Performance Measurement and Optimization Report

Project: Advanced Tic Tac Toe Game

1. Purpose

The purpose of this document is to report on the performance measurements and optimizations applied during the development of the Advanced Tic Tac Toe Game. We aim to ensure smooth gameplay, fast response times, minimal resource usage, and overall efficient performance.

2. Measurement Setup

Metric	Tool/Method
Response Time	Manual profiling using Qt's event loop observation + Stopwatch/QElapsedTimer.
Memory Usage	Task Manager (Windows) / Activity Monitor (macOS) during game sessions.
CPU Utilization	System Monitor during gameplay (especially AI move calculation).
FPS/UI Responsiveness	Visual inspection; Qt provides native fast UI unless overloaded.

Tests were done:
- During login/register operations.
- During player moves.
- During AI move computation.
- While viewing game history.

3. Performance Benchmarks

Scenario	Average Response Time	CPU Spike / Memory Usage
Login/Register	~50ms	<1% CPU / +2 MB
Player Move (vs Human)	~15ms	<1% CPU / Negligible
Player Move (vs AI Easy Board State)	~50-100ms	3-5% CPU / +2 MB
Player Move (vs AI Complex Board State)	~200-400ms	8-12% CPU / +4 MB
Game History Retrieval (10+ entries)	~70ms	<2% CPU / +1 MB

4. Optimization Techniques Applied

4.1 AI Optimization

- Implemented alpha-beta pruning to reduce computation.
- Reduced branch evaluation when a definitive win/loss path is found.
- Used depth-aware scoring in Minimax to prioritize faster wins.

4.2 UI Optimization

- Pre-loaded UI elements.
- Minimal runtime stylesheet recalculations.
- Event-driven programming model ensuring non-blocking operations.

4.3 Database Optimization

- Persistent database connection during app lifecycle.
- Prepared SQL statements to minimize parsing overhead.

5. Potential Future Optimizations

Area	Possible Improvement
AI Computation	Introduce move ordering heuristics for faster alpha-beta pruning.
Memory Usage	Lazy load game history if records exceed 500 entries
Security	Hash passwords with SHA-256 or bcrypt
Multithreading	Offload AI computation to a background thread for better UI responsiveness.

6. Conclusion

Overall, the Advanced Tic Tac Toe Game meets its performance goals:

- Instant player move feedback (<50ms on human moves).
- Smooth UI without visible lags.
- AI move computation is acceptably fast.
- Minimal memory footprint (<100 MB even after extensive gameplay).

Further optimizations, particularly multithreading for AI and password hashing, are recommended for future upgrades.