ADP MINI-PROJECT

PASSMAN

TOPIC: PASSWORD MANAGEMENT SYSTEM

Team Members:

- BALAADITYA M - Web Design, Database, Integration
- DHEV SABHARISH S - Check Strength, Input Validation, View/Edit Passwords
- JAWAHAR KISHORE K - Random Password Generator, Idea of the Project

Project Overview:

Passman performs actions that allows the users to:

- Save their password along with website and email Id
- Check the strength of the password dynamically
- Randomly generate password according to their need
- View the saved passwords
- Edit/update the pre-existing passwords
- Delete the pre-existing passwords

Files:

- index.html
- home.html
- page2.html
- checkstrength.html
- about.html
- header.css
- main.css
- styles2.css
- style.css
- table.css
- script2.js
- pwstrength.js
- validation.js
- config.php

- db.php
- displaytable.php
- update.php
- editpass.php
- delete.php

Utility Core Functionalities:

Landing Page:

index.html

Getting website, email and password from user and storing in database:

home.html

validation.js: to validate the website name and email id provided by the user

pwstrength.js: to check the strength of the password provided by the user

db.php: create database(adp), table(data) and insert the data collected from the user in the table

Generating password according to user's need and storing in database:

page2.html

script2.js: generate random password according to the checkboxes selected by the user and displaying it to user

db.php: create database(adp), table(data) and insert the data collected from the user in the table

Displaying the password stored, edit and delete it:

displaytable.php: to display the data from the table(data) in the form of table to the user

editpassword.php: gets the new password from the user

update.php: updates the new password with respect to the website.

delete.php: delete the password chosen in displaytable.php

Checking the strength of the password:

Checkstrength.html

pwstrength.js: to check the strength of the password provided by the user

About:

about.html: brief description of the project

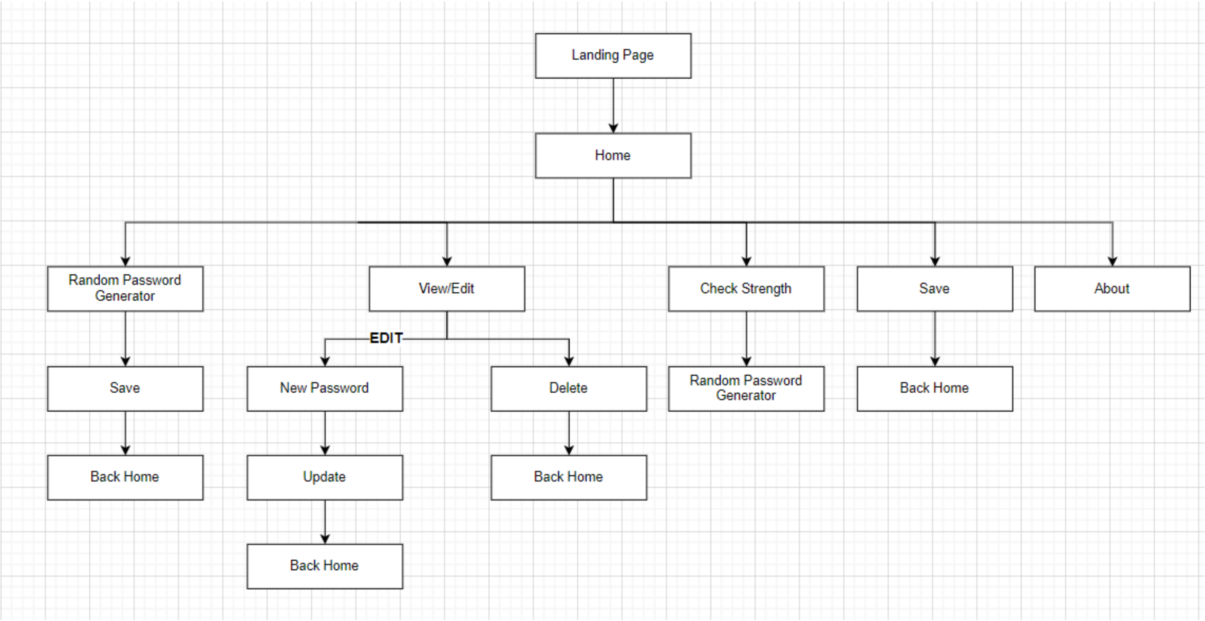
Database:

Database: adp

Table: data

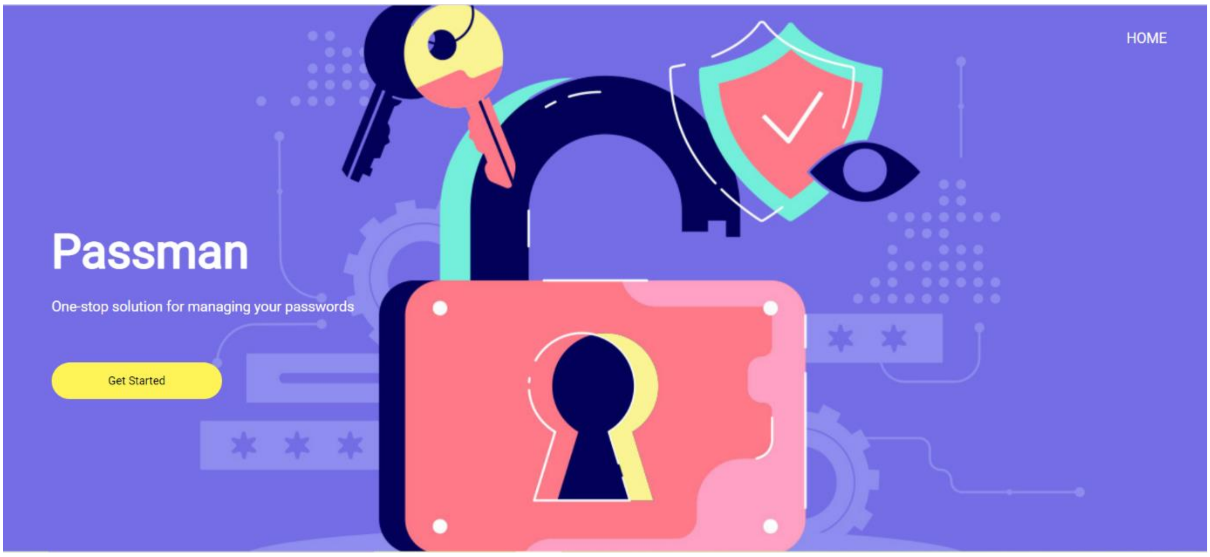
To store website, email and password.

FLOW OF THE APP



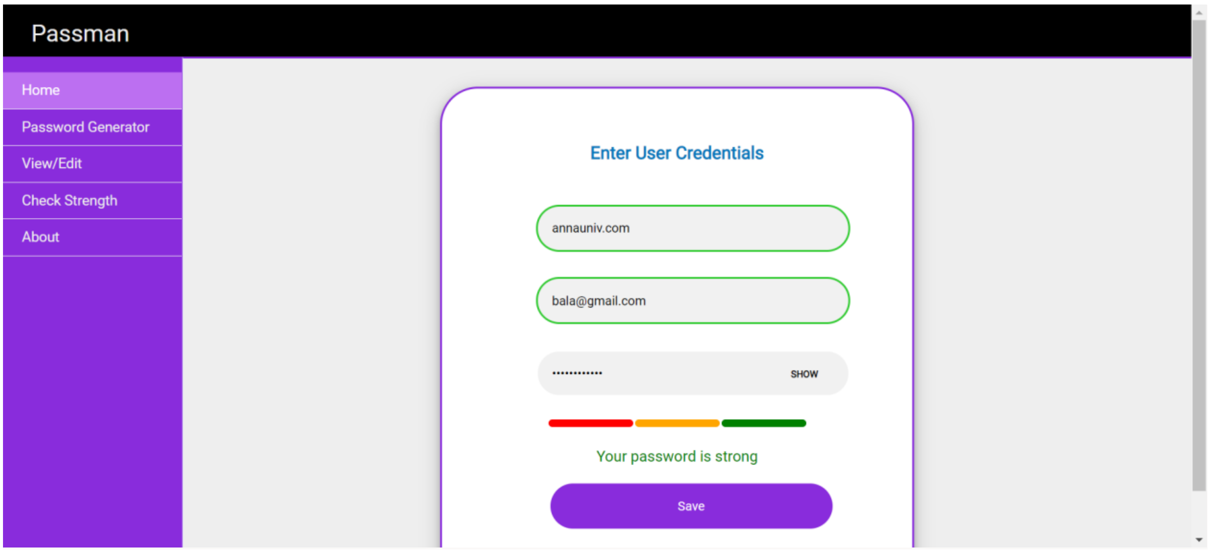
FUNCTIONS AND FLOW:

1.First, user is taken to the landing screen. On clicking **get started**, user is directed to the home page.



2.Home page: Save the password along with website name and email Id and check the strength of the password

- User need to provide the website name and email id in the given textboxes
- As the password is being typed, the strength of password is determined and displayed to the user
- on clicking **save**, the password for the provided website and email is saved



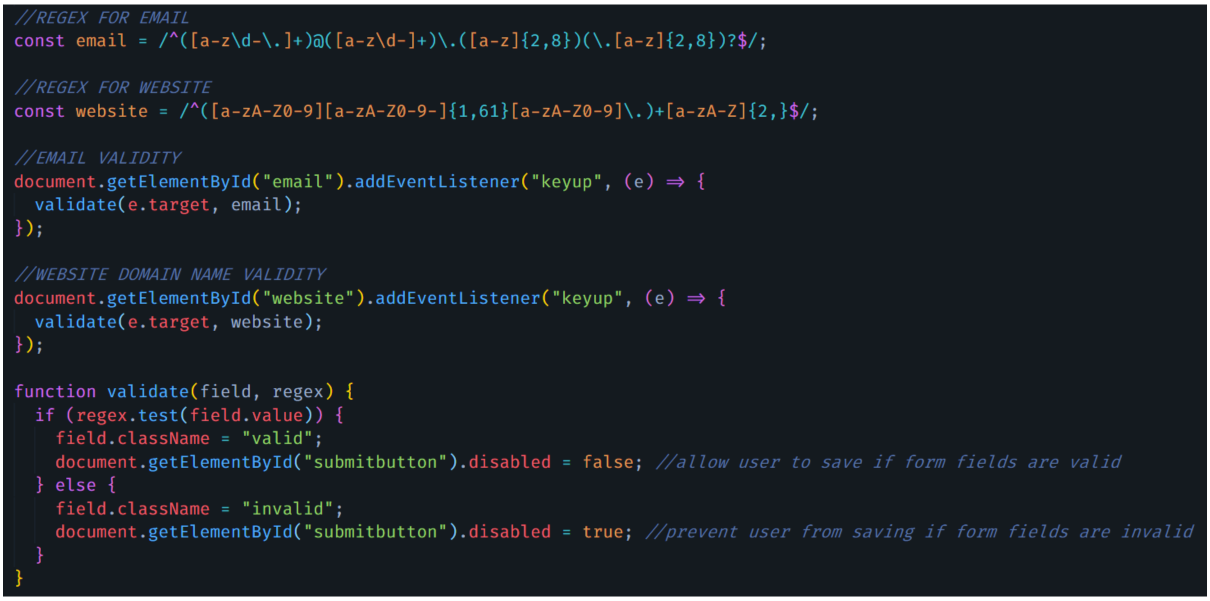
Enter User Credentials

annauniv.com

bala@gmail.com

••••••••••••

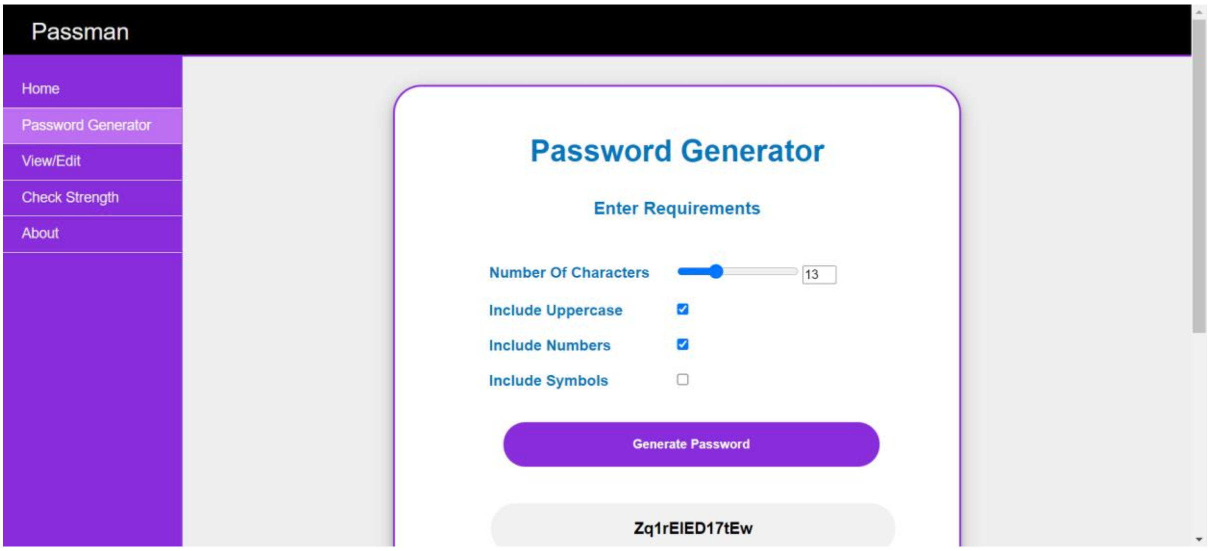
Your password is strong



```
//REGEX FOR EMAIL
const email = /^([a-z\d-\.]+)@([a-z\d-]+)\.([a-z]{2,8})(\.[a-z]{2,8})?$/;

//REGEX FOR WEBSITE
const website = /^([a-zA-Z0-9][a-zA-Z0-9-]{1,61}[a-zA-Z0-9]\.)+[a-zA-Z]{2,}$/;

//EMAIL VALIDITY
document.getElementById("email").addEventListener("keyup", (e) => {
  validate(e.target, email);
});

//WEBSITE DOMAIN NAME VALIDITY
document.getElementById("website").addEventListener("keyup", (e) => {
  validate(e.target, website);
});

function validate(field, regex) {
  if (regex.test(field.value)) {
    field.className = "valid";
    document.getElementById("submitbutton").disabled = false; //allow user to save if form fields are valid
  } else {
    field.className = "invalid";
    document.getElementById("submitbutton").disabled = true; //prevent user from saving if form fields are invalid
  }
}
```

3.Password generator: randomly generates the password according the user's need and save it along with the website and email id.

- On clicking **Password Generator** on the side navigation bar, user is taken to the password generator page
- User needs to select the length and the requirements for the password and click **generate password**
- The password will be generated and will be displayed.
- User further needs to enter valid website and email id and click on **save** button to the save the randomly generated along with the website and email



Password Generator

Enter Requirements

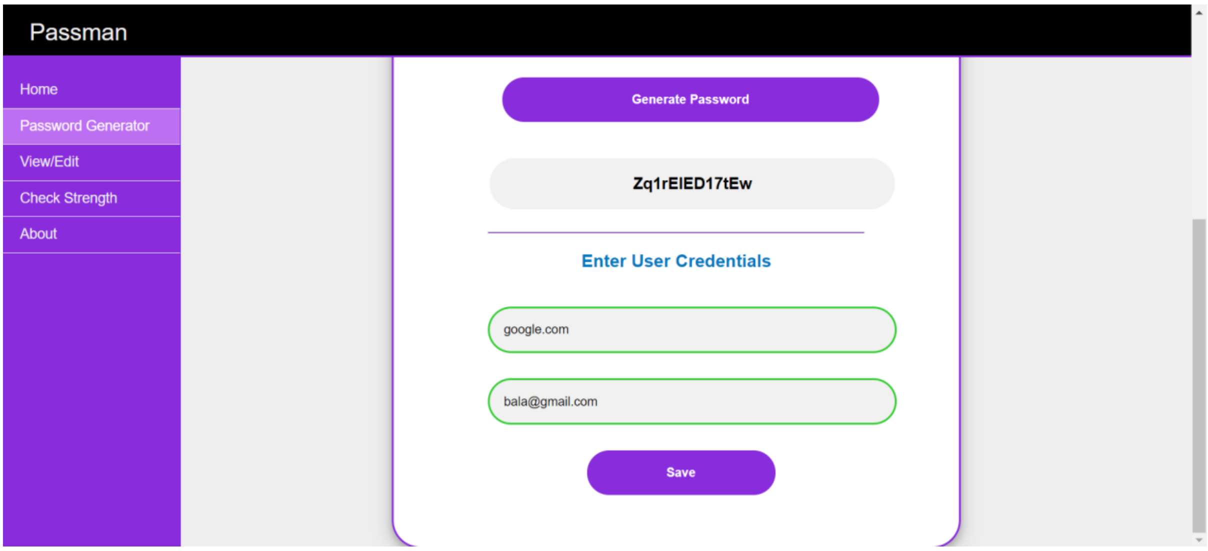
Number Of Characters 13

Include Uppercase ☑

Include Numbers ☑

Include Symbols □

Zq1rEIED17tEw

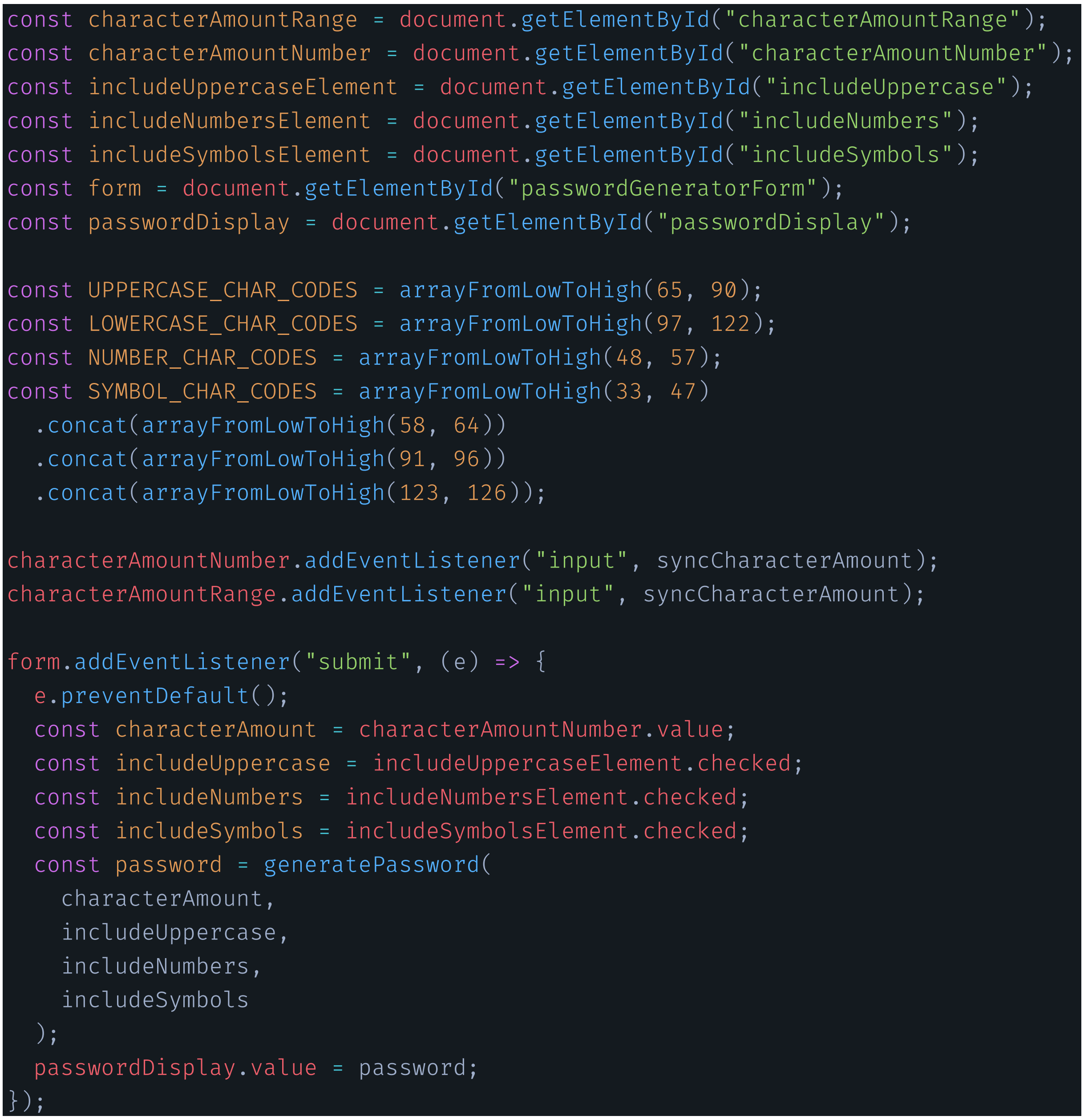


Zq1rEIED17tEw

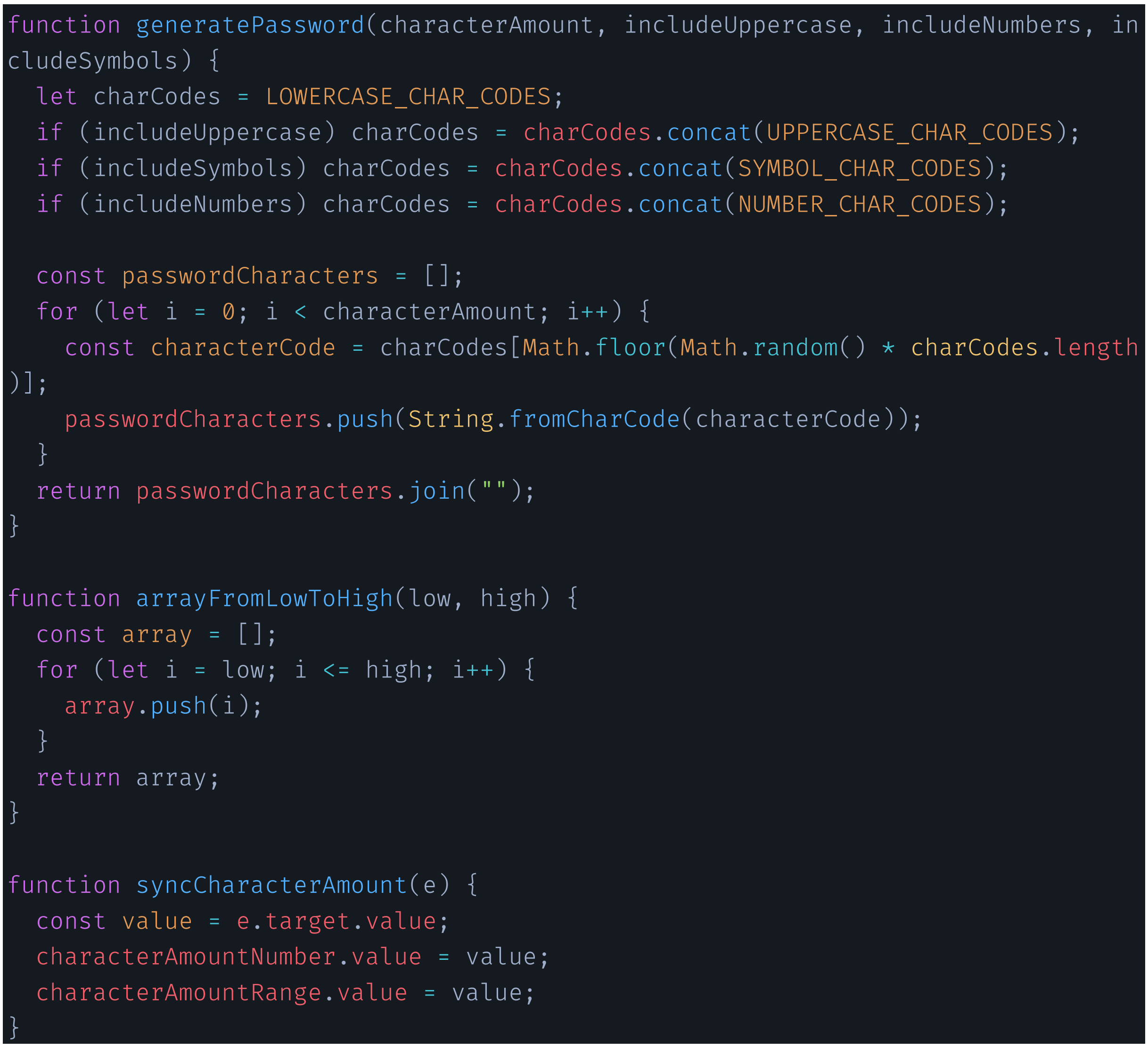
Enter User Credentials

google.com

bala@gmail.com



```
const characterAmountRange = document.getElementById("characterAmountRange");
const characterAmountNumber = document.getElementById("characterAmountNumber");
const includeUppercaseElement = document.getElementById("includeUppercase");
const includeNumbersElement = document.getElementById("includeNumbers");
const includeSymbolsElement = document.getElementById("includeSymbols");
const form = document.getElementById("passwordGeneratorForm");
const passwordDisplay = document.getElementById("passwordDisplay");

const UPPERCASE_CHAR_CODES = arrayFromLowToHigh(65, 90);
const LOWERCASE_CHAR_CODES = arrayFromLowToHigh(97, 122);
const NUMBER_CHAR_CODES = arrayFromLowToHigh(48, 57);
const SYMBOL_CHAR_CODES = arrayFromLowToHigh(33, 47)
  .concat(arrayFromLowToHigh(58, 64))
  .concat(arrayFromLowToHigh(91, 96))
  .concat(arrayFromLowToHigh(123, 126));

characterAmountNumber.addEventListener("input", syncCharacterAmount);
characterAmountRange.addEventListener("input", syncCharacterAmount);

form.addEventListener("submit", (e) => {
  e.preventDefault();
  const characterAmount = characterAmountNumber.value;
  const includeUppercase = includeUppercaseElement.checked;
  const includeNumbers = includeNumbersElement.checked;
  const includeSymbols = includeSymbolsElement.checked;
  const password = generatePassword(
    characterAmount,
    includeUppercase,
    includeNumbers,
    includeSymbols
  );
  passwordDisplay.value = password;
});
```



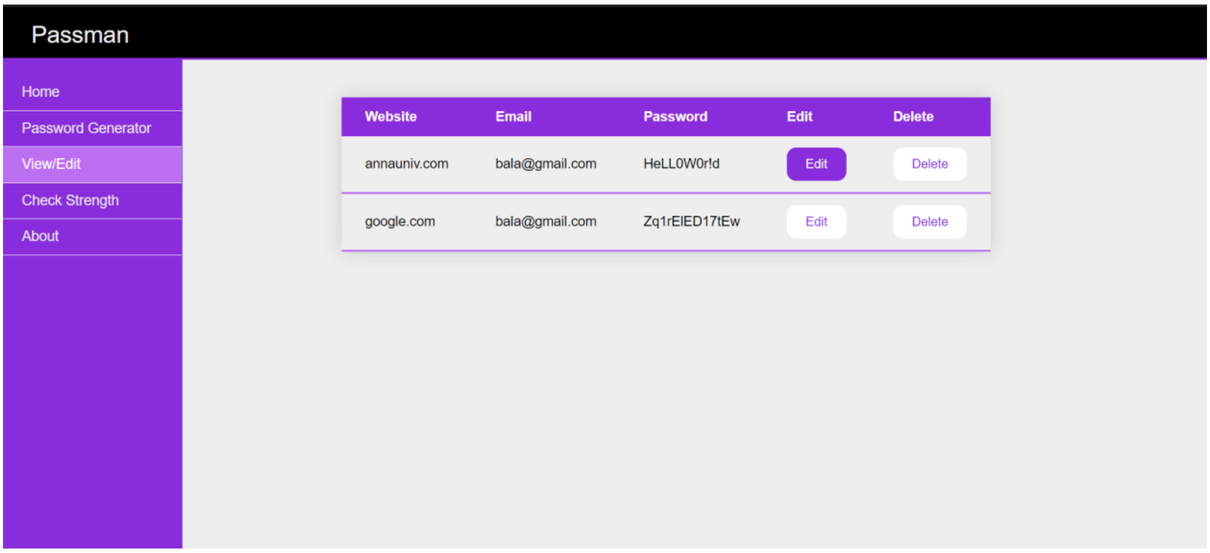
```
function generatePassword(characterAmount, includeUppercase, includeNumbers, includeSymbols) {
  let charCodes = LOWERCASE_CHAR_CODES;
  if (includeUppercase) charCodes = charCodes.concat(UPPERCASE_CHAR_CODES);
  if (includeSymbols) charCodes = charCodes.concat(SYMBOL_CHAR_CODES);
  if (includeNumbers) charCodes = charCodes.concat(NUMBER_CHAR_CODES);

  const passwordCharacters = [];
  for (let i = 0; i < characterAmount; i++) {
    const characterCode = charCodes[Math.floor(Math.random() * charCodes.length)];
    passwordCharacters.push(String.fromCharCode(characterCode));
  }
  return passwordCharacters.join("");
}

function arrayFromLowToHigh(low, high) {
  const array = [];
  for (let i = low; i <= high; i++) {
    array.push(i);
  }
  return array;
}

function syncCharacterAmount(e) {
  const value = e.target.value;
  characterAmountNumber.value = value;
  characterAmountRange.value = value;
}
```

4.View/Edit and Delete: allows user to view the previously stored passwords and edit/update them

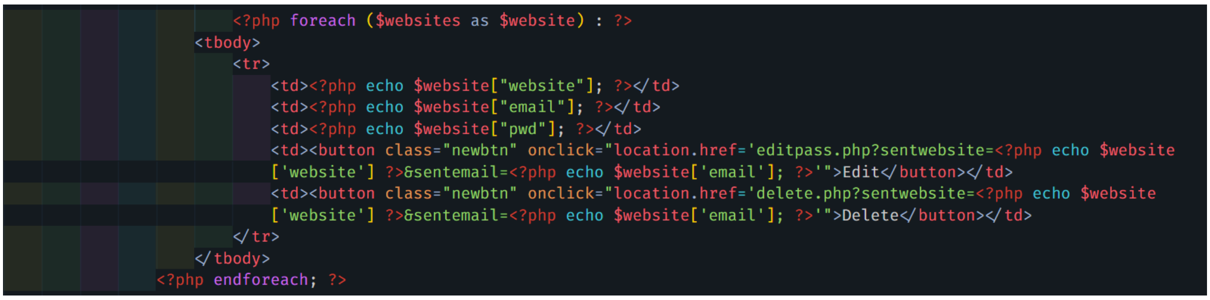
- On clicking **View/Edit** on the side navigation bar, user is taken to the View/Edit page.
- Here user can see the previously stores passwords along with respect to the website and email.
- User can edit the password by clicking the **edit** button for the specific row.
- User can now enter the new password whose strength will be dynamically checked and displayed.

- On clicking the save button, the new password will be updated

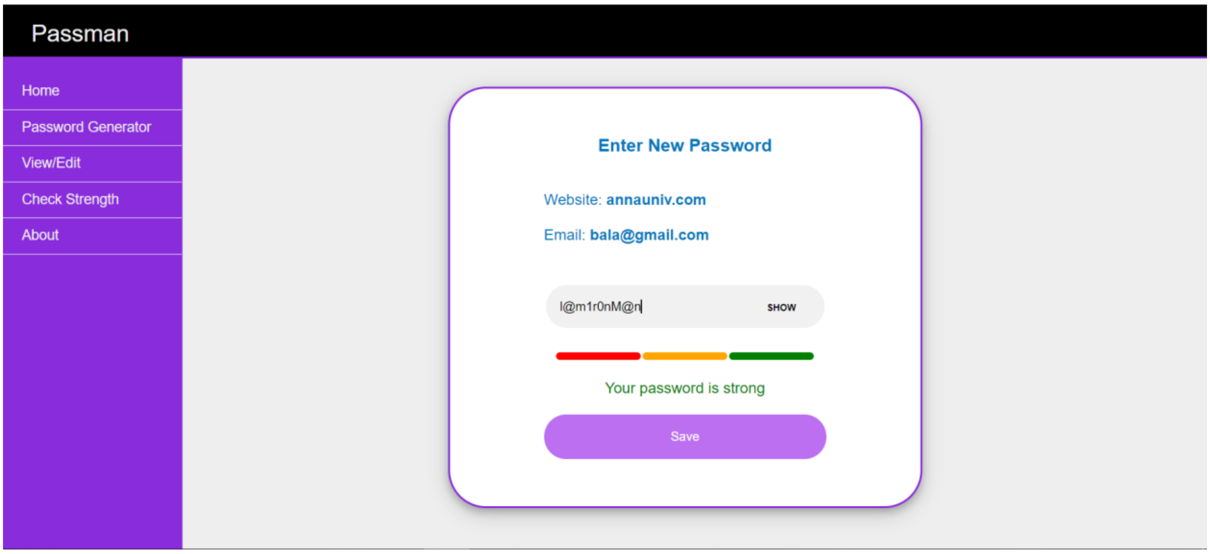
Editing user data for annauniv.com:



Website	Email	Password	Edit	Delete
annauniv.com	bala@gmail.com	HeLL0W0r!d	Edit	Delete
google.com	bala@gmail.com	Zq1rElED17tEw	Edit	Delete



```
        <?php foreach ($websites as $website) : ?>
    <tbody>
        <tr>
            <td><?php echo $website["website"]; ?></td>
            <td><?php echo $website["email"]; ?></td>
            <td><?php echo $website["pwd"]; ?></td>
            <td><button class="newbtn" onclick="location.href='editpass.php?sentwebsite=<?php echo $website
            ['website'] ?>&sentemail=<?php echo $website['email']; ?>'">Edit</button></td>
            <td><button class="newbtn" onclick="location.href='delete.php?sentwebsite=<?php echo $website
            ['website'] ?>&sentemail=<?php echo $website['email']; ?>'">Delete</button></td>
        </tr>
    </tbody>
<?php endforeach; ?>
```



Enter New Password

Website: **annauniv.com**

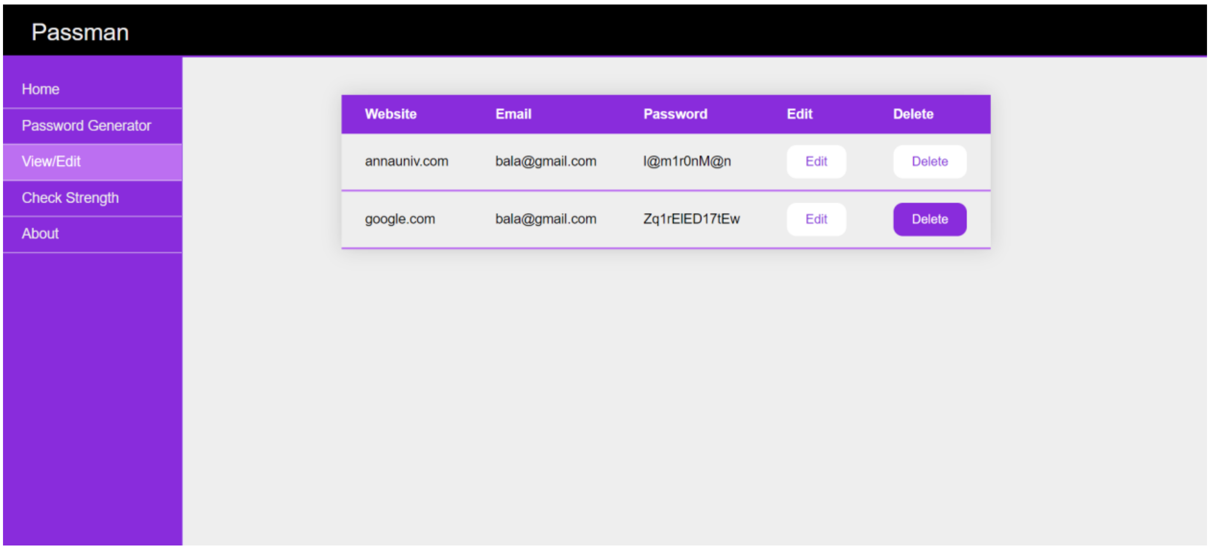
Email: **bala@gmail.com**

I@m1r0nM@n SHOW

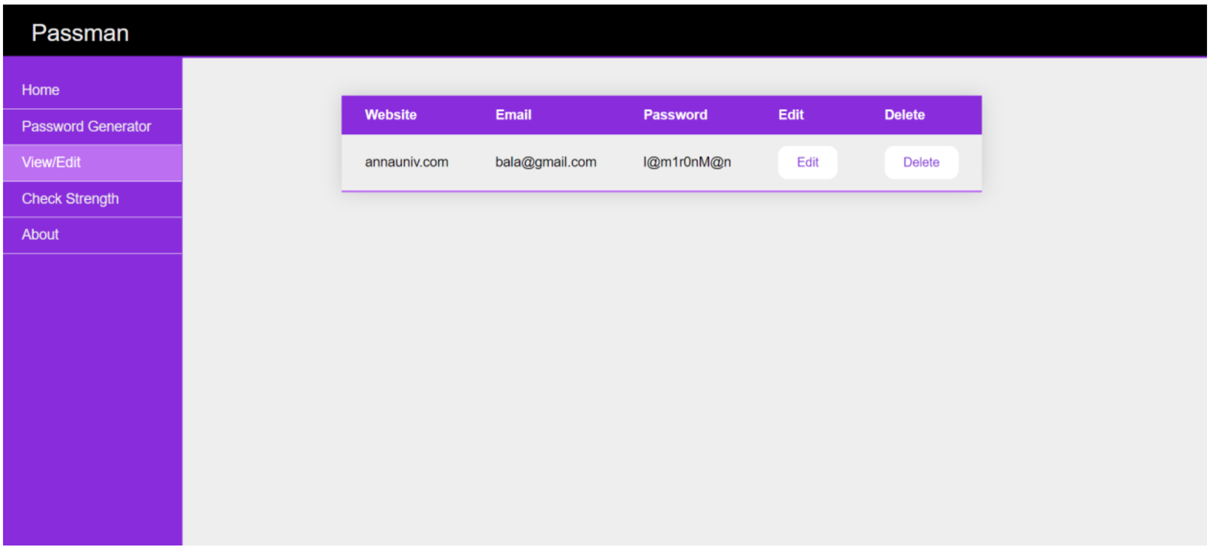
Your password is strong

Save

Deleting user data for google.com:



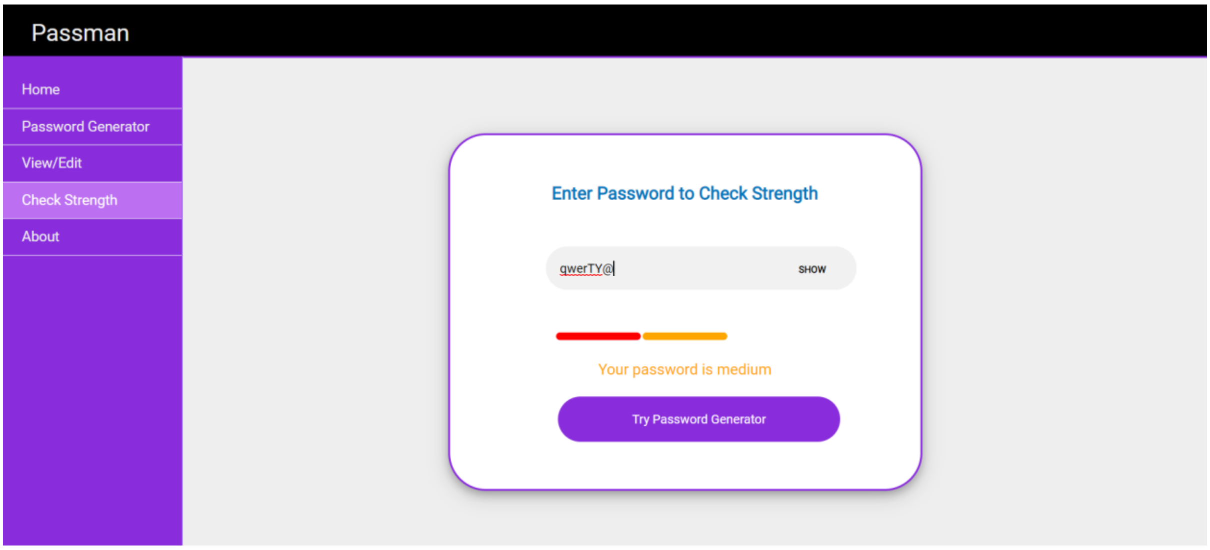
Website	Email	Password	Edit	Delete
annauniv.com	bala@gmail.com	l@m1r0nM@n	Edit	Delete
google.com	bala@gmail.com	Zq1rElED17tEw	Edit	Delete



Website	Email	Password	Edit	Delete
annauniv.com	bala@gmail.com	l@m1r0nM@n	Edit	Delete

5.Check Strength: checks the strength of the password and displays it to the user

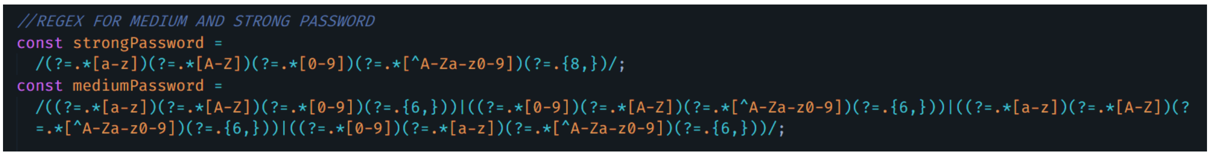
- On clicking check strength on the side navigation bar, user is taken to the check strength page.
- User enters the password and strength will be checked and displayed to the user



Enter Password to Check Strength

qwerTY@

Your password is medium



```
//REGEX FOR MEDIUM AND STRONG PASSWORD
const strongPassword =
  /(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[^A-Za-z0-9])(?=.{8,})/;
const mediumPassword =
  /((?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.{6,}))|((?=.*[0-9])(?=.*[A-Z])(?=.*[^A-Za-z0-9])(?=.{6,}))|((?=.*[a-z])(?=.*[A-Z])(?
  =.*[^A-Za-z0-9])(?=.{6,}))|((?=.*[0-9])(?=.*[a-z])(?=.*[^A-Za-z0-9])(?=.{6,}))/;
```