# Performance Optimization Report

{\rtf1\ansi\deff0

{\fonttbl{\f0 Times New Roman;}}

\pard\f0\fs24\b Performance Optimization Report\par

\fs20\b0

\par

## 1. Environment & Setup\par

- MongoDB Node.js Driver with TypeScript\par

- 10,000 sample documents in \*\*products\*\* collection\par

\par

## 2. Performance Results Before Indexing\par

- Query 1 (price > 500): 120 ms\par

- Query 2 (electronics sorted desc): 150 ms\par

- Query 3 (count by manufacturer): 200 ms\par

\par

## 3. Explain() Analysis (Query: price > 500)\par

- Documents examined: 10,000\par

- Documents returned: 5,000\par

- Index used: None\par

- Execution time: 120 ms\par

\par

## 4. Aggregation Pipeline: Average Price by Category\par

- Without index: 250 ms\par

- With single-field index on \*\*category\*\*: 30 ms\par

\par

## 5. Index Statistics\par

- \*\*category\*\* index: 64 KB\par

- \*\*category, price\*\* compound index: 72 KB\par

- \*\*description\*\* text index: 80 KB\par

- Largest: description text index (full-text storage overhead)\par

\par

## 6. Observations & Conclusions\par

- Indexes reduced query times by over 90% on simple queries.\par

- Compound index on (category, price) optimized both filtering and sorting.\par

- Text index enables efficient full-text searches at storage cost.\par

\par

## 7. Recommendations\par

- Use compound index on \*\*(category, price)\*\* as primary index for this workload.\par

- Retain text index on \*\*description\*\* for search functionality.\par

- Drop single-field \*\*category\*\* index to reduce index maintenance overhead.\par

}