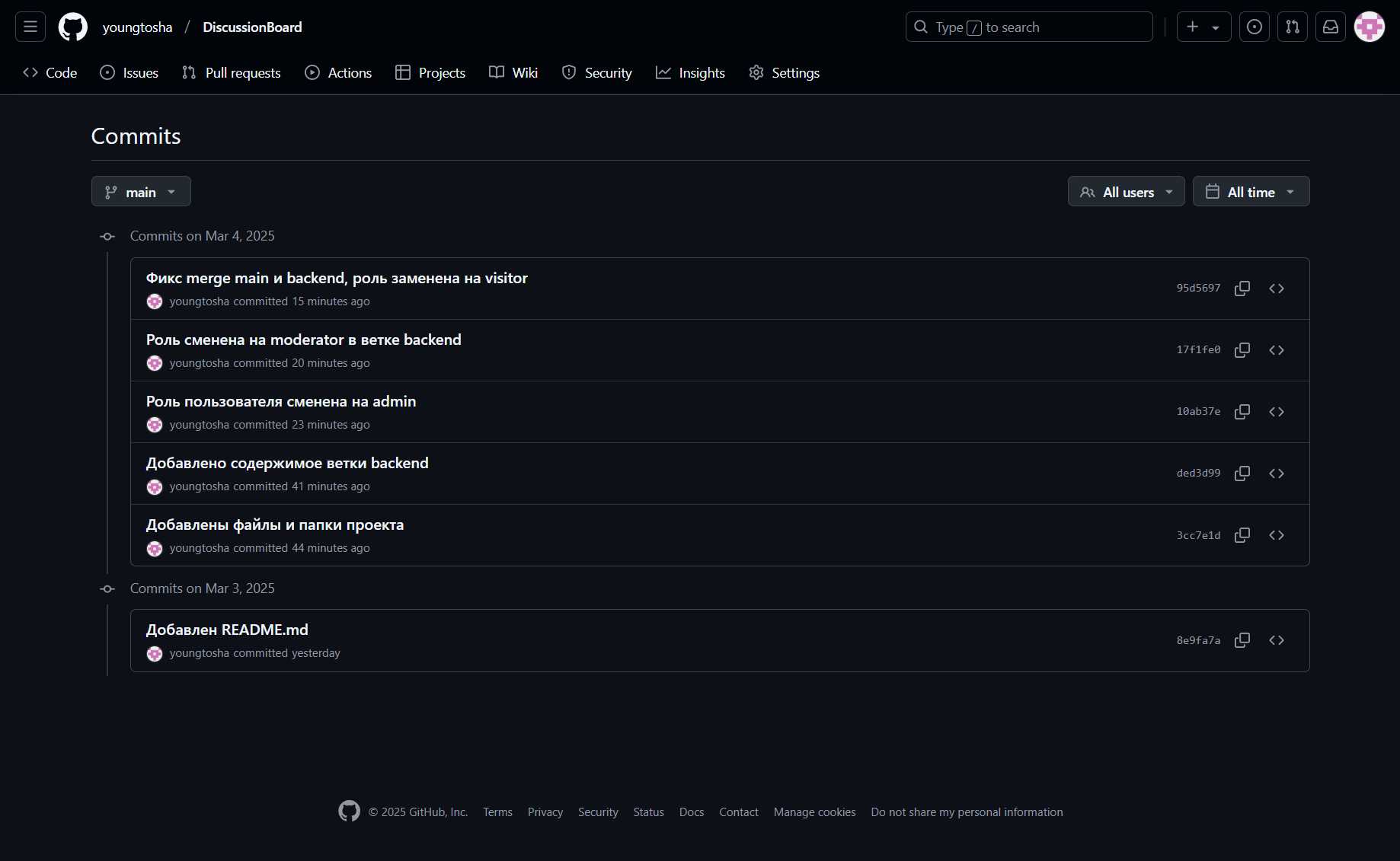


Рисунок 13 – Коммиты ветки main

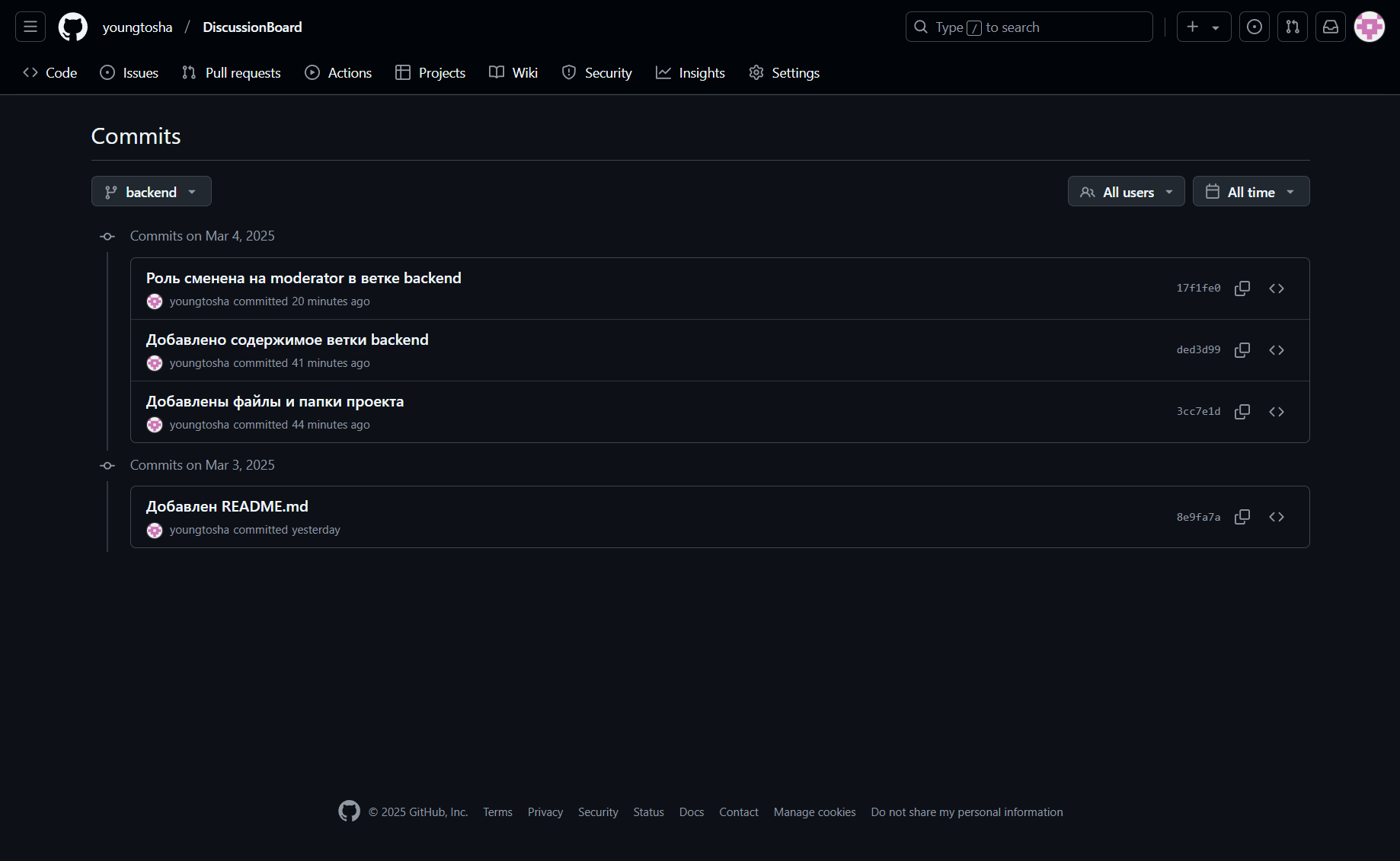


Рисунок 14 – Коммиты ветки backend

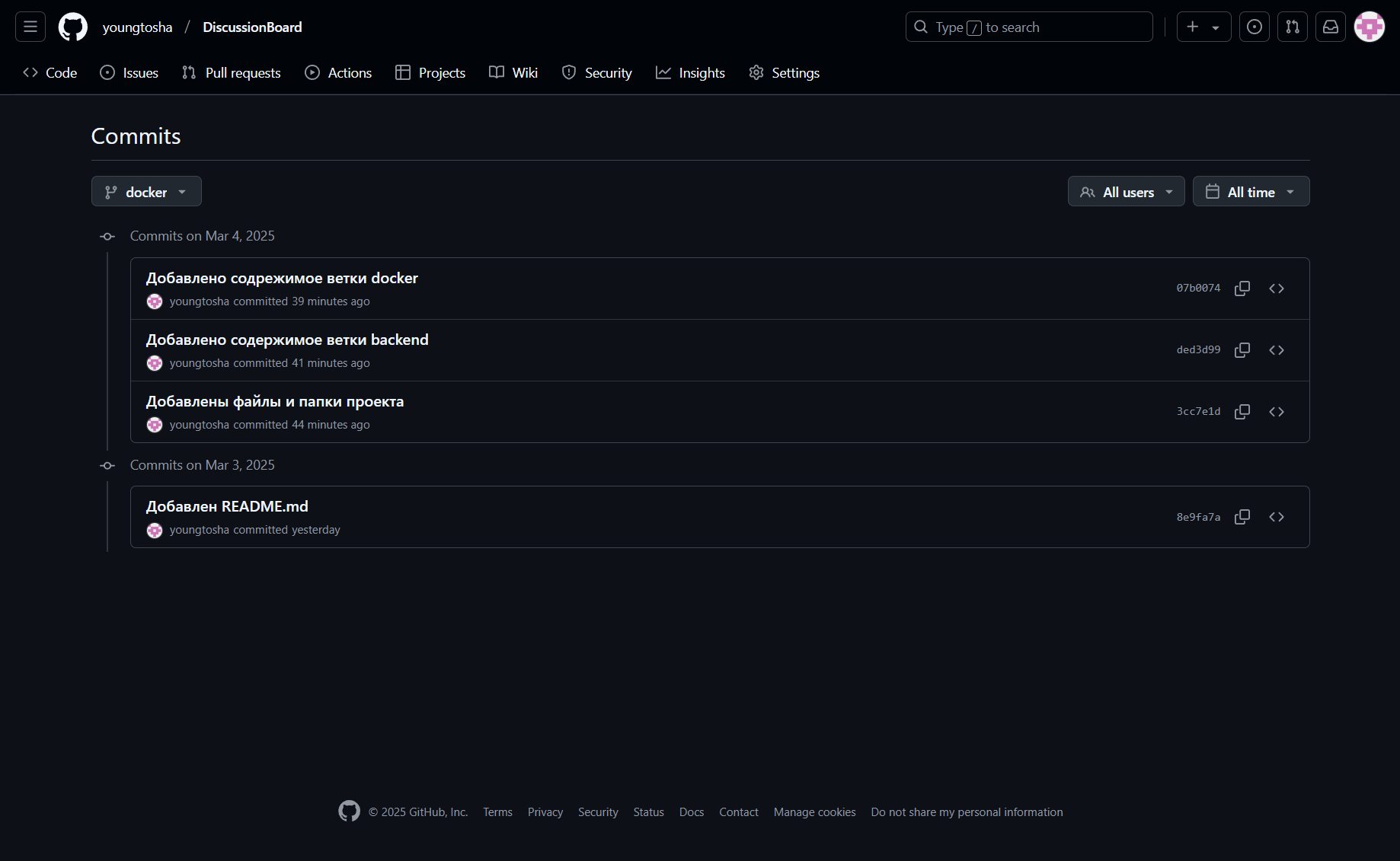


Рисунок 15 – Коммиты ветки docker

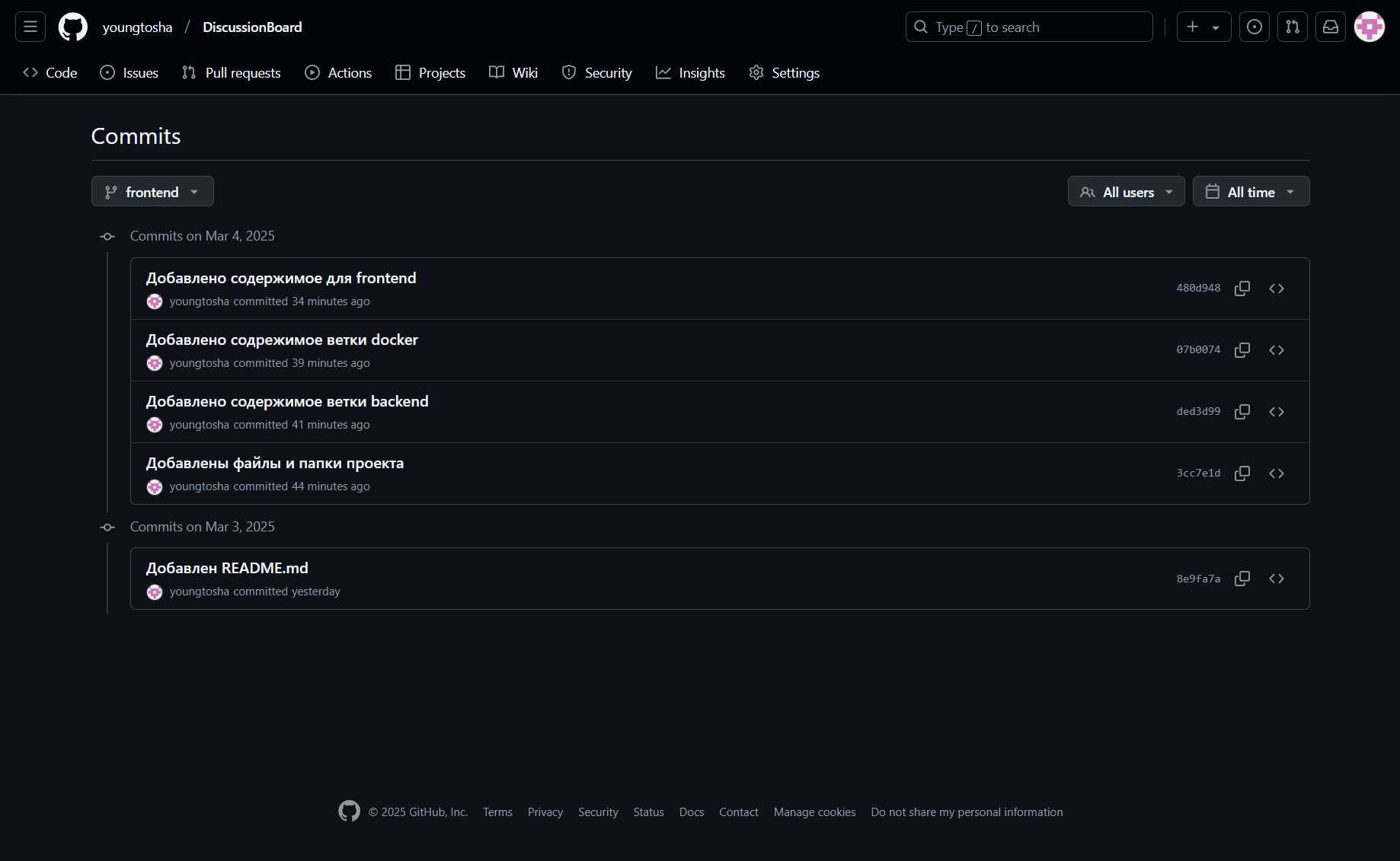


Рисунок 16 – Коммиты ветки frontend