A. Take-Home Exercise (2 Days)

1. API Design & Implementation

a. Requirements:

i. Build a small "Book Review" service with:

1. GET/books (list all books)

2. POST/books (add a new book)

3. GET/books/{id}/reviews

4. POST/books/{id}/reviews

ii. Use Python + FastAPI (or Flask) and TypeScript + NestJS (or Express), picking one for implementation.

iii. Include OpenAPI/Swagger documentation (or GraphQL schema).

2. Data Modeling & Persistence

a. Requirements:

i. Use PostgreSQL/SQLite with an ORM (SQL Alchemy, TypeORM, etc.).

ii. Design and implement migrations.

iii. Add an index on the reviews table to optimize fetching reviews by book.

3. Integration & Error Handling

a. Requirements:

i. Integrate with a mock external cache (e.g. Redis).

ii. On fetching/books, first attempt to read from cache; if miss, read DB and populate cache.

iii. Demonstrate proper error handling if the cache is down.

4. A4. Automated Tests

a. Requirements:

i. Write unit tests for at least two endpoints.

ii. Write an integration test covering the cache-miss path.utomated Tests.

Deliverables:

- A Git repository (public or private link) containing instructions in a README file.
- Clear documentation detailing how to run the service, migrations, and tests.
- A Live Walk-Through video Demo

B. live walk through (5 minutes)

Code Review:

a. The candidate explains their key design decisions, chosen patterns, error handling structure, and performance considerations of their code.

Whiteboard/Design Question:

b.The candidate is prompted to sketch how they would extend a service to support GraphQL subscriptions for new reviews in real-time. This includes discussing schema changes, authentication, and scaling concerns.

EVALUATION CRITERIA

Code Quality & API Design: Emphasizes readability, modularity, separation of concerns, RESTful conventions, semantic HTTP status codes, and documentation.

Data Modeling & Error Handling: Looks for appropriate schema, indexing, migrations, ORMs, robustness, fallback strategies, and clear error messages.

Testing & Collaboration: Evaluates test coverage, meaningful assertions, CI friendliness, clarity in presentations,and the ability to justify choices.