

Technical Assessment – Senior Developer (.NET / Azure)

Um deine technischen Fähigkeiten besser einschätzen zu können, bitten wir dich, die folgenden beiden Aufgaben auszuarbeiten (Beide Aufgaben sind verpflichtend)

Aufgabe 1: Architektur & Domain Design – Monolith zu Microservices

Stell dir vor, unser bestehendes Hotel-Buchungssystem ist als klassischer Monolith aufgebaut. Es umfasst Module wie:

- Kundenverwaltung
- Buchung- & Zimmer-Verwaltung
- Zahlungen
- Rechnungsstellung

Teil A – Microservice-Architekturentwurf:

Bitte beschreibe, wie du diesen Monolithen in eigenständige Microservices aufteilen würdest. Gehe dabei auf folgende Punkte ein:

- Welche Microservices ergeben sich aus deiner Sicht? (Begründung)
- Wie kommunizieren die Services miteinander? (REST, Messaging, Eventing – synchron/asynchron)
- Wie stellst du Datenkonsistenz sicher (z. B. Buchung ↔ Zahlung)?
- Welche Infrastruktur-Komponenten würdest du einsetzen? (API Gateway, Message Broker, Service Registry, etc.)
- Wie würdest du Authentifizierung und Autorisierung realisieren?

Teil B – Domain Design eines ausgewählten Microservices:

Wähle einen der von dir vorgeschlagenen Microservices aus und entwerfe dessen fachliches Domänenmodell im Sinne von Domain Driven Design.

Beschreibe:

- Die wichtigsten Entities, Value Objects und den Aggregate Root
- Mögliche Domain Events
- Ein Beispiel für einen typischen Use Case / Command Handler
- Erstelle: ein UML-Klassendiagramm oder Codebeispiel zur Veranschaulichung

Aufgabe 2: Cloud & DevOps – Bereitstellung in Azure

Dein Architekturentwurf aus Aufgabe 1 soll in Microsoft Azure produktiv betrieben werden.

Deine Aufgabe:

Beschreibe deinen bevorzugten Deployment-Ansatz in Azure:

- Welche Azure-Dienste würdest du einsetzen (AKS, App Services, Cosmos DB, Redis etc.)?
- Wie würdest du CI/CD implementieren? (Azure DevOps, Docker, GitOps, IaC etc.)
- Wie würdest du Monitoring, Logging und Alerting aufbauen? (Application Insights, Azure Monitor etc.)
- Wie stellst du sicher, dass das System skalierbar, ausfallsicher und wartbar ist?



Hinweise

- Abgabe als PDF, Markdown oder GitHub-Repository.
- Diagramme und Codebeispiele sind willkommen.
- Wir legen Wert auf Klarheit, sauberes Denken und gute Kommunikation.