Quarkus Exercises Proposal

Quarkus Advantage-Focused Exercises

**1. Dev Mode & Live Reload (Developer Joy)**

* **Task:** Build a simple GET /hello endpoint.
* Then change the message *while the app is running* (mvn quarkus:dev).
* Show how the change is reflected immediately — no restart needed.
* *Advantage:* Instant feedback loop vs. slow restarts in other frameworks.

**2. Configuration Profiles**

* **Task:** Inject a @ConfigProperty (like greeting.message).
* Add different messages for dev, test, and prod profiles.
* Switch profiles without code changes.
* *Advantage:* Seamless environment management for microservices/cloud.

**3. Database with Panache ORM**

* **Task:** Create a Fruit entity and REST CRUD endpoints.
* Use Panache’s active record style (Fruit.listAll(), Fruit.persist()) instead of boilerplate repositories.
* *Advantage:* Less boilerplate than JPA, faster developer productivity.

**4. REST Client + External API**

* **Task:** Call a public API (e.g., JSON Placeholder) with Quarkus REST Client.
* Show simple interface-based client with annotations (no big frameworks).
* *Advantage:* Easy, type-safe HTTP calls without tons of config.

**5. Startup Time & Native Compilation**

* **Task:** Build app as native (mvn package -Pnative).
* Measure startup time & memory (compare to JVM mode).
* *Advantage:* Lightning-fast boot & small footprint, perfect for serverless/Kubernetes.

**6. Reactive Streaming (Optional Advanced)**

* **Task:** Create an endpoint that streams a live event (like a counter or random numbers) using Multi.
* *Advantage:* Reactive programming out of the box, with minimal code.

**Presentation Tie-ins**

For each exercise, you can highlight the **Quarkus value-add**:

* **Dev Mode:** Quarkus makes developers faster.
* **Config:** Built-in cloud-native patterns.
* **Panache ORM:** Reduced boilerplate, cleaner domain logic.
* **REST Client:** Quarkus integrates modern microservice tooling smoothly.
* **Native:** Key selling point → startup in milliseconds, low memory.
* **Reactive:** Ready for modern async workloads.

*Presentation plan:*

1. **What is Quarkus(slide) and Why Use It(slide)?(T)**
   * Supersonic Subatomic Java.
   * Cloud-native, container-first.
2. **Developer Joy: Quarkus Dev Mode(T)**
   * Hot reload and continuous testing.
   * Live coding demo.
3. **Quarkus in the Cloud(T)**
   * Optimized for Kubernetes/OpenShift.
   * Native image benefits.
4. **Building Microservices with Quarkus(M)**
   * RESTEasy, Panache, CDI, Config.
   * How Quarkus simplifies microservices development.
5. **Reactive Programming with Quarkus** (if team is into modern async stuff).**(M)**
6. **Quarkus with Hibernate(M)**
7. **Compare** with Spring Boot Different services.

* Advantages
* Disadvantages

1. **Best Practices** for Quarkus **(T)**
2. **Limitation/Drawbacks). (M)**
3. **Quick quiz**
4. Referencess!!!