

# INFOSOFT IT SOLUTIONS

**Training | Projects | Placements**

Revathi Apartments, Ameerpet, 1<sup>st</sup> Floor, Opposite Annapurna Block,

Info soft It solutions, Software Training & Development 905968394,918254087

## **GRAPHQL TRAINING**

### **1: Introduction to Graph QL**

1. Overview of Graph QL
2. History and evolution
3. Graph QL vs REST
4. Key concepts:
  - Schema
  - Queries
  - Mutations
  - Subscriptions
  - Types (Scalar, Object, Enum, Interface, Union)
5. Graph QL ecosystem and tooling

### **2: Setting Up Graph QL Server**

1. Choosing a Graph QL server implementation (e.g., Apollo Server, Express Graph QL, etc.)
2. Environment setup (Node.js, npm/yarn)
3. Initializing a Graph QL project
4. Defining a Graph QL schema
5. Resolver functions
6. Data sources and connectors

### **3: Graph QL Queries**

1. Anatomy of a Graph QL query
2. Basic queries
3. Query variables
4. Fragments
5. Aliases
6. Directives
7. Query best practices and optimization

## **4: Graph QL Mutations**

1. Anatomy of a Graph QL mutation
2. Writing basic mutations
3. Input types
4. Nested mutations
5. Mutation best practices and optimization

## **5: Graph QL Subscriptions**

1. Introduction to real-time updates with Graph QL subscriptions
2. Setting up subscriptions in a Graph QL server
3. Subscribing to events from the client
4. Handling subscription payloads
5. Subscription best practices and optimization

## **6: Authentication and Authorization in Graph QL**

1. Authentication methods (JWT, OAuth, etc.)
2. Middleware for authentication
3. Role-based access control
4. Authorization directives
5. Securing subscriptions

## **7: Error Handling and Testing**

1. Error handling in Graph QL
2. Error formatting and responses
3. Unit testing Graph QL schema and resolvers
4. Integration testing Graph QL APIs
5. Mocking Graph QL server for testing purposes

## **8: Advanced Graph QL Concepts**

1. Batch fetching and Data Loader
2. Pagination
3. Caching strategies
4. Federation and schema stitching
5. Performance optimization techniques

## **9: Graph QL in Practice**

1. Case studies of Graph QL implementations in real-world applications
2. Best practices for designing Graph QL APIs
3. Tooling for Graph QL development (Graph QL Playground, Apollo Client Dev Tools, etc.)
4. Monitoring and debugging Graph QL APIs
5. Handling versioning and deprecation

## **10: Graph QL and Frontend Integration**

1. Consuming Graph QL APIs from client-side applications
2. Graph QL client libraries (Apollo Client, Relay, etc.)
3. Querying data in React/Vue/Angular applications
4. Managing local state with Graph QL
5. Real-time updates in client applications using subscriptions

## **11: Deploying Graph QL Applications**

1. Deployment strategies (Heroku, AWS, Docker, etc.)
2. Containerization of Graph QL servers
3. Continuous integration and deployment (CI/CD) for Graph QL applications
4. Scaling Graph QL servers
5. Monitoring and performance optimization in production

## **1: Advanced Schema Design**

1. Schema design patterns for complex domains
2. Advanced type definitions (interfaces, unions, custom scalars)
3. Schema-first vs. code-first approaches
4. Extending and customizing the Graph QL schema
5. Schema stitching and federation in distributed architectures

## **2: Performance Optimization**

1. Advanced resolver optimization techniques
2. Caching strategies for Graph QL APIs
3. Data Loader and batch data fetching
4. Query complexity analysis and rate limiting
5. Monitoring and profiling Graph QL performance

### **3: Authorization and Security**

1. Role-based access control (RBAC) with Graph QL
2. Fine-grained authorization with custom directives
3. JWT authentication and OAuth integration
4. Securing Graph QL subscriptions
5. Best practices for securing Graph QL APIs in production

### **4: Real-time GraphQL**

1. Implementing real-time updates with Graph QL subscriptions
2. WebSocket protocols and connections
3. Pub/sub systems for real-time data synchronization
4. Scaling real-time Graph QL applications
5. Handling error scenarios and retries in real-time updates

### **5: Advanced Query Techniques**

1. Dynamic query generation and composition
2. Advanced filtering and sorting options
3. Pagination strategies for large datasets
4. Nested mutations and atomic updates
5. Implementing custom query directives

### **6: Federation and Microservices**

1. Understanding Graph QL federation
2. Implementing federated Graph QL schemas
3. Microservices architecture with Graph QL
4. Service discovery and routing in federated setups
5. Best practices and challenges of federated Graph QL

### **7: Graph QL Tooling and Ecosystem**

1. Advanced tooling for Graph QL development
2. Graph QL schema management and versioning
3. Mocking Graph QL APIs for testing
4. Code generation for Graph QL clients
5. Performance monitoring and tracing tools

### **8: Advanced Error Handling and Testing**

1. Error handling strategies in Graph QL
2. Custom error formatting and responses
3. Unit testing Graph QL resolvers and schemas
4. Integration testing Graph QL APIs
5. Mocking and stubbing Graph QL servers for testing

## **9: Graph QL Internals**

1. Understanding the Graph QL execution process
2. Query parsing and validation
3. Execution lifecycle of Graph QL queries
4. Introspection and schema introspection queries
5. Customizing GraphQL execution with middleware

## **10: Graph QL in Production**

1. Deployment strategies for production Graph QL APIs
2. Horizontal and vertical scaling of Graph QL servers
3. High availability and fault tolerance
4. Monitoring and alerting strategies
5. Handling schema changes and deprecation in production

## **11: Graph QL Governance and Best Practices**

1. Implementing Graph QL governance processes
2. Versioning and deprecation strategies
3. Documentation and API discovery best practices
4. Collaboration and communication in Graph QL teams
5. Reviewing and auditing Graph QL APIs