

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018**



**A PROJECT REPORT
ON
DIGITAL TOKEN SYSTEM
BY**

Vinutha N
4SF19CS187

Sinchana Rai B
4SF19CS161

In the partial fulfillment of the requirement for V Sem. B. E. (CSE)

DBMS LABORATORY WITH MINI PROJECT
Under the guidance of

Mrs. Pooja N S
Asst. Professor, Dept. of CS&E



**Department of Computer Science & Engineering
SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
An Autonomous Institution
Adyar, Mangaluru-575007
2021-22**

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
An Autonomous Institution
Adyar, Mangaluru – 575007

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled “**DIGITAL TOKEN SYSTEM**” is submitted in partial fulfillment for the requirement of V sem. B. E. (Computer Science & Engineering), “DBMS LABORATORY WITH MINI PROJECT” during the year 2021 – 22 is a result of bonafide work carried out by

Vinutha N

4SF19CS187

Sinchana Rai B

4SF19CS161

.....
Mrs. Pooja N S
Asst. Prof. Dept. of CS&E
SCEM, Mangaluru

.....
Dr. Pushpalatha K
HOD, Dept. of CS&E
SCEM, Mangaluru

Signature of the Examiners

1.

2.

ABSTRACT

A database is an organized collection of data, generally stored and accessed electronically from a computer system. The database management system (DBMS) is a software that interacts with end user, applications and the database itself to capture and analyze the data and provides facilities to administer the database. Primary goal of database management system is to provide a way to store and retrieve database information that is convenient and efficient

Our project is about DIGITAL TOKEN SYSEYEM' as it is very important to have valid document. A small error may cause major problem especially Aadhar card, ration card and voter id, but in order to solve this problem we need to wait for a long time Infront of respective government offices. So inorder to reduce the time of people we have come up with an idea of DIGITAL TOKEN SYSTEM

So that if there are any issues in the document it can be sorted out where people need not wait for a long time as they are provided with token number along with a timeslot. Thus, people can come at particular time that they have provided and solve their issues.

We developed our project DIGITAL TOKEN SYSTEM that could help the people by providing token number along with timeslot by using NetBeans IDE, java, MySQL.

ACKNOWLEDGEMENT

Before we get in depth with the project, we might want to include few expressions of appreciation for the people who have been a piece of this project appropriate from its inception. The written work of this project has been one of the huge academic challenges we have faced and without the help, patience and guidance of the people involved, this assignment would not have been completed satisfactorily.

It is with great satisfaction that we are submitting the Mini Project Report on “**DIGITAL TOKEN SYSTEM**”. We have completed it as a part of the V semester DBMS Laboratory with Mini Project (18CSL58) of Bachelor of Engineering in Computer Science & Engineering of Vishveshwarya Technological University, Belagavi.

We are profoundly indebted to our guide, **Mrs. Pooja N S**, Assistant Professor, Department of Computer Science and Engineering for innumerable acts of timely advice and encouragement. We sincerely express our gratitude.

We express our sincere gratitude to **Dr. Pushpalatha K**, Head and Associate Professor, Department of Computer Science and Engineering for his invaluable support and guidance.

We sincerely thank **Dr. Rajesha S**, Principal, Sahyadri College of Engineering and Management.

Finally, we express our heartfelt thanks to our family, friends for our well- wishers who have provided encouragement throughout the work.

Vinutha N
4SF19CS187
V Sem, B.E., CSE
SCEM,Mangaluru

Sinchana Rai B
4SF19CS161
V Sem, B.E., CSE
SCEM,Mangaluru

PAGE INDEX

Chapter No.	Topic	Page No.
1.	Introduction	1
2.	Design	2
2.1	E-R diagram	3
2.2	Relational schema (ER to relational schema)	4-5
2.3	Schema diagram	6
3.	Normalization	7-9
4.	Implementation	10-16
5.	Results	17-20
6.	Conclusion	21
	References	22

CHAPTER 1

INTRODUCTION

INTRODUCTION TO DBMS

Database is a collection of data and Management System is a set of programs to store and retrieve those data. Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data.

Database Management System DBMS is a collection of inter-related data and set of programs to store and access those data in an easy and effective manner.

For Example: MySQL etc. are popular commercial DBMS used in different applications.

DBMS allows users the following tasks:

Data Definition: It helps in creation, modification and removal of definitions that define the organization of data in database.

Data Updating: It helps in insertion, modification and deletion of the actual data in the database.

Data Retrieval: It helps in retrieval of data from the database which can be used by applications for various purposes.

User Administration: It helps in registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control and recovering information corrupted by unexpected failure.

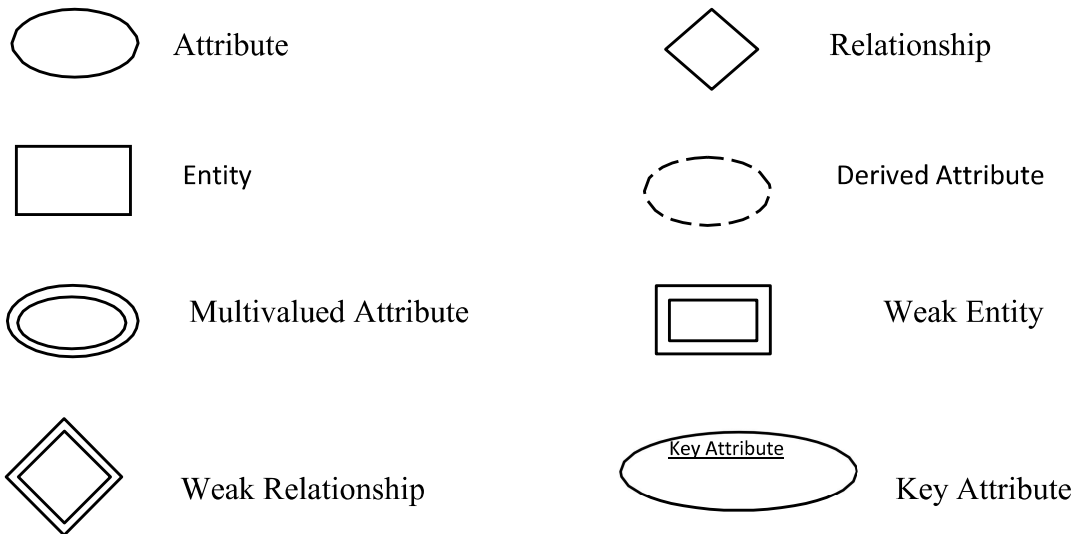
Software used is MySQL and NetBeans.

MySQL is free and open-source software under the terms of the GNU General Public License. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for *Linux, Apache, MySQL, Perl/PHP/Python*. MySQL is used by many database-driven web applications, including Drupal, Joomla, phub, and WordPress.

CHAPTER 2

DESIGN

THE ERD model consists of the following major components



Attribute: Attributes describe the instances in the column of a database.

Relationship: Relationships allow relational databases to split and store data in different tables, while linking disparate data items.

Entity: entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes.

Derived attribute: Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database.

Weak Entity: weak entity is an entity that cannot be uniquely identified by its attributes alone.

Key attribute: A key is an attribute in ER diagrams whose values are distinct for each individual entity in an entity set.

ER DIAGRAM

ER Diagram: An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. ER diagram of women harassment database is shown in below Figure 2.1.

ER DIAGRAM

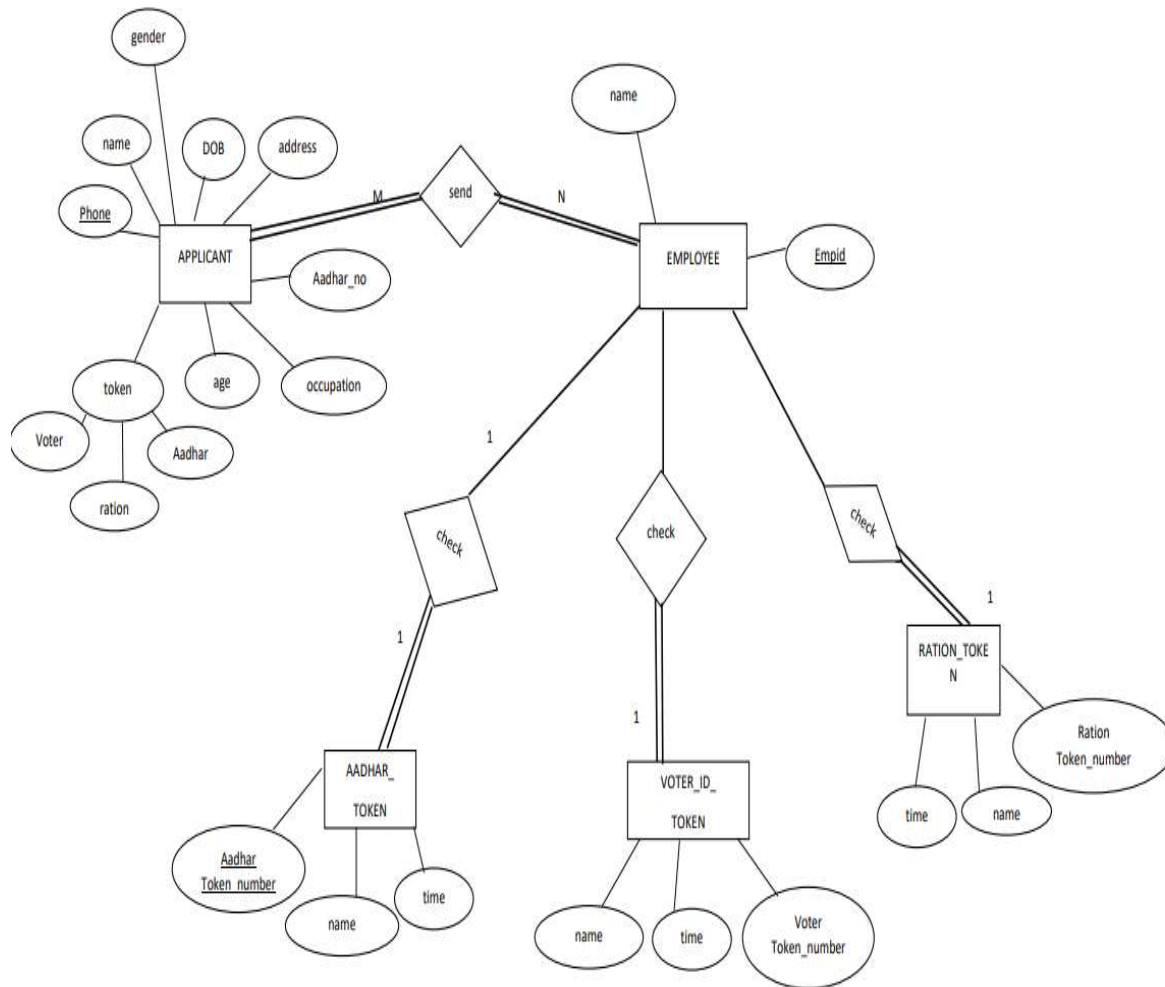


Figure 2.1: ER diagram of digital token system

ER-RELATIONAL SCHEMA

Steps to Draw Schema Diagram

Step 1: Convert strong entity sets into relation only simple attributes will be mapped in case of composite attributes. Exclude multi-valued attributes from mapping into relations.

APPLICANT

name	<u>phone</u>	DOB	gender	address	Aadhar_no	age	voter	ration	aadhar	occupation
------	--------------	-----	--------	---------	-----------	-----	-------	--------	--------	------------

EMPLOYEE

<u>Empid</u>	name
--------------	------

AADHAR TOKEN

time	name	<u>Aadhar_token_number</u>
------	------	----------------------------

VOTER TOKEN

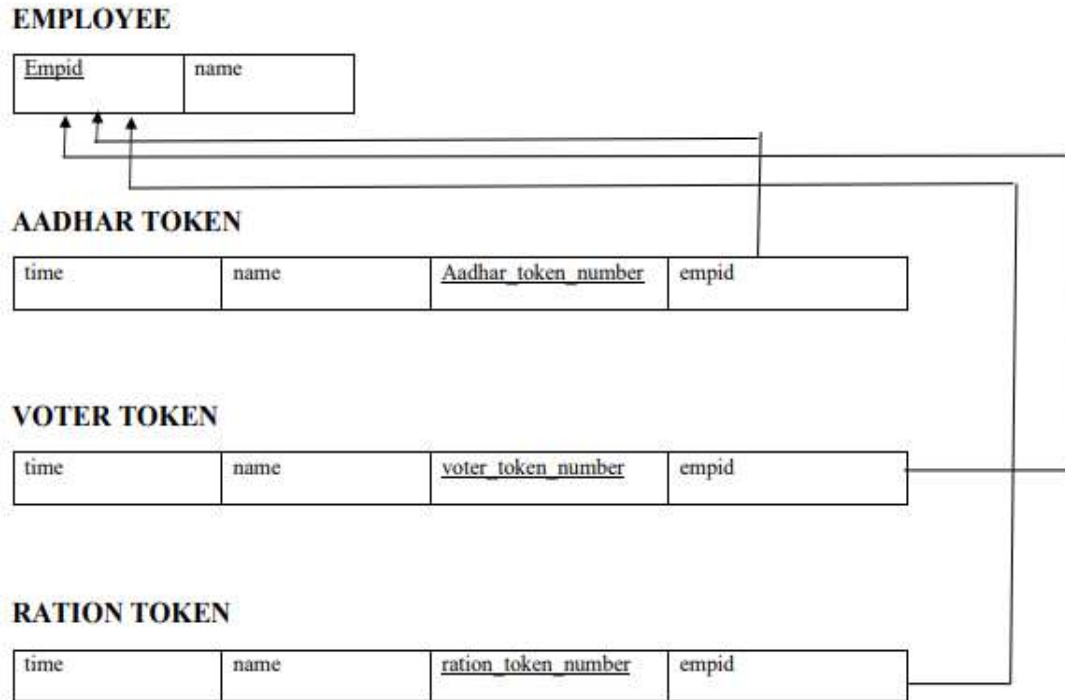
time	name	<u>voter_token_number</u>
------	------	---------------------------

RATION TOKEN

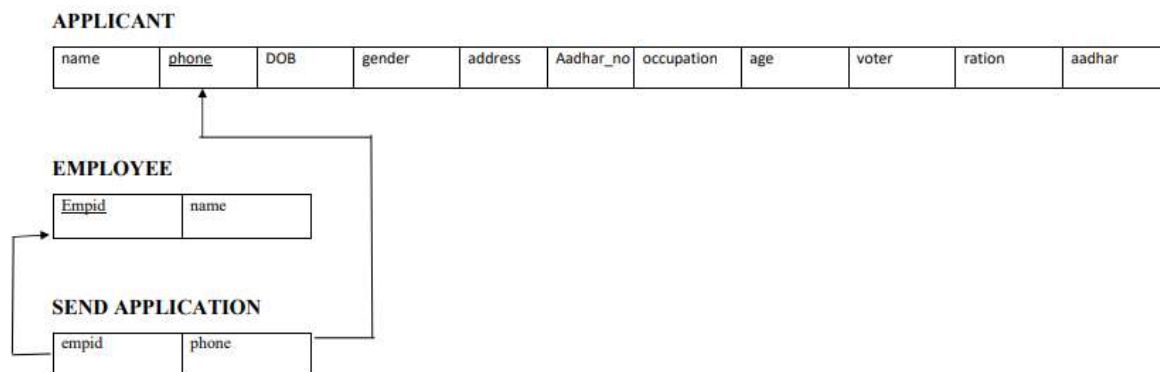
time	name	<u>ration_token_number</u>
------	------	----------------------------

Step 2: Mapping of weak entity types

Since there is no weak Entity, we have Excluded step 2

Step 3: Mapping of 1:1 relation type**Step 4: Mapping 1: N relation type**

Since there is no 1:N relation type, we have excluded step 4.

Step 5: Mapping of M: N Relationship types**Step 6: Mapping Multivalued Attributes**

Since there is no multivalued attribute, we exclude step 6.

Step 7: Mapping N-ary Relation

Here we don't have any N-ary Relationship, therefore there will be no mapping.

SCHEMA DIAGRAM

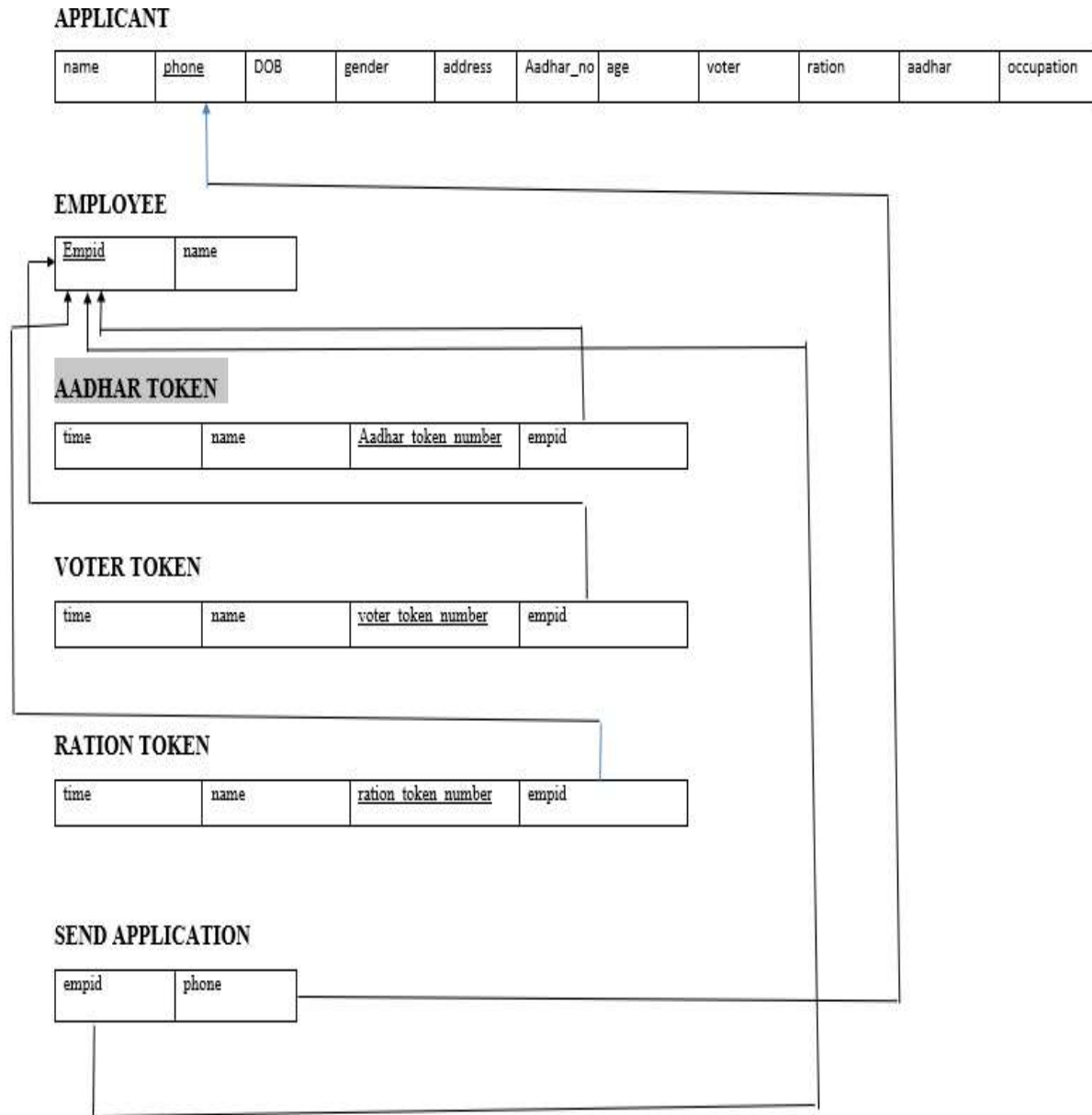


Figure 2.2: Final schema diagram of digital token system

The above figure 2.2 shows the schema diagram of digital token system.

CHAPTER 3

NORMALIZATION

STEPS OF NORMALIZATION

FIRST NORMAL FORM (1NF)

If the domain of each atomic values and the value of each attribute contains only single attribute from the domain

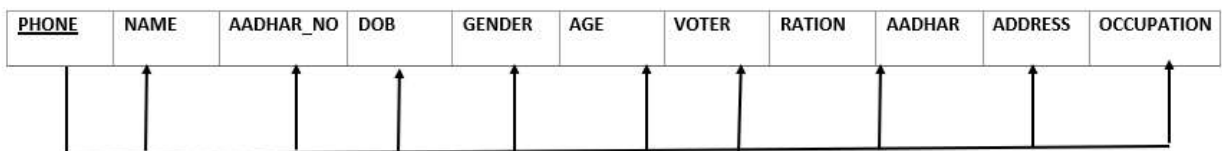
SECOND NORMAL FORM (2NF)

If the non-prime attribute is dependent on any proper subset of any candidate key of the relation

THIRD NORMAL FORM (3NF)

If the entity in 2NF and all the attribute in the table are determined only by the candidates of that relation and by any non-prime attributes, then the relation is in 3NF since there is no transitive dependency. Therefore, the relation used in this project are normalized and are in 1NF, 2NF, 3NF.

Applicant



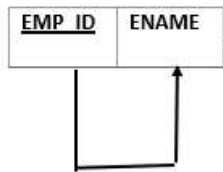
FD1

FD1 : phone->{name,aadhar_no,dob,gender,age,voter,ration,aadhar,address,occupation}

The given schema is in 1NF because it does not have any multi valued or composite attribute.

The given schema is in 2NF because every non-prime attribute have full functional dependency with the primary key. The given schema is in 3NF because it does not have any transitive dependency.

Employee



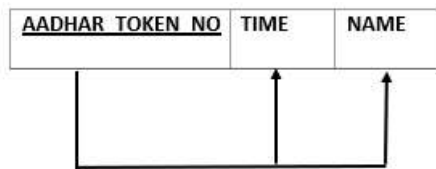
FD2

FD2: empid-> {name}

The given schema is in 1NF because it does not have any multi valued or composite attribute.

The given schema is in 2NF because every non-prime attribute have full functional dependency with the primary key because it does not have transitive dependency

Aadhar Token



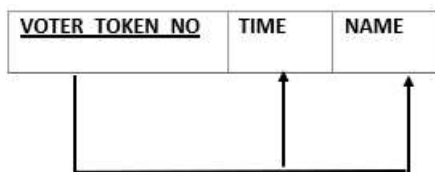
FD3

FD3: aadhar_token_number -> {time, name}

The given schema is in 1NF because it does not have any multi valued or composite attribute.

The given schema is in 2NF because all non-prime attributes have full functional dependency with the primary key. The given schema is in 3NF because it does not have any transitive dependency.

Voter Token

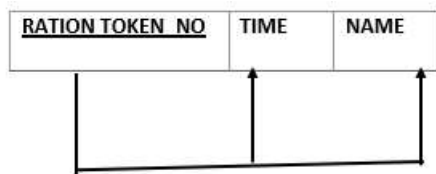


FD4

FD4 : voter_token_number- \rightarrow {time, name }

The given schema is in 1NF because it does not have any multi valued or composite attribute.

The given schema is in 2NF because every non-prime attribute have full functional dependency with the primary key. The given schema is in 3NF because it does not have any transitive dependency.

Ration Token**FD5**

FD5: ration_certificate_no \rightarrow { time, name }

The given schema is in 1nf because it does not have any multivalued or composite attribute.

The given schema is in 2NF because every no prime attribute gave full functional dependency with primary key.

The given schema is in 3NF because it does not have any transitive dependency.

CHAPTER 4

IMPLEMENTATION

System Specification

Operating System: Windows XP/7/8/10/MacOS/Linux.

Memory: Minimum of 1GB of RAM, Minimum of 2GB hard disk space.

Backend: MYSQL Frontend: APACHE NetBeans 12.2

Programming Language: Java

Table Structure

USERDASH

```
mysql> desc userdash;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name       | varchar(100)  | YES  |      | NULL    |      |
| AADHARNO   | varchar(20)   | YES  |      | NULL    |      |
| address    | varchar(150)  | YES  |      | NULL    |      |
| occupation | varchar(50)   | YES  |      | NULL    |      |
| token      | varchar(50)   | NO   |      | NULL    |      |
| gender     | varchar(50)   | YES  |      | NULL    |      |
| age        | int           | YES  |      | NULL    |      |
| phone      | varchar(20)   | NO   | PRI  | NULL    |      |
| dob        | date          | YES  |      | NULL    |      |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

EMPLOYEE

```
mysql> desc employee;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| employeeid | int           | NO   | PRI  | NULL    |      |
| name       | varchar(50)   | YES  |      | NULL    |      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

VOTER

```
mysql> desc voter;
```

Field	Type	Null	Key	Default	Extra
token	int	NO	PRI	NULL	
name	varchar(50)	YES		NULL	
time	varchar(50)	YES		NULL	

3 rows in set (0.00 sec)

AADHAR

```
mysql> desc aadhar;
```

Field	Type	Null	Key	Default	Extra
token	int	NO	PRI	NULL	
name	varchar(50)	YES		NULL	
time	varchar(20)	YES		NULL	

3 rows in set (0.00 sec)

RATION

```
mysql> desc ration;
```

Field	Type	Null	Key	Default	Extra
token	int	NO	PRI	NULL	
name	varchar(50)	YES		NULL	
time	varchar(50)	YES		NULL	

3 rows in set (0.00 sec)

Functionalities

Connecting to Database

The “Digital token system” has been developed by java. It uses MySQL database for storing the data and it is connected by the following code.

```
Class.forName("com. mysql. jdbc.Driver");
```

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/final","root",  
"vinutha2001");
```

INSERT

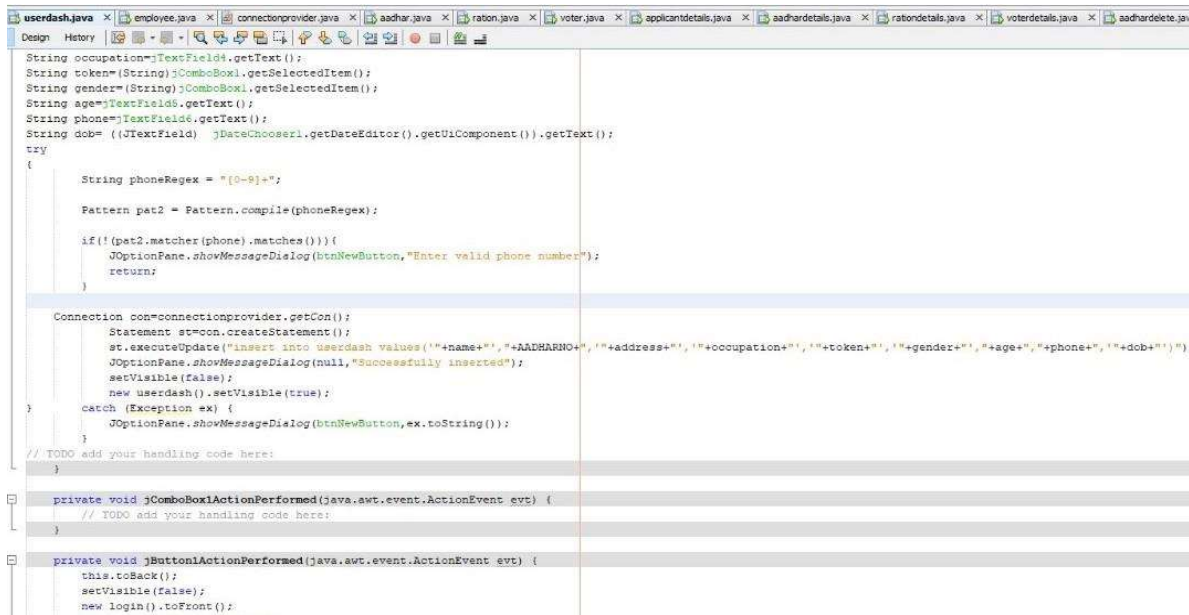
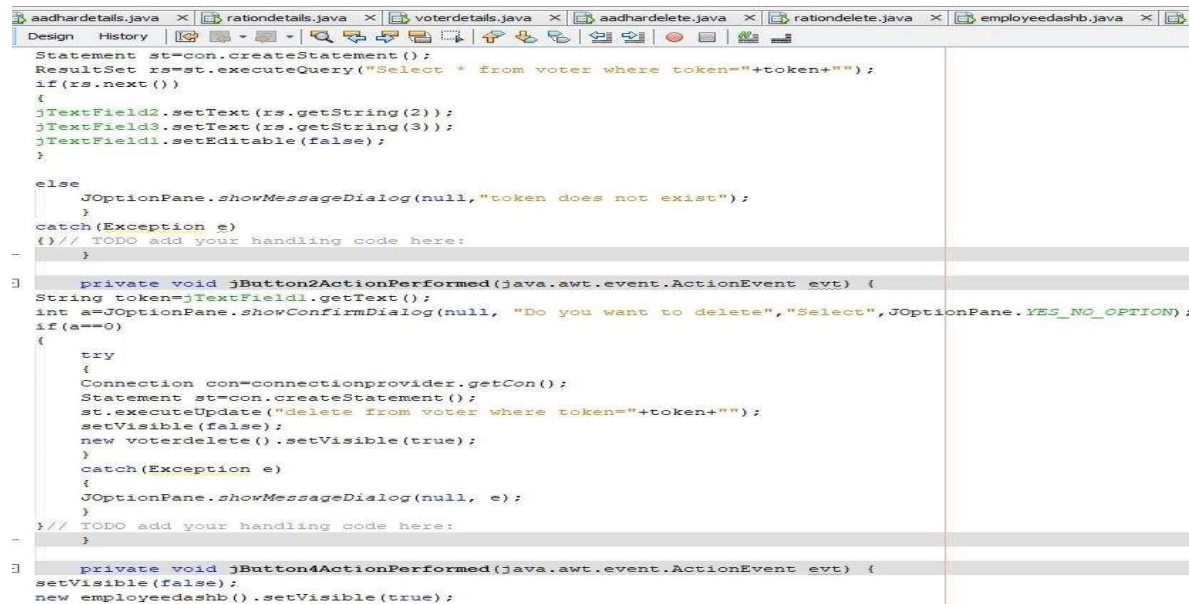


Figure 4.1: code for insert

The above figure 4.1 refers the backend code for insert table to the database.

DELETE



```

aahardetails.java x rationdetails.java x voterdetails.java x aahardelete.java x rationdelete.java x employeeedashb.java x
Design History
Statement st=con.createStatement();
ResultSet rs=st.executeQuery("Select * from voter where token="+token+"");
if(rs.next())
{
    jTextField2.setText(rs.getString(2));
    jTextField3.setText(rs.getString(3));
    jTextField1.setEditable(false);
}
else
{
    JOptionPane.showMessageDialog(null,"token does not exist");
}
catch(Exception e)
{
    // TODO add your handling code here:
}

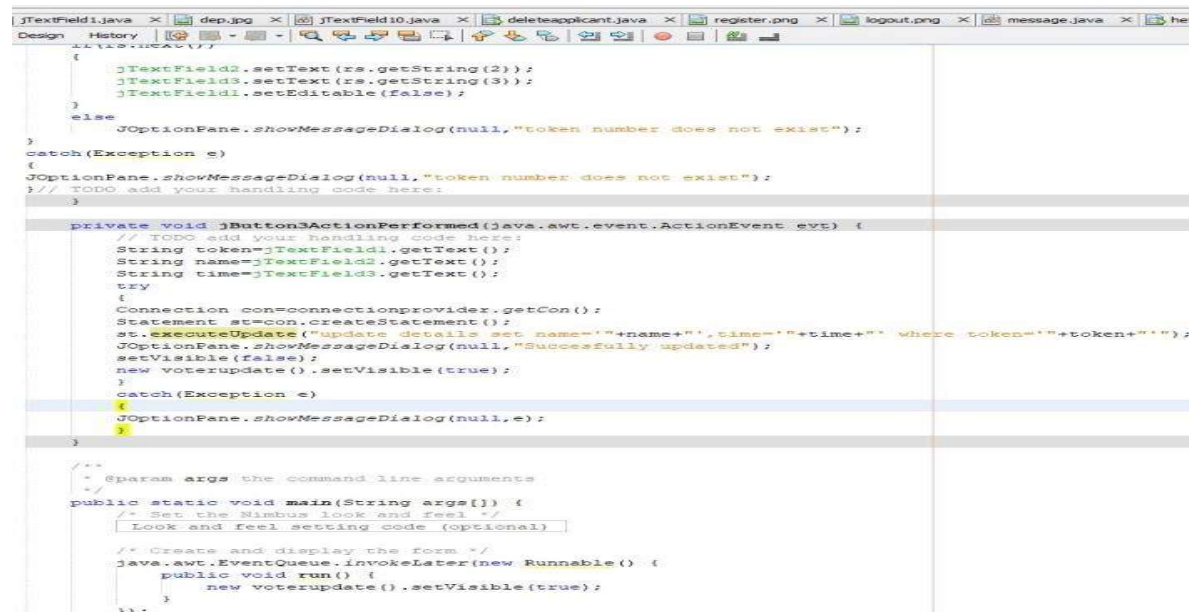
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
String token=jTextField1.getText();
int a=JOptionPane.showConfirmDialog(null, "Do you want to delete","Select",JOptionPane.YES_NO_OPTION);
if(a==0)
{
    try
    {
        Connection con=connectionprovider.getCon();
        Statement st=con.createStatement();
        st.executeUpdate("delete from voter where token="+token+"");
        setVisible(false);
        new voterdelete().setVisible(true);
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
}
// TODO add your handling code here:
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
setVisible(false);
new employeeedashb().setVisible(true);
}
  
```

Figure 4.2: code for delete

The above figure 4.2 refers the backend code for the delete operations to tables of database.

SEARCH



```

jTextField1.java x dep.jpg x jTextField10.java x deleteapplicant.java x register.png x logout.png x message.java x hel
Design History
{
    jTextField2.setText(rs.getString(2));
    jTextField3.setText(rs.getString(3));
    jTextField1.setEditable(false);
}
else
{
    JOptionPane.showMessageDialog(null,"token number does not exist");
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,"token number does not exist");
}
// TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
String token=jTextField1.getText();
String name=jTextField2.getText();
String time=jTextField3.getText();
try
{
    Connection con=connectionprovider.getCon();
    Statement st=con.createStatement();
    st.executeUpdate("update details set name='"+name+"',time='"+time+"' where token='"+token+"'");
    JOptionPane.showMessageDialog(null,"Successfully updated");
    setVisible(false);
    new voterupdate().setVisible(true);
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,e);
}
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    /* Look and feel setting code (optional) */

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new voterupdate().setVisible(true);
        }
    });
}
  
```

Figure 4.3: code for search

The above figure 4.3 refers to the backend code for the search values stored in the tables of the database.

TRIGGER



```

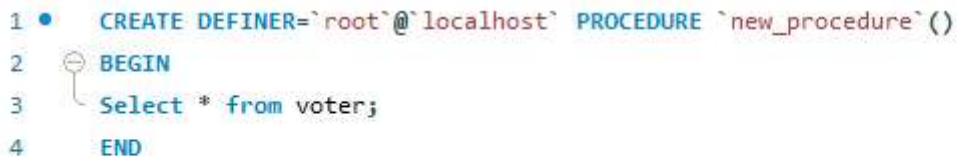
1 CREATE DEFINER='root'@'localhost' TRIGGER `userdash_BEFORE_INSERT` BEFORE INSERT ON `userdash` FOR EACH ROW BEGIN
2     IF(LENGTH(NEW.phone) < 10)
3     THEN
4         signal sqlstate '45000';
5         set message_text = 'Phone Number is not valid';
6     END IF;
7     IF(LENGTH(NEW.phone) > 10)
8     THEN
9         signal sqlstate '45000';
10        set message_text = 'Phone Number is not valid';
11    END IF;
12    IF(LENGTH(NEW.AADHARNO) < 12)
13    THEN
14        signal sqlstate '45000';
15        set message_text = 'AADHARNO is not valid';
16    END IF;
17    IF(LENGTH(NEW.AADHARNO) > 12)
18    THEN
19        signal sqlstate '45000';
20        set message_text = 'AADHARNO is not valid';
21    END IF;
22 END

```

Figure 4.4: code for trigger

The above figure 4.4 refers to the backend code for the trigger create a trigger and store it in the database.

PROCEDURE



```

1 CREATE DEFINER='root'@'localhost' PROCEDURE `new_procedure`()
2 BEGIN
3     Select * from voter;
4 END

```

Figure 4.5: code for procedure

The above figure 4.5 refers to the backend code for the procedure create a procedure and store it in the database.

The above figure 4.6 refers to the backend code to the display of all information stored in the database.

CHAPTER 5

RESULTS

The DIGITAL TOKEN SYSTEM stores the applicant details, employee details, token details.

The output or the snapshot for all the table is as follows.

SNAPSHOTS

FIRST PAGE

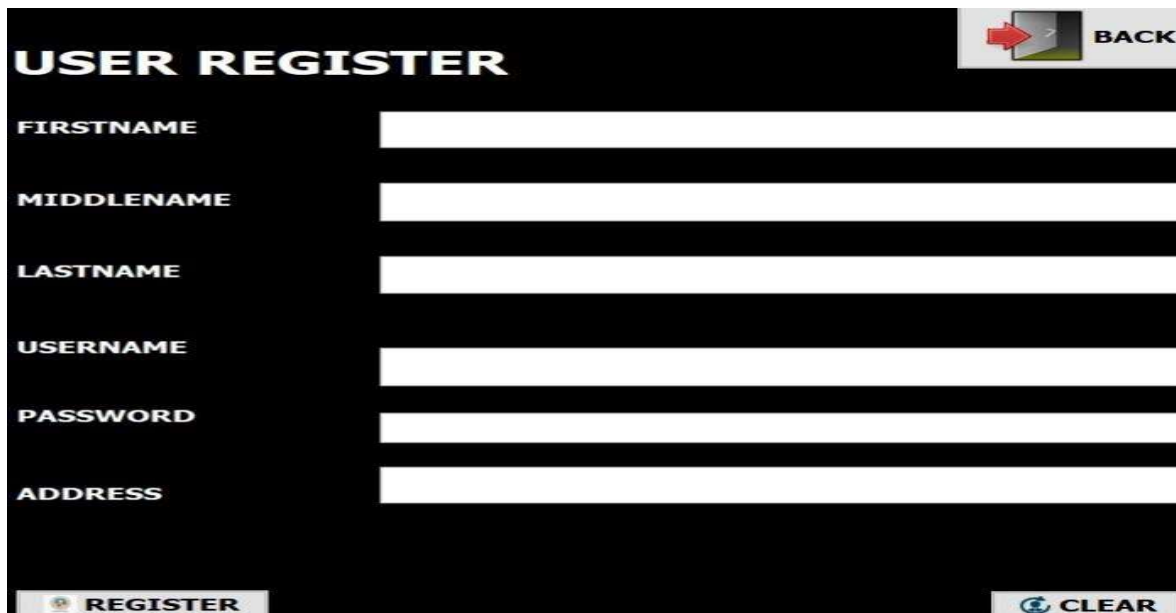


The screenshot displays the home page of the Digital Token System. At the top, there is a navigation bar with a user icon, a 'DEPARTMENT LOGIN' button, and an 'EXIT' button with a red arrow icon. The main content area has a black background with the text 'USER LOGIN' in green. Below this, there are two white input fields for 'USERNAME' and 'PASSWORD'. A 'show password' checkbox is located below the password field. At the bottom, there are three buttons: 'LOGIN' with a green arrow icon, 'REGISTER' with a user icon, and 'CLEAR' with a blue circular arrow icon.

Figure 5.1: Home page

Figure 5.1 shows a snapshot of a home page which directs the user to department login, login, register.

USER REGISTER PAGE

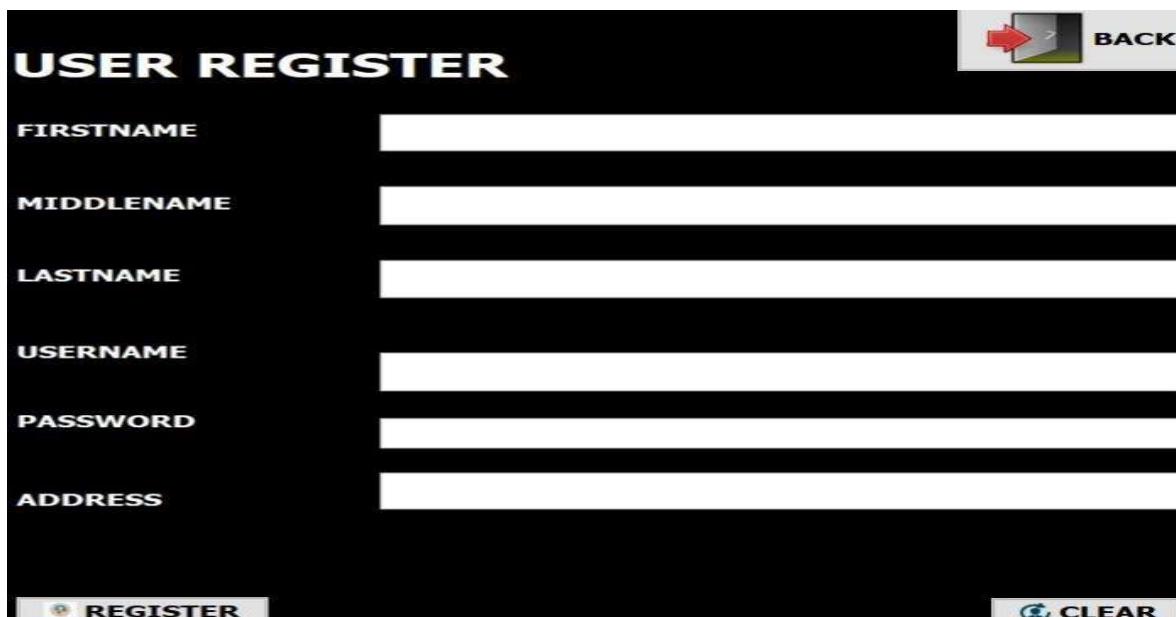


The screenshot shows a web form titled "USER REGISTER" in a large, bold, black font at the top left. In the top right corner, there is a "BACK" button with a red arrow icon pointing left. The form contains six input fields, each with a label to its left: "FIRSTNAME", "MIDDLENAME", "LASTNAME", "USERNAME", "PASSWORD", and "ADDRESS". Each label is in a bold, black font. The input fields are white with black borders. At the bottom left of the form, there is a "REGISTER" button with a green circular icon containing a white checkmark. At the bottom right, there is a "CLEAR" button with a blue circular icon containing a white eraser.

Figure 5.2: user register page

The above figure 5.2 directs the user to the user register page where he enter the his/her details.

DEPARTMENT REGISTER PAGE

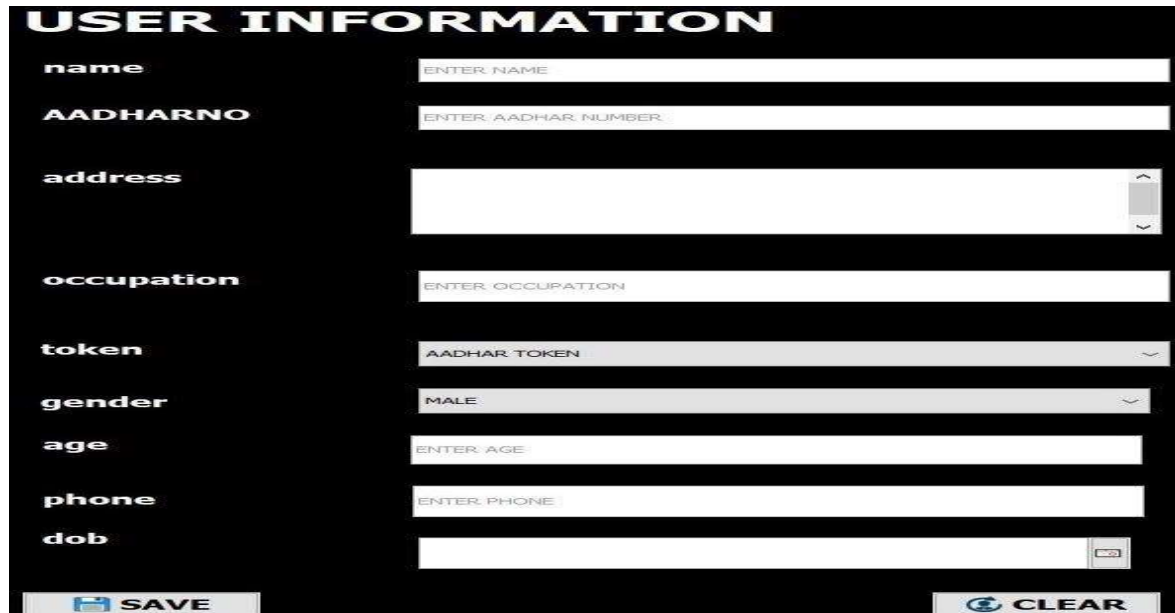


This screenshot is identical to the one in Figure 5.2, showing the "USER REGISTER" form. It features the same title, "BACK" button, input fields for "FIRSTNAME", "MIDDLENAME", "LASTNAME", "USERNAME", "PASSWORD", and "ADDRESS", and "REGISTER" and "CLEAR" buttons at the bottom.

Figure 5.3: department register page

The above Figure 5.3 directs to the department people for the use registration page and stores in the database.

USER APPLICATION PAGE

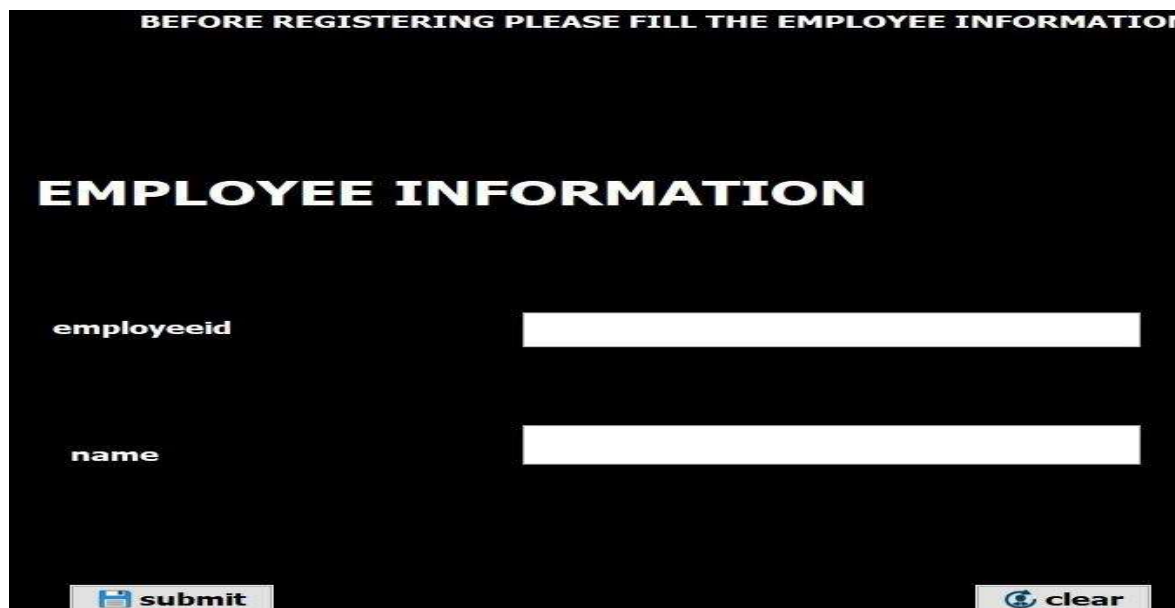


The screenshot shows a web form titled "USER INFORMATION" with a dark background. The form contains several input fields and dropdown menus. The labels on the left are: name, AADHARNO, address, occupation, token, gender, age, phone, and dob. The corresponding input areas on the right are: a text box with placeholder "ENTER NAME", a text box with placeholder "ENTER AADHAR NUMBER", a text box with placeholder "ADDRESS", a text box with placeholder "ENTER OCCUPATION", a dropdown menu with "AADHAR TOKEN", a dropdown menu with "MALE", a text box with placeholder "ENTER AGE", a text box with placeholder "ENTER PHONE", and a date picker for "dob". At the bottom, there are two buttons: "SAVE" with a floppy disk icon and "CLEAR" with a circular arrow icon.

Figure 5.4: user application page

The above Figure 5.4 directs the user to the use application page of the frontend and enter details.

EMPLOYEE INFORMATION



The screenshot shows a web form titled "EMPLOYEE INFORMATION" with a dark background. At the top, there is a header text: "BEFORE REGISTERING PLEASE FILL THE EMPLOYEE INFORMATION". The form contains two input fields: "employeeid" and "name". At the bottom, there are two buttons: "submit" with a floppy disk icon and "clear" with a circular arrow icon.

Figure 5.5: employee information

The above Figure 5.5 directs the employee to enter details in employee information page.

INSERTION PAGE

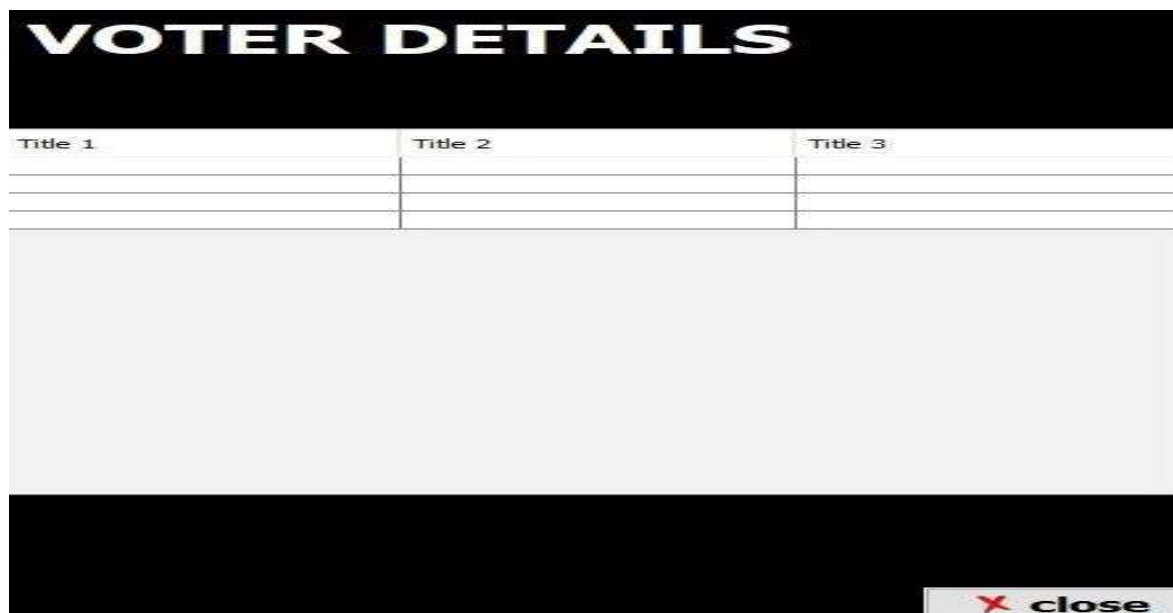


The image shows a web form titled "VOTER DETAILS" on a black background. It contains three input fields: "tokenno" with the value "100", "name" with a placeholder "enter name", and "time" which is empty. At the bottom, there are three buttons: "save" (with a magnifying glass icon), "clear" (with a circular arrow icon), and "close" (with a red 'X' icon).

Figure 5.6: insertion page

The above Figure 5.6 directs to the insertion page and stores in the database.

DISPLAY PAGE



The image shows a web page titled "VOTER DETAILS" on a black background. Below the title is a table with three columns labeled "Title 1", "Title 2", and "Title 3". The table has four rows, with the first row containing data and the subsequent three rows being empty. Below the table is a large grey rectangular area. At the bottom right, there is a "close" button with a red 'X' icon.

Title 1	Title 2	Title 3

Figure 5.7: display page

The above Figure 5.7 displays the details which are stored in the database.

UPDATE PAGE



VOTER DETAILS UPDATE

token

name

time

The form is titled "VOTER DETAILS UPDATE" in large, bold, black letters. Below the title, there are three input fields labeled "token", "name", and "time". To the right of the "token" field is a "search" button with a magnifying glass icon. At the bottom of the form, there are three buttons: "update" with a green plus icon, "reset" with a blue circular arrow icon, and "close" with a red X icon.

Figure 5.8 update page

The above Figure 5.8 directs the user to the update page which are stored in the database.

CHAPTER 6

CONCLUSION

Our project “DIGITAL TOKEN SYSTEM” stores the applicant details, user details, ration details, voter details which can view by department. So that the department can access the required information easily. Thus, applicant can apply for the token at given time slot.

For later references and completion of procedures of applying, they can easily access the previously applied fields.

REFERENCES

- [1] www.youtube.com
- [2] www.wikipedia.com
- [3] <https://www.mysqltutorial.org>
- [4] Fundamentals of Data base Systems (Fifth edition)