

# Team Task Plan: Chatbox Application

Project: Chatbox Application (Python Server + Web Client)

Team Division Plan:

Person 1 - Server Developer (Python)

- Build TCP server
- Handle client connections & threading
- Broadcast logic
- Logging / debugging

Person 2 - Bridge Developer (WebSocket to TCP)

- Create WebSocket to TCP bridge (Node.js or Python with websockets)
- Test with TCP server
- Ensure message sync

Person 3 - Frontend Developer (UI)

- Design and implement index.html, style.css
- Create clean responsive UI
- Handle scroll, message formatting

Person 4 - Frontend JS Logic / Integration

- Write script.js
- Connect WebSocket to bridge
- Handle send/receive events

## Team Task Plan: Chatbox Application

- Integrate with UI

### Collaboration Plan:

- Daily stand-ups (What's done, blockers, next tasks)
- Use shared GitHub repo or cloud folder
- Test integration progressively

### Integration Flow:

1. Server developer confirms tcp\_server.py works standalone
2. Bridge developer builds and tests ws\_bridge.js
3. Frontend team integrates and tests with WebSocket bridge
4. Final integration and polishing

### Suggested 5-Day Timeline:

Day 1: Server setup + Bridge setup + UI mock

Day 2: TCP server complete + WebSocket relay built

Day 3: UI and JS integration

Day 4: Full integration testing

Day 5: Polish, bug fix, add optional features

### Optional Features:

- Username prompt before joining chat
- Message timestamps

## **Team Task Plan: Chatbox Application**

- Emoji parsing
- Chat clear button
- Dark/light mode toggle