



Dokumentation LB 245

Julian Zurkinden

Datum	Version	Autor	Bemerkung
31.03	1.0	Zurkinden	
01.04	2.0	Zurkinden	

Inhaltsverzeichnis

Problembeschreibung.....	3
Anforderungen Prototyp	4
Art und Vorgehensweise Prototyping.....	5
Resultat	6
Projekt	6
Demo Video	6
Wichtige Codeauschnitte	7
Fazit/ Reflexion	9

Problembeschreibung

Die heutige digitale Gesellschaft ist von Passwortsicherheit und Datenschutz geprägt. Viele Nutzer verwenden jedoch unsichere Passwörter oder wiederholen dieselben Passwörter auf verschiedenen Plattformen, was ihre Sicherheit gefährdet. Darum ist es notwendig, eine Anwendung zu entwickeln, die Nutzern hilft, sichere Passwörter zu erstellen und ihre Sicherheit zu überprüfen. Mit solch einer Anwendung könnte man viele geklaute Daten verhindern und Leute besser vor der Kriminalität Online schützen.

Anforderungen Prototyp

Nummer	Muss/Kann	Funktional, Qualität, Rand	Beschreibung
1	Muss	Funktional	Das Programm kann das eingegebene Passwort überprüfen und daraufhin angeben, ob das Passwort sicher ist oder nicht.
2	Muss	Funktional	Ich als Nutzer habe die Möglichkeit per Knopfdruck das eingegebene Passwort umwandeln zu lassen.
3	Muss	Funktional	Das Programm kann das eingegebene Passwort mit einer Passwort liste abgleichen und dem Nutzer mitteilen, ob es geleakt wurde oder nicht.
4	Kann	Qualität	Das Programm erkennt das eingegebene Passwort und kann ausgeben, wie lange ein Computer benötigen würde, um das Passwort mittels Brute-Force-Angriff zu knacken. Diese Funktion soll auf einer hypothetischen Formel basieren.
5	Kann	Qualität	Der Benutzer kann auf einen Button klicken, bei dem weitere Tipps für einen sicheren Umgang mit Passwörtern angezeigt werden.

Art und Vorgehensweise Prototyping

Auswahl der Technologie:

Ich habe mich für eine Webapplikation entschieden da ich mein Produkt möglichst einer grossen Menge an Personen möglichst einfach zur Verfügung stellen möchte.

Anforderungen:

Ich habe mir als nächstes darüber Gedanken gemacht, was mein Prototyp braucht um den nutzen zu überliefern der dann das Fertige Produkt ebenfalls haben sollte. Nach Absprache mit der Lehrperson habe ich mich dann auf die Anforderungen festgelegt.

Entwickeln:

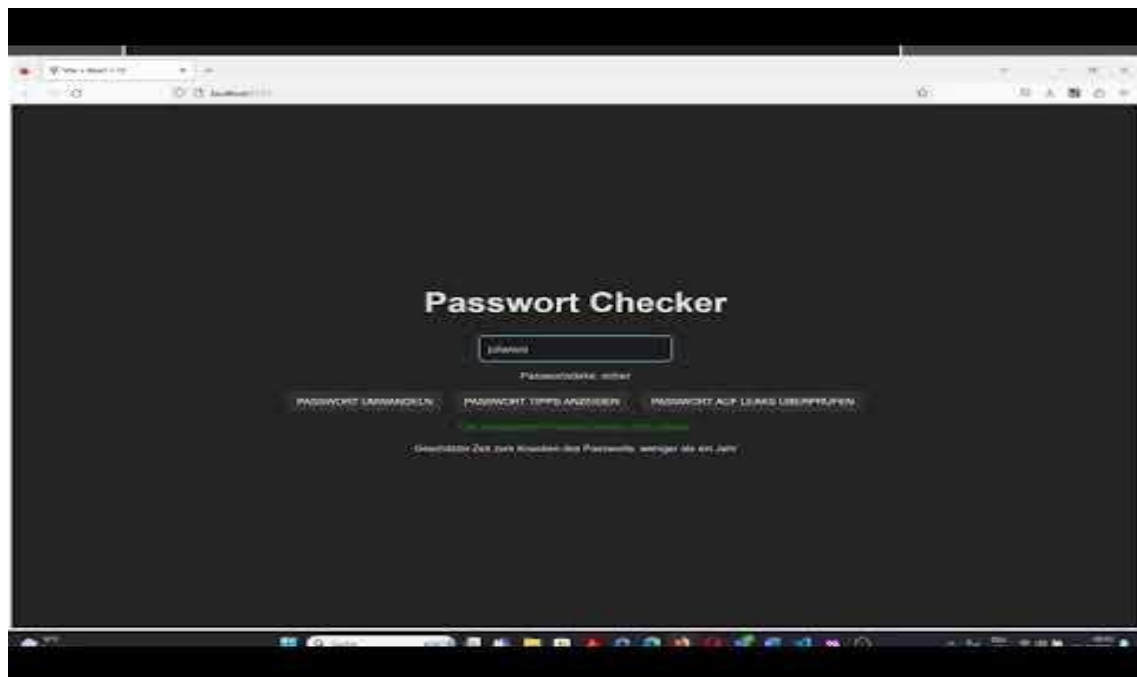
Ich habe sofort mit dem Entwerfen des Prototyps begonnen. Dabei habe ich bewusst Technologien verwendet, die ich bereits zuvor eingesetzt habe, um den Prototypen so effizient wie möglich zu entwickeln. Zuerst habe ich mich auf die Grundfunktionen und die Struktur meiner Anwendung konzentriert und von dort aus weiterentwickelt, bis mein Prototyp funktionsfähig war und alle Anforderungen abdeckte.

Resultat

Projekt

<https://github.com/julian05z/PasswortKontrolle>

Demo Video



Wichtige Codeauschnitte

Berechnung:

Im folgenden Codeausschnitt sieht man die **Berechnung** für die Dauer, um das Passwort zu knacken. Es handelt sich um eine Hypothetische Rechnung da man das so genau nicht sagen kann.

```
const calculateCrackTime = (password: string): string => {  
  
  // Annahme: Ein Angreifer kann 10 Milliarden Passwörter pro Sekunde  
  ausprobieren  
  const guessesPerSecond = 10_000_000_000;  
  
  const passwordLength = password.length;  
  
  const characterSetSize = 94;  
  
  // Berechnung der Anzahl der Zeichen in jedem Zeichensatz  
  const lowerCaseLetters = /[a-z]/.test(password) ? 26 : 0;  
  const upperCaseLetters = /[A-Z]/.test(password) ? 26 : 0;  
  const numbers = /\d/.test(password) ? 10 : 0;  
  const specialCharacters = /^[^a-zA-Z0-9]/.test(password) ? 32 : 0;  
  
  // Berechnung der Grösse des verwendeten Zeichensatzes  
  const usedCharacterSetSize =  
    lowerCaseLetters + upperCaseLetters + numbers + specialCharacters;  
  
  // Berechnung der möglichen Kombinationen  
  const possibilities =  
    BigInt(usedCharacterSetSize) ** BigInt(passwordLength);  
  
  const timeInSeconds = possibilities / BigInt(guessesPerSecond);  
  
  const timeInYears = Number(timeInSeconds) / (60 * 60 * 24 * 365);  
  
  return timeInYears >= 1  
    ? `${timeInYears.toFixed(2)} Jahre`  
    : `weniger als ein Jahr`;  
};
```

Umwandlung:

Hier wird das eingegebene Password umgewandelt und nach einem gewissen Schema Buchstaben ersetzt.

```
const transformPassword = (password: string): string => {  
  // Hier wird das Passwort umgewandelt.  
  const transformedPassword = password  
    .replace(/s/g, "5")  
    .replace(/a/g, "q")  
    .replace(/i/g, "!");  
  return transformedPassword;  
};  
  
export default transformPassword;
```


Fazit/ Reflexion

Ich war seit Beginn des Moduls richtig gespannt auf die Entwicklung des Prototypen und bin mit meiner Arbeit sehr zufrieden. Ich finde, dass mir der Prototyp besonders gut gelungen ist. Da ich bereits früh eine klare Vorstellung davon hatte, wie meine Anwendung aussehen sollte, fand ich problemlos einen Einstieg in die Realisierung. Schwierigkeiten traten erst bei der Berechnung der Dauer für das Knacken eines Passworts auf. Obwohl es im Internet Angaben dazu gibt, wie man diese Berechnungen anstellt, fiel es mir schwer, diese in mein Programm zu implementieren. Mit der Unterstützung von ChatGPT gelang es mir jedoch, eine geeignete Formel zu implementieren. Im weiteren Verlauf gab es keine grösseren Schwierigkeiten mehr, und ich konnte den Prototypen in kurzer Zeit fertigstellen. Für die Zukunft nehme ich mir vor, den Prototypen zu einem vollständigen Programm auszubauen, da ich Spass an der Entwicklung der Idee hatte und denke, dass es ein spannendes Projekt wäre, um kontinuierlich weiterzuentwickeln.