DBS Project

PASSWORD MANAGER

Documentation

Submitted By:

Madhav Gupta - 2020A7PS0106P

Pulkit Sinha - 2020A7PS1678P

System requirement specification (SRS)

We have developed the frontend using react framework of javascript and to integrate it with MySQL we have used Nodejs to handle the backend part of the project.

In order to run the project properly, a system must have

i. React

ii. Node js

iii. MySQL

Installed and running properly.

Dependencies for **Client Side (frontend - React)** ->

```
"dependencies": {
    "axios": "^0.26.1",
    "babel-preset-react": "^6.24.1",
    "cra-template": "1.1.3",
    "react": "^18.0.0",
    "react-dom": "^18.0.0",
    "react-scripts": "^5.0.1"
  },
```

Dependencies for **Server Side(backend - Node js)** ->

```
"dependencies": {
    "cors": "^2.8.5",
    "express": "^4.17.3",
    "mysql": "^2.18.1",
    "nodemon": "^2.0.15"
  }
```

Password Manager

A password manager is an advanced tool that is used to store and manage passwords at individual as well as business levels. Such a software has been in demand in the recent years because of the fact that today a large number of websites require you to login before using them and it is not possible for a person to remember passwords for every different website. A password manager takes in the password and **encrypts** it before storing it in the database and when the user wants to access the password it is **decrypted** and shown to the user. All a user needs to know to access all the passwords is one master pin, apart from the pin user also has a **master login** which is useful incase user forgets the pin.

Functionality:

Our webpage allows a user to **add a new password** to the database for which it compulsorily takes the name of the website, username and the password and gives user an option to enter notes and/or memo for the same. In case the user doesn't enter the password or the username or the website name, the webpage alerts the user to enter all the three.

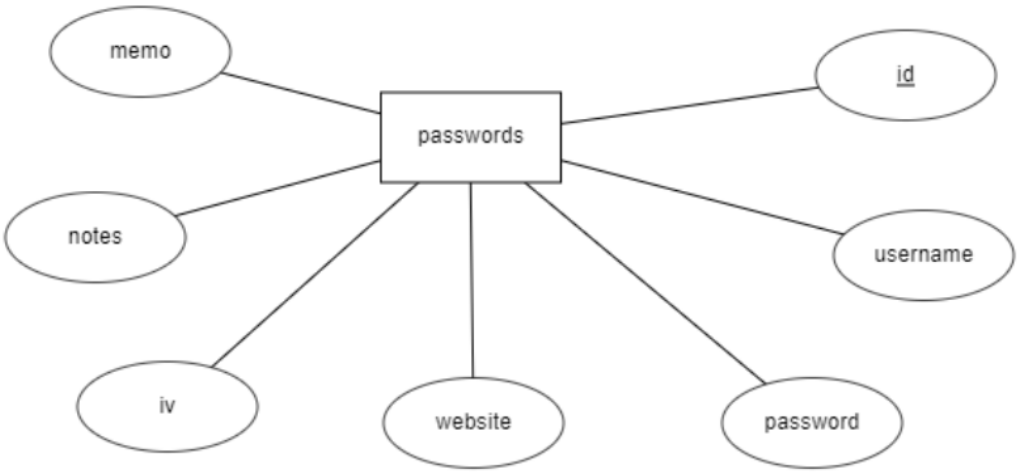
After adding a new password the website is added to the saved list of websites (data maintained in mysql database) and the relevant information like the username, notes and memo are visible throughout. For every

website, user is provided with a textbox to enter the pin and if the **pin is correct** the user can access the password and also a button is provided next to the password to directly **copy the password to the clipboard** once the correct pin is entered.

In case the user forgets the pin that is to be used to access all other passwords, we have a mechanism to deal with this scenario. We have provided the user with a **master username and password** which can be used to directly login into the system and upon login user is told the pin that has to be used to access the passwords making the software user friendly and able to handle the worst case scenario.

System Modelling

1. ER Diagram



2. Schema Design

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	☑	☑	☐	☐	☐	☐	☑	☐	
username	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
password	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
website	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
iv	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
notes	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
memo	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	☐	

3. Data Normalization

In our passwords table, id is a candidate key and is suitable for being used as a primary key. At first look, it might look like that password can also be used as a candidate key but actually, if a user uses the same passwords for different websites which is often the case, then in such case the password after AES encryption that is stored in the database will also be same which makes it unfit for being a candidate key. There are **no partial dependencies** which ensures

that our table is in **2NF** because the candidate key is a single attribute so there is no question of the existence of partial dependencies.

Coming to **transitive dependencies,** website, username, and password can not be dependent on each other as it is often a common practice of having the same username/password for different websites. Also, memos and notes cant be said to be dependent on any other attribute, and in our design, it is not compulsory to add memos/notes for each website, by default these entries will be null.
Apart from these attributes, the attribute iv which is the identifier for encryption is also not dependent on any other attribute as it is a random 16 bytes value used for encryption. So basically, we do not have any transitive dependencies which makes us believe that our table is in **3NF**.

If we dig deep into our table, we realize that all the dependencies that exist are related to the id itself, so basically, the LHS of any functional dependency is a superkey which makes our table in **BCNF** form which is quite good in removing redundancies.

4. List of Tables Required

Our project requires a **single table called passwords**, which contains the names of the website for which a password is stored in encrypted format along with the username user uses on that website and an option to add notes/ memo alongside the saved passwords. These passwords can only be accessed upon entering a pin.

The table password contains 7 attributes namely **id, username, password, website, iv, notes, memo** out of which **id is being used as the primary key** and is set to **not null** and **auto increment** also other attributes like username, password, website, iv are also set to not null and the default values of notes and memo is set to be NULL.

5. Additional Components

We have used an additional pin that must be known to the user in order to access the passwords he has previously stored. He can add passwords without knowing the **pin(BITS)**, but to access those passwords he must enter the pin. The passwords are internally stored in the database after AES encryption and as soon as the correct pin is entered, they are decrypted back and printed to the user. Apart from the pin, we have also used a **master username(DBS Project)** and a **master password(PR18)** which is being used to handle the case when the user forgets the pin.

Encryption

All the passwords entered by the user into the system have been encrypted and then stored into the database. For this purpose AES Encryption has been used. The Advanced Encryption Standard(AES) is a symmetric block cipher that is used to protect classified information.
All the passwords that are stored in the table named passwords are in the aes encrypted format and they are decrypted upon entering the correct pin and are then shown to the user. For each of these passwords, an iv(identifier) is also generated that is also being stored in the table. The identifier is later used to decrypt the password before showing it to the user.