GraphQL Developer – 100+ Lab Exercises Roadmap

☐ Basic Level (30+ Exercises)

Objective: Gain foundational knowledge of GraphQL concepts, schema design, and basic query/mutation operations.

Core Concepts & Setup

- Understand GraphQL basics: queries, mutations, subscriptions.
- Set up a GraphQL server using Node.js with Apollo Server or Express-GraphQL.
- Write simple queries and mutations against a sample schema.
- Explore GraphiQL or Playground for API testing.

Schema Design & Types

- Define GraphQL types: Scalars, Objects, Enums, Input types.
- Implement basic schema with Query and Mutation types.
- Use arguments in queries and mutations.
- Understand and implement nullable vs non-nullable fields.

Resolvers

- Write resolver functions for fields.
- Understand resolver chaining and delegation.
- Handle nested queries and relationships between types.

Data Fetching & Integration

- Connect resolvers to mock or in-memory data sources.
- Implement CRUD operations on sample datasets.
- Practice error handling in resolvers.

☐ Intermediate Level (40+ Exercises)

Objective: Advance skills in schema complexity, security, and integrating GraphQL with databases and other services.

Advanced Schema Design

- Use Interfaces and Unions for flexible schemas.
- Implement custom scalar types.
- Practice schema stitching and schema federation.

Performance & Pagination

- Implement efficient data fetching with batching (DataLoader).
- Apply pagination techniques (cursor-based, offset-based).
- Practice caching strategies at resolver level.

Authentication & Authorization

- Implement authentication middleware (JWT, OAuth).
- Apply field-level and type-level authorization.
- Handle role-based access control (RBAC).

Integrations

- Connect GraphQL API to relational (PostgreSQL, MySQL) and NoSQL (MongoDB) databases.
- Use ORMs like Prisma or Sequelize with GraphQL.
- Integrate REST APIs with GraphQL schema (wrapping REST in GraphQL).

Testing & Tooling

- Write unit and integration tests for GraphQL resolvers.
- Use Apollo Engine or other monitoring tools.
- Document GraphQL API with tools like GraphQL Voyager.

☐ Advanced Level (40+ Exercises)

Objective: Master enterprise-grade GraphQL architecture, real-time features, and ecosystem contributions.

Real-Time & Subscriptions

- Implement GraphQL subscriptions using WebSockets.
- Handle real-time updates in client and server.

Schema Governance & Evolution

- Manage schema versioning and backward compatibility.
- Use schema directives for advanced logic.
- Design schema for multi-tenant applications.

Security & Best Practices

- Prevent common attacks (query complexity, injection).
- Apply rate limiting and depth limiting to queries.
- Secure subscriptions and real-time channels.

Scaling & Performance

- Architect GraphQL APIs for horizontal scaling.
- Implement distributed tracing and logging.
- Use persisted queries and CDN caching.

Ecosystem & Community

- Contribute to GraphQL open source projects.
- Develop custom Apollo Server plugins or directives.
- Build client apps with Apollo Client, Relay, or URQL.

Capstone Projects

- Build a full-stack app with React + Apollo Client + GraphQL server.
- Design a microservices architecture with GraphQL federation.
- Implement a real-time collaborative app using subscriptions.
- Develop a multi-tenant SaaS API with advanced auth and data isolation.

Recommended Tools & Libraries

- Apollo Server & Apollo Client
- Express-GraphQL
- Prisma ORM, Sequelize
- DataLoader (batching)
- GraphiQL, GraphQL Playground
- Hasura (GraphQL engine over PostgreSQL)
- Relay, URQL (GraphQL clients)