**Project Development Phase**

**Utilization Of Algorithms, Dynamic Programming, Optimal Memory Utilization**

|  |  |
| --- | --- |
| Date | 3 November 2023 |
| Team Id | NM2023TMID03725 |
| Project Name | |  | | --- | | Create a Google My Business profile | |  | |

**Utilization Of Algorithms:**

Google My Business (GMB) is a free tool that businesses can use to manage their online presence across Google Search and Maps. By creating and optimizing a GMB profile, businesses can make it easier for potential customers to find them online, and learn more about their products or services.

Google makes use of several algorithms to display GMB profiles in Search and Maps. Some of the most important factors that influence how and where a GMB profile is displayed include:

* Relevance: How relevant is the GMB profile to the user's search query?
* Distance: How close is the GMB profile to the user's location?
* Prominence: How well-known and established is the business?
* Reviews: How many positive reviews does the business have?
* Completeness: How complete is the GMB profile?

**Dynamic Programming:**

Dynamic programming is a technique for solving problems by breaking them down into smaller, subproblems that have already been solved. This can be useful for creating a Google My Business profile for a soap shop because it allows you to reuse code and avoid recomputing the same results multiple times.

One way to use dynamic programming to create a Google My Business profile is to create a function that takes the soap shop's name, address, and phone number as input and returns a Google My Business profile object. This function can then use dynamic programming to cache the results of each subproblem, so that they only need to be computed once.

**Optimal Memory Utilization:**

To minimize memory usage when creating a Google My Business profile for a soap shop, you should use the following techniques:

* Use a data structure that is efficient for storing the profile data. For example, you could use a dictionary to store the profile's attributes.
* Avoid creating unnecessary copies of the profile data. When passing data between functions, pass references to the data instead of copying it.
* Use a caching mechanism to store frequently accessed data. This can help to reduce the number of database queries that you need to perform.