Practical Task: QA Testing

Objective

The task will focus on testing the functionality, performance, data accuracy, error handling, and security of some of the [CoinGecko](#) APIs. The outcomes will evaluate the candidate's skills based on the approaches they use. A basic report summarizing key results, findings, and any recommendations will be sufficient.

Scope

- **Functionality**: Verify API endpoints work as expected.
- **Performance**: Measure response times and system load.
- **Data Accuracy**: Check if the data returned is correct.
- **Error Handling**: Validate proper error messages for invalid requests.
- **Security**: Identify vulnerabilities (e.g., SQL injection, unauthorized access).

List of API Endpoints to Test

- **Simple Price**: `GET /simple/price`
- **Coin Markets**: `GET /coins/markets`
- **Coin List**: `GET /coins/list`

Tools to Use

- **Scripting language:** Javascript/Typescript
- **Manual Testing**: Postman or similar
- **API automated Testing**: Postman/Newman or similar
- **Performance Testing**: K6 or similar

Test Plan

1. Setup

- Obtain API keys from coingecko by signing up to a free account.
- Set up testing environments (Development, Staging, Production) in Postman for reusability.

2. Manual Testing

- Create functional test cases for one endpoint.

3. API automated Testing

- Write test scripts for one endpoint using Postman.
- Execute tests using **Newman** and integrate with any **CI/CD** tool.
- Analyze results: Pass/fail status, response times, anomalies.

4. Performance Testing

- Write non-functional test scripts for one endpoint.
- **Load Testing**: Simulate multiple users using **K6** or similar tools.
- **Stress Testing**: Gradually increase load to find API's breaking point.
- Analyze: Response times, error rates, and system behavior.

5. Security Testing (Nice to have)

- Write security test scripts for one endpoint.
- Check for vulnerabilities (e.g., SQL injection).
- Test access control (authentication/authorization).
- Ensure sensitive data is encrypted.

API Endpoints Details

1. **Simple Price Endpoint**

   - **Parameter**: `ids=ethereum, vs_currencies=usd`
   - **Description**: Retrieve the current price of Ethereum (ETH) in USD.
   - **Expected Result**: `{"ethereum": {"usd": value}}`
   - **Method**: Automated/Manual

2. **Coins List Endpoint**

   - **Description**: Fetch the list of all available coins.
   - **Expected Result**: List of coins with `id`, `symbol`, and `name`.
   - **Method**: Automated/Manual

3. **Coin Markets Endpoint**

   - **Parameter**: `vs_currency=usd`
   - **Description**: Get market data for a specific coin.
   - **Expected Result**: Market data with `id`, `symbol`, `current_price`, etc.
   - **Method**: Automated/Manual

Sample Report Structure

1. **Test Scripting**
   - Description of testing methods
   - Tools and environments used
2. **Test Execution**
   - Summary of test cases executed
   - Results from automated and manual tests
3. **Performance Analysis**
   - Results from load and stress testing
4. **Security Analysis**
   - Summary of security tests performed
   - Identified vulnerabilities and mitigation steps
5. **Deliverables**
   - To provide the test scripts and reports in a zip file or via a Git repository.