Pine Labs
Summer Internship

Project Report on

Log Monitor Application

By Adit Goyal

Under the Guidance of

Ms. Shalini Agrawal

ABSTRACT

Systems, both client and server, generate a huge number of events, and it's incredibly easy for the useful information to be completely lost in the signal to noise ratio; there's quite a lot of noise. Too often important information can be lost in the sea of superfluous errors without the help of management software to sift through it all. On the software-focused side of things, event logging is incredibly useful when applications just aren't cooperating with the user.

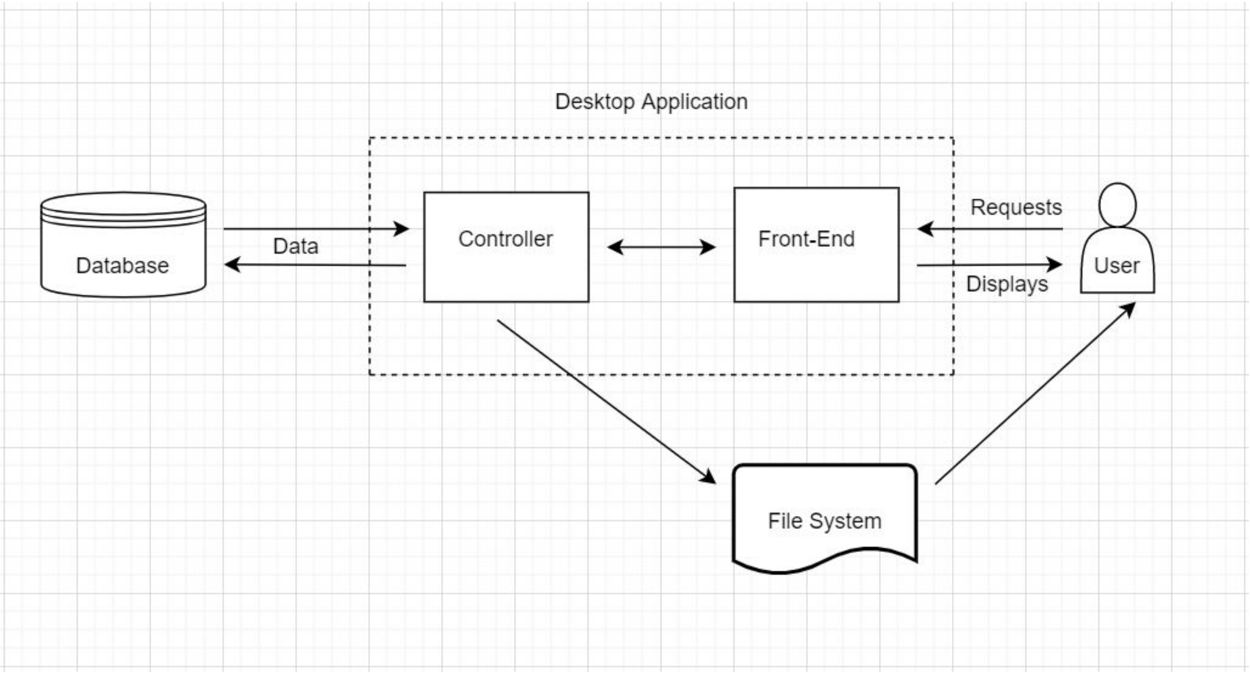
Ultimately event logs tend to be just too unwieldy and time consuming to peruse in their raw state. That's where Event Log Management makes any technician's life, and job, easier – software that can quickly, intelligently, and reliably make the proverbial needle in a haystack search far easier while simultaneously monitoring.

It can view logs and view patterns in the event data as well as query through the data to get the required results.

INTRODUCTION

Log monitoring system provides a solution for log analysis, and reporting. It gives us the means to act with speed as soon as possible when threats, errors, or oversights are detected. This feature is an essential cornerstone of IT security. This application provides means for advanced search options for quick locating and sorting of pertinent data. The collection of logs from multiple sources, which can be unified into a single universal log resource can further be used for trend analysis by data scientists to predict future system failures before they occur. Easier creation of more comprehensive troubleshooting methods and helping to differentiate between common false positives and actual bugs and similar problems can be very useful to analyze and report the data properly.

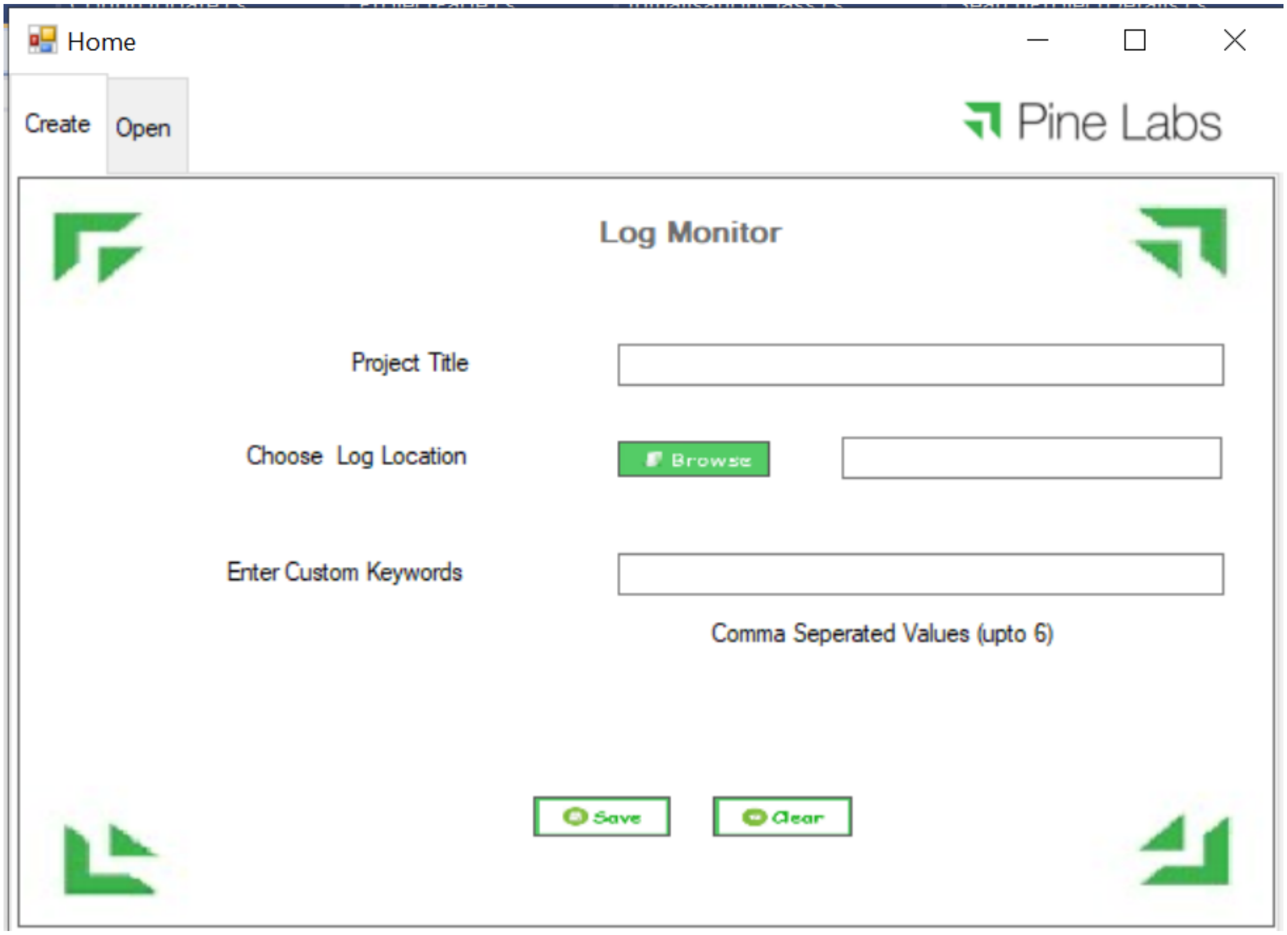
Project Architecture



The user requests data from application through the Windows Application Form (Front-End) which processes the code from the Back-End (Controller) that interacts with the Database Storage through SQL Server Linq Queries and makes the changes as requested in the tables and if needed, writes the search results in an External File-System for user to access in the future for further analysis and the form window Displays the Required Output to the user.

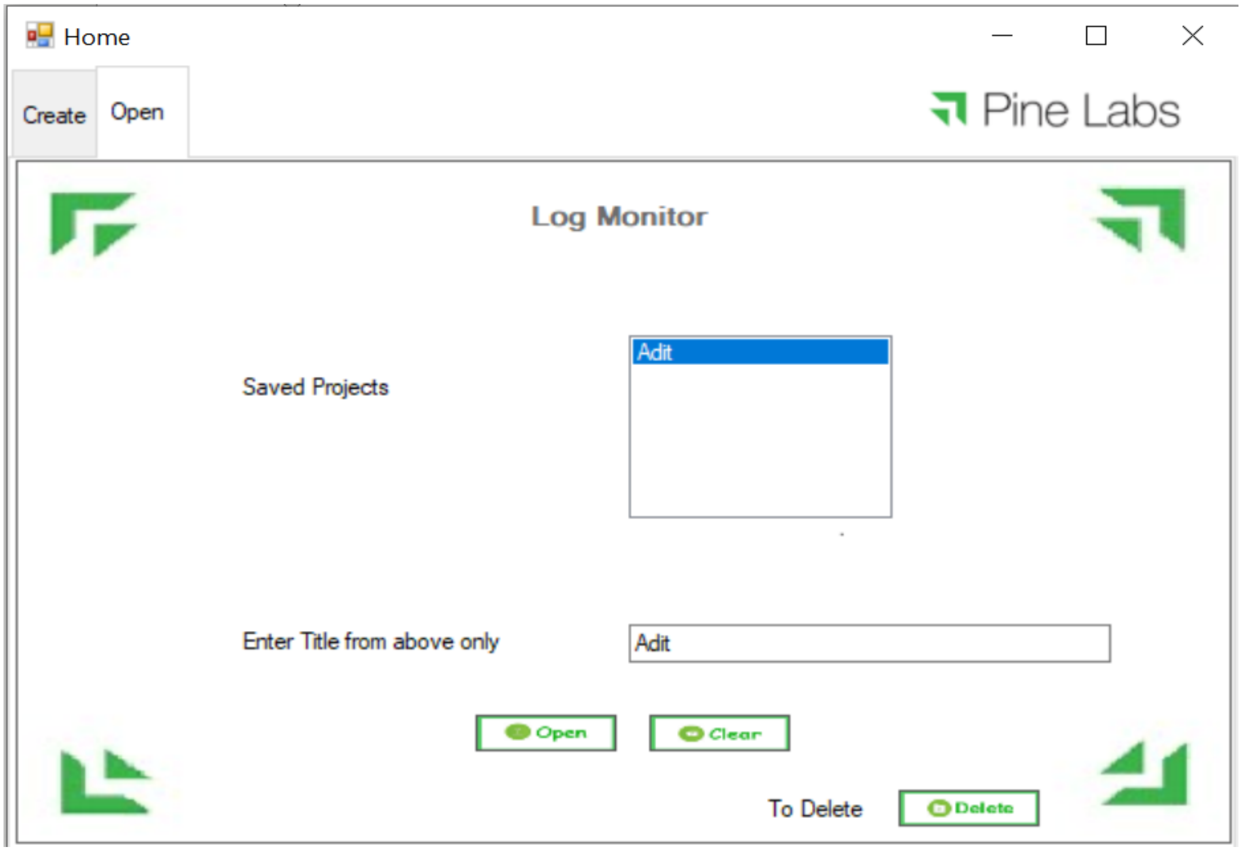
OVERVIEW and APPROACH USED

1. Home Page
    a. Create Tab
        i. Project Title
        ii. Folder Location for Log Reading
        iii. Comma separated values(up to 6) for custom keywords to perform search



b. Open Tab

   i. Fetch details from SQL server database and display all available projects
   ii. Select the one user wants to open
   iii. Delete the selected project from the records

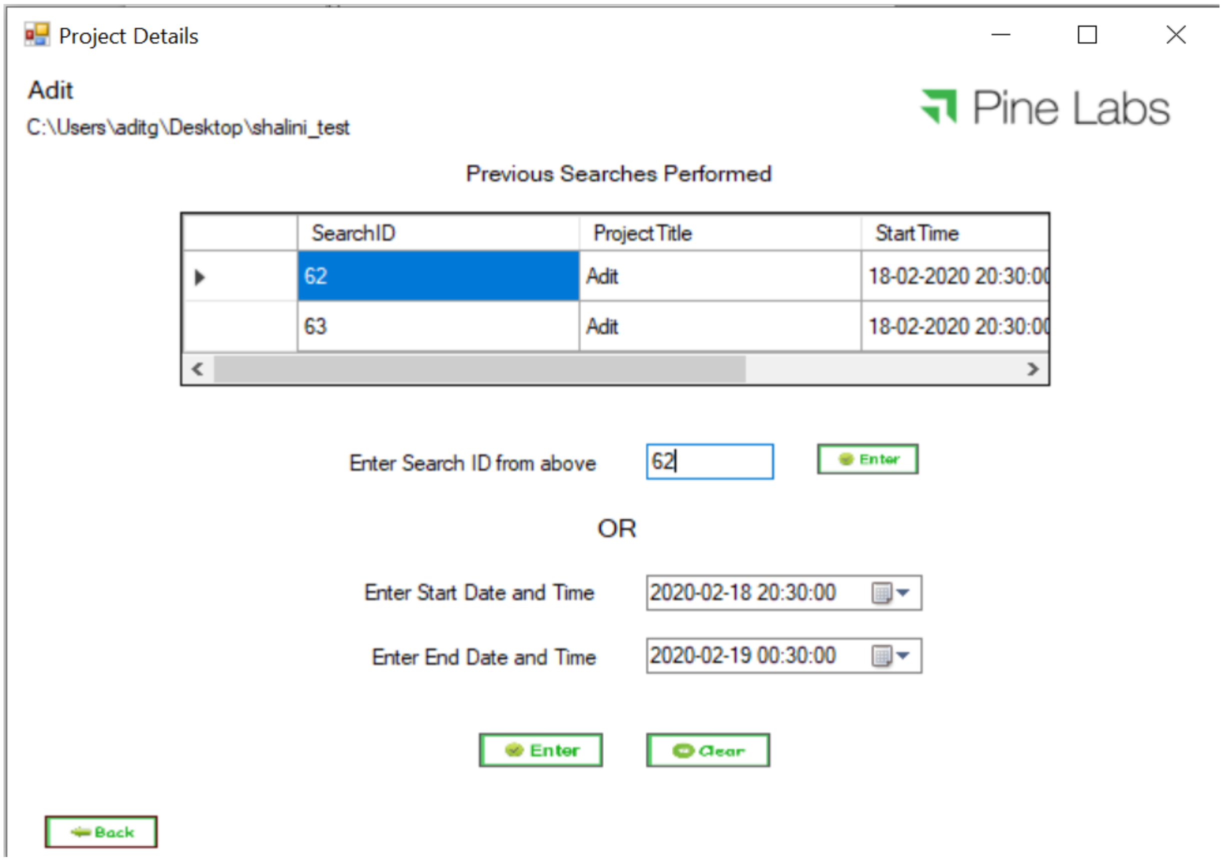


Saved Projects: Adit

Enter Title from above only: Adit

2. Project Page

a. Project Title
b. Log Folder Location
c. Select from all previous searches for this project or an option to perform a new search.
d. Go back to the previous page



Adit

C:\Users\aditg\Desktop\shalini_test

Previous Searches Performed

	SearchID	Project Title	Start Time
▶	62	Adit	18-02-2020 20:30:0
	63	Adit	18-02-2020 20:30:0

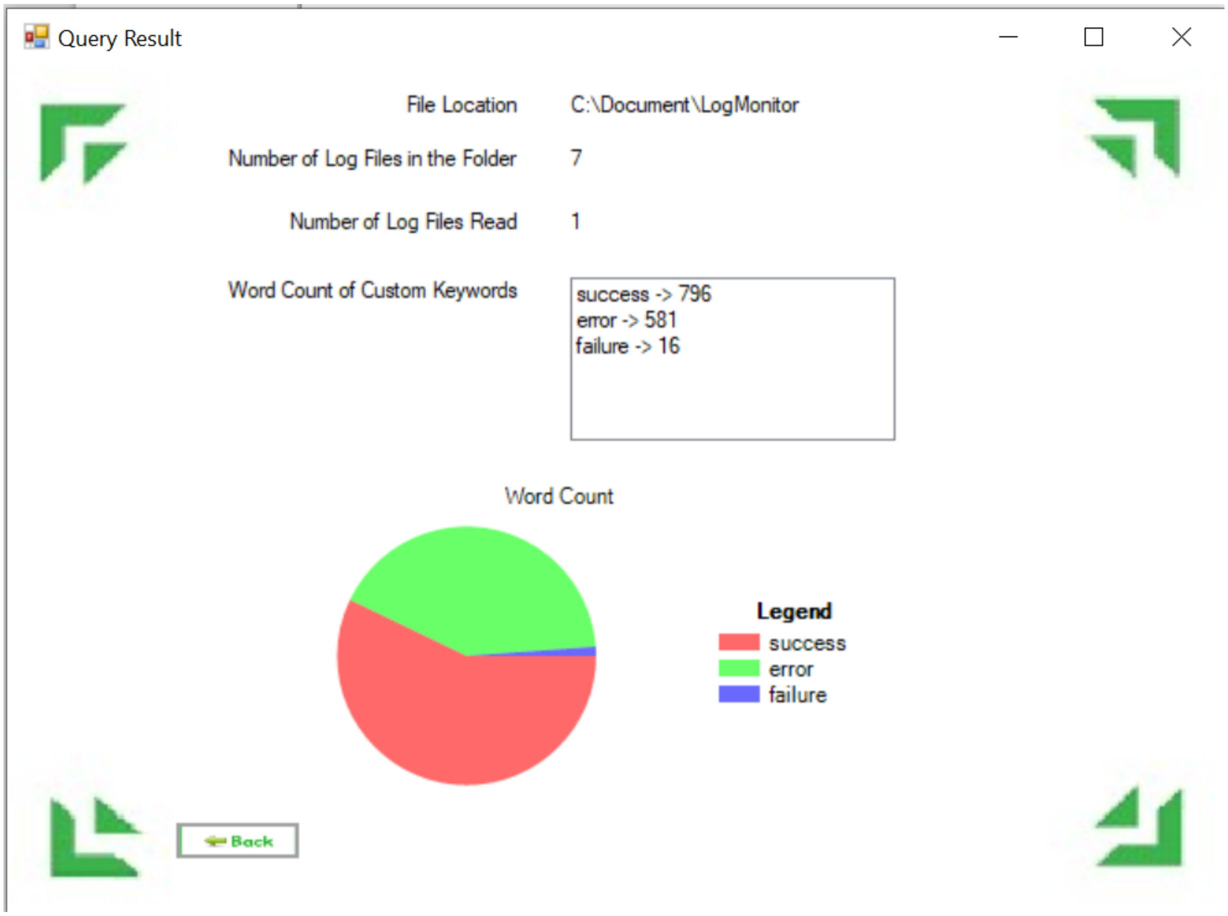
Enter Search ID from above: 62

OR

Enter Start Date and Time: 2020-02-18 20:30:00

Enter End Date and Time: 2020-02-19 00:30:00

3. Result Page
    a. Perform the search for a new search or for an old search, fetch the details from the database
    b. Location where file is created
    c. Files Present in the Log Folder
    d. Files Read according to the Time frame set by the user
    e. Word count and write relevant information in the file
    f. Pie chart to show percentage of each word count

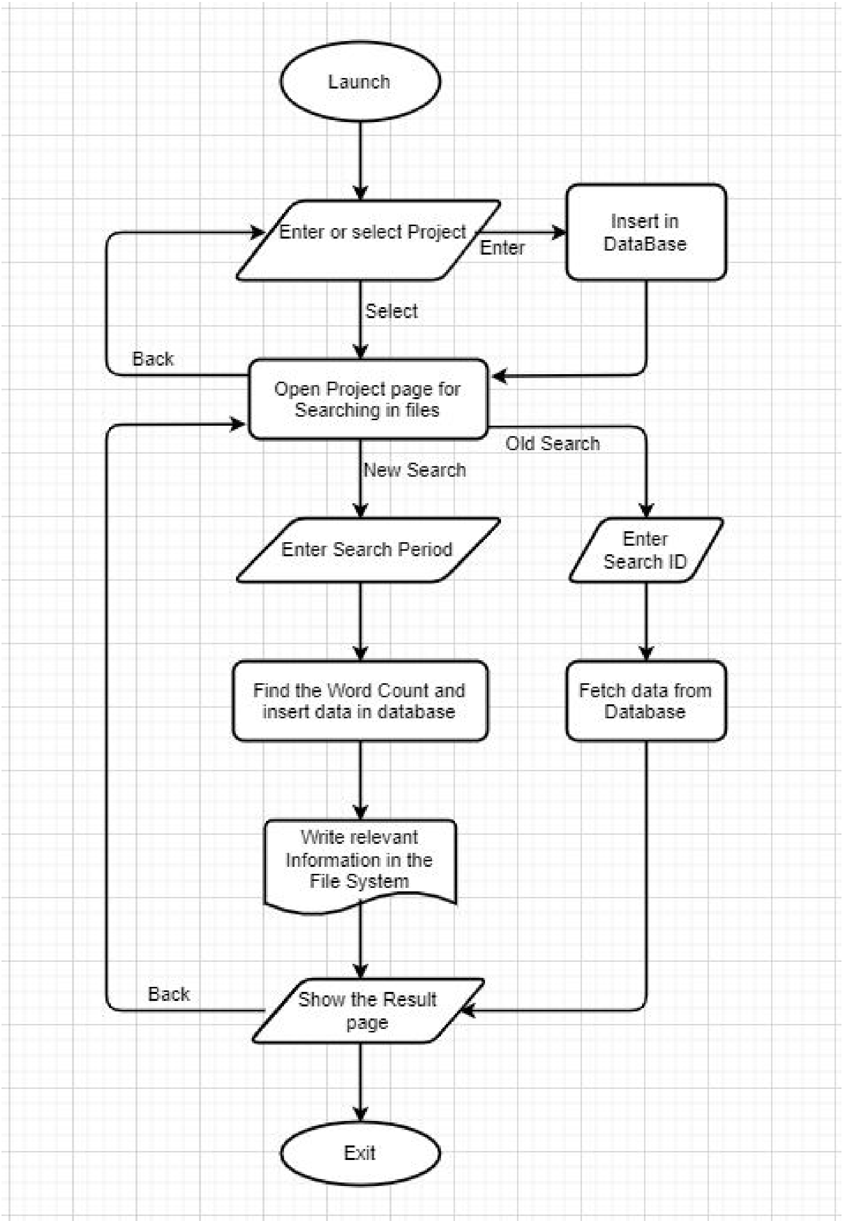


Query Result

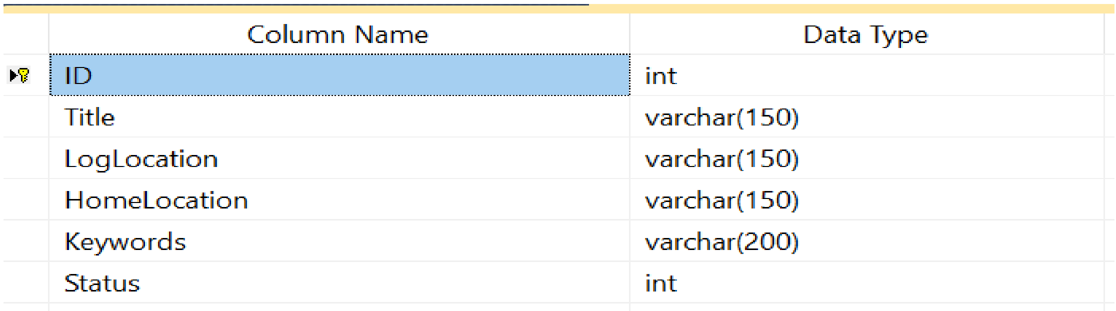
File Location	C:\Document\LogMonitor
Number of Log Files in the Folder	7
Number of Log Files Read	1
Word Count of Custom Keywords	success -> 796 error -> 581 failure -> 16

4. Auto-Delete: Delete Files and records from the Database and system after 30 days and load the Home Page.

Application Flowchart

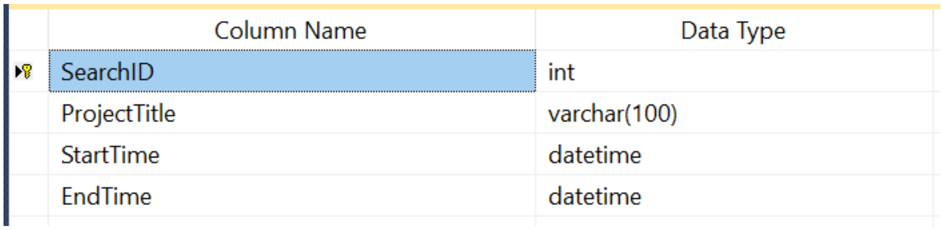


DATABASE AND FILE-SYSTEM



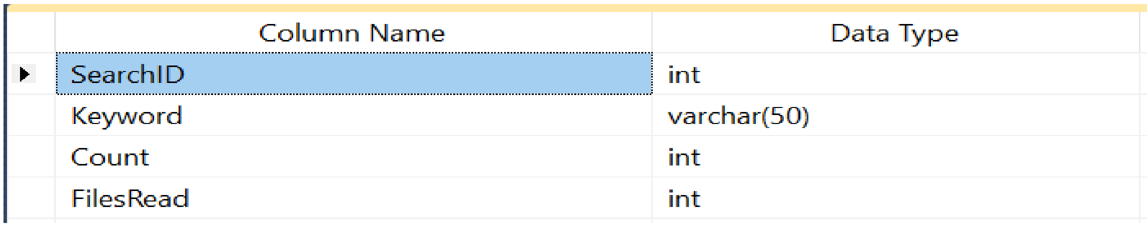
	Column Name	Data Type
▶🔑	ID	int
	Title	varchar(150)
	LogLocation	varchar(150)
	HomeLocation	varchar(150)
	Keywords	varchar(200)
	Status	int

Projects Table



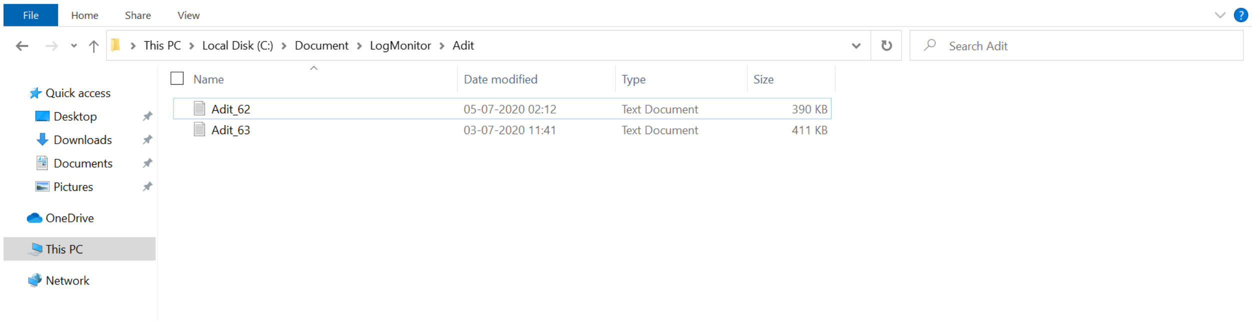
	Column Name	Data Type
▶🔑	SearchID	int
	ProjectTitle	varchar(100)
	StartTime	datetime
	EndTime	datetime

Search History Table



	Column Name	Data Type
▶	SearchID	int
	Keyword	varchar(50)
	Count	int
	FilesRead	int

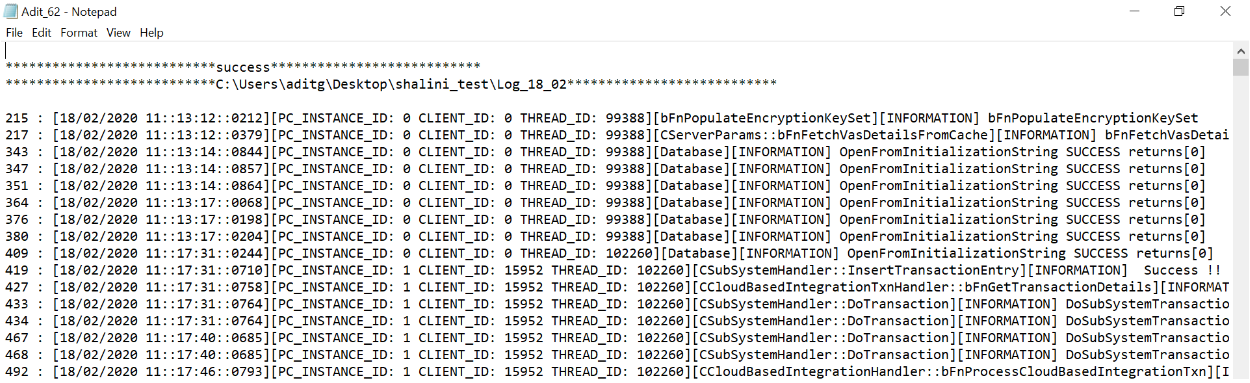
Count History Table



This PC > Local Disk (C:) > Document > LogMonitor > Adit

Name	Date modified	Type	Size
Adit_62	05-07-2020 02:12	Text Document	390 KB
Adit_63	03-07-2020 11:41	Text Document	411 KB

Output Files Generated



```
**************************success**************************
*************************C:\Users\aditg\Desktop\shalini_test\Log_18_02*************************

215 : [18/02/2020 11::13:12::0212][PC_INSTANCE_ID: 0 CLIENT_ID: 0 THREAD_ID: 99388][bFnPopulateEncryptionKeySet][INFORMATION] bFnPopulateEncryptionKeySet
217 : [18/02/2020 11::13:12::0379][PC_INSTANCE_ID: 0 CLIENT_ID: 0 THREAD_ID: 99388][CServerParams::bFnFetchVasDetailsFromCache][INFORMATION] bFnFetchVasDetai
343 : [18/02/2020 11::13:14::0844][PC_INSTANCE_ID: 0 CLIENT_ID: 0 THREAD_ID: 99388][Database][INFORMATION] OpenFromInitializationString SUCCESS returns[0]
347 : [18/02/2020 11::13:14::0857][PC_INSTANCE_ID: 0 CLIENT_ID: 0 THREAD_ID: 99388][Database][INFORMATION] OpenFromInitializationString SUCCESS returns[0]
351 : [18/02/2020 11::13:14::0864][PC_INSTANCE_ID: 0 CLIENT_ID: 0 THREAD_ID: 99388][Database][INFORMATION] OpenFromInitializationString SUCCESS returns[0]
364 : [18/02/2020 11::13:17::0068][PC_INSTANCE_ID: 0 CLIENT_ID: 0 THREAD_ID: 99388][Database][INFORMATION] OpenFromInitializationString SUCCESS returns[0]
376 : [18/02/2020 11::13:17::0198][PC_INSTANCE_ID: 0 CLIENT_ID: 0 THREAD_ID: 99388][Database][INFORMATION] OpenFromInitializationString SUCCESS returns[0]
380 : [18/02/2020 11::13:17::0204][PC_INSTANCE_ID: 0 CLIENT_ID: 0 THREAD_ID: 99388][Database][INFORMATION] OpenFromInitializationString SUCCESS returns[0]
409 : [18/02/2020 11::17:31::0244][PC_INSTANCE_ID: 0 CLIENT_ID: 0 THREAD_ID: 102260][Database][INFORMATION] OpenFromInitializationString SUCCESS returns[0]
419 : [18/02/2020 11::17:31::0710][PC_INSTANCE_ID: 1 CLIENT_ID: 15952 THREAD_ID: 102260][CSubSystemHandler::InsertTransactionEntry][INFORMATION]  Success !!
427 : [18/02/2020 11::17:31::0758][PC_INSTANCE_ID: 1 CLIENT_ID: 15952 THREAD_ID: 102260][CCloudBasedIntegrationTxnHandler::bFnGetTransactionDetails][INFORMAT
433 : [18/02/2020 11::17:31::0764][PC_INSTANCE_ID: 1 CLIENT_ID: 15952 THREAD_ID: 102260][CSubSystemHandler::DoTransaction][INFORMATION] DoSubSystemTransactio
434 : [18/02/2020 11::17:31::0764][PC_INSTANCE_ID: 1 CLIENT_ID: 15952 THREAD_ID: 102260][CSubSystemHandler::DoTransaction][INFORMATION] DoSubSystemTransactio
467 : [18/02/2020 11::17:40::0685][PC_INSTANCE_ID: 1 CLIENT_ID: 15952 THREAD_ID: 102260][CSubSystemHandler::DoTransaction][INFORMATION] DoSubSystemTransactio
468 : [18/02/2020 11::17:40::0685][PC_INSTANCE_ID: 1 CLIENT_ID: 15952 THREAD_ID: 102260][CSubSystemHandler::DoTransaction][INFORMATION] DoSubSystemTransactio
492 : [18/02/2020 11::17:46::0793][PC_INSTANCE_ID: 1 CLIENT_ID: 15952 THREAD_ID: 102260][CCloudBasedIntegrationHandler::bFnProcessCloudBasedIntegrationTxn][I
```

File Content for the search with line No, Keyword, Log File Location for the Places where match was found

The use of database management and file handling will be useful for efficient storage of data and ease of access for addition of further functionalities to make this more resourceful and detail-oriented for the user.

THANK YOU