TECHGIG

Nielson – Store Transaction Imputation

Jishan Shaikh
jishanshaikh9893@gmail.com
Hackathon - Presentation

Team Details

- **Team Name: Jishan Shaikh**
- **Number of team members: 1**
- **Team Members: Jishan Shaikh.**
- **Delieverables:**
  - Source_code (+ input and output files)
  - CSV File of output data
  - Presentation
  - Screenshots
  - Supplements

Problems Addressed

- How to select values from all scaled data inputs?
- Features selection: which features to select, and which not to?
  - Features selected: MONTH, STORE, VALUE, GROUP
- Tools and Technologies
  - C++ (Data manipulation), TXT and CSV (output)
- Result calculation and proper data structure selection
  - Arrays in C++, double for values (for data integrity)
  - File Handling: Taking input/output from/to file as STDIN and STDOUT.

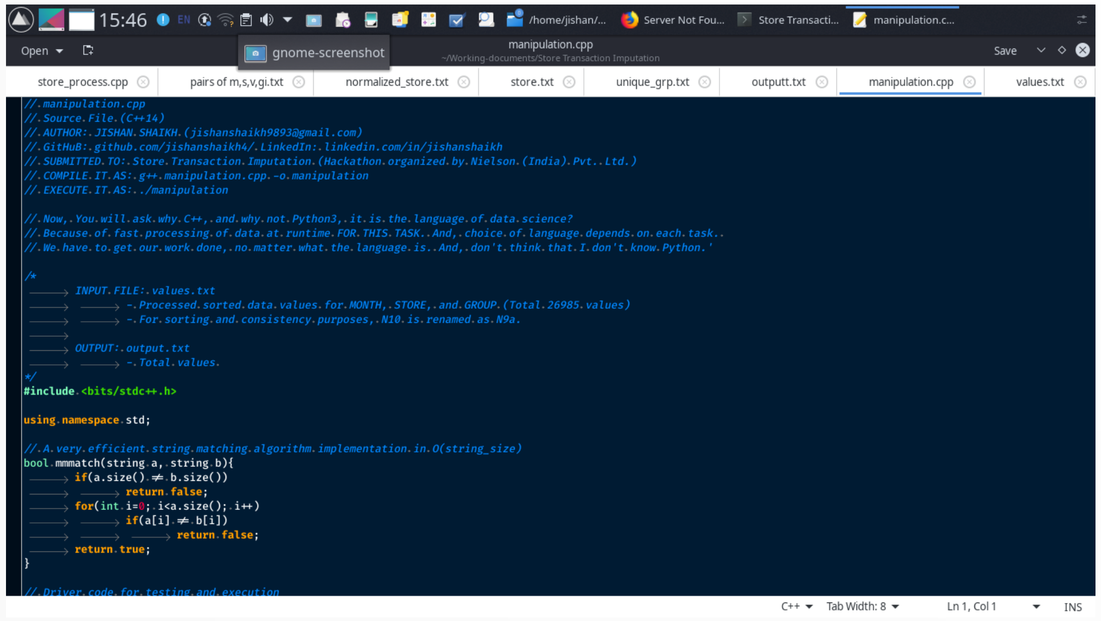
Problem-solving approach

- Data science project life-cycle activities (Order of execution):
  - Preprocessing (To values.txt from working_data.csv for C++)
  - Normalization (e.g. Converting N10 to N9a for convenience)
  - Feature Selection (e.g. SUB_GROUP are abandoned)
  - Result Calculations (e.g. proper data structures and manipulations)
- Used Regression solving approach for handling of small scale data
  - E.g. Replaced 0 values with mean of the complete values

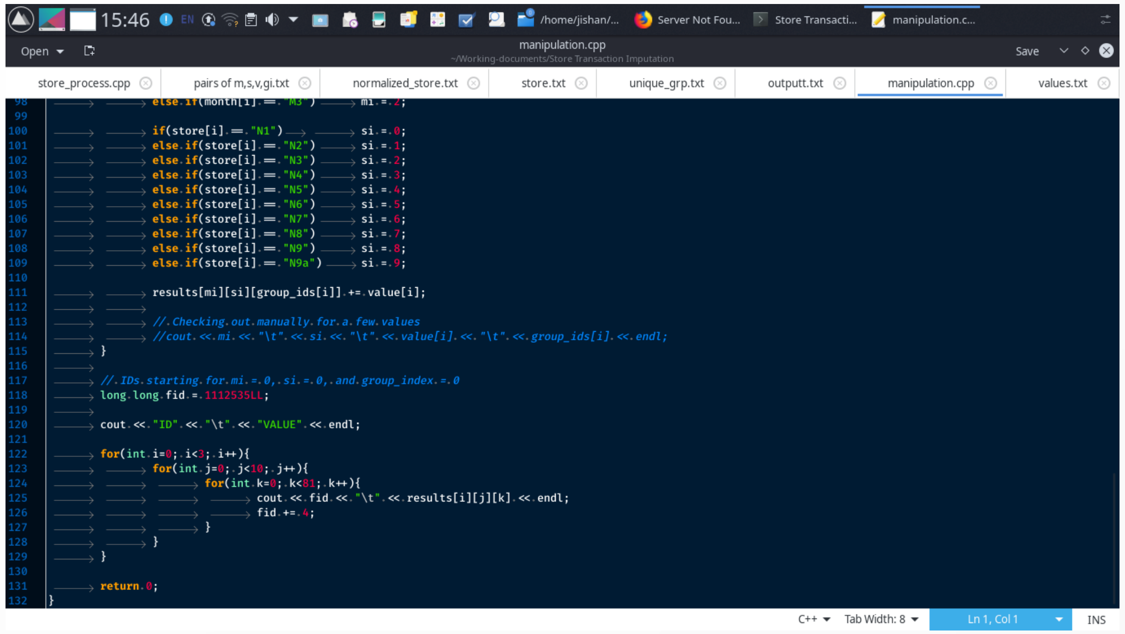
Solution Explanation

- Created custom dataset (values.txt) from working_data.csv
  - Sorted it in order (STORE, MONTH, GROUP) for ease of requirements.
  - Normalized values such as decimal values to double and N10 to N9a.
  - Extracted 81 unique groups from 26985 sample values.
  - Indexed all groups to extracted 81 groups, Months (M1, M2, and M3) to (0, 1, 2), and STORE (0, 1, 2, ... , 9).
  - Calculated result for each pair of [store][month][group] as sum of values for corresponding pairs.
- Provided required output
  - Storing ouput to outputt.txt and then converting it to CSV format.

Screenshots (1 of 5)



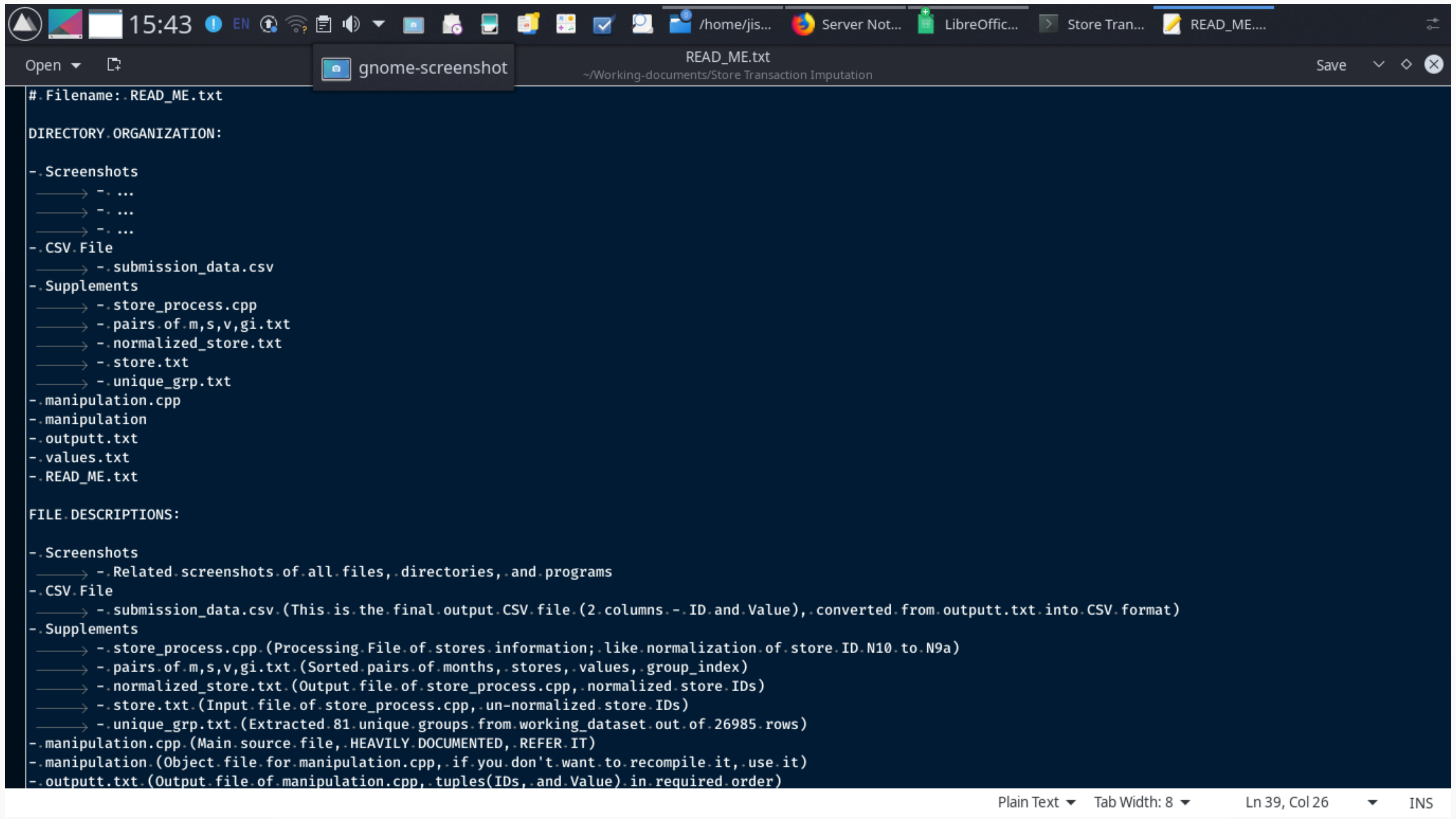
```cpp
// manipulation.cpp
// Source File (C++14)
// AUTHOR: JISHAN SHAIKH (jishanshaikh9893@gmail.com)
// GitHuB: github.com/jishanshaikh4/ LinkedIn: linkedin.com/in/jishanshaikh
// SUBMITTED TO: Store Transaction Imputation (Hackathon organized by Nielson (India) Pvt. Ltd.)
// COMPILE IT AS: g++ manipulation.cpp -o manipulation
// EXECUTE IT AS: ./manipulation

// Now, You will ask why C++, and why not Python3, it is the language of data science?
// Because of fast processing of data at runtime FOR THIS TASK. And, choice of language depends on each task.
// We have to get our work done, no matter what the language is. And, don't think that I don't know Python.'

/*
	INPUT FILE: values.txt
		- Processed sorted data values for MONTH, STORE, and GROUP (Total 26985 values)
		- For sorting and consistency purposes, N10 is renamed as N9a.
	
	OUTPUT: output.txt
		- Total values.
*/
#include <bits/stdc++.h>

using namespace std;

// A very efficient string matching algorithm implementation in O(string_size)
bool mmmatch(string a, string b){
	if(a.size() ≠ b.size())
		return false;
	for(int i=0; i<a.size(); i++)
		if(a[i] ≠ b[i])
			return false;
	return true;
}

// Driver code for testing and execution
```



```cpp
		else if(month[i] == "M3")	mi = 2;

		if(store[i] == "N1")		si = 0;
		else if(store[i] == "N2")	si = 1;
		else if(store[i] == "N3")	si = 2;
		else if(store[i] == "N4")	si = 3;
		else if(store[i] == "N5")	si = 4;
		else if(store[i] == "N6")	si = 5;
		else if(store[i] == "N7")	si = 6;
		else if(store[i] == "N8")	si = 7;
		else if(store[i] == "N9")	si = 8;
		else if(store[i] == "N9a")	si = 9;

		results[mi][si][group_ids[i]] += value[i];
		
		// Checking out manually for a few values
		//cout << mi << "\t" << si << "\t" << value[i] << "\t" << group_ids[i] << endl;
	}
	
	// IDs starting for mi = 0, si = 0, and group_index = 0
	long long fid = 1112535LL;
	
	cout << "ID" << "\t" << "VALUE" << endl;

	for(int i=0; i<3; i++){
		for(int j=0; j<10; j++){
			for(int k=0; k<81; k++){
				cout << fid << "\t" << results[i][j][k] << endl;
				fid += 4;
			}
		}
	}

	return 0;
}
```

Manipulation.cpp (source-file)

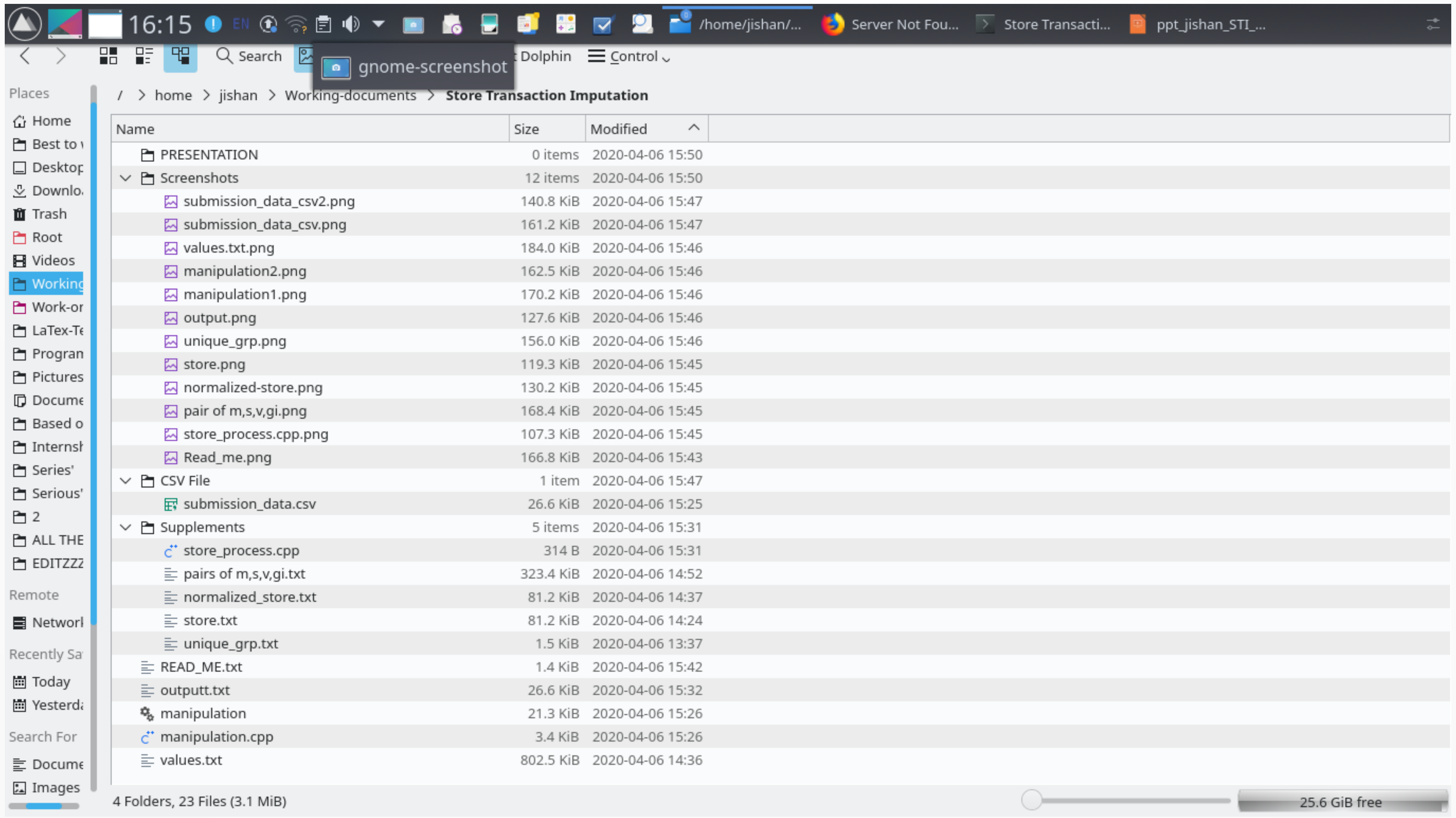
Screenshots (2 of 5)



```
# Filename: READ_ME.txt

DIRECTORY ORGANIZATION:

- Screenshots
        - ...
        - ...
        - ...
- CSV File
        - submission_data.csv
- Supplements
        - store_process.cpp
        - pairs of m,s,v,gi.txt
        - normalized_store.txt
        - store.txt
        - unique_grp.txt
- manipulation.cpp
- manipulation
- outputt.txt
- values.txt
- READ_ME.txt

FILE DESCRIPTIONS:

- Screenshots
        - Related screenshots of all files, directories, and programs
- CSV File
        - submission_data.csv (This is the final output CSV file (2 columns - ID and Value), converted from outputt.txt into CSV format)
- Supplements
        - store_process.cpp (Processing File of stores information; like normalization of store ID N10 to N9a)
        - pairs of m,s,v,gi.txt (Sorted pairs of months, stores, values, group_index)
        - normalized_store.txt (Output file of store_process.cpp, normalized store IDs)
        - store.txt (Input file of store_process.cpp, un-normalized store IDs)
        - unique_grp.txt (Extracted 81 unique groups from working_dataset out of 26985 rows)
- manipulation.cpp (Main source file, HEAVILY DOCUMENTED, REFER IT)
- manipulation (Object file for manipulation.cpp, if you don't want to recompile it, use it)
- outputt.txt (Output file of manipulation.cpp, tuples(IDs, and Value) in required order)
```

READ_ME.txt (README file)

Screenshots (3 of 5)



/ > home > jishan > Working-documents > **Store Transaction Imputation**

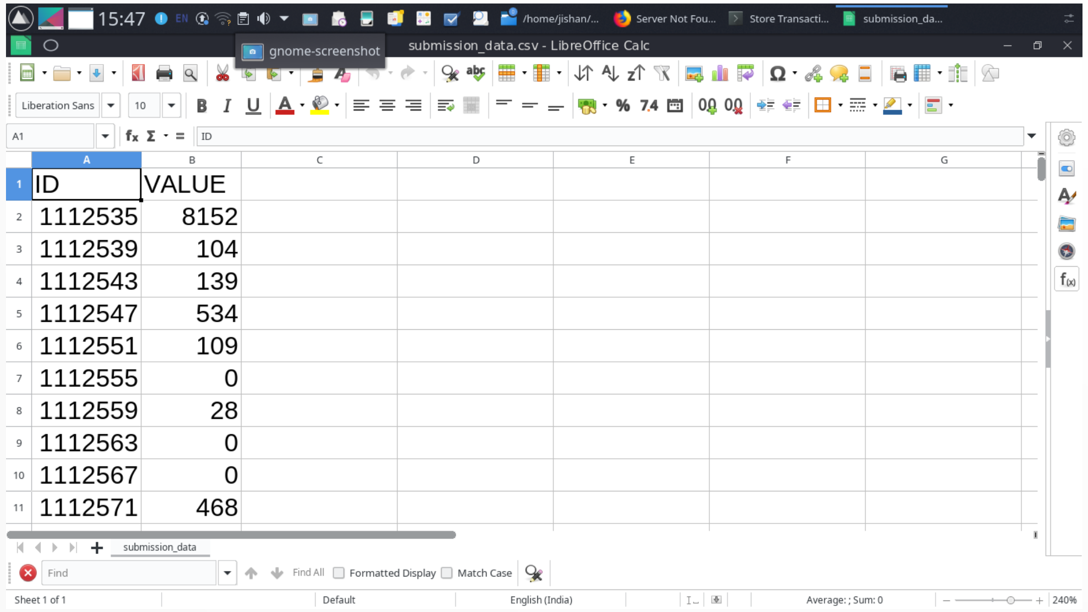
Name	Size	Modified
PRESENTATION	0 items	2020-04-06 15:50
Screenshots	12 items	2020-04-06 15:50
submission_data_csv2.png	140.8 KiB	2020-04-06 15:47
submission_data_csv.png	161.2 KiB	2020-04-06 15:47
values.txt.png	184.0 KiB	2020-04-06 15:46
manipulation2.png	162.5 KiB	2020-04-06 15:46
manipulation1.png	170.2 KiB	2020-04-06 15:46
output.png	127.6 KiB	2020-04-06 15:46
unique_grp.png	156.0 KiB	2020-04-06 15:46
store.png	119.3 KiB	2020-04-06 15:45
normalized-store.png	130.2 KiB	2020-04-06 15:45
pair of m,s,v,gi.png	168.4 KiB	2020-04-06 15:45
store_process.cpp.png	107.3 KiB	2020-04-06 15:45
Read_me.png	166.8 KiB	2020-04-06 15:43
CSV File	1 item	2020-04-06 15:47
submission_data.csv	26.6 KiB	2020-04-06 15:25
Supplements	5 items	2020-04-06 15:31
store_process.cpp	314 B	2020-04-06 15:31
pairs of m,s,v,gi.txt	323.4 KiB	2020-04-06 14:52
normalized_store.txt	81.2 KiB	2020-04-06 14:37
store.txt	81.2 KiB	2020-04-06 14:24
unique_grp.txt	1.5 KiB	2020-04-06 13:37
READ_ME.txt	1.4 KiB	2020-04-06 15:42
outputt.txt	26.6 KiB	2020-04-06 15:32
manipulation	21.3 KiB	2020-04-06 15:26
manipulation.cpp	3.4 KiB	2020-04-06 15:26
values.txt	802.5 KiB	2020-04-06 14:36

4 Folders, 23 Files (3.1 MiB)

25.6 GiB free

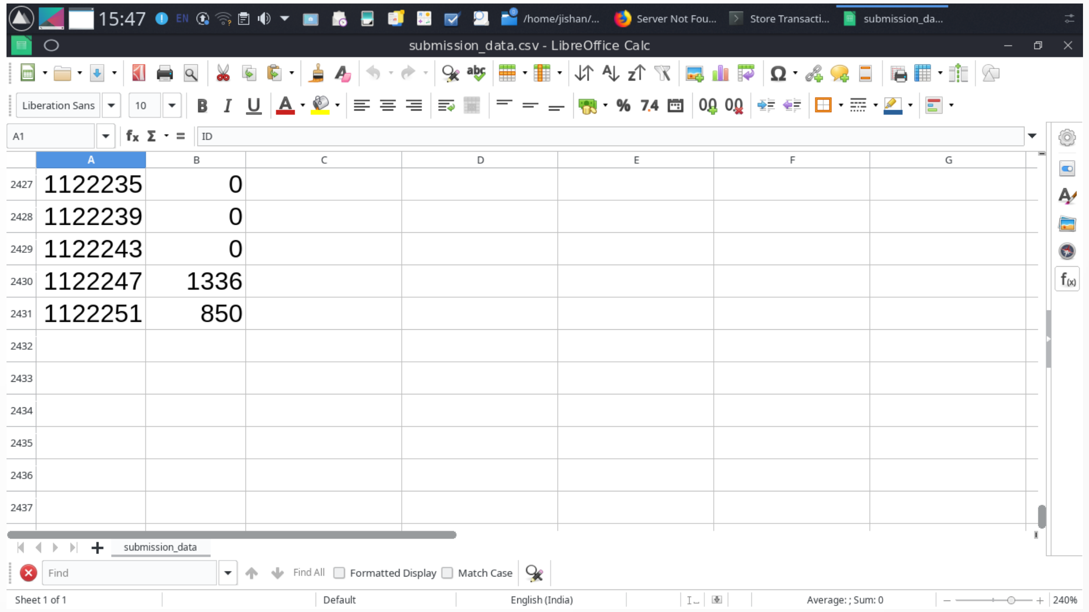
Delieverables’ file organization

Screenshots (4 of 5)



submission_data.csv - LibreOffice Calc

	A	B
1	ID	VALUE
2	1112535	8152
3	1112539	104
4	1112543	139
5	1112547	534
6	1112551	109
7	1112555	0
8	1112559	28
9	1112563	0
10	1112567	0
11	1112571	468

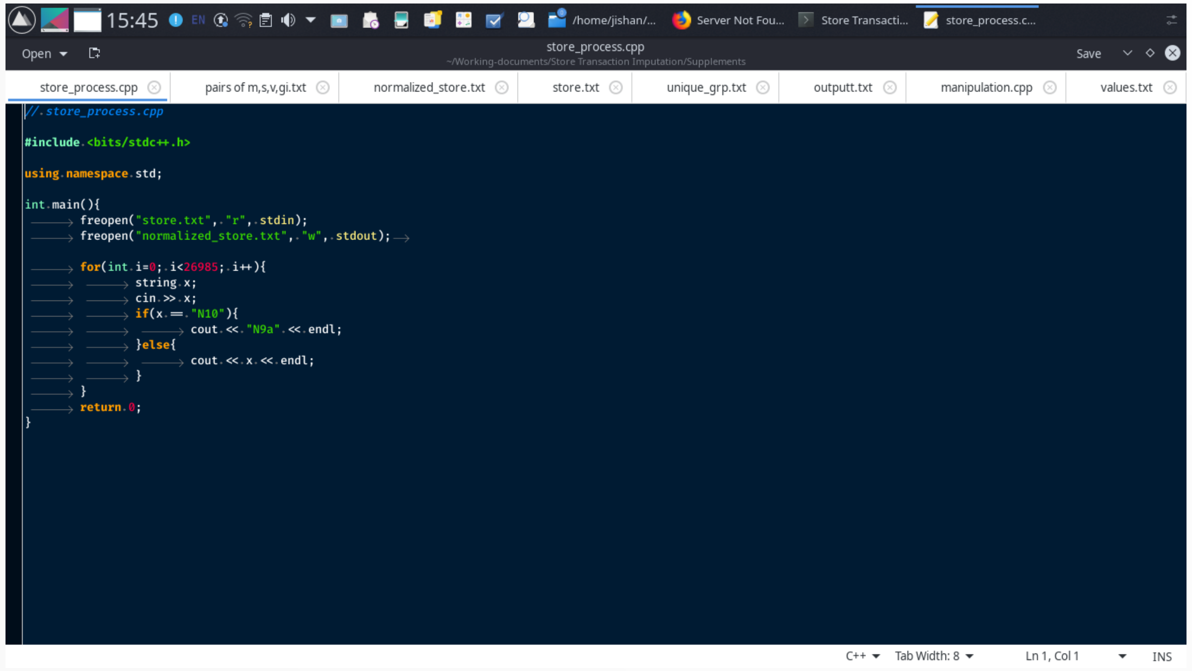


submission_data.csv - LibreOffice Calc

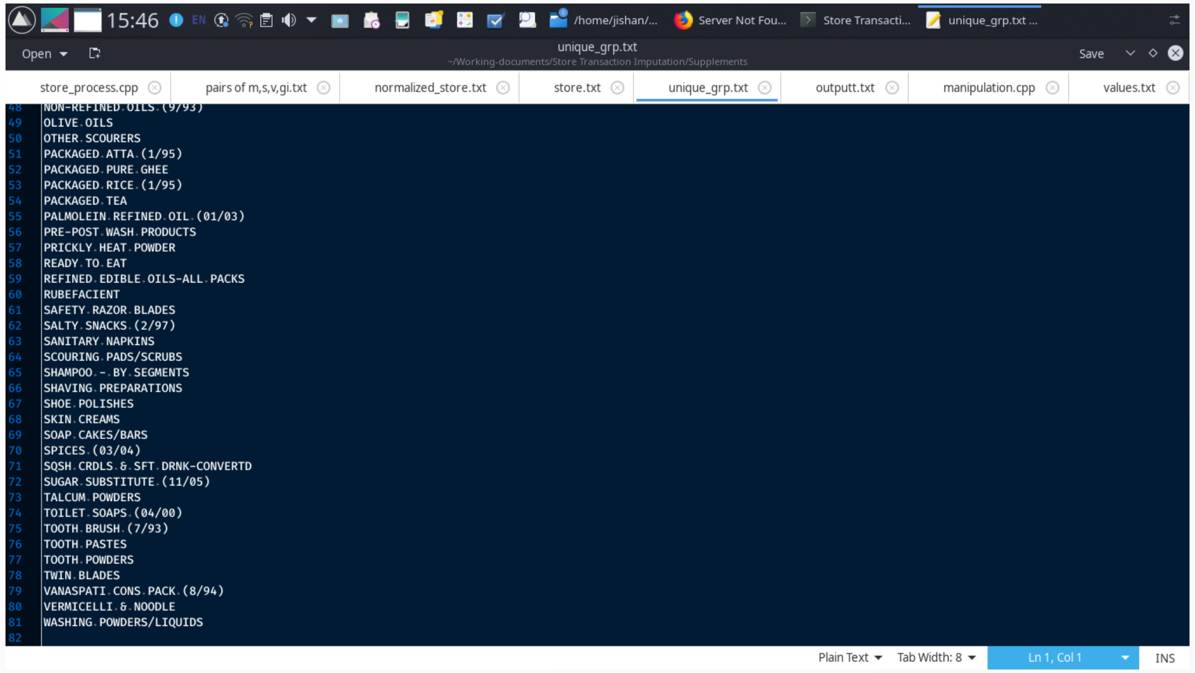
	A	B
2427	1122235	0
2428	1122239	0
2429	1122243	0
2430	1122247	1336
2431	1122251	850
2432		
2433		
2434		
2435		
2436		
2437		

CSV File (submitted_data.csv)

Screenshots (5 of 5)



```cpp
// store_process.cpp

#include <bits/stdc++.h>

using namespace std;

int main(){
	freopen("store.txt", "r", stdin);
	freopen("normalized_store.txt", "w", stdout);	

	for(int i=0; i<26985; i++){
		string x;
		cin >> x;
		if(x == "N10"){
			cout << "N9a" << endl;
		}else{
			cout << x << endl;
		}
	}
	return 0;
}
```



```
NON-REFINED OILS (9/93)
OLIVE OILS
OTHER SCOURERS
PACKAGED ATTA (1/95)
PACKAGED PURE GHEE
PACKAGED RICE (1/95)
PACKAGED TEA
PALMOLEIN REFINED OIL (01/03)
PRE-POST WASH PRODUCTS
PRICKLY HEAT POWDER
READY TO EAT
REFINED EDIBLE OILS-ALL PACKS
RUBEFACIENT
SAFETY RAZOR BLADES
SALTY SNACKS (2/97)
SANITARY NAPKINS
SCOURING PADS/SCRUBS
SHAMPOO - BY SEGMENTS
SHAVING PREPARATIONS
SHOE POLISHES
SKIN CREAMS
SOAP CAKES/BARS
SPICES (03/04)
SQSH CRDLS & SFT DRNK-CONVERTD
SUGAR SUBSTITUTE (11/05)
TALCUM POWDERS
TOILET SOAPS (04/00)
TOOTH BRUSH (7/93)
TOOTH PASTES
TOOTH POWDERS
TWIN BLADES
VANASPATI CONS PACK (8/94)
VERMICELLI & NOODLE
WASHING POWDERS/LIQUIDS
```

Supplements (code and unique_groups)

Thank You

TechGig and Nielson (India)