**To Fetch CSV File in GitHub Program:-**

**SRS (Software Requirements Specifications):-**

**1. System Overview:-**

The purpose of this document is to outline the software requirements for a program that reads a CSV file containing repository and user information, interacts with the GitHub API to fetch information about each repository, processes the data, and generates an output CSV file with the required information. The program will also create folders as needed and log events to a log file.

**2. Functional Requirements:-**

**Input Validation:**

* The program should validate the input CSV file to ensure it is in the correct format.
* The input CSV file should contain a header row with the column names "repo\_name" and "user\_name".

**GitHub API Interaction:-**

* The program should send requests to the GitHub API for each repository and user pair specified in the input CSV file.
* The program should handle successful and unsuccessful API responses appropriately.
* If the API response is successful (HTTP status code 200), the program should extract the relevant information from the response.
* If the API response is unsuccessful, the program should log a warning and continue processing the next repository and user pair.

**Output Generation:-**

* The program should generate an output CSV file with the required information.
* The output CSV file should contain a header row with the column names "repo\_name", "user\_name", "full\_name", and "description".
* For each valid repository and user pair, the program should extract the full name and description from the GitHub API response and write them to the output CSV file.

**Folder Creation:-**

* The program should create the output folder specified by the user if it does not already exist.
* If the output folder already exists, the program should proceed without modifying the existing folder.

**Logging:-**

* The program should log every event to a logfile.
* The logfile should be located in the output folder.
* The program should log events at different severity levels, such as info, warning, and error.

**3. Non-functional Requirements:-**

**Usability:-**

* The program should provide clear and meaningful error messages if there are any issues with the input or during the API interactions.
* The program should handle a large number of repository and user pairs efficiently.

**Reliability:-**

* The program should handle API failures gracefully and continue processing other repository and user pairs.
* The program should handle unexpected errors or exceptions and log them appropriately.

**Performance:-**

* The program should be able to fetch information from the GitHub API in a reasonable amount of time.
* The program should process the CSV file and generate the output CSV file efficiently.

**Portability:-**

* The program should be implemented in a portable programming language, such as Python, and should run on different operating systems.

**4. Constraints:-**

* The program is dependent on an internet connection to interact with the GitHub API.
* The user should have appropriate access permissions to read the input CSV file, create folders, and write the output CSV file.

**5. Assumptions:-**

* The input CSV file provided by the user will be correctly formatted with the required columns.
* The user will have the necessary permissions and credentials to interact with the GitHub API.
* The GitHub API responses will be in the expected format.