

PROJECT REPORT

CS6308 - JAVA PROGRAMMING

MiniDBMS – YaS DBMS

Team Members,

Sakthi Dasan B A (2019503555)

Sham Ganesh M(2019503559)

Yogeeswar S (2019503573)

Acknowledgement:

With the given opportunity we would like to thank Dr.Jayachitra VP, Assistant professor ,Department of computer Technology, Chennai for their proper guidance and constant assistance to complete this project.

Abstract:-

A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information relevant to an enterprise. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both *convenient* and *efficient*.

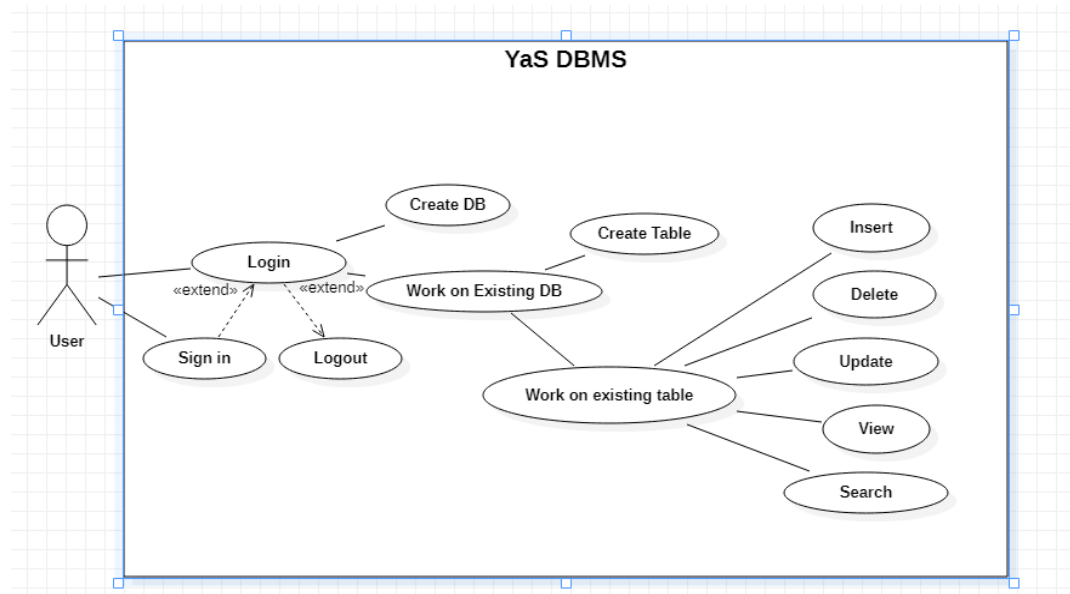
Database systems are designed to manage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results.

Because information is so important in most organizations, computer scientists have developed a large body of concepts and techniques for managing data. These concepts and techniques form the focus of this book. This chapter briefly introduces the principles of database systems.

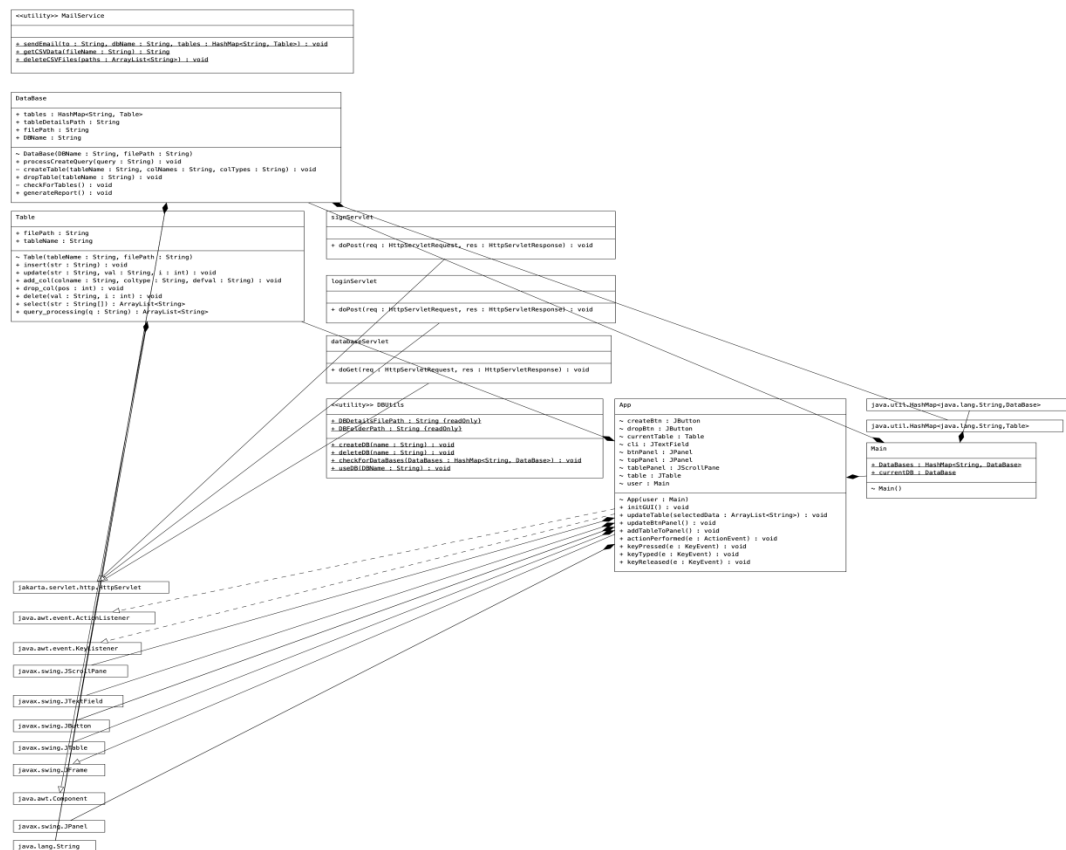
In YaS DBMS, users will be able to create an account by signing up and can login. After logging in a user can either create a new database or can use work on existing database, after selecting the database user can create a new table or work on existing table. With the selected table, users can insert, delete, update, view, search, data.

All the above mentioned functionality can be done with the help of GUI or command line interface, when using CLI user has to write a query similar to but less complex than SQL.

UseCase Diagram:



Class Diagram:

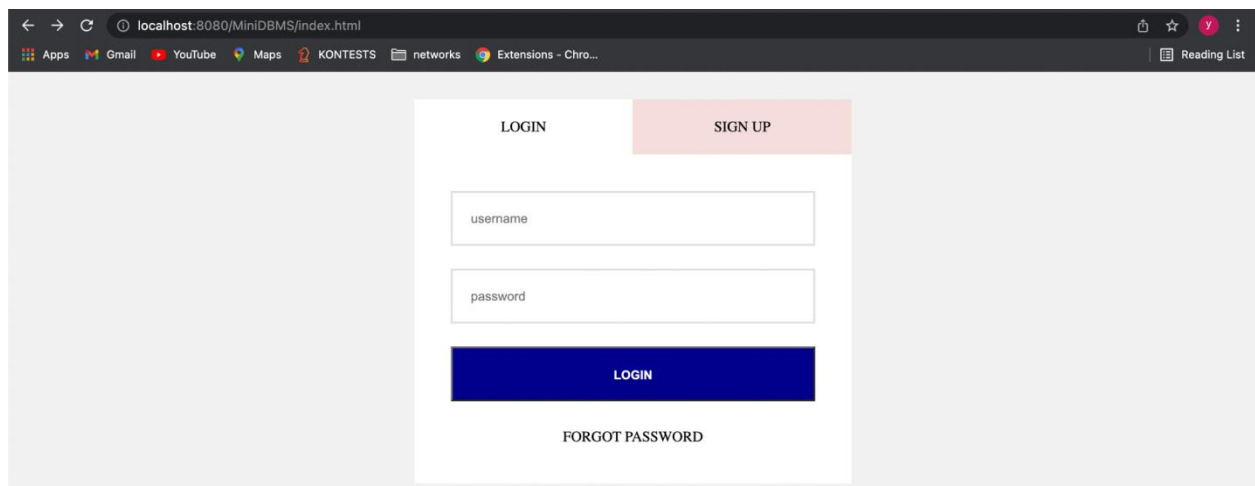


Features of our project:-

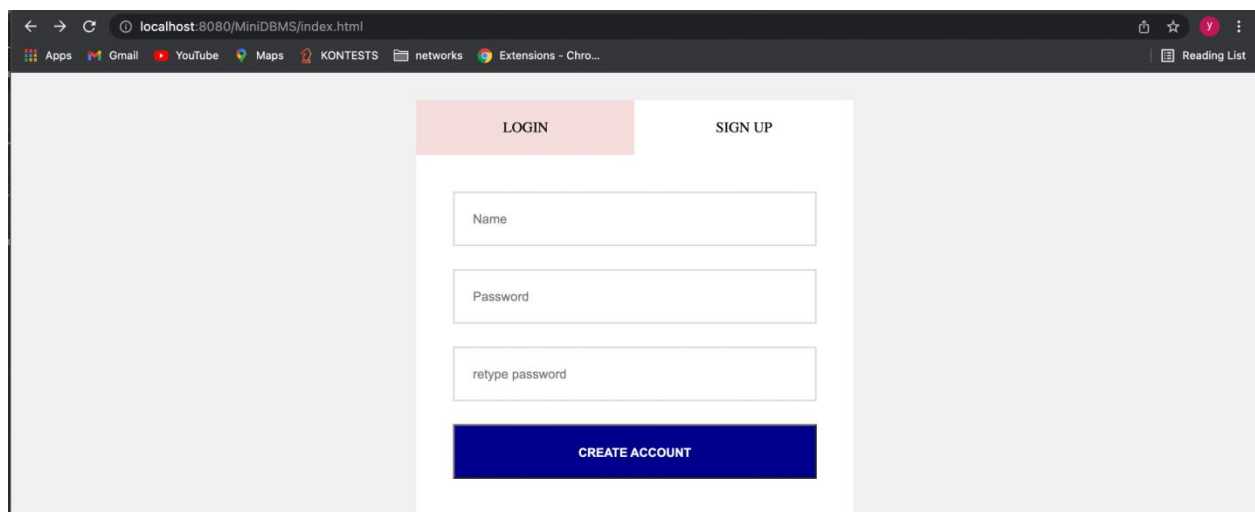
Our project is an improved version of the current existing DBMS with a clean and simple user interface which the user can use with effectiveness. Our DBMS has the functionalities such as creating databases, tables and can insert, update and delete values and can also modify the structure of the table. The features of the DBMS are listed below.

Database:

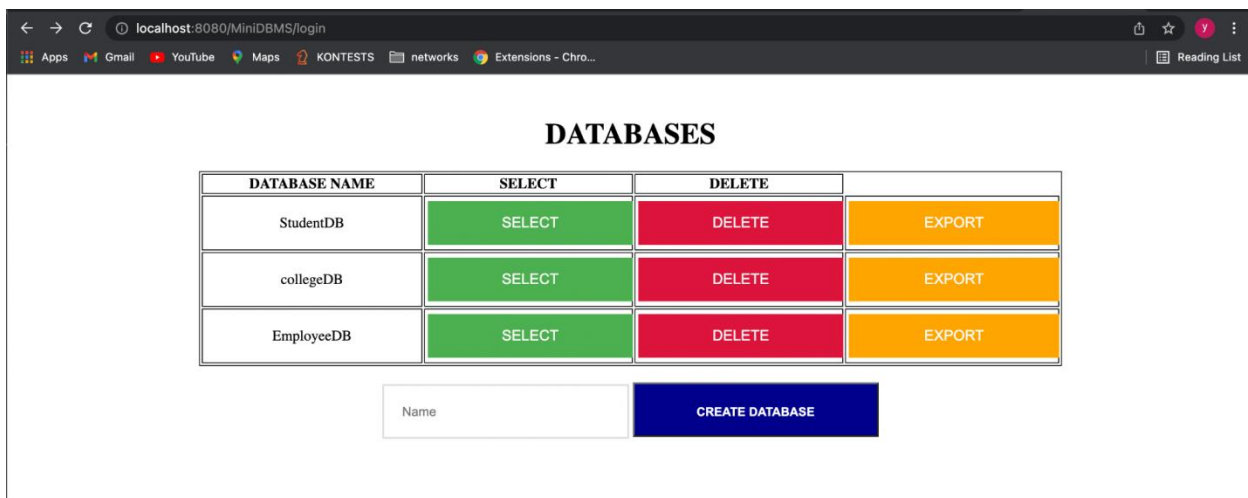
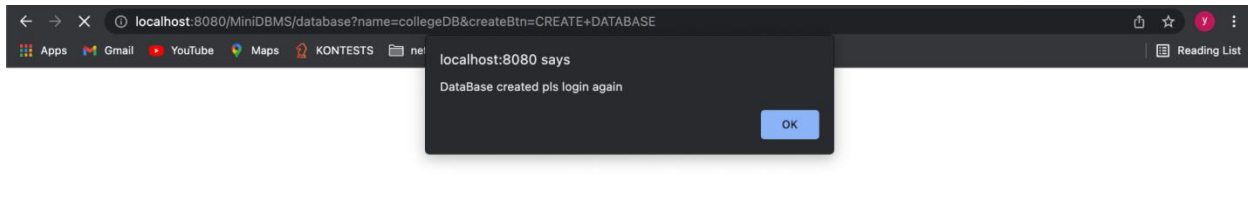
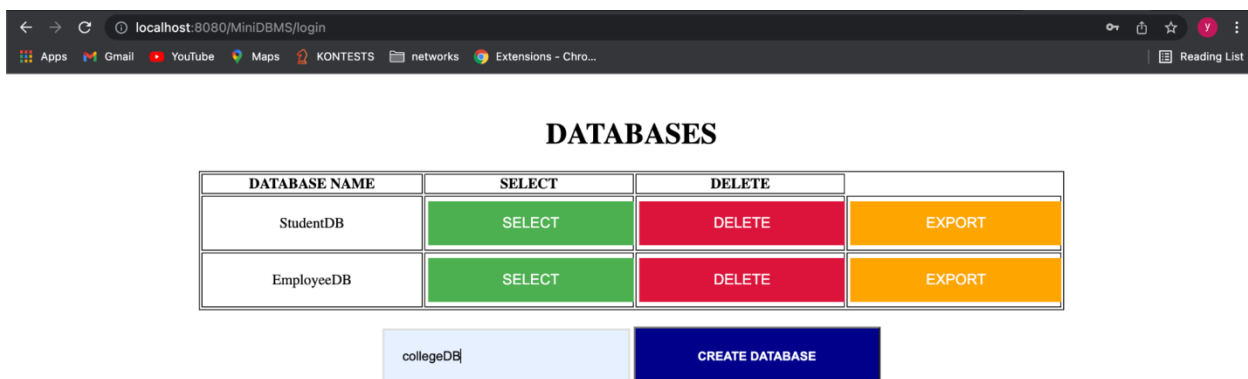
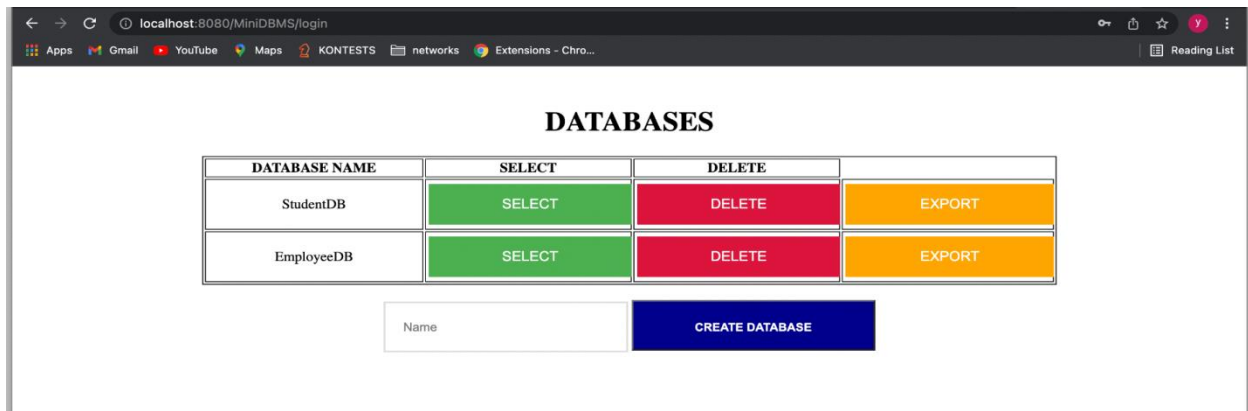
Creating a database is the first step on using a DBMS. In order to create a database user should first login if he/she has already created an account or user has to sign up. Creating a database creates a folder in the local system and after creating the database a user will be able to create tables. The users will be able to drop the created database or create another database with the current user name. User will also be able to export tables in a particular database as csv files to his email.



A screenshot of a web browser displaying the login page of a MiniDBMS application. The browser's address bar shows 'localhost:8080/MiniDBMS/index.html'. The page has a light gray background. In the center, there is a white rectangular box containing the login form. At the top of this box, there are two tabs: 'LOGIN' (active, white background) and 'SIGN UP' (inactive, light red background). Below the tabs, there are two input fields: 'username' and 'password'. Below these fields is a blue button with the text 'LOGIN'. At the bottom of the white box, there is a link that says 'FORGOT PASSWORD'.



A screenshot of the same web browser displaying the sign-up page of the MiniDBMS application. The address bar remains the same. The 'SIGN UP' tab is now active, highlighted with a light red background, while the 'LOGIN' tab is inactive. Below the tabs, there are three input fields: 'Name', 'Password', and 'retype password'. Below these fields is a blue button with the text 'CREATE ACCOUNT'.

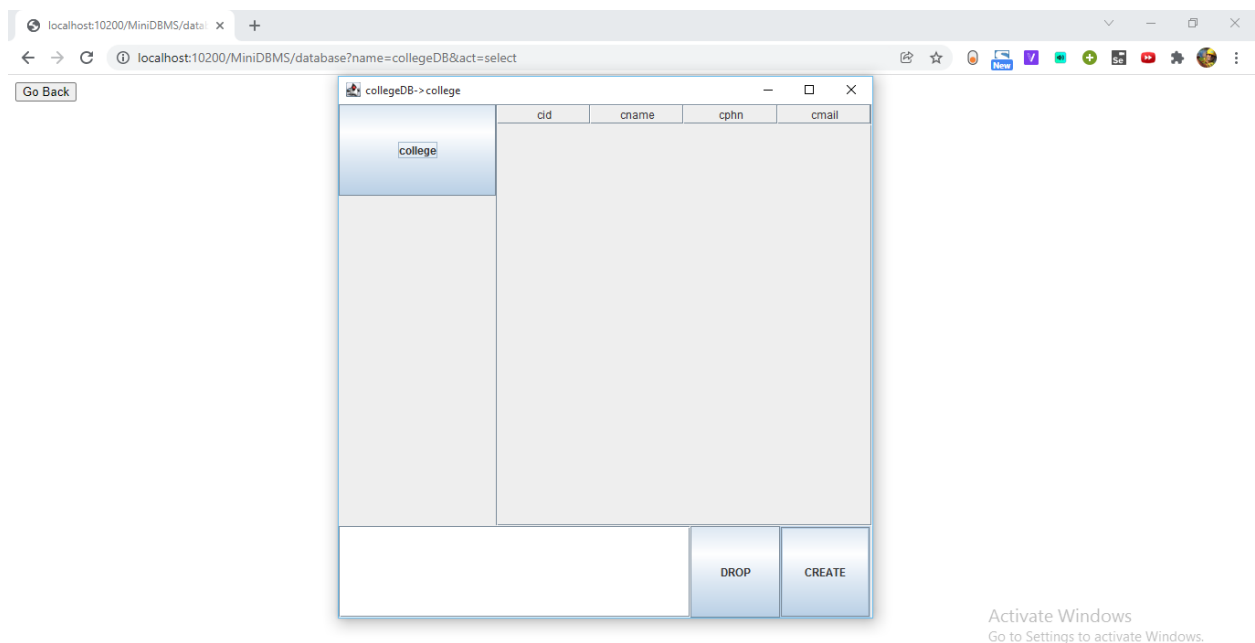
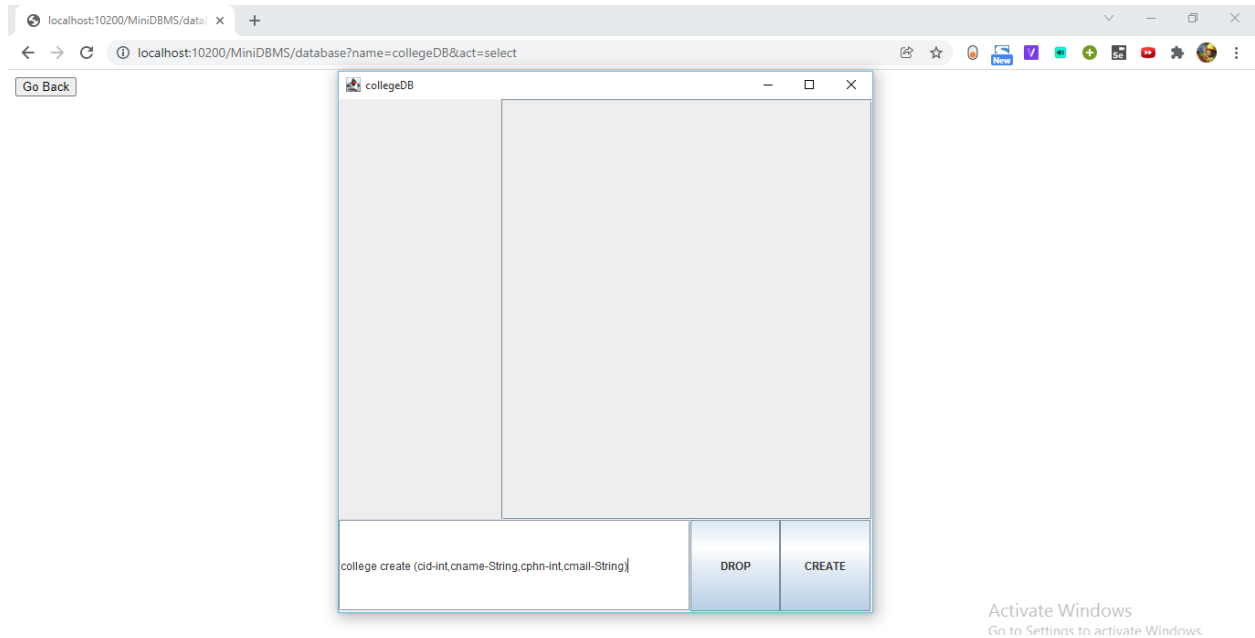


Create table:

Tables are containers which store the data of similar domains. Here while creating a table users have to specify the column names and datatypes of the column. After creating the table, the user can insert the values in the columns.

Syntax: Table-name create (Column names)

Example: college create (cid-int,cname-String,cphn-int,cmail-String)



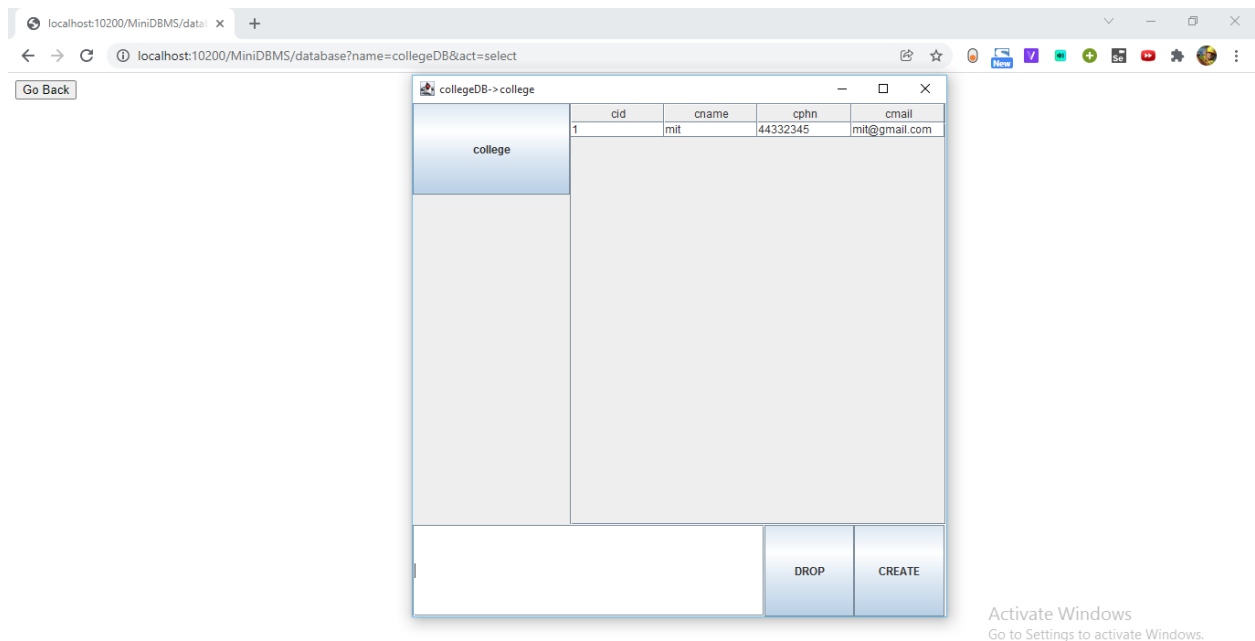
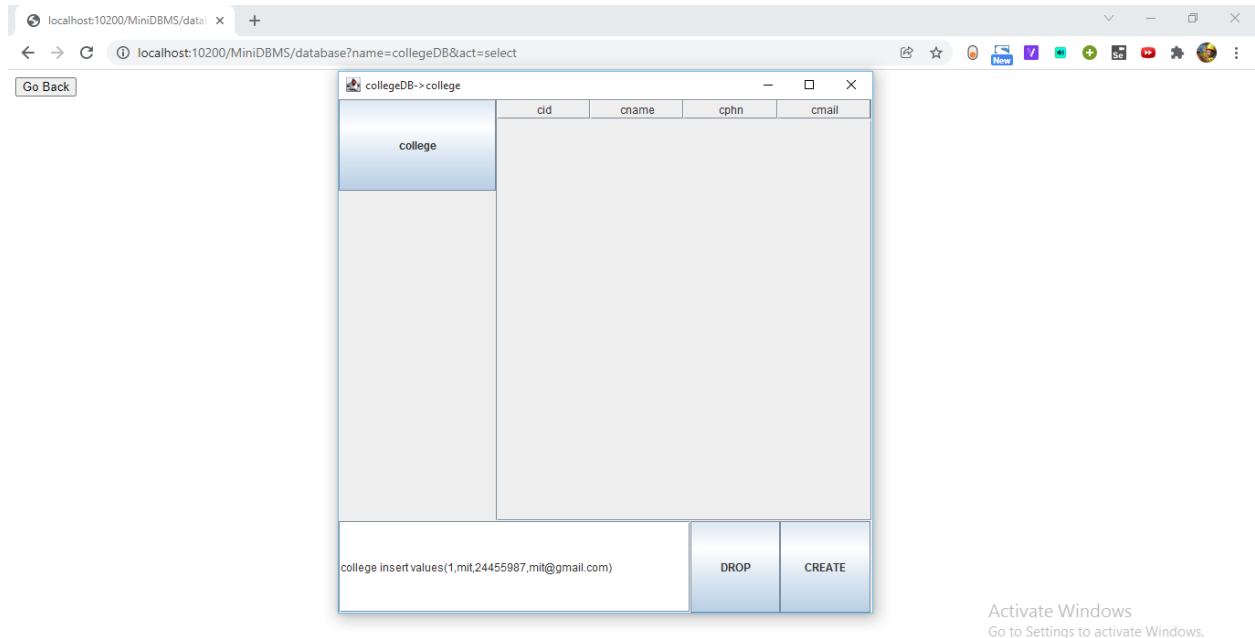
Insert:

Insert is the command used here to insert the values into the table. Insert can also be used to insert values in the specific columns.

(I)

Syntax: Table-name insert values (values to be inserted)

Example: college insert values (1,mit ,24455987,mit@gmail.com)



(II)

Syntax: Table-name insert (Column names) values (values to be inserted)

Example: college insert (cid,cname) values (2,ceg)

localhost:10200/MiniDBMS/data/ x +

← → ↻ localhost:10200/MiniDBMS/datab collegeDB->college

Go Back

	cid	cname	cphn	cmail
college	1	mit	44332345	mit@gmail.com

college insert (cid,cname) values (2,ceg)

DROP CREATE

Activate Windows

localhost:10200/MiniDBMS/data/ x +

← → ↻ localhost:10200/MiniDBMS/datab collegeDB->college

Go Back

	cid	cname	cphn	cmail
college	1	mit	44332345	mit@gmail.com
	2	ceg		

DROP CREATE

Activate Windows
Go to Settings to activate Windows.

Add Column:

This allows user to add column after the creation of table.

(I)

Syntax: Table-name add column (column_name,column_type)

Example: college add column (clocation,String)

localhost:10200/MiniDBMS/data/ x +

localhost:10200/MiniDBMS/database?name=collegeDB&act=select

Go Back

collegeDB->college

	cid	cname	cphn	cmal	
1	mit		44332345	mit@gmail.com	
2	ceg				
3	ssn		44982356	ssn@gmail.com	
4	srm		43574357	srm@gmail.com	
5	vit		87348734	vit@gmail.com	

college add column (clocation,String)

DROP CREATE

Activate Windows
Go to Settings to activate Windows.

localhost:10200/MiniDBMS/data/ x +

localhost:10200/MiniDBMS/database?name=collegeDB&act=select

Go Back

collegeDB->college

	cid	cname	cphn	cmal	clocation
1	mit		44332345	mit@gmail.c...	
2	ceg				
3	ssn		44982356	ssn@gmail...	
4	srm		43574357	srm@gmail...	
5	vit		87348734	vit@gmail.co...	

DROP CREATE

Activate Windows
Go to Settings to activate Windows.

(II)

Syntax: Table-name add column (**column_name,column_type**) val=value

Example: college add column (fees,int) val=20000

localhost:10200/MiniDBMS/data/ x +

localhost:10200/MiniDBMS/database?name=collegeDB&act=select

Go Back

collegeDB->college

	cid	cname	cphn	cmail
1	mit		44332345	mit@gmail.com
2	ceg			
3	ssn		44982356	ssn@gmail.com
4	srm		43574357	srm@gmail.com
5	vlt		87348734	vlt@gmail.com

college add column (fees,int) val=20000

DROP CREATE

Activate Windows
Go to Settings to activate Windows.

localhost:10200/MiniDBMS/data/ x +

localhost:10200/MiniDBMS/database?name=collegeDB&act=select

Go Back

collegeDB->college

	cid	cname	cphn	cmail	fees
1	mit		44332345	mit@gmail.c...	20000
2	ceg				20000
3	ssn		44982356	ssn@gmail...	20000
4	srm		43574357	srm@gmail...	20000
5	vlt		87348734	vlt@gmail.co...	20000

DROP CREATE

Activate Windows
Go to Settings to activate Windows.

Drop Column:

This allows user to delete a column after the creation of table.

Syntax: Table-name field **Column-name** drop

Example: college field clocation drop

localhost:10200/MiniDBMS/data/ x +

localhost:10200/MiniDBMS/database?name=collegeDB&act=select

Go Back

collegeDB-> college

	cid	cname	cphn	cmail	clocation
college	1	mit	44332345	mit@gmail.c...	
	2	ceg			
	3	ssn	44982356	ssn@gmail...	
	4	srm	43574357	srm@gmail...	
	5	vit	87348734	vit@gmail.co...	

college field clocation drop

DROP CREATE

Activate Windows
Go to Settings to activate Windows.

localhost:10200/MiniDBMS/data/ x +

localhost:10200/MiniDBMS/database?name=collegeDB&act=select

Go Back

collegeDB-> college

	cid	cname	cphn	cmail
college	1	mit	44332345	mit@gmail.com
	2	ceg		
	3	ssn	44982356	ssn@gmail.com
	4	srm	43574357	srm@gmail.com
	5	vit	87348734	vit@gmail.com

DROP CREATE

Activate Windows
Go to Settings to activate Windows.

Update:

Update is the feature which allows the user to edit the entered value in the table. The update command works with a condition given in the command line.

Syntax: Table-name update **value to be updated** where **condition**

Example: college update cphn=44568787 where cid=2;

localhost:10200/MiniDBMS/data/ x +

← → ↻ localhost:10200/MiniDBMS/data/ collegeDB-> college

Go Back

	cid	cname	cphn	cmail	fees
1	mit		44332345	mit@gmail.c.	20000
2	ceg				20000
3	ssn		44982356	ssn@gmail.	20000
4	srm		43574357	srm@gmail.	20000
5	vit		87348734	vit@gmail.co.	20000

college update cphn=44568787 where cid=2

DROP CREATE

Activate Windows
Go to Settings to activate Windows.

localhost:10200/MiniDBMS/data/ x +

← → ↻ localhost:10200/MiniDBMS/data/ collegeDB-> college

Go Back

	cid	cname	cphn	cmail	fees
1	mit		44332345	mit@gmail.c.	20000
2	ceg		44568787		20000
3	ssn		44982356	ssn@gmail.	20000
4	srm		43574357	srm@gmail.	20000
5	vit		87348734	vit@gmail.co.	20000

DROP CREATE

Activate Windows
Go to Settings to activate Windows.

Delete:

Delete is the command used to delete a row from the table. Also here a condition is given for what row has to be deleted.

Syntax: Table-name delete record where **condition**

Example: college delete record where cid=4

localhost:10200/MiniDBMS/data/ x +

localhost:10200/MiniDBMS/database?name=collegeDB&act=select

Go Back

collegeDB->college

	cid	cname	cphn	cmail	fees
1	mit	44332345	mit@gmail.c.	20000	
2	ceg	44568787		20000	
3	ssn	44982356	ssn@gmail...	20000	
4	srm	43574357	srm@gmail...	20000	
5	vit	87348734	vit@gmail.co.	20000	

college delete record where cid=4

DROP CREATE

Activate Windows
Go to Settings to activate Windows.

localhost:10200/MiniDBMS/data/ x +

localhost:10200/MiniDBMS/database?name=collegeDB&act=select

Go Back

collegeDB->college

	cid	cname	cphn	cmail	fees
1	mit	44332345	mit@gmail.c.	20000	
2	ceg	44568787		20000	
3	ssn	44982356	ssn@gmail...	20000	
5	vit	87348734	vit@gmail.co.	20000	

DROP CREATE

Activate Windows
Go to Settings to activate Windows.

Select:

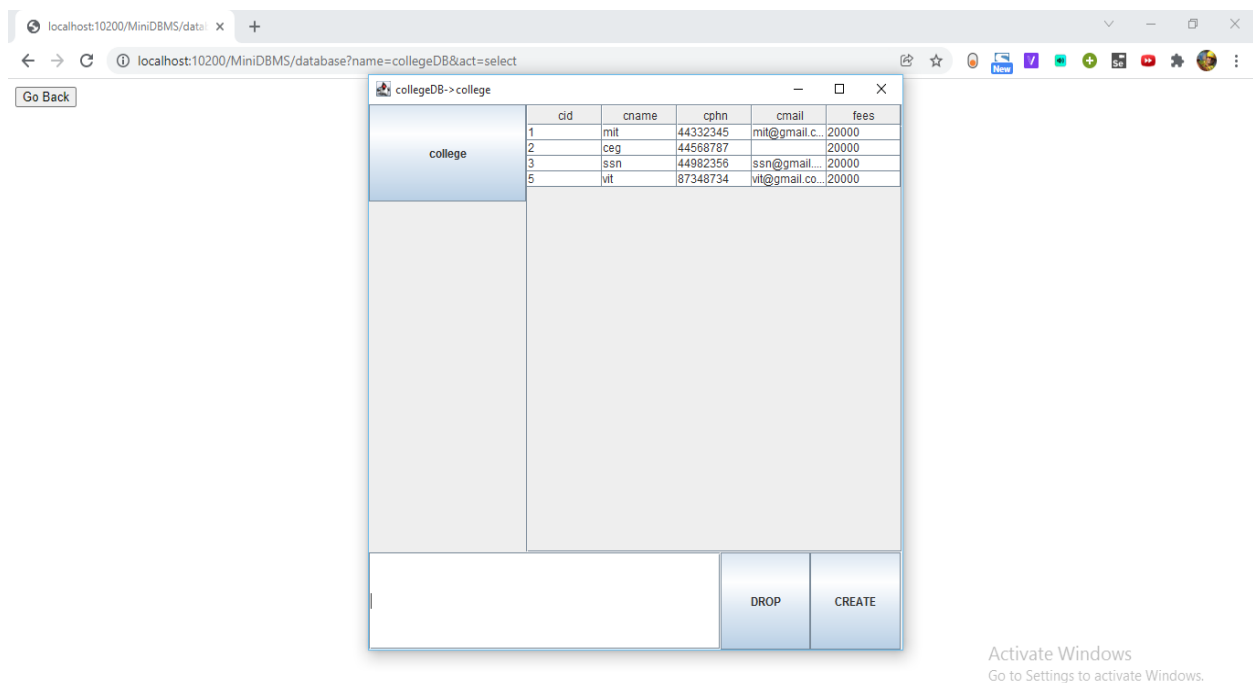
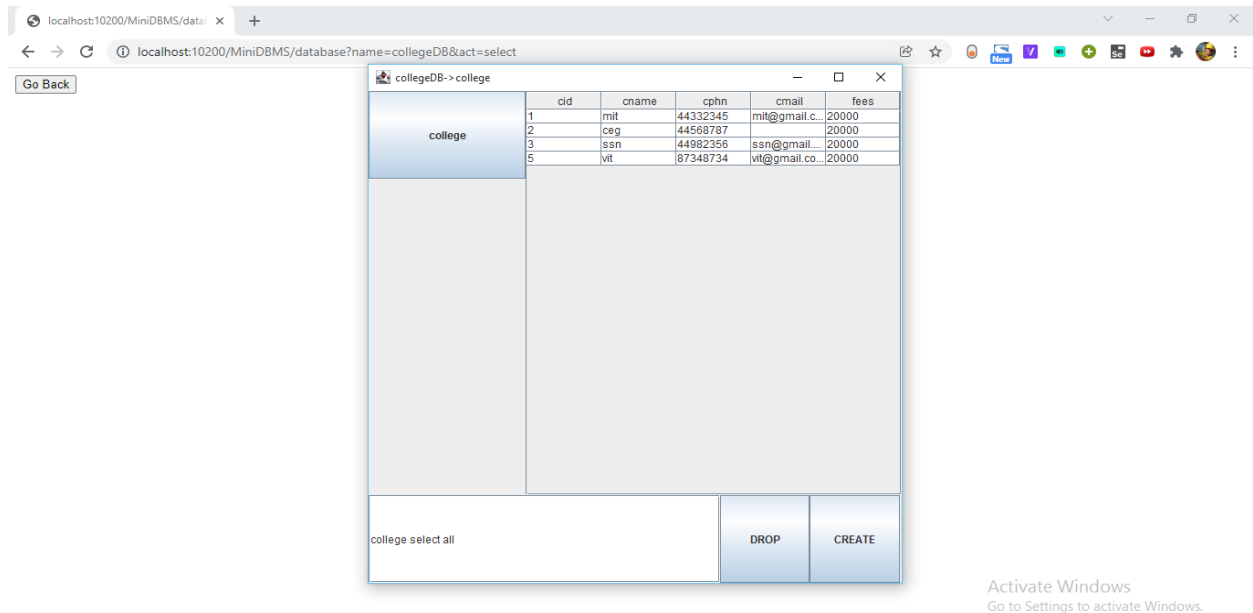
Select is a feature which allows the user to view the entered records in the table.

- **View all values :**

The user can view all the records inserted in the table

Syntax: Table-name select all

Example: college select all

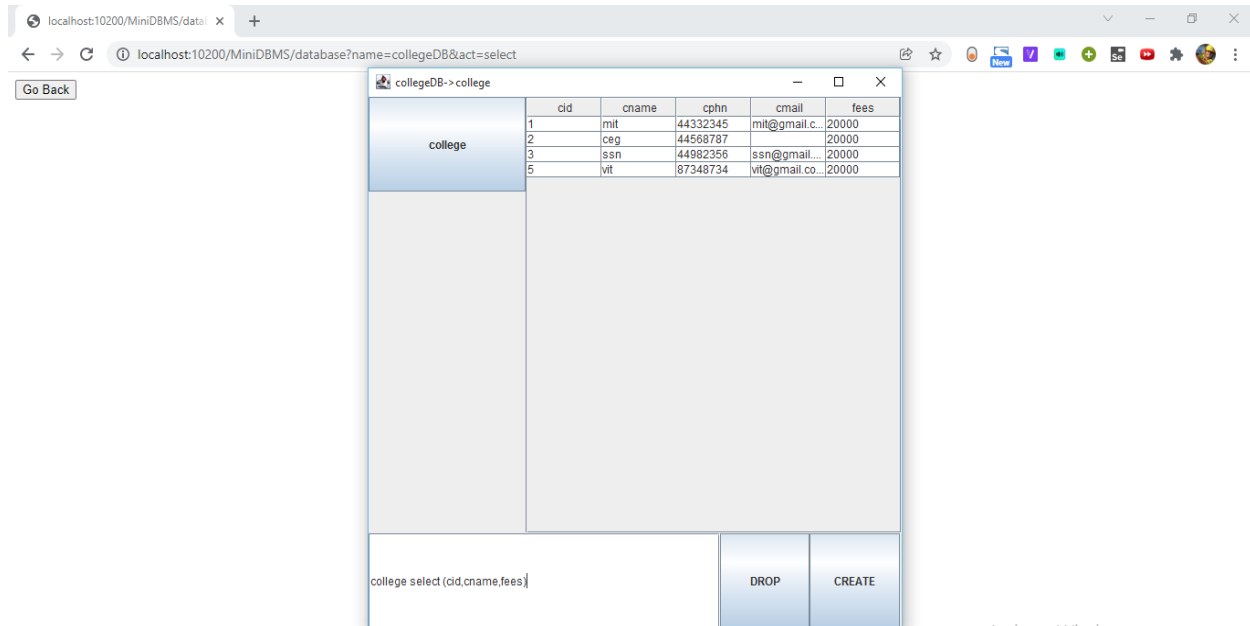


- **View Selected columns:**

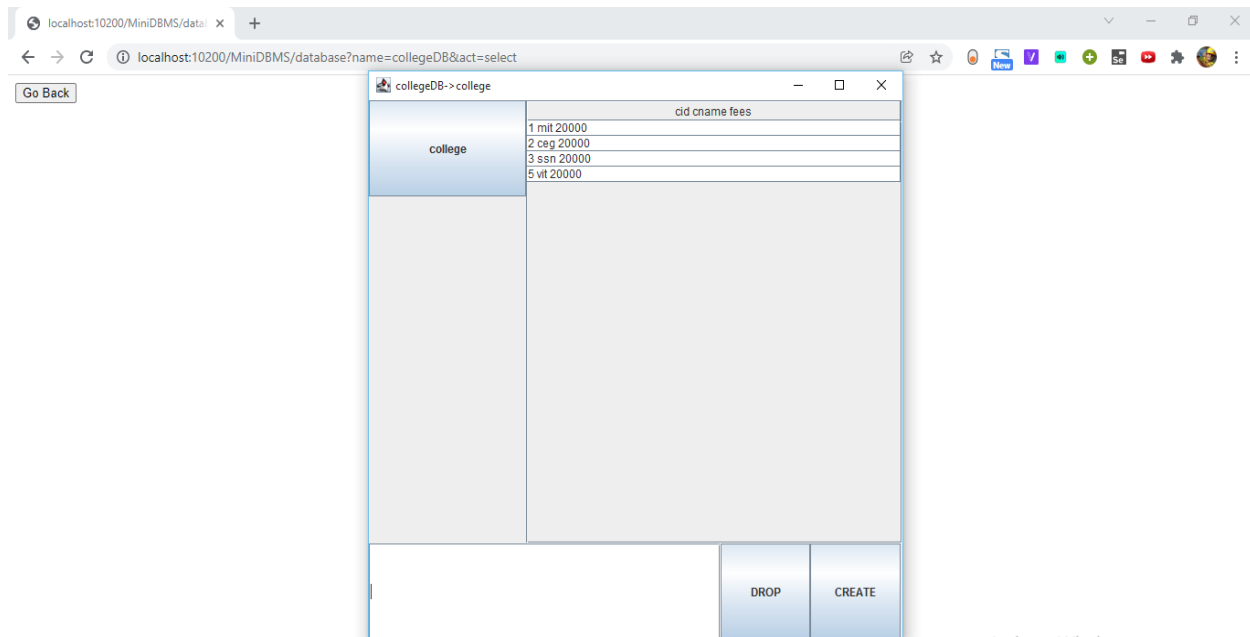
The user can select Records of which columns to be displayed.

Syntax: Table-name select (column_names)

Example: college select (cid,cname,fees)



Activate Windows
Go to Settings to activate Windows.



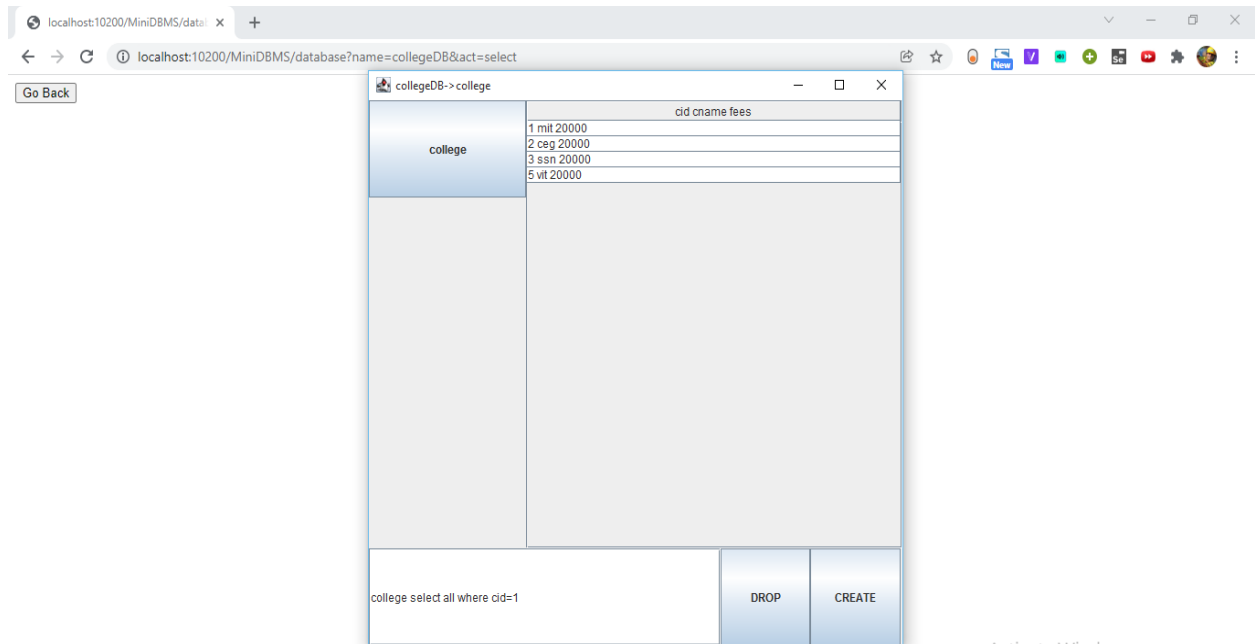
Activate Windows
Go to Settings to activate Windows.

- **View all records with a condition:**

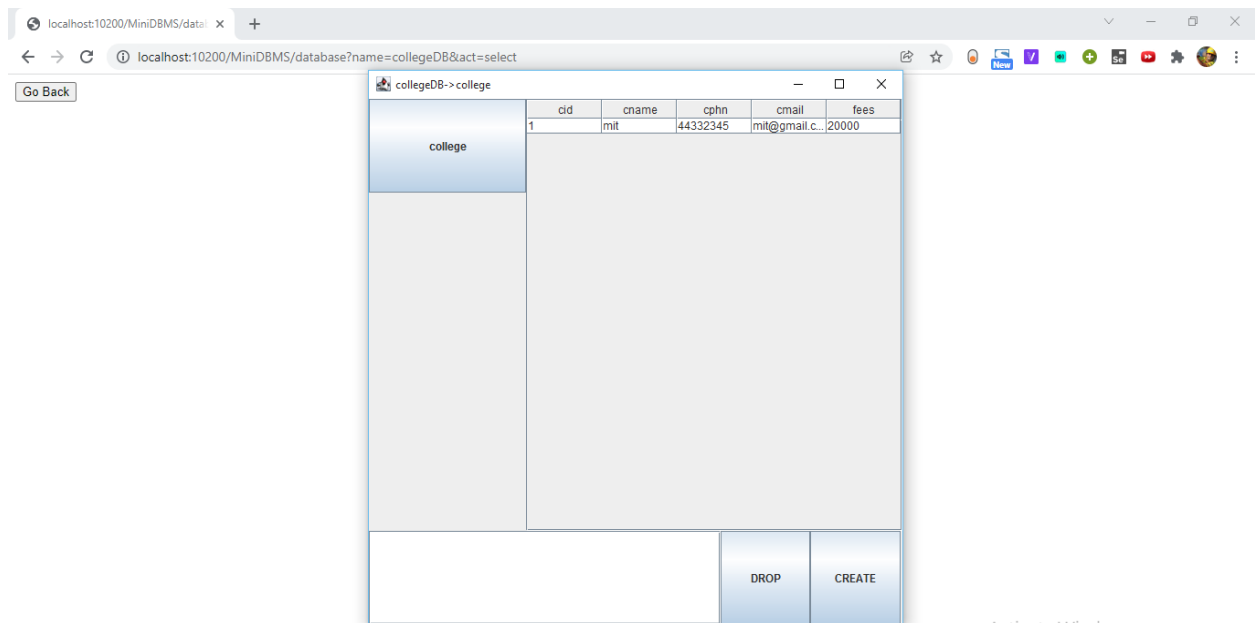
The user will be able to view all the records which satisfy the given condition.

Syntax: Table-name select all where **condition**

Example: college select all where cid=1



Activate Windows
Go to Settings to activate Windows.



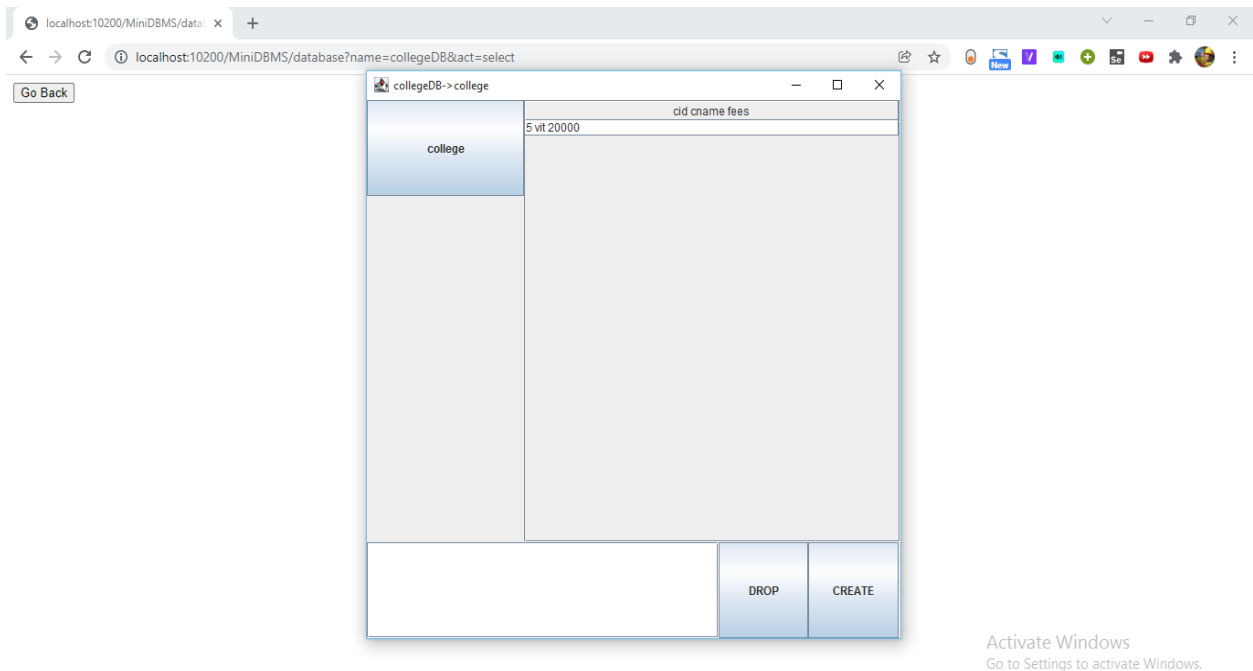
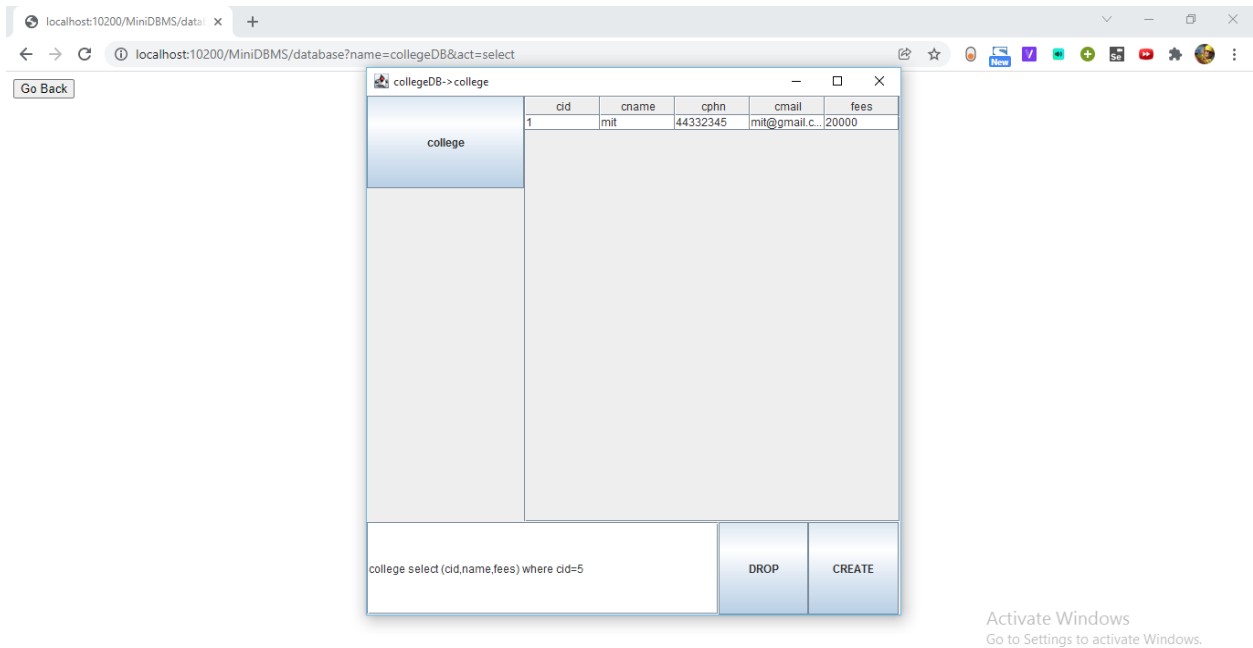
Activate Windows
Go to Settings to activate Windows.

- **View selected columns with a condition:**

The user will be able to select columns which has to be displayed and can give a condition to display the records.

Syntax: Table-name select (**column name**) where **condition**

Example: college select (cid,name,fees) where cid=5



Comparison of Table:

The user will be able to compare two tables within the same database or with different databases. This feature is implemented in the GUI. This feature enables the user with ease to compare the records insert and saves the extra work for comparison.

Report generation:

This feature allows the user to generate a csv file of the tables which will be sent to the given email. The user can access the file from his mail even after closing the DBMS.

The image displays two screenshots of a web application interface. The top screenshot shows the 'DATABASES' page of the MiniDBMS application, which includes a table of existing databases and a form to create a new one.

DATABASE NAME	SELECT	DELETE	
StudentDB	SELECT	DELETE	EXPORT
collegeDB	SELECT	DELETE	EXPORT
EmployeeDB	SELECT	DELETE	EXPORT

Below the table is a form with a text input labeled 'Name' and a blue button labeled 'CREATE DATABASE'.

The bottom screenshot shows a Gmail inbox on a mobile device. The top email is titled 'collegeDB DB TABLES EXPORTED' and contains the text 'collegeDB DB TABLES 1. college All the above tables are exported.' with a 'college.csv' attachment. Below it are three other emails: 'Sample Mail - Hello World from Sample Mail.', 'Critical security alert - Access for less secure apps setting has been turned on demouser241@gmail.com', and 'Security alert - A new sign-in on Windows demouser241@gmail.com'.

