

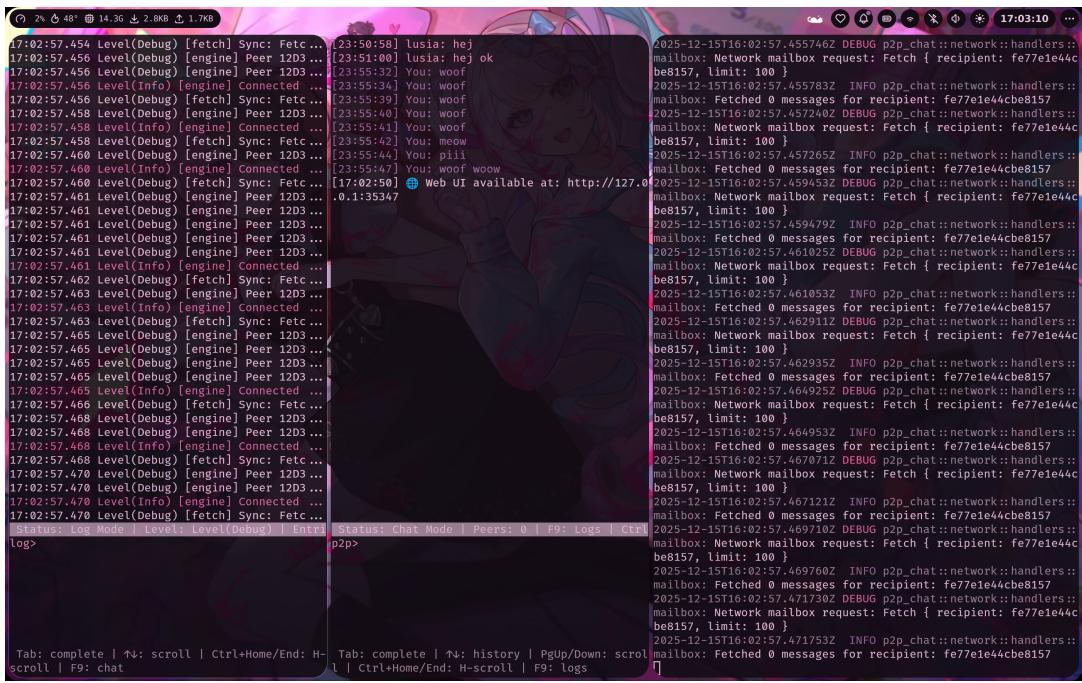
## Załącznik 2. Zrzuty ekranu aplikacji p2p-chat

### 1. Interfejs tekstowy (TUI)

Na rysunku 1 przedstawiono przykładowy widok interfejsu tekstowego aplikacji p2p-chat. Zrzut ekranu obejmuje jednocześnie pracę trzech węzłów uruchomionych w jednym terminalu:

- po lewej stronie widoczny jest węzeł użytkownika pracujący w trybie podglądu logów, prezentujący zdarzenia sieciowe oraz komunikaty diagnostyczne;
- w środkowym oknie pokazano tryb czatu (ChatMode) dla węzła klienta, obejmujący listę wiadomości oraz pole wprowadzania nowych komunikatów;
- po prawej stronie widoczny jest węzeł pełniący rolę *mailboxa*, odpowiedzialny za przechowywanie wiadomości dla nieobecnych uczestników.

Widok ten ilustruje sposób pracy systemu w środowisku terminalowym, w którym użytkownik ma równoczesny dostęp do historii rozmów, bieżących logów oraz informacji o pracy węzła *mailbox*.



Rysunek 1: Interfejs tekstowy (TUI) z trzema węzłami: użytkownik w trybie logów, klient w trybie czatu oraz węzeł mailbox.

## 2. Interfejs webowy

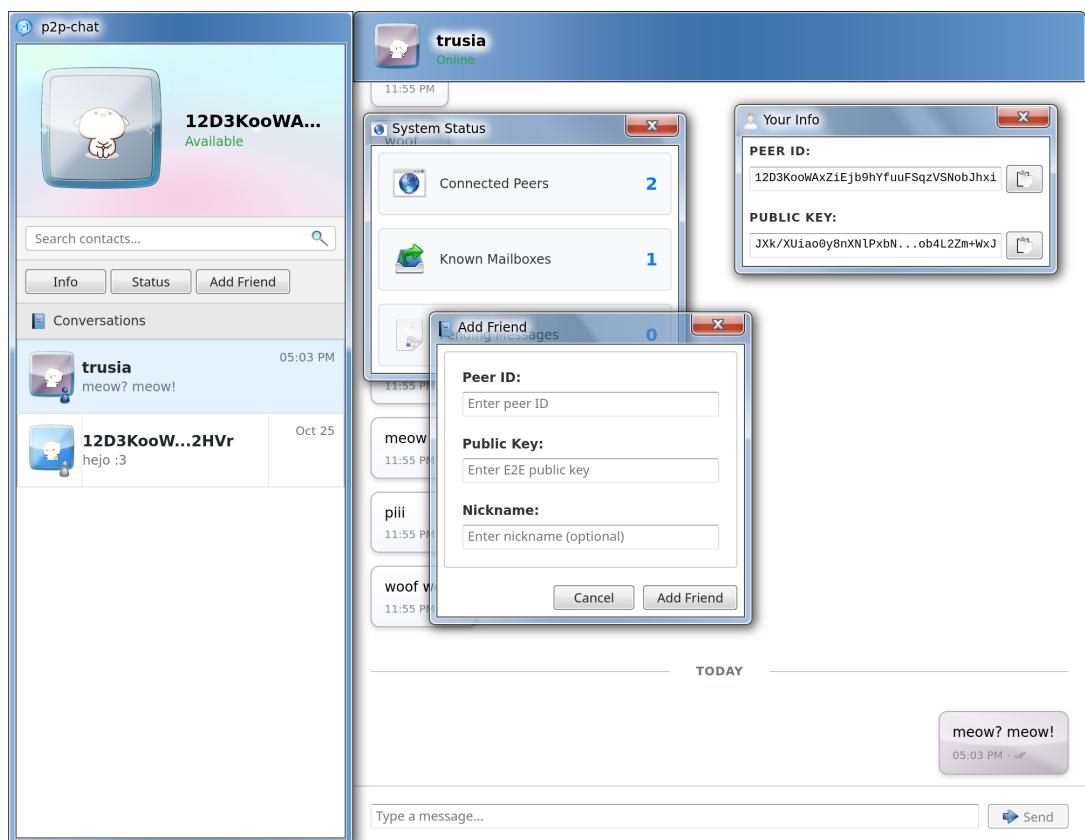
Rysunek 2 przedstawia widok interfejsu webowego komunikatora p2p-chat. Na zrzucie ekranu widoczna jest aktywna konwersacja pomiędzy dwoma użytkownikami, prezentowana w oknie czatu wraz z przypisanymi kolorami oraz awatarami dopasowanymi do identyfikatorów *PeerId*. Kolorystyka oraz grafiki awatarów są deterministycznie wyprowadzane z identyfikatora węzła, co ułatwia wizualne rozróżnianie rozmówców.

W prawym obszarze interfejsu widoczne są dodatkowe okna:

- **System Status** – panel prezentujący bieżące informacje o stanie węzła, połączeniu z siecią P2P oraz działaniu modułów pomocniczych;
  - **Add Friend** – okno umożliwiające dodanie nowego znajomego na podstawie jego identyfikatora, w tym klucza publicznego używanego do szyfrowania end-to-end;

- **Your Info** – panel wyświetlający dane lokalnego użytkownika, w szczególności identyfikator *PeerId* oraz klucz publiczny wykorzystywany do nawiązywania bezpiecznych połączeń.

Zrzut ekranu obrazuje docelowy sposób korzystania z aplikacji przez użytkownika końcowego – komunikacja odbywa się w przeglądarce, natomiast logika sieciowa i kryptograficzna jest realizowana przez lokalny węzeł Rust uruchomiony w tle.



Rysunek 2: Interfejs webowy z aktywną konwersacją oraz otwartymi panelami System Status, Add Friend i Your Info.