**Event Ticketing System Using Neo4j Graph Database**

**1. Introduction**

This project develops a ticket booking system using Neo4j to improve how event data relationships (users, seats, venues) are managed. It addresses speed limitations in traditional database systems during high-demand sales.

**2. Background**

Current ticketing platforms (e.g., Eventbrite) use SQL databases that struggle with:

* Complex queries for seat dependencies
* Slow performance during peak sales
* Limited real-time updates

Graph databases like Neo4j optimize relationship-heavy operations, making them ideal for this use case.

**3. Problem Statement**

Relational databases cause:  
 Delays in seat reservation conflicts  
 Inefficient friend-group bookings  
 Poor scalability during high traffic

This project tests if Neo4j can solve these issues.

**4. Objectives**

1. Build a functional ticket booking system with:
   * Real-time seat selection
   * User authentication
   * Payment simulation
2. Compare Neo4j vs. SQL performance
3. Document results for academic use

**5. Methodology**

**Step 1:** Design database schema for:

* Events
* Seats (as nodes with relationships)
* Users

**Step 2:** Develop using:

* **Frontend:** Vue.js/Angular
* **Backend:** Python + FastAPI
* **Database:** Neo4j on Azure

**Step 3:** Test with:

* 100+ simulated users (Locust)
* Query speed comparisons

**6. Resources Needed**

| **Type** | **Items** |
| --- | --- |
| Software | Neo4j AuraDB (free), VS Code, Python 3.x |
| Hardware | Laptop (8GB RAM minimum) |
| Cloud | Azure Free Tier (hosting) |
| APIs | Stripe test mode (payments) |