**3.6 Session 6 reflection**

* Look into the following concepts in the context of developing and operating web services that may access one or more databases. Provide, in your own words, a definition for each, together with a brief discussion of what you perceive its relevance to be to our present area of study.

Please clearly cite all sources.

1. CGI.

The Common Gateway Interface, or CGI, is a standard for external gateway programs to interface with information servers such as HTTP servers.

Server that processes data requested, summited, or clicked by user to back and sent a respond back to the web browser. Connect Frontend with Backend

Ref: <https://www.tutorialspoint.com/python/python_cgi_programming.htm>

https://www.oreilly.com/library/view/cgi-programming-on/9781565921689/04\_chapter-01.html

1. ODBC.

Open Database Connectivity (ODBC) is an open standard application programming interface (API) that allows application programmers to access any database.

This store connection information required to DB

Ref: https://support.microsoft.com/en-us/office/administer-odbc-data-sources-b19f856b-5b9b-48c9-8b93-07484bfab5a7

1. Rollback (instead of commit).

Undo the transactions that have not been saved in the database. The command is only used to undo changes since the last COMMIT and returns the database to some previous state.

Ref: https://support.microsoft.com/en-us/office/administer-odbc-data-sources-b19f856b-5b9b-48c9-8b93-07484bfab5a7

* Next, look into the following additional programming languages that are also used with database technologies, sometimes even at back ends of web services. In your response, discuss why you think they were not highlighted as interesting options in the course outline. Also, identify some feasible options to connect to a DBMS with them.

4. C, C++, and C#.

There are the variations of C programming language

C++: C++ is a cross-platform programming language for developing high-performance applications, a portable programming language that may be used to create applications that run on a variety of systems.

C#: is a high-level, object-oriented programming language. Typically sees use in internal or enterprise applications, rather than commercial software.

**Option to connect to a DBMS**

SQLAPI++ is a C++ library (basically a set of header files) for accessing multiple SQL databases (Oracle, SQL Server, DB2, Sybase, Informix, InterBase, SQLBase, MySQL, PostgreSQL, SQLite, SQL Anywhere and ODBC). It is easy to implement and simple.

Ref: <https://csharp-station.com/understanding-the-differences-between-c-c-and-c/>

* <https://www.simplilearn.com/tutorials/cpp-tutorial/c-sharp-vs-cpp>
* https://www.geeksforgeeks.org/database-connectivity-using-cc/

5. Fortran.

Fortran is a computer programming language that is extensively used in numerical, scientific computing.

Use for over six decades in computationally intensive areas such as [numerical weather prediction](https://en.wikipedia.org/wiki/Numerical_weather_prediction), [finite element analysis](https://en.wikipedia.org/wiki/Finite_element_method), [computational fluid dynamics](https://en.wikipedia.org/wiki/Computational_fluid_dynamics), [geophysics](https://en.wikipedia.org/wiki/Geophysics), [computational physics](https://en.wikipedia.org/wiki/Computational_physics), [crystallography](https://en.wikipedia.org/wiki/Crystallography) and [computational chemistry](https://en.wikipedia.org/wiki/Computational_chemistry)

**Option to connect to a DBMS**

**ForDBC :** Fortran programmer can read and write tables of databases using common SQL commands.

ForDBC is based on the ODBC API

Ref: https://en.wikipedia.org/wiki/Fortran

6. Cobol.

COBOL (Common Business-Oriented Language) is a high-level programming language for business applications

**Option to connect to a DBMS**

**DBMS**

Micro Focus COBSQL technology

Micro Focus External Compiler Module (ECM)

This programing language using on backend need more deeply knowledge and based on my read is the base of programming.

* Finally, pick one of the programming languages discussed in this session as alternatives to NodeJS (or one of the three from the previous block of questions, if you really want to) and replicate functionality similar to the example code of Session 5 with that technology and the DBMS of your choice. Host your code on a public repository such as GitHub or similar, together with a setup script or instructions to install and configure the necessary components.

Discuss the following three aspects of your experience.

7. What is the URL of the repository? Did you have any difficulties creating or populating it?

8. Was it easier in some way with the technologies you chose than with NodeJS and SQLite?  
Was anything about it harder? Please discuss what was different and how.

9. If you were to start over with yet another technology pair (programming language and DBMS), which ones would you pick and why? If you in fact experimented with several before choosing one, how did that go?

Python example

A screenshot of a computer

Description automatically generated with medium confidence