

De-Dupe Engine



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

De-Dupe Engine

Team Members:

Chandru S	19BCE1878
Priyanshu Prasad	19BLC1017
Ujjwal Gupta	19BCE1133

Submitted to: **Dr Sivabalakrishnan M**

Under the course

CSE2004 – Database Management System

B. TECH - Computer Science and Engineering

In

Fall Semester 21-22

Acknowledgment

I would like to express my special thanks to our professor **Dr Sivabalakrishnan M** who gave me the golden opportunity to do this wonderful project on the topic De Dupe Engine which helped me in doing a lot of research and I came to know about so many new things. I am really thankful to sir.

Secondly, I would also like to extend my gratitude to my parents, friends and my team mates who helped me a lot in finalizing this project within the limited time frame and all possible resources.

INDEX

Sr. No.	Content	Page No.
1	Synopsis	4
2	Introduction	5
3	Literature Survey	6
4	Proposed Model	8
5	ER Diagram	9
6	Tech Stack	10
7	Working Modules	12
8	Output Screenshots	15
9	Conclusion	23
10	References	24

Synopsis

No SQL Databases is an emerging technology which is set to replace the older relational databases in this field. It helps us to scale and make robust data management system. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads. Presence of duplicate entries can be an important issue when observed from a storage perspective. Storage requires lot of capital investment. Hence the need for optimization in storage crucial. With the presence of duplicate data, the database gets larger unnecessarily. This poses hindrance for any kind of searching / sorting algorithm as their efficiency is inversely proportional to the size of the data.

Above all, the project revolves around the prevention of duplication in No SQL Databases like MongoDB. We have implemented a Web application towards solving this problem. It connects with the database at the backend and analyses the duplicates for single user entry and batch mode over a large database.

Introduction

No SQL Databases is an emerging technology and is an upcoming technology in every other field in the industry for building scalable and robust systems. For normal Relational database, each data can be assigned unique ID's like the primary key and foreign keys to check for duplicate data. No SQL databases work in a different way and need some duplication prevention system before entering data into the system in bulk or even for single user system.

Presence of duplicate entries can be an important issue when observed from a storage perspective. Storage requires lot of capital investment. Hence the need for optimization in storage crucial. With the presence of duplicate data, the database gets larger unnecessarily. This poses hindrance for any kind of searching / sorting algorithm as their efficiency is inversely proportional to the size of the data.

For many organisations this problem arises from, their varied source of acquisition of data which causes accumulation of the exact same files. For others, it's a collection of files that aren't completely identical yet contain pieces of the same data. Normal Conventional String matching algorithms like Rabin Karp or KMP on Boolean Logic. When used, the most it can say is, whether two strings are exactly matching or not. It returns either true or false.

Literature Survey

We went through the following papers to learn more about the duplication in No SQL Databases and the methods that can be deployed to solve the problem and made the decisions accordingly.

- *Backialakshmi N and Manikandan M, "Data de duplication using NOSQL Databases in Cloud," 2015 International Conference on Soft-Computing and Networks Security (ICSNS), 2015, pp. 1-4, doi: 10.1109/ICSNS.2015.7292436.*— This paper gives an overview about how to maximally reduce the amount of duplicates in one type of NoSQL DBs, namely the key-value store, to maximally increase the process performance such that the backup window is marginally affected, and to design with horizontal scaling in mind such that it would run on a Cloud Platform competitively.
- *Gu, Yunhua & Wang, Xing & Shen, Shu & Ji, Sai & Wang, Jin. (2015). Analysis of data replication mechanism in NoSQL database MongoDB. 66-67. 10.1109/ICCE-TW.2015.7217033.* – This conference paper analyses data replication mechanism in NoSQL database
- *A. Burdakov, U. Grigorev, A. Ploutenko and E. Ttsviashchenko, "Estimation Models for NoSQL Database Consistency Characteristics," 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), 2016, pp. 35-42, doi: 10.1109/PDP.2016.23.*— This paper is about solving NoSQL database replication problems. It analyzes the influence of the N, W, R replication parameters on the consistency characteristics of database record replicas.

- Y. Gu, X. Wang, S. Shen, S. Ji and J. Wang, "Analysis of data replication mechanism in NoSQL database MongoDB," *2015 IEEE International Conference on Consumer Electronics - Taiwan, 2015*, pp. 66-67, doi: 10.1109/ICCE-TW.2015.7217033.– It focusses on the important concept of NoSQL database MongoDB includes Master/Slave structure and Replica Set. Write operation implement on Master, Slaves will send the synchronize data command asynchronously to Master to update its data. Read operation just implement on Master to provide the strong consistency, while read operation implement on Slave to provide the eventual consistency. Replica set is a group of servers which run Mongod and store the copy of the same data with automatic failover and automatic recovery of member nodes.

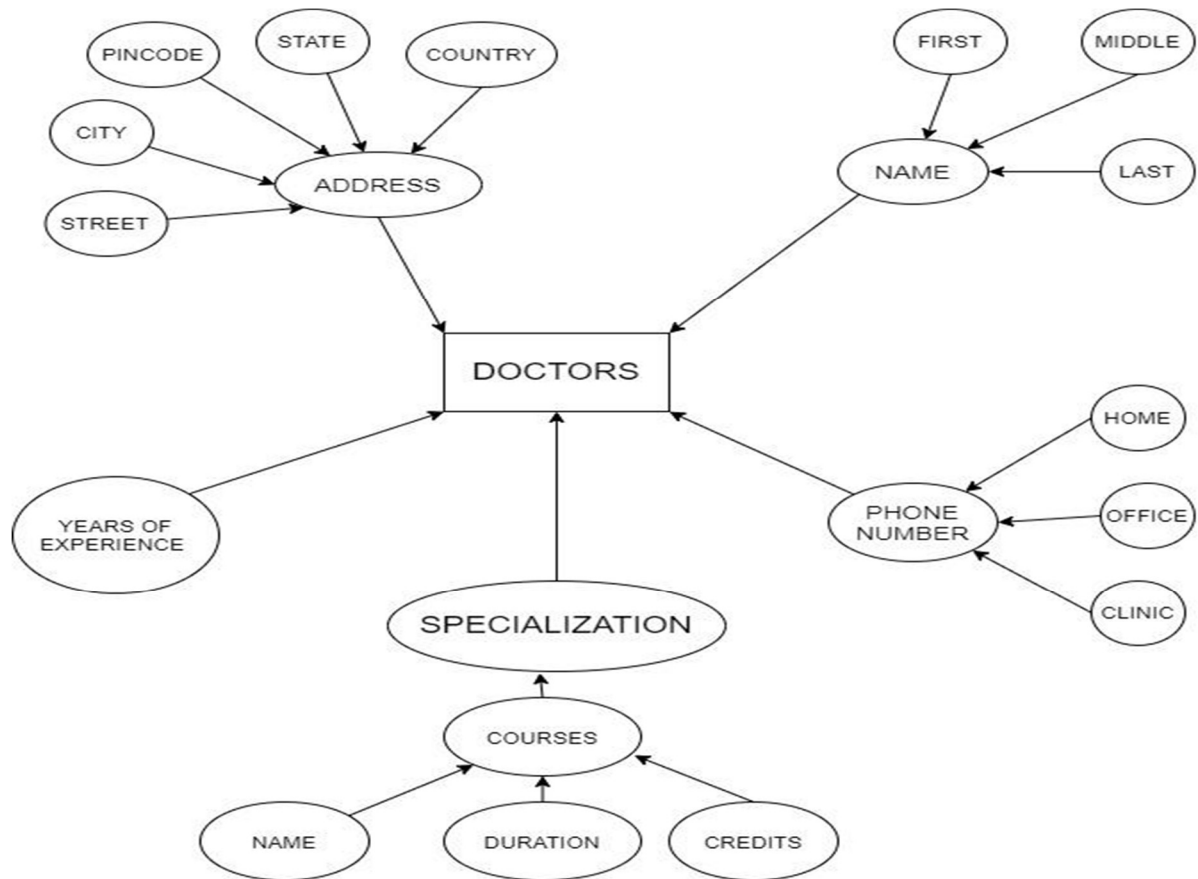
Proposed Model

Our proposed model aims to implement a complete standalone de duplication system where there will be two modes of operation for the service. One will be a single mode and the other will be a batch mode. The single mode is useful when a service needs to signup a user for their database and give access to features. They will be able to check if the user is the same person or a different individual based on the data entered. This ensures the trust of the service and prevent unnecessary duplicates from getting entered into the database.

The other mode of operation is the Batch Mode which focusses on the bulk upload from a user in the form of a excel/CSV sheet. This means that the data needs to be processed with respect to an already existing huge database. On successful comparison a report is generated for the admin to look into and remove verify the details of the person for getting added to the database.

The special features of our application is that the user/admin can tune the weights and set a bias according to the knowledge about the significance of the column attributes. The uploaded data undergoes thorough checking for any duplicates and the rows not fulfilling the threshold criteria for uniqueness are returned as error. The data entered has any duplicates the corresponding rows data is displayed with a message along with the similarity score else an affirmation message is displayed.

ER Diagram



Tech Stack

- Mongo DB



MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server-Side Public License

- Flask Server



Flask

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. It's has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features. It has many features like URL routing, template engine.

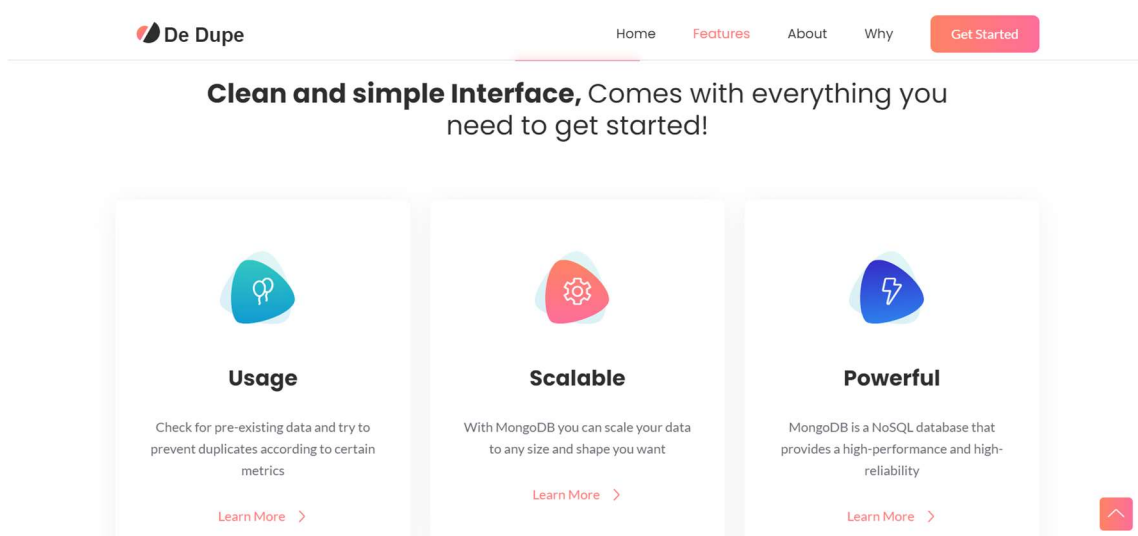
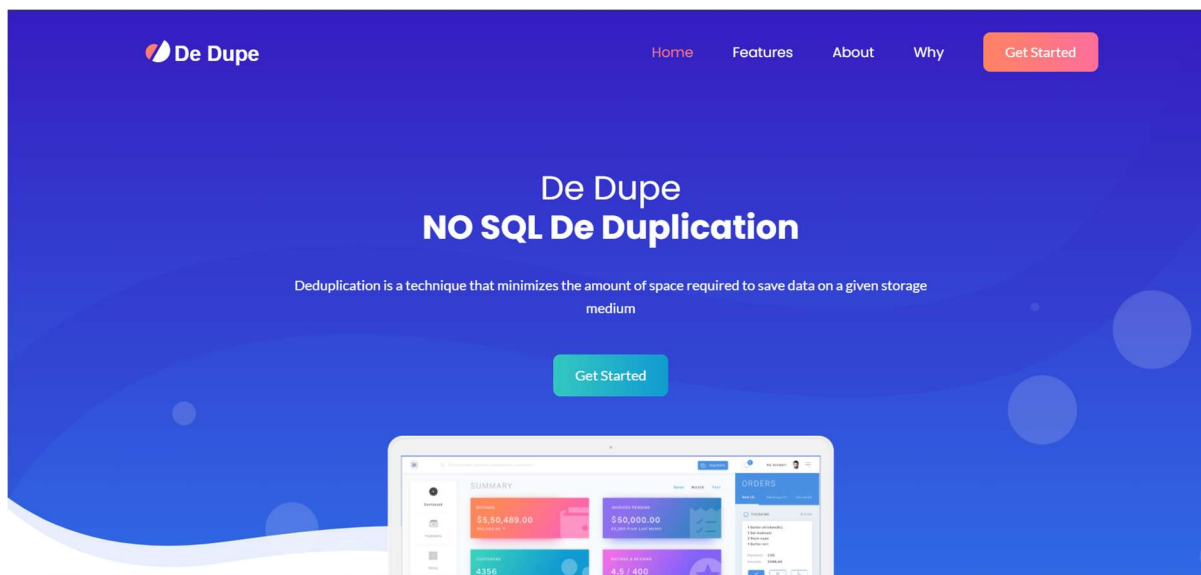
- Bootstrap



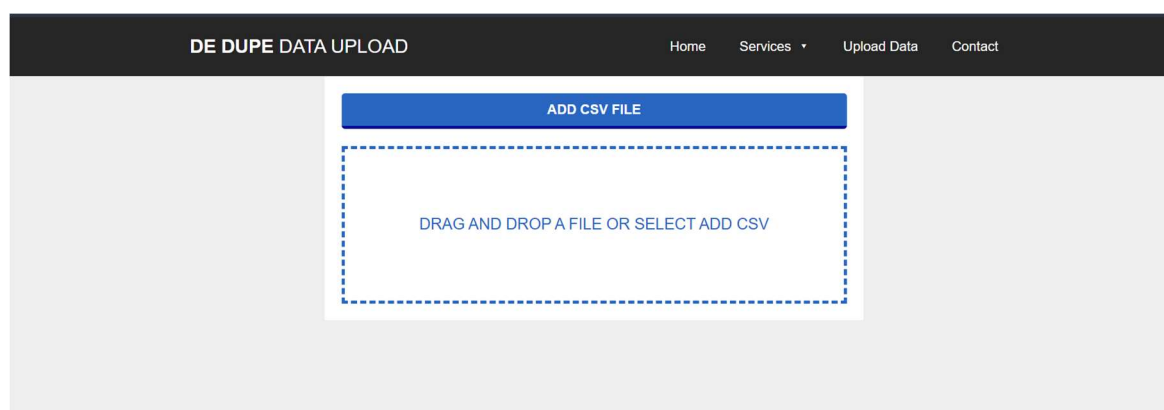
Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Working Modules

Module 1: Landing Page



Module 2: Data Upload Populating Database



Module 3: Single Mode

DE DUPE SINGLE USER SYSTEM

[Home](#)[Services](#)[Upload Data](#)[Contact](#)

MRN Number	<input type="text"/>	<div><div></div><div>0</div></div>
First Name	<input type="text"/>	<div><div></div><div>0</div></div>
Last Name	<input type="text"/>	<div><div></div><div>0</div></div>
DOB	<input type="text" value="dd-mm-yyyy"/>	<div><div></div><div>0</div></div>
State	<input type="text"/>	<div><div></div><div>0</div></div>
Pincode	<input type="text"/>	<div><div></div><div>0</div></div>
Pincode	<input type="text"/>	<div><div></div><div>0</div></div>
Phone	<input type="text"/>	<div><div></div><div>0</div></div>
Years of Exp	<input type="text"/>	<div><div></div><div>0</div></div>
Specialization	<input type="text"/>	<div><div></div><div>0</div></div>
Education	<input type="text"/>	<div><div></div><div>0</div></div>

Check Duplication

Module 4: Batch Mode

DE DUPE BATCH USER SYSTEM

HomeServicesUpload DataContact

ADD CSV FILE

DRAG AND DROP A FILE OR SELECT ADD CSV

Upload CSV

MRN Number

0.01

MRN Number

0.01

First Name

0.01

Last Name

0.01

DOB

0.01

State

0.01

Pincode

0.01

Phone

0.01

Years of Exp

0.01

Specialization

0.01

Phone

0.01

Years of Exp

0.01

Specialization

0.01

Education

0.01

Check for Bulk Duplication and Generate Report

Implementation

Setup of the Modules

- **Module 1: Single User Data Upload:**

This module is for the single user data checking. The user can check for duplication against the already existing database. Each attribute can be assigned weights for checking the similarity score. This helps in identifying the unique data and can be inserted into the database if it does not exist.

DE DUPE SINGLE USER SYSTEM

[Home](#) [Services](#) [Upload Data](#) [Contact](#)

MRN Number

w7fWWD

0.7

First Name

Michael

0.61

Last Name

Baeza

0.51

DOB

24-12-1972

0.52

State

Bihar

0.54

Pincode

758346

0.41

State

Bihar

0.54

Pincode

758346

0.41

Phone

2423271092

0.57

Years of Exp

14

0.58

Specialization

General Physician

0.45

Education

MBBS, MS Pediatric surgeon

0.53

Check Duplication

Figure 1: Input of Details

De-Dupe Engine

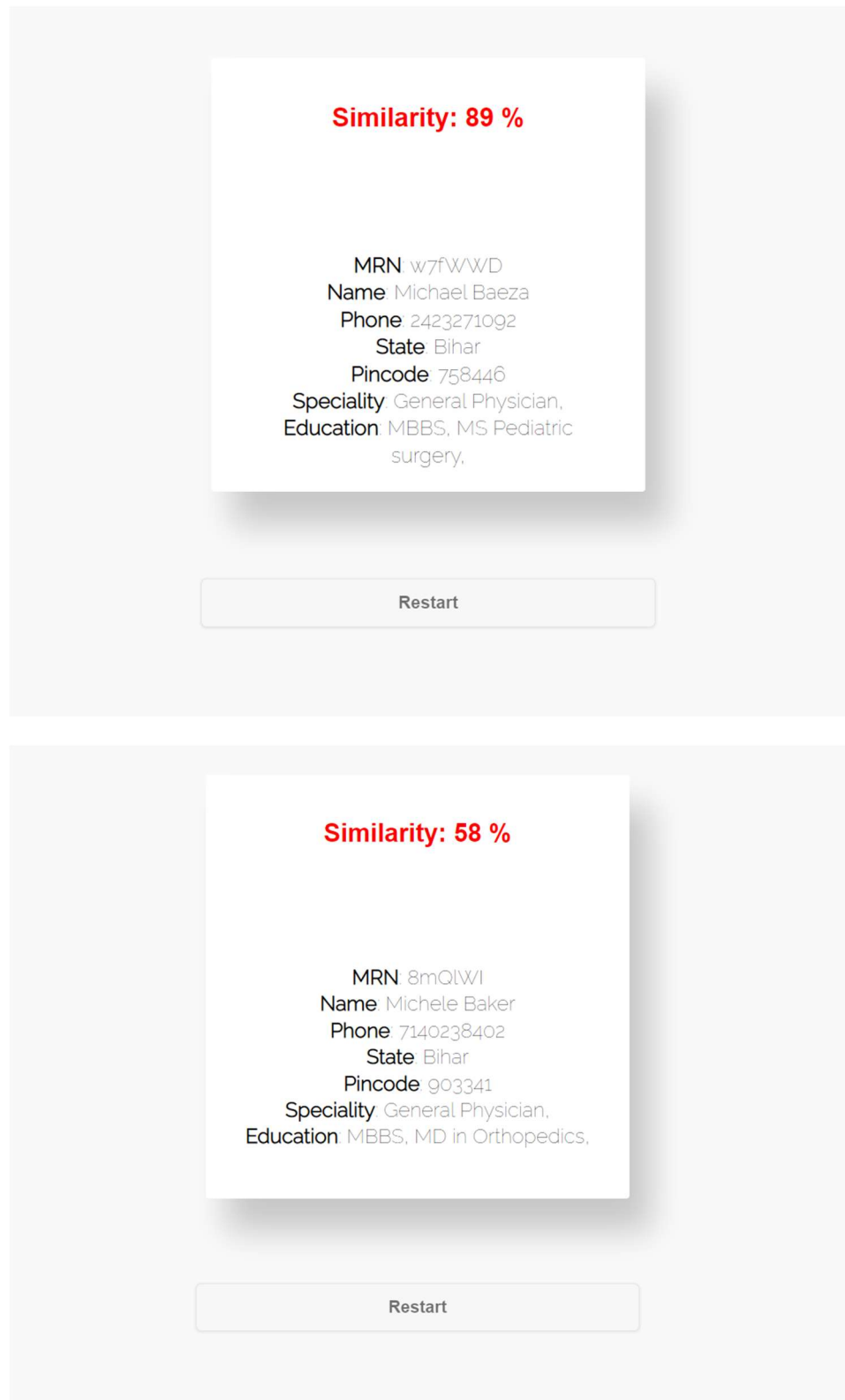


Figure 2: Getting the similar data entries

De-Dupe Engine

- Similarity: 58 % MRN: 8mQIWName: Michele BakerPhone: 7140239402State: BiharPincode: 903344Speciality: General Physician, Education: MBBS, MD in Orthopedics;
- Similarity: 56 % MRN: m8eYWCName: Michael GrantPhone: 7367497730State: BiharPincode: 288246Speciality: General Physician, Education: MBBS, MS-Cosmetic surgery;
- Similarity: 55 % MRN: wllVAHName: Michael HydePhone: 9616181460State: BiharPincode: 626067Speciality: General Physician, Education: MBBS, MD in Dermatology;
- Similarity: 54 % MRN: hA7zweName: Michael DavenportPhone: 6687080004State: BiharPincode: 709066Speciality: Orthopedic, Education: MBBS, MS-Cardiac surgery;

☒ Show all

Restart

Save as *.pdf

Print

Figure 3: Printing the report

DE DUPE SINGLE USER SYSTEM Home Services ▾ Upload Data Contact

MRN Number	<input type="text" value="w7fWWD"/>	<input type="range" value="0.57"/>	0.57
First Name	<input type="text" value="Ayan"/>	<input type="range" value="0.55"/>	0.55
Last Name	<input type="text" value="Sadhukhan"/>	<input type="range" value="0.52"/>	0.52
DOB	<input type="text" value="23-12-1972"/>	<input type="range" value="0.63"/>	0.63
State	<input type="text" value="West Bengal"/>	<input type="range" value="0.45"/>	0.45
Pincode	<input type="text" value="700123"/>	<input type="range" value="0.53"/>	0.53
Phone	<input type="text" value="+917890032256"/>	<input type="range" value="0.65"/>	0.65
Years of Exp	<input type="text" value="14"/>	<input type="range" value="0.65"/>	0.65
Specialization	<input type="text" value="Computer"/>	<input type="range" value="0.62"/>	0.62
Education	<input type="text" value="B.Tech"/>	<input type="range" value="0.66"/>	0.66

Figure 4: On Inputting unique Data

De-Dupe Engine

State: West Bengal, 0.45

Pincode: 700123, 0.53

Phone: +917890032256, 0.65

Years of Exp: **Data Unique!** (Data is unique and has been added to the database.), 0.65

Specialization: , 0.62

Education: B.Tech, 0.66

[Check Duplication](#)

Figure 4: Modal showing unique data and entering it to the database

- Module 2: Batch Mode Data Upload:**

This module is for the multi user data checking. The admin of the system can upload the data in bulk and generate report based on the already existing database.

DE DUPE SINGLE USER SYSTEM

Home Services Upload Data Contact

Single Mode

Batch Mode 57

MRN Number: w7fWWD

First Name: Ayan, 0.55

Figure 5: Batch Mode Service

De-Dupe Engine

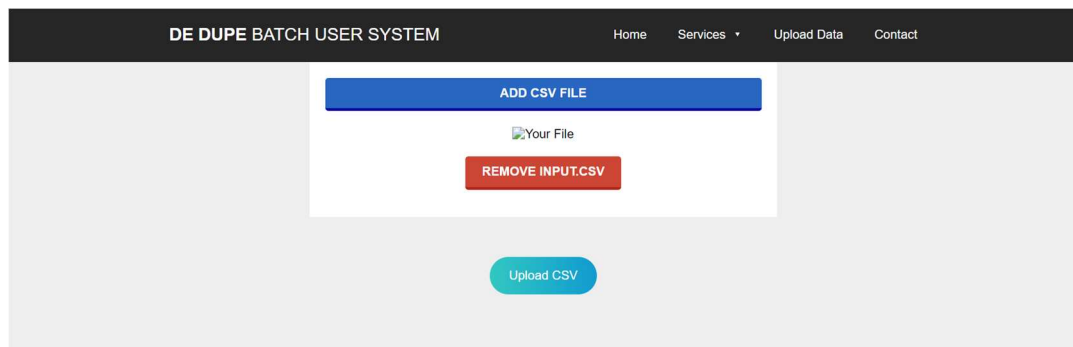


Figure 6: Uploading a CSV file for the record

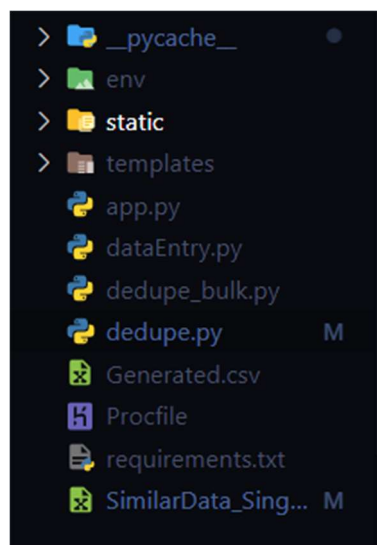
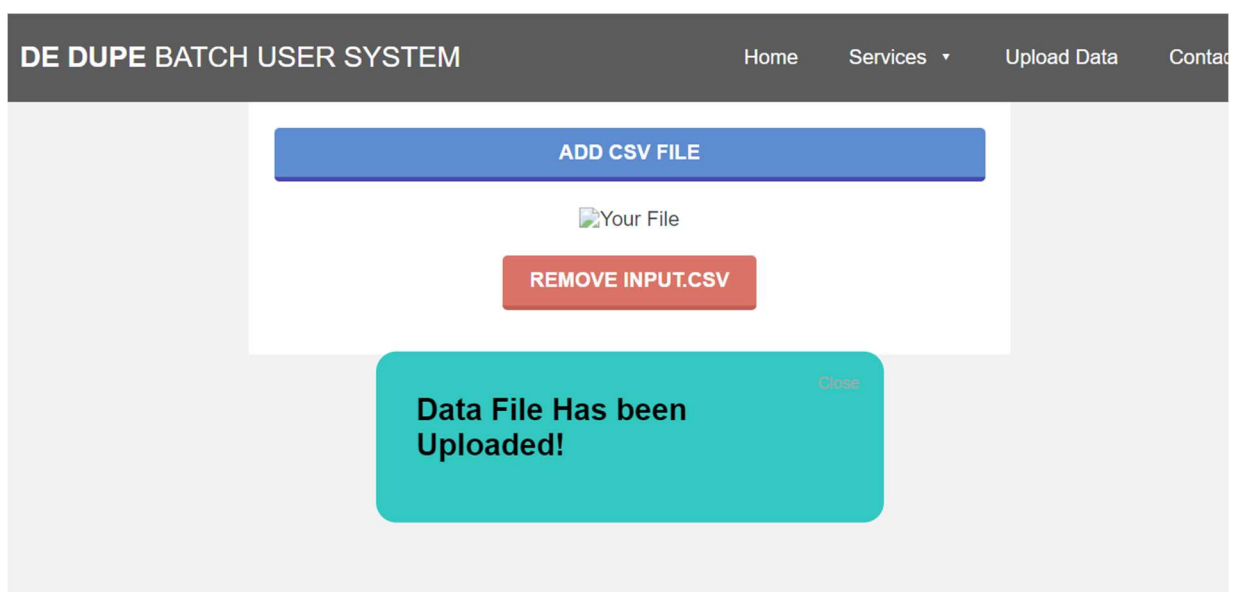


Figure 7: Initial Root Directory



De-Dupe Engine

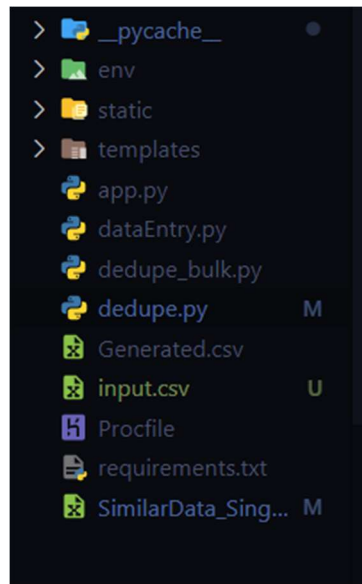


Figure 8: CSV gets stored in the Root Directory



Figure 9: Adding weight to each attribute

De-Dupe Engine

```

def checkDuplicates(MRN,firstName,lastName,DOB,State,Pincode,
                    Phone,YOE,Specialization,Education,MRNScale,fNamescale,
                    lNamescale,DOBscale,Statescale,PincodeScale,Phonescale,
                    YOEScale,Specializationscale,Educationscale):
    # getting the details etc
    mrnWeight = float(MRNScale)
    fNameWeight = float(fNamescale)
    lNameWeight = float(lNamescale)
    dobWeight = float(DOBscale)
    phoneWeight = float(Phonescale)

    [20 rows x 10 columns]>
    Output file generated. Check directory.
    Report Generated! Check file in directory
    127.0.0.1 - - [04/Dec/2021 15:10:43] "POST /check_bulk_duplication HTTP/1.1" 200 -
    127.0.0.1 - - [04/Dec/2021 15:10:43] "GET /static/images/favicon.png HTTP/1.1" 304 -
  
```

Figure 10: Report is generated

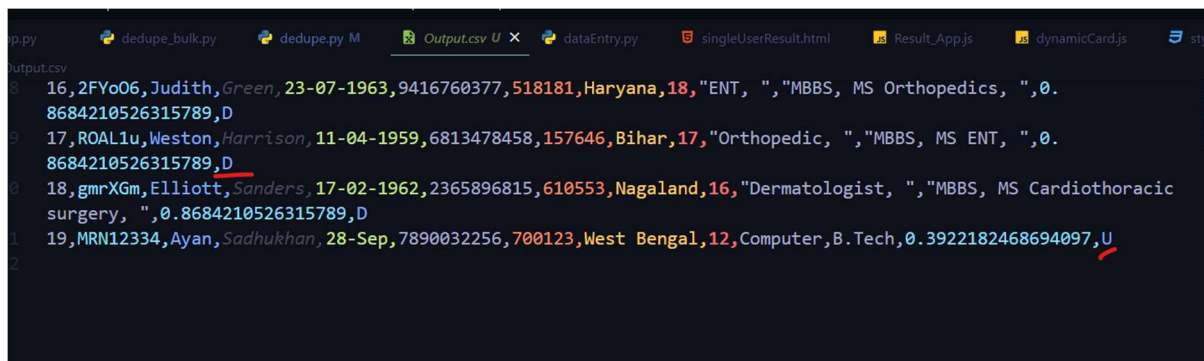
```

1 -----Summary of Input Data-----
2
3 Total numbers of rows in the input data set: 20.
4 Unique Entry Count: 1 (5.0 %)
5 Duplicate Entry Count: 19 (95.0 %)
6 Partial Similarity Entry Count: 0 (0.0 %)
7
  
```

```

1 ,MRN,First Name,Last Name,DOB,Phone Number,Pincode,State,Years of Exp.,Specialization,Education,
  SimilarityScore,DUP
2 0,H99czl,Adele,Haight,20-06-1961,3526188712,943298,Orissa,24,"Paediatrician, ", "MBBS, MS Gynecology, ",0.
  8684210526315789,D
3 1,mDADfi,Debra,Pontillo,10-06-1966,1765564229,491414,Maharashtra,10,"Psychiatrist, Urologist, ", "MBBS, MD in
  General Surgery, ",0.8684210526315789,D
4 2,QX2lup,Leon,Bulson,07-04-1986,1467540836,999467,Kerala,21,"Gynaecologist, ENT, ", "MBBS, MS Cosmetic surgery,
  ",0.8684210526315789,D
5 3,QoHFPA,Mary,Mccormick,21-08-1960,7881064414,125294,Uttar Pradesh,21,"General Physician, Orthopedic, ", "MBBS,
  MD in Aerospace Medicine, ",0.8684210526315789,D
6 4,zsAkyX,Wilbur,Mendoza,13-12-1959,7883177344,359318,Himachal Pradesh,11,"Sexologist, Dermatologist, ", "MBBS,
  MD in Aerospace Medicine, ",0.8684210526315789,D
7 5,71IMkj,Edward,Craig,12-02-1958,5413985181,4231,Assam,16,"Orthopedic, ", "MBBS, MD in Geriatrics, ",0.
  8684210526315789,D
8 6,w7fWMD,Michael,Baeza,28-10-1972,2423271092,758446,Bihar,14,"General Physician, ", "MBBS, MS Pediatric
  surgery, ",0.8684210526315789,D
9 7,DX8FDa,Gladys,Lewis,19-12-1990,9018297804,520330,Madhya Pradesh,16,"General Physician, ", "MBBS, MS Plastic
  surgery, ",0.8684210526315789,D
10 8,up6NvV,Santiago,Roberts,19-09-1961,7240134782,463570,Sikkim,7,"Urologist, Paediatrician, ", "MBBS, MS
  
```

De-Dupe Engine



```
16,2FY006,Judith,Green,23-07-1963,9416760377,518181,Haryana,18,"ENT, ", "MBBS, MS Orthopedics, ",0.
8684210526315789,D
17,ROAL1u,Weston,Harrison,11-04-1959,6813478458,157646,Bihar,17,"Orthopedic, ", "MBBS, MS ENT, ",0.
8684210526315789,D
18,gmrXGm,Elliott,Sanders,17-02-1962,2365896815,610553,Nagaland,16,"Dermatologist, ", "MBBS, MS Cardiothoracic
surgery, ",0.8684210526315789,D
19,MRN12334,Ayan,Sadhukhan,28-Sep,7890032256,700123,West Bengal,12,Computer,B.Tech,0.3922182468694097,U
```

Figure 11: Report Details

Conclusion

We have successfully implemented de duplication engine with report generation for both single user and multi user system.

The UI for the different modes can be improved for a better user experience with downloadable reports. The bulk processing speed can be improved using multithreading and a faster response time. Since we used this algorithm, we are trying to use more better and faster techniques to speed up our process. As said the project can be configured with the cloud DB's like MongoDB Atlas and be made easily accessible for any user to use our web application.

References

- [1] <https://docs.mongodb.com/manual/core/bulk-write-operations/>
- [2] <https://docs.mongodb.com/manual/replication/>
- [3] <https://github.com/bgracz/TinderCards>
- [4] <https://getbootstrap.com/docs/4.1/layout/overview/>
- [5] https://codepen.io/ig_design/pen/VwedgWj
- [6] <https://uideck.com/templates/basic/>