Spezifikation und Design

Projektziele

Es soll eine Weboberfläche zur Darstellung aller Filme, die im Kino laufen implementiert werden. Bei dieser Implementation sollen sich die Besucher registrieren können und somit Karten über die Website kaufen können. Der Administrator soll sowohl die Filme als auch die Kategorien verwalten können.

Personas

Rebekka Holzwurm

- 38-jährige Mutter
- 2 Kinder
- Bürofachangestellte
- Organisiert und strukturiert
- Vorliebe für Unterhaltungsmedien
- Effiziente Planerin
- Mag keine Warteschlangen

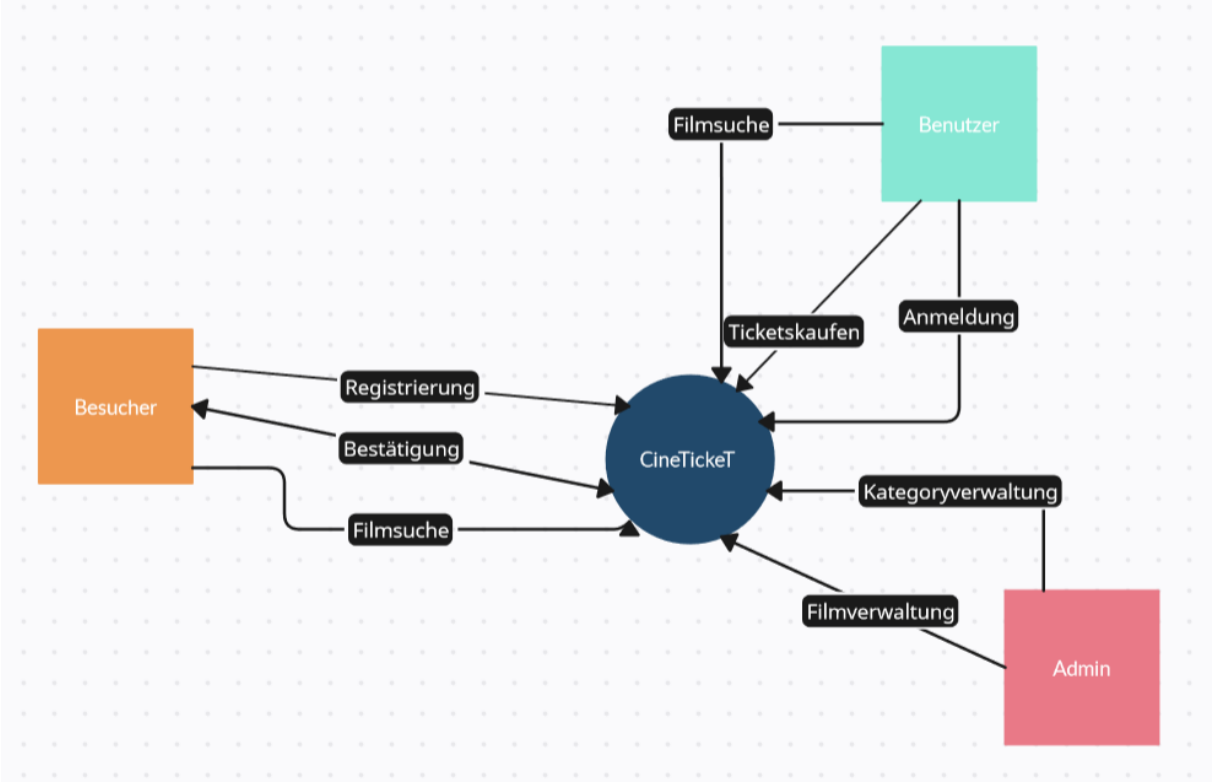
Karl Schneider

- 20-jähriger Wirtschaftsstudent
- Ist gerne unter Freunden
- Bestellt regelmäßig in Onlineshops
- Ist bekannt als der Organisator
- Zeit ist Geld

Gerhard Mönchhause

- 75-jähriger Opa
- Informiert sich gerne mit der Zeitung am morgen
- Enkel Peter hat ihm ein Smartphone zum 75. Geburtstag geschenkt
- In der Ruhe liegt die Kraft
- Angelt gerne
- Möchte seine Enkelkinder beeindrucken

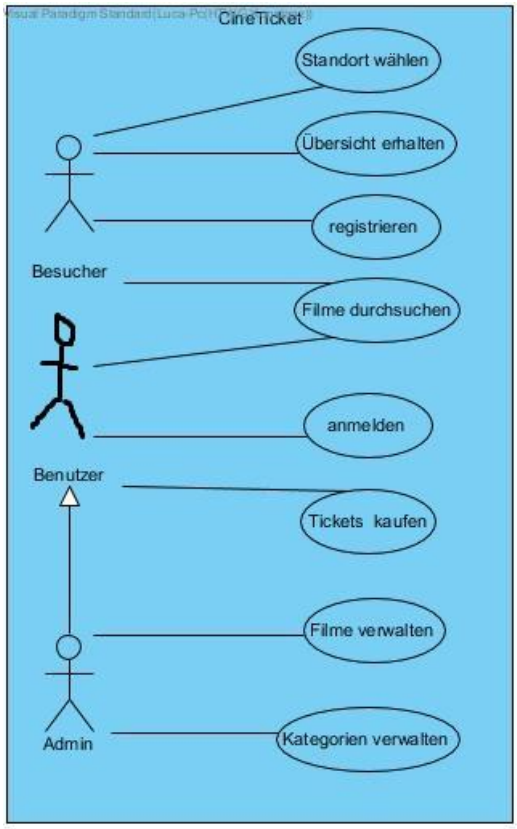
Kontextdiagramm:



Akteure:

Besucher	Dieser möchte einen Überblick über die angebotenen Filme erhalten. Dieser möchte sich über anstehende Filme informieren können.
Registrierter Benutzer	Dieser möchte Tickets kaufen können.
Administrator	Dieser kann Filme bearbeiten.

Use-Case-Diagramm:



Use-Case-Spezifikationen

- Name: Übersicht erhalten
- Zusammenfassung: Die Übersicht über alle Veranstaltungen wird dargestellt.
- Akteur: Besucher
- Auslösendes Ereignis: Besuchen der Website

- Name: registrieren
- Zusammenfassung: Unter Angabe von persönlichen Daten findet eine Registrierung statt.
- Akteur: Besucher
- Auslösendes Ereignis: Sign-Up Button drücken

- Name: Standort wählen
- Zusammenfassung: Den gewünschten Standort auswählen
- Akteur: Besucher
- Auslösendes Ereignis: Standort-Knopf drücken

- Name: anmelden
- Zusammenfassung: Die Login-Daten werden eingegeben.
- Akteur: Benutzer
- Auslösendes Ereignis: Log-In Button drücken

- Name: Ticket kaufen
- Zusammenfassung: Sitzplätze für einen spezifischen Film werden gekauft.
- Akteur: Benutzer
- Auslösendes Ereignis: Tickets kaufen Button drücken

- Name: Filme verwalten
- Zusammenfassung: Die Filme werden geändert.
- Akteur: Admin
- Auslösendes Ereignis: Film ändern Button drücken

- Name: Kategorien verwalten
- Zusammenfassung: Die Kategorien können angepasst werden
- Akteur: Admin
- Auslösendes Ereignis: Kategorie ändern Button drücken

- Name: Filme durchsuchen
- Zusammenfassung: Filme werden nach Suchbegriffen sortiert.
- Akteur: Besucher, Benutzer, Admin
- Auslösendes Ereignis: In Suchfeld tippen

Design

Moodboard:

Moodboard:

Style Sheet:
https://coolors.co/2f6690-3a7ca5-d9dcd6-16425b-81c3d7

CineTicket Style Tyle:Version 1

Possible Colors:

This is an Example of a Header

Font Name: "Helvetica Neue", Helvetica, Arial, sans-serif # 16425B

This is an Example of a Sub Head

Font Name: "Helvetica Neue", Helvetica, Arial, sans-serif #00000

Willkommen bei CineTicket! Im CineTicket können Sie schon jetzt Ihr Erlebnis des Jahres buchen. Gestalten Sie ihren unvergesslichen Aufenthalt schon heute ganz bequem zu Hause vom Sofa. Für Spaß von Groß und Klein ist garantiert.

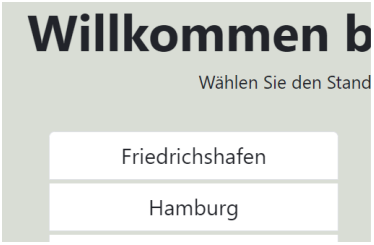
Font Name: "Helvetica Neue", Helvetica, Arial, sans-serif #000000

This is an example of a Text link

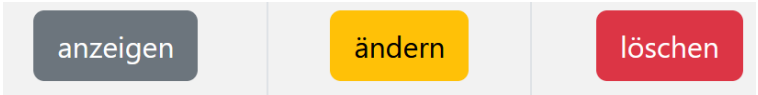
Adjective

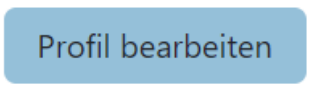
Freundlich Strukturiert Effektiv

Possible Patterns:



Possible Buttons:





Page Structure:

Logo Film Suchen Adminbereich Login

Bild

Wilkommen bei CineTicket!

Wählen Sie den Standort Ihres Kinos aus:

Auswahl 1

Auswahl 2

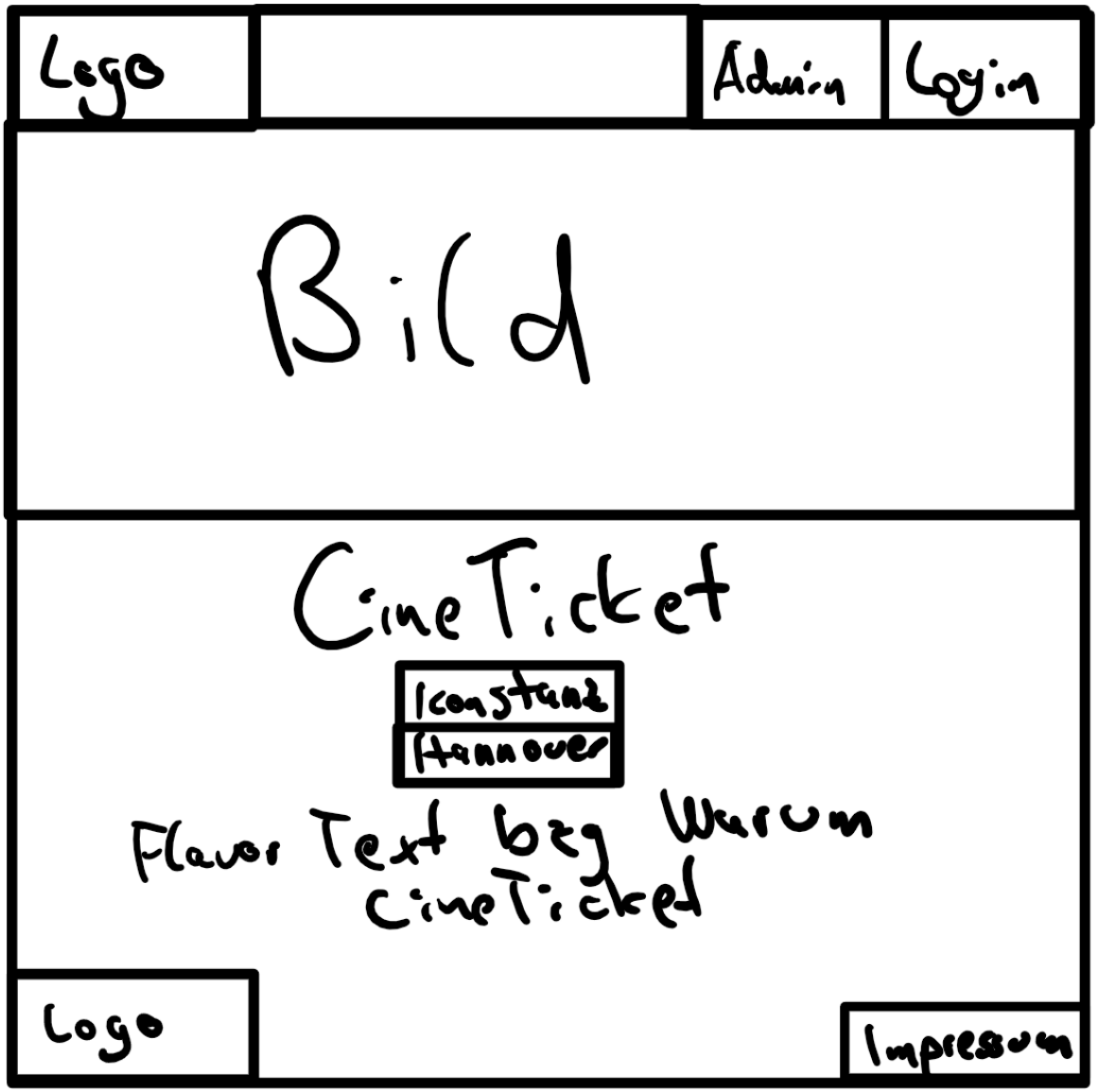
Auswahl 3

Warum CineTicket?

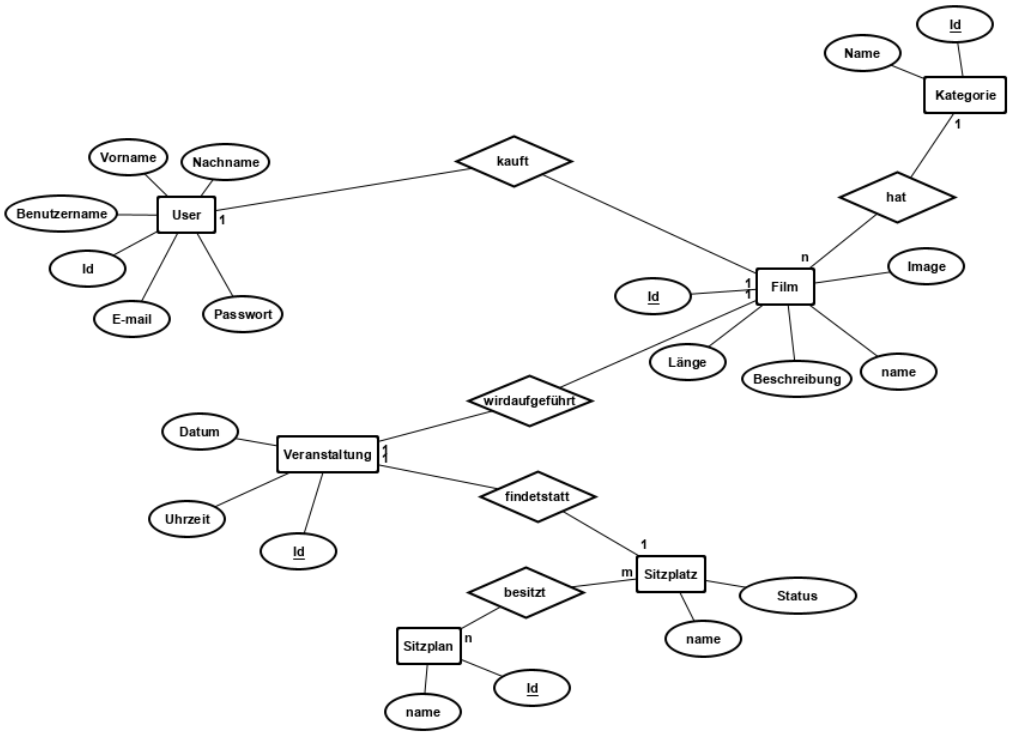
Flavor Text

Logo Impressum

Low-Fi-Mockup



ER-Diagramm:



https://www.figma.com/file/hGyJztABpdsqPU8narFKbd/CineTicket?type=design&node-id=0%3A1&mode=design&t=e3H3j4P8dA1zyxwM-1

Projekt live auf :
https://cineticket-github.onrender.com/

Admin Login:
test@test.de pw: test123

Normal Login:

abc@abc.de pw:abc

Implementierung

Für die Kategorie Verwaltung wurde ein Controller erstellt, der alle Standardoperationslogiken enthält:

z.B.

```
updateOne: async function (req, res) {
    sails.log.debug("Update single Category....")
    try {
        let category = await Category.updateOne({ id: req.params.id
}).set(req.body);
        sails.log.debug("Updated category:", category);
        res.view('pages/category/show', { category: category });
    } catch (err) {
        sails.log.error("Error updating category:", err);
        res.serverError(err);
    }
```

Passend dazu wurde die jeweiligen .ejs Dateien erstellt die das bearbeiten auf der Frontendseite möglich machen

Für die Film Verwaltung wurde ein Controller erstellt, der alle Standardoperationslogiken enthält:

```
destroy: async function (req, res) {
    let movie = await Movie.destroyOne({ id: req.params.id });
    res.redirect('/movie');
  },
```

Für die Möglichkeit zur Registrierung und Anmelden von Besucher und Benutzern wurde die Logik, die von Ihnen bereitgestellt wurde kopiert.

Für die Suche von Filmen wurde die Funktion find hinzugefügt:

```
find: async function (req, res) {
    let movies;
    let categories = await Category.find();

    if (req.query.q && req.query.q.length > 0) {
      if (req.query.catQ > 0) {
        movies = await Movie.find({
          name: {
            'contains': req.query.q
          },
          category: req.query.catQ
        }).populate('category');
      } else {
        movies = await Movie.find({
          name: {
            'contains': req.query.q
          }
```

```
        }).populate('category')
      }
    }
    else if (req.query.catQ > 0) {
      movies = await Movie.find({
        category: req.query.catQ
      }).populate('category');
    }
    else {
      movies = await Movie.find().populate('category');
    }
    res.view('pages/search', { movies: movies, categories: categories });
  },
```

Für das Kaufen von Kinokarten mit der dazugehörigen Sitzplatzauswahl wurde eine neue Vue-Komponente erzeugt, die Logik dafür findet sich im separaten SeatController:

```
updateSeat: async function (req, res) {
    try {
      console.log('Update Seat Action aufgerufen');
      const seatId = req.params.id;
      const updatedStatus = req.body.status;
      console.log('Seat ID:', seatId);
      console.log('Updated Status:', updatedStatus);

      const updatedSeat = await Seat.updateOne({ id: seatId }).set({ status:
updatedStatus });

      if (!updatedSeat) {
        return res.status(404).json({ error: 'Sitzplatz nicht gefunden.' });
      }

      return res.status(200).json(updatedSeat);
    } catch (error) {
      console.error(error);
      return res.status(500).json({ error: 'Interner Serverfehler' });
    }
  },
```

Diese und weitere Funktionen werden dann in den einzelnen Methoden der Vue-Komponente aufgerufen

Optimierung

SEO

Es wurden folgende Mittel eingesetzt, um die Website für die Suchmaschine zu optimieren:

1. Hinzufügen von einer Meta-Beschreibung

```
<meta name="description" content="Entdecken Sie das einzigartige CineTicket-Kinoreservierungssystem! Machen Sie Ihren Kinobesuch in Friedrichshafen, Hamburg, Hannover, Konstanz, München, Radolfzell, Singen oder Stuttgart stressfrei und aufregend. CineTicket ermöglicht bequeme Online-Ticketkäufe für jeden Filmfan, von Blockbustern bis zu zeitlosen Klassikern.">
```

2. Hinzufügen von Meta-Keywords(tags)

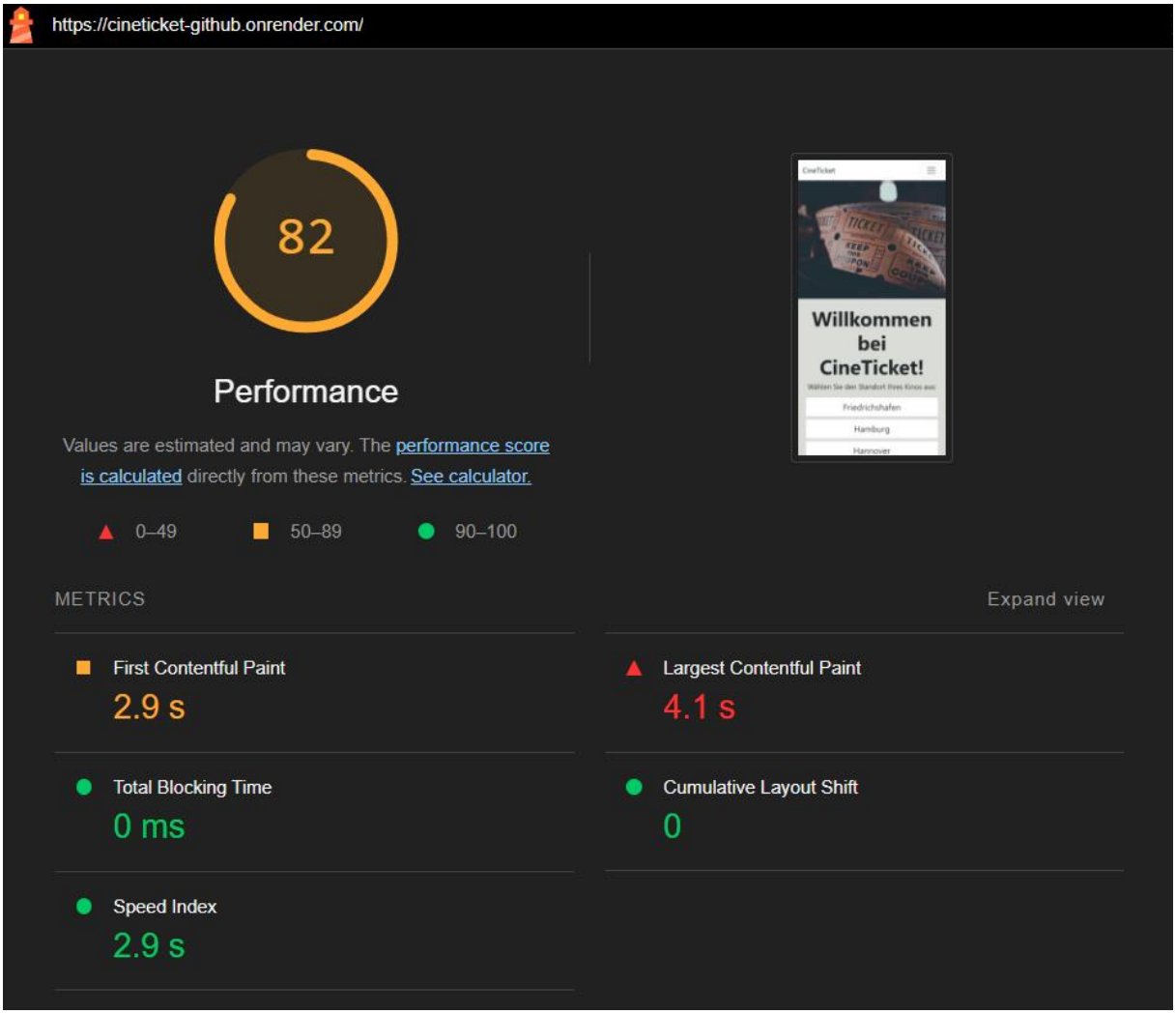
```
<meta name="keywords" content="CineTicket, Kino, Online-Ticketreservierung, Filmerlebnisse, Kinobesuch, Familie, Umgebung, Kinokarten">
```

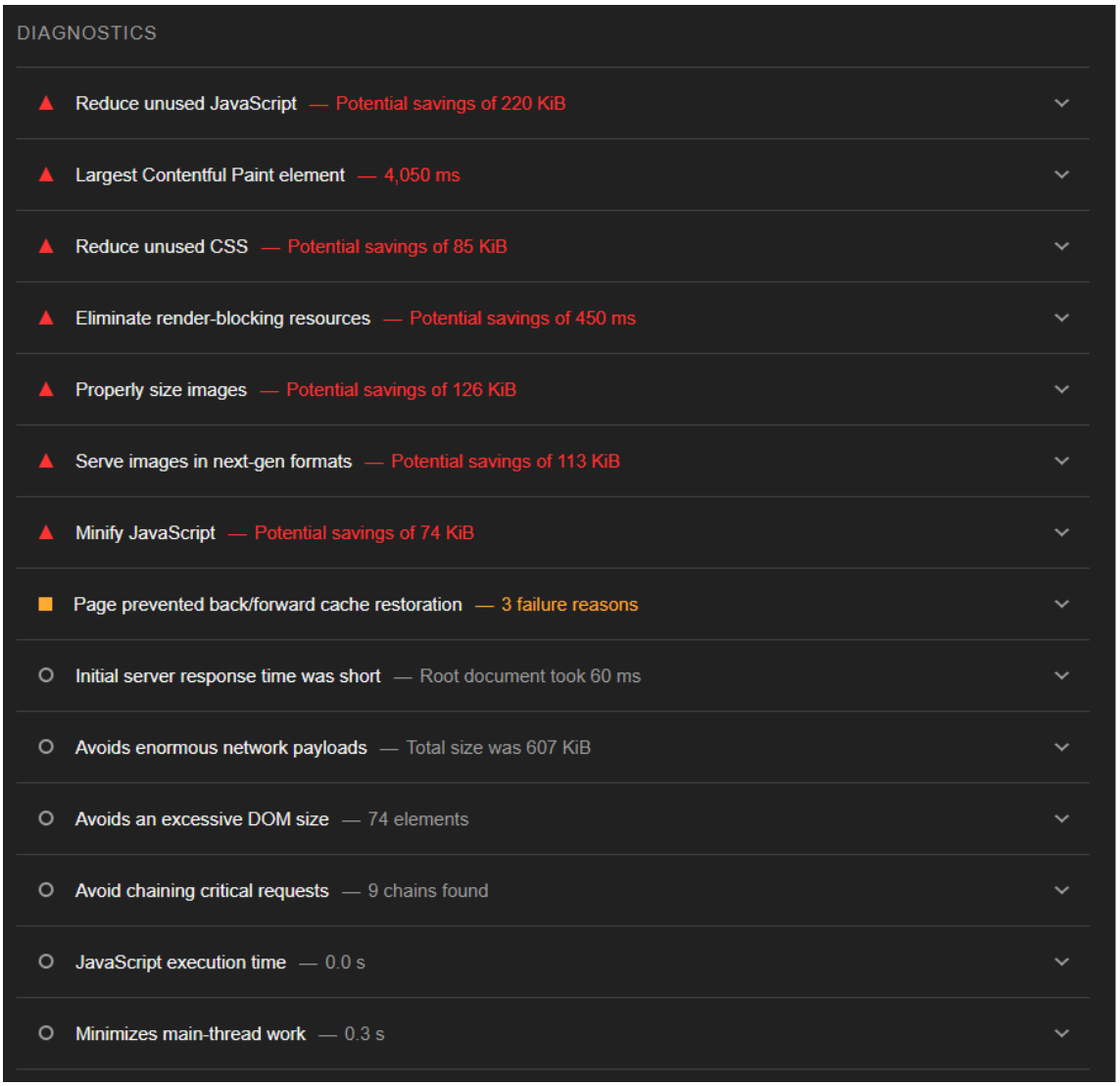
3. Anpassung der Überschriftenstruktur zu h1/h2 usw
4. Alt-Text für die Bilder auf der Mainpage hinzugefügt damit Suchmaschinen den Inhalt verstehen können
5. Einfügen von Schlüsselwörtern damit die Suchmaschine tags finden kann

```
<span class="keyword">CineTicket</span>
```

Performance

Report von Lighthouse (Lighthouse performance mobil):





Um die Largest Contentful Paint Elementzeit zu verbessern wurde dynamische Bild größer hinzugefügt, die das Laden schneller macht:

```
<img class="home-slideshow" srcset="images/home_ticket_320.jpg 320w,
images/home_ticket_640.jpg 640w,
images/home_ticket.jpg 1280w"
sizes="(max-width: 320px) 320px,
(max-width: 640px) 640px,
(max-width:1280) 1280px"
src="images/home_ticket.jpg" alt="CineTicket Kinokarten">
```

Weitere Optimierungsmaßnahmen wäre Sachen wie minifyjs auszuführen. Um auf einen Wert von 100 zu kommen, muss man hauptsächlich am Largest Contentful Paint sowie am First Contentful Paint arbeiten, alle anderen Metriken wie CLS oder SI oder TBT sind perfekt.