Final Report

Designing Emulated Distributed File System for exploration of Outbreaks: COVID-19, Ebola, and SARS

Team Members: Tushar Sharma, Laxmi Garde, Raajitha Rajkumar

A. Project Overview

The project provides a study and analysis of impacts of COVID-19, SARS and Ebola outbreaks. The study examines the available datasets and provides relationships between the deaths, geography, recovered cases, and total infected people due to different diseases. The project is based on an emulated distributed file system implemented using Firebase and MySQL, and uses the file system operations implemented and provides these features to the end users.

B. Project Design / Emulated Partition Systems Implementations

The implementation of all operations mentioned in Task 1, Task 2 and Task 3 of the Project have been completed successfully. Details on the Project timelines are mentioned in the Project Tracker (Section C). The implementation details include:

B.1 MySQL Implementation

For our MySQL implementation, we are using two tables created with the following structure:

fid	name	parent	parentid	content	file
1	root		0	/	

pid	fid	tableName	partitionName	part

• mkdir()

Creating a new directory is very simple as we just insert a new row into the directory table after gathering all the column information. After parsing through user input and tracking parent information and determining content information, we just insert it.

```
sql = "INSERT INTO dir (name, parent, parentid, content, file) VALUES (%s, %s, %s, %s, %s)"
val = (name, parent, int(parentid), content, file)
```

• Is()

We are able to grab the children files from a directory by simply tracking the parentid grabbed from the file id where the user put in the directory name.

```
cursor.execute("SELECT fid FROM dir WHERE content LIKE '"+ dirname +"'")
result = cursor.fetchall()
dirid = result[0][0]
sql = "SELECT name FROM dir WHERE parentid = " + str(dirid)
```

• rm()

For rm() we use a simple delete statement to delete the row from the dir table

sql = "DELETE FROM dir WHERE content LIKE '" + filename + "'";

• *put()*

- 1. Take filename and create a tableName for new table
- 2. Create a new row in dir table with filename and tableName
- 3. Create a new table for file contents (.csv, .txt) and assign partition number
- 4. Insert values into table
- 5. For each partition, create a new table and insert a new partition location in the partition table

```
# create partition tables
sql = "SELECT fid FROM dir WHERE content LIKE '" + content + "'"
cursor.execute(sql)
fid = cursor.fetchall()
for x in range(k):
    partitionName = tableName + "_" + str(x)
    cursor.execute("CREATE TABLE " + partitionName + " SELECT * FROM " + tableName +" PARTITION(p" + str(x) + ")")
    sql = "INSERT INTO partitions (file_id, tableName, partition_name, part) VALUES (" + str(fid[-1][0])
    + ", '" + tableName + "', " + partitionName + "', " + str(x+1) + ")"
    cursor.execute(sql)
    db.commit()
```

Results of put():

```
: put("archive/day_wise.csv", "/user", 3)
 processing table into directory ...
 done
 dir()
:
    fid
                name parent parentid
                                                            file
                                                content
 0
      1
                root
                                 0
                                                     1
 1
      2
                user root
                                  1
                                                  /user
 2
                                   2 /user/day_wise.csv day_wise
      4 day_wise.csv
                      user
 partitions()
:
    pid file id tableName partition name
                                        part
 0
      1 4 day_wise day_wise_0
                                         1
 1
      2
             4 day wise
                            day wise 1
                                           2
              4 day wise day wise 2
 2
      3
                                           3
```

• cat()

For cat(), we use a simple select statement to output the table

• getPartitionLocations()

To get the location of the partitions in the EDFS, we simple map to their IDs in their table

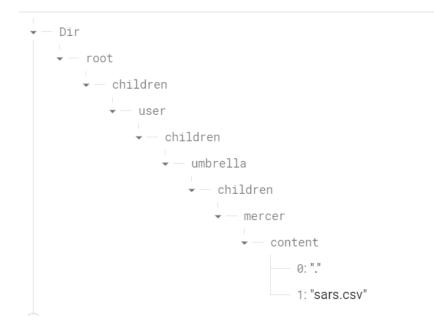
• readPartitions()

To read a partition, we map to the partition table created in put, and map to the part specified by the user

B.2 Firebase Implementation

For our Firebase implementation, we are using 3 cards:

1. Dir -> Directory tree , stores children of a node and its content structure



2. Files -> Metadata tree stores the files and their partition names



3. **Store** -> Location where the partitioned file contents are stored

```
https://dsci-project-28e72-default-rtdb.firebaseio.com/
    Dir
    Files
    Store
    ___user_-_umbrella_-_mercer_-_sars_1: "["{\"Country\\/Region\":\"Australia\",\"Cumulative male case:
    ___user_-_umbrella_-_mercer_-_sars_2: "["{\"Country\\/Region\":\"New Zealand\",\"Cumulative male c
    ___user_-_umbrella_-_mercer_-_sars_3: "["{\"Country\\/Region\":\"South Africa\",\"Cumulative male c;
    ___user_-_umbrella_-_mercer_-_sars_4: "["{\"Cumulative male c;
    ___user_-_umbrella_-_mercer_-_sars_4: "["{\"Cumulative male c;
    ___user_-_umbrella_-_mercer_-_sars_4: "["{\"Cumulative male c;
    ___user_-_umbrella_-_mercer_-_sars_4: "["{\"Cumulative male c;
    ___user
```

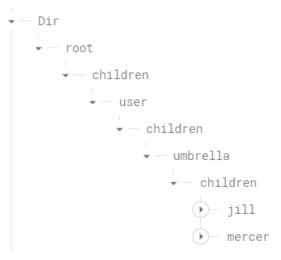
The above 3 cards and their structure is essential for the following commands to run properly.

• mkdir()

Creating a new directory involves first querying the Database for a pre-existing directory. If the directory is new and does not exist, the function proceeds to create the whole directory tree, including and new components.

For eg : /user/umbrella/mercer as an input would create all directories including user, umbrella even if they did not exist. (placeholder acts as an identity file allowing firebase to actually create directory structure. This cannot be done if the card does not have any content)





Result of mkdir

• Is()

Using the directory structure we first need to modify the input to match our directory structure with '/children/' for each directory giving access to its children.

Once done we can list the contents of any directory.

For empty directories we display a message stating the current directory has no children.



D:\MS\DSCI-551\HW\Project≻python tushar_prjkt.py /user/umbrella/mercer {'files': ['.', 'sars.csv']}

• rm()

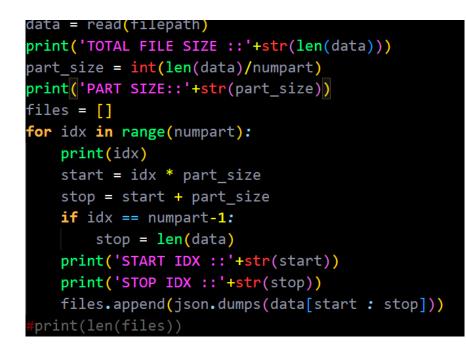
For rm we have to do a 3 pronged approach and delete the file from 3 places:

- a. Dir structure from content list of the path card
- b. Files structure to remove the partitions
- c. Stores structure to remove the actual partitions

• *put()*

Put is one of the more complex implementations, it has the following steps:

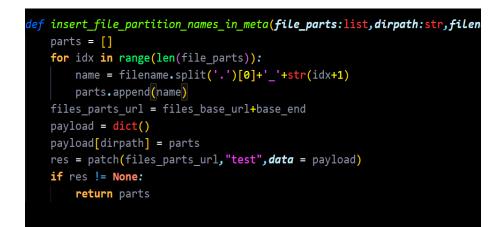
1. Split file into given number of partitions:



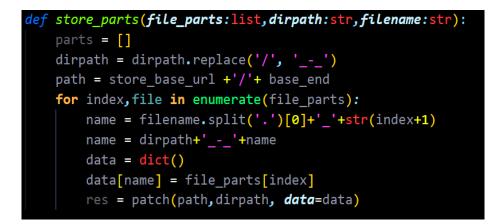
2. Insert file into the Dir Structure:

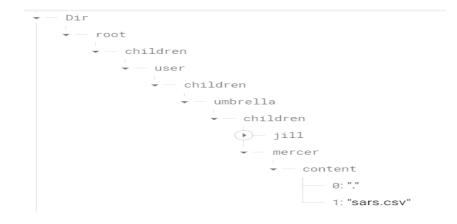


3. Insert file partition names into the Files Structure:

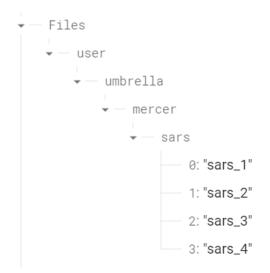


4. Insert data into store with partitions:





Directory Structure after put request



Files Structure after put request



Store Structure after put request

• getPartitionLocations()

We simply access the two Cards Files and Store to get the Partition Locations.

```
def getPartitionLocations(file : str):
    filename = file.split('/')[-1].split('.')[0]
    dirPath = file.removesuffix('/'+filename+'.csv')
    #print(dirPath)
    data = ls(dirPath)
    #print(data)
    locs = None
    if "files" in data and filename+'.csv' in data['files']:
        #now return the partitions
        partsNames = getParts(filename,dirPath)
        locs = generateLocs(dirPath,partsNames)
    return locs
```

```
D:\MS\DSCI-551\HW\Project>python tushar_prjkt.py /user/umbrella/mercer sars.csv
{'files': ['.', 'sars.csv']}
{'sars_1': 'Store/user/umbrella/mercer/sars_1', 'sars_2': 'Store/user/umbrella/mercer/sars_2', 'sars_3': 'Store/user/umb
rella/mercer/sars_3', 'sars_4': 'Store/user/umbrella/mercer/sars_4'}
```

• readPartitions()

For reading a numbered partition of the given file, we first ensure that the file exists. Then we use the generated getPartitionLocations() to fetch data from the location.



• cat()

Cat uses the existing function of getPartitionLocations and readPartitions to reconstruct the complete file and return it.

<pre>D:\MS\DSCI-551\HW\Project>python tushar_prjkt.py /user/umbrella/mercer/sars.csv ['{\"Country\\/Region\":\"Australia\",\"Cumulative male cases\":4,\"Cumulative female case\":2,\"Cumulative total cases \":6,\"No. of deaths\":e,\"Case fatalities ratio (%)\":e,\"Date onset first probable case\":12003-02-26\",\"Date onset last probable case\":12003-04-011",\"Median age\":15.0,\"Age range\":\"1-45\","Number of Imported cases\":6,0,\"Percent tage of Imported cases\":160.0,\"Number of HGW affected\":0,\"Percentage of HGW affected\":0,\"Cumulative total cases\":6,0,\"Percent ':CanadA',"Cumulative male cases\":11,\"Cumulative female cases\":100,\"Cumulative total cases\":25.0,\"Date onset 13, probable case\","Cumulative male cases\":10,"Cumulative total case\":251,\"No. of deaths\" ':2003-06-12\",\"Median age\":49.0,\"Age range\":1-98\",\"Number of Imported cases\":5.0,\"Percentage of Imported cases\":5.0,\"Percentage of Imported cases\":5.0,\"Cumulative total case\":\"Cumulative total cases\":251,\"No. of deaths\","Cumulative total case\":251,\"No. of deaths\","Cumulative total case\":251,\"No. of deaths\","Cumulative total case\":2603-06-12\",\"Date onset first probable case\":149,\"Cumulative total cases\":267,\"Cumulative total cases\":2603-06-23\", "Cumulative total cases\":267,\"Cumulative female cases\":2607,\"Cumulative total cases\":251,\"No. of deaths\":249,\"Cumulative total cases\":267,\"Cumulative total cases\":2603-06-32\", "Motore onset last probable case\":267,\"Cumulative total cases\":267,\"Cumulative total cases\":267,\"Cumulative total cases\":27,\"Cumulative total cases\":2003-06-33\", "Motore onset last probable case\":2003-06-33\", "Motore onset last probab</pre>
cases\":128,\"Cumulative total cases\":346,\"No. of deaths\":37,\"Case fatalities ratio (%)\":11,\"Date onset first pro
arrected :20; { { Country/\/Region : / France , < CumUlative male cases :1, < CumUlative female cases :0, < CumUlative vental cases :1, < CumUlative female cases :0, < CumUlative vental cases :1, < CumUlative female

B.3 Application User Interface Implementation

For the application's user interface, we used Python3, HTML5/CSS3, Javascript, JQuery, Bootstrap libraries, Axios, and the application uses the Flask web framework as a local development server.

The file structure for the project follows the same structure as specified in the Flask documentation. The Flask web application project structure is as follows:

V DSCI551-PROJECT	
>pycache	
> .ipynb_checkpoints	
> archive	
> covid	
> ebola	
> sars	
\checkmark static	•
JS firebase.js	м
JS mysql.js	м
# styles.css	
\sim templates	•
💠 explanation.html	
firebase.html	м
index.html	м
🔷 mysql.html	м
> uploads	
.gitignore	
🗬 app.py	
covid.csv	
🗬 firebase_api.py	м
🗬 mysql_api.py	м
🖶 mysql-nb.ipynb	м
 README.md 	
≡ requirements.txt	
III sars.csv	
sqltable.csv	U

Flask Project Structure

The *static* folder contains the static items that are served by the server. The firebase.js and mysql.js files contain the Javascript, JQuery implementations that render and update the HTML based on the API responses. The styles.css provides the CSS styling template for the project. The *templates* folder contains index.html, firebase.html, mysql.html, and explanation.html code for front-end. The *app.py* file creates the Flask application instance as it imports the Flask objects and registers all the front-end paths. The firebase_api.py and mysql_api.py files are the backend code of the application, and they have the REST APIs for each feature implementation.

To start the Flask application instance the following commands are required to be executed on the command line:



The application can be accessed on URL : <u>http://127.0.0.0.1:5432</u>

The URL takes the users to the Project Home page which contains two buttons, which allows users to click on the specific EDFS implementation.

The Home page also contains a 'About Operations' button which redirects the user to the explanation page. The explanation page gives the list of features provided by the application.

hreene File Edit View History Bookmanks Profiles Tab Window Help	Image: Antiperson of the second se	••• Statester Note x +		
COCCI SEI Project Home x +	v		,	0 * 4 *
				😲 GRS Practice Text 🔊 Sign in for Text Ta 🧐 New Tab 🔽 How-to instal Ha 🌒 YACC 🔹 🛅 Of
anarks 🥳 Education in the U. 🔣 WWW Software pr., 🥎 PROSEAUMING L., 🎘 Data Structure an., 🚺 DES Practice Test. 🥏 Sign in for Test	et Ta 🚷 New Tab 🙀 How-to Install Ha 🚷 VADC 🛛 > 🖽 Other Bookmarks	a socialità di tassassi interes. 📓 nen so	and the second s	
DSCI 551 Project Emulated Distributed File System for exploration of Outbr COVID 19, Ebola, and SARS	vreaks			nation Page pplication features available to the end users
Explore our App Features enabled by EDFS (Firebase and MySQL) by selectin About Operations 0	ing one of the given options		Applica	ation Features
		Feature/Command	Feature Description	Feature Details
Firebase MySQL		mixtir	Create a directory in the file system	Click on the 'New Folder' button and enter the directory to be created in the file system. The alert message informs the directory creation status to the users.
		la .	Listing contents of a given directory	In File Explorer section, enter the path of the directory and click on 'Search'. The files in that directory can be viewed below.
		cat	Displays contents of a file	In File Explorer section, in the search bar enter the file path to be displayed, and click on 'Open' button. The file contents are displayed in the 'File Content' section below.
		m	Remove a file from the file system	In File Explorer section, in the search bar enter the file path to be deleted, and click on 'Delete' button.
		put	Uploading a file to the file system	Click on the 'Upload' button and enter the file path, number of partitions, choose a file to upload from your local machine, and click on 'Upload' button. The alert message informs the upload status to the users.
		getPartitionLocations(file)	Returns the locations of partitions of the file	In File Explorer section, in the search bar enter the file path and hit enter. The partition location details can be viewed on the information section of the webpage.
		readPartition(file, partition#)	Returns the content of partition # of the exactlined file.	In the Information section that displays the partition location details, enter the number of the partition that you need to seed. The partition protects are displayed below.

Project Home Page

Explanation Page

By clicking on the Firebase or MySQL button on the Project Home page, the user can go to one of the implementations.

🖆 Chrome File Edit View History Bookmarks Profiles Tab Window Help	🕒 🗃 🚨 🕴 🕑 🦸 100% 1688 😨 💂 Man New 28 6-43 AM	🔹 Chrome File Edit View History Bookmarks Profiles Tab Window Help	🕒 📟 🙆 🤻 🕢 💲 1997 1888 😪 🗟 Man Nev 28 644 AM
••• Firebase File pystem x +	۰ ۲	••• OMySOL File system x +	v
← → C (0 127.0.0.1:5432)Tirebase	0 🖈 🖡 🖬 🔞 E	← → C (0 127.0.0.15432/mysql	0 x 🔹 🖬 🖬 🚷 🗄
🖈 Bookmanks 🦉 Education in the U. 🦉 WWW Software pr. 🔇 PRODERANUMUD L. 📓 Data Structure an. 👔 GRE Practice Test. 💣 Sign in for Tes	Ta 🤣 New Tab 👱 How-to Install Ha 🧐 YADC 🔹 🗎 Other Bookmarks	★ Bookmarks 🐻 Education in the U. 📓 WWW Software pr., 🔇 PROGRAMMWIG L. 📜 Data Structure an 🍟 GRE Practice Te	st 🧬 Sign Infor Test Ta 🔇 New Tabi 🙀 How-to Install Ha 🔇 YACC 🔹 🗎 Other Bookmarks
Firebase File System		MySQL File S	ystem
File Operations ?		File Operation:	\$?
File Explorer		File Explorer	
Enter path to use with Is	٩	Enter path to use with Is	٩
Enter a path and click on the search button		Enter a path and dick on the se	arch button
Enter a file path	Open Delete	Enter a file path	Open Delete

Firebase Operations Page

MySQL Operations Page

• mkdir()

mkdir on the front-end is given by the 'New Folder' button and by clicking on it, an input field appears where the path of the new directory to be created is specified. After that by clicking on the 'Create' button, the directory is created. The button click internally calls the Firebase (or MySQL) /mkdir REST API which takes 'path' as a parameter, backend processes the request and the API response in JSON format is parsed and rendered on the interface.

🔹 Chrome File Edit View History Boolstrarks Profiles Tab Window Help	S Mon Nev 28 6:52 AM	Sev History Bookmarks Profiles Tab Window Help	🕒 🖬 🖪 🗳 💮 💲 100% 🕬 👳 💂 Man Nav 28 652 AN
• • • Sirotaan File system x +	🗸 🔍 🔮 Firebase File sp	sten X +	*
	★ ★ ■ ③ :		0 x 🗴 🕯 🖬 🙆 E
therease Calcular and Constraints Constraint Constraint Constraint Constraints Constrai	+ Boner Boshneits 🕈 Boshneits 🔀 Básatón h.t	No. C www.browsz. I monosowa 1272.0215422 ktys Bratsky cawled soccessfully	rti. Əl Nərtis 🖬 nəvci kazilik. Əl NAC — I 🛅 Oler Bodrado
File Operations ?	12 New Fal	File Operations ?	
Auser/garde/demo/report	Create	File Explorer	
File Explorer	Enter path	to use with is	٩
Enter path to use with is	٩	This directory is empty	
This directory is empty			
k Pauline Paul	Enter a file	s path	Open Delete

Mkdir directory path entry

Directory created successfully

• Is()

The Is command can be executed by checking in the File explorer section, by entering the file system path for which the user wants to check the contents for. By clicking on the search icon, the contents of the file are displayed in the section below. If the directory is present, 'This directory is empty' message is displayed.

Automatication Calification	e File Edit View History Bookmarks Profiles Tab Window Help	🕒 🖬 🖸 🌾 🕞 🂲 100% KRA 😌 😰 Mon Nov 20 7.11 AM	🔹 Chrome File Edit View History Bookmarks Profiles Tab Window Help	• • • • • • • •
	Ø Firebase Resystem x +	*	••• • Privileau File again x +	
rt hy piñ File Explorer Tis directo js engly thr s file piñ thr s file piñ	💿 127.0.0.1:1432/Trebose 🐼 Exkusion in the U. 📓 WWW Schware pr. 😵 FROCENAMINO L. 🔀 Data Structure an. 👔 ORE Practice Test. 👘			nactice Test. 🕐 Sign in for Test Ta 😵 New Tab. 🎦 How-Is in
File Explorer File Explorer File Explorer File Explorer File Explorer File Explorer File Explorer File Explo	🗈 New Folder 🗘 Upload		New Folder	
inter a fla partie	Enter the path	Create	Enter the path	
This directory is empty International Coper Codes	File Explorer		File Exp	orer
Inter a file public Total Contemport	Juser(ligsróle)demo/report	٩	/user/lgardejdemo	
Enter a lis public	This directory is empty		m report	
Enter a lis public	Enter a Ga noth	Drup Dates		
RECONTING RECONTING RECONTING				
			FILE CONTENT	FORMATION

Empty Directory

Directory contents displayed

• rm()

The rm command can be executed using the 'Delete' button in the FIIe Explorer section of the interface. The parameter the /rm API accepts is the filepath. By clicking on the Delete button, if the delete file operation is successful, the alert box is displayed. After deleting the file, check if the file is deleted by using the ls command.

		0 🖸 🖬 🖾 🌵 Θ 💲 100% 8880 😪	😨 Mon Nox 28 7:19 AM			🖸 🖬 🖸 🍦 🕢 🕴 2004 KND 🖓 😂 Mon No
Firebase File system x +			v •••			
) ≱ \$ □ (8) E (+ → c	0 127.0.0.1.5432/lirebase		
a 💽 Education in the U., 🐷 WWW Software pr., 🔶 PRC	0K	t Ta. 👌 New Tab 😰 How-Is Install Ha 🤣 YACC	* 🖿 Other Bookmarks 🖈 Bookmarks	is 🗿 Education in the U. 🔯 WAW Software pr. 🔗 PRODULUMING L.	👷 Deta Structure an. 👔 GRE Practice Text. 📄 Sign Infer Text Ta.	🕲 New Tab. 🚡 How-to Install Ha 🕲 MCC 🔹 🛅 C
	File Operations?				File Operations ?	
New Folder Lipload	/user/garde/demo	Choose File covid.csv	Upload	🔁 New Folder 🗘 Upicad		
	File Explorer				File Explorer	
			_	/user/kgarde/demo		٩
/user/lgarde/demo			٩			
				Treport test		
report test covid.csv						
				Enter a file path		Open Dekte
In report Intest Covid.csv		Open	Gøre	Enter a file path		Open Delete
	RUE CONTENT INFORMATION	Среп	Dena	Enter il li path	FLE CONTENT INFORMATION	Open Delete

File deleted successfully

Verifying by using Is command

• *put()*

The put command can be used by the user by clicking on the 'Upload' button. The number of partitions, file path, and selecting a file from the local system to post the contents is performed by clicking on Upload. Once the upload is successful, alert message is displayed.

É Chrome	e File Edit	View History Bo	okmanks Profiles	Tab Window Help		0 0 0 4 0 1	: 100% 🛤 😌 🕃 Mon Nov 28 7:15 AM	🗯 Chrome File Edt Vie	w History Bookmarks Profiles Tab	Window Help	0 0 8 0 4 0 *	100% 📾 🗧 🗟 Man Nav 28 7-16 AV.
• • •							*	🖲 🔍 🌒 🛞 Finbase File syste	n x +			
							0 x 4 x 0 6 :	€ → C (0 127.0.0.1:543	2)Trebase			0 x 🕨 🕯 🖬 🔞 E
* Bookmanis	🛛 🗃 Courts	enneu. 🔞 www.s	etaan (s. 🛛 🔞 250	stations III 🗿 Sau Sau	aun un 👖 Gris Practice teal.	🥐 Oprin to Sei Cu. 🔞 New Sol. 🖪 Hawkon	ent Hall 🔕 1965 - 🔸 🗎 Other Basimurka -	* Bookmarks 🔞 Education in the	U., 📓 WWW Software pr., 🚷 PROGRAM	127.0.0.1:5432 says	t Ta 🧿 Mew Tab 👖 How-to Ins	tal Hs 🔇 WCC 🔹 🛅 Other Bookmarks
										ок		
											ox	
	🖬 New I	Folder 🕹 Up	load					🗈 New Folde	r 🗘 Upload			
			Favorites		✓ demo-upload-files	Q Search						
	2		 lavmigarde Desktop 		covid.csv		Upload	2		/user/lgarde/demo	Choose File covid.csv	Upload
			 Downloads 	Ine >	🗄 ebala.csv	FILE						
			A Applications	ct-551 rishot 200.38 AM.prg	sars.csv	- in the second				File Explorer		
			Documents	inshot 200.52 AM.prg inshot 216.56 AM.prg								
			Cloud	inshot 21.17.24 AM.png			٩	Enter path t	n uce with ic			٩
			E* Shared	inshot 230.47 AM.prg inshot 234.32 AM.prg		CSV		Line pairs				
			Locations	inshot 243.54 AM.prg						Enter a path and click on the search b	itton	
			C Google Drive	inshot 244.02 AM.prg inshot 252.04 AM.prg		covid.csv						
			🗁 dataOne	inshot 252.16 AM.prg inshot 27.11.28 AM.prg		CSV Document - 14 KB Information						
			Media JI Music	and at 5 741 \$1 \$14 ann		Created November 25, 2022 at 3:53 PM						
	Enter a	a file path	Photos			Cancel Open	Open Delete	Enter a file p	aath			Open Delete
					E CONTENT INFORMATIO	N				FILE CONTENT INFORMATION		

Upload file from local

Upload done successfully

🗯 Chrome	File Edit View	History Bookmarks	Profiles Tab Windo	w Help		0	• • •	\$ ⊙ \$ 100%	60 ? B	Mon Nov 28 7:16 AM
•••	🕉 Firebase File system	× +								
	0 127.0.0.1:5432/fir									a 🛪 🗉 🛞 E
* Bookmarks	W Education in the U	WWW Software pr	S PROGRAMMING L.	Data Structure an	GRE Practice Test	Sign In for Test Ta	S New Tab	Mow-to Install Ha	🚱 ҮАСС 🕛	Dther Bookmarks
	New Folder	1 Upload								
	2		/use	r/lgarde/demo		Choose File	covid.csv		Up	load
				File	Explorer					
	/user/lgarde/der	mo							C	ર
	repor	t 🖿 test 🗋 cov	id.csv							
	Enter a file path	1							Open Delet	•
				FILE CONTE)N				

Uploaded file checked using Is command

• cat()

cat command can be executed by the user by clicking on the 'Open' button present. The output of the cat command is displayed in the 'FILE CONTENT' tab below.

					ndow Help					•	• • •	* • *	100% (658)	~ • ~
• •	Firebase File system	× +												
	127.0.0.1:5432/fire													
okmarks 🛛	 Education in the U., /user/lgarde/dem 	_	are pr 🦿	PROGRAMMING	i L. 💽 Di	ita Structure	an. 🖬	ORE Practice Tes	el 🥏 Sk	pri In for Test Ta	S New Teb	Mow-to Insta	тны. 🚷	YACC + Q
	Jusei/igai dejdern	10												<u> </u>
	Treport	test	Covid.c	sv										
	/user/lgarde/dem	io/covid.csv											Ope	n Delete
	/acol/Baraclacity	0,001101001											ope	001010
						FILE CO	INTENT	INFORMA	TION					
						FILE C	ONTENT	INFORMA						
		Cumulative				FILE C	NTENT	New		Recovered / 100	Deaths / 100	Confirmed		1 week %
	Country	total cases	deaths	Recovered		New cases	New deaths	New recovered	Deaths / 100 Cases	/ 100 Cases	100 Recovered	last week	change	% increas
	Country Afghanistan			Recovered 25198	Active 9796	New	New	New	Deaths / 100	/ 100	100			%
		total cases	deaths			New cases	New deaths	New recovered	Deaths / 100 Cases	/ 100 Cases	100 Recovered	last week	change	% increas
	Afghanistan	total cases 36263	deaths	25198	9796	New cases 106	New deaths 10	New recovered	Deaths /100 Cases 3.5	/ 100 Cases 69.49	100 Recovered	last week 35526	change 737	% increas 2.07
	Afghanistan Albania	total cases 36263 4880	deaths 1269 144	25198 2745	9796 1991	New cases 106 117	New deaths 10 6	New recovered 18 63	Deaths / 100 Cases 3.5 2.95	/100 Cases 69.49 56.25	100 Recovered 5.04 5.25	last week 35526 4171	change 737 709	% increas 2.07 17
	Afghanistan Albania Algeria	total cases 36263 4880 27973	deaths 1269 144 1163	25198 2745 18837	9796 1991 7973	New cases 106 117 616	New deaths 10 6 8	New recovered 18 63 749	Deaths /100 Cases 3.5 2.95 4.16	/100 Cases 69.49 56.25 67.34	100 Recovered 5.04 5.25 6.17	last week 35526 4171 23691	change 737 709 4282	% increas 2.07 17 18.07
	Afghanistan Albania Algeria Andorra	total cases 36263 4880 27973 907	deaths 1269 144 1163 52	25198 2745 18837 803	9796 1991 7973 52	New cases 106 117 616 10	New deaths 10 6 8 0	New recovered 18 63 749 0	Deaths /100 Cases 3.5 2.95 4.16 5.73	/100 Cases 69.49 56.25 67.34 88.53	100 Recovered 5.04 5.25 6.17 6.48	last week 35526 4171 23691 884	<pre>change 737 709 4282 23</pre>	% increas 2.07 17 18.07 2.6

cat command output

• getPartitionLocations() and readPartition()

The output for these two commands can be viewed by the user in the 'Information' tab. To view the contents of a particular partition, enter the partition number and click on the search icon. The contents of the partition would be displayed below in tabular format.

e File Esit View History Bookman	a notes no willow hop		• • • • • • • • • •			Chrome File Edit View	_		nies las wi	indow He	•				•			10276 (668)	~ *
	epz 🔇 FROGRAMMING L 🕱 Data Structure en 👔 GRE	Practice Test. 💣 Sign in for	r Test Ta 🔇 New Tab 🙍 How-to install He		• 🗉 🔞 🗄 🛛 🗧	 Piebase File system C (0 127.0.0.1:5432) colorada (0 Education in the U.)) FROSPANNEN	or Ma						😵 New Tab	Now-to Inst		
/user/ligarde/demo) covid.cov			Q		/user/lgande/de	mo/covid.csv											Ope	an Deleti
/user/figarde/demo/covid.csv				Open Delete		Partition			Partition	Path	FILE CO	INTENT	INFORMA	TION					
	FILE CONTENT	INFORMATION				cavid_1 cavid_2			Store/user Store/user										
Partition covid_1	Partition Path Store/user/ligarde/demo/cov/id_1					2													
covid_2	Store/user/lgarde/demo/covid_2			œ		Country	Cumulative total cases		Recovered	Active	New cases	New deaths	New	/100	Recovered / 100 Cases	100	Confirmed last week		1 week % increas
Line person munuel or re	90					Kyrgyzstan Laos	33295 20	1301 0	21205 19	10790 1		24 0	817 0	3.91 0	63.69 95	6.14 0	27143 19		22.67 5.26
	Sear	ch				Latvia	1219	31	1045	143	0	0	0	2.54	85.73	2.97	1192	27	2.27
Find the number of deaths per of	country ~		RESULT									Sea	arch						

getPartitionLocation() output

readPartition() output

The search and analysis interface implementation details are present in section C. Both MySQL and Firebase pages have '?' symbols on top for users to access the Explanation Page.

B.4 Project Integration APIs

BASE_URL=	http://127.0.0.1:5432/	
-----------	------------------------	--

Function	Endpoint	Met hod	Query Paramete r	Sample Request URL	Sample JSON Responses
MySQL mkdir	/mysql/mkdir	GET	path	BASE_URL + mysql/mkdir?path=/demo	{ "status": "Directory created successfully!" }
Firebase mkdir	/firebase/mkdir			BASE_URL + firebase/mkdir?path=/demo	
MySQL Is	/mysql/ls	GET	path	BASE_URL + mysql/ls?path=/covid	{ "directories": [], "files":["covid_19_clean_comp lete.csv"] }
Firebase ls	/firebase/ls			BASE_URL + firebase/ls?path=/covid	
MySQL cat	/mysql/cat	GET	file	BASE_URL + mysql/cat?file=/user/test/sars.c sv	["{\"Country\":\"Australia\",\" Cumulative male cases\":4,\"Cumulative female cases\":2,\"Cumulative total cases\":6,\"No. of deaths\":0,\"Case fatalities
Firebase cat	/firebase/cat			BASE_URL + firebase/cat?file=/user/test/sars .csv	ratio (%)\":0,\"Date onset first probable case\":\"2\\/26\\/2003\",\",\ "Median age\":15.0,\"Age range\":\"Jan-45\",\"Number of Imported cases\":6.0}"]
MySQL rm	/mysql/rm	GET	file	BASE_URL + mysql/rm?file=/user/test/sars.c sv	{"status": "OK"}
Firebase rm	/firebase/rm			BASE_URL + firebase/rm?file=/user/test/sars .csv	
MySQL put	/mysql/put	POST	file, dirPath, numPart	BASE_URL + mysql/put?dirPath=/user/test& numPart=2 Body: file: sars.csv	{"status": "OK"}
Firebase put	/firebase/put			BASE_URL + firebase/put?dirPath=/user/test &numPart=2	

				Body: file: sars.csv	
MySQL getPartitionLo cations	/mysql/getPartition Locations	GET	file	BASE_URL + mysql/getPartitionLocations?file =/user/test/ebola.csv	{ "ebola_1": "Store/user/test/ebola_1", "ebola 2":
Firebase getPartitionLo cations	/firebase/getPartiti onLocations			BASE_URL + firebase/getPartitionLocations?f ile=/user/test/ebola.csv	ebola_2: "Store/user/test/ebola_2" }
MySQL readPartition	/mysql/readPartitio n	GET	file, partNumber	BASE_URL + mysql/readPartition?file=/user/ test/ebola.csv&partNumber=2	["{\"Country\":\"Italy\",\"Date \":\"6\\/22\\/2015\",\"Cumul ative total cases\":1.0,\"No. of
Firebase readPartition	/firebase/readParti tion			BASE_URL + firebase/readPartition?file=/use r/test/ebola.csv&partNumber=2	deaths\":0}"]
MySQL search function 1	/mysql/countrydea thcount	GET	dataset, country	BASE_URL + mysql/countrydeathcount?data set=covid&country=India	{"mapper": ["{}", "{\"Country\":{\"0\":\"India\" },\"No. of deaths\":{\"0\":33408},\"Cum ulative total
Firebase search function 1	/firebase/countryd eathcount			BASE_URL + firebase/countrydeathcount?da taset=covid&country=India	cases\":{\"0\":1480073}}", "{}", "{}"], "reducer": "{\"Country\":{\"0\":\"India\" },\"No. of deaths\":{\"0\":33408},\"Cum ulative total cases\":{\"0\":1480073}}"}
MySQL search function 2	/mysql/findcountri esbetween	GET	dataset, limit1, limit2	BASE_URL + mysql/findcountriesbetween?d ataset=covid&limit1=1000&limi t2=5000	Outputs too large.
Firebase search function 2	/firebase/findcount riesbetween			BASE_URL + firebase/findcountriesbetween? dataset=covid&limit1=1000&lim it2=5000	
MySQL analysis functions	/mysql/analysisdea thpercountry /mysql/analysisrec overy	GET	-	BASE_URL + /mysql/analysisdeathpercountry BASE_URL + mysql/analysisrecovery	Outputs too large.
Firebase analysis functions	/firebase/analysisd eathpercountry /firebase/analysisr ecovery			BASE_URL + firebase/analysisdeathpercount ry BASE_URL + firebase/analysisrecovery	

C. Search and Analytics Functions

Search Functions:

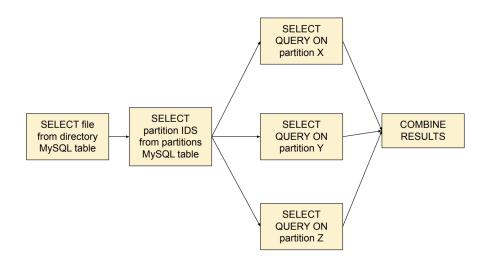
- 1. Death count for a specific disease (of 3) for a specific country.
- 2. Find countries with a death count between two numbers for a specific disease.

Analytics Functions:

- 1. Analyze the count of deaths per country for each disease
- 2. Analyze the number of people who recovered after being affected by the disease for each disease.

C1. MySQL Implementation

For the MySQL implementation for partition-based map and reduce on data stored in our EFDS, we performed our following search and analytic functions on our dataset partitions and combined the results.

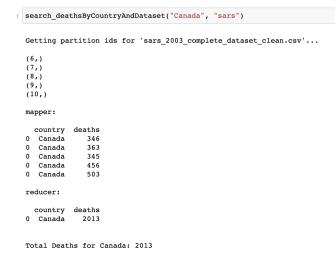


Above is my flow of how I coded the search and analytic functions for MySQL. As you can see, I go through all the tables in my database to get to my final answer.

Search Functions:

1. parameters(disease, country)

For example, if a user puts in the country "Canada" and the disease "sars" then we can look through the sars partition ids, query all the death counts from canada and combine them easily in python.



You can see in the example above, there are 5 partitions and 5 death counts mapped and reduced to a total death count.

2. parameters(disease, upper/lower thresholds)

Following the same workflow, you can see that there is only 1 partition for covid, and for all the sums of deaths per country, it gets filtered between the upper and lower limits decided by the user. The example here uses covid with limits of 2000 and 5000.

	tries_cas	esBetween	VT ("covid", 2000	, 5000)	
Gett	ing parti	tion ids	for	'covid/covid	_19_clea	an_complete.csv'
(1,)						
		count		cases		
0		Afghanist	an	1936390.0		
1				1179755.0		
2		Andor	ra	94404.0		
3		Ango	la	22662.0		
4	Antigua	and Barbu	da	4487.0		
••			••			
182		Yem		67180.0		
183		Comor	os	15823.0		
184		Tajikist	an	383026.0		
185		Lesot	ho	6794 0		
186		Alban	ia	196702.0		
[187	rows x 2	columns]				
Cou	ntries wi		bet	ween 2000 and	5000	
	ntries wi		bet			
Cou	ntries wi	th cases 1		country	cases	
Cou 2000 5	ntries wi	th cases 1		country and Barbuda	cases 4487.0	
Cou 2000	ntries wi	th cases 1		country and Barbuda Belize	cases 4487.0 2636.0	
Cou 2000 5 17 19	ntries wi	th cases 1		country and Barbuda Belize Bhutan	cases 4487.0 2636.0 4971.0	
Cou 2000 5 17	ntries wi	th cases 1		country and Barbuda Belize Bhutan Dominica	cases 4487.0 2636.0 4971.0 2059.0	
Cou 2000 5 17 19 49	ntries wi	th cases 1		country and Barbuda Belize Bhutan Dominica Fiji	cases 4487.0 2636.0 4971.0	
Cou 2000 5 17 19 49 59	ntries wi	th cases 1		country and Barbuda Belize Bhutan Dominica Fiji	cases 4487.0 2636.0 4971.0 2059.0 2266.0 4845.0	
Cou 2000 5 17 19 49 59 63	ntries wi	th cases 1		country and Barbuda Belize Bhutan Dominica Fiji Gambia Grenada	cases 4487.0 2636.0 4971.0 2059.0 2266.0 4845.0	
Cou 2000 5 17 19 49 59 63 69	ntries wi	th cases 1		country and Barbuda Belize Bhutan Dominica Fiji Gambia Grenada	cases 4487.0 2636.0 4971.0 2059.0 2266.0 4845.0 2466.0 2229.0	
Cou 2000 5 17 19 49 59 63 69 94	ntries wi	th cases 1	gua	country and Barbuda Belize Bhutan Dominica Fiji Gambia Grenada Laos	cases 4487.0 2636.0 4971.0 2059.0 2266.0 4845.0 2466.0 2229.0 2236.0	
Cou 2000 5 17 19 49 59 63 69 94 141	ntries wi	th cases 1	gua	country and Barbuda Belize Bhutan Dominica Fiji Gambia Grenada Laos Saint Lucia	cases 4487.0 2636.0 4971.0 2266.0 2266.0 2466.0 2229.0 2236.0 2771.0	

Analytics Functions:

1. In order to find the total deaths per country across all datasets, we found the total deaths per country for each dataset, and combined all three results of each dataset. Here is an example of a SQL query we used...

"SELECT CountryRegion, SUM(Deaths)

FROM partition

GROUP BY CountryRegion"

2. In order to find the average number of people recovered from all datasets, we query all partitions in such a way...

"SELECT AVG(Confirmed), AVG(Recovered)

FROM partition"

We used the average number of confirmed cases as a way to analyze the data. We take all the averages and combine them to make a total average across all datasets to get an output like this:

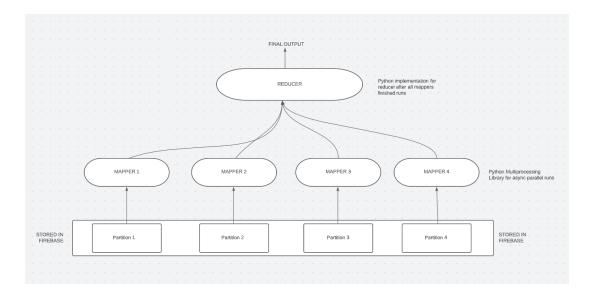
```
averageNumberofRecoveredPerOutbreak()
```

```
{'mapper': {'sars': [[175, 275], [107, 154], [142, 264], [100, 177], [154, 238]], 'ebola': [[299, 2863], [205, 2863],
[184, 2861], [171, 2863]], 'covid': [[7915, 16885]]}, 'reducer': {'sars': [136, 222], 'ebola': [215, 2863], 'covid':
[7915, 16885]}}
```

C2. Firebase Implementation

The general implementation involves creating a mapper and a reducer function for each specific search and analysis function.

The workflow needs to emulate a hadoop implementation and as such needs to run mappers asynchronously. For this I opted for the multiprocessing library of python and its associated thread pools to run multiple jobs asynchronously for different partitions.



The general solution is represented in the flowchart above. The sample workflow for the specific queries is described below.

- 1. Death counts for specific diseases for a specified country:
 - a. First we ask the user to select a Dataset for which they want to search the death count.
 - b. Next we ask the user to enter the name of a country to look for deaths for that specific country.
 - c. The following steps are taken to search for the result:
 - i. We run parallel processes for mapping (1 for each partition the file is divided into) which does a select filter over the partitioned data to find the country mentioned in the search input.



- ii. After this the results of all the mappers are collected in a list and sent as input to the reducer, which then further searches for the results in the mapper results.
- 2. Find countries with a death count between two numbers for a specific disease.
 - a. For this search query the user enters lower and upper limits for death count for a specific dataset.
 - b. The mapper then runs through all the partitions searching for countries that match the condition. The result for the mappers is then collected by the reducer and returned to the API call.
- 3. Analyze the count of deaths per country for each disease
 - a. The analysis questions are posed to repeat the action for all 3 disease datasets Sars, Covid, and Ebola.
 - b. First the mapper runs through the partitions and collects the number of deaths in the current partition for each country.
 - c. Then the reducer sums over the mapper results to produce the total number of deaths per country for the dataset.
- 4. Analyze the number of people who recovered after being affected by the disease for each disease.
 - a. The analysis questions are posed to repeat the action for all 3 disease datasets Sars, Covid, and Ebola.
 - b. First the mapper runs through the partitions and finds the number of people who survived the disease.
 - c. Then we collect the total number of countries present in our partition, we return these three values to our mapper to combine and return the average number of people who recovered from the specific disease.
 - d. Then the reducer sums over the mapper results to produce the total number of people recovered for the diseases, it then sums over the total country count to calculate and return the average number of people who

^{[&}quot;mapper": {"sars": [[7693.0, 6964.0], [33.0, 31.0], [259.0, 224.0], [111.0, 103.0]], "ebola": [[1387413.0, 831470.0], [1737814.0, 1021368.0], [1603719.0, 954201.0], [1596516.0, 963003.0]], "covid": [[4167167.0, 4022094.0], [3364889.0, 3209857.0], [1874660.0, 1784999.0], [7073769.0, 6809499.0]]}, "reducer": {"sars": [252.48275862068965, 8096.0], "ebola": [1522.0193782801775, 6325462.0], "covid": [84633.41711229946, 16480485.0]}]

recovered from the disease.

C3. Application User Interface Implementation

1. Search Functions:

The search functions are present in the Search section of the user interface. The two functions described above in Section C1 and Section C2 are displayed using the accordion HTML component. The output of the search function is displayed in the 'RESULT' section.

27.0.0.1:5432/fire ducation in the U		are pr <table-cell></table-cell>	PROGRAMMING	iL. 🗵 De	ta Structure	an 🖬	ORE Practice Tes	n. 🔊 Si	ge in for Test Te	😒 New Tab	Now-to Insta		() ☆ () () YACC ~ () ()
Country	Cumulative total cases		Recovered	Active	New cases	New deaths	New recovered	/100	Recovered / 100 Cases	100	Confirmed last week		
Kyrgyzstan	33296	1301	21205	10790	483	24	817	3.91	63.69	6.14	27143	6153	22.67
Laos	20	0	19	1	0	0	0	0	95	0	19	1	5.26
Latvia	1219	31	1045	143	0	0	0	2.54	85.73	2.97	1192	27	2.27
						Sea	arch						
Find the numbe	er of deaths pe	r country				\sim				RESULT			
Find countries f	for the number	of cases	selected			\sim							

Search Section

• Finding Death counts for specific disease for a user specified country

The user can select the country and disease for checking the death counts of that country. By clicking on the 'Submit' button the user can see the number of deaths for that country.

4 (hum	File Edit Wew History Bookmarks Profiles Tab Window	114			\$ 1006 BB	den Har 26 7 46 41
_	⊘ Distant File system × +			0-0.0		
e ⇒ e	 ○ 127.0.0.15432/finibase Biokasio in the U B WWW Software pr S PROBAMMIND L 	🕅 Data Structure an 🚺	OHE Practice Test 🔊 Sign In for Test Ta	. 🔕 New Tab 👖 How-10		🖈 🖬 👸
		Se	arch			
	Find the number of deaths per country	^		RESULT		
	Choose the disease: COVID-19	*	Total Deaths in INDIA - Cumula Number of Deaths: 33408	tive: 1480073		
	Choose the country: India	*				
	Submit					
	Find countries for the number of cases selected	~				
		Ana	alysis			
				RESULT		
	Total Deaths per Country					

Search death count of the country

• Finding countries with a death count between two numbers for a specific disease.

This function takes in three parameters, disease, limit1 and limit2. By clicking on the 'Submit' button, the countries that have the number of disease cases between limit1 and limit2 are displayed.

	e Edit View History Bookmarks Profiles Teb Window ebase file system X +	и Нефр	ଦ		698 😤 😫 Mon Nov 28 7:48 AM
< → c ⊙	127.D. 0. 1:5432//iirebase Education in the U WWW Software pr 🗞 PROGRAMMING L	Data Structure an	🛛 ORE Practice Test 🧳 Sign In for Test Ta.	. 🤣 New Tab 🔀 How-to Install Ha	O ☆ ▲ ★ ■ G I
		S	earch		
	Find the number of deaths per country	~		RESULT	
	Find countries for the number of cases selected	^	Country	No. of Deaths	
	Choose the disease:		Afghanistan	1269	
	COVID-19	~	Algeria	1163	
	Specify the range below:		Argentina	3059	
	1000		Bangladesh	2965	
	5000				
	Submit				
		Ar	nalysis		
				RESULT	
	Total Deaths per Country			NEOCEI	

Search countries in the user specified range

2. Analysis Functions:

The analysis functions are present in the Analysis section of the user interface. The two functions described for analysis in Sections C1 and C2 are displayed using block buttons. The output of the analysis functions is displayed in the 'RESULT' section.

		_	Profiles Tab Window	Help		0	• • 6	¥ © \$ 73071	169 - 2	Mon Nov 28 St 10 AM
← → ♂ (0 12	ebase File system 127.0.0.1:5432/fireb									
	Find the number	of deaths per count	ry	~	GRE Practice Test	Sign in for Test Ta	RESULT	M How-to Install Ha.	S YACC *	i 🛅 Other Bookmarks
	Find countries for	r the number of cas	es selected	A	nalysis					
	,		ns per Country of Recovered Ca	ses			RESULT			

Analysis Section

• Analysis of the death count per country for each disease

To view the analysis of death count per country, the user can click on the 'Total Deaths per Country' button. The result of the analysis (as explained in Sections C1 and C2) is displayed in the 'RESULT' section.

🔹 Chrome File Edit View History Bookmarks Profiles Tab Window Help			🕴 🌩 😰 Man Nov 28 9:10 AM
• • • • S Firebase File system × +			
← → C (0) 127.0.0.1:5432/firebase			0 🛠 🔺 🗖 🔞 🗄
★ Bookmanks 🔞 Education in the U., 🔝 WWW Boltware pr., 🤌 PROBRAMMING L., 🐚 Data Structure an.,	COUNTRY COUNTRY	Test Ta 🔇 New Tab 📉 How-to install Ha 🤅 NO. OF Deatins	🔊 YACC 🔹 🛅 Other Bookmarks
Choose the disease:	Afghanistan	1269	
COVID-19 ~	Algeria	1163	
Specify the range below:	Argentina	3059	
1000	Bangladesh	2965	
5000			
Submit			
A	nalysis		
		050147	
Total Deaths per Country		RESULT	
iotal Deauls per Country	SARS		
	Country	No. of Deaths	
Average Number of Recovered Cases	Australia	0	
	Canada	43	
	China	349	

Number of Deaths per Country output

• Analysis on the number of people who recovered after being affected by the disease

The user can click on the 'Average Number of Recovered Cases' button to view the analysis of the number of people who recovered after being affected by the disease. The result of the analysis is displayed in the 'RESULT' section of the interface.

•••	Firebase File system	× +								
	© 127.0.0.1:5432/fir									* 🗉 🛞 E
Bookmarks	W Education in the U	WWW Software pr	S PROGRAMMING L	Data Structure a	⊶ <u>∎</u> e	IRE Practice Test 🧶	🔊 Sign in for Test Ta 🥱 New Tab	Mow-to Install Ha	S YACC +	E Other Bookmarks
	Find the numb	er of deaths per coun	try		~		RESULT			
	Find countries	for the number of ca	es selected		~					
					Ana	lysis				
							RESULT			
		Total Deat	hs per Country			NEMULEI	RESULI			
						Dataset	Average Recovered Cases	Total	Cases	
		Average Numbe	of Recovered Ca	ses		sars	7322	8096		
						ebola	3770042	63254	162	
						covid	15826449	16480	1485	

Number of recovered cases

D. Project Tracker:

Date	Task Description	Responsibilities	Status
9/19	Proposal	Group Meeting on zoom to write out proposal	DONE
9/19 - 10/3	Research	Tushar: Individual research on Firebase Laxmi: Individual research on web apps Raajitha: Individual research on MYSQL ** Research will include: storing datasets, how we want to use commands, functions to use, user-flow, interface languages, database formatting, contents display, and storing**	DONE
10/3 - 10/17	Task-1 Implementati on	 Tushar: implement commands from task one on firebase Laxmi: set up web browser app and create structures/interface for potential user Raajitha: implement commands from task one on MYSQL **Actually start coding and setting up the commands with their executions. We can use this time to communicate with each other, ask questions and make changes** 	DONE
10/17- 10/31	Task 1 Completion, Task 2 Implementati	Tushar: Complete firebase implementation Laxmi: Start connecting interface with databases Raajitha: Complete MYSQL implementation ** will start to add updates to report **	DONE

	on, Midterm Report		
10/31	Midterm Report	Group meeting to finalize and submit midterm report	DONE
10/31 - 11/14	Task 2 Completion, Task 3 Implementati on	Work on PMR operations together	DONE
11/14 - 11/28	Complete Task 3 & Integration. Final Report	Tushar: Test and finalize task 2 Laxmi: Test and finalize task 3 & Integration Raajitha: Test and finalize task 2	DONE
11/28	Final Report	Group meeting to finalize report and presentation video	DONE
11/28 - 12/1	Work on Peer Evaluations	** individually work on peer evaluations **	-
12/2	Peer Evaluations	Submit evaluations	-

E. Changes to previous project work

1. MySQL:

we changed the put() function to allow lots of data to be loaded fast. I was manually inserting the lines before, but now we just use the "LOAD LOCAL INFILE" to load the csv file extremely quickly because before that, the long datasets were not loading at all.

2. Firebase:

- One of the major changes needed was to introduce children and content in the directory system. This was needed to handle multiple subdirectories and files easily in the EDFS. We further split the location of partitions and

the actual partitions themselves into separate cards. This prevents errors due to path matches and gives a clean and extendable implementation.

 As expected implementation of map reduce emulation required the help of the multiprocessing library in python which essentially allowed the mappers to run in parallel and emulate the working of a distributed file system.

3. Application User Interface

The Tree view presented and statically generated during the Midterm is changed on the user interface due to challenges faced during dynamic updates and integration to update and render the Treeview file system successfully. The new approach provides users with intuitive

F. Challenges, Learnings & Future Work:

1. MySQL:

- AWS RDS Instance connection: This took a lot of time for me to figure out in terms of connecting it to python and forming a proper database using python and MySQL because I do not have windows. I learned a lot with how AWS works and how I can build projects in the future.
- Loading CSV File: I had to dabble a lot with the MySQL permissions in order to allow a whole csv file to be loaded into a table. I learned a lot about how you can bulk load csv files easily into MySQL tables instead of inserting each line.
- Search & Analytics: These functions were hard to do because some datasets had different data like cumulative columns versus regular columns. I had to do different functions for different datasets. This helped me learn a lot about aggregate functions and how to query on python and MySQL as well as concat multiple tables.

2. Firebase:

- One of the biggest challenges was to get the directory substructure right.
- Next big challenge was to get all the urls of the firebase system right to ensure the 3 big cards (Dir, Files and Store) are handled correctly

- Thinking about the edge cases for put() function was especially challenging, and it resulted in a big multi-step function.
- The analytics were challenging to process due to the nature of distributed data and cumulative functions, some stats like averages also required me to collect the count of inputs distributed over the dataset partitions. Creating a standardized structure for these distributed requests was a great learning experience.

3. Application User Interface:

- The initial design of the user interface of considering the EDFS commands and putting it into an intuitive user interface for the users was a bit challenging.
- As we were new to the Flask web framework and related development, it was a new learning curve for us as a team. We all are relatively new to front-end technologies, and building user interfaces, the initial development phase was a bit challenging for us.
- Javascript, JQuery implementations, handling Axios and rendering the correct response on the front-end, integration and testing were challenging.
- Future work is to host this application on a web application server i.e. on platforms like Heroku, PythonAnywhere, GCP. Improving the performance of the interface, new errors/exception handling, and making the application more intuitive and user friendly.

The project work had a good learning curve, and gave us an opportunity to work together as a team.

G. Useful Links and Resources

Demo Video Link:

https://www.youtube.com/watch?v=bPm5M0QuKhg&feature=youtu.be

Project Code Google Drive Link:

https://drive.google.com/drive/u/0/folders/1M1kANNc5BqycbigY_GKEqcThQ6RSp-z5

GitHub Code Link:

<u>https://github.com/laxmigarde/dsci551-project</u> <u>https://github.com/laxmigarde/dsci551-project/tree/integration-frontend</u>

Resources and materials referred:

- 1. Flask Documentation: <u>https://flask.palletsprojects.com/en/2.2.x/</u>
- 2. Bootstrap Documentation:<u>https://getbootstrap.com/docs/5.0/getting-started/introduction/</u>
- 3. JQuery: <u>https://www.w3schools.com/jquery/default.asp</u>
- 4. Axios: <u>https://axios-http.com/docs/intro</u>
- 5. Javascript MDN: <u>https://developer.mozilla.org/en-US/docs/Web/JavaScript</u>
- 6. Font Awesome: <u>https://fontawesome.com/</u>
- 7. MySQL APIs: <u>https://webdamn.com/create-restful-api-using-python-mysql/</u>
- Python web application using Flask: <u>https://www.digitalocean.com/community/tutorials/how-to-create-your-first-we</u> <u>b-application-using-flask-and-python-3</u>