**Discovery Script**

**Functional Approach**:

The objective of this project is to find out all the objects on the GitHub side, including projects, repositories, branches, and files. To achieve this goal, we can follow the below functional approach:

1. Identify the GitHub account and access rights: The first step is to identify the GitHub account, for which we want to find out the objects. We also need to ensure that we have the necessary access rights to access the account and its associated repositories.
2. Accessing GitHub API: GitHub provides a REST API that can be used to retrieve information about the account and its repositories. We need to use the API to retrieve the required information.
3. Retrieving Projects: To retrieve all the projects, we need to use the Projects API endpoint. We can iterate over all the pages of the endpoint to retrieve all the projects.
4. Retrieving Repositories: To retrieve all the repositories, we need to use the Repositories API endpoint. We can iterate over all the pages of the endpoint to retrieve all the repositories.
5. Retrieving Branches: To retrieve all the branches for a repository, we need to use the Branches API endpoint. We can iterate over all the repositories and their branches to retrieve all the branches.
6. Retrieving Files: To retrieve all the files for a repository, we need to use the Contents API endpoint. We can iterate over all the repositories and their files to retrieve all the files.
7. Saving the information: Once we have retrieved all the required information, we can save it in a file for further analysis.

**Technical Approach:**

To implement the above functional approach, we can use the following technical approach:

1. Authentication: We need to authenticate with the GitHub API to access the required information. We can use an access token or a personal access token (PAT) to authenticate with the API.
2. Retrieving Projects: We can use the Projects API endpoint (<https://api.github.com/projects>) to retrieve all the projects. We can use the requests library to make a GET request to the endpoint and retrieve the JSON response. We can iterate over all the pages of the response until we have retrieved all the projects.
3. Retrieving Repositories: We can use the Repositories API endpoint (<https://api.github.com/user/repos>) to retrieve all the repositories for a user account. We can use the requests library to make a GET request to the endpoint and retrieve the JSON response. We can iterate over all the pages of the response until we have retrieved all the repositories.
4. Retrieving Branches: We can use the Branches API endpoint ([https://api.github.com/repos/{owner}/{repo}/branches](https://api.github.com/repos/%7Bowner%7D/%7Brepo%7D/branches)) to retrieve all the branches for a repository. We can use the requests library to make a GET request to the endpoint and retrieve the JSON response. We can iterate over all the repositories and their branches to retrieve all the branches.
5. Retrieving Files: We can use the Contents API endpoint ([https://api.github.com/repos/{owner}/{repo}/contents/{path}](https://api.github.com/repos/%7Bowner%7D/%7Brepo%7D/contents/%7Bpath%7D)) to retrieve all the files for a repository. We can use the requests library to make a GET request to the endpoint and retrieve the JSON response. We can iterate over all the repositories and their files to retrieve all the files.
6. Saving the information: We can save the retrieved information in a file. For example, we can save the information in a CSV file or a JSON file for further analysis.
7. Error Handling: We need to handle errors and exceptions while retrieving the information from the GitHub API. We can use try-except blocks to catch any exceptions and log them for debugging purposes.