

IBM Project

Report

On

Data Redundancy Removal System

Developed By: -

Meet Prajapati (20162121010)

Akshar Patel (20162171013)

Raval Mihir (20162101017)

Guided By:-

Prof. Sonam Singh (Internal)

Mr. Anoj Dixit (External)

Submitted to
Faculty of Engineering and Technology
Institute of Computer Technology
Ganpat University



**Institute of
Computer
Technology**



Year – 2024



CERTIFICATE

This is to certify that the IBM/Industry Project work entitled “**Data Redundancy Removal System**” by Akshar Patel (Enrolment No. 20162171013), Meet Prajapati (Enrolment No. 20162121010), and Mihir Raval (Enrolment No. 20162101017) of Ganpat University, towards the partial fulfillment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, carried out by them in the CSE (CBA/BDA/CS) Department. The results/findings contained in this Project have not been submitted in part or full to any other University/Institute for the award of any other Degree/Diploma.

Name & Signature of Internal Guide

Name & Signature of Head

Place:
ICT -
GUNI

Date:

ACKNOWLEDGEMENT

The IBM project is a golden opportunity for learning and self-development. We consider ourselves very lucky and honored to have so many wonderful people lead us through in completion of this project. First and foremost, we would like to thank Dr. Rohit Patel, Principal, ICT, and Prof. Dharmesh Darji, Head, ICT who gave us an opportunity to undertake this project. Our grateful thanks to Prof. Sonam Singh & Mr. Anoj D (Internal & External Guides) for their guidance in project work Predicting Application Rating of Google Play Store, who despite being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where we would have been without his/her help. The CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

INDEX

Sr. No	Title		Page No.
1.	Vision		4
	1.1.	Introduction	4
	1.2.	Scope	4
	1.3.	Purpose	4
	1.4.	Overview	4
2.	Positioning		5
	2.1.	Business Opportunity	5
	2.2.	Problem Statement	5
3.	User Description		6
	3.1.	Market Demographics	6
	3.2.	User Environment	6
	3.3.	Users Need	6
4.	Product Features		7
	4.1.	Features	7
	4.2	Others Features	8
5.	Constraints and Product Requirements		9
	5.1.	System Requirement	10
	5.2.	Performance Requirement	10
6.	System Design		11
	6.1.	UI Flowchart	11
	6.2.	System Overview Flowchart	12

	6.3	MongoDB Collection Structure	16
	6.4	Figma Design and Screenshots	17
7.		Deployment On AWS Cloud	29
	7.1	Creating an EC2 instance of AWS Cloud	29
	7.2	Using Nginx & Guni corn for load balancing	30
8.		Project Conclusion	31
9		Refrences	32

CHAPTER 1. VISION

1.1 INTRODUCTION

In the ever-evolving landscape of data management, organizations are continually faced with the challenge of navigating through vast volumes of information while ensuring its integrity, security, and efficiency. The "Data Redundancy Removal System" project emerges as a strategic initiative aimed at addressing one of the fundamental issues pervasive in contemporary data architectures – redundancy.

In the digital realm where data proliferation is exponential, redundant data poses not only operational inefficiencies but also risks to security and scalability. This project endeavors to provide a comprehensive solution, leveraging advanced technologies in Big Data Analytics (BDA), fortified by robust Cyber Security measures, and implemented through a Cloud-Based Application (CBA) framework.

1.2 SCOPE

The scope of the Data Redundancy Removal System, integrated with Python and leveraging relevant Python libraries, is to systematically identify and eliminate redundant data within a dataset or database. This system aims to provide a flexible and customizable solution using Python's programming capabilities and data processing libraries for efficient redundancy detection and removal.

1.3 PURPOSE / GOAL

In addition to the previously mentioned goals and objectives, the integration of Python and associated libraries introduces specific focuses:

Customization through Python: Leverage Python's flexibility to allow for customization of redundancy detection algorithms, catering to the unique characteristics of different datasets.

Scalability: Utilize Python's scalability for processing large datasets efficiently, ensuring that the redundancy removal system can handle varying data sizes.

Community Support: Benefit from the extensive Python community and ecosystem for continuous improvement, support, and the availability of diverse libraries.

Integration with ML: Seamlessly integrate machine learning algorithms using Python libraries to enhance the system's ability to intelligently identify and predict redundancy.

1.4 OVERVIEW

In response to the escalating complexities of modern data management, the Data Redundancy Removal System emerges as a transformative solution, strategically designed to optimize data efficiency, fortify security, and unlock the full potential of organizational data assets. This overview provides a comprehensive insight into the key components, features, and anticipated impacts of this innovative project.

CHAPTER 2. POSITIONING

2.1 BUSINESS OPPORTUNITY

In the ever-expanding digital landscape, organizations are faced with a dual challenge – the proliferation of data and the persistence of redundancy within it. The Data Redundancy Removal System not only addresses these challenges but strategically positions itself as a pivotal business opportunity, unlocking unprecedented value in data management.

Cost Savings and Resource Optimization: Redundant data incurs unnecessary costs related to storage, processing, and management. The Data Redundancy Removal System presents a tangible business benefit by reducing these costs through the elimination of superfluous data. The efficient utilization of resources allows organizations to redirect budget allocations toward strategic initiatives.

Enhanced Data Security and Compliance: In an era where data breaches and regulatory compliance are critical concerns, the system provides a business opportunity to fortify data security. By integrating robust cybersecurity measures, organizations not only mitigate security risks but also enhance their compliance posture. This becomes a strategic advantage in building trust and safeguarding sensitive information.

Scalability for Future Growth: The Cloud-Based Application (CBA) framework incorporated into the system offers a distinct business opportunity for scalability. Organizations can seamlessly adapt to evolving data needs, whether it's an expansion of data volumes or the integration of new data sources. This scalability positions the system as a catalyst for future growth and innovation.

Strategic Decision-Making: As data becomes a critical asset for informed decision-making, the Data Redundancy Removal System creates a business opportunity by providing cleaner, more reliable data. Decision-makers can rely on accurate insights, enabling strategic and data-driven decision-making. This positions the system as an enabler of organizational success in a data-centric business environment.

2.2 PROBLEM STATEMENT

Customization Challenge: Existing redundancy removal systems lack the necessary flexibility for customization. Many algorithms struggle to adapt to the unique characteristics of diverse datasets.

Scalability Bottleneck: Current systems face challenges in efficiently processing large datasets. The scalability of redundancy removal algorithms needs improvement to handle varying data sizes effectively.

Integration Gap with Python: Lack of seamless integration with Python hampers the utilization of its extensive community support and ecosystem. Limited availability of diverse libraries for continuous improvement in redundancy removal systems.

Machine Learning Integration Deficiency: Current redundancy removal solutions often lack integration with machine learning algorithms. The absence of intelligent redundancy identification and prediction through machine learning is a notable gap in existing systems. In summary, the problem statement revolves around the development of a redundancy removal system that addresses customization challenges, scalability bottlenecks, integration gaps with Python, and deficiencies in integrating machine learning for intelligent redundancy identification and prediction. The goal is to create a comprehensive solution that overcomes these limitations, offering a more effective and adaptable approach to redundancy removal.

CHAPTER 3. USER DESCRIPTIONS

3.1 MARKET DEMOGRAPHICS

The Data Redundancy Removal System targets a diverse market encompassing industries such as IT, finance, healthcare, manufacturing, and telecommunications. It caters to large enterprises with extensive datasets, mid-sized companies facing scaling challenges, and startups seeking cost-effective data solutions. The global market includes multinational corporations with complex data management needs, while regional markets focus on localized data redundancy issues. The system addresses regulatory requirements in highly regulated industries, adapting to evolving standards in others. It resonates with technology-driven organizations and those undergoing digital transformations, offering modern data solutions. Its applicability extends to organizations dealing with highly sensitive data, providing robust security measures, and to those with moderately sensitive data prioritizing privacy. In a landscape shaped by varying organization sizes, geographical presence, regulatory environments, technological sophistication, and data sensitivity, the Data Redundancy Removal System provides a tailored approach to streamline data, enhance efficiency, and meet the unique challenges of each market segment.

3.2 USER ENVIRONMENT

3.2.1 Clear Layout

Incorporate a clear layout with labeling on your website to make it easier for manager or users to find information. Your website should have a natural flow of information.

3.2.2 Be Efficient

Keeping element to be attractive effective and satisfactory to meet the standard of UI.

3.2.3 Simple and Understandable

Application should be simple user interface and easy to use for user.

3.3 USERS NEED

- **User-Friendly Interface:** Need an intuitive and easy-to-use interface for uploading data files, selecting fields, and viewing redundancy removal results.
- **Customization Options:** Require the ability to customize the display of redundancy removal results based on specific fields or criteria.
- **Efficient Redundancy Analysis:** Need a system that efficiently analyzes data to identify and remove redundancy, providing accurate and reliable results.
- **Security Assurance:** Expect robust security measures to protect sensitive data during the redundancy removal process.
- **Collaboration Features:** Benefit from features that support collaboration, such as sharing workspaces and commenting on data analysis.

CHAPTER 4. PRODUCT FEATURES

4.1 FEATURE

1. User Interface (UI/UX):

File Input Module:

- Allows users to upload files containing data with potential redundancy.

Supports various file formats (e.g., CSV, Excel).

Field Selection Module:

- Enables users to choose fields for redundancy analysis.
- Provides a user-friendly interface for field selection.

Result Display Module:

- Displays the final result after redundancy removal.
- Allows customization of the display based on user-selected fields.

2. Security Integration:

Data Security Module:

- Implements Python libraries for enhanced data security.
- Ensures encryption and secure handling of sensitive information.

3. Redundancy Removal:

Analysis Module:

- Conducts analysis to identify redundant data.
- Utilizes algorithms to detect and mark duplicate entries.

Redundancy Removal Module:

- Identifying Functional Dependencies:

We iterated over each pair of columns in the dataset to check for functional dependencies.

Functional dependencies were identified by assessing whether one column (X) uniquely determines another column (Y). If the combination of values in X uniquely determines the values in Y, a functional dependency is established.

Each identified functional dependency was stored in a dictionary for further analysis.

- Data Normalization:

To achieve normalization, we aimed to eliminate partial dependencies for 2NF or transitive dependencies for 3NF.

Tables were created based on the identified functional dependencies. Each table comprised the primary key and attributes that were functionally dependent on it.

We ensured that each table had a unique primary key by checking if the primary key uniquely identified records in the table.

Finally, normalized tables were written into separate CSV files for further analysis.

Other Functions:

1. Creates a New Table or CSV File with Redundant Entries Removed:

Upon processing a dataset, the Duolicay Removal System is capable of generating a new table or CSV file where redundant entries have been effectively eliminated. This feature aids in streamlining data sets, making them more manageable and conducive to analysis.

2. Ensures Data Integrity and Consistency:

One of the primary objectives of the Duolicay Removal System is to uphold data integrity and consistency. By removing redundant entries, it helps maintain the accuracy and reliability of the dataset, ensuring that it remains trustworthy for analytical and decision-making purposes.

3. Exact Match Data Redundancy Removal:

The system employs an exact match data redundancy removal technique to identify and eliminate duplicate records that are identical in all fields. This precise method ensures that only true duplicates are removed, leaving behind the most relevant and distinct data entries.

4. Fuzzy Method to Remove Data:

In addition to exact match removal, the Duolicay Removal System utilizes a fuzzy method to identify and eliminate redundant entries that may not be exact matches but exhibit similarities. This fuzzy matching algorithm employs techniques such as string similarity or phonetic matching to identify potential duplicates based on similarity thresholds.

5. KNN Clustering Method:

The system implements the K-Nearest Neighbors (KNN) clustering method to group similar data points together. By clustering data based on their proximity in a multidimensional space, the system can identify redundant entries and remove them effectively. This method is particularly useful in scenarios where exact matching or fuzzy methods may not be suitable.

6. CWOA Hospital Data Redundancy Removal Method:

The Duolicay Removal System incorporates a specialized redundancy removal method tailored for healthcare datasets, such as those from hospitals. The CWOA (Chaotic Whale Optimization) method is designed to efficiently identify and eliminate redundant patient records or medical data entries while considering the unique characteristics and complexities of healthcare data.

Column-Based Data Redundancy Removal:

Apart from removing redundant entries across the entire dataset, the system offers column-based redundancy removal capabilities. This feature allows users to specify particular columns or fields where redundancy should be addressed, enabling more targeted and customizable data cleansing processes.

By integrating these diverse functionalities, the Duolicay Removal System provides a comprehensive solution for effectively identifying, managing, and eliminating redundant data entries across various types of datasets, ensuring enhanced data quality and facilitating more accurate and insightful analysis.

4.2 OTHER FEATURE

■ **User Authentication:**

- Implements secure user authentication for system access.

■ **Audit Trail:**

- Logs user activities and changes made to the data.
- Enhances accountability and traceability.

■ **Notification Module:**

- Sends notifications/alerts for successful redundancy removal.
- Alerts users in case of any potential security issues

CHAPTER 5. CONSTRAINTS AND PRODUCT REQUIREMENT

CONSTRAINTS:

The constraints for the project from the perspective of Cyber Security, Big Data Analytics, and Cloud-Based Applications:

Cyber Security Constraints:

- **Data Privacy:** Ensuring that sensitive information within the .csv files is protected from unauthorized access or breaches during upload, processing, and storage. **Secure Communication:** Implementing secure channels for communication between the client uploading the file, the server processing it, and the storage system.
- **Authentication and Authorization:** Verifying the identity of users accessing the system and restricting access to authorized personnel only.
- **Vulnerability Management:** Regularly updating and patching software components to address any potential security vulnerabilities that could be exploited by attackers.
- **Secure Configuration:** Ensuring that the Python code, MongoDB database, and AWS services are configured securely to prevent common security misconfigurations.

Big Data Analytics Constraints:

- **Scalability:** Ensuring that the system can efficiently handle large volumes of data without compromising performance during redundancy removal and processing.
- **Data Quality:** Maintaining the accuracy, consistency, and completeness of the data throughout the redundancy removal process to ensure reliable results.
- **Algorithm Efficiency:** Optimizing the MDNN, exact match, fuzzy match, CWOA match, and selected column removal algorithms to handle big data efficiently and effectively.
- **Resource Utilization:** Efficiently managing computing resources such as CPU, memory, and storage to minimize costs and maximize performance during data processing.

Cloud-Based Applications Constraints:

- **Availability:** Ensuring high availability of the application on AWS to minimize downtime and ensure continuous access for users.
- **Data Transfer Costs:** Minimizing data transfer costs associated with uploading .csv files to the cloud and transferring processed data back to users.
- **Cloud Service Dependencies:** Managing dependencies on AWS services and ensuring compatibility and interoperability with other cloud platforms or services.
- **Compliance:** Adhering to regulatory requirements and compliance standards for data storage, processing, and transmission in the cloud environment.
- **Performance Optimization:** Optimizing the application's performance on AWS by leveraging cloud-native services, auto-scaling, and load balancing to meet user demand efficiently.

These constraints should be considered to ensure the security, efficiency, and reliability of the data redundancy removal system across different domains.

5.1 SYSTEM REQUIREMENT

Software	Windows 7+, Brevo mailer, NGINX
Database	MongoDB 6.0+
FrontEnd	HTML, CSS, JS, JQuery, Bootstrap
BackEnd/Framework	Django, Python

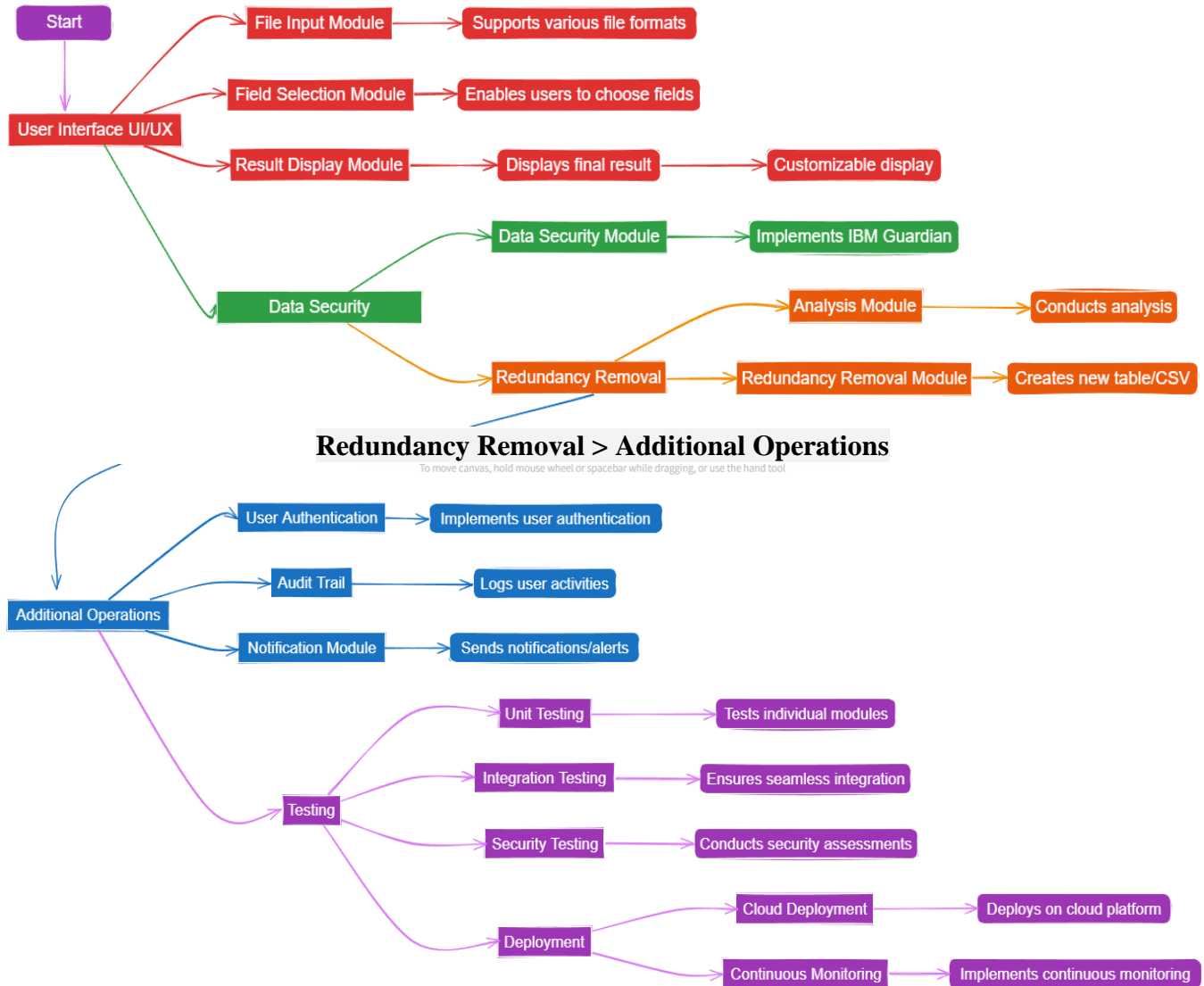
5.2 PERFORMANCE REQUIREMENT

4GB system RAM is enough to run the software, but more memory will increase the speed and performance of the software.

- Chrome/Mozilla v90+ is required for smooth web performance.

CHAPTER 6. SYSTEM DIAGRAM

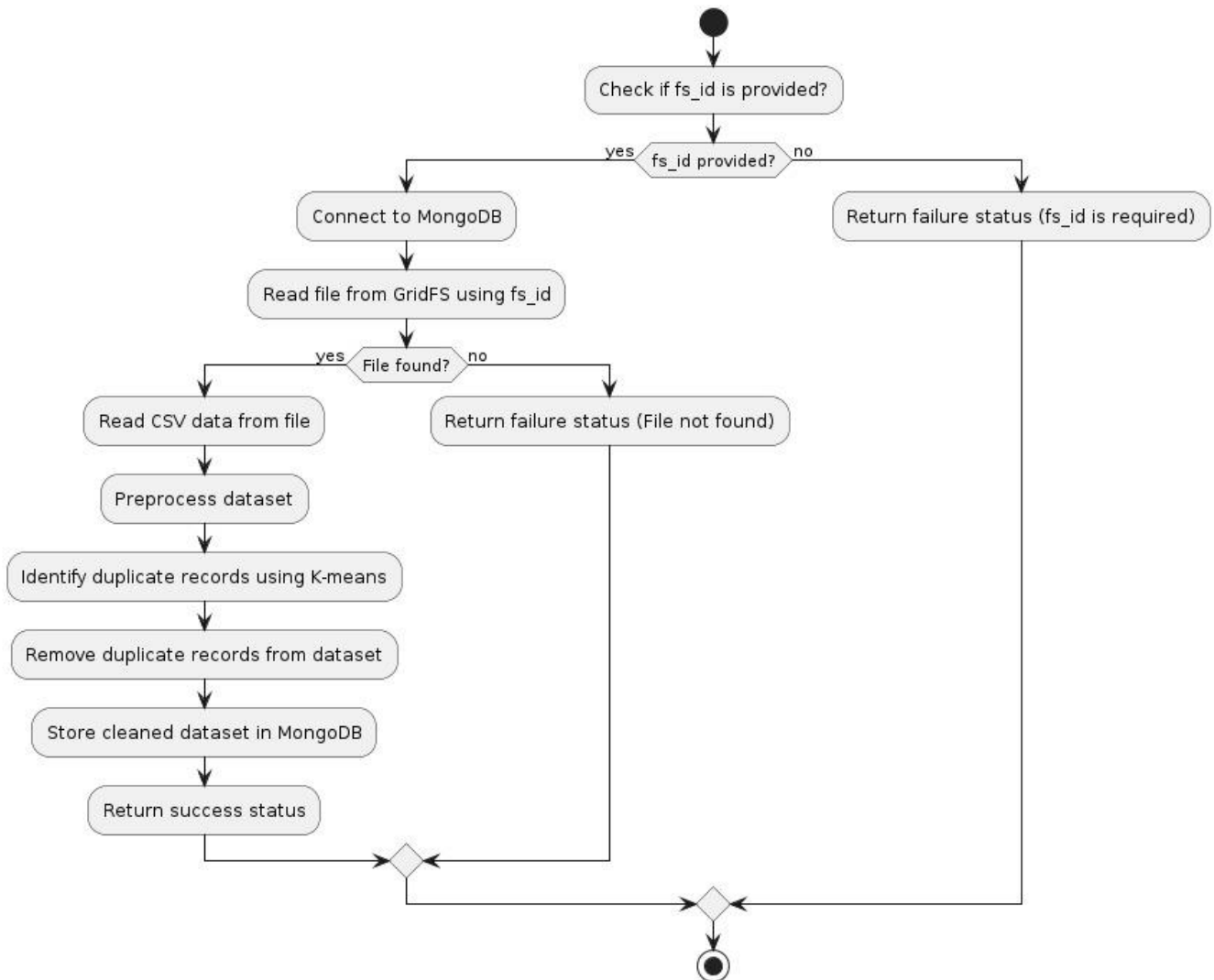
6.1 UI FLOWCHART



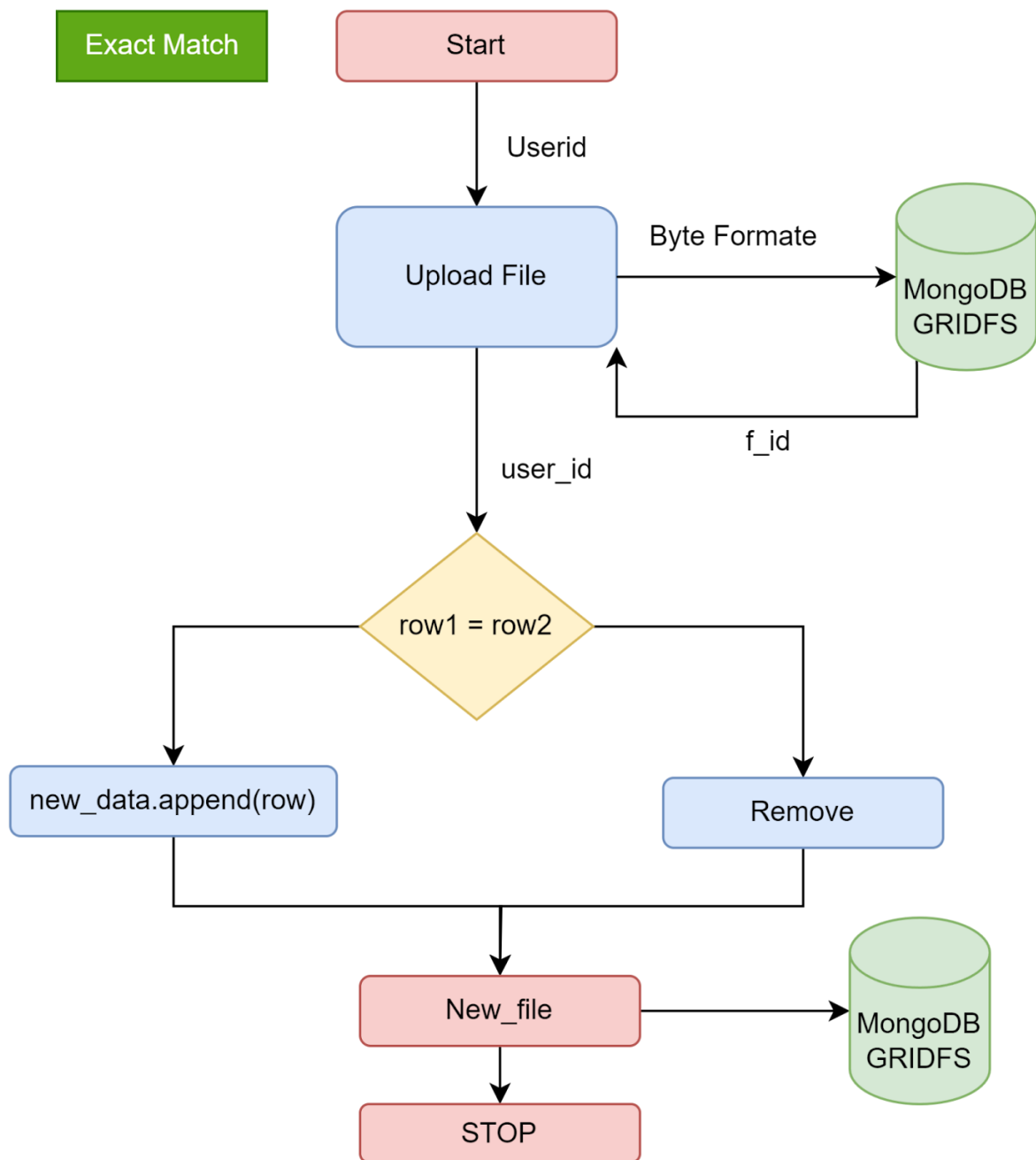
LINK: [Data Redundancy Removal UI Flowchart](#)

6.2 SYSTEM OVERVIEW FLOWCHART

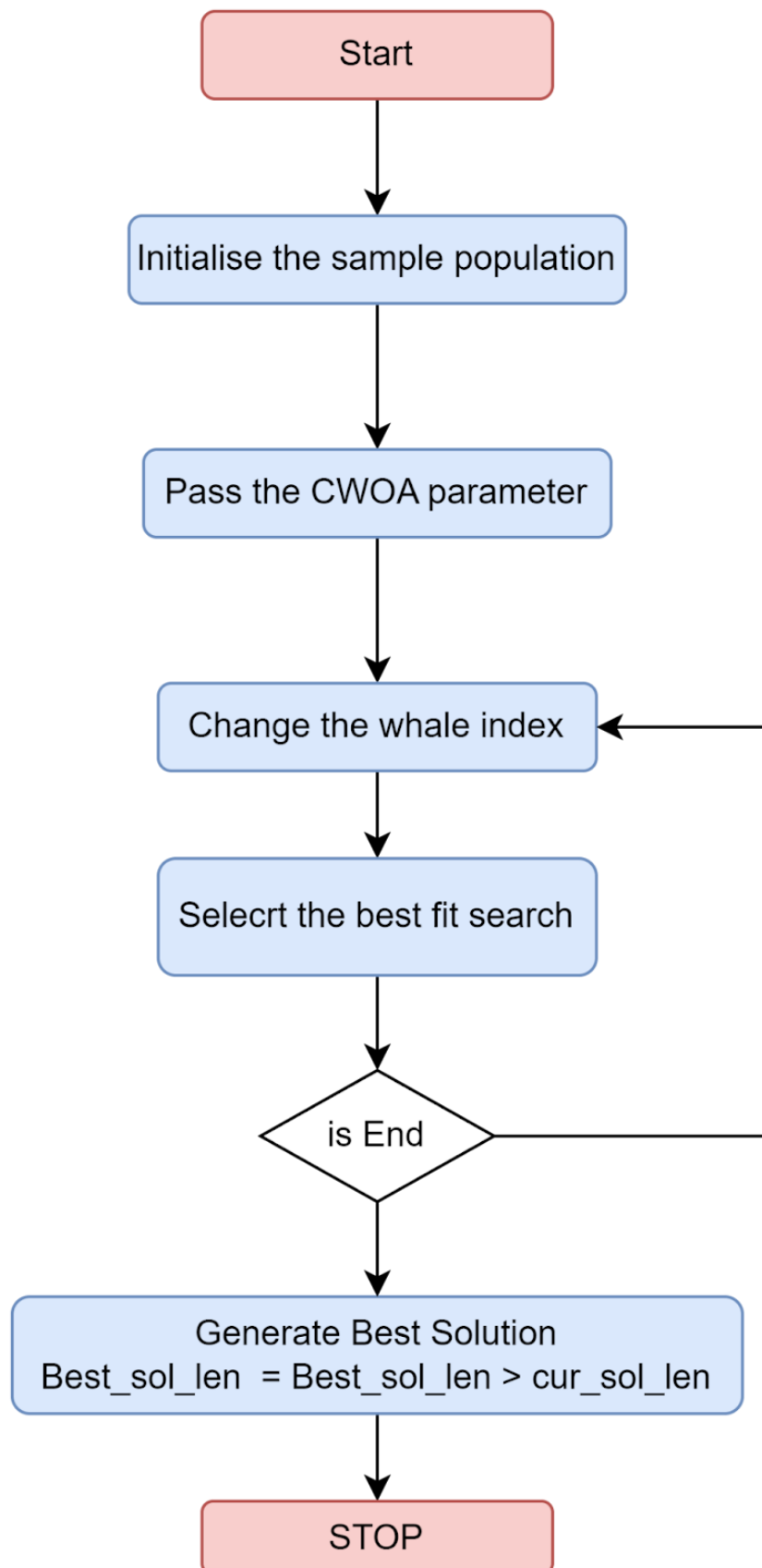
[ML] CSV File Duplicate:



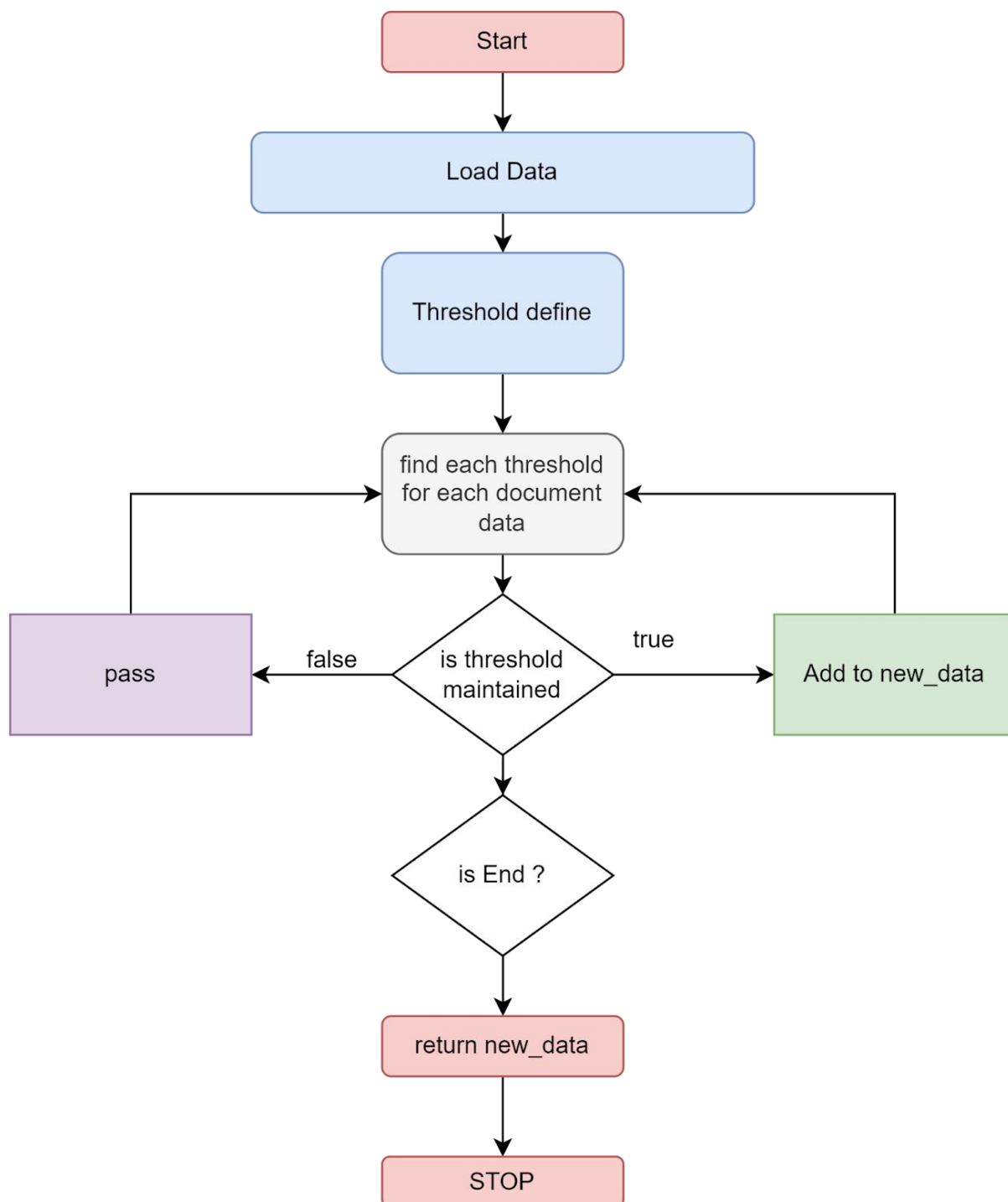
Exact Match Removal:



[Medical Industry Method] CWOA Method:



[Fuzzy Method] Data duplicate removal:



6.3 MongoDB Collection Structure

6.3.1 USERS

- `_id`: user id ObjectID()
- `firstName`: String
- `lastName` : String
- `email`: String
- `phone_number`: String
- `password`: SHA256() hash value
- `occupation`: String
- `ProfilePhoto`: Blob()
- `request`: [{
 `fileUploadedAt`: Date(),
 `file_id`: <GridFS_id>,
 `changes`: [undo_perform, remove column, data_type_change],
}]
- `userCreatedAt`: Date()

6.3.2 GridFS database

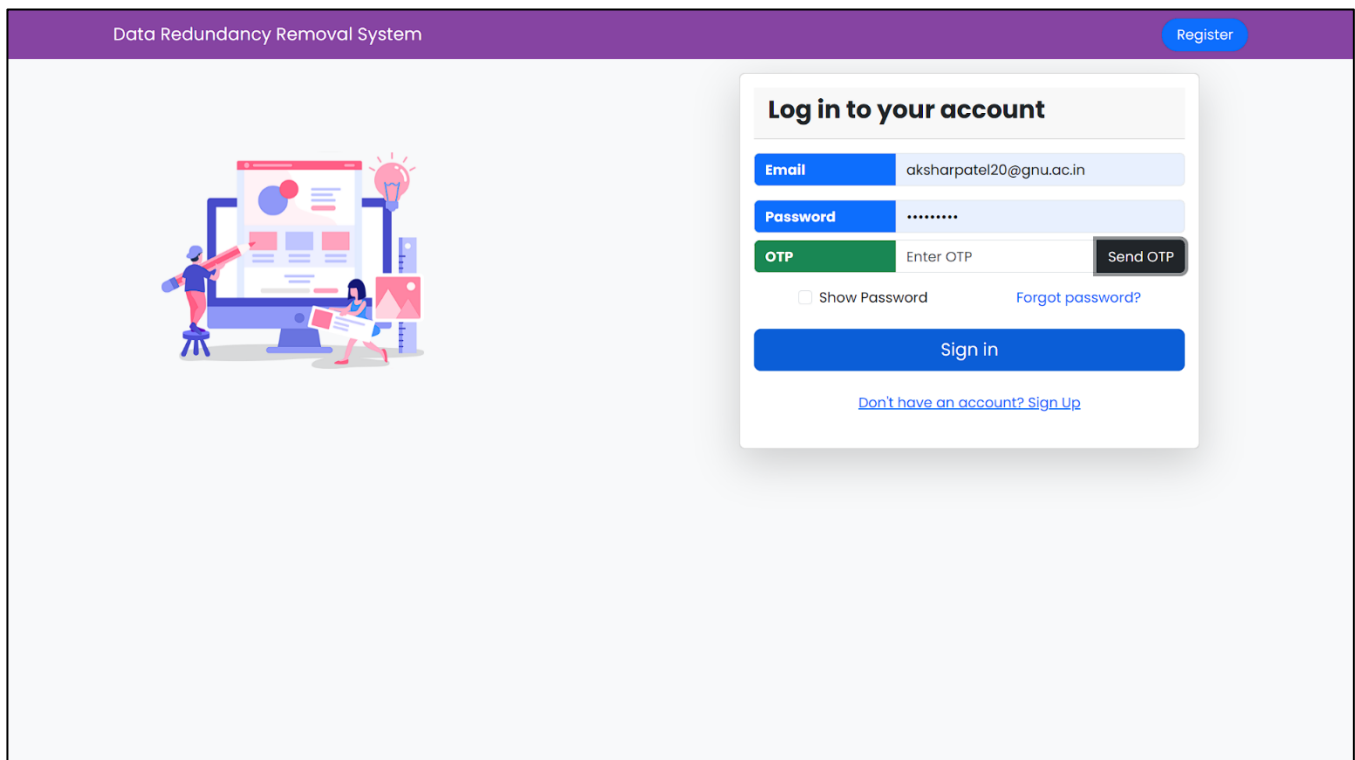
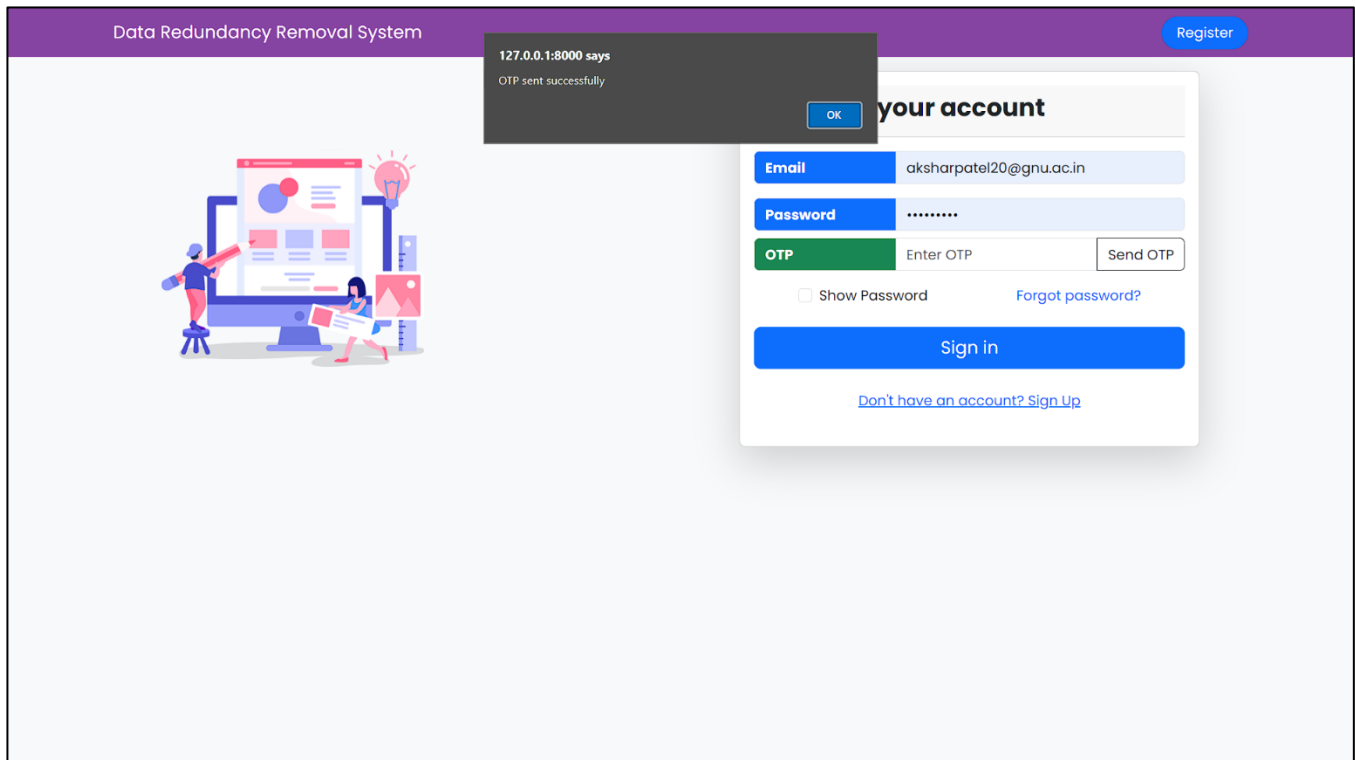
- `fileMetadata` { `file_id`, `finalname`, `datatype`, `storageType` }
- File Chunks

6.3.3 log_storage

- `log_id`
- `log_title`
- `log_content`
- `createdAt`
- `userid`
- `module_title`

6.4 FIGMA DESIGN

6.4.1 Login Page





Log in to your account

Email aksharpatel20@gnu.ac.in

Password

OTP A07D07

Send OTP

☐ Show Password

[Forgot password?](#)

Sign in

[Don't have an account? Sign Up](#)

6.4.2 Registration Page:



Create your account

First Name Akshar

Last Name Patel

Email aksharpatel20@gnu.ac.in

Phone 4512451245

Password

☐ Show Password

[Forgot password?](#)

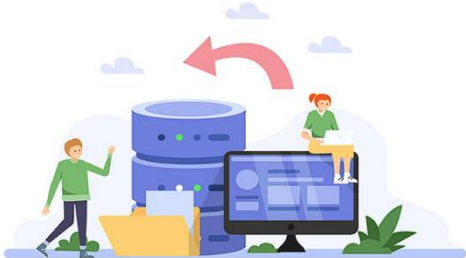
Sign Up

[Already account created? Login](#)

6.4.3 Home Page:

DRRSHomeFeaturesGet Started

Akshar Patel




Data Redundancy Removal System

This system is designed to help you identify and eliminate redundant data within your database. Redundant data can lead to inefficiencies and inaccuracies, so it's important to regularly clean and optimize your database.


Take control of your data and streamline your operations with the Data Redundancy Removal System!

About Us


Features



CWOA




Exact Matching



Fuzzy Matching

DRRSHomeFeaturesGet Started


Akshar Patel



CWOA


An approach to remove duplication records in healthcare dataset based on Mimic Deep Neural Network (MDNN) and Chaotic Whale Optimization

Read More



Exact Matching

Identify and remove exact duplicate records from your database to improve data quality and reduce errors.




Fuzzy Matching

Find and eliminate similar records using fuzzy matching algorithms, which can identify records that are similar but not identical.

Get Started

Drag & Drop Files Here

 DRRS

[Home](#) [Features](#) [Get Started](#)

Akshar Patel

CWOA

An approach to remove duplication records in healthcare dataset based on Mimic Deep Neural Network (MDNN) and Chaotic Whale Optimization

Read More

Exact Matching

Identify and remove exact duplicate records from your database to improve data quality and reduce errors.

Fuzzy Matching

Find and eliminate similar records using fuzzy matching algorithms, which can identify records that are similar but not identical.

Get Started

Drag & Drop Files Here

or

Choose File


No file chosen

View Description

Duplicate Removal

File Description

File Name	-
-----------	---

 DRRS

[Home](#) [Features](#) [Get Started](#)

Akshar Patel

View Description

Duplicate Removal

File Description

File Name	-
File Type	-
Size	-
No. of Rows	-
No. of Column	-

Column Description

No data available

Numeric Column

No data available

DRRS
Home
Features
Get Started
Akshar Patel

Choose File
No file chosen

View Description

Duplicate Removal

Exact match removal

Extract Unique Records

MDNN match removal

Extract Unique Records

Start CWOA

Start CWOA

- Step 1: Initialize Whales
- Step 2: Execute CWOA
- Step 3: Store Unique Records

For seleted columns

Extract Unique Records

Fuzzy match removal

Threshold

0.8

Extract Unique Records

6.4.4 Edit Profile:

DRRS
Home
Features
Get Started
Akshar Patel

Profile
Logout

Data Redundancy Removal System

This system is designed to help you identify and eliminate redundant data within your database. Redundant data can lead to inefficiencies and inaccuracies, so it's important to regularly clean and optimize your database.

Take control of your data and streamline your operations with the Data Redundancy Removal System!


About Us

Features

CWOA

Exact Matching


Fuzzy Matching

 DRRS

[Home](#) [Features](#) [Get Started](#)

Akshar Patel

Welcome, aksharpatel20@gnu.ac.in!



Email

aksharpatel20@gnu.ac.in

First Name

Akshar

Last Name

Patel

Mobile Number

5412541745

Joined

2024-05-02 13:24:48

Last Updated

May 2, 2024, 1:24 p.m.


Last Login

First Login

Edit Profile

History


File name	Date	Actions
No files uploaded		

 DRRS

[Home](#) [Features](#) [Get Started](#)

Akshar Patel

Welcome, aksharpatel20@gnu.ac.in!



Email

aksharpatel20@gnu.ac.in

First Name

Akshar

Last Name

Patel

Mobile Number

5412541745

Joined

2024-05-02 13:24:48

Last Updated

May 2, 2024, 1:24 p.m.

Last Login

First Login

Save

Cancel

History

File name	Date	Actions
No files uploaded		

DRRS

Home

Features

Get Started

127.0.0.1:8000 says

File uploaded successfully.

OK

Choose File

test.csv

View Description

Duplicate Removal

File Description

File Name	-
File Type	-
Size	-
No. of Rows	-
No. of Column	-

Column Description

No data available

DRRS

Home

Features

Get Started

qwert qwsa

View Description

Duplicate Removal

File Description

File Name	test.csv
File Type	csv
Size	1.36 KB
No. of Rows	41
No. of Column	6

Column Description

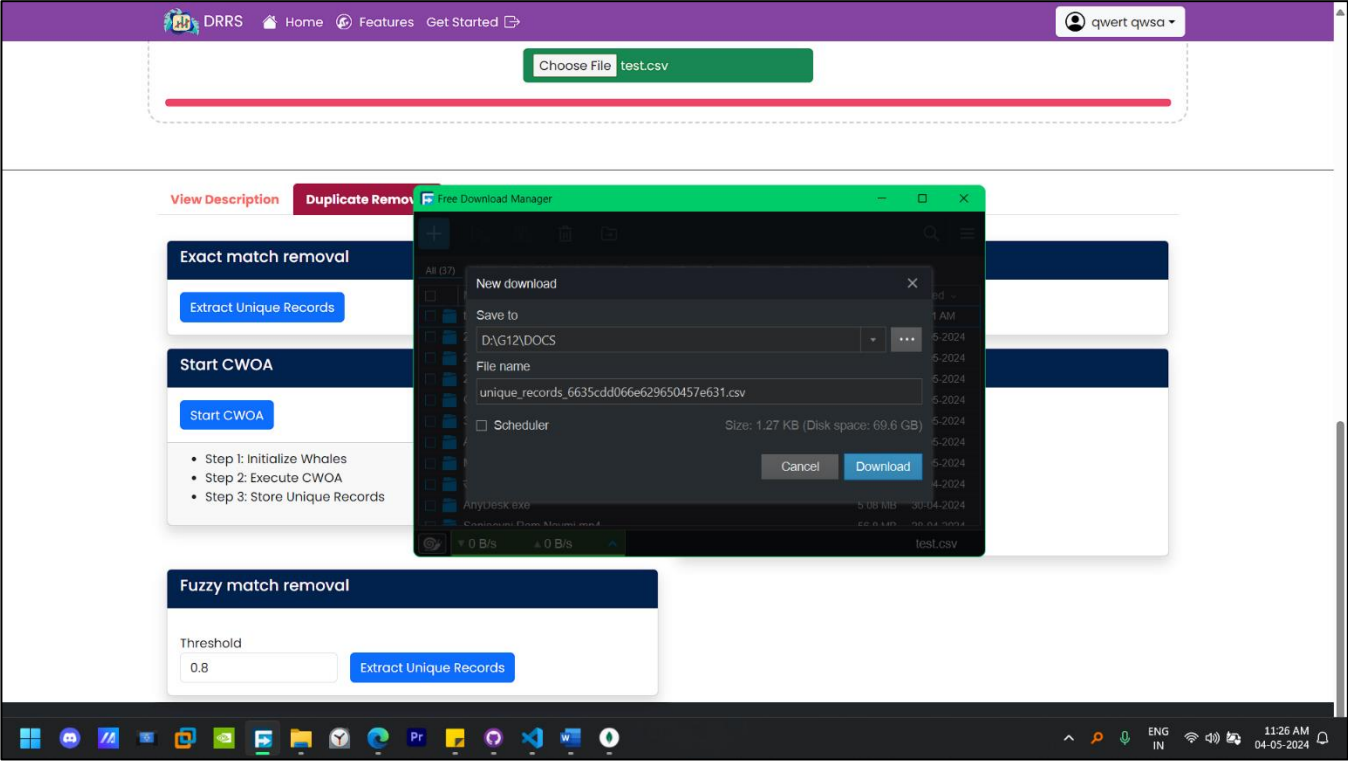
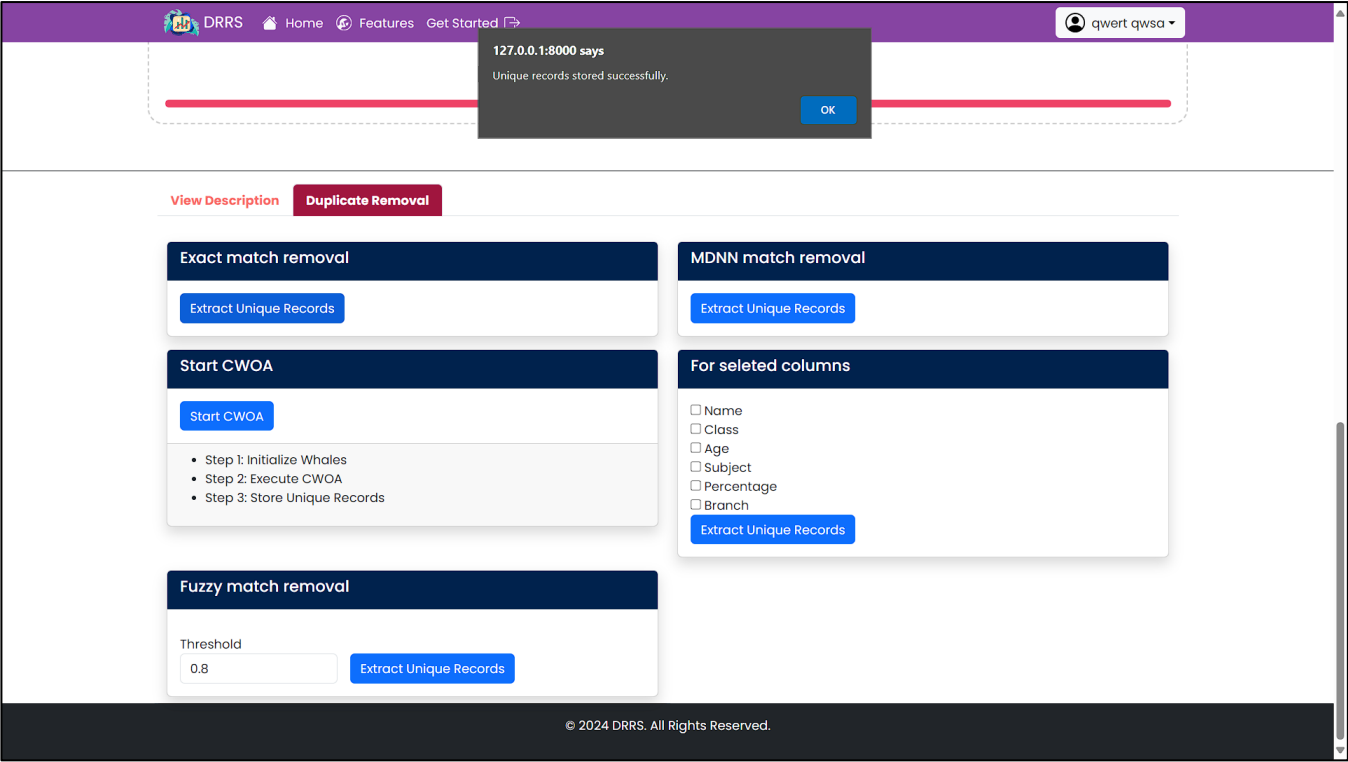
Column Name	Data Type	Null	Unique
Name	object	0	38
Class	int64	0	3
Age	int64	0	3
Subject	object	0	38
Percentage	int64	0	10
Branch	object	0	3

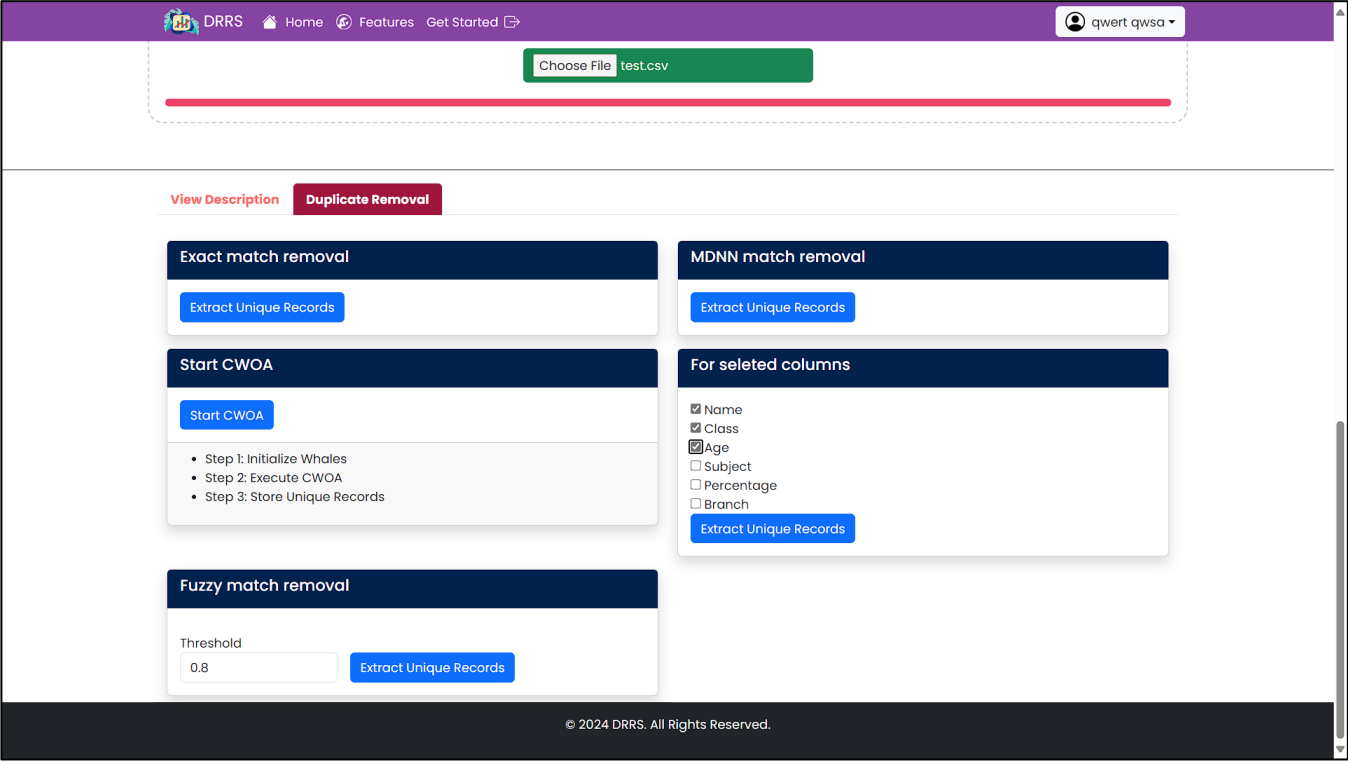
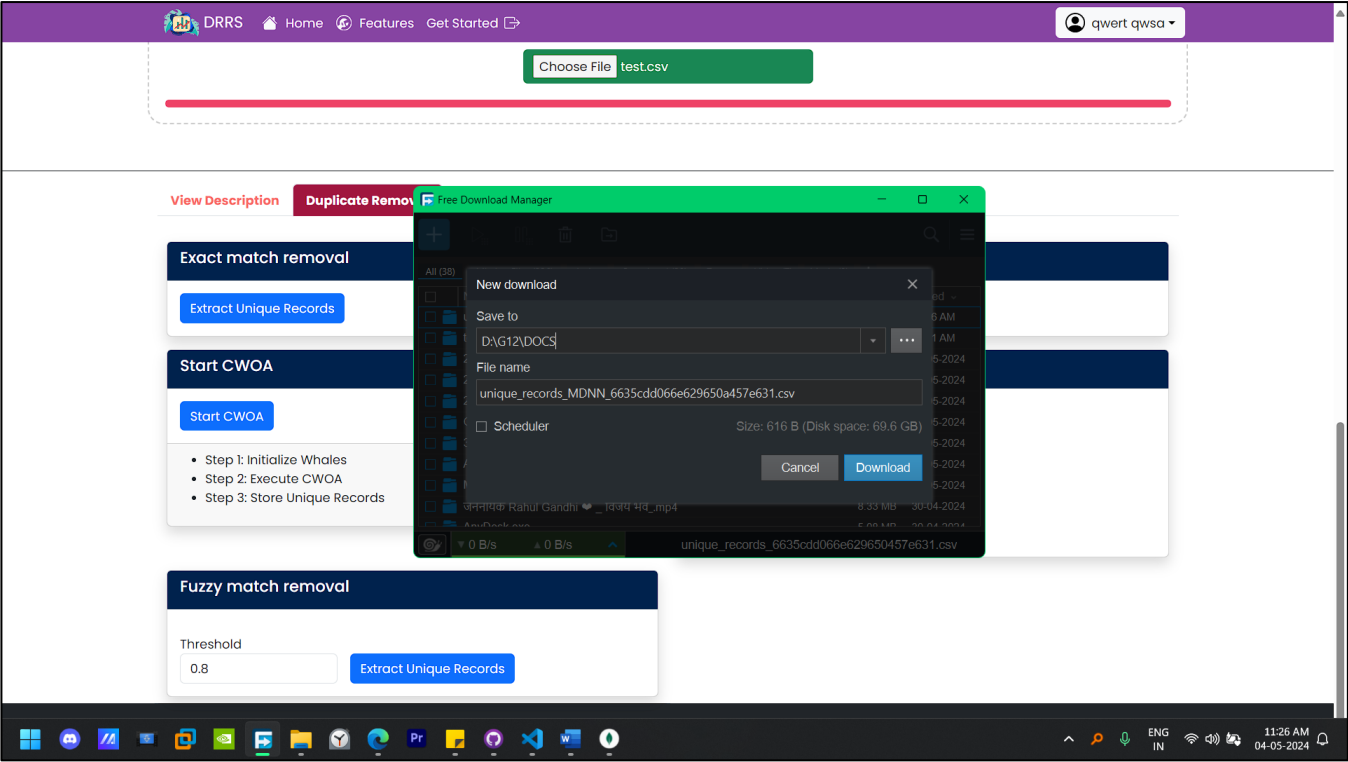
Numeric Column

Column Name	Max	Min	Mean	Median	Mode	Standard Deviation	Skewness
Class	12	10	10.98	11.0	10	0.82	0.05
Age	17	15	15.98	16.0	15	0.82	0.05
Percentage	94	85	89.76	90.0	90	2.78	-0.17

© 2024 DRRS. All Rights Reserved.

Page | 23





MongoDB Collection

The screenshot shows the MongoDB Compass interface for a local host at port 27017. The left sidebar displays the database structure, with the 'drr' database selected. The main panel shows a list of collections within 'drr':

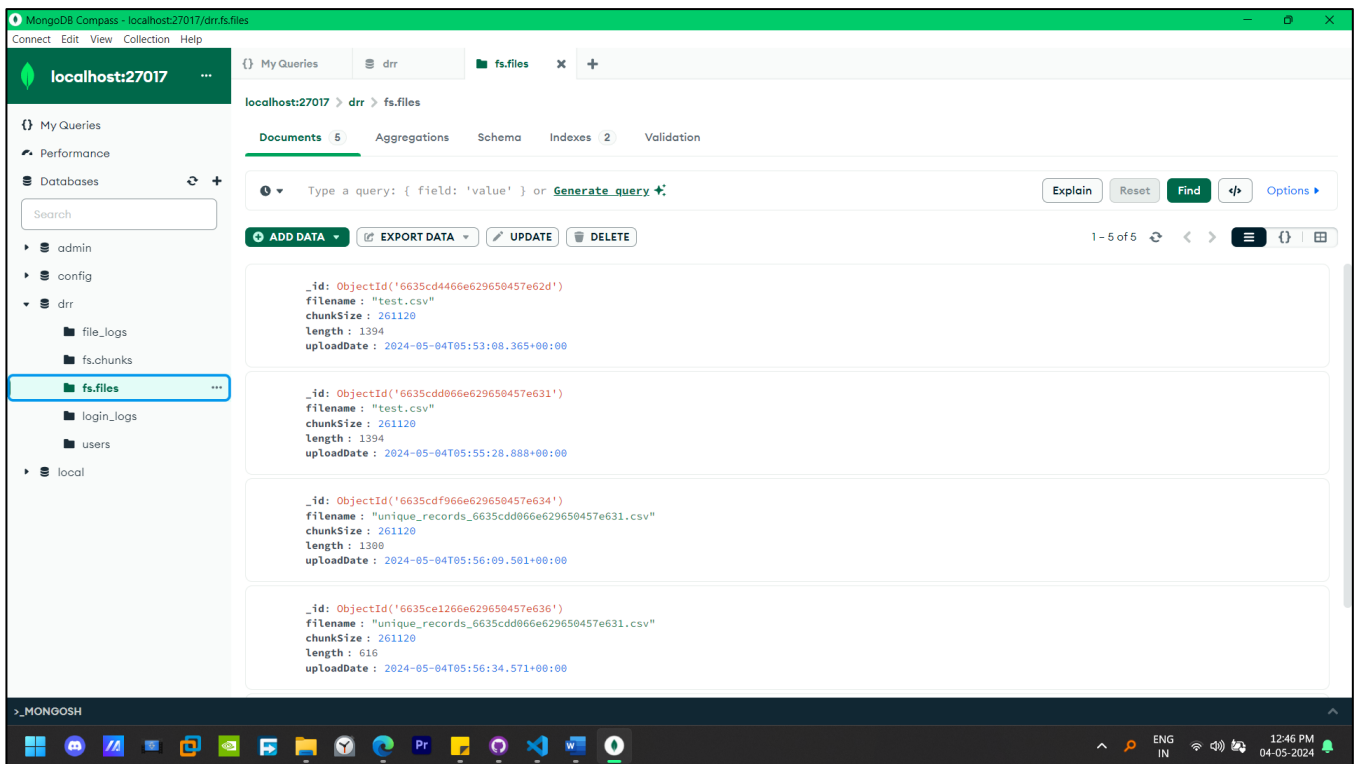
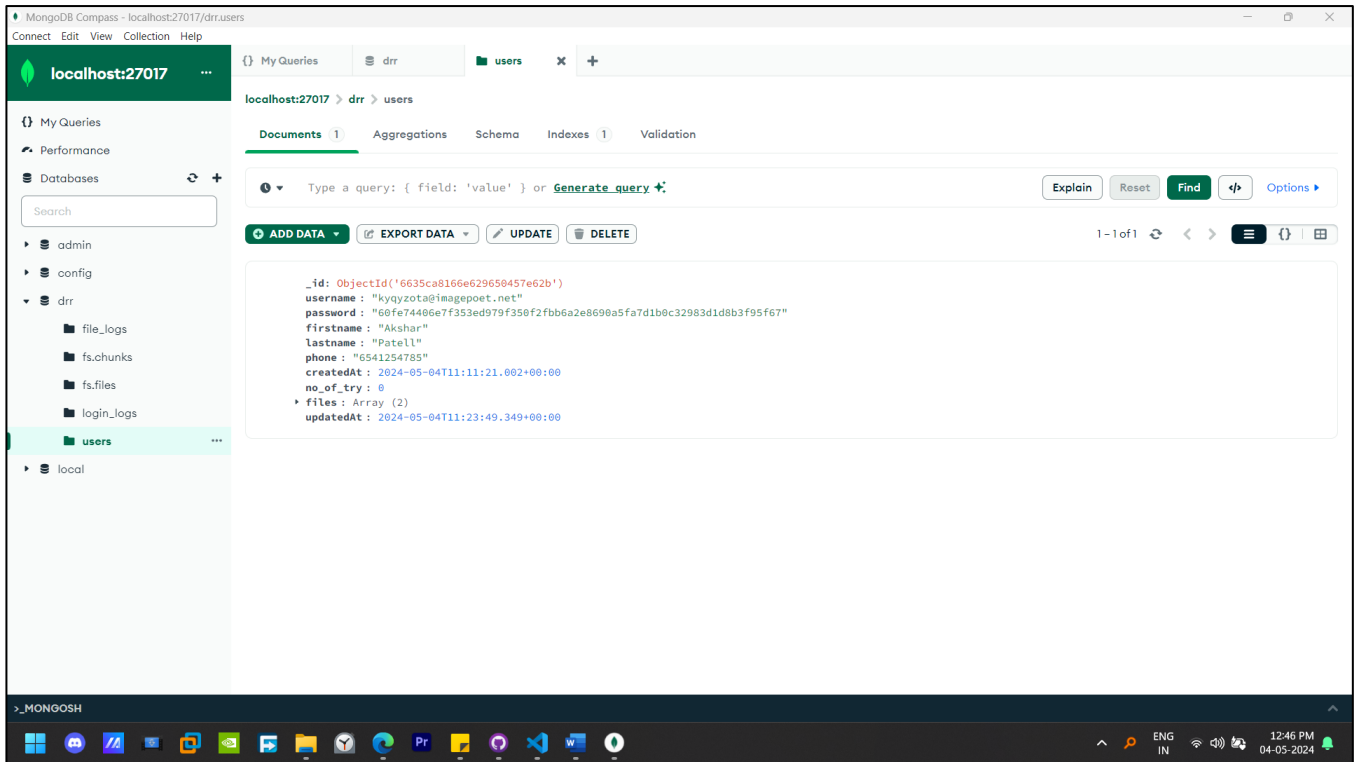
Collection Name	Storage size	Documents	Avg. document size	Indexes	Total index size
file_logs	20.48 kB	2	156.00 B	1	36.86 kB
fs.chunks	20.48 kB	5	1.26 kB	2	73.73 kB
fs.files	20.48 kB	5	117.00 B	2	73.73 kB
login_logs	20.48 kB	1	148.00 B	1	20.48 kB
users	20.48 kB	1	148.00 B	1	20.48 kB

The bottom status bar shows the terminal with the command '>_MONGOSH' and the system clock at 12:45 PM on 04-05-2024.

This screenshot shows the 'login_logs' collection selected within the 'drr' database. The 'Documents' tab is active, displaying a single document:

```
{
  "_id": ObjectId("6635ca9366e629658457e62c"),
  "username": "kyqyzota@imagepoet.net",
  "login_time": "2024-05-04T11:11:39.588+00:00",
  "ip_address": "127.0.0.1",
  "city": "",
  "state": "",
  "status": "success"
}
```

The interface includes a query bar at the top with the placeholder 'Type a query: { field: 'value' } or Generate query'. Below the document, there are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The bottom status bar shows the terminal with the command '>_MONGOSH' and the system clock at 12:45 PM on 04-05-2024.



MongoDB Compass - localhost:27017/drr:fs.files

Connect Edit View Collection Help

localhost:27017

My Queries Performance Databases Search

admin config drr file_logs fs.chunks fs.files login_logs users local

My Queries drr fs.files

Documents 5 Aggregations Schema Indexes 2 Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

1 - 5 of 5

```
{ "_id": ObjectId("6635cd4466e629650457e62d"),
  "filename": "test.csv",
  "chunkSize": 261120,
  "length": 1394,
  "uploadDate": "2024-05-04T05:53:08.365+00:00" }
{ "_id": ObjectId("6635cd0866e629650457e631"),
  "filename": "test.csv",
  "chunkSize": 261120,
  "length": 1394,
  "uploadDate": "2024-05-04T05:55:28.888+00:00" }
{ "_id": ObjectId("6635cdf966e629650457e634"),
  "filename": "unique_records_6635cd0866e629650457e631.csv",
  "chunkSize": 261120,
  "length": 1308,
  "uploadDate": "2024-05-04T05:56:09.501+00:00" }
{ "_id": ObjectId("6635ce1266e629650457e636"),
  "filename": "unique_records_6635cd0866e629650457e631.csv",
  "chunkSize": 261120,
  "length": 616,
  "uploadDate": "2024-05-04T05:56:34.571+00:00" }
```

>_MONGOSH

Windows taskbar: 12:46 PM, 04-05-2024

MongoDB Compass - localhost:27017/drr:file_logs

Connect Edit View Collection Help

localhost:27017

My Queries Performance Databases Search

admin config drr file_logs fs.chunks fs.files login_logs users local

My Queries drr file_logs

Documents 2 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

1 - 2 of 2

```
{ "_id": ObjectId("6635cd4466e629650457e62f"),
  "username": "kyqyzota@imagepoet.net",
  "file_name": "test.csv",
  "uploaded_at": "2024-05-04T11:23:08.818+00:00",
  "ip_address": "127.0.0.1",
  "city": null,
  "status": "success" }
{ "_id": ObjectId("6635cdd166e629650457e633"),
  "username": "kyqyzota@imagepoet.net",
  "file_name": "test.csv",
  "uploaded_at": "2024-05-04T11:25:29.873+00:00",
  "ip_address": "127.0.0.1",
  "city": null,
  "status": "success" }
```

>_MONGOSH

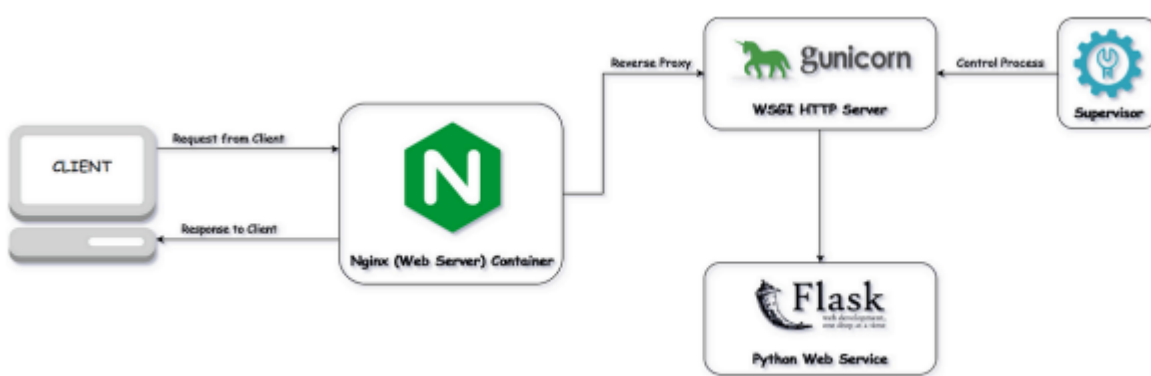
Windows taskbar: 12:46 PM, 04-05-2024

Chapter 7 Deployment on AWS Cloud

7.1 Instance Details

Instance Type:	EC2 t2.micro
Instance Name:	drrs
Instance Platform:	Ubuntu
Platform Details:	Linux/UNIX – SERVER: NGINX

7.2 Load Balancing with Nginx and Gunicorn in a Django Application



In our Django application, we employed a combination of Nginx and Gunicorn for efficient load balancing. This setup optimizes the distribution of incoming web traffic across multiple instances of our Django application, ensuring high availability, improved performance, and fault tolerance.

Nginx:

Nginx acts as a reverse proxy server, handling client requests and distributing them among multiple Gunicorn workers. It efficiently manages incoming connections, optimizes resource utilization, and provides additional functionalities such as caching, SSL termination, and request routing.

By leveraging Nginx's event-driven architecture and asynchronous processing capabilities, we achieved low-latency response times and high throughput, even under heavy loads. Furthermore, Nginx's robust configuration options allowed us to fine-tune our load balancing strategy according to our application's specific requirements.

Gunicorn:

Gunicorn, short for Green Unicorn, serves as the WSGI HTTP server for our Django application. It interfaces between Nginx and the Django web framework, handling incoming requests, processing them, and generating responses. Gunicorn utilizes multiple worker processes to concurrently serve incoming requests, effectively utilizing the available system resources and maximizing performance.

Through Gunicorn's seamless integration with Django and its ability to scale horizontally by adding

more worker processes or deploying multiple instances, we ensured scalability and responsiveness, even during periods of increased traffic or sudden spikes in demand.

Benefits:

- **Scalability:** The combination of Nginx and Gunicorn allowed us to horizontally scale our Django application by adding more server instances or worker processes as needed, accommodating growing user traffic and maintaining consistent performance levels.
- **Fault Tolerance:** By distributing incoming requests across multiple server instances, our load balancing setup enhanced the fault tolerance of our application, mitigating the impact of server failures or downtime and ensuring uninterrupted service availability.
- **Performance Optimization:** Nginx's efficient request handling and Gunicorn's multi-worker architecture optimized the performance of our Django application, delivering fast response times, low latency, and high throughput, even under heavy loads.

CHAPTER 8. PROJECT CONCLUSION

The project revolves around the development of a data redundancy removal system aimed at optimizing non-optimized .csv files. To achieve this, a suite of methods including MDNN removal, Exact match removal, Fuzzy match removal, CWOA Match removal, and Selected column removal are employed. These methods collectively serve to streamline the data by eliminating duplicate entries and enhancing data integrity.

Technologically, Python is utilized for scripting the system's functionalities, while MongoDB serves as the local database management system. Furthermore, AWS is leveraged for deploying the system publicly on the internet, making it accessible to users beyond local environments.

However, several constraints need to be addressed to ensure the system's effectiveness and viability. Performance optimization stands out as a primary concern due to the potentially large datasets the system will encounter. Scalability is also crucial to accommodate increasing loads and data volumes as the project gains traction. Additionally, robust data security measures are imperative to safeguard sensitive information processed by the system, both locally and in the cloud.

Furthermore, attention must be given to designing a user-friendly interface to facilitate seamless file uploading, method selection, and result visualization, enhancing overall user experience. Lastly, prudent cost management strategies must be implemented, especially concerning AWS usage, to optimize expenditure and ensure the project's financial sustainability in the long run. Addressing these constraints effectively is paramount for the successful implementation and widespread adoption of the data redundancy removal system.

CHAPTER 9. REFERENCES

1. <https://bhuvan.nrsc.gov.in/hackathon/iisf2023/topics/Topic12.pdf>
2. [An approach to remove duplication records in healthcare dataset based on Mimic Deep Neural Network (MDNN) and Chaotic Whale Optimization (CWO)] Anto Praveena M.D, and Bharathi B, <https://journals.sagepub.com/doi/pdf/10.1177/1063293X21992014>
3. GRIDFS-FILE Management MongoDB: <https://www.mongodb.com/docs/manual/core/gridfs/>
4. Pandas library of Python: https://pandas.pydata.org/docs/getting_started/index.html
5. DEDUPLICATING DATA AND REMOVING REDUNDANCY IN CLOUD - Arun Singh Kaurav, T.Santhosh Kumar, V.Yadigiri Assistant Professor, Assistant Professor, Assistant Professor Computer Science and Engineering Guru Nanak Institutions Technical Campus, Hyderabad, India
<https://www.ijcrt.org/papers/IJCRT1892578.pdf>
6. <https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-gunicorn-on-ubuntu>, Erin Glass, Jamon Camisso, and Easha Abid