Advanced Database Systems

Lab Assignment 3

Creation of a Data Lake

Group No. 7

Members:

Aashay Kaurav	**(2022H1030100)**
Bhagirath Parmar	**(2022H1030076)**
Gourav	**(2022H1030071)**
Parth Goswami	**(2022H1030066)**

Birla Institute of Technology & Science, Pilani
Hyderabad Campus

CONTENTS

Introduction

The idea of evolving above a data warehouse is swiftly gaining popularity as a means of desinging and creating the future creed of systems to deal with fresh large volume data issues. Large firms are aiming to develop these architectures because they handle and work on information with an increasing volume, diversity, and a velocity that has left them mesmerized.

However, a data lake's shared storage and a computational framework (typically distributed) provide the fundamental framework required for the sharing and reuse of large datasets.

The growing usage of data lakes has given rise to some fascinating new challenges for data management research. As a result of data lakes' introduction there are new challenges like dataset discovery, the need for solutions to classic issues like data extraction, making data lean, integrating it, information versoning, and meta-data handling are changing.

Data Lake

It is a huge pool of data swarms that:
1) may be kept in various hardware,
2) may have different forms,
3) may not have any helpful metadata connected to them,
4) may have that metadata represented in different formats, and
5) may change over time.

Businesses are increasingly utilizing the lakes for multiple business reasons. Firstly, they divide data creators (such as working systems) from the users (like reprting and systems that analyse and give an outcome). That would be essential, especially while using outdated mainframe operating systems that may not even be owned by the business, which is common in several sectors, including banking and finance.

In the context of analytics of data, they give a useful layer of means for storing sandbox data, which is also the stimulus and result of anlysing data and gaining knowledge operations. Without consulting with other systems or analysts, data may be produced and used independently.

Capabilities of Data Lake

At a time when there are huge volumes of data, businesses must continuously collect and analyze new forms of data. In an online company, data lakes were initially used to manage web data, but as time went on, many more varieties of data suits were found. As a result, data lakes became more well-liked in the business data management ecosystem.

The data lake is capable of supporting the following features:

1) To mass-collect and affordably store unstructured data: Due to the quick increase in data volume, it is more important than ever to consider data storage costs.
2) A range of data forms, including text, graph, and video data as well as multi-structured and structured data from common DBMS, may be kept in the same repository. Different processing techniques are needed for these various kinds of data.
3) For applying transformations to the data: Main use is cleaning before and some change of information for upcoming system examination.
4) By establishing the structure of the data as it is needed, a technique known as on learn scheme, it eliminates extensive, expensive useful and comprehensive work.
5) To support new types of data processing, the data lake must be able to manage all of the data and all of the data processing techniques.
6) Because it is not known how valuable the information contained the swarm is, individuals will need to generate analytics that are subject-specific in order to discover the most effective way to put the data to use.

Data Lake Architecture

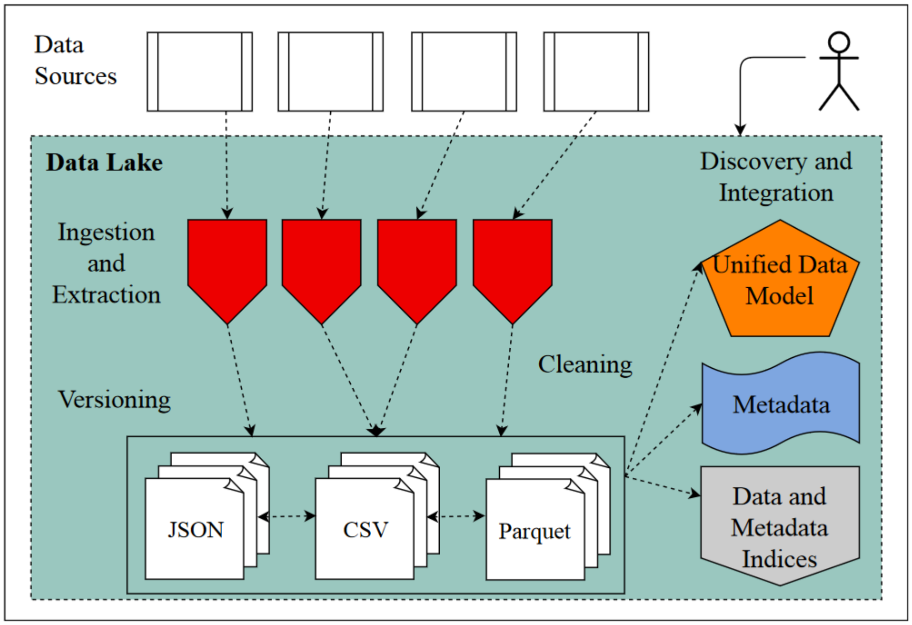


Figure 1: Example Data Lake Management System.

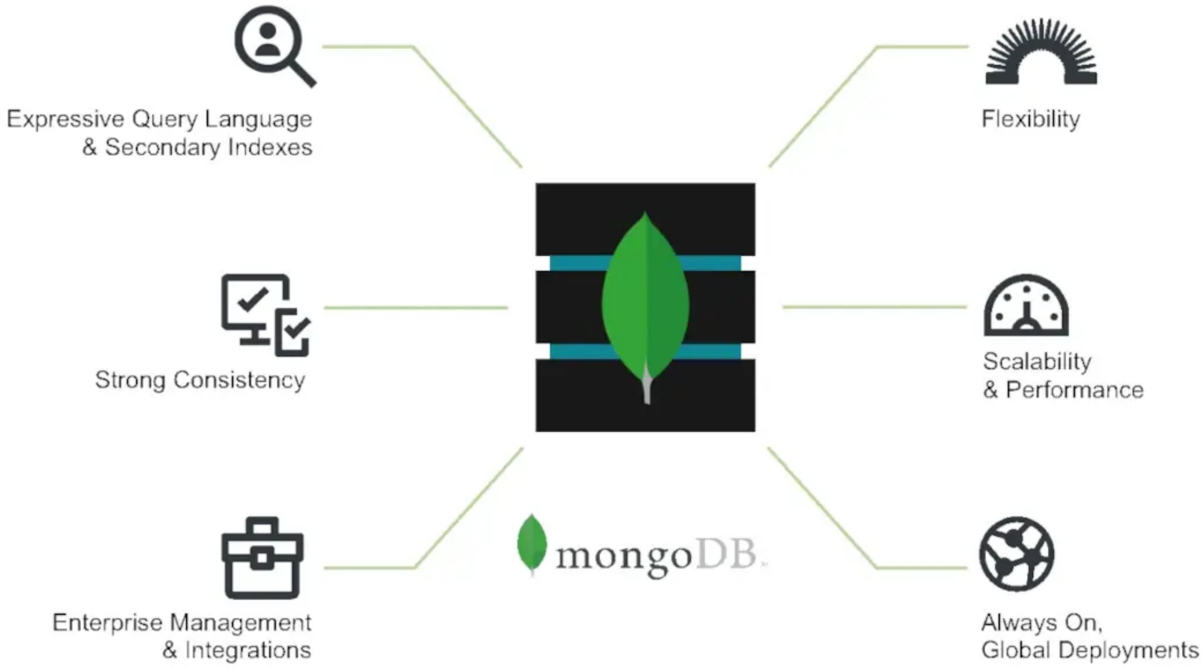
The data may have originated from out-of-date operating systems (running in Cobol or other formats), online scraping from social media and other sources, or from commercial data brokers like Thompson Reuters and Lexis-Nexis.

Other data can include unstructured logs, social media content, and pure papers. The ability of various data lakes to provide a unified perspective over the entire lake or specific areas of the lake differs.

Databases

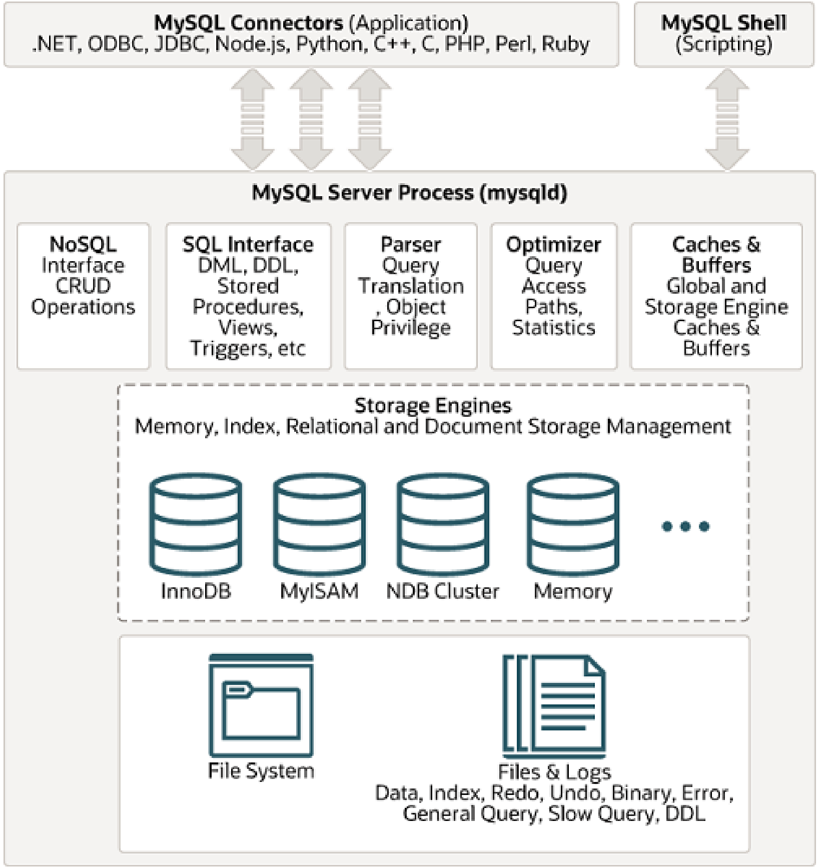
MongoDB

MongoDB is a cross-platform NoSQL database, and it is the newest database that is increasing at the rate that is considered to be the most rapid all across the world. If you have experience using other relational database management system (RDBMS) solutions like MySQL, you will find that the rich document-oriented structure of MongoDB as well as its dynamic queries are quite familiar to you.



MySQL

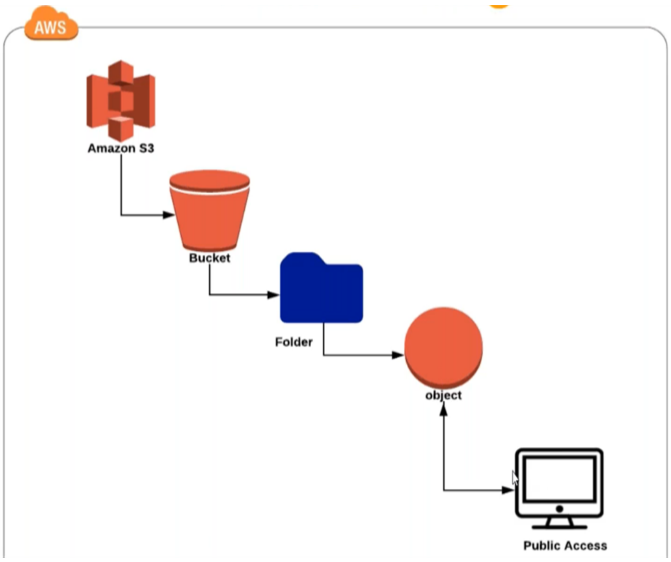
Relational database management systems have emerged as the most effective and extensively used solutions for the persistence of data as a direct result of the vast volume of data flow that has been seen. Even though systems whose code is available are not as widely used as archaic systems such as Oracle or Server, certain systems, such as MySQL, have amassed an incredible amount of popularity over the course of their history. This is because open-source RDBMS systems are free to use and modify by anyone.

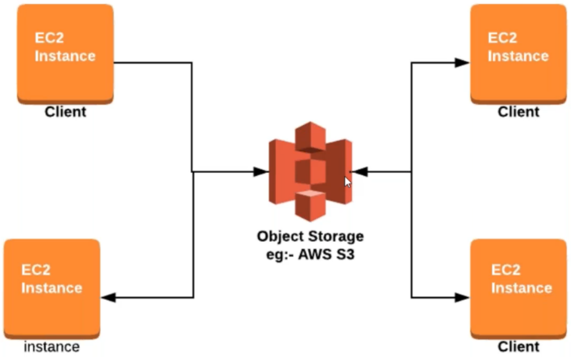

MySQL Connectors (Application)
.NET, ODBC, JDBC, Node.js, Python, C++, C, PHP, Perl, Ruby
MySQL Shell
(Scripting)
MySQL Server Process (mysqld)
NoSQL Interface CRUD Operations
SQL Interface DML, DDL, Stored Procedures, Views, Triggers, etc
Parser Query Translation , Object Privilege
Optimizer Query Access Paths, Statistics
Caches & Buffers Global and Storage Engine Caches & Buffers
Storage Engines
Memory, Index, Relational and Document Storage Management
InnoDB
MyISAM
NDB Cluster
Memory
File System
Files & Logs
Data, Index, Redo, Undo, Binary, Error, General Query, Slow Query, DDL

Amazon S3

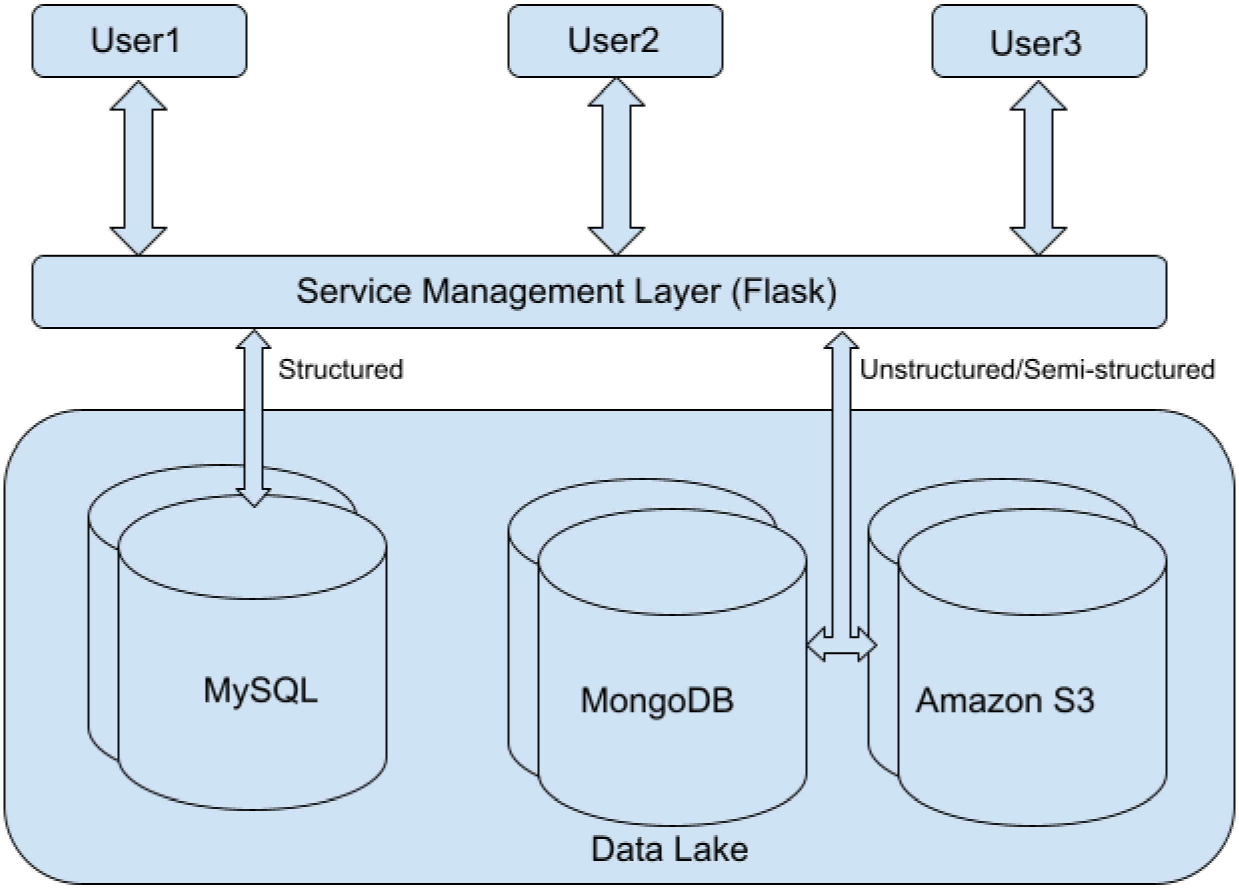
Amazon.com's AWS services provide access to user-level calculation, magnetic tape and cohesion facilities, respectively. These services may be used by any individual or corporation located anywhere in the world that has a payment card that is still active.

Service inconsistencies, not having an agreement, and a troublesome license issue are all holding back this promising Amazon Web Services (AWS) subset from reaching its full potential.





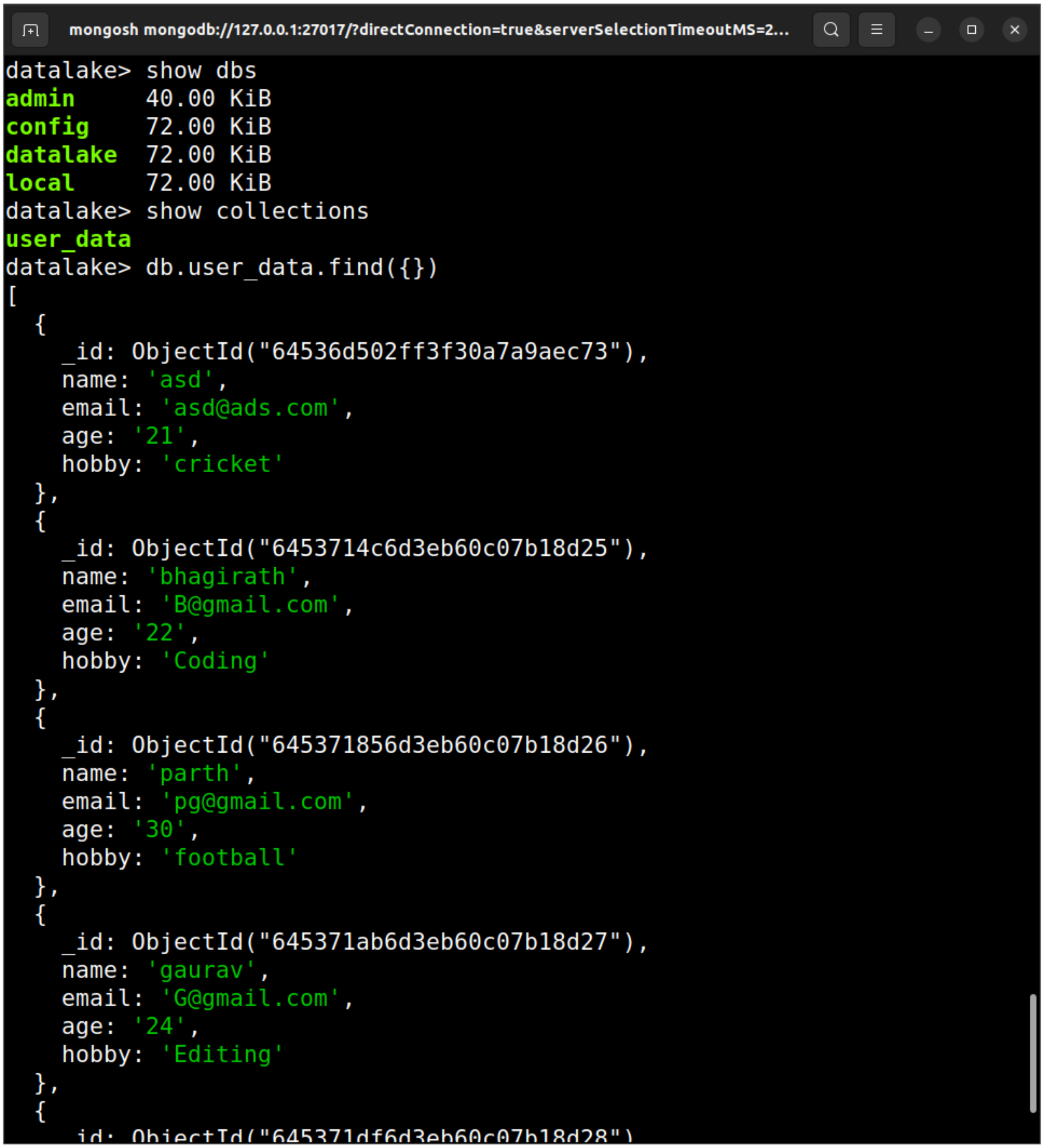
Workflow


User1
User2
User3
Service Management Layer (Flask)
Structured
Unstructured/Semi-structured
MySQL
MongoDB
Amazon S3
Data Lake

Data Lake setup on system

Setting up MongoDB

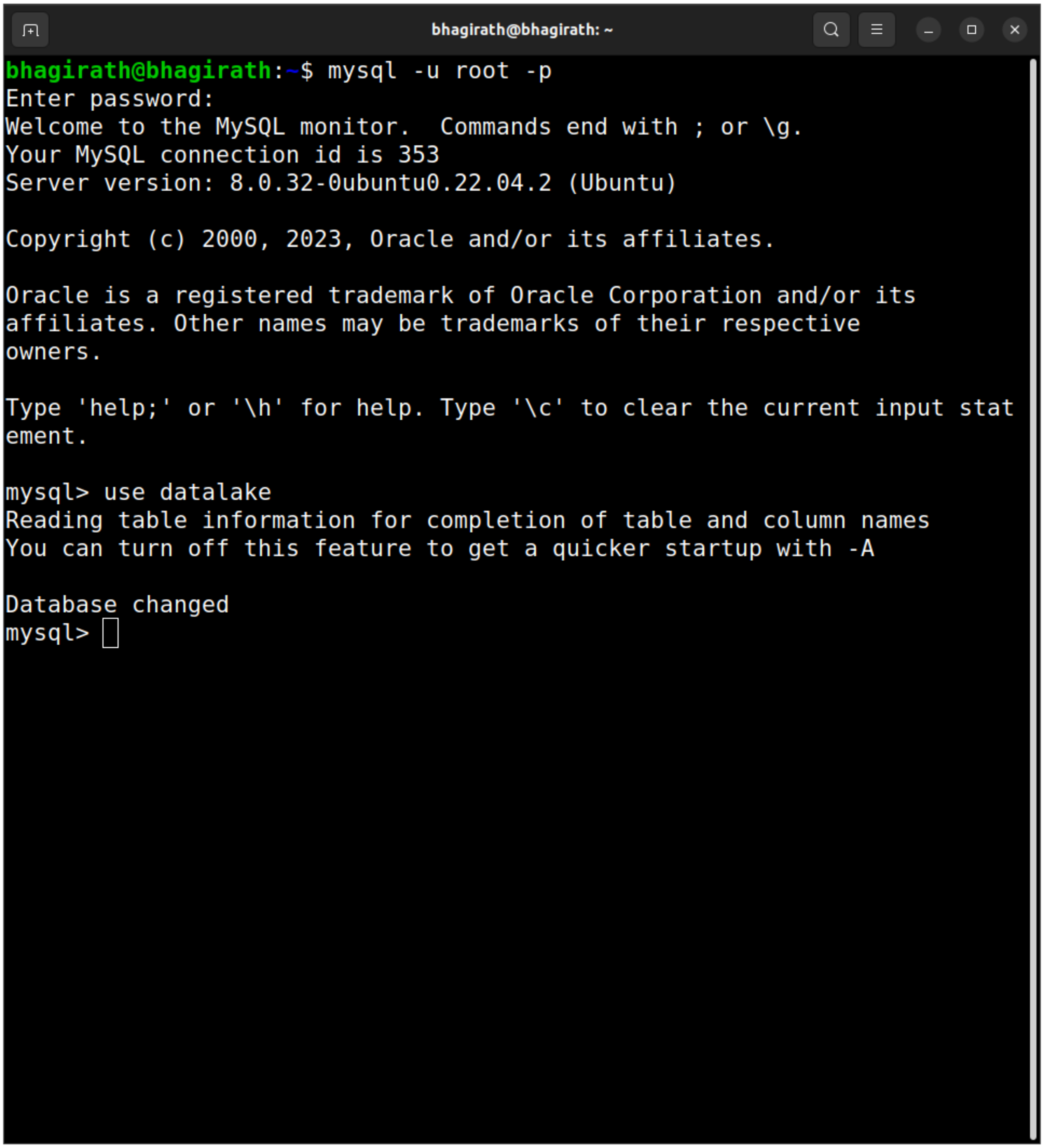
- Here we are setting the MongoDB database and viewing the collection.


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2...

```
datalake> show dbs
admin     40.00 KiB
config    72.00 KiB
datalake  72.00 KiB
local     72.00 KiB
datalake> show collections
user_data
datalake> db.user_data.find({})
[
  {
    _id: ObjectId("64536d502ff3f30a7a9aec73"),
    name: 'asd',
    email: 'asd@ads.com',
    age: '21',
    hobby: 'cricket'
  },
  {
    _id: ObjectId("6453714c6d3eb60c07b18d25"),
    name: 'bhagirath',
    email: 'B@gmail.com',
    age: '22',
    hobby: 'Coding'
  },
  {
    _id: ObjectId("645371856d3eb60c07b18d26"),
    name: 'parth',
    email: 'pg@gmail.com',
    age: '30',
    hobby: 'football'
  },
  {
    _id: ObjectId("645371ab6d3eb60c07b18d27"),
    name: 'gaurav',
    email: 'G@gmail.com',
    age: '24',
    hobby: 'Editing'
  },
  {
    _id: ObjectId("645371df6d3eb60c07b18d28")
```

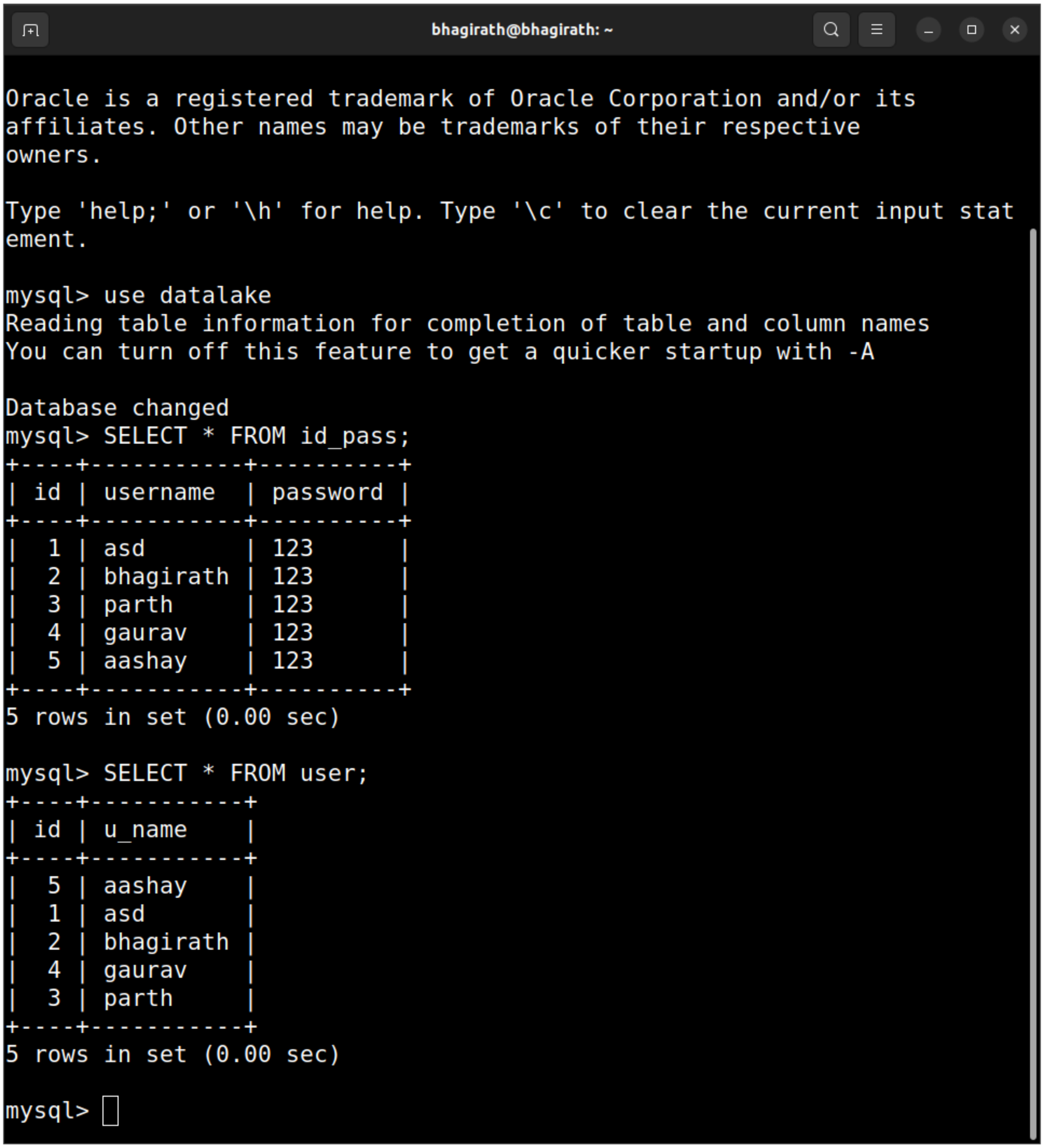
Setting up MySQL

- Here we are setting up a MySQL database.



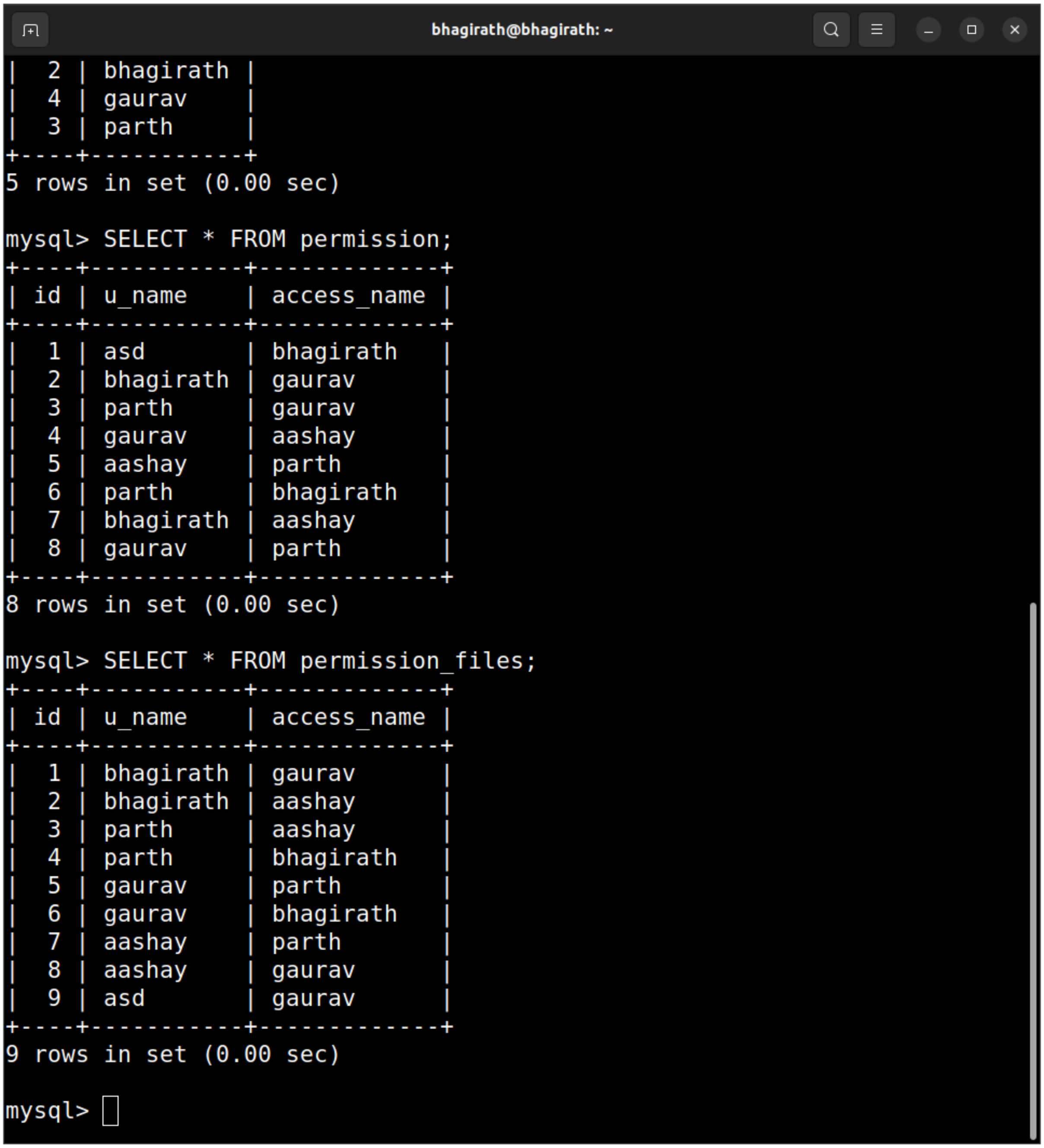
```
bhagirath@bhagirath:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 353
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input stat
ement.

mysql> use datalake
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> 
```

- Now we are viewing our table “users” from MySQL database “datalake”.



```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input stat
ement.

mysql> use datalake
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM id_pass;
+----+-----------+----------+
| id | username  | password |
+----+-----------+----------+
|  1 | asd       | 123      |
|  2 | bhagirath | 123      |
|  3 | parth     | 123      |
|  4 | gaurav    | 123      |
|  5 | aashay    | 123      |
+----+-----------+----------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM user;
+----+-----------+
| id | u_name    |
+----+-----------+
|  5 | aashay    |
|  1 | asd       |
|  2 | bhagirath |
|  4 | gaurav    |
|  3 | parth     |
+----+-----------+
5 rows in set (0.00 sec)

mysql> 
```

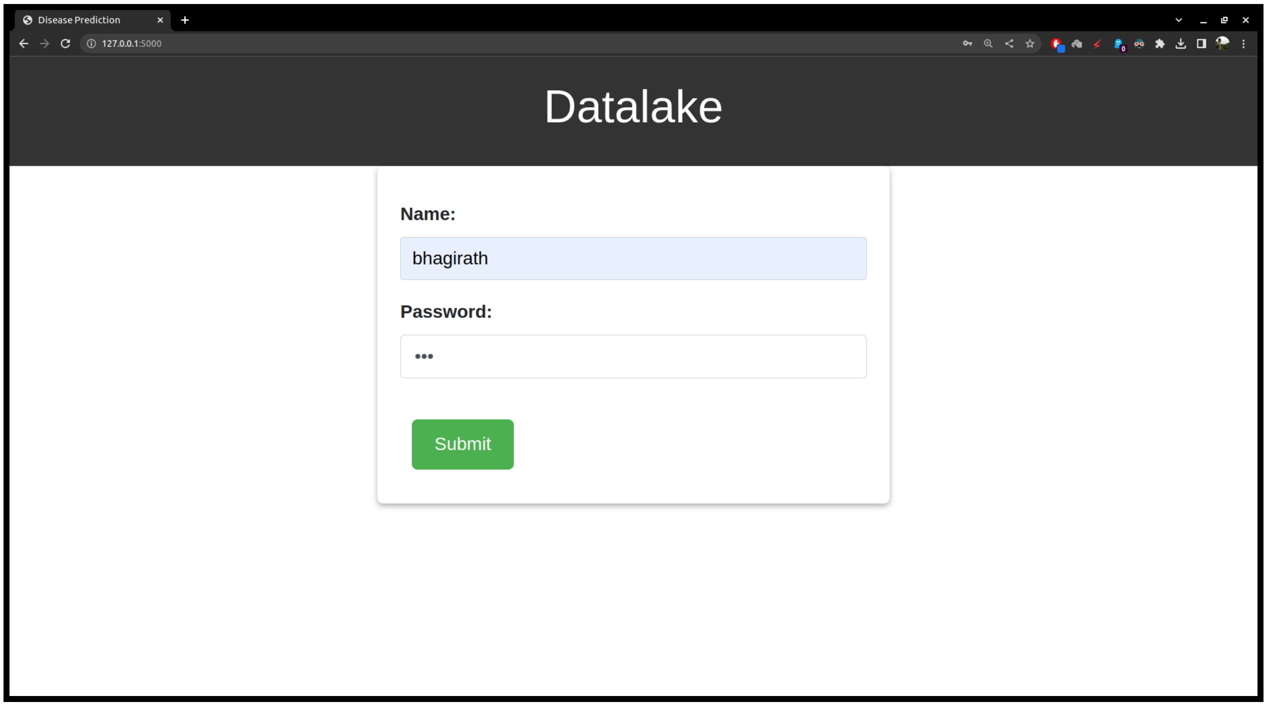
- Here we are viewing database tables “permission” and “permission_files”.



```
|  2 | bhagirath |
|  4 | gaurav    |
|  3 | parth     |
+----+-----------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM permission;
+----+-----------+-------------+
| id | u_name    | access_name |
+----+-----------+-------------+
|  1 | asd       | bhagirath   |
|  2 | bhagirath | gaurav      |
|  3 | parth     | gaurav      |
|  4 | gaurav    | aashay      |
|  5 | aashay    | parth       |
|  6 | parth     | bhagirath   |
|  7 | bhagirath | aashay      |
|  8 | gaurav    | parth       |
+----+-----------+-------------+
8 rows in set (0.00 sec)

mysql> SELECT * FROM permission_files;
+----+-----------+-------------+
| id | u_name    | access_name |
+----+-----------+-------------+
|  1 | bhagirath | gaurav      |
|  2 | bhagirath | aashay      |
|  3 | parth     | aashay      |
|  4 | parth     | bhagirath   |
|  5 | gaurav    | parth       |
|  6 | gaurav    | bhagirath   |
|  7 | aashay    | parth       |
|  8 | aashay    | gaurav      |
|  9 | asd       | gaurav      |
+----+-----------+-------------+
9 rows in set (0.00 sec)

mysql> 
```

Data lake environment UI

- This is the login page of our Datalake application where user can login into.

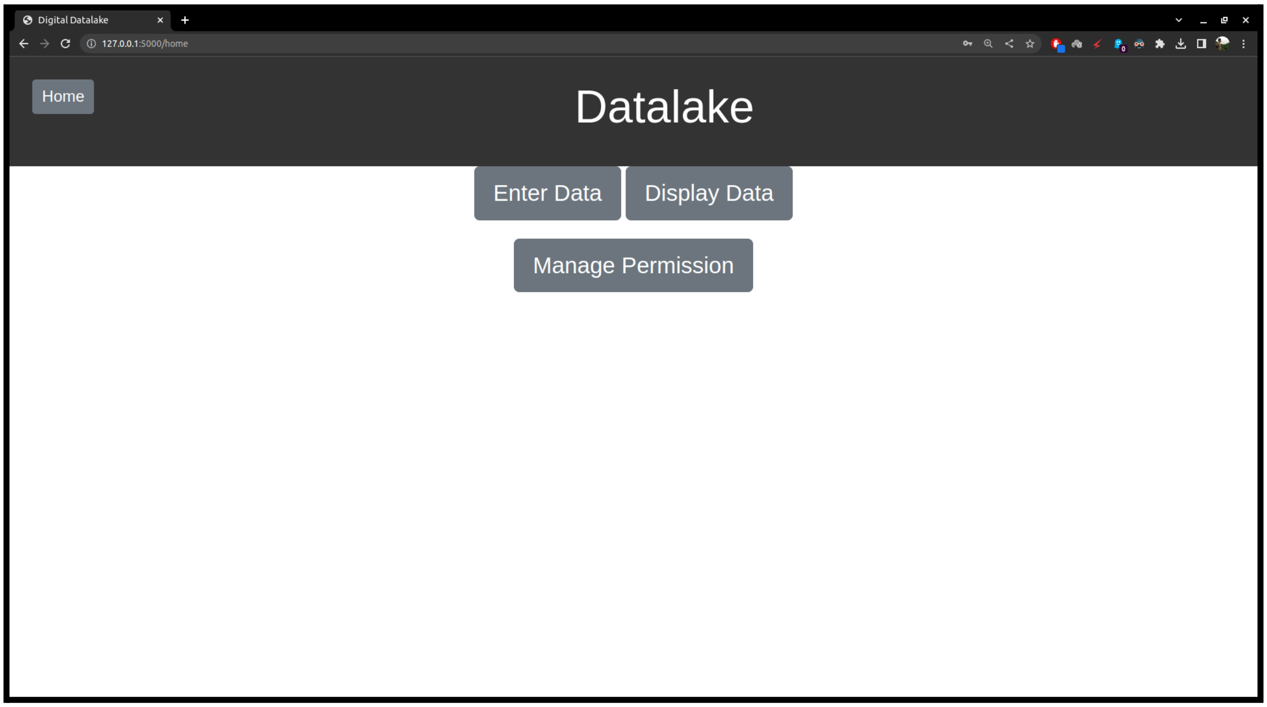


Datalake

Name: bhagirath

Password: •••

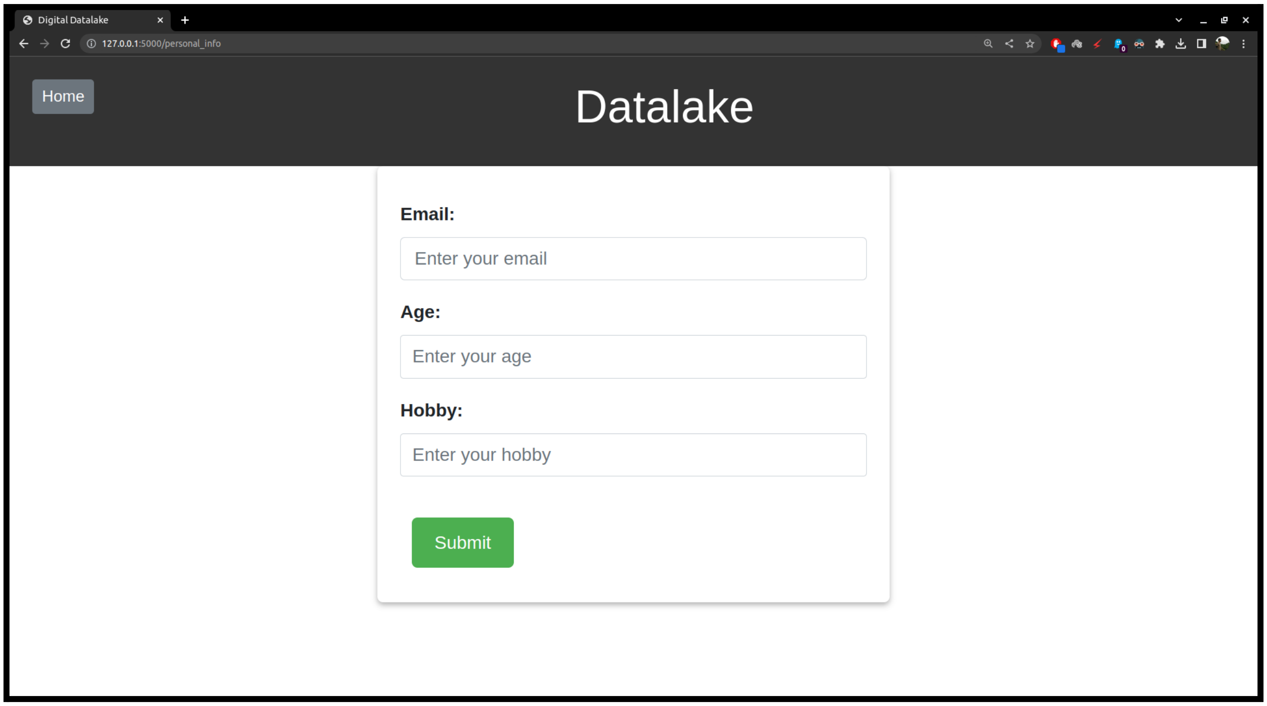
- This is the home page after the successful login where the user can now access the database if he/she has permission.



- We can upload data in two ways into the database.

□ we can simply upload personal information through form.

□ We can upload any file from local storage to the database server.



Datalake

Email:

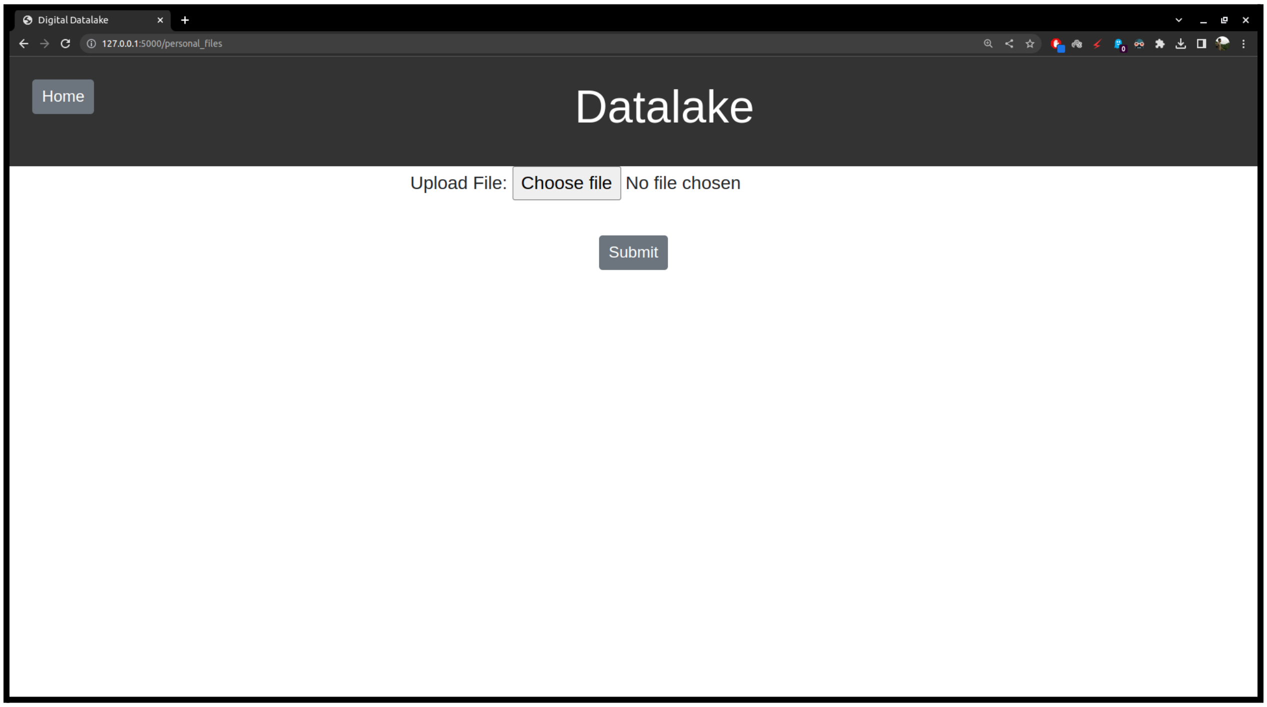
Enter your email

Age:

Enter your age

Hobby:

Enter your hobby



Datalake

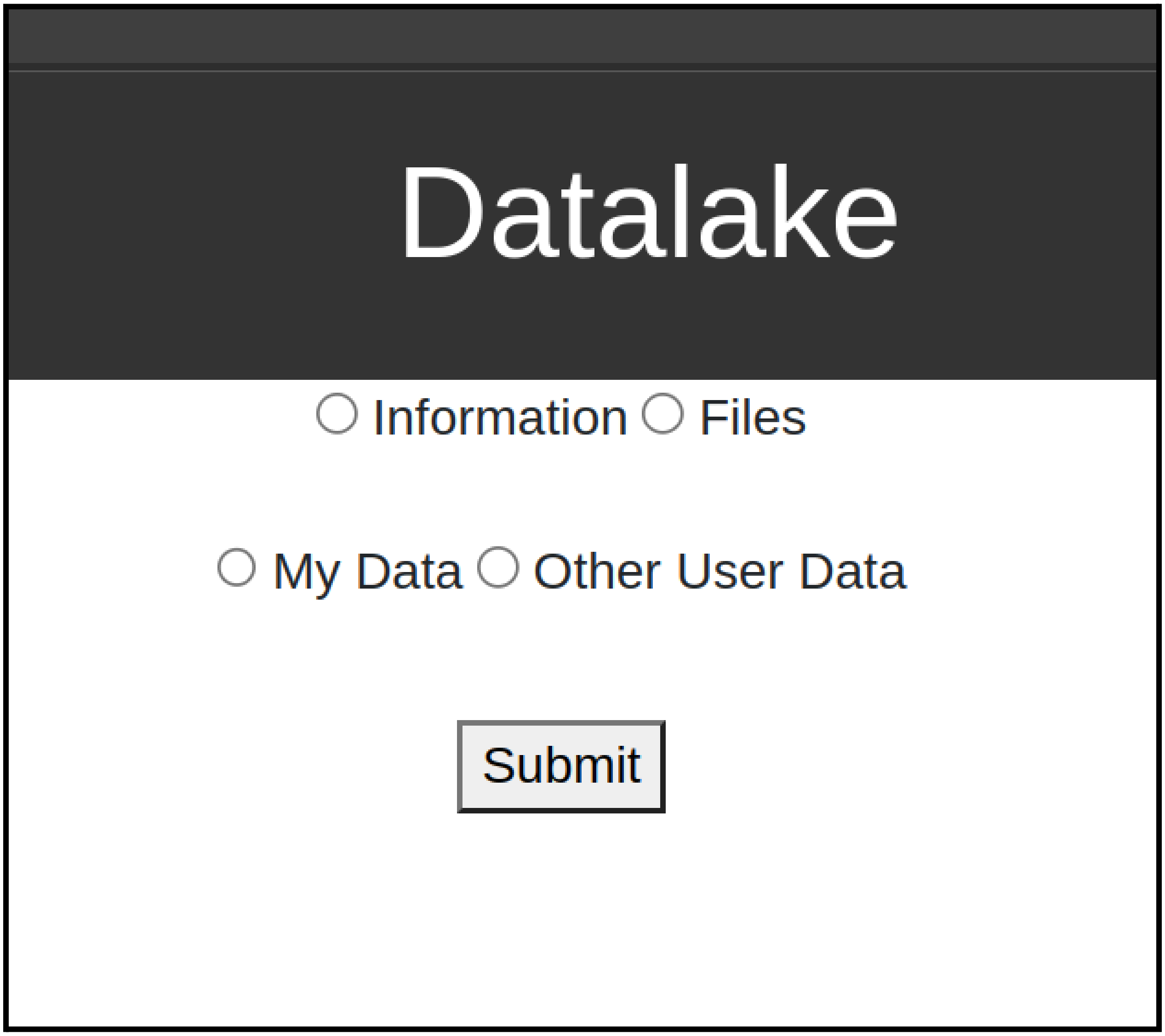
Upload File: Choose file No file chosen

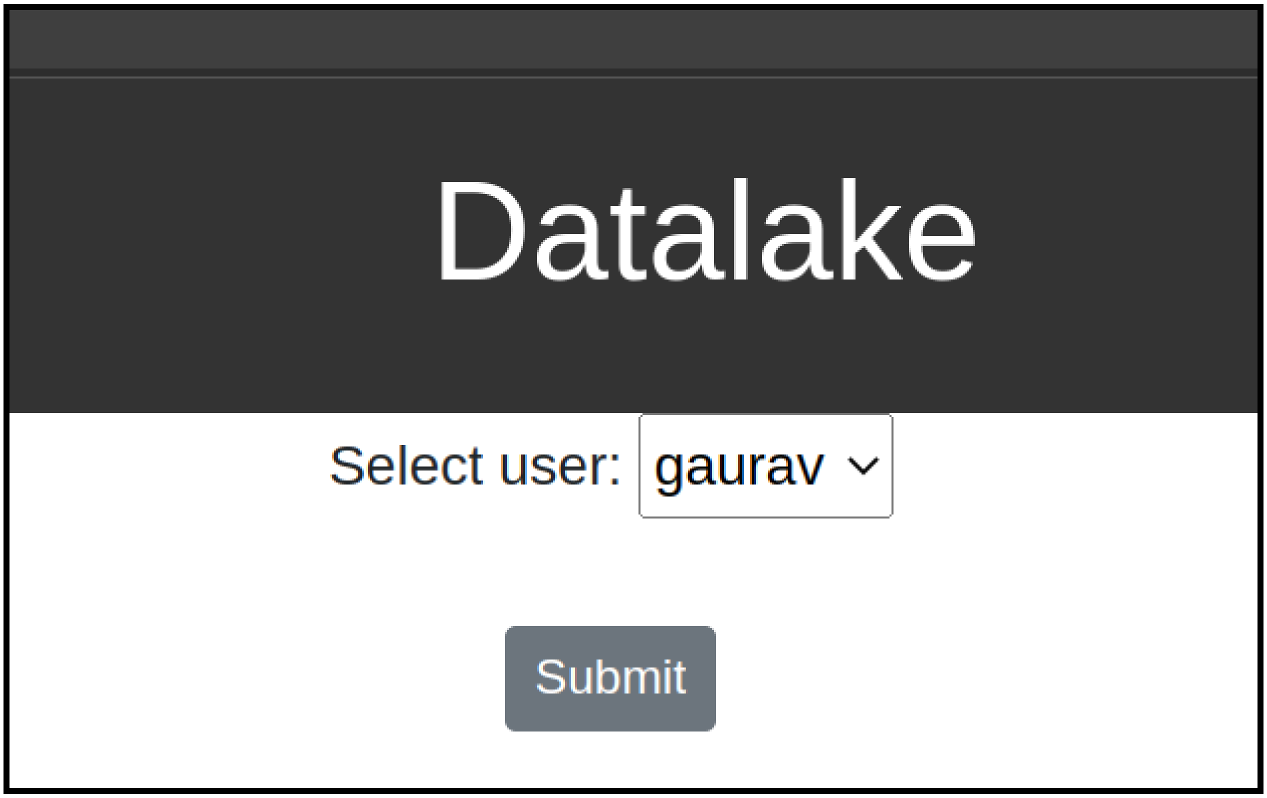
By choosing display data from the home page we can see the databases that are available in the database.

Here we have two options:

□ First option is whether we want to display personal information or files that are stored in database

□ Second option is whether data we want to see from our own database or other user data that you have access to.

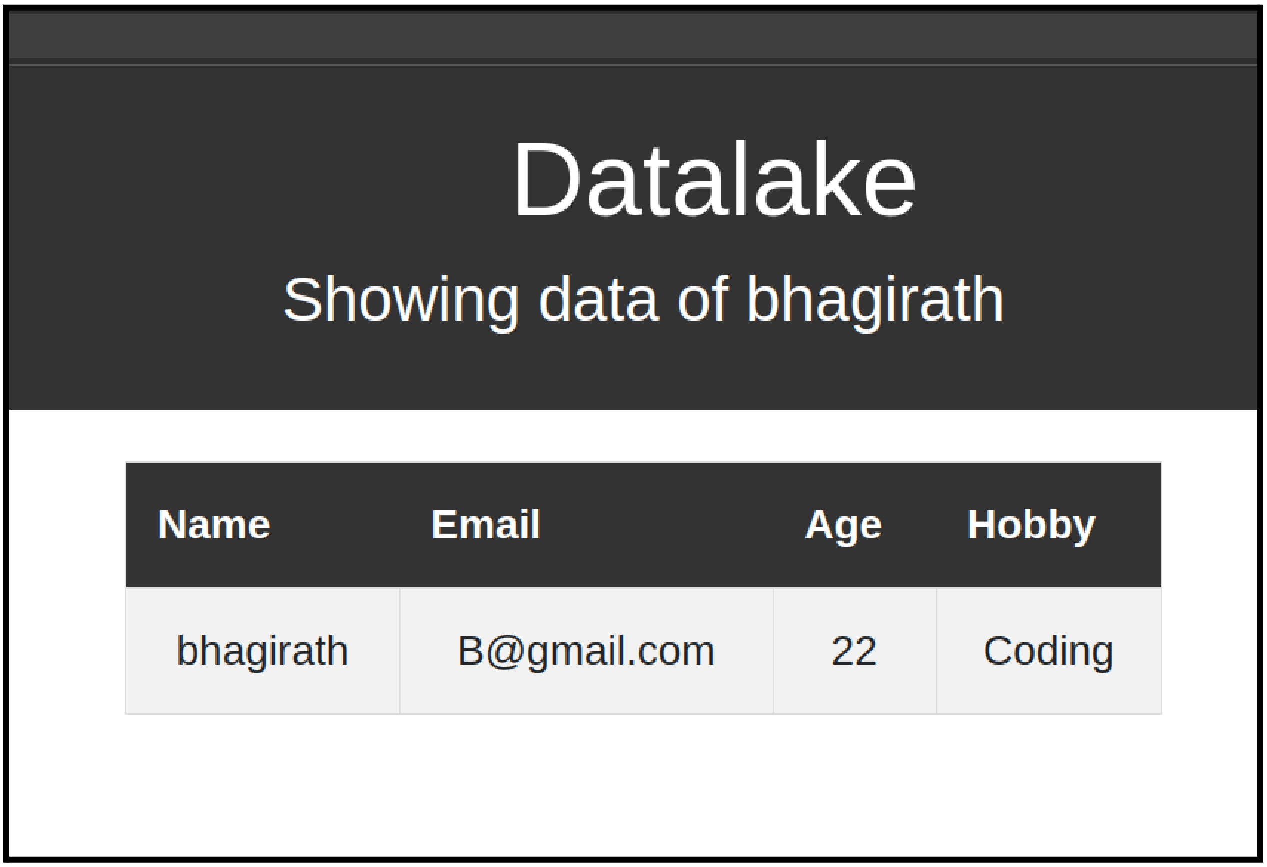




Datalake

Select user: gaurav

- Based on the choice of these two options we can see the data from the database of a particular user.

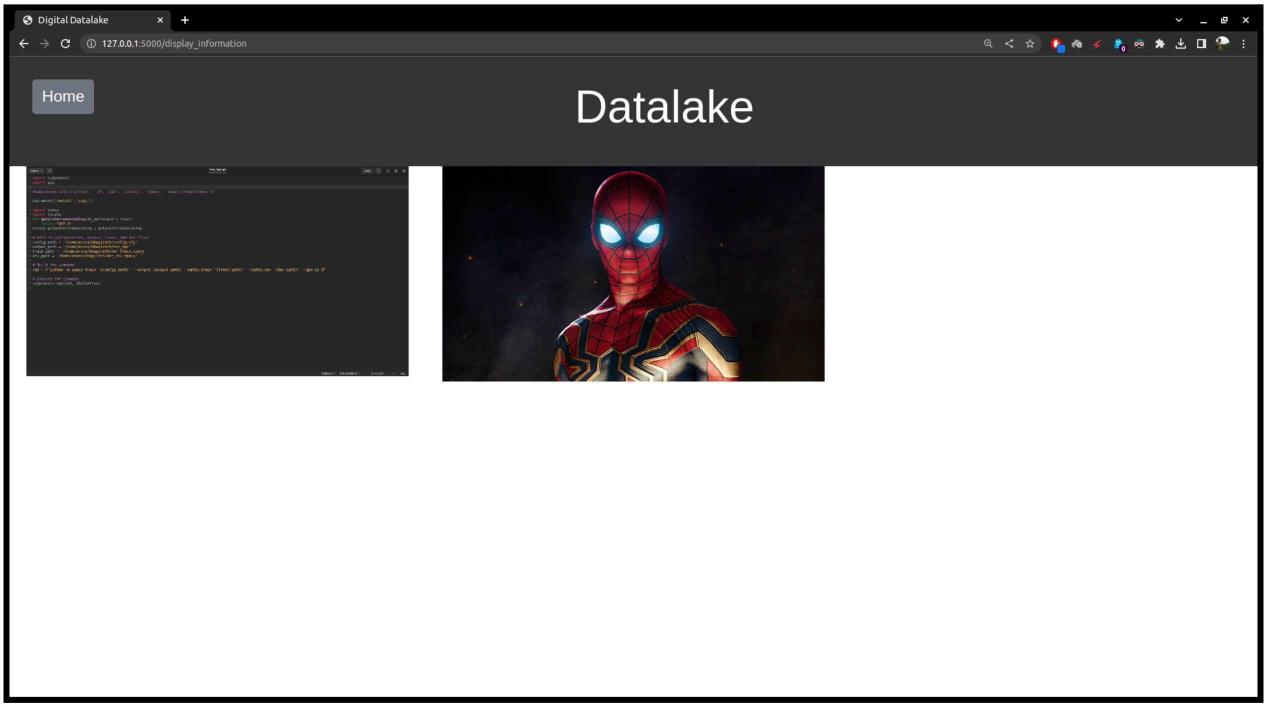


Datalake

Showing data of bhagirath

Name	Email	Age	Hobby
bhagirath	B@gmail.com	22	Coding

- Based on the choice of these two options we can see the files which are fetched from the database.



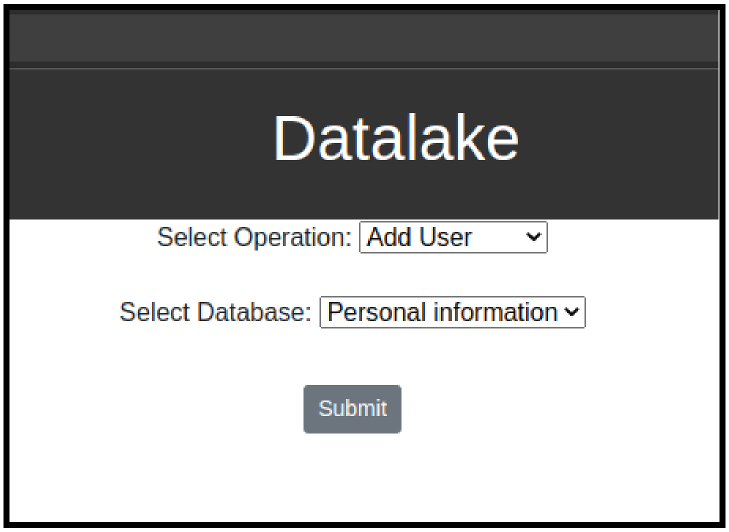
```
import subprocess
import pip

#subprocess.call(['python', '-m', 'pip', 'install', 'spacy', 'spacy-transformers'])

pip.main(['install','cupy'])

import spacy
import locale
def getpreferredencoding(do_setlocale = True):
    return "UTF-8"
locale.getpreferredencoding = getpreferredencoding

# path to configuration, output, train, and dev files
config_path = '/home/aruna/bhagirath/config.cfg'
output_path = '/home/aruna/bhagirath/ner_hpc'
train_path = '/home/aruna/bhagirath/ner_train.spacy'
dev_path = '/home/aruna/bhagirath/ner_dev.spacy'

# Build the command
cmd = f"python -m spacy train '{config_path}' --output {output_path} --paths.train '{train_path}' --paths.dev '{dev_path}' --gpu-id 0"

# Execute the command
subprocess.run(cmd, shell=True)
```

Sharing mechanism and confidentiality

Sharing mechanism

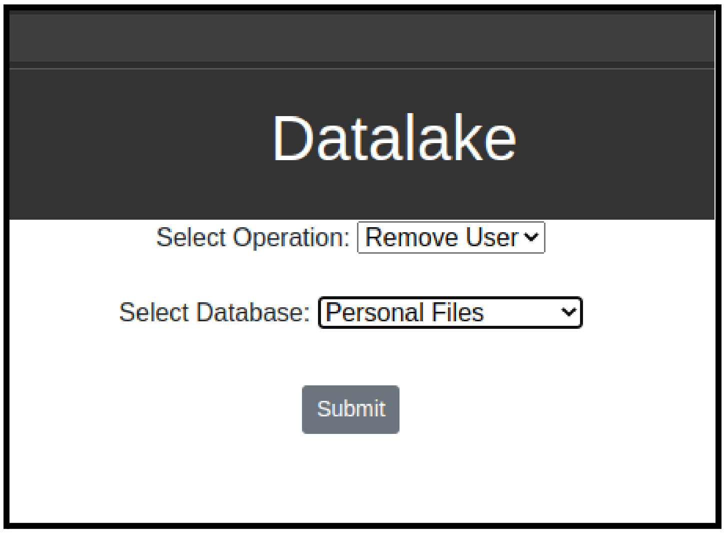
For the sharing mechanism we have one feature which allows users to give or remove access to their data to other users.

First option is that users can choose to add user access permission or remove access permission.
Second option is that from which database users want to add user permission or remove permission.



Select Operation: Add User

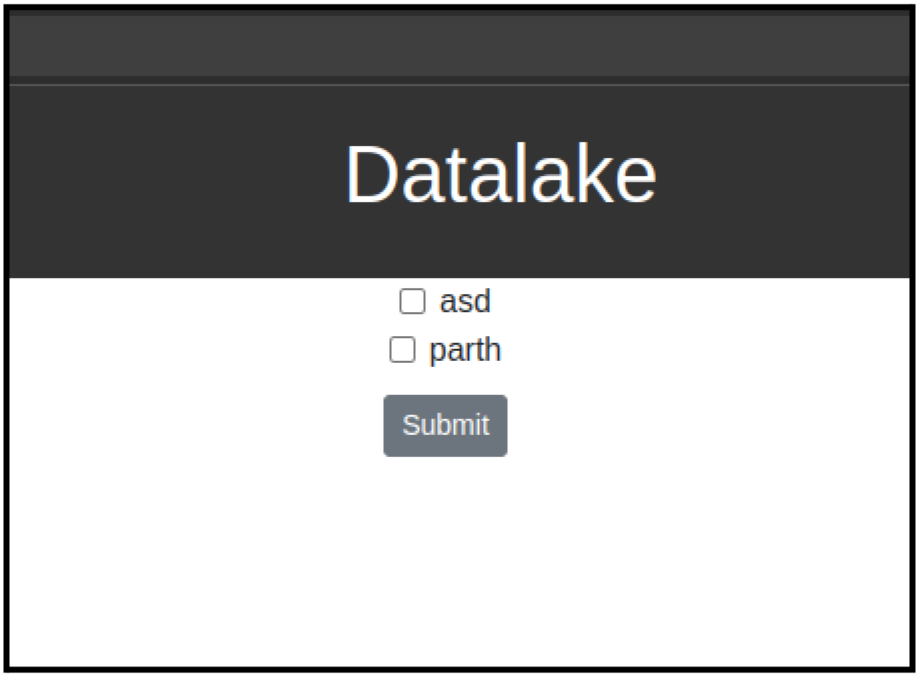
Select Database: Personal information



Select Operation: Remove User

Select Database: Personal Files

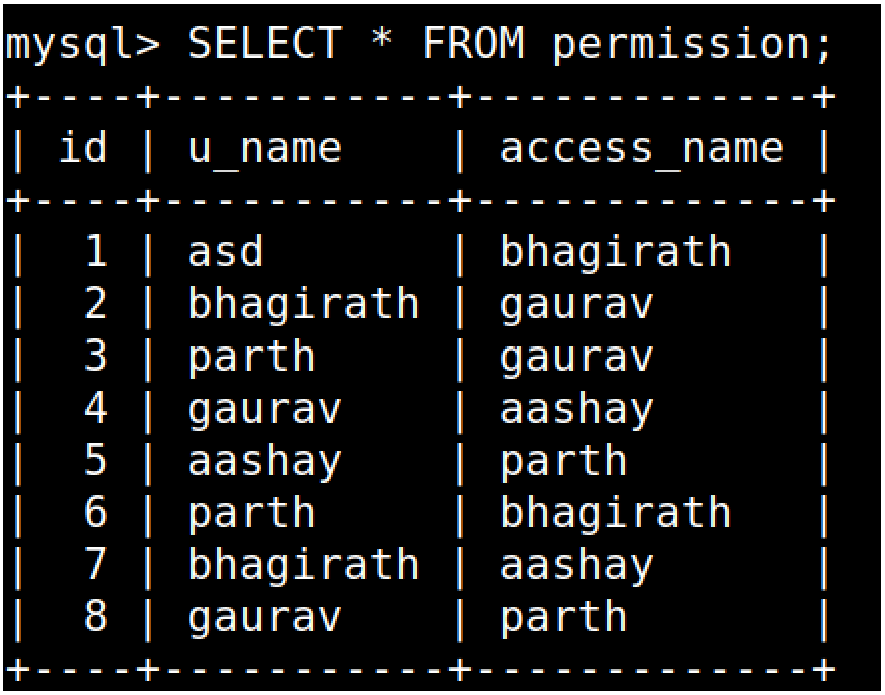
According to the choice of the previous two option user can see for which users it can add user permission or remove permission.



Confidentiality

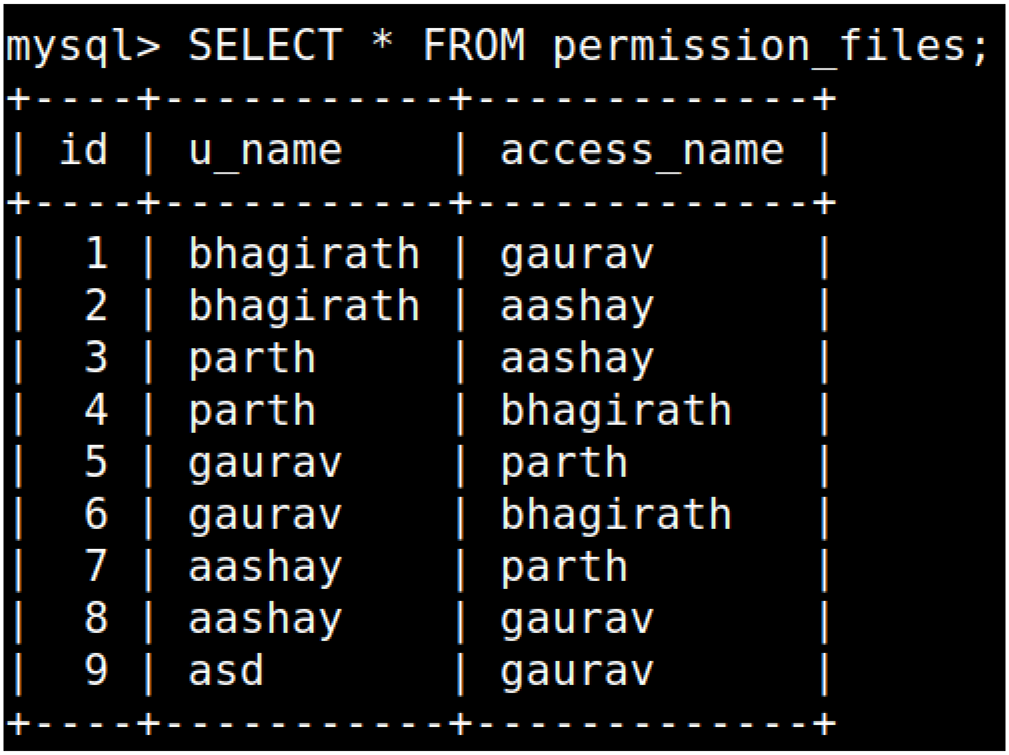
For confidentiality we have two different tables.
Users can access any other user's data if and only if that user has an access entry in the permission table.

Like if one user wants to access any other **user's personal info** then the access entry of that user should be present in the permission table.



```
mysql> SELECT * FROM permission;
+----+-----------+-------------+
| id | u_name    | access_name |
+----+-----------+-------------+
|  1 | asd       | bhagirath   |
|  2 | bhagirath | gaurav      |
|  3 | parth     | gaurav      |
|  4 | gaurav    | aashay      |
|  5 | aashay    | parth       |
|  6 | parth     | bhagirath   |
|  7 | bhagirath | aashay      |
|  8 | gaurav    | parth       |
+----+-----------+-------------+
```

Like if one user wants to access any **other user's files** that are stored on a database server then the access entry of that should be in the permission_files table.



```
mysql> SELECT * FROM permission_files;
+----+-----------+-------------+
| id | u_name    | access_name |
+----+-----------+-------------+
|  1 | bhagirath | gaurav      |
|  2 | bhagirath | aashay      |
|  3 | parth     | aashay      |
|  4 | parth     | bhagirath   |
|  5 | gaurav    | parth       |
|  6 | gaurav    | bhagirath   |
|  7 | aashay    | parth       |
|  8 | aashay    | gaurav      |
|  9 | asd       | gaurav      |
+----+-----------+-------------+
```