

---

# VERSICHERUNGSRECHNER

---

Giacun Masüger



10. MÄRZ 2023

BBW & KBW

# Inhaltsverzeichnis

|  |           |
|--|-----------|
| <b>Recherche im Web .....</b>                        | <b>2</b>  |
| <b>Allianz.ch .....</b>                              | <b>2</b>  |
| <b>Axa.ch .....</b>                                  | <b>2</b>  |
| <b>Migros-versicherungen.ch .....</b>                | <b>3</b>  |
| <b>Helvetia.com .....</b>                            | <b>3</b>  |
| <b>Zeitplan .....</b>                                | <b>4</b>  |
| <b>GUI-Ideen .....</b>                               | <b>4</b>  |
| <b>Variante 1:.....</b>                              | <b>4</b>  |
| <b>Variante 2:.....</b>                              | <b>5</b>  |
| <b>Das Problem-Statement: .....</b>                  | <b>5</b>  |
| <b>Die User-Story:.....</b>                          | <b>6</b>  |
| <b>Eingabefelder Versicherungssummenmodell .....</b> | <b>7</b>  |
| <b>Eingabefelder Quadratmetermodell .....</b>        | <b>9</b>  |
| <b>Dark und Light Mode .....</b>                     | <b>11</b> |
| <b>Das Bild.....</b>                                 | <b>12</b> |
| <b>Reflexion .....</b>                               | <b>13</b> |

## Recherche im Web

Allianz.ch

Ich möchte gerne eine Hausratversicherung abschliessen.

Ich bin 35 Jahre alt und wohne als Mieter in 3600 Thun.

Der Haushalt umfasst 2 Personen sowie 3,5 Zimmer.

[ANGEBOTE ANZEIGEN](#)

[→ LETZTE BERECHNUNG LADEN](#)

Axa.ch

### Hausratversicherung

|                  |              |           |          |
|------------------|--------------|-----------|----------|
| <b>1</b> Angaben | 2 Leistungen | 3 Adresse | 4 Fertig |
|------------------|--------------|-----------|----------|

**Gerne helfe ich Ihnen bei der Prämienberechnung. Dafür brauche ich ein paar Angaben.**



LOS GEHT'S



## Prämienrechner



Welche Versicherung möchten Sie? ⓘ

Hausrat & Haftpflicht

Nur Hausrat

Nur Privathaftpflicht

Helvetia.com

### 1. Versicherungsdetails

**CHF 186.00**

Selbstbehalt  
CHF 1'000



Versicherungssumme  
100'000



CHF

oder

Versicherungssumme berechnen

## Zeitplan

Ich werde mich im Laufe dieses Projekts an das Drehbuch halten und falls es dennoch Punkte gibt, bei denen ich nicht ganz fertig wurde, werde ich dies zuhause nachholen.

## GUI-Ideen

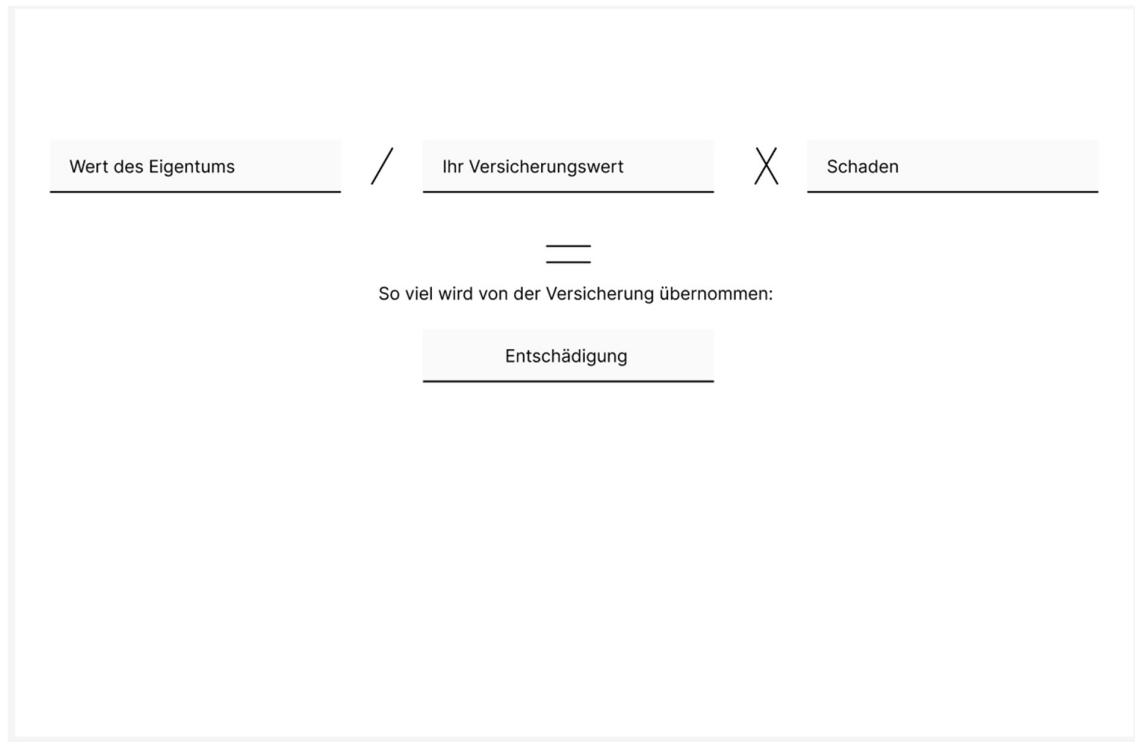
### Variante 1:

The mockup shows a dark gray background with a white toggle switch in the top right corner. Below it, there are four light gray input fields arranged in a 2x2 grid. Each field has a question mark icon in its top right corner. The text inside the fields is as follows:

|                                    |                               |
|------------------------------------|-------------------------------|
| Wert des Eigentums                 | Wie viel zahlen Sie pro Monat |
| Wie viel ist Ihr Versicherungswert | Entschädigung                 |

Below the input fields is a horizontal white line. Underneath this line, on the left, is a white downward arrow icon followed by the text "Anderer Rechner".

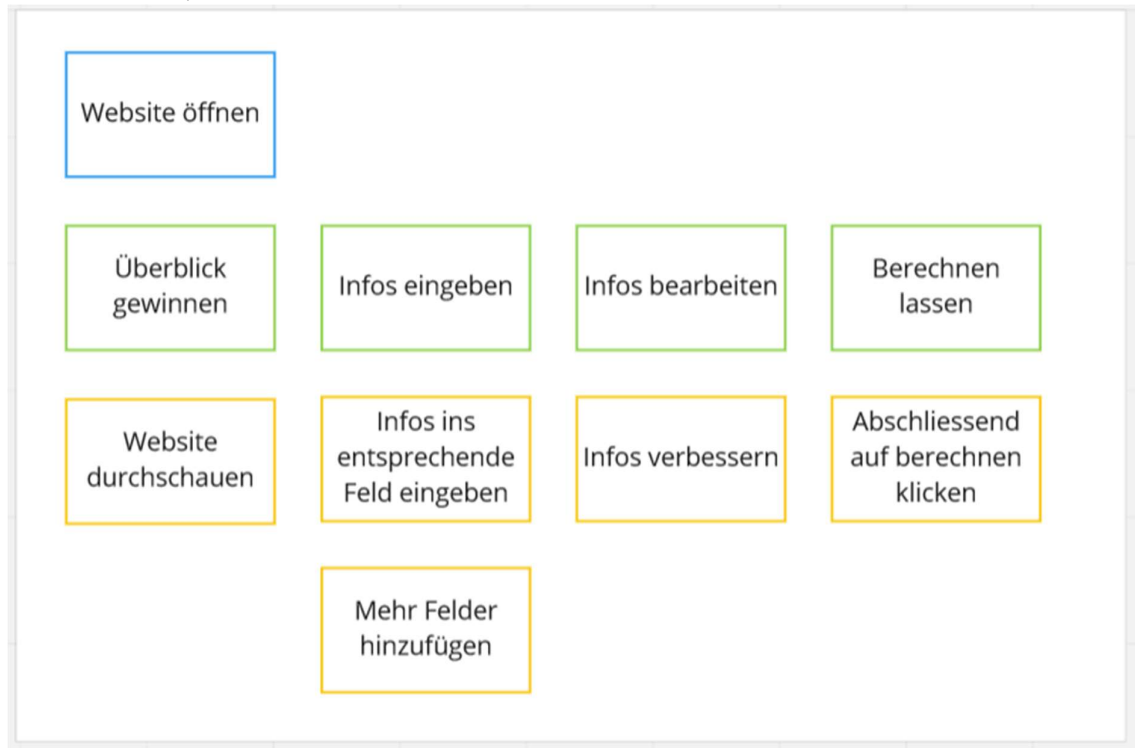
## Variante 2:



## Das Problem-Statement:

| Nutzer                                    | Problem   | Lösungsansätze  | Metriken  | Stakeholder                   | Randbedingungen              | Risiken  |
|---|---|---|---|-------------------------------|------------------------------|--|
| Der Kunde nutzt die Website auf dem Handy | Der Kunde sieht nicht alles (Eingabefelder)           | Die Website für Handy optimieren                        | Die Optimierung für jedes Gerät                               | Designer                      | Kein neues Projekt aufsetzen | Die einzelnen Felder sind zu klein   |
| Der Kunde hat speziell Sachen             | Die Website bietet nicht genügend Felder an           | Individuell Felder hinzufügen                           | Der Kunde kann so viele weitere Felder wie er will hinzufügen | Entwickler                    | ?                            | Es könnte zu viele Felder haben, welche dann zu einem Absturz führen könnten |
| Nutzer hat keine Ahnung                   | Nutzer weiss nicht, für was die einzelnen Felder sind | Über jedem Eingabefeld ein Infozeichen mit Beschreibung | Für jedes Feld ein Infotext erstellen                         | Entwickler schreibt Hilfetext | So kurz wie möglich          | Der Kunde versteht es immer noch nicht                                       |

## Die User-Story:



## Eingabefelder Versicherungssummenmodell

Ich habe die Inputs für den ersten Versicherungsrechner implementiert und darauf immer geachtet, dass es mit meinem GUI-Plan übereinstimmt, dies wurde im CSS dann angepasst. Weiter habe ich die Eingabefelder in ein Form gepackt, welches bei der Ausführung eine Funktion aufruft. Die einzelnen Inputs übergeben die eingegebenen Werte der Funktion, welche danach überprüft.

```
<form onSubmit={Rechnen}>
  <span className='fragezeichen' title='Den Betrag, den Sie mit der Versicherung abgemacht haben.'?></span>
  <input type='number' placeholder='Versicherungssumme' value={vs} min='0' onChange={ (e) => setVs(e.target.value)} />
  <br />
  <span className='fragezeichen' title='Wie teuer wäre es, wenn Sie ihr gesamten Haushalt nochmals kaufen müssten?'?></span>
  <input type='number' placeholder='Versicherungswert' value={vw} min='0' onChange={ (e) => setVw(e.target.value)} />
  <span className='fragezeichen' title='Wie hoch ist der Schaden?'?></span>
  <input type='number' placeholder='Schaden' value={schaden} min='0' onChange={ (e) => setSchaden(e.target.value)} />
  <input type='submit' value='Berechnen' />
  <input type='button' value='Löschen' onClick={Clear}/>
  <br />
  <label className='ergebnis'>Die Versicherung übernimmt:&nbsp;
    <p>
      <output>{es}</output>
      &nbsp;&nbsp;&nbsp;CHF
    </p>
  </label>
  <label className='infos'>
    {info}
  </label>
  <br />
</form>
```

Dazu werden Informationen angegeben, so dass der Benutzer sieht, ob es sich bei seinen Angaben um eine Unter- oder Überversicherung handelt, wie auch ob alle Felder ausgefüllt sind.

# Versicherungsrechner als Versicherungssummenmodell

?

Versicherungssumme

?

Versicherungswert

?

Schaden

Berechnen

Löschen

Die Versicherung übernimmt: **0 CHF**

Bitte geben Sie alle Angaben an.



Hier habe ich die Funktion «Rechnen» programmiert:

```
const [vs, setVs] = useState(''); //vs = Versicherungssumme
const [vw, setVw] = useState(''); //vw = Versicherungswert
const [schaden, setSchaden] = useState('');
const [es, setEs] = useState(0); //es = Entschädigung
const [info, setInfo] = useState('');

const Rechnen = (event) => {
  if (vs === '' || vw === '' || schaden === '') {
    event.preventDefault();
    setInfo('Bitte geben Sie alle Angaben an.');
```

You, 13 hou

```
  } else {
    if (vs < vw) {
      event.preventDefault();
      setInfo('Sie haben eine Unterversicherung.');
```

setEs(vs / vw \* schaden);

```
    } else {
      if (vs > vw) {
        event.preventDefault();
        setInfo('Sie haben eine Überversicherung.');
```

setEs(schaden);

```
      } else {
        event.preventDefault();
        setEs(vs / vw * schaden);
        setInfo('');
```

}

```
      }
    }
  }
};
```

Die Funktion «Rechnen», überprüft, ob Werte angegeben wurden und berechnet, danach den Betrag, welche die Versicherung zahlt. Darüber hinaus, wird noch überprüft, ob es sich um eine Unter- oder Überversicherung handelt.

Mit der Funktion «Clear» werden alle Werte, die in den UseState Variablen gespeichert sind, zurückgesetzt.

```
function Clear() {
  setVs('');
  setVw('');
  setSchaden('');
  setEs(0);
  setInfo('');
}
```

## Eingabefelder Quadratmetermodell

Ich habe wie beim Versicherungssummenmodell die Inputfelder gleich angepasst, da es so ein einheitlicherer Anblick gibt.

# Versicherungsrechner als Quadratmetermodell

?

Versicherungssumme

?

Quadratmeteranzahl

Berechnen

Löschen

Die Versicherungssumme beträgt: **0 CHF**

Diesen Rechner kann man durch das Aufklappen eines Abschnittes anzeigen lassen:

Weitere Versicherungen öffnen

```
<div className="collapsible collaps_posi">  
  <div className="header" {...getToggleProps()}>  
    {isExpanded ? 'Weitere Versicherungen schliessen' : 'Weitere Versicherungen öffnen'}  
  </div>  
  <div {...getCollapseProps()}>  
    <div className="content">  
      <p className='titel'>  
        Versicherungsrechner als Quadratmetermodell  
      </p>  
      <form onSubmit={Rechnen}>  
        <span className='fragezeichen' title='Den Betrag, den Sie mit der Versicherung abgemacht haben.'?></span>  
        <input type='number' placeholder='Versicherungssumme' value={vs} min='0' onChange={ (e) => setVs(e.target.value)} />  
        <br />  
        <span className='fragezeichen' title='Wie viele Quadratmeter Wohnfläche haben Sie?'?></span>  
        <input type='number' placeholder='Quadratmeteranzahl' value={vw} min='0' onChange={ (e) => setVw(e.target.value)} />  
        <input type='submit' value='Berechnen' className='submitButton' />  
        <input type='button' value='Löschen' onClick={Clear} />  
        <br />  
        <label className='ergebnis'>Die Versicherungssumme beträgt:&nbsp;&nbsp;&nbsp;<br>  
          <p>  
            <output>{be}</output>  
            &nbsp;&nbsp;&CHF&nbsp;&nbsp;&  
          </p>  
        </label>  
        <label className='infos'>  
          {info}  
        </label>  
      </form>  
    </div>  
  </div>  
</div>
```

Des Weiteren braucht es bei diesem Modell nicht drei sondern nur zwei Inputs, welche dann multipliziert und ausgegeben werden.

```
const { getCollapseProps, getToggleProps, isExpanded } = useCollapse();
const [vs, setVs] = useState(''); //vs = Versicherungssumme
const [vw, setVw] = useState(''); //vw = Versicherungswert
const [be, setBe] = useState(0); //be = Betrag
const [info, setInfo] = useState('');

const Rechnen = (event) => {
  if (vs === '' || vw === '') {
    event.preventDefault();
    setInfo('Bitte geben Sie alle Angaben an.');
```

Da dieses Aufklappen aufgerufen werden muss, ist dies auch wieder eine Funktion.

```
function Collapsible() {
```

Zu Guter Letzt wird diese Komponente dann exportiert:

```
export default function Quadra() {
  return (
    <Collapsible/>
  );
}
```

## Dark und Light Mode



Mit diesem Knopf ist es dem Benutzer erlaubt, die Website im Light-Mode oder Dark-Mode anzusehen. Diese Feature wurde vor allem aus Interesse komplett selbst entwickelt, was es wahrscheinlich etwas unprofessionell macht.

```
export default function Mode({callback}) {
  const [text, setText] = useState('Light');
  const [hintergrund, setHintergrund] = useState('');

  function Dtol() { //Dark to Light und Text
    //Text
    if (text == 'Light') {
      setText('Dark');
      //Farbe
      setHintergrund('light');
      callback(hintergrund);
    } else {
      setText('Light');
      //Farbe
      setHintergrund('dark');
      callback(hintergrund);
    }
  }

  // Damit die Farbe beim ersten Mal gändert wird
  useEffect(() => {
    callback(hintergrund);
  }, [hintergrund, callback]);

  return (
    <button className='dark_light' onClick={Dtol}>{text}</button>
  );
}
```

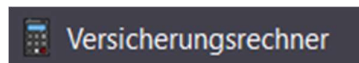
Beim Aktivieren dieser Funktion wird der ClassName der obersten Komponente verändert und so wird die Textfarbe und Hintergrundfarbe verändert. Dies kann mithilfe von einem Callback gemacht werden.

```
const [hintergrund, setHintergrund] = useState('rgb(63, 65, 79)');

return (
  <div className={hintergrund}>
    <Mode callback={setHintergrund}/>
    <Eingabefelder />
    <Bilder />
    <Quadra />
    <Footer />
  </div>
);
}
```

## Das Bild

Ganz am Schluss habe ich noch das Favicon und das Bild zur Website hinzugefügt.



## Reflexion

Das Projekt zu entwickeln, zu planen und Ideen zu sammeln, war eine spassige Abwechslung und vor allem, dass es in Docker funktioniert hat, war richtig toll. Ich konnte leider nicht ganz alles erreichen, was mein Ziel war, aber im Grossen und Ganzen konnte ich, in meinen Augen, das Projekt zeitlich erfolgreich abschliessen.