

Praktische Arbeit - Farbgenerator

Caroline Bengart, Anna-Lena Gassner, Aaron Kreuzer, Dieter Levin,
Fabian Meiler, Patrick Pohl und Harun Tacli

Abstract—Überblick über unser Projekt, in welchem wir eine Farbgenerator-Webanwendung programmieren wollen, mit den entsprechenden Inspirationen, Anforderungen und Technologien.

I. EINLEITUNG

Ziel unserer Projektarbeit ist die Entwicklung eines Farbgenerators. Die Bedienung und das Erstellen der Farbschemata erfolgen dabei intuitiv und ermöglichen damit sowohl Laien als auch professionellen Anwendern einen leichten Zugriff. Unsere Webanwendung besteht aus zwei Seiten, wobei die erste Seite die Möglichkeit bietet, Farbschemata zu suchen und diese auch zu bewerten. Auf der zweiten Seite wird die eigentliche Farbpalette erstellt, welche auch individuell angepasst werden kann. Zu jedem Schema gibt es auch eine Vorschau, in der das Schema auf eine kleine Beispiel-Webseite angewendet wird, um somit einen besseren Eindruck zu erhalten. Eine Kommentarsektion schließt unsere Webseite ab. Unser Farbgenerator hebt sich von der Masse ab, in dem er folgende 3 Punkte, die sonst nur einzeln aufzufinden sind, kombiniert:

- Intuitives Erstellen eines Schemas
- Bewertungssystem nach mehreren Kriterien
- Umfangreiche, realitätsnahe Vorschau.

II. VERWANDTE ARBEITEN

Unsere Übersichtsseite orientiert sich an der Seite colorhunt.co [1], da diese eine übersichtliche und saubere Möglichkeit bietet, mehrere Farben in zusammengefassten Schemata mit Like-Funktion anzeigen zu lassen. Außerdem können hier Paletten mit unterschiedlicher Anzahl an Farben dargestellt werden.

Der Color-Picker wird durch Adobe Color CC [2] inspiriert. Daran gefällt uns besonders die Aufteilung der Website in Color-Wheel und das dynamische Anzeigen der ausgewählten Farben im unteren Teil.

Ergänzt wird diese Ansicht durch eine kleine, abstrakte Website, welche die gewählten Farben live anzeigt. Dies soll dem Benutzer die Möglichkeit bieten, direkt beurteilen zu können, wie gut die gewählten Farben miteinander harmonieren. Die Idee ist an [w3schools](https://www.w3schools.com/colors/colors_compound.asp) [3] angelehnt, welche dieses Konzept unserer Meinung nach bereits sehr gut umgesetzt haben.

III. ANFORDERUNGEN

- Als Benutzer möchte ich eine Farbpalette erstellen, um das Zusammenspiel der Farben besser zu erkennen.

- Als Benutzer möchte ich die einzelnen Farben individuell ändern, um einen großen Gestaltungsfreiraum zu haben.
- Als Benutzer möchte ich die Anzahl der unterschiedlichen Farben ändern, um individuell auf mein Produkt eingehen zu können.
- Als Benutzer möchte ich eine Vorschau der Farbpalette angezeigt bekommen, um sofort zu sehen, wie die ausgewählten Farben miteinander harmonieren.
- Als Benutzer möchte ich meine erstellte Farbpalette speichern, um sie später wieder abrufen zu können.
- Als Benutzer möchte ich die einzelnen Farbwerte sehen, um sie in mein Projekt kopieren zu können.
- Als Benutzer möchte ich eine bereits von mir erstellte Farbe einfügen, um die Farbpalette an dieser zu orientieren.
- Als Benutzer möchte ich andere Farbpaletten sehen, um mich von diesen inspirieren zu lassen.
- Als Benutzer möchte ich Bewertungen der Farbpaletten sehen, um die besten Farbkombinationen zu finden.
- Als Benutzer möchte ich die Farbpaletten anderer bewerten, um Ihnen eine Einschätzung über ihre Farbkombination zu geben.
- Also Benutzer möchte ich ohne Registrierung eine Farbpalette erstellen, da ich gerne anonym bleiben möchte.

IV. METHODEN

Für unser Frontend verwenden wir React.js und im Backend Express.js mit MongoDB. Für die Kommunikation zwischen Client und Server verwenden wir eine REST-Schnittstelle. Hier werden die vom Benutzer erstellten Farbpaletten abgelegt mit den dazugehörigen Bewertungen und eventuell Kommentaren. Wir haben uns für den MERN-Stack (MongoDB, Express, React und Node) entschieden, da es weit verbreitet ist und es viele Ressourcen dazu gibt.

REFERENCES

- [1] <https://colorhunt.co>
- [2] <https://color.adobe.com/de/create/color-wheel>
- [3] https://www.w3schools.com/colors/colors_compound.asp