Part 2: General Programming Questions

Algorithmic Problem (10 points):

● Implement a function in a language of your choice that checks if the given string is a palindrome.

Language: Python

main.py

```python
given_string = input("Enter a word: ")

reverse_string = ""

for each in given_string:
    reverse_string = each + reverse_string

if(given_string == reverse_string):
    print("The given string is a palindrome")
else:
    print("The given string is not palindrome")
```

```
Enter a word: malayalam
The given string is a palindrome


...Program finished with exit code 0
Press ENTER to exit console.
```

Database Concepts (5 points):
● Explain the difference between SQL and NoSQL databases. Provide examples of use cases for each.

SQL Database – Structured Query Language are the relational databases that are stored in tables in rows and columns format. In this type of database, data is organized in tables with predefined schemas.

Examples: MySQL, Oracle, and PostgreSQL

This type of database is majorly used in the **banking sector** to manipulate transaction data.

NO SQL Database – Not Only Structured Query Language are the non-relational databases where the schema is not fixed. In this type of database, data is organized in the object format.

Example: Mongo DB, Cassandra, and Redis

This database is majorly used in big data applications which are highly dynamic in nature.

Web Technologies (5 points):
● What is CORS, and how does it work? How can you handle CORS-related issues in a web application?

CORS – Cross Origin Resource Sharing helps the web pages to make requests to a particular domain. It prevents unauthorized requests

When a web page hosted on one domain makes a request to another domain (via XML, HTTP Request, Fetch API, or other methods), the browser enforces the Same-Origin Policy, which restricts such requests. CORS is a mechanism that allows servers to specify who can access their resources.

Some of the CORS-related issues that can be handled are:

1. **Client-Side Configuration:**
    - On the client side, use appropriate APIs (e.g., Fetch API) that automatically handle CORS. For example, the Fetch API will make preflight requests as needed.
2. **Avoiding JSONP (JSON with Padding):**
    - While JSONP is a workaround for CORS, it has security implications and is generally considered outdated. It's better to use modern CORS mechanisms.

Coding Best Practices (5 points):
● Discuss three coding best practices that you consider important for maintaining a clean and efficient codebase.

1. Breaking the code into smaller chunks:
   This helps to debug the code more easily.
   Makes the code more readable and understandable.
2. Functions with minimal arguments:
   Each function should have only one utility so that it can be used anywhere when required.
3. Comments – Comments help to make the code more readable which is very useful when the codebase is very big and complex.