Mini Project-2 Report

System Overview:-

The system is a MongoDB - NoSQL based where we are responsible of build and operating a document store built using python language. It supports a single user who can view basic information based on the tasks assigned to work on. We start off by making a Document Store with 2 argument inputs, which will take in a json format file and our local port number as an argument. This phase would be responsible to connect us to the server and generate a database named 291db. We also create a new collection named articles.

Now to operate the Document Store, we write various lines of code for the user to carry out the following tasks: Search the most commonly used words from the type of media. Count the number of news & blog articles and which media has the most articles, with its associated difference, all depending on the date inputted by the user. Search for the top 5 news sources by taking into consideration the most articles published in 2015. Lastly, the latest articles with a limit and tie of 5.

User Guide:-

The system offers one user role accessed through a simple menu interface.

Options Page

Users can directly enter the menu of options and exit the program if wanted

Design of the Software:-

The architecture of the system has a display layer, application layer and the data layer. Users interact though a command line menu interface, while all data is stored in the database.
At the presentation layer, the main options menu acts as the entry point. It displays the initial choices given to the users like 4 options assigned to work on from the phase 2. This routes them to the appropriate functions. All user inputs and outputs are handled here with no direct database access.

The application layer contains the main logic of the system(i.e. Functional modules) and manages all user operations. It includes carrying out search, authenticating the date entered, and basic comparison with the items from the database Together these modules ensure that all user actions are correctly working through the application logic before interacting with the database.

All functions use a shared database, which stores information into a singular table with dictionary storage of its associated data, which is accessed through parameterized NoSQL queries to maintain integrity, and enforce relationships among the functions.

Primary functions with their responsibilities, interface and the structure and relationships among them:-

UserOptions () – This function serves as the starting point of the program. It displays the main menu, allowing users to use other functions for basic search or potentially exiting the entire program.

Common_words() – This function handles the top 5 words found in the news or blog section while also displaying the total number of those common words found in the article.

Article_Count() –This function is responsible for counting the number of articles that have been published between News and Blog in terms of the date provided by the user. This displays the number of articles for news or blog, and which media type has more articles, with its count.

News_Sources() – This function works to access the top 5 (with tie) news sources with the most articles published during the year 2015

Most_Recent_Articles() – This function allows the user to retrieve the top 5 most recent articles published from the source provided by the user

Testing Strategy:-

We adopted a functional and database-level testing approach to verify both user interface behaviors and backend data integrity. We tested all functions for invalid inputs such as leaving fields empty and pressing enter, entering alphanumeric or special characters and checked for case insensitivity in the program. Some examples of scenarios where we did the above testing would be:-

Scenarios being Tested	Coverage of Test Cases
All input functions	Test various alphanumeric, date formats, keywords, upper/lower case words & letters, numbers, and quick repetitive Enter key presses to test the boundaries of our program and output appropriate feedback.

However during all this heavy testing, we faced many bugs like the search function not working accurately, date format entered incorrectly without providing a feedback and crashing, not displaying tied number of articles, and many more which was debugged and fixed moving forward.

Group Work Strategy:-

Person	Work Item	Estimated Time spent	Progress

Henil	Phase 1 Phase 2 part 1 & 2	20 hr	Created the entire functionality for Phase 1 and the starting parts for Phase 2. Worked on handling errors for the functions previously written
Shaz	Phase 2 part 3 & 4 Report Writing	15 hr	Completed Phase 2 with the final bits, while also debugging and fixing errors. Completed the Report

For communication purposes, we mostly used discord. We used to meet 2 times a week after classes for active discussions and brainstorming (time decided mutually across via text) in the library. We mainly organized and tracked our tasks through github. After completion, we tested the full program as a group so that we could check for any errors and update the functions accordingly for final submission.