Phase 1: Setup & Connection

1. Verify `credentials.json` is valid and authorized for YouTube API v3.
2. Start the bot with a live stream running:
3. Confirm console prints: `[INFO] Connected to active live broadcast <id>`.
4. Confirm “Hello, world!” message appears in chat.
5. Stop the live stream and restart bot:
6. Confirm fallback to `video-id` triggers.
7. Confirm warning message `[WARN] No active live stream. Falling back to video-id from YAML`.

Phase 2: Chat Reading & Logging

1. Send normal chat messages.
2. Confirm console output `[CHAT] @username: message`.
3. Check `youtube-chat-full.txt` logs all messages.
4. Send `!play` and `!swap` commands.
5. Confirm `youtube-chat.txt` only logs swap/play messages.
6. Confirm Lua plugin can read `youtube-played.txt`.

Phase 3: !play Command

1. Send `!play <exact game>`:
2. Confirm matched game logged.
3. Confirm Lua receives game in `youtube-played.txt`.
4. Confirm API sends chat message.
5. Send `!play <partial/fuzzy game>`:
6. Confirm proper partial match occurs.
7. Send `!play <unknown game>`:
8. Confirm "No match found" message sent to chat.
9. Confirm `youtube-played.txt` writes `__NO_MATCH__`.
10. Confirm strike applied.

Phase 4: !swap Command

1. Send `!swap`:
2. Confirm chat message confirmation.
3. Confirm Lua receives the command.

Phase 5: Cooldowns & Global Cooldowns

1. Issue multiple `!play` from the same user quickly:
2. Confirm per-user cooldown prevents repeated commands.
3. Confirm warning message with remaining seconds.
4. Confirm strike applied.
5. Issue multiple `!play` from different users quickly:
6. Confirm global cooldown prevents immediate successive commands.

Phase 6: Strikes, Timeouts & Bans

1. Trigger strikes by violating cooldown:
2. Confirm strikes increment.
3. After `STRIKES_BEFORE_TIMEOUT`, confirm timeout applied.
4. Confirm timeout message sent.
5. Trigger multiple timeouts:
6. Confirm `TIMEOUTS_BEFORE_BAN` triggers a ban.
7. Confirm API call bans the user.
8. Confirm chat notification sent.
9. Confirm timed-out users cannot issue commands.
10. Confirm banned users cannot issue commands.

Phase 7: Host Commands

1. Test `!lock`:
2. Confirm commands from other users ignored.
3. Confirm chat message confirms lock.
4. Test `!unlock`:
5. Confirm commands from users accepted again.
6. Test `!timeout <user>`:
7. Confirm user is timed out.
8. Test `!ban <user>`:
9. Confirm user banned via YouTube API.
10. Confirm notification sent.

Phase 8: Lua Integration

1. Ensure `youtube-played.txt` updates with every !play.
2. Confirm Lua plugin reads new lines and calls `swap_game` with correct game.
3. Confirm `__NO_MATCH__` lines trigger Lua notifications.

Phase 9: Edge Cases

1. Send consecutive commands if disabled.
2. Test locked state.
3. Test multiple users violating cooldown/timeouts.
4. Confirm console logs match expected messages.

Optional Stress Tests

1. Rapid-fire !play commands from multiple users.
2. Simulate API errors by revoking credentials temporarily.
3. Confirm bot reconnects if chat connection fails.