

The Number of Functional Features Included In The Solution:

- **Create a Google My Business account:** This feature allows users to create a Google My Business account, which is a free tool that helps businesses manage their online presence across Google
- **Add your business information:** This feature allows users to add their business information to their Google My Business profile, such as their business name, address, phone number, website, and hours of operation.
- **Verify your business:** This feature allows users to verify their business with Google, which helps to ensure that their business information is accurate and up-to-date.
- **Optimize your profile:** This feature allows users to optimize their Google My Business profile for search engines, which can help their business show up higher in search results.
- **Manage your profile:** This feature allows users to manage their Google My Business profile, such as responding to reviews, posting updates, and running promotions.
- **Use Google Posts:** This feature allows users to create and share posts on their Google My Business profile, which can help them to connect with potential customers.
- **Create Google My Business Insights reports:** This feature allows users to create reports that provide insights into their Google My Business performance, such as how many people are viewing their profile and how many people are clicking on their website.

Code-Layout, Readability And Reusability

The code for creating a Google My Business Profile should be well-organized and easy to read. This will make it easier for other developers to understand and modify the code in the future. The code should also be reusable, meaning that it can be used to create other Google My Business Profiles without having to be rewritten from scratch.

Utilization of Algorithms, Dynamic Programming, Optimal Memory Utilization

The code for creating a Google My Business Profile should use efficient algorithms and data structures to minimize the amount of time and memory required to create a profile. Dynamic programming can be used to optimize the code for creating profiles with a large number of fields.

Debugging & Traceability

Debugging is the process of finding and fixing errors in a program. Traceability is the ability to follow the flow of data through a program. These two concepts are closely related, as debugging often involves tracing the flow of data to find the source of an error.

There are many different tools and techniques that can be used for debugging and traceability. Some common debugging tools include:

- **Debuggers:**

Debuggers allow developers to step through their code line by line, set breakpoints, and examine the values of variables.

- **Logging:**

Logging allows developers to print messages to a console or file, which can help them to track the execution of their code.

- **Assertions:**

Assertions are statements that developers can use to check for conditions that should always be true. If an assertion fails, an exception is thrown.

Traceability can be achieved through a variety of techniques, such as:

- **Code instrumentation:** This involves adding code to the program that tracks the flow of data.
- **Static analysis:** This involves analyzing the program code to identify potential problems.
- **Dynamic analysis:** This involves running the program and collecting data about its execution.

Debugging and traceability are essential for developing high-quality software. By using these techniques, developers can find and fix errors more quickly and easily, and they can also ensure that their code is reliable and performance

Exception Handling

Exception handling is a programming technique that allows developers to handle errors gracefully. When an error occurs, an exception is thrown. The developer can then catch the exception and handle it in a way that prevents the program from crashing or causing other problems.

There are three main parts to exception handling:

- **Throwing an exception:**
 - This is the act of notifying the program that an error has occurred.
- **Catching an exception:**
 - This is the act of intercepting an exception and handling it.
- **Handling an exception:**
 - This is the act of taking action to deal with an exception.

There are many different ways to handle exceptions. Some common ways to handle exceptions include:

- Logging the exception
- Displaying an error message to the user
- Trying to recover from the error